Energy Services: A Case Study in Real-World Service Configuration*

Ziv Baida[1], Jaap Gordijn[1], Hanne Sæle[2],
Andrei Z. Morch[2], and Hans Akkermans[1]

[1] Free University, FEW/Business Informatics
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{ziv,gordijn,elly}@cs.vu.nl
[2] Dep. of Energy Systems, SINTEF Energy Research
7465 Trondheim, Norway
{Hanne.Saele,azm}@sintef.no

Abstract. Current eCommerce is still mainly characterized by the trading of commodity goods. Many industries offer complex compositions of goods based on customers' specifications. This is facilitated by a component-based description of goods, supported by a variety of product classification schemes, e.g., UNSPSC and eCl@ss. These focus on physical goods – wrongly referred to as *products* – rather than on services. *Services* are intangible products, for instance insurances, transportation, network connectivity, events hosting, entertainment or energy supply. Due to major differences between goods and services, product classification schemes cannot support automated service scenarios, such as a customer who wishes to define and buy a set of independent services, possibly supplied by multiple suppliers, via one website. To enable such eCommerce scenarios for services, a service ontology is required that supports a component-based structure of services. Defining a set of services is then reduced to a *configuration* task, as studied in the knowledge management literature. In this paper we use a case study from the Norwegian energy sector to describe how a component-based ontological description of services facilitates the automated design of a set of services, a so called *service bundle.*

1 Introduction

Although some e-business initiatives have failed, the massive diffusion of the Internet still opens up many new opportunities for businesses, such as in the field of online service provisioning. So far, the Internet has mainly been used as a channel for selling physical goods. It is for instance quite common that customers can configure a complex good (e.g. a PC) out of more elementary

* This work has been partially supported by the European Commission, as project No. IST-2001-33144 OBELIX (Ontology-Based ELectronic Integration of compleX products and value chains). We thank especially Nieves Peña (LABEIN, Bilbao, Spain) for useful discussions on the topic of this paper.

A. Persson and J. Stirna (Eds.): CAiSE 2004, LNCS 3084, pp. 36–50, 2004.

components and order such a good online. Examples can be found on websites of market leaders such as Dell and Cisco. Such an eCommerce scenario requires a component-based ontology of *goods*, specifically suited for *classification* (to allow customers to find goods) and *configuration* (to facilitate in composing complex goods). Examples of such supporting ontologies are UNSPSC [3] and eCl@ss [2].

Unlike physical goods, services are not supported well by these ontologies due to major differences between services and goods: their intangible nature (vs. tangible goods), customer involvement in the production process of services (vs. standard, off the shelf goods), difficulties in ensuring standard quality levels due to the important role of employees in the service process, and more [19].

However, from an economical perspective, services grow more and more in importance [4], and will be offered and deployed via the Internet increasingly. It is therefore important to extend current goods-biased classification and configuration ontologies with a service perspective. This paper proposes an ontological foundation for service configuration.

It is important to understand that our interpretation of the term *service* stems from business science, a community that has been researching the notion of services extensively already for decennia [15, 12, 23, 19, 21, 16, 8]. So, we do *not* take *web-services* publications such as [22] as our starting point; web-services take hardly a commercial/business value perspective on services, a perspective that is needed by customers and suppliers to configure valuable, complex services.

Based on service marketing and service management literature we have created a generic *service ontology* that describes services both from a supply-driven and demand-oriented perspective. We combine our business-driven conceptualization of the service sector with knowledge management research, specifically work on configuration task ontologies [13]. As a result, by using our service ontology as well as already existing configuration ontologies we can configure complex services (called a *service bundle*) out of more elementary services, possibly offered by multiple suppliers.

In the remainder of this paper we explain how the service configuration process takes place, and how it is facilitated by a service ontology. We demonstrate how we put our theory into practice by presenting a case study from the energy sector. We first present in Section 2 a top-level description of our service ontology, followed by an explanation of service terminology that underlies our work (Section 3). Subsequently, in Section 4 we explain how services can be described as components. Then, in Section 5 we introduce our case study domain, the electricity market, and we analyze potential service bundles in this market using our configuration-biased service ontology. Finally, in Section 6 we present our conclusions.

2 Service Ontology

Using the service management and marketing literature as a starting point, we have developed in earlier work a generic component-based service ontology [7, 6]. The ontology incorporates both a customer perspective and a supplier perspective on services, and it includes unique characteristics of services (compared to goods), e.g., the intangible nature of services. It allows customers to configure compound services, based on their specific requirements and expectations.

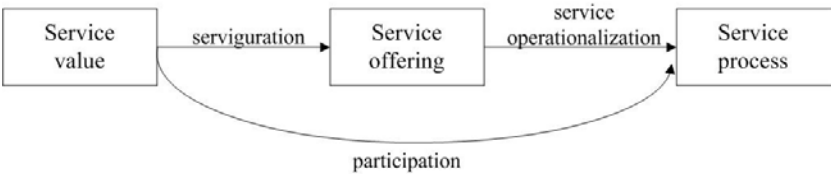


Fig. 1. Three top-level ontological distinctions to be made in a generic service ontology: the customer-value perspective, the supply-side perspective, and the joint operationalization of these viewpoints in terms of the actual service production process

On a high level of abstraction, a service ontology must embody three interrelated top-level perspectives: *service value, service offering* and *service process* (see Figure 1). The *service value* perspective describes the service from a customer's point of view in terms of a customer's needs and demands, his quality descriptors and his acceptable sacrifice, in return for obtaining the service (including price, but also intangible costs such as inconvenience costs and access time). The *service offering* perspective describes a service from a supplier's perspective; it provides a hierarchy of service components (service elements) and outcomes, as they are actually delivered by the service provider in order to satisfy customers' needs. The *service process* perspective describes how the service offering is put into operation (the *operationalization* relation in Figure 1) in terms of business processes that can be modeled using existing technologies as ebXML [1], WSFL [17] or BPEL [5]. Customers often take active part in the service production process (the *participation* relation in Figure 1).

Service configuration, or *serviguration*, is the process of defining sets of service elements (a supply-side description of services, part of the *service offering* perspective), that satisfy the customer description of his desired service (*service value* perspective). Serviguration can be split into two sub-processes: (1) transformation process between a customer description of the requested service and a supplier terminology for describing the service; and (2) defining zero or more sets of service elements that satisfy this supplier description of the requested service(s), and thus also the customer description of his requested service(s). This paper focuses on the second sub-process of the serviguration process: a task of configuring elementary services into a complex *service bundle.* It is important to understand that elements of the service offering perspective – which we model – cannot be modeled using business process modeling techniques, as the essence of value-oriented models is different from that of business process models. For a thorough explanation see [11]. Our work includes also the first sub-process of the serviguration process, a mapping task between customer requirements and available services. However, due to scope limitations, we do not discuss it in the present paper.

3 Service Terminology

In this section, we briefly review the core ontological concepts in our service ontology. For a more detailed discussion see [7, 6].

Service element represents what a supplier offers to its customers, in supplier terminology. It is what the business literature defines as *service*, a business activity (performance) of mostly intangible nature. A service element can be a composite concept; it can be decomposed to smaller service elements, as long as the smaller service elements can be offered to customers separately or by different suppliers. The business science literature discusses the notions *core service, supporting service* and *enhancing service.* A core service represents the reason for the supplier's presence on the market. For example, the core service of an insurance company is providing insurances; the core service of an airline is providing transportation. Supporting services are needed in order to enable the core service consumption. In the absence of these services, the core service consumption is no longer possible. For example room cleaning services in hotels. Enhancing services are often considered to be the elements of the service that define it and make it competitive. They increase the value of the service, or differentiate it from those of competitors [12]; the core service can nevertheless be consumed without them. An example is providing credit card holders with a "free" travel insurance. Neither supporting services nor enhancing services are offered independently of a core service. We adopt a customer-oriented definition of *core service*: a core service is not a main service that a supplier sells, but rather a main service a customer is interested in. For example, when a customer buys an airline ticket with a travel insurance, his core service is transportation, rather than insurance. However, from the insurance company's perspective, the travel insurance is a core service, since this is a core activity of that company.

Resource is either a pre-requisite for the provisioning of some service element, or the outcome (result) of a service element. We call resources *service inputs* or *service outcomes.* A resource may be the outcome of some service element, as well as the input of another service element. Very often the outcomes of a service reflect the customer benefits from a service. Resources may be of several types: *physical goods* (sometimes defined as 'those things that can be dropped on the floor'), *human resources*, *monetary resources*, *information resources* (e.g., customer information or a weather report service), *capability resources* (the ability to do something, which is of value to some actor, e.g., the ability to connect to the Internet 24x7), *experience resources* (e.g., an added value of going to Euro Disney is having fun) and *State-change resource.* The latter type requires an explanation. Services are "activities... of bringing about a desired change in – or on behalf of – the recipient of the service" [19]. Sometimes the change can be related to a property of some resource (e.g., a car's state changes in a car repair service), whereas in other services the subject of the state-change is not a resource, e.g., a passenger taking a flight undergoes a state change. In such cases the customer value of a service is a change of state: the customer *was* in Amsterdam, and now he *is* in Sydney. He pays for this change of state.

Service bundle is a set of one or more service elements that are offered together, possibly by more than one supplier. The service elements need not be related to each other; however, there will always be some logic behind the decision to bundle services. Services may be bundled because they depend on each other, to make better use of existing resources, for marketing goals, because legislation requires it and more.

4 Configuration of Services

Using the above terminology, we can describe services and service bundles in a similar way to how components and configurations are described in the knowledge systems literature. In particular, configuration is a constructive task, based on the availability of a set of predefined components, connections, and associated parameters and constraints [20, 18, 13]. We will describe service elements as components, so that a configuration process can create service bundles. Our discussion can be split into two parts: describing service elements (elementary and complex) and describing constraints on connections between service elements.

4.1 Service Element

Components, as described in the knowledge engineering literature [13, 18, 9, 10], have ports, constraints and properties. We claim that the component-based nature is inherent to services. As such, we can identify ports, constraints and properties for service elements.

Ports. Every service element has *ports* of two types: input ports and outcome ports. The provisioning of a service element requires core resources, and results in the availability of other resources. A port indicates a certain resource that is either a pre-requisite for carrying out this service element (input port), or the result of carrying out this service element (outcome port). A service element cannot be provisioned if not all required inputs are available; it results in the availability of all outcome resources. The set of all input ports, respectively all outcome ports of a service element form the element's *input interface*, respectively *outcome interface.*

Constraints. We distinguish between constraints that are internal to some service element (e.g., constraints on associated resources) and constraints on the relationships between service elements. The latter type is referred to as *functions* (see Section 4.3).

We can also identify **properties** of service elements. Some properties are generic (e.g., quality, productivity, sacrifice), whereas others are domain-specific. We found that all service properties can be described as resource properties as well. Hence, the properties of a service are expressed as the properties of its associated resources. The concept *service element* is visualized in Figure 2.

4.2 Service Bundle

A service bundle is a complex service element, that includes one or more (possibly complex) service elements. Hence, a service bundle has an input interface and

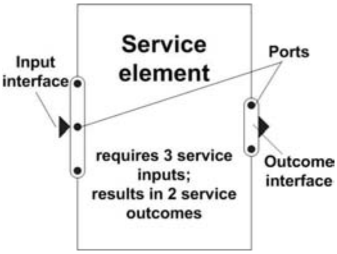


Fig. 2. Service element

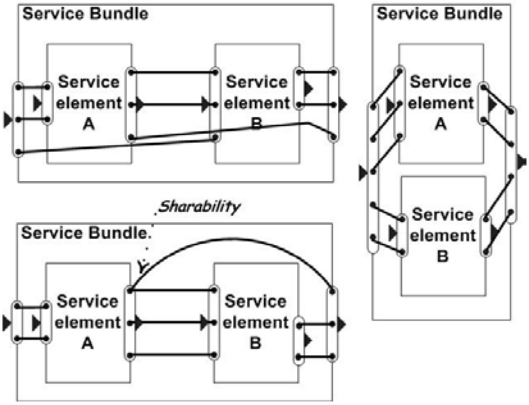


Fig. 3. Service bundles

an outcome interface. The input interface, respectively outcome interface of a service bundle, are identical to the union of the input interfaces, respectively outcome interfaces, of all service elements included in a service bundle. Two deviations from this rule exist:

(1) when resources have the *sharability* property, they may be consumed after they had already been consumed. For example a *pricing model* is a static information resource that can be used multiple times and still be available.
(2) when resources have the *compositeness* property, multiple resources of the same type may be modeled as one resource. For instance, when two service elements are bundled, and both require a *payment* input, these two inputs can be composed into one *payment* resource. Very often the price in case of such bundling is lower than the sum of both prices.

The input interface of a service bundle must provide all inputs of all service elements that are part of this bundle, unless they are provided internally (one service element may produce an outcome that is consumed as an input by a different service element). Examples of service bundles are shown in Figure 3. Links between ports mean that one port uses a resources that another port provides.

4.3 Functions

Function is a relationship that defines dependencies between two service elements. It represents a constraint on how these service elements may or may not be bundled, rather than on the service elements themselves. A function is a formula that receives two arguments of type ‘service element’ (A - the *dependee*, and B - the *dependent*), and produces as output a set of possible configurations of these two inputs. We defined the following functions (with A, B as service elements) based on business science literature [12, 14, 15, 19] and on case studies:

1. **Core/enhancing (A, B):** B is an enhancing service of A (and thus A is a core service of B). Hence service A is a main service a customer is interested

in, whereas service B: (1) is not required for the provisioning of service A; and (2) adds value to A; and (3) is an optional service element, next to A; and (4) is not offered independently. If a customer wishes to consume service element A, he is presented with the option to consume also B, but he is not obliged to consume B.
Notation: $A \rightarrow_{enh} B$
Output: {A},{A,B}

2. **Core/supporting (A, B):** B is a supporting service of A (and thus A is the core service of B). In business terms it means that A cannot be provisioned without B and that B is not offered independently. Very often B will not present value as such for customers (e.g., billing services), but yet it must be provided to enable the provisioning of A. If a customer wishes to buy service A, he is obliged to consume service B as well.
Notation: $A \rightarrow_{supp} B$
Output: {A,B}
3. **Bundled (A, B):** If a customer wishes to buy service element A, he is obliged to buy also B. This is similar to the *Core/supporting* function. However, in this case B may be offered independently, and the reason for the obligatory consumption of B is different. In the case of *bundled* services, the bundling is required due to some business logic, such as cost efficiency reasons, marketing reasons, legislation and more.
Notation: $A \rightarrow_{bund} B$
Output: {A,B}
4. **OptionalBundle (A, B):** Two services A and B are offered separately, but also as an optional combination (bundle). In most cases, the bundling of two such service elements presents added value to the supplier (e.g., lower operational costs) as well as to the customer (lower price). Unlike the *core/enhancing* function, in the *optional bundle* case service B can also be offered independently of service A.
Notation: $A \rightarrow_{optBund} B$
Output: {A},{A,B}
5. **Substitute (A, B):** The benefits presented by A (in terms of service outcomes) to a customer are also presented by B (but B possibly offers more benefits). B can therefore be bought instead of A; customers can choose which one of them they prefer.
Notation: $A \rightarrow_{subst} B$
Output: {A},{B}
6. **Excluding (A, B):** If service element A is consumed, service element B may not be consumed, for example due to business reasons (A and B are competing services, and the supplier does not want to provide them together) or legislation that prohibits selling both services together.
Notation: $A \rightarrow_{excl} B$
Output: {A}

5 Case Study: Bundling Electricity Supply with Other Services

5.1 Differentiation of the Electricity Product in Norway

Electricity becomes more and more a commodity product. Competing suppliers are offering electricity to end-customers but the price per kWh is (nearly) the same. Additionally, it is delivered according to the same standard and consumed through the same electricity socket in a customer's home.

As a result, it is for electricity suppliers hard to compete with each other. Consequently, many suppliers are seeking for ways to differentiate their product. One way to do so is to add complementary and additional services such as Internet access and home comfort management. In many cases, suppliers can use existing infrastructure and/or available business processes to deploy such extra services, so bundling these services can be done with relatively modest effort. The study presented in the next sections utilizes and exemplifies our service ontology as well as existing work on configuration theory from AI, using a project we carried out for an electricity supplier in Trondheim, Norway. First, we elicit elementary service elements, their outcomes and inputs, constraints and functions. Knowing these elements, we configure *bundles* of service elements and outcomes, in effect various combinations of the outcome 'electricity' with additional outcomes.

5.2 Energy Service Elements

Multiple suppliers offer a variety of services that can be bundled with *electricity supply* in the energy sector. Some of the services are required in order to make possible the supply of electricity, while others add value or help suppliers differentiate themselves in this market. We identify the following service elements:

- **Electricity supply**; the selling of electricity to end-consumers.
- **Electricity transmission** to end-consumers. The Norwegian law forbids electricity supply and transmission to be done by the same company.
- **Heat pump** uses electricity for space heating. It provides maintained comfort with less use of electricity.
- **Energy control system** enables controlling the temperature and switching appliances on/off. It gives the customer the possibility to reduce electricity consumption and maintain a comfortable temperature.
- **Broadband (Internet) access** is offered in a limited geographic area.
- **Hot water** for room heating and tap water requires some technical infrastructure that is available in certain regions only. It provides the same functionality as heating water with electricity, but for a lower price.
- **Remote control**: web-based control of home appliances.
- **Contracting**. Many services require a contracting service element. However, the description of most services (in our case study) is quite different from that of electricity supply and transmission. Hence, we model three contracting service elements: electricity supply contracting, electricity transmission

contracting, and a generic contracting service element that all other services can use.

- **Billing** is also a service element that many other services require; it always requires some information about the contract, and results in an invoice. We model one generic billing service element.

We identified and modeled more services, e.g., ASP-services, safety check of electrical installations and more, but we do not discuss them further for the sake of brevity. In the remainder of this section we present two detailed examples of how service elements can be modeled as components. Figure 4 shows visualizations of two more such service elements.

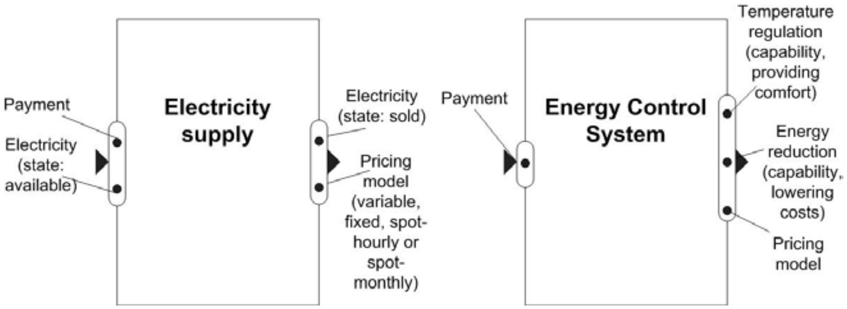

Payment
Electricity (state: available)
Electricity supply
Electricity (state: sold)
Pricing model (variable, fixed, spot-hourly or spot-monthly)
Payment
Energy Control System
Temperature regulation (capability, providing comfort)
Energy reduction (capability, lowering costs)
Pricing model

Fig. 4. Service elements: electricity supply, energy control system and contracting

Service Element: Broadband Access

Broadband Internet access is offered as an independent service (not necessarily bundled with electricity supply). The service is provided in a limited geographic region, where the required infrastructure is available.

Service Inputs:
Payment. *Type*: Monetary resource. *Compositeness*: Formula (when bundling two or more services, their payment inputs can be composed into one payment resource, based on a pricing-formula of the supplier).
Customer Information. Information about the customer, such as type of customer (household or industrial), name and address, postcode (to verify the geographic constraint of this service). *Type*: Information resource.

Service Outcomes:
Pricing Model. *Type*: Information resource. *State*: The chosen product for broadband access. Possible alternatives are *basic, regular* etc, implying differing download/upload speeds. *Sharability*: Infinite (The pricing model is determined in this service element, and then serves as input for the contracting and billing service elements. Being a static information resource, it can be consumed an infinite number of times).

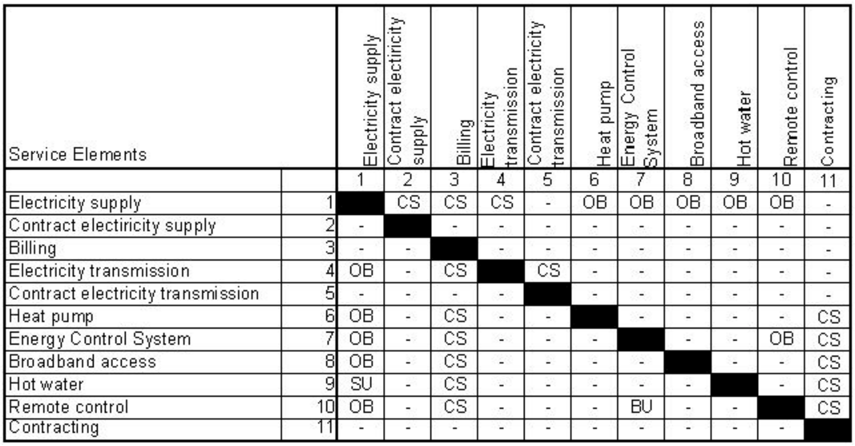
Broadband Internet. *Type*: Capability resource. *State*: Available. *Productivity*: 256/128 kbps (download/upload), 512/128 kbps or 704/256 kbps. *Medium*: wireless OR fiber optics.

Service Element: Billing
Billing is a generic *supporting service element*; it is required to enable the consumption of several other service elements.

Service Inputs:
Pricing Model: *Type*: Information resource. *State*: Defined per service element.
Customer Information: Data about the consumption for the specified customer. *Type*: Information resource.

Service Outcomes:
Invoice: *Type*: Monetary resource. *Productivity:* Frequency of invoices (monthly, four times a year, yearly).

The above description of service elements is a generalization. In reality, when we model these service elements, we have multiple *broadband access* service elements, multiple *electricity supply* service elements etc, with varying values for the same resource properties. For example, instances of the *broadband access* service element exist with differing quality levels: basic, regular or luxurious.



Service Elements		Electricity supply	Contract electiricity supply	Billing	Electricity transmission	Contract electricity transmission	Heat pump	Energy Control System	Broadband access	Hot water	Remote control	Contracting
		1	2	3	4	5	6	7	8	9	10	11
Electricity supply	1	■	CS	CS	CS	-	OB	OB	OB	OB	OB	-
Contract electiricity supply	2	-	■	-	-	-	-	-	-	-	-	-
Billing	3	-	-	■	-	-	-	-	-	-	-	-
Electricity transmission	4	OB	-	CS	■	CS	-	-	-	-	-	-
Contract electricity transmission	5	-	-	-	-	■	-	-	-	-	-	-
Heat pump	6	OB	-	CS	-	-	■	-	-	-	-	CS
Energy Control System	7	OB	-	CS	-	-	-	■	-	-	OB	CS
Broadband access	8	OB	-	CS	-	-	-	-	■	-	-	CS
Hot water	9	SU	-	CS	-	-	-	-	-	■	-	CS
Remote control	10	OB	-	CS	-	-	-	BU	-	-	■	CS
Contracting	11	-	-	-	-	-	-	-	-	-	-	■

Fig. 5. Constraints on bundling services in the energy sector

5.3 Functions: Constraints on Service Bundling

Having modeled service elements in the energy sector, we can now look at dependencies between service elements. These are formulated in terms of the functions presented in Section 4.3. Together with domain experts we created a matrix of service elements, and every slot in the matrix represents the value of a function between two service elements. We use the following abbreviations: CS (Core/supporting), OB (Optional bundle), SU (Substitute), BU (Bundled). The notation '-' means that there is no function between the two associated service elements. This matrix is presented in Figure 5. It has to be read as follows: every slot defines a function *Function(row, column)*. In other words, service elements in the rows are the first argument of a function, and service elements in a column are the second argument of a function. For example: service element *electricity transmission* has an optionalBundle function with *electricity supply* and a core/supporting function with the service elements *billing* and *contract electricity transmission*. Multiple service elements naturally have a core/supporting function with *billing* and with *contracting*. Defining the set of functions (see Section 4.3) is a conceptual modeling task, mostly based on existing business science research. Instantiating the model as done in Figure 5, on the other hand, is done by mapping domain knowledge into the structures of our service ontology.

5.4 Configuring Service Bundles in the Electricity Market

A main reason behind our study of service bundles in the energy sector is to develop offerings so that our case study partner can differentiate herself from competitors. The same methodology can also be used to offer customers the possibility to define a set of services that they are interested in. In this section we analyze how services can be bundled in a scenario in which a customer is interested in electricity, as well as in broadband Internet. Similar scenarios can be created for any of the other service elements we modeled.

The input for the configuration process includes three parts:

1. A set of all available service elements, including their associated resources.
2. A set of all functions between any pair of service elements (see Figure 5).
3. A set of initial requirements. In our scenario we require the service outcomes *electricity* and *broadband Internet*. Deriving these requirements, based on a modeling of customer requirements, is facilitated by a mapping between concepts of two perspectives in our ontology: the *service value* perspective and the *service offering* perspective. As mentioned before, this process is not discussed in the present paper due to scope limitations.

The service outcomes 'electricity' and 'broadband Internet' are the results of service elements 'electricity supply' and 'broadband access' respectively. Consequently we would like to create a service bundle with these service elements.

Electricity supply has the following functions:
$electricity\_supply \rightarrow_{supp} contract\_electricity\_supply$
$electricity\_supply \rightarrow_{supp} billing$
$electricity\_supply \rightarrow_{supp} electricity\_transmission$
$electricity\_supply \rightarrow_{optBund} heat\_pump$
$electricity\_supply \rightarrow_{optBund} energy\_control\_system$

$electricity\_supply \rightarrow_{optBund} broadband\_access$
$electricity\_supply \rightarrow_{optBund} hot\_water$
$electricity\_supply \rightarrow_{optBund} remote\_control$

Consequently, any service bundle for electricity supply needs to include also instances of the service elements *contract electricity supply, billing* and *electricity transmission*, and possibly – but not necessarily – any combination of the following service elements: heat pump, energy control system, broadband access, hot water and remote control.

Broadband access has the following functions:
$broadband\_access \rightarrow_{supp} billing$
$broadband\_access \rightarrow_{supp} contracting$
$broadband\_access \rightarrow_{optBund} electricity\_supply$

Consequently, any service bundle for broadband access needs to include also instances of the service elements *billing* and *contracting*, and possibly *electricity supply.* However, since *electricity supply* is already part of our bundle, this function adds no new possibilities. Next, for every service element that we add to the bundle due to one of the above functions, we have to check recursively whether the functions of that service element pose new restrictions or add new possibilities. These service elements are: contract electricity supply, billing, electricity transmission and contracting. Three of them (contract electricity supply, billing, contracting) have no functions. But *electricity transmission* does:
$electricity\_transmission \rightarrow_{optBund} electricity\_supply$
$electricity\_transmission \rightarrow_{supp} contract\_electricity\_transmission$
$electricity\_transmission \rightarrow_{supp} billing$

The first function need not be addressed, since electricity supply *already* is part of our bundle. Both other functions are of type *core/supporting*, so their second arguments (*contract electricity transmission* and *billing*) must be added to any bundle. Since these two service elements have no functions, the recursive check of functions terminates, unless we also consider service elements that have an *optionalBundle* function with *electricity supply.* The result of this process is that any service bundle that satisfies the above mentioned requirements must include instances of the following nine service elements: (1) one instance of *electricity supply*; (2) one instance of *contract electricity supply*; (3) one instance of *electricity transmission*; (4) one instance of *contract electricity transmission*; (5) one instance of *broadband access*; (6) three instances of *billing* (for *electricity supply*, for *electricity transmission* and for *broadband access*); and (7) one instance of *contracting.*

Assuming that multiple instances exist for all these services (differing in quality levels or in other properties), the configuration process will result in a set of possible service bundles. Any such service bundle needs to include nine service elements as mentioned. An example for such a service bundle is presented in Figure 6. For clarity, we did not model how input/outcome interfaces provide the required resources to all service elements. Instead, we modeled only the three most important resources: the service input *payment* and the service outcomes *electricity* and *broadband Internet.* As can be seen, the resource *electricity* in a

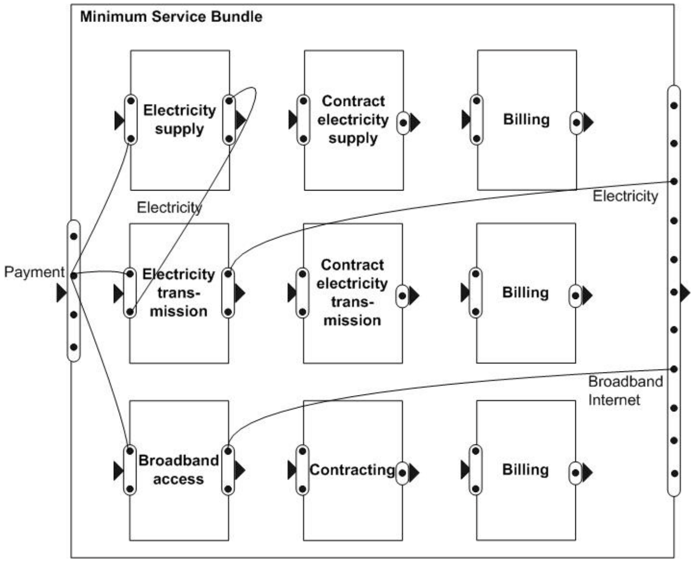


Fig. 6. A service bundle with electricity supply and broadband Internet access

certain quantity and in state *sold* is a service outcome of the service element *electricity supply*, as well as a service input of the service element *electricity transmission*. Subsequently, it is also a service outcome of the service element *electricity transmission* (and of the whole bundle), with state *transmitted*.
However, more possible service bundles exits. Instances of any of the *optionally bundled* service elements may be added to the bundle. This would require recursively adding service elements that may be required by the functions of any optional service element. For example, the service element *hot water* would require another instance of the service element *contracting*, and another instance of the service element *billing*. It is important to understand that Figure 6 is not a visualization of business processes, but a supplier description of offered services. These will be realized by business processes. There is no 1-on-1 mapping between service elements and business processes.

6 Conclusions

Services have traditionally been subject to research in business science. Discussions about services, as can be found in the business literature, are characterized by the use of natural language, which is not suitable for automated support of services. Business knowledge needs to be conceptualized and formalized as a first step towards online offering of complex service scenarios. However, scenarios in which a variety of services is offered by multiple suppliers require more than a formal description of business knowledge. For software to define sets of independent services, it is necessary to describe formally business knowledge on (1) what services are; and (2) under which circumstances services may be sold together. In other words, services need to be described as components, and their interdependencies as constraints. Our service ontology supports these two necessities through a formal conceptualization of business knowledge, based on structures from the knowledge management literature.

In this paper we have presented examples of modeling services from the energy sector, as part of a large-scale analysis of this sector. Based on our component-based description of services in the case study at hand, it seems possible to have software that configures *service bundles* that satisfy customer criteria. Currently, we are building in the EC-IST funded project OBELIX a tool that (1) allows modeling of elementary service elements and supporting constructs, and (2) configures service bundles using these elements as well as customer requirements.

The case study we have presented in this paper is part of a larger case study in the energy sector, in which we use other ontologies as well. We also investigated how the subjective customer description of services can be mapped into an objective supplier description of services; we then combined that process with the configuration of services. Among the results were new insights into the possible service bundles that are commercially viable for service suppliers, as well as feasible in the sense of interdependencies between services. Hence, our service ontology can be used not only for a customer-triggered process of configuring services, but also for an analysis of possible business scenarios.

References

1. *ebXML website.* 2003. http://www.ebXML.org
2. *eCl@ss website.* 2003. http://www.eclass.de
3. *UNSPSC website.* 2003. http://www.unspsc.net
4. An assessment of services trade and liberalization in the united states and developing economies. *World Trade Organization, Council for Trade in Services*, 31 March 2003. Document TN/S/W/12, available via http://www.wto.org/english/docs_e/docs_e.htm, last visited February 2004.
5. Tony Andrews, Francisco Curbera, Hitesh Dholakia, Yaron Goland, Johannes Klein, Frank Leymann, Kevin Liu, Dieter Roller, Doug Smith, Ivana Trickovic, and Sanjiva Weerawarana. *Business Process Execution Language for Web Services Version 1.1.* 2003. http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/, last visited February 2004.
6. Z. Baida, H. Akkermans, A. Bernaras, J. Aguado, and J. Gordijn. The configurable nature of real-world services: Analysis and demonstration. In *The First International Workshop on e-services*, pages 46–56, Pittsburgh, PA, 2003. Carnegie Mellon University.
7. Z. Baida, H. Akkermans, and J. Gordijn. Serviguration: Towards online configurability of real-world services. In *Proceedings of the Fifth International Conference on Electronic Commerce (ICEC03)*, pages 111–118, Pittsburgh, PA, 2003. ACM.
8. Leonard Berry. Perspectives on the retailing of services. In Ronald W. Stampfl and Elizabeth C. Hirschman, editors, *Theory in Retailing: Traditional and Nontraditional Sources*, Chicago, 1981. American Marketing Association.

9. Pim Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, Universiteit Twente, Enschede, NL, 1997.
10. W. N. Borst, J. M. Akkermans, and J. L. Top. Engineering ontologies. *International Journal of Human-Computer Studies*, 46:365–406, 1997.
11. J. Gordijn, H. Akkermans, and J. van Vliet. Business modelling is not process modelling, In: Conceptual modeling for e-business and the web (ECOMO-2000), Springer-Verlag, LNCS 1921, Salt Lake City, USA, October 9-12, 2000, pp. 40-51., 2000.
12. C. Grönroos. *Service Management and Marketing: A Customer Relationship Management Approach, 2nd edition*. John Wiley & Sons, Chichester, UK, 2000.
13. T. Gruber, G. Olsen, and J. Runkel. The configuration design ontologies and the VT elevator domain theory. *International Journal of Human-Computer Studies*, 44:569–598, 1996.
14. H. Kasper, P. van Helsdingen, and W. de Vries jr. *Service Marketing Management: An International Perspective*. John Wiley & Sons, Chichester, UK, 1999.
15. P. Kotler. *Marketing Management: Analysis, Planning, Implementation and Control, 6th edition*. Prentice Hall, Englewood Cliffs, NJ, 1988.
16. Theodore Levitt. *Your Factories in the Field: Customer Service and Service Industries*, chapter 3, in Marketing for Business Growth, pages 51–70. McGraw-Hill, New York, 1973.
17. Frank Leymann. *Web Services Flow Language (WSFL 1.0)*. IBM, 2001. http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf, last visited February 2004.
18. C. Löckenhoff and T. Messer. Configuration. In J. Breuker and W. Van de Velde, editors, *The CommonKADS Library for Expertise Modelling — Reusable Problem Solving Components, Chapter 9*, Amsterdam, The Netherlands, 1994. IOS Press.
19. C. Lovelock. *Services Marketing, People, Technology, Strategy, 4th edition*. Prentice Hall, Englewood Cliffs, NJ, 2001.
20. S. Mittal and F. Frayman. Towards a generic model of configuration tasks. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 1395–1401, San Francisco, CA, 1989. Morgan Kaufmann.
21. W. Earl Sasser, R. Paul Olsen, and D. Daryl Wyckoff. *Management of Service Operations: Text, Cases, and Readings*. Allyn & Bacon, 1978.
22. Steffen Staab, Wil van der Aalst, V. Richard Benjamins, Amit Sheth, John A. Miller, Christoph Bussler, Alexander Maedche, Dieter Fensel, and Dennis Gannon. Web services: Been there, done that? *IEEE Intelligent Systems*, 18(1):72–85, 2003.
23. V.A. Zeithaml and Mari Jo Bitner. *Services Marketing*. MGraw-Hill Companies, New York, NY, 1996.