

## Association for Information Systems AIS Electronic Library (AISeL)

---

ACIS 2011 Proceedings

Australasian (ACIS)

---

2011

# Services for Business Processes in EA – Are They in Relation?

Lam-Son Lê

University of Wollongong, [lle@uow.edu.au](mailto:lle@uow.edu.au)

Follow this and additional works at: <http://aisel.aisnet.org/acis2011>

---

### Recommended Citation

Lê, Lam-Son, "Services for Business Processes in EA – Are They in Relation?" (2011). *ACIS 2011 Proceedings*. 4.  
<http://aisel.aisnet.org/acis2011/4>

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2011 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

## Services for Business Processes in EA – Are They in Relation?

Lam-Son Lê

School of Computer Science and Software Engineering  
Faculty of Informatics, University of Wollongong  
New South Wales 2522, Australia  
Email: [lle@uow.edu.au](mailto:lle@uow.edu.au)

### Abstract

*Business services arguably play a central role in service-based information systems as they would fill in the gap between the technicality of Service-Oriented Architecture and the business processes captured in Enterprise Architecture. Business services have distinctive features that are not typically observed in plain Web services. The representation of business services requires that we view human activity and human-mediated functionality through the lens of computing and systems engineering. We give insights into the modeling of business services and relationships between them. This work sheds light on the analysis, design and reusability of business-aware services that business owners, entrepreneurs and business architects alike would find useful when dealing with their service ecosystem.*

### Keywords

Service Ecosystem, Services Evolution, Business Services, Business Processes, SOA, Enterprise Architecture

### INTRODUCTION

In the last few years, service-oriented computing has become an emerging research topic in response to the shift from product-oriented economy to service-oriented economy. On the one hand, we now live in a growing services-based economy in which every product today has virtually a service component to it (Paulson, 2006). In this context, services are increasingly provided in different ways in order to meet growing customer demands. Business domains involving large and complex collection of loosely coupled services provided by autonomous enterprises are becoming increasingly prevalent (Singh & Huhns, 2005). On the other hand, Information Technology (IT) has now been thoroughly integrated into our daily life (Hansmann et al., 2011) and gradually gives rise to the paradigm of ubiquitous computing. As such, business services are essentially IT-enabled making the border between business services and IT-enabled services blurred. At the high-level operationalization of a business service, we see business activities happening between service stakeholders. We may or may not witness IT operations at this representational level. At lower levels, the operationalization of these services are eventually translated into IT operations as we have seen in the cases of banking services, recruitment services, library services, auctioning services, etc.

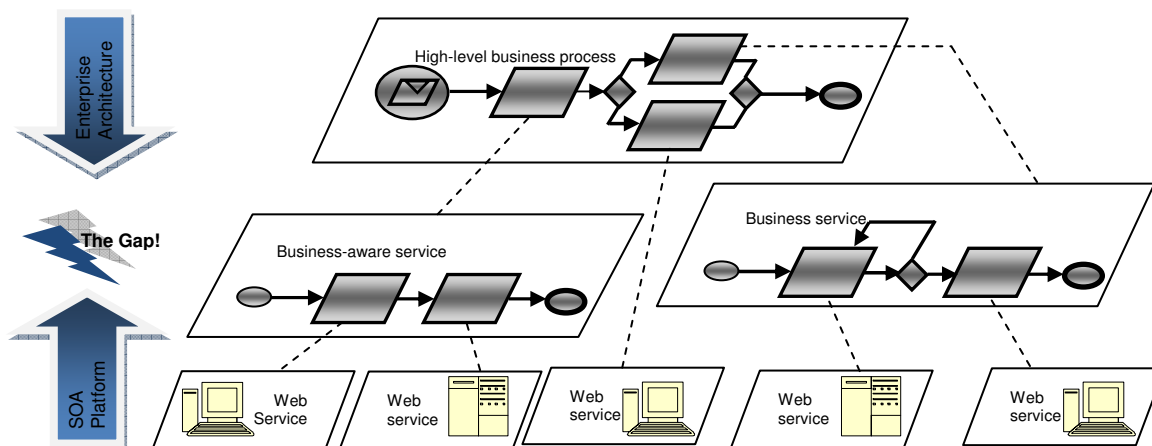


Figure 1. The gap between high-level business services in EA and Web services in SOA.

Let us take a look at the landscape of IT services from another angle - the connection of Service-Oriented Architecture (SOA) to Enterprise Architecture. The underlying rationale of SOA is to provide business-aware services at the granularity targeted at business cognizant rather than software development (Grigoriu, 2007). Unfortunately, the SOA has adopted the Web as its implementation technology, virtually making Web services

the primary elements in SOA. The consensus on this matter is that Web services provide a platform on which business processes shall be implemented using executable languages such as Business Process Execution Language (BPEL) (Sarang et al., 2006). Typically, the execution of business processes involves the invocation of Web services as a computerization of the business logic expressed in these processes. As such, IT services are generally considered as low-level, implementation-bound representation for the business logic. In reality, this BPEL-inspired standpoint may not be sufficient since executing business processes requires the involvement of not only Web services but also human-mediated actors, especially when we have messages exchanged between processes (Kloppmann et al., 2005). As such, the business logic of a single process is only partially computerized by means of Web services. Explicitly or implicitly, this computerization is jointly done with other, perhaps less business-oriented, processes that in turn might be computerized by means of their own Web services. In this way, business processes are enabled by non-Web business-aware services that serve as access points for business functions as well as IT capabilities (see Figure 1). This is an emerging topic called Service-Oriented Enterprise Architecture. We call these services *business IT-enabled services*, or simply business services.

Unfortunately, there has not been much work in modeling high-level services from a business perspective. The operationalization of business services have distinctive features that are not typically observed in plain Web services. Most notably, business services occur for a noticeable period of time, not spontaneously as Web services do. Their occurrences feature incremental human-mediated developments. As such, the representation of business services requires that we view human activity and human-mediated functionality through the lens of computing and systems engineering. In this paper, we propose how to model business services and relationship between them. This work sheds light on the analysis and design of business services that business owners, entrepreneurs and managers alike would find useful when dealing with their service ecosystem.

## OVERVIEW OF THE 3 X 3 X 3 MODELING APPROACH

To make the paper self-contained, we briefly present a multi-perspective representational grid that we base our work on (and also improved it) to reason about service relationships (Lê et al., 2010). This grid takes into consideration the fact that occurrences of business services usually involve multiple stakeholders and social issues. Typical stakeholders of a business service are service providers and service consumers. The providers and the consumers have different perspectives on a service they provide and consume, respectively. These two perspectives are the main sources of representational concepts for business services. The context where services are provided/consumed is another source of modeling concepts that address the social issues of business services. The provider, the consumer and the context of services form a 3-dimensional representational space of business services (see Figure 2).

The dimension of the service consumers has three concepts: *received value*, service *input/output* and service *bundling/unbundling*. Service values capture the values added to consumers' business after they consumes a service. Services values are not tangible but they can be cognitively experienced or sensed and thus are rather subjective. Service input and output are items that are created or exchanged during the occurrence of a business service. Unlike service values, service input and output are tangible or perceivable. Business services can be consolidated into service bundles (Kohlborn et al., 2009a). This exercise is called service bundling. Service unbundling refers to the practice of splitting a relatively big (and expensive) service to allow service consumers to make their own choices in consuming certain parts of the service they really want to without paying the whole price.

In the dimension of the service providers, the concepts that matter are the *returned/co-created* value, the service *capability* and service *composition/decomposition*. Returned/co-created values of a business service capture what the service provider gains after providing the service. They could be returned values such as payments/acknowledgments or values that are co-created by the consumers of the service. Service capability is the modeling concept that permits reasoning about what a service does in order to deliver its values and provision its outputs. The concept of service capability takes its root from business modeling. Homann (2006) conceives the business as a network of capabilities. Pohle (2005) stresses that business components have capabilities. Service capability may not give imperative details of how service values and outputs can be achieved. To formally describe service capability, we can annotate precondition (i.e. conditions in which the service may occur), postcondition (i.e. conditions and/or effects when the service has occurred) and schedules (i.e. checkpoints of functionality delivery, payments and penalties in case expected functionality is not delivered). Service decomposition refers to the practice of breaking down a relatively big service with high complexity for the ease of its design, operationalization or management in a top-down fashion. Service composition is an opposite practice in a bottom-up manner.

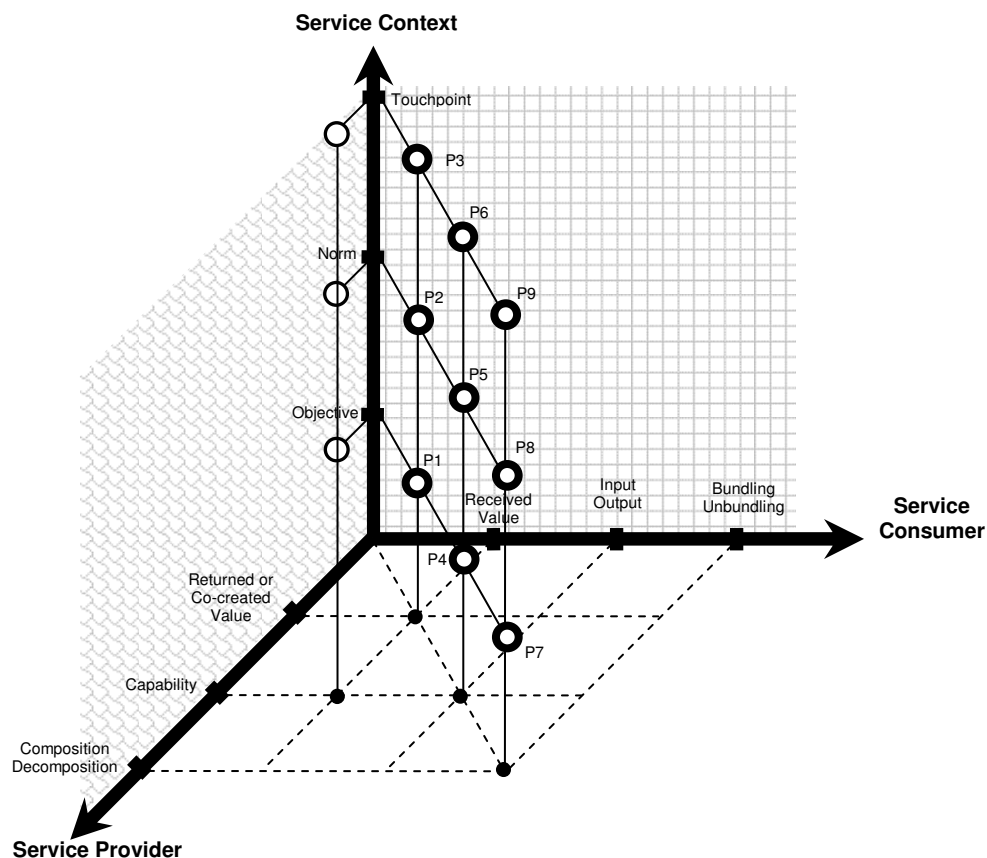


Figure 2: The representational space of business services has a total of 27 representational points (3 x 3 x 3). For the sake of simplicity, only 12 points (out of 27) are illustrated in this figure.

The dimension of service context features *objective*, *norm* and *touchpoint*. We can bear on the analysis of why business services are designed, operationalized and consumed by adding business objectives to the representation of services. Service stakeholders (i.e. providers and consumers) interact with one another in order to create values and exchange input/output in a business service. Their behaviors in providing or consuming a service are regulated through norms. Obligations, prohibitions and permissions are typical norms. Another kind of norms that we find relevant in the dimension of service context is the concept of assumption. In our industry engagements with government agencies, we have found what were documented as "client responsibilities" in their service catalog could best be expressed as service assumptions. These assumptions state what service consumers are expected to do, in order to enable the service provider to fulfill relevant service capabilities. In general, norms need to be monitored during the occurrence of a business service - if a violation is detected, the service may have to be aborted. Service touchpoint is the place where service interactions happen (Bitner, 1990). Through touchpoints, the service is experienced and perceived with all the senses.

Table 1: Intuitive meanings of the main nine points in the 3 x 3 x 3 representational space

Point	Indicative Meaning
P1	Business objectives relate values, co-created values and returned values of business services.
P2	Norms are assigned to co-created values and returned values of business services.
P3	Touchpoints help judge values, co-created values and returned values of business services.
P4	Business objectives help identify capabilities and input/output of business services.
P5	Norms can be put on capabilities and input/output of business services.
P6	Touchpoints help us define capabilities and input/output of business services.
P7	Business objectives give hint on how business services are (un-)bundled and (de-)composed.
P8	Norms help us judge the ways services are (un-)bundled and (de-)composed.
P9	Touchpoints help us identify potential (un-)bundling and (de-)composition of services.

Figure 2 illustrates the representational space of a business service. In our approach, a service representation can intuitively be interpreted as the set of points in the 3-dimensional space formed by service provider, service consumer and service context. There are a total of 9 main points in this representational grid. They are spotted by small circles with thick lines and enumerated as P1, P2... P9 and all lie on a geometric plane. Table 1 gives intuitive meanings conveyed by these points. Note that these points simply give some guidance of how we

identify and analyze business services. They are not meant to dictate a specific methodology for service identification, analysis and design.

## RELATIONSHIPS BETWEEN BUSINESS SERVICES

When it comes to service-oriented methodologies, services should be treated as the first citizen (Chang & Kim, 2007) (Kohlborn et al., 2009b), in the same manner that we treat objects in object-oriented methods (Booch, 2007). Managing relationships between objects plays a central role in object-oriented modeling. Reaching a consensus on the notions of subtyping, polymorphism, inheritance, aggregation and dependency was a major cornerstone in the evolution of the object-oriented paradigm. It gave rise to the proliferation of object-oriented modeling methods in the last two decades. Benefits of explicitly expressing relationships between objects include modularity, information hiding and reusability in software development. Most notably, we witnessed these advantages in the success stories of windowing programming libraries (e.g Java Swing or Microsoft Foundation Class), object-oriented design patterns (Gamma et al., 1995), as well as telecommunication standards such as the Open Distributed Processing – Reference Model (ISO/IEC, 2010). To be able to bring the service-oriented paradigm to a similar level of maturity, we need more insight into how business services relate to one another in a service ecosystem. In essence, we not only describe them individually but also represent relationships between them. In this section, we discuss how relationships between business services could be defined in terms of modeling concepts of the 3 x 3 x 3 grid presented in the previous section.

### Service Generalization/Specialization

A business services is said to be a specialization of another service in an ecosystem if the former can substitute for the latter without affecting the business of all services stakeholders involved. In other words, the former can safely be used wherever the latter is expected. If this is the case, we also say the latter service is a generalization of the former service. A business service can substitute for another if the following clauses hold

- Either the former service possesses all the capabilities specified for the latter service plus additional capabilities, or the capabilities specified for the former entail those specified for the latter. Entailment between service capabilities are determined based on their preconditions and postconditions. Intuitively speaking, the postcondition of the entailing capability entails that of the entailed capability, and the precondition of the entailed capability entails that of the entailing capability.
- The received values and co-created values experienced from the former include those experienced from the latter.
- Either the former produces more output than the latter does, or the output produced by the latter subsumes the output produced by the former.
- Either the former takes less input than the latter does, or the input taken by the former subsumes the input taken by the latter.
- The former service assumes weaker norms than the latter one does.
- The former service's objectives entail the latter service's objectives.

*Example 1.* Let us consider a boarding service provided by an airlines company. Through this service, the airlines company provides their passengers with check-in facility at counters that are attended by their staff. The company exploits an aircraft terminal to facilitate the boarding process. Now, the airlines company would like to deploy an upgraded version of this service. In the upgraded service, the passengers use the same check-in counters (i.e. touchpoints) and boarding gates as the original service but they have two choices for checking-in: either having a machine print their boarding pass or queuing at check-in counters. They also have two choices when boarding: either walking through the aircraft terminal to enter the aircraft or choosing a staircase leading to the rear door of the aircraft, whichever is more convenient for them. In this case, the upgraded service is actually a specialization of the original one in the sense that the deployment of this upgraded service still delivers what the passengers expect. The upgraded service actually provides additional capabilities (i.e. two choices for checking-in and boarding), creates more values (i.e. greater passengers' satisfaction, lessons learned in optimizing the boarding process for the airlines company) than the original one does.

*Example 2.* Let us consider two different services: one offers (passenger) car rental and the other vehicle rental. The first service (called `Car Rental`) accepts only customer's credit card and provides customers with passenger cars. The second service (called `Vehicle Rental`) accepts a wider range of cards (e.g. debit, credit) and provides not only passenger cars but also SUV & pickup vehicles. In terms of norms, the first service assumes less customers' responsibility than the second does in unusual scenarios such as vehicle breakdown.

Figure 3 depicts the input and output elements of these two services at the type level. Input/output types are drawn under rectangles; services under ellipses; input/output objects under arrows; service norms under folder-

like pictograms with text. Note that we use different line patterns for the illustration of these services: dashed lines for anything related to the first service whilst solid lines for the second. Triangle-headed arrows stand for subtyping: *Passenger Car* is subtype of *Vehicle* and likewise *Credit Card* is subtype of *Bank Card*. In terms of input/output, service *Vehicle Rental* can substitute for service *Car Rental* because *Vehicle* and *Bank Card* subsume *Passenger Car* and *Credit Card*, respectively. In terms of service norms, this substitutability also holds as the second service assumes weaker customer's responsibility on vehicle condition than the first one does thanks to the installment of monitoring devices on vehicles it manages. We also use a triangle-headed arrow to visually express that service *Vehicle Rental* is a specialization of service *Car Rental*. Note also that the triangle-headed arrow applied to the two services and the ones to the input/output types do not run in the same direction. This might look counterintuitive at the first place, but the substitutability rule is rigorously stated in the aforementioned clauses.

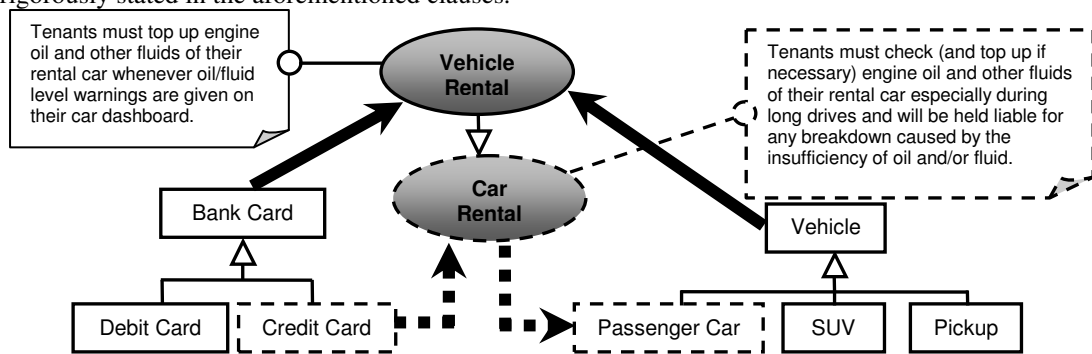


Figure 3: Service *Vehicle Rental* can substitute for *Car Rental* in terms of input/output objects and norms.

### Service Aggregation

An aggregation exists between two business services if one of them is conceptually part of the other. If this is the case, the lifecycle of the former service is well within that of the latter. We investigate this relationship from two different perspectives. As pointed out in the 3 x 3 x 3 modeling grid, service (de-)composition constitutes aggregation from the service provider's perspective whereas service (un-)bundling define aggregation from the service consumers' perspective. Table 2 elaborates this standpoint by explaining the four different kinds of service aggregation.

Table 2: Taxonomy of service aggregation

Service Aggregation	Explanation	Perspective
Service decomposition (top-down)	The exercise of breaking down a relatively complex service into component services through outsourcing, subcontracting, delegation, etc.	Service provider's
Service composition (bottom-up)	The practice of combing existing services (e.g. those that are available from a service catalog) to fulfill a service contract.	Service provider's
Service bundling (bottom-up)	The exercise of grouping existing related services to provide service consumers with more comprehensive service offerings.	Service consumers'
Service unbundling (top-down)	The practice of ungrouping a service allows service consumers to partially consume the service without paying for the whole price.	Service consumers'

*Example 3.* Let us revisit service *Vehicle Rental* that is depicted in Figure 3. On the one hand, for the sake of service operationalization, the provider of this service may break it into several component cooperating services and subcontract them to other service providers. On the other hand, this service can be bundled with other related services to produce a more comprehensive service offering (called *Travel Package*), which is also known as service bundling. This service decomposition and service bundling are illustrated in Figure 4.

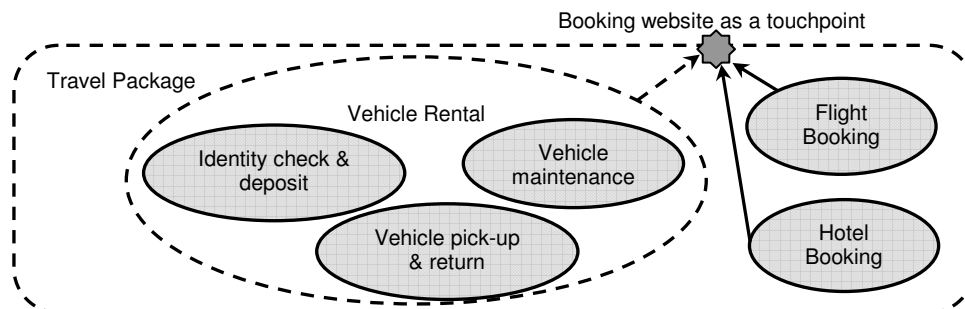


Figure 4: An example of service decomposition and service bundling

Note that bundled services and the resulting service offering all have a common touchpoint: a website that enables customers to make bookings. This touchpoint is drawn under an 8-point star in Figure 4. The pictogram

of service bundling is slightly different from that of service decomposition (i.e. a rounded rectangle versus an ellipse) but they have two features in common: line pattern (i.e. using dashed lines) and nested pictograms (i.e. pictograms of their component services are nested inside their pictograms).

### Service Dependency

Dependency is a relationship between two business services one of which exists because of the existence of the other. Quite often, we see a dependency relationship between services that share one or more service stakeholders. If the shared stakeholders are service consumers, they are likely to find that values and/or output received from the first service need to be paired with those received from the second service. For instance, there is a dependency between a service that sells Blue-ray players and another service that sells or leases movies formatted in Blue-ray. A person who buys a Blue-ray player will find it pointless not to buy or to rent movies formatted in the Blue-ray standard and vice versa. Note that this dependency relationship is not identical to the relatedness of services that are bundled for the convenience of the consumption of these services. The bundled services produce values or outputs that are related but they may exist independently of one another. Yet another example in this category is a dependency between online shopping (as a service) and shipping & tracking (as another service). The company offering the online shopping service relies on another company that delivers purchased goods to customers who have paid for their order. We call this relationships *value-driven dependency*.

If a consumer of one business service becomes the provider of another service, then the shared stakeholder plays a bridging role by connecting the consumers of the second service to the provider of the first one. For example, a travel agency serves as the service provider in making multiple bookings for tourists but as a consumer in arranging tours (e.g. scenic tour, scuba drive tour) with tour companies. In this case, the service provider of the first service consumes the second one. We call these relationships *stakeholder-driven dependency*.

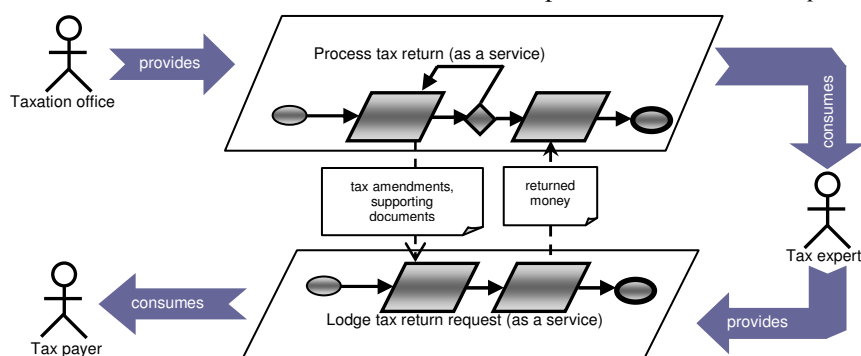


Figure 5: Stakeholder-driven dependency between Log tax return request and Process tax return. The two services are illustrated as parallelograms enclosing illustrative processing details.

*Example 4.* Figure 5 gives a more comprehensive example of stakeholder-driven dependency between two business services. They are *Process tax return (as a service)* and *Lodge tax return request (as another service)*. A tax expert provides the latter service and consumes the former one, which is provided by a taxation office. The expert helps their customers prepare tax return requests and submits them to the taxation office. During the occurrence of these two services, the lawyer also acquires additional information from their client while answering any questions posted by the taxation office. These two services almost occur in parallel. They are illustrated as parallelograms revealing some processing details in Figure 5. The dash arrows stand for data flows between the two services during their occurrences. Data (e.g. tax amendments, supporting documents, money returned) exchanged on these flows are drawn under folder-like pictograms. We use the BPMN (Business Process Model and Notation) to express processing details of the two services. However, we do not represent the granularity of task/activity of these two services. All the tasks are drawn for illustration purposes only.

## META-MODELING FOR BUSINESS SERVICES

In this section, we present a meta-model for business services and discuss the roles that service relationships play in the engineering and the evolution of service ecosystems. Figure 6 gives such a meta-model that is expressed using the Unified Modeling Language. Each business service (i.e. UML class *Business Service*) has capabilities and is associated with objectives, norms, touchpoints and parameters. A capability has pre/post condition and is associated with delivery schedules, payment schedules and penalties. A service parameter can specifically be described as either input (i.e. class *Service Input*) or output (i.e. class *Service Output*). A boolean attribute called *tangible* will tell whether it is tangible (e.g. a physical object) or perceivable (e.g. an electronic item). A business service is associated with service stakeholders who play the either the consumer's

role (i.e. subclass *Service Consumer*) or the provider's role (i.e. subclass *Service Provider*). This association features a UML association class (named *Value*) that represents the value of the associated service. This class has a boolean attribute that can tell whether the represented value is a co-created value or a returned value. Being an association class, *Value* is connected to class *Business Service* and a subclass of *Service Stakeholder*.

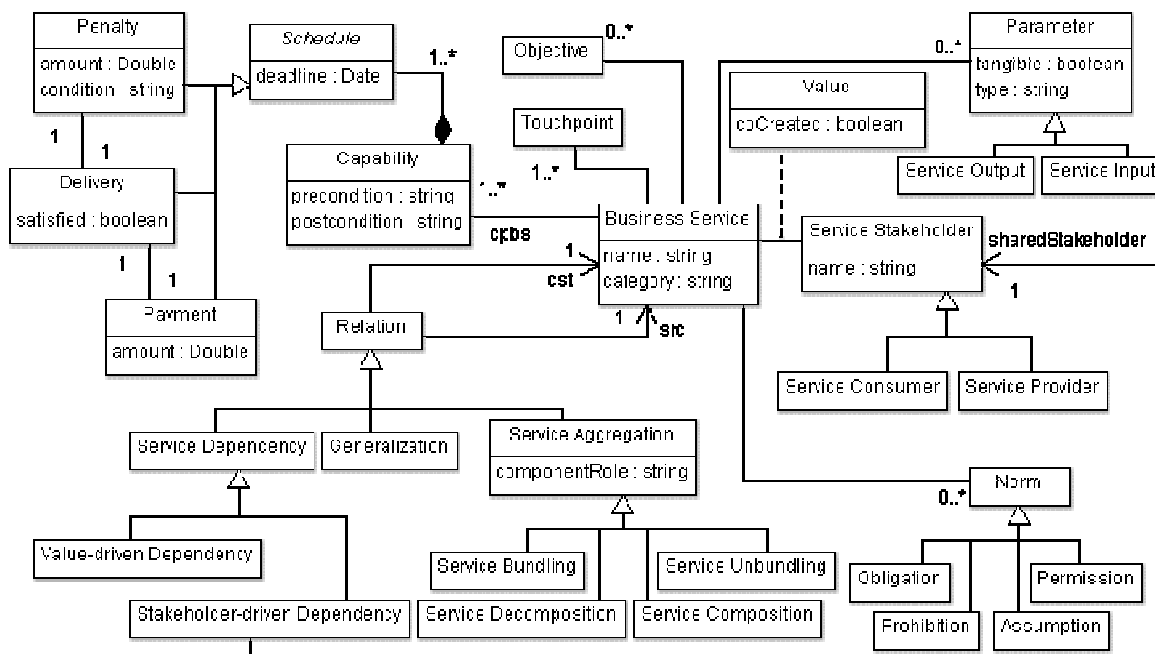


Figure 6: A meta-model for business services and their relationships.

A service relation (i.e. UML class *Relation*) literally connects two business services. The aforementioned three kinds of service relationship are expressed as three subclasses of *Relation*. Additional UML subclasses represent more refined forms of service aggregation and service dependency. Note that each stakeholder-driven dependency is associated with a stakeholder via role name *sharedStakeholder*.

### The roles of service relationships in the engineering and the evolution of service ecosystems

Business domains involving large and complex collection of loosely coupled services provided by autonomous enterprises are becoming increasingly prevalent. Such interactions among and between independent services are what define a service ecosystem (Sawatani, 2007). The emergence of an ecosystem context can be seen in various places such as in the form of Shared Service Centers providing central, standardized services from different agencies or departments in the public sector (Janssen et al., 2004). In order to effectively manage the operationalization of a service ecosystem, we should be able to figure out the core to bear on the analysis (Ghose, 2011). The core of a service ecosystem is a set of basic services that can be combined or extended via service aggregation and service specialization in order to operationalize the businesses of an organization or a collective business system. We might observe dependencies among service members of the core.

*Example 5.* Let us look at the operations of an airport as an example of service ecosystem. The flight (as a service) can be decomposed into a few component services: ticketing, boarding and aircrafts operation. Unbundling the flight yields low-cost flight, speedy boarding, heavy luggage check-in and onboard meals. To realize the boarding service, we can compose two services: security screening and baggage handling. We may introduce an improved boarding service explained in Example 1 as specialization of the standard boarding service. In this example, the core of the ecosystem of an airport includes: security screening, baggage handling, the standard boarding service, speedy boarding, heavy luggage check-in and onboard meals.

The notion of ecosystem would make more sense in service engineering if we relate the importance of service relationships to the evolution of a service ecosystem. The lifecycle of business services may span analysis, design, implementation, publishing, operation and retirement (Kohlborn et al., 2009b). Changes can be made to services at any stage of their entire lifecycle. A change made to a certain service initially breaks some relationships to which the service is connected, leading to a perturbation of the service ecosystem (i.e. the ecosystem is temporarily perturbed). Provided we have explicitly and systematically represented service relationships in the ecosystem, the well-defined relationships model will tell us how to propagate this initial change to other services – probably the ones that are in relation with the service to which the initial change was



made, in order to restore the ecosystem equilibrium. Note that this change propagation process may take a few or a very large number of attempts, as the initial change would trigger a secondary change, which in turn might spark another change and so on so forth. The process carries on until we reach new equilibrium of the ecosystem.

Intuitively speaking, identifying generalization/specialization relationships attributes to the reusability of a service ecosystem as the more generalized a service is the more frequently it will be consumed. Service aggregation has a direct influence over the modularity and the cohesion of an ecosystem in the following manners: service bundling/composition helps modularize services (i.e. the principle of low coupling) whereas unbundling/decomposition helps break down services that offer a relatively larger number of functionalities (i.e. the principle of high cohesion). Service dependency helps measure coupling between services of an ecosystem.

## APPLICATION

We present our work on a case-study in which we have applied the proposed modeling approach. The case-study is about an agency who provide business services under the government body of an Australian state. They proposed a total of 29 services and documented them using their own ad-hoc template. We studied this document and populated all documented services using our modeling approach.

Table 2: Summary for the representation of a service catalog using the 3 x 3 x 3 modeling grid. For the sake of confidentiality, we choose to enumerate all the services and present only the number of fields populated.

Service	Stake	Decomp	Capab	Assumpt	Obligation	Input	Output
FI1	2	–	5	7	3	–	1
FI2	2	–	7	5	6	–	1
FI3	2	–	5	6	13	–	1
FI4	4	–	7	9	7	–	1
FI5	2	–	8	7	7	–	2
FI6	4	–	6	3	7	–	1
FI7	4	–	5	6	1	–	1
FI8	3	–	6	2	–	–	1
FI9	3	–	6	3	6	–	1
HR1	4	4	10	8	7	–	2
HR2	3	2	20	8	8	–	2
HR3	3	–	10	12	8	–	1
HR4	2	–	8	14	2	–	2
HR5	3	–	9	7	5	–	1
FIM1	2	–	3	6	–	–	2
FIM2	2	2	4	11	4	–	1
FIM3	2	3	10	11	2	1	3
FIM4	2	–	4	2	1	–	1
FIM5	2	–	5	1	1	–	1
FIM6	3	–	9	11	7	–	1
FIM7	2	2	6	4	8	–	3
IS1	2	–	18	9	3	–	1
IS2	2	–	11	9	3	–	1
IS3	2	2	21	13	3	–	1
IS4	2	–	11	6	2	–	1
IS5	2	2	5	5	5	–	2
IS6	2	2	7	2	4	–	2
IS7	2	2	3	6	5	–	1
AD1	2	–	5	7	8	–	3

Table 3 gives a summary of how many fields we have populated for each business services in the catalog. Trivial fields such as service name and service category are omitted in this summary. The most left enumerate business services in the catalog. Other columns stand for (from left to right) numbers of service stakeholders, constituent services via decomposition, service capabilities, assumptions, obligations, input and output, respectively. We came to the following methodological remarks while doing this case-study.

- Some fields in the document could straightforwardly be populated, including service capability, service stakeholder and input/output. However, the document contains inconsistent and potentially ambiguous fields. In particular, we came across a field that means service capability for some entries in the document but is closer to service assumption for others. What are documented as client's responsibility in the catalog was best populated as service assumptions.

- We found an interesting way to extract what are described as service measurements in the service catalog to service assumptions, including delivery schedules and obligations after having harmonized these service measurements.
- It was unclear how the services documented in this catalog can be unbundled or decomposed at the first place. We augmented this representation by decomposing a total of 9 services and populated the constituent services at the same level of details as we did to the decomposed services. The decomposition of these services was made by grouping their service capabilities, services assumptions in a way that maximizes the cohesion of constituent services.
- The most populated fields in this case-study are service capability, service assumption and obligations. Service output is more often described than service input.

## RELATED WORK

Our interpretation of the generalization/specialization relationship between business services was inspired by the notion of *subtyping* that was first made popular in programming language theory and later extended to conceptual modeling. We do not directly link this interpretation to the definition of object-oriented inheritance or subclass, which is in fact a mechanism to achieve subtyping. In essence, we may not apply a pairwise comparison of operational details to the two services in question. Instead, we compare their contractual specs to see if the specialized service can safely be used in a context where the generalized service is expected (i.e. substitutability). To the best of our knowledge, this interpretation is in line with the concept of *projection inheritance* in workflows (van-der-Aalst & Basten, 2002) and the notion of *function sybtyping* in research on behavioral compatibility (Liskov & Wing, 1994).

Service *bundling* has been investigated in the form of service portfolio management (Kohlborn et al., 2009a) and the realm of banking (Altinkemer, 2001). In our view, service bundling is a special case of service aggregation. We propose a taxonomy of service aggregation featuring bundling, unbundling, composition and decomposition. The taxonomy reflects the perspectives on both the service provider and the service consumer.

## CONCLUSION

This paper discusses relationships between business services in a service ecosystem. The concept of *business service* emerges amid the weakness of software-driven SOA in dealing the granularity of business cognizant in the EA. When it comes to modeling a service ecosystem, we argue that IT-enabled business services should be treated as first-class elements in the same manner that objects are treated in object-oriented methodologies. We should describe services and identify relationships between them. We have refined our previous work on the representation of business services and come up with a model of relationships between services.

Work is currently underway to formalize the three relationships between business services we have defined in this paper. Once achieved, this work would open a door for automatically identifying services relationships as development of an engine that supports change propagation in the evolution of service ecosystems. Another piece of future work is to correlate our definition of service relationships to the cooperating nature of business processes. In our group, we have defined a richer set of relationships for business process models. As illustrated in Figure 1, we may view a business service as wrapper for low-level business process models. This would lead to a new understanding of how the cooperation and relationships between underlying business processes might surface and help determine emerging relationships between the business services that wrap them.

## REFERENCES

- Altinkemer K. 2001. Bundling e-banking Services. *Communications of the ACM* 44 (6), pp 45-47
- Bitner M. 1990. Evaluating service encounters: the effects of physical surroundings and employee responses. *Journal of Marketing* 54 (2), pp 69-82
- Booch G. 2007. *Object-Oriented Analysis and Design with Applications*. (3rd Edition): Addison-Wesley Professional, isbn 978-0201895513
- Chang SH, Kim SD. 2007, A Systematic Approach to Service-Oriented Analysis and Design, *proceedings of 8th International Conference on Product-Focused Software Process Improvement*, pp 374-388, Riga, Latvia, July 2007
- Gamma E, Helm R, Johnson R, Vlissides J. 1995. *Design Patterns*: Addison-Wesley, isbn 0-201-63361-2
- Ghose A. 2011, A formal approach to designing and managing service ecosystems, *proceedings of 19th International Conference on Information Systems Development*, pp Prague, Czech Republic, August 2011

- Grigoriu A. 2007. *SOA or SOEA, Service Oriented Enterprise Architecture*, (White paper), Toolbox for IT.
- Hansmann U, Merk L, Nicklous M-S, Stober T. 2011. *Pervasive Computing: The Mobile World*: Springer, isbn 978-3642055256
- Homann U. 2006. *A Business-Oriented Foundation for Service Orientation*, (White paper), Microsoft MSDN Library.
- ISO/IEC. 2010. *ITU-T X.902 — ISO/IEC 10746-2 Information Technology - Open Distributed Processing - Reference Model - Foundations*, (International Standard), SC 7 and ITU.
- Janssen, M., Wagenaar, R. 2004, An analysis of a shared services centre in e-government, *proceedings of 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 5*, pp IEEE Computer Society, Hawaii, USA,
- Kloppmann M, Koenig D, Leymann F, Pfau G, Rickayzen A, Riegen C, Schmidt P, Trickovic I. 2005. *WS-BPEL Extension for People – BPEL4People*, (White paper), IBM and SAP.
- Kohlborn T, Fielt E, Korthaus A, Rosemann M. 2009a, Towards a Service Portfolio Management Framework, *proceedings of 20th Australasian Conference on Information Systems*, pp AIS Electronic Library (AISeL), Melbourne, Australia, December 2009
- Kohlborn T, Korthaus A, Rosemann M. 2009b, Business and Software Service Lifecycle Management, *proceedings of 13th IEEE International Enterprise Distributed Object Computing Conference*, pp 87-96, IEEE Computer Society, Auckland, New Zealand, September 2009
- Lê L-S, Dam HK, Ghose A. 2010, On Business Services Representation – The 3 x 3 x 3 Approach, *proceedings of 21st Australasian Conference on Information Systems*, pp 58, AIS Electronic Library (AISeL), Brisbane, Australia, December 2010
- Liskov BH, Wing JM. 1994. A behavioral notion of subtyping. *ACM Transactions on Programming Languages and Systems* 16 (6), pp 1811-1841
- Paulson LD. 2006. Services Science: A New Field for Today's Economy. *Computer* 39 (8), pp 18-21
- Pohle G. 2005. *Component business models - Making specialization real*, (White paper), IBM® Institute for Business Value.
- Sarang P, Juric M, Mathew B. 2006. *Business Process Execution Language for Web Services BPEL and BPEL4WS*. 2nd edition: Packt Publishing, isbn 978-1904811817
- Sawatani Y. 2007, Research in service ecosystems, *proceedings of Management of Engineering and Technology (PICMET'07)*, pp 2763–2768, Portland, USA,
- Singh MP, Huhns MN. 2005. *Service-Oriented Computing: Semantics, Processes, Agents*. (1st edition): Wiley, isbn 978-0470091487
- van-der-Aalst WMP, Basten T. 2002. Inheritance of Workflows - An approach to tackling problems related to change. *Theoretical Computer Science* 270 (1-2), pp 125-203

## ACKNOWLEDGEMENTS

This research has been funded by the research initiative Smart Services CRC, established and supported under the Australian Government's Cooperative Research Centres Programme.

## COPYRIGHT

Lam-Son Lê © 2011. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.