Service as a Resource

VMBO Amsterdam 2009. Based on [caise09]

Hans Weigand[1], Paul Johannesson[2], Birger Andersson[2], Maria Bergholtz[2]

[1] Tilburg University, P.O.Box 90153,
5000 LE Tilburg, The Netherlands
H.Weigand@uvt.nl
[2] Royal Institute of Technology
Department of Computer and Systems Sciences, Sweden
pajo,ba,maria@dsv.su.se

REA

The Resource-Event-Agent (REA) ontology was formulated originally in [Mc82] and has been developed further, e.g. in [UM03, Ge99, Hr06]. It was originally intended as a basis for accounting information systems and focused on representing increases and decreases of value in an organization. REA has been extended with patterns to form a foundation for enterprise information systems architectures [Hr06], and it has also been applied to e-commerce frameworks [UM03]. The following is a short overview of the core concepts of the REA ontology based on [caise09].

A *resource* is any object that is under control of an agent and regarded as valuable by some agent. The value can be monetary or of an intangible nature, such as status, health state, and security.

Resources are modified or exchanged in processes. A *conversion process* uses some input resources to produce new or modify existing resources, like in manufacturing. An *exchange process* occurs as two agents exchange resources. To acquire a resource an agent has to give up some other resource.

The constituents of processes are called *economic events*. An economic event is carried out by an agent and affects a resource. The notion of stockflow is used to specify in what way an economic event affects a resource. REA identifies five stockflows: produce, use, consume, take and give, where the first three occur in conversion processes and the latter two in exchange processes. REA recognizes two kinds of duality between events: conversion duality and exchange duality. An *agent* is an individual or organization capable of having control over economic resources, and transferring or receiving the control to or from other agents.

Services in REA

In REA, a service is a resource as it is viewed as valuable by some agent and can be transferred between agents [caise09]. This is captured in the ontology by modeling Service Type as a subtype of Resource Type. As such, it inherits all features of resources, in particular:

- it can be exchanged between agents. Customers pay for a service as they pay for a good. There is no transfer of ownership in the exchange, as the service itself is not something owned. However, it is under control of an agent.
- it is part of a conversion process. To produce a service, other resources have to be spent, and by consuming the service, other resources are being modified. This is not different from goods. What is special is that the production and consumption events have to coincide in time.
- the service delivery is governed by a contract – in REA a bundle of commitments. The fulfillment of the commitments is typically not instantaneous but takes a period of time.

If we would model a service in a different way, all these features would have to be doubled for resources and for services. So although we recognize differences between services and goods, there are strong arguments to view them both as resources in the REA sense.

A distinctive feature of a service is that it has a goal to modify other resources by converting or exchanging them. We model this by the association *hasGoal* from Service to Economic Event. A service does not specify how it is to be realized, i.e. how its goals are to be achieved. Instead, a service can be realized in many different ways, i.e. by many different processes.

The exchange of a service is simple: the service is exchanged between agents in return for money. All the coordination services that can be used within an exchange process apply to service exchanges as well. At the same time, the service is a resource produced in a conversion process by the provider, and consumed in a conversion process by the customer. REA usually renders only one perspective, but for the understanding of the service interfacing between the provider and customer, we have included both perspectives (indicated by dotted rectangles) in one figure. The economic increment event for creating the service stands in conversion duality to a resource use event. At the customer side, we distinguish between service use and service consumption. Both

can be used to add value to some resource (production event), typically in combination of some effort of the customer himself - that is why we also include a resource use event here. We use the <goal> relationship as a short-hand for the production event at the customer side (as in [caise09]) as well as the resource use event at the provider's side.

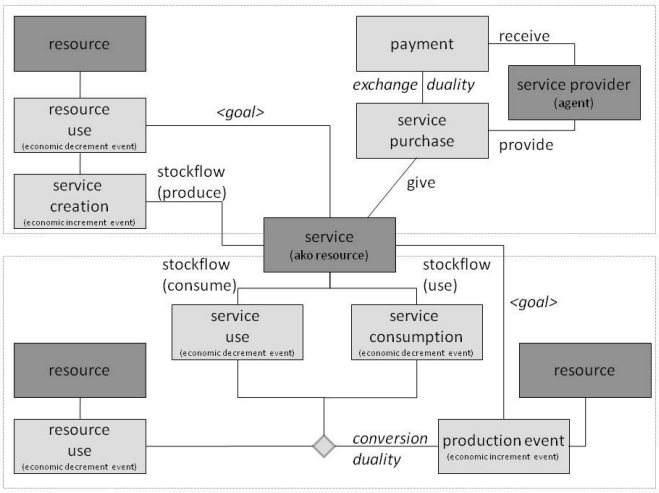

Fig. 1 Service ontology in REA

The difference between service consumption and service use can be illustrated by the example of [FG09] of a fire brigade. This could be a service hired by the municipality for a certain period. Service consumption is then a matter of time: at the end of the period, it is completely consumed. During the period, the fire brigade may become active in the case of an emergency, as stipulated in the service contract. This is an instance of service use. The effect of service use is a particular house being rescued, whereas the effect of the service consumption is the increased security of all houses in town. Details e.g. on the maximum emergency response time may be written down in an SLA. Security of the resource in this case does not mean that no incident could happen but that the damage will be limited.

Service systems

The notion of service system as proposed recently by Spohrer and colleagues [S08] is based on Vargo's Service-Dominant Logic [VL06]. There is no explicit ontology or modeling technique published to support it, but we can identify a number of key concepts. A *system* is a configuration of *resources*. Some resources are *operants* that act on other (operand or operant) resources. A *service* is the application of resources to bring about changes that have *value* for another system. Services are performed in the context of *economic exchanges* – the mutual value creation by two or more interacting systems. So value is created in *interactions* between service systems. As a first rough approximation, we can make the following mapping to REA: resource - resource, service – service as a resource, produced and used/consumed in business events, system – agent, and exchange – exchange process. From Fig. 1, we can immediately see that some resources are operant (both at provider and customer side) whereas others are operand (at customer side). The model also makes clear that value is typically *co-created* in services (although co-creation is not confined to services: we could also have a non-service resource exchanged, and keep the same economic events, including the use of customer resources).

Unified notion of services

The service ontology unifies the notion of business service (like hotel rooms, loans and hair dressing) and software service (e.g. web service by means of which hotel reservations can be made). The ontological representation of service is in line with the SOA idea of services as a way of hiding process details to the service customer.

Service classification

The starting point of the service classification in Fig. 2 is that any actor has to provide a number of services to its environment, thereby producing value for other actors. These services are called core services. The reason for viewing these services as core is that they provide the raison d'être for an actor in a value network, as they specify what value the actor is able to provide to the network.

Following e3value, we assume that services exchanged between actors are typically bundled. A set of services offered together by a service provider is a *service bundle*. For instance, hairdressing is a service bundle

that includes not only hair cutting but also hair washing, a magazine service for customers while they are waiting, a cup of coffee, etc..

Given a set of core services, there is a need for a number of services that can improve on the core services. We divide these services into four categories: complementary services enhancing services, support services, and coordination services.

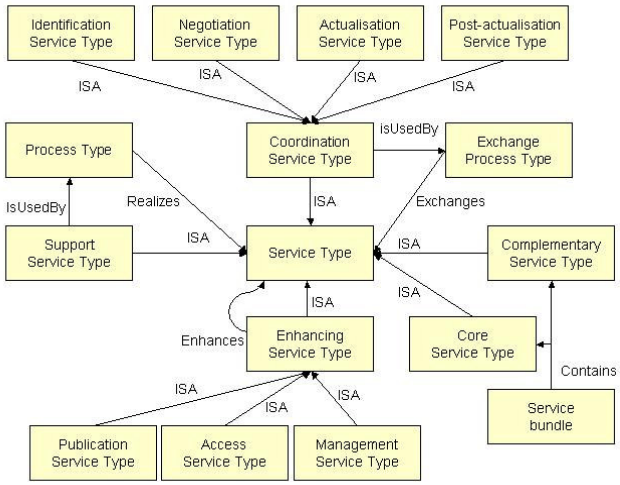

Fig. 2. Service Ontology (highlighting different dimensions of the Service concept)

Complementary service

A service complements another service if they are part of the same service bundle and their goals concern the same resource [We07]. For example, a gift-wrapping service complements a book sales service by having as goal to improve the book by packaging it in an attractive way. Thus, both services concern the same resource, the book.

Enhancing service

An enhancing service is a service that adds value to another service (instead of some other type of resource) In other words, it has an effect on the quality of another service or some feature like visibility or accessibility. By definition, it is existence-dependent on the other service.

Support services

A service A supports a service B if A has as goal to produce B, or if A has as goal to produce a resource that is used in a process that produces B.

Coordination services

A coordination service is any service that supports an exchange process (of a service or good). It is used for ensuring that communicating parties in a business relation are coordinated or synchronized. Coordination services can be defined in terms of commitments that they create ore manipulate. For example, a hotel reservation service creates a commitment from the hotel to provide a room of a certain type at some specified date.

Service Layer Architecture

For a Service Layer Architecture that integrates business services and software services, we draw upon the enterprise ontology of Dietz [Di06] that distinguishes a social (performational) level, an informational level and a formational level. To illustrate: an order is a request at the social level, a message at the informational level and a document or file at the formational level.

In the context of IS design, an *informational* service is a software service that has as goal to produce information or enhance communication. A *utility* service is a service that is produced by means of IT hardware and supports informational services by storing, processing or transferring data. A *business service* is an economic service provided by an economic actor to fulfill a customer need. Both informational and utility services can be classified as supporting services (cf. section 3.2).

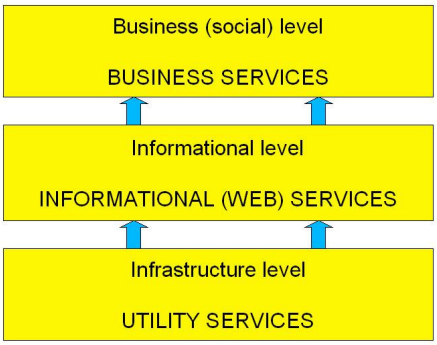

Fig. 3 Service Layer Architecture

Information services are software services that are characterized by some economic autonomy. They can use other information services and utility services as resources Information services can be supported by management services whose aim is to maintain and improve the quality level of the managed service over time, and other enhancing services

Utility services are software services at the infrastructure level that are characterized by a certain economic autonomy (which makes them suitable candidates for outsourcing) and usually support more than one informational service. The value provided by the utility service is the storage of data or the execution of programs. Also at this level we can have managing and other enhancing services.

Informational services support the business level in different ways, depending on their focus [caise09]. We mention in particular that, within a proper institutional context, informational services can implement coordination services. Coordination services manipulate commitments, as we said. Informational services manipulate information. However, within a certain context, this information exchange can count as a communicative act with a certain social effect. In this way, the IS department can realize a coordination service, although the end responsibility to the customer is still with the business.

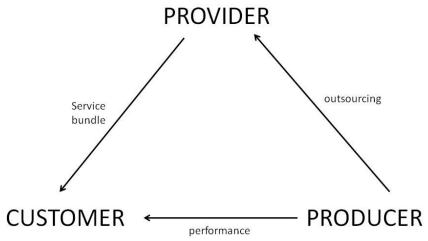

Fig. 4 delegation triangle

Service delegation

Service description becomes more complicated when services are being delegated. The dyadic relationship between *service customer* and *service provider* is then extended into a triad that includes a *service producer* (Fig. 4). In the outsourcing literature [e.g. [LC09]], the service provider is also called the *buyer* and the service producer the *supplier,* since the service producer is delivering something to the buyer (provider) in return for money. Examples of this kind of delegation are: a bank organization outsourcing its customer help desk to India, or Amazon using a courier to deliver ordered books to the customer. Another example [FG09] mentioned above is the municipality that sets up a contract with a fire brigade service.

What services should be distinguished here? Does the producer offer a service to the customer *and* one to the provider/buyer? Economically speaking, the service to the customer is not provided by the producer, although he may be most visible, but by the provider typically as part of a bundle of services paid for by the customer. Any claims from the customer should be directed to the provider, not the producer. So that means the producer does not have a service relationship with the customer, only with the provider. A possible problem with this approach is that it is not immediately clear what the producer offers to the provider. An alternative analysis is modeling the service to the customer as something provided by the producer. After all, it's the fireman rescuing the citizens' house. This service is paid for by the municipality: exchange duality is ensured, although the exchange is not purely reciprocal (with 3 agents involved). Finally, a kind of synthesis of these two

approaches is possible: the producer provides service S to customer. He gets money from the provider for providing S, in other words, he performs a service S' to the provider of "giving service S tot customers" (in our service model, the latter is the goal of the former). This service S' is what the provider can also put into his service bundle to the customer. This becomes service S": the service of arranging S, meaning: the service of contracting an S' agent. For the customer, the indirect effect of S" is that S is provided and the goal of S is realized.

Delegation often plays a role in web services. It is often the case that the web service is hosted (produced) by an IT company or department on behalf of some other agent, e.g. a hotel. Then the web service is both a service offered by the hotel to its customers ("book online") and a service offered by the IT company to the hotel.

References

[Di06] J. Dietz. *Enterprise Ontology - Theory and Methodology*. Springer, Berlin (2006).

[Ge99] Geerts, G., McCarthy, W. E. An Accounting Object Infrastructure For Knowledge-Based Enterprise Models. *IEEE Int. Systems & Their Applications*, pp. 89-94, (1999).

[FG09] Ferrario, R, Guarino, N., Fernandeze Barrera, M. Towards an Ontological Foundation for Service Science: the Legal Perspective.

[Hr06] Hruby, P. *Model-Driven Design of Software Applications with Business Patterns*. Springer Verlag ISBN: 3540301542. (2006).

[Mc82] McCarthy W. E., The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. *The Accounting Review* (1982).

[S08] J. Spohrer et al, The Service System Is the Basic Abstraction of Service Science. Proc. *HICSS*, (2008)

[UM03] UN/CEFACT Modelling Methodology (UMM) User Guide. Available at http://www.unece.org/cefact/umm/UMM_userguide_220606.pdf 2008-02-19. (2003)

[VL06] S. L. Vargo, R. F. Lusch, and F. W. Morgan, "Historical Perspectives on Service-Dominant Logic," in *The Service-Dominant Logic of Marketing*, R. F. Lusch and S. L. Vargo, Editors, M. E. Sharpe, Armonk, NY (2006), pp. 29–42

[We06] Weigand H., Johannesson, P., Andersson, B., Bergholtz, M., Edirisuriya, A., Ilayperuma, T. On the Notion of Value Object. Proc. CAiSE'06 (2006), pp,321-335.

[We07] Weigand, H. et al: Strategic Analysis Using Value Modeling--The c3-Value Approach. HICSS 2007: 175

[caise09] Weigand, H. et al: Value-based Service Modeling and Design: Toward a Unified View of Services. Proc. CAiSE 09.