Running head: CUSTOMER NEEDS IN E-SERVICES

How e-Services Satisfy Customer Needs: a Software-aided Reasoning

Ziv Baida[1,2], Jaap Gordijn[3], Hans Akkermans[3], Hanne Sæle[4] and Andrei Z. Morch[4]

[1]Faculty of Economics and Business Administration, Free University Amsterdam, The Netherlands
[2]e-Rational, Amsterdam, The Netherlands
[3]Faculty of Sciences, Free University Amsterdam, The Netherlands
[4]SINTEF Energy Research, Norway

ABSTRACT

We outline a rigorous approach that models how companies can electronically offer packages of independent services (service bundles). Its objective is to support prospective Website visitors in defining and buying service bundles that fit their specific needs and demands. The various services in the bundle may be offered by different suppliers. To enable this scenario, it is necessary that software can reason about customer needs and available service offerings. Our approach for tackling this issue is based on recent advances in computer and information science, where information about a domain at hand is conceptualized and formalized using ontologies and subsequently represented in machine-interpretable form. The substantive part from our ontology derives from broadly accepted service management and marketing concepts from business studies literature. In earlier work, we concentrated on the service bundling process itself. In the present chapter, we discuss how to ensure that the created bundles indeed meet customer demands. Experience of Norwegian energy utilities shows that severe financial losses can be caused when companies offer service bundles without a solid foundation for the bundle-creation process and without an in-depth understanding of customer needs and demands. We use a running case example from the Norwegian energy sector to demonstrate how we put theory into practice.

KEYWORDS: electronic services, conceptual model, knowledge management, ontology theory, requirements engineering, service industry

How e-Services Satisfy Customer Needs: a Software-aided Reasoning

INTRODUCTION

More and more businesses nowadays offer their services via the Internet, either parallel to or instead of traditional physical channels. Statistics show an immense growth in the percentage of households with Internet access that actually shop online; from 27% in 1998 to nearly 50% in 2000 (Xue *et al.*, 2003). Almost 30% of Internet users in the EU use online banking services, with the Nordic countries as leaders; nearly 65% of Internet users in Finland use online banking (Centeno, 2003). Airlines sell more and more tickets online instead of through traditional travel agencies; check-in is performed online rather than at the check-in counter in the airport. Companies as DHL and FedEx allow customers to follow their shipments through a so-called track-and-trace system. Governments are considering online voting. These are all examples showing the dominant and growing role and importance of e-services in a variety of industries.

Online service offerings introduce a new challenge, with which traditional service suppliers do not have to deal. It no longer is sufficient that only service personnel understands customer needs; if a supplier wishes to offer customized services through an automated online process, software must be able to reason about these customer needs and about the possible service offerings satisfying such needs, so that the whole process can be provided online. The need for an automated process becomes even greater when a customer wants to buy a service bundle (Grönroos, 2000), a package of more elementary services, , which may be offered by multiple suppliers. Each supplier offers its added value, and together suppliers provide a complete answer for a customer need. In such a case, software should be able to decide whether and how to combine services of multiple suppliers into one service bundle.

Our study on creating customer-driven service bundles aims at this new challenge. We present a method for formalizing customer needs and available service offerings, and we relate the two to each other. We do not directly address the problem of how to elicit and understand customer needs (although, as we will show, our method helps gain insights into these needs) but focus on the issues of conceptualizing and formalizing customer needs, such that software can configure service bundles satisfying customer needs.

Our research uses well-known and accepted knowledge, concepts, ideas, and terminology from business research literature (Grönroos, 2000; Kotler, 1988; Zeithaml *et al.*, 1990) to describe services from a supplier perspective as well as from a customer perspective. The idea is to conceptualize and formalize well-known business research concepts, not to invent new ones. Additionally, we use practices and ideas from computer science as a means to process this knowledge in order to enable automated support for the bundling process of customer-driven service bundles. One of these practices is the use of an ontology, which is a formal, shared conceptualization of something we assume to exist (Borst, 1997; Quine, 1961), in our case, needs and e-services. The unique contribution of this chapter is in the combination of well-known business research terminology on services with the modeling and conceptualization rigor of computer science.

The work presented in this chapter is not limited to e-services, but can be applied to traditional services as well. Nevertheless, our work is of much greater importance for e-services, since the realization of e-service offerings requires automating processes that may otherwise be performed

in the minds of service personnel. For e-services realization, it is absolutely necessary that business knowledge is conceptualized, formalized, and made machine-readable and machine-processable. This is what we aim to achieve in our work.
Our method consists of three steps to be performed in advance, followed by one runtime step to be performed each time a customer wants to create (design) a bundle for need satisfaction:

1. Identify and model customer needs and demands;
2. Identify and model available services;
3. Identify and model relations between demands and available services;
4. Create service bundles out of available services, based on customer needs and demands.

Whereas our earlier work (Akkermans *et al.*, 2004; Baida *et al.*, 2003a; Baida *et al.*, 2004b); focused on steps 2 and 4 of the presented method, in the current chapter we present the whole method, and focus on steps 1 and 3.
In the remainder of this chapter we will use a case study in the energy domain to present our work. After introducing the energy domain, we discuss our research approach, followed by a discussion of a service ontology. We then present a four-steps method for ensuring that e-services are demand-driven and discuss it using examples from the energy case study. Finally, we review related work, and present our conclusions.

CASE STUDY: BUNDLING ELECTRICITY SUPPLY WITH OTHER SERVICES

Since the deregulation of the electricity market in Norway in 1991, production and trade of electric energy have been liberalized, while the transmission and distribution are maintained as regulated monopolies. Nowadays, after evolving for almost 15 years of deregulation, the Norwegian power market becomes mature. The electricity generation and supply sectors are characterized by a fierce competition, due to which the difference in electricity retail prices per kWh between different suppliers is diminishing. Also in other European countries power is shifting from suppliers to customers, and more and more end-user customers in Europe are able to choose a preferred electricity supplier.
Commercially, one of the disadvantages of the electricity product is that for power supply companies it is hard to distinguish themselves, due to the anonymous nature of this product: electricity from different suppliers is delivered according to the same standard, with the same physical characteristics, and is consumed through the same electricity socket in a customer's home. Therefore, companies face difficulties in competing with each other. Consequently, many suppliers are seeking for ways to improve marketing via differentiation of their product, to increase their market share. One way to differentiate is to offer additional services such as Internet access, (software) application service provisioning and home comfort management. Another way to improve marketing is to create more complex and elaborated electricity retail contracts, which are more beneficial to customers because they fit better to their needs. At the same time, choosing the best electricity contract becomes a demanding task for electricity consumers.
Many of the additional services can be ordered and provisioned via the Internet. Moreover suppliers can use existing infrastructure and/or available business processes to deploy such extra services, so bundling these services with the traditional electricity product can be done with relatively modest effort. Experience however shows that the bundling of services without sound logical fundaments of the bundles design process and disregarding customers' demands may cause severe financial losses, as can be seen by the example of KanKan (Flæte & Ottesen, 2001).

KanKan was launched on January 23rd 2001 as a new market offer of one of the biggest Distribution System Operators in Norway. It was marketed as an integrated bundle of services, including electricity supply and transmission, Smart Home features, home insurance, telephone and an ISP service. Despite the expectations and costly market campaigns, very few households showed interest in the new service offering. After several attempts to revise the concept, it was removed from the market (Flæte & Ottesen, 2001; Martinussen, 2002). Several reasons for the failure were identified later; misunderstanding of customer needs and meeting them in product offers was the most visible one. The need for such a solid and formal foundation for a successful online process is the driving force behind our study in the energy sector. Furthermore, the KanKan example highlights the necessity for evaluation methods for the feasibility of offering service bundles, a topic which we have addressed in Baida *et al.* (2004a). In this chapter, therefore, we focus on the aspect of customer demands.

NOTES ON RESEARCH METHOD AND DESIGN

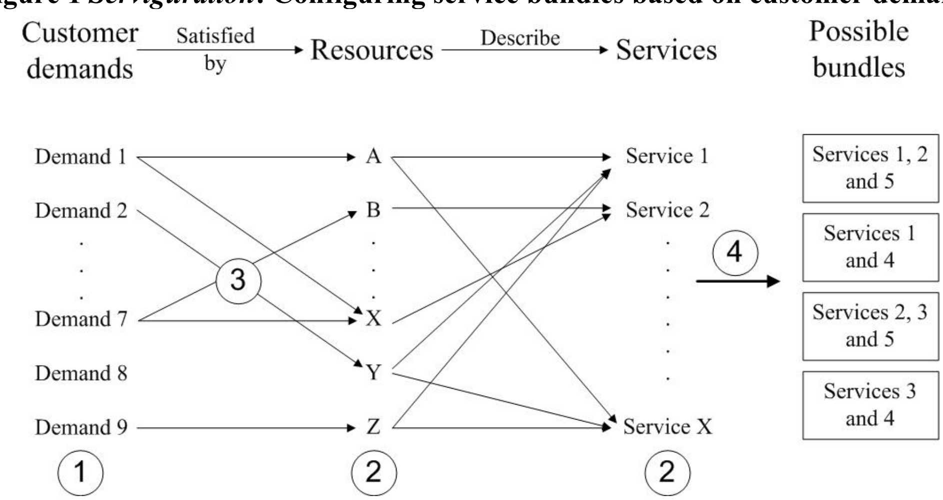
Our research approach represents a departure from traditional quantitative as well as qualitative modes of scientific research in information systems (IS) on several scores. First, the nature and role of theory; we employ formal ontology as a device for rigorous theory articulation. Ontologies are formal conceptualizations of a real-world domain such that they have a computational representation that is fit for automated reasoning. This work is much helped now that there are international standards such as RDF and OWL for knowledge representation on the Web (developed in the context of W3C's Semantic Web effort; OWL stands for Ontology Web Language and was finalized in February 2004). As theories, ontologies are formal (in a logical and/or knowledge-based inferencing sense) yet typically are not expressed in the variable and measure parlance of the common quantitative modes of social and business research (although, of course, this is not strictly excluded). So, usually, ontologies are formalized qualitative theories concerning conceptual structures shared by a community of practice in a domain.

Yet, this does not imply (at least not necessarily) that they are congruent with the interpretivist or naturalist perspectives common in qualitative research. Ontologies are intended to be reusable (this is the typical computer science term) (i.e., generalizable across other settings, contexts, and applications). Therefore, they formalize the agreed-upon (explicit or more often implicit) common understanding in a domain. For example, the ontology partly discussed in the present chapter only reflects and formalizes consensus aspects of service management and marketing as, for example, typically found in textbooks; it does not attempt to express the latest issues as debated in academic literature on services where there is no consensus, nor does it represent highly domain-specific or even organization-specific elements that one will undoubtedly encounter in any thick-description field empirical study. This implies a clear difference in the resulting theory from a strict interpretivist or naturalist perspective. Ontology is better seen as a model-based approach, whereby the quality and success of the model is assessed in terms of whether it is good enough to help in problem solving, as posed by the research goals. This notion of a model-based stance that is different from the standard fare in both quantitative and qualitative approaches has already been recognized and debated a long time ago in the knowledge systems literature (Ford & Bradshaw, 1993; Schreiber *et al.*, 2000) and references therein.

Further, qualitative and quantitative approaches have in common that they (often tacitly) assume that scientific aims lie in (different forms of) explanation. In contrast, our ontology approach is more tailored toward problem solving and innovation in business and industry practice. Thus, its aim is closer to what Hevner *et al.* (2004) call design science in IS. We mention in passing that, based on previous research, engineering science, and industry experiences, we would take issue with some of these authors' proposed guidelines for academic quality design research, in particular design as a search process and as an (instantiated) artifact, but this is beyond the scope

of the present chapter. But certainly in e-business and e-service research, where the field is in a constant state of change and emergence, research goals that go beyond observation, measurement, statistical-variable, or qualitative-interpretive explanation are of prime importance.
All this does have implications for the empirical and test/validation parts of research studies in IS. Ontologies can be tested by computer tooling, modeling, simulation, and analysis. This establishes what is sometimes called their computational adequacy and some aspects of their theoretical adequacy (soundness, consistency, completeness). Their empirical, epistemological adequacy can be tested by (as in our research) case studies in the field. Given the different nature and role of our approach to theory formation, such case studies do not sit well with the conventional typology of exploratory, descriptive, or explanatory case study. They serve a dual goal. On the one hand, they help validate (part of) generalizable ontological theory. To this extent, they might be viewed as tending toward being explanatory (although not necessarily in terms of causal explanation). On the other hand, however, they aim at helping to solve problems and achieve goals, as specifically perceived by our partners or clients in the study, which are, in the present case, not of an explanatory but of a business development and design nature. A consequence of this positioning of our empirical research is that case study design is not along the traditional lines of external-observer style empirical research but has much more in common with action research.

Figure 1 *Serviguration*: Configuring service bundles based on customer demands



FROM CUSTOMER NEEDS TO E-SERVICES

In this chapter we present a four-step approach (see Figure 1) to find alternative bundles of e-services that satisfy customer needs. Our approach is based on the following key ideas:

1. A service can be seen as a bundle of benefits (Kasper *et al.*, 1999), which satisfy customer needs.
2. When customers buy products (services or goods), in fact they are not interested in the products themselves, but in the benefits – the value – that these products presents for them (Lancaster, 1966; Teare, 1998). These benefits are satisfiers of customer needs.
3. A customer view on services differs from a supplier view (Vasarhelyi & Greenstein, 2003); thus two service descriptions are required for automated service provisioning. Typically, a supplier description is required for selecting

and comparing service instances. A customer description is required in order to decide which available services fit specific customers.

4. Services differ from goods in their intangibility (Grönroos, 2000; Kasper et al., 1999; Kotler, 1988; Lovelock, 2001; Zeithaml & Bitner, 1996). As a result, services cannot be described by their physical properties – as is the case with goods – so that customers and suppliers can refer to them unambiguously. Services therefore need to be described differently. We describe services by the benefits (value) they provide, and by the sacrifices (value) they require.

In short, with the help of business experts, we first model needs and demands of customers in a given sector, and then describe available services. Since customer demands are satisfied by providing some customer value, we identify relations between demands and outcomes of services (*resources*, as we call them), that reflect a customer's benefits from a service. When searching for possible services or service bundles that satisfy customer needs, demands (described possibly by subjective quality criteria) are used as selection criteria for resources (benefits, described by objective quality criteria). In other words, instead of saying "demand X can be satisfied by service Y" we say "demand X can be satisfied by resource Y", and we then search for all services that provide resource Y. This is possible because resources are descriptors of services. Hence, selecting specific demands implies not only selecting certain resources but also certain services that must or can be part of a bundle. Then, based on business knowledge on inherent dependencies between services (Baida et al., 2004b), other services may be included in bundles, or substituting services may be suggested as solutions. The causal chain, from needs and demands through resources to services, ensures that the offered service bundles will, indeed, meet customer needs.

Formalizing Business Knowledge Using a Service Ontology

We formalize business knowledge on services using a service ontology (Baida, 2006; Baida et al., 2003a). On a high level of abstraction, our service ontology embodies two inter-related top-level viewpoints or perspectives: service value and service offering.

The service value perspective (see Figure 2) captures knowledge about adding value. It represents a customer viewpoint on value creation by expressing customer needs, expectations, and experiences, and is driven by a customer's desire to buy a certain service of a certain, often vaguely defined quality, in return for a certain sacrifice (including price, but also intangible costs such as inconvenience costs, and access time).

The service offering perspective, in contrast, represents the supply-side viewpoint; it describes service components (a core service and supplementary services) and outcomes, as they are actually delivered by the service provider in order to satisfy customers' needs.

The service value perspective and the service offering perspective will be presented shortly in the following subsections.

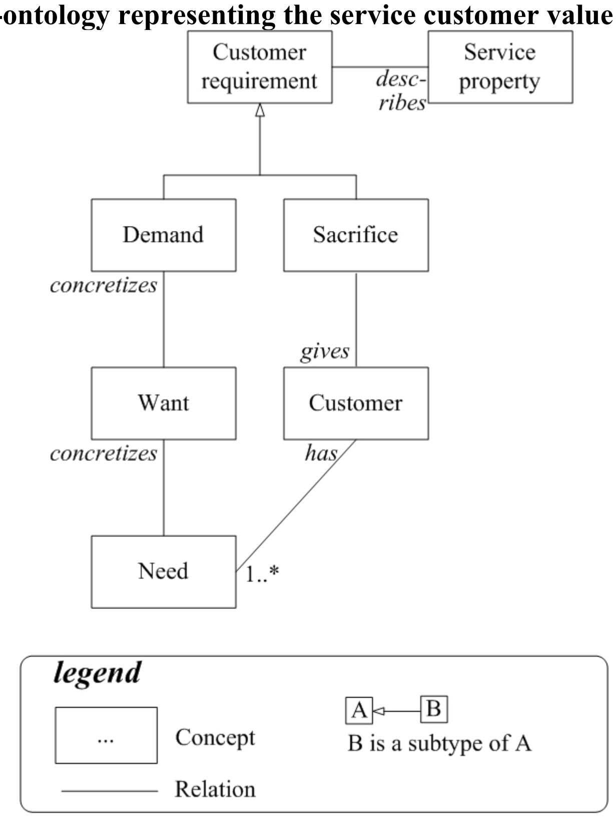
Service Value Perspective

The sub-ontology representing the service value (customer) perspective is sketched in Figure 2. Its main concepts are discussed below.

Needs, wants and demands. The starting point for the discipline of marketing, whether it refers to services or not, lies in the human needs and wants (Kotler, 1988). The term *need* refers to what humans need and want (to buy) and is quite straightforward. A formal definition is given by Kotler (1988), who distinguishes needs, wants, and demands:

- A human need is a state of felt deprivation of some basic satisfaction.
- Wants are desires for specific satisfiers of these deeper needs.
- Demands are wants for specific products that are backed up by an ability and willingness to buy them.

Figure 2 Service sub-ontology representing the service customer value perspective



Needs are often vague; the need for financial security, for example, can be interpreted in many ways. Customers concretize their needs by transforming them into wants and demands, based, for example, on their exposure to existing services and to marketing campaigns. In many cases, when a customer is interested in some service, he or she has already transformed needs into wants and demands. As a matter of fact, the customer already has a solution in mind for his or her need (e.g., indoor comfort [need]; lighting [want]; energy supply [demand]).

Sacrifice. The customer's long-term sacrifice includes the price of the service as well as relationship costs. These can be direct (e.g., investment in office space, additional equipment), indirect (related to the amount of time and resources that the customer has to devote to maintaining the relationship), or psychological costs (lack of trust in a service provider; unpleasant sensory experiences such as noise) (Grönroos, 2000) (e.g., time spent waiting to be served, travel costs, switching costs from one supplier to another).

Service quality. Service quality is the degree and direction of the discrepancy between a customer's expectations and the perception of the service (Bigné *et al.*, 1997). Customer expectations embrace several different elements, including desired service, predicted service, and a zone of tolerance that falls between the desired and adequate service levels (Berry & Parasuraman, 1991). Expectations are based on word-of-mouth communications, personal needs, past experiences, and external communications from service providers (Zeithaml et al., 1990). At least two widely accepted generic methods for defining service quality exist: that of the Nordic school (Grönroos, 2000) and that of the North American school (SERVQUAL) (Zeithaml et al., 1990). Nevertheless, quality definition is domain- and market-specific (e.g., high level of reliability, highly individualized service, and fancy conference location).

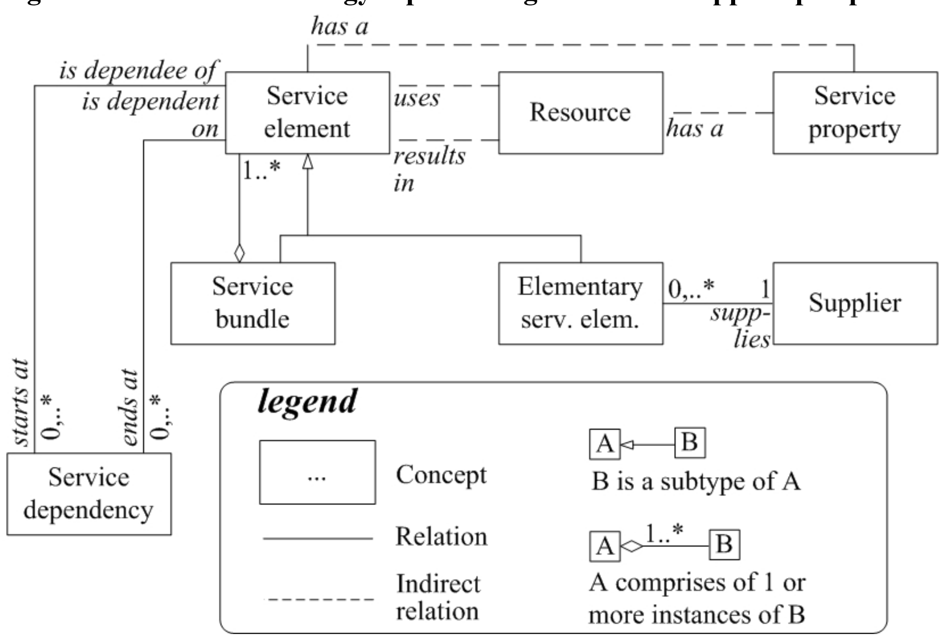
Next to quality description, other criteria also may play a role (e.g., quantitative description, or time: when the service should be provided). For this reason, we have introduced the concept **service property** in our ontology; *service quality* is described as *service properties*.

Similarly to demands, also sacrifices may be described by service properties. We refer to demands and sacrifices as **customer requirements**. Service quality, technically speaking, is a property of a customer requirement. In the rest of this chapter, whenever we use the term *desired quality*, we refer also to non-qualitative service properties.

Service Offering Perspective

The service offering (supplier) perspective, lengthily discussed in Baida et al. (2004b), describes how a business intends to add value (see Figure 3). It is centered on the concept of **service element**, which is what the service marketing literature defines as business activities, deeds and performances of a mostly intangible nature (Grönroos, 2000; Kasper et al., 1999; Kotler, 1988; Zeithaml et al., 1990). We showed in Akkermans et al. (2004) that service bundling can in fact be seen as a component configuration task, once the business essence of a service is described with constructs from configuration theory. Configuration is a constructive task, where predefined components are assembled (configured) into a larger, complex component, based on the availability of a set of predefined connections, and associated parameters and constraints (Gruber *et al.*, 1996; Löckenhoff & Messer, 1994; Mittal & Frayman, 1989).

Figure 3 Service sub-ontology representing the service supplier perspective

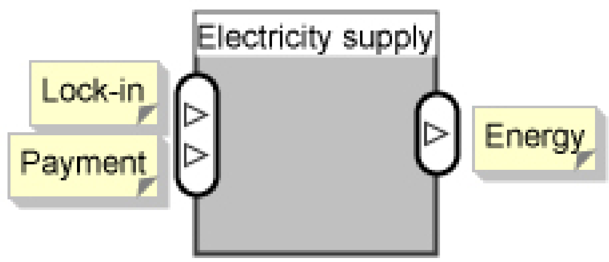


A service element is a business activity that involves the exchange of values between the actors involved. Hence, it requires a set of *service inputs*, and results in the availability of a set of *service outcomes*. Very often, the outcomes of a service reflect the customer benefits from a service, whereas the customer sacrifice is expressed as service inputs (e.g., payment). Service inputs and service outcomes are referred to as **resources**. Resources are described using objective, measurable parameters. For example, the service element *broadband Internet access* has an outcome resource *broadband Internet capability* with properties *download speed* and *up-load speed*, specified in Kbps. Hence, the resource description provides the objective and measurable benefit of a service; this objective benefit may be interpreted differently by customers who have differing expectations and quality perceptions, leading to their subjective value perceptions of the same service.

Service elements can be offered as a bundle and thus form a complex service element. To facilitate automated reasoning about bundling, the **service dependency** is used; it is a relation between service elements. For instance, a substitute dependency between elements A and B represents that service element B is a substitute service for service A (but not necessarily vice versa).

Figure 4 is an example service element from our energy study – the supply of electricity. The service is described by its resources. Two service inputs are required to provide this service (payment and lock-in, a commitment to consume this service for a predefined period), and it results in the availability of one service outcome (energy of type electricity).

Figure 4 Example service element: electricity supply



Relating the Service Value and Service Offering Perspectives

The process of service configuration – Serviguration (Baida, 2006; Baida et al., 2003a) – spans both perspectives: service value and service offering. Serviguration is the process of defining bundles 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 (see Figure 1) can be split into four steps: (1) identify and model customer needs and demands; (2) identify and model available services; (3) identify and model relations between demands and available services; and (4) use knowledge of steps 1, 2 and 3 to create service bundles out of available services, based on a given set of customer needs and demands. Whereas our earlier work concentrated on the steps 2 and 4 (Baida et al., 2004a), the present chapter concentrates on steps 1 and 3.

STEP 1: IDENTIFY AND MODEL CUSTOMER NEEDS AND DEMANDS

Understanding customer needs has been acknowledged by service marketing and service management researchers as an important early phase in business initiatives (Aschmoneit & Heitmann, 2002; Kotler, 1988; Mentzer *et al.*, 1997; Teare, 1998). But also in the field of Requirements Engineering (RE), a sub-discipline of computer science, significant effort has been put into understanding stakeholder needs to be satisfied by information systems (Liu & Yu, 2001; Mylopoulos *et al.*, 2001; Mylopoulos *et al.*, 1999; Sharp & Galal, 1999; Van Lamsweerde, 2000). A specific contribution of RE is Goal Oriented Requirements Engineering (GORE). In GORE, needs are called goals, and formal and semi-formal modeling techniques are used to model goals and relations between these. We employ these techniques to represent and to reason about needs. The advantage of doing so is that this enables us to reuse existing mechanisms for reasoning about such needs.

Table 1 Customer needs, wants, and demands for the energy utility TrønderEnergi

Customer Needs	Customer Wants	Customer Demands
Indoor comfort (H,I)	Lighting (H,I) Home services (cooking, washing etc.) (H) Comfort temperature (H,I)	Energy supply (H,I) Hot tap water (H,I) Room heating (H,I) Air conditioning (H,I)
	Energy regulation for budget-control (H,I)	Energy regulation for budget control (H,I), with different characteristics (manual / automated, on-site regulation / location-independent)
	Temperature regulation for increased comfort (H,I)	Temperature regulation (H,I) with different characteristics (manual / automated, on-site regulation / location-independent)
Social contacts and Recreation (H) Business contacts (I)	Communication (H,I)	Telephone line (H,I) Mobile phone line (H,I) Internet (broadband) (H,I) Email facilities (H,I)
Safety (H,I)	Increased security (H,I) Reduced insurance premium (H)	Safety check of electrical installation (H) Internal control of electrical installation (I)
IT support for business (I)	IT-services (I)	ASP-services (I) Hardware (I) Software (I)

In the first step of our method, we identify and model customer needs. Needs identification has been studied by marketing researchers (Kärkkäinen & Elfvengren, 2002; Kotler, 1988; McCullough, 2002; Murthi & Sarkar, 2003; Reynolds & Gutman, 1988; Teare, 1998), and is beyond the scope of our study. Instead, we consider customer needs to be known in advance by business experts. We then use need hierarchies to model these needs in accordance with our service ontology (needs, wants, and demands). Table 1 presents our hierarchy of needs, wants, and demands for the energy utility at hand. The notations H/I refer to the customer type: household or industrial. As can be seen from Table 1, some demands relate to concrete services (e.g., a demand for a mobile phone line), while others are more abstract when a customer does not necessarily know which service can satisfy his or her need, or when a diversity of solutions exits (e.g., the demand *temperature regulation* does not specify a concrete service; it can be satisfied by a variety of services).

Table 1 shows examples of needs, wants and demands, as we modeled in the energy study. As can be seen from the table, customers specify demands in their own terminology (e.g., 'room heating') or in supplier terminology (e.g., 'telephone line'). The latter happens when customers are already familiar with available services that can satisfy their needs. In our study, the energy utility TrønderEnergi wanted to explore possible ways to bundle electricity supply with other (not energy related) services, such that the bundles provide a good solution for customer needs. Therefore, the list of needs, wants and demands presented in Table 1 is not complete; it includes only those needs, wants and demands that TrønderEnergi considered to satisfy through existing or new service offerings.

Customer requirements for services are captured by (1) needs, wants, and demands; and (2) acceptable sacrifice. Each may further be described by quality criteria or by other service properties. Demands often describe the functionality of a desired solution, whereas the desired quality prescribes the expected performance-level of a service. Hence, the desired quality describes a certain level that applies to demands. The acceptable sacrifice captures the price, switching costs, psychological costs, and more to be paid for satisfaction of a need. Two important remarks have to be made:

- While our discussion in this chapter concentrates on deriving a set of desired service outcomes (resources) based on customer demands, we use the same mechanisms also to transform the customers' acceptable sacrifice (in customer terminology) to a set of acceptable service inputs (in supplier terminology).
- Conceptually, resources provide solutions for demands. Hence we discuss relations between demands and resources. However, due to computational considerations the service ontology relates the concept 'demand', through its super-type 'customer requirement', to the concept 'requirement expression'. The latter is related to a 'design element', the super-type of 'resource'.

Reasoning about Customer Needs

Often when customers first indicate requiring something, only partial (or no) knowledge of their *concrete* demands exists. We then need (1) to reason about relations between needs, wants and demands, and derive concrete demands based on more abstract needs, and subsequently (2) to match between these demands and available service offerings of service suppliers. In the rest of this section we show how we perform the first of these reasoning processes.

The relation between needs, wants and demands can be described by a hierarchy, "a structure by which classes of objects are ranked according to some subordinating principle" (Stephens & Tripp, 1978). Need hierarchies comprise of three levels of aggregation, using the above definitions of needs, wants and demands as a subordinating principle.

Similar hierarchies have been used in the field of Goal Oriented Requirements Engineering

(GORE) to transform high-level organizational needs to concrete system requirements (Donzelli, 2004). *Needs* capture the answer for the question why a service (either an elementary one or a service bundle) is offered. Similarly, in system/software design *goals* represent why a system/software is needed.

Similar to customer needs, also goals are defined at different levels of abstraction. They capture the various objectives that the system under consideration should achieve (Van Lamsweerde, 2000, 2001). Unlike GORE literature on goal hierarchies (Fuxman *et al.*, 2003), the marketing literature discusses hierarchies (of needs) (Kotler, 1988) without providing well-defined relations between elements in the need hierarchies, while such well-defined relations are required for software reasoning about needs. We fill this gap by introducing AND/OR/XOR refinements. An AND decomposition means that all siblings of a higher-level object (need, want) must be satisfied to satisfy the higher-level object. An OR decomposition means that a higher-level object can be satisfied by satisfying an arbitrary number of its siblings. A XOR decomposition means that exactly one of the siblings of a higher-level object must be satisfied to satisfy the higher-level object. These constructs can be combined, for example need N1 may be decomposed into wants W1, W2, W3 and W4 as follows: (W1 AND W2) XOR (W3 AND W4).

We model need hierarchies similar to goal trees. In our case hierarchies are directed graphs, rather than trees, because a demand or want may be related to more than one want or need respectively, so multiple paths may exist between two nodes, which is not allowed in trees. Needs are the top level nodes of the graph; then come wants; and finally demands are leafs. AND/OR/XOR refinements describe the relations between a node in the hierarchy (graph) with related nodes in an adjacent level of the hierarchy. Edges that connect nodes have the semantics “concretized by”. This relation does not apply to nodes of the same level, because they have the same level of granularity. Therefore we do not connect nodes of the same hierarchical level. Using this technique and knowledge that business experts possess, we can reason about how an abstract customer need can be specified by more concrete demands, for which a solution (satisfier) can be searched. The left part of Figure 5 presents a visualization of part of Table 1 as a need hierarchy.

Our experience from studies in the energy sector, the health sector (Baida, 2006) and online service provisioning (Baida *et al.*, 2003b) shows that the use of above refinement structures requires adding a *context* dimension, since customer needs (or: stakeholder needs, as in De Bruin *et al.* (2002)) differ per customer type, and thus the refinement changes per customer type. Different needs, wants, demands and their decompositions may apply to different customer types. In fact, per customer group (or: per stakeholder) we may define a separate need hierarchy. Customer grouping criteria may differ per case. Examples are the nature of consumption (e.g., households vs. industrial customers), the customer’s role (e.g., a patient vs. an informal carer of that patient) or the customer’s age group (e.g., teenagers typically have a different interpretation of their needs than adults). For example, the customer want for ‘communication’ can be refined to several demands, including (landline) telephone line, mobile phone line and Internet access. Whereas one customer may require a landline, another may want Internet access and a mobile phone line, and no landline. Consequently, reasoning on need satisfaction (i.e., which service can satisfy a customer want for communication) should be done on the level of customer demands rather than on the level of (more abstract) customer wants or needs. Note that quality criteria also typically describe demands; wants or needs are often too abstract to be described by some well-defined desired quality criteria.

STEP 2: IDENTIFY AND MODEL AVAILABLE SERVICES

In step two of our method, we use the service offering perspective of the presented service ontology to model available services of suppliers. We describe services by their resources – their required inputs and their outcomes. Our study of the energy domain involves a group of financially independent enterprises that provide a variety of services. Together with business experts, we investigated and modeled services, including electricity supply, electricity transmission through a high voltage network, hot water supply (for room heating), energy control (for controlling the temperature; that is, to lower the temperature during the night and to switch appliances on and off), temperature remote control, broadband Internet access, ASP (Application Service Provider) services, and more. A detailed description of this step can be found in (Baida et al., 2004b). For our current discussion, we will provide a shorter summary.

When a customer searches for a service or a service bundle to buy through a Website, he is, in fact, not interested in the service itself but in the value that the service presents. This principle was acknowledged in the literature (Holbrook, 1999; Kotler, 1988; Lancaster, 1966; Teare, 1998), and can be traced back in the acknowledgment of how important customer value is in e-service offerings. The customer value of a service is reflected very often by the benefits of the service. Benefits often are expressed as the service outcomes (Kasper et al., 1999). The term *benefit* has to be understood in a broad sense; a benefit may also be negative. For example, some services require customers to perform some of the work by themselves (e.g., self-service restaurants). Also, payment – a sacrifice in terms of the service ontology – is seen as a negative benefit. Thus, the benefits of a service reflect not only the positive value of a service (from a customer's perspective), but also the negative value thereof. We describe benefits with resources. A service thus is described by its resources – its required service inputs and its produced service outcomes. Example resources from the current study are energy (of type hot water or electricity), air conditioning, and payments. Since resources represent a supplier perspective on services, they are described in objective terms rather than as a customer perceives them to be – subjective. The objective description is necessary in order to compare services, calculate prices, and provide specifications of the delivered service. Every resource is described by generic attributes (i.e., name and type) and possibly domain specific properties (to describe a state, productivity, speed, etc.). Accordingly, the quality level of a service is described by the properties of the resources associated with the service.

In other words, resources specify not only the functional benefits of a service (e.g., ability to surf on the Internet), but also an objective description of its quality (e.g., download speed). Consequently, a list of resources – including required positive benefits and acceptable negative benefits – can be used as requirements for service selection when bundling (configuring) elementary services into a value-adding service bundle. To summarize, since resources (inputs and outcomes) describe the customer benefits of a service, they will be used for the selection of services to include in a service bundle.

The use of resources to select services can be manifested by the following example. Both the service electricity supply and the service hot water have an outcome: energy. However, the service *electricity supply* provides an energy resource with the property *type: electricity*, whereas the service *hot water* provides an energy resource with the property *type: hot water*. These are, in fact, two different resources. Suppose now that a customer is interested in energy. A reasoning engine – Software that can use business logics and business rules to derive solutions suitable for customers – will then look for services with the outcome resource *energy* (without specifying the resource properties). If a variety of *electricity supply* and *hot water* services are available (possibly provided by different suppliers), each of them will have the outcome *energy*, so each of them will be a suitable solution. If, on the other hand, the property *type: electricity* also is specified, any of the *electricity supply* services (but not the *hot water* services) may be (part of) a solution.

We created a prototype software tool for modeling services in accordance with the presented service ontology. The tool (available at http://www.baida.nl/research/serviguration.html) presents a user-friendly graphical user interface that hides the technical details of the underlying service ontology. Once services are graphically modeled by the user, the tool is capable of creating a service-ontology-based machine-readable version of the model using the RDFS-W3C standard. This is an XML-based standard used for describing information; it adds a layer of semantics to information, and it is suitable for reasoning with ontologies over the Web.

STEP 3: IDENTIFY AND MODEL RELATIONS BETWEEN CUSTOMER DEMANDS AND AVAILABLE SERVICES

The purpose of building a need hierarchy is twofold. First the hierarchy is used to find context depending demands, based on more abstract wants and needs. Second, concrete demands are used to search for services that provide satisfiers (service outcomes, resources) for these demands and for more abstract needs.

To this end, we use another requirements engineering technique; namely, feature-solution graphs (De Bruin & Van Vliet, 2001; De Bruin & Van Vliet, 2002; De Bruin *et al.*, 2002). A transformation between customer demands (the satisfaction of which is the goal of the service offering) and resources (descriptors of available services, or solutions) can be viewed as a production system consisting of ***production rules***, a knowledge representation formalism used in the AI field. Production rules have the form: if situation X is encountered then select solution Y. De Bruin et al. suggested the use of context-aware *Feature-Solution graphs* (FS-graphs) to model these production rules (De Bruin & Van Vliet, 2002; De Bruin et al., 2002). The suggested graph captures and documents context-sensitive business knowledge so that it becomes possible to reason about feasible solutions and the demands (requirements) they support. A feature-solution graph (adapted to our case) includes three spaces, organized in hierarchies of AND/OR/XOR decompositions:

1. **Feature Space.** Describes the desired properties of the system (or: service) as expressed by the user. In our case, these are customer demands.
2. **Solution Space.** Contains the internal system (services) decomposition into resources that are required for or delivered by available services.
3. **Context Space.** Contextual domain knowledge that influences relations between elements of the feature space and elements of the solution space (e.g., customer type, geographic restrictions).

Relations between elements of the feature space (demands) and elements of the solution space (resources) may have the semantics of selection (if demand *A* is requested, select resource *B*), rejection (if demand *A* is requested, then do not select resource *B*), or weaker relations (positively influenced by or negatively influenced by). These are further referred to as SEL(demand, resource), REJ(demand, resource), POS(demand, resource) and NEG(demand, resource). An example FS-graph, adapted for our case, can be found in Figure 5. For visualization reasons, we explicitly mention the type of hierarchy (AND/OR/XOR) only in a few places. Note the context space, where context information as location or the type of customer may influence the behavior of a relation. The hierarchy uses AND/OR/XOR structures. As explained before, an AND structure means that all lower-level elements (demands or resources) must be satisfied in order to satisfy the higher-level element. An OR structure means that any, or a combination of the lower-level elements, can be satisfied in order to satisfy the higher-level element. A XOR structure means that any, but not more than one, of the lower-level elements can be satisfied to satisfy the higher-level element.

Our experience in using FS-graphs with business experts shows that graphs are a good means to visually *communicate ideas*, but when a substantial number of production rules is involved, and in the absence of a dedicated software tool to support this task, the use of *Excel* sheets is preferred by business experts, because the graph becomes too complex to comprehend and to manage. Yet *Excel* also presents a difficulty: it is two dimensional, while the FS-graph is three dimensional. To provide automated support for modeling production rules, we added constructs of the FS-graph to the earlier presented service ontology. Figure 6 shows how we incorporate FS-graph structures in the service ontology.

In step one of our method, we identified and modeled customer need hierarchies using the service value perspective of our service ontology. These will now be considered as features. In step two, we used the service offering perspective of our service ontology to model available services described by resources. The latter will now be considered as solutions. In the third step of our method, we define relations between demands (features) and resources (solutions), as can be seen in Figure 5. These relations have the advantage that they can easily be formalized by logical and programming constructs, making it possible to do a systematic analysis of customer needs and their corresponding solutions (available services) and to automate the reasoning for the selection of resources (and thus, services) to meet certain customer demands.

Figure 5 Partial FS-graph of the energy study

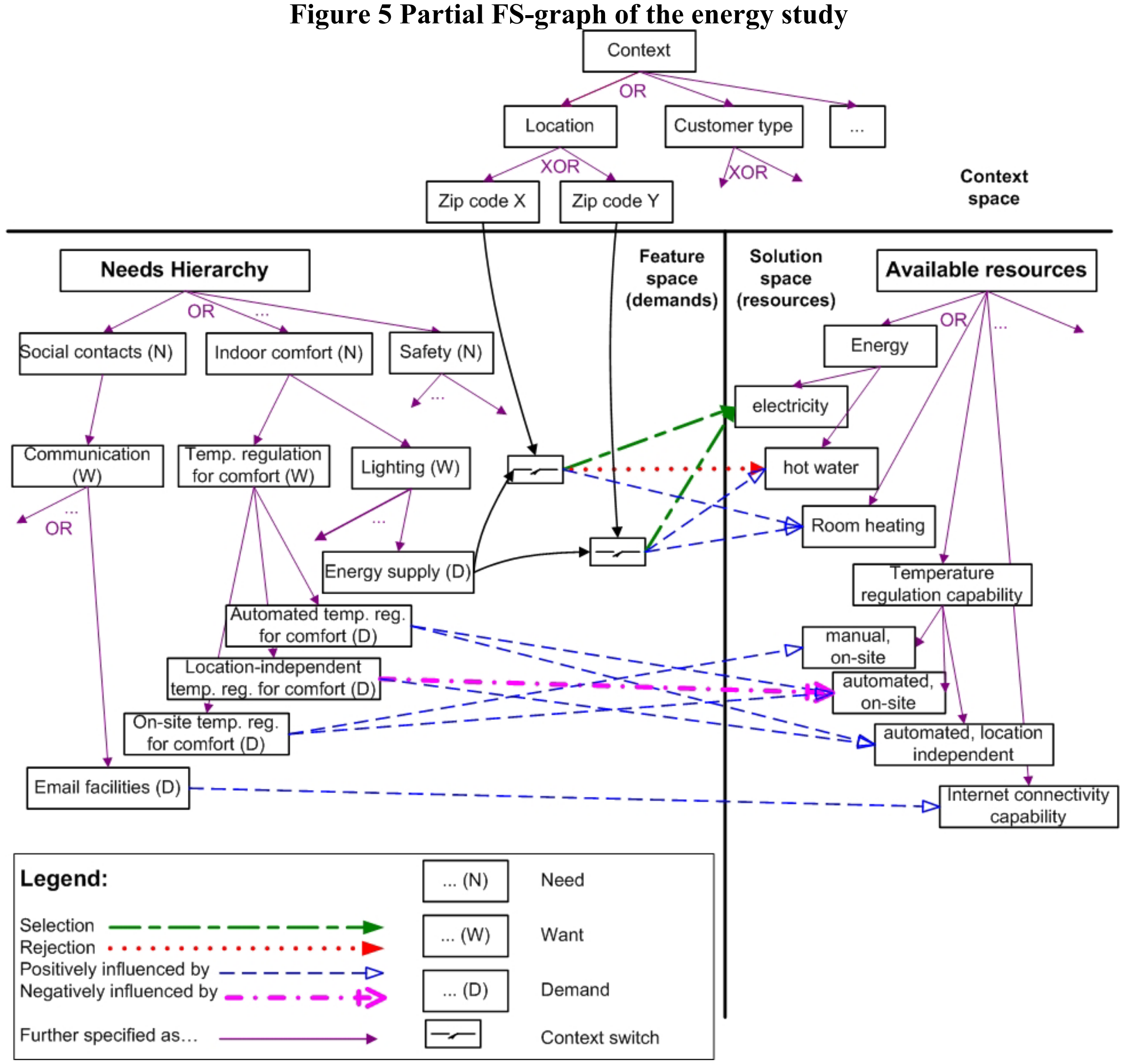
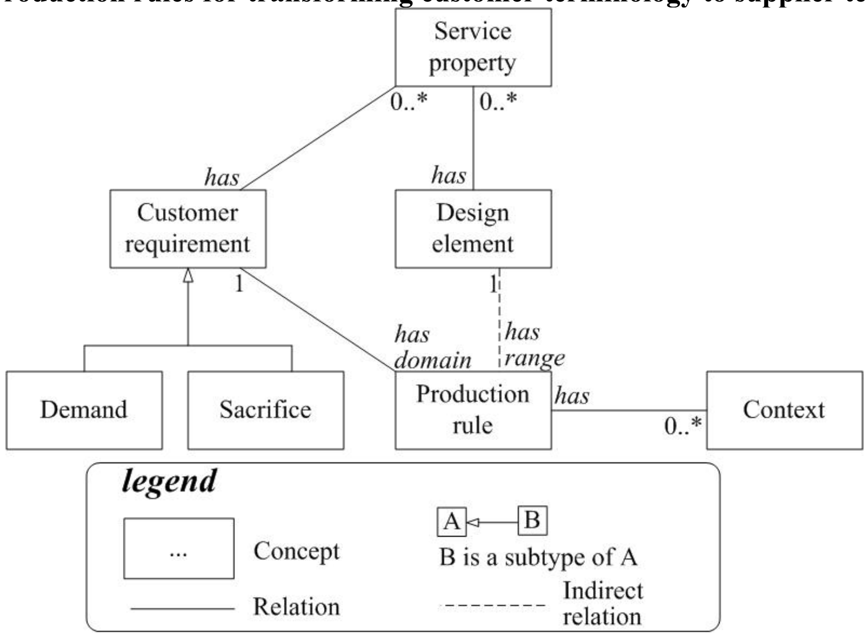


Figure 6 Production rules for transforming customer terminology to supplier terminology



Complexity in Reasoning with Production Rules

Very often demands and resources include qualitative and/or quantitative descriptors (referred to as service properties in the service ontology). For instance, in Table 1 and in Figure 5 we can find the demand for temperature regulation, specified by the descriptors ‘manual’, ‘automated’, ‘on-site’ and ‘location-independent’. Service properties may influence production rules. For example, imagine a demand for ‘email facilities’ that may be specified by the service property ‘capacity: small enterprise’, and an ‘Internet connectivity capability’ resource that may be specified by the service property ‘connection type: ISDN’. We model two production rules between these demand and resource:

1. SEL(‘email facilities’, ‘Internet connectivity capability’): if a customer has a demand for ‘email facilities’, any solution bundle must include a service that provides an ‘Internet connectivity capability’ resource.
2. NEG(‘email facilities’ with property ‘capacity: small enterprise’, ‘Internet connectivity capability’ with property ‘connection type: ISDN’): the availability of a service that provides the resource ‘Internet connectivity capability’ with property ‘connection type: ISDN’ in a bundle has a negative influence on satisfying the customer demand for ‘email facilities’ for a small enterprise.

Hence, two different production rules apply to these demand and resource, depending on the question whether or not the demand and resource are described by service properties. If a customer asks for ‘email facilities’ for a small enterprise, we search a service that provides an ‘Internet connectivity capability’ resource without service property ‘connection type: ISDN’. This example shows that it does not suffice to model one production rule between any pair (demand, resource). Service properties that describe demands and resources need to be taken into consideration as well. As demands and resources may be described by multiple service properties, theoretically every pair (demand D with service property Qx; resource R with service property

Qy) may require a production rule. This discussion can also be extended to demands and resources that are described by more than one service property. For example a capability resource “Internet connectivity” may be described by a service property ‘download speed: 8000 Kbit/s’, as well as by a service property ‘upload speed: 1024 Kbit/s’. A very large number of production rules may have to be modeled, resulting in an extensive modeling effort.

Also in the domain of telecommunication services the problem of explosion of combinations has been studied (Keck & Kuehn, 1998), and suggested solutions include tools for context generation and information acquisition. Our experience from large scale studies in the energy sector and in the health sector (Baida, 2006) is that the majority of the combinations (demand D1 with property Qx, resource R1 with property Qy) require no production rule, so the modeling effort is reasonable. Customer demands and available services that we model are described typically on a higher level of abstraction than in the case of (executable) telecommunication services as in Keck & Kuehn (1998). For example, we model demands as ‘(landline) telephone line’ and resources as ‘Internet connectivity’ with a certain download speed and upload speed, but when these services are made operational, a much richer description of QoS (Quality of Service) and desired/available features is required, resulting in a much larger number of feature combinations to deal with.

A study we carried out in the health sector (Baida, 2006) yielded a means to decrease complexity. Demands can be divided into clusters, where a cluster includes all demands that are related to a single need. Because resources are solutions for demands, very often also clusters of resources can be observed, that are related (by production rules) to clusters of demands. An important observation from our study in the health sector is that the vast majority of production rules exist between single clusters of demands and single clusters of resources. Only a small number of production rules exist between the same cluster of demands and other clusters of resources.

An important conclusion from this observation is that most modeling work can be performed by modeling experts with a reasonable effort and time investment. We can divide the space of demands and resources into clusters, identify related clusters of demands and resources, and first focus the modeling effort on production rules between these clusters. The vast majority of production rules will be modeled between pairs of clusters. Since clusters are sets of related demands and solutions for these demands, in the health study identifying clusters was natural for business experts.

Conflict Identification in Reasoning with Production Rules

Conflicts may arise in three situations when reasoning with production rules. The first situation occurs when various production rules involve the same resource (which may or may not be described by some service properties). This may cause conflicts between production rules. Imagine that we have two demands, D1 and D2, one resource R1, and the following production rules: SEL(D1, R1) (meaning that resource R1 must be selected if demand D1 is triggered) and REJ(D2, R1) (meaning that resource R1 mustn’t be selected if demand D2 is triggered). A conflict occurs when a customer has demands D1 *and* D2. On the one hand resource R1 must be part of any service bundle, and on the other hand it may not be part of a solution (bundle). In cases that we modeled in the health sector and in the energy sector, this situation was only theoretical, but it did not appear in practice. Namely, in reality when two conflicting production rules involve two different demands D1 and D2, business experts declared that these demands cannot co-exist, so the conflicting production rules involving (D1, R1) and (D2, R1) will not be triggered at the same time.

The second situation is similar to the first one, but the conflict occurs due to two different service properties Q1 and Q2 that specify the same demand D1. Conflicts occur when two production rules involve the same demand D1 with different service properties (e.g., D1Q1 and D1Q2) and a single resource R1, specified by service property Q3. Demand D1Q1 may require resource R1Q3, while demand D1Q2 has a rejection relation with R1Q3. What must be done when both D1Q1 and D1Q2 apply? This situation is different from the first situation, because here the conflicting production rules involve the same demand (only with different service properties), while the first situation involved two completely different demands.

Some terminology needs to be introduced for the discussion on a third situation of conflict. We distinguish between *global* production rules and *local* production rules.

- *Global* production rules relate a demand D1 with a resource R1, when neither the demand nor the resource is specified by any service properties (this production rule applies independently of any service property).
- *Local* production rules relate a demand D1 with a resource R1, when either the demand, or the resource, or both are specified by service properties (this production rule applies only when specific service properties are specified).

The third situation occurs when a demand and a resource have a global production rule, as well as a local production rules. Let us take as an example the earlier presented demand 'energy consumption regulation for budget control' (D1) with the service property 'mode of operation: manual' (Q1), and resource 'temperature regulation capability' (R1) with service property 'mode of operation: automated' (Q4). Two production rules are relevant here:

- POS(D1, R1): resource R1 has a positive influence on satisfying demand D1. This is a global production rule: it does not take into consideration the properties of D1 and of R1.
- REJ(D1Q1, R1Q4): when demand D1 is specified by property Q1, the solution must not include a resource R1 with property Q4. This is a local production rule: it holds for D1 and R1, only when they are specified by service properties Q1 and Q4 respectively.

To automate reasoning with production rules, one must know how the two production rules should be used together. Does the POS production rule apply when a user specifies demand D1 with quality descriptor Q1, because it is global (it holds for any pair (D1, R1), independent of their properties), or does the REJ production rule apply because it is a strong relation (while POS is a weak relation) or because it is more specific? Similar conflicts may occur also between other pairs of production rules, i.e., NEG and SEL, or POS and NEG.

Conflict Resolution in Reasoning with Production Rules

As mentioned above, case studies we performed showed that the first situation is theoretical. In order to solve conflicts of the second situation, we classify conflicts based on their *severeness*, as done by Baida et al. (2003b) who used FS-graphs for an assessment of an e-business case study:

- A *major conflict* is a conflict between two strong production rules. It involves a SEL relation and a REJ relation. No solution is possible, so no service bundle can satisfy the given demands.
- A *minor conflict* is a conflict between two weak production rules. It involves a POS relation and a NEG relation. Satisfying the demands is possible, but it requires compromises (typically, the suggested service is not "exactly" what the customer wanted; yet the customer may accept this solution if no better option exists or if its price is significantly lower than the price of other solutions).
- The third type of conflict involves a strong production rule and a weak production rule: either a SEL relation and a NEG relation, or a REJ relation and a POS relation. In these cases we analyze the impact of the conflict, and classify it as a major one or as a minor one. We refer to this as "the third type of conflict".

In order to resolve conflicts of the second situation mentioned above, we modeled production rules in studies in the health sector (Baida, 2006) and in the energy sector, analyzed the nature of conflicts, and applied the above classification of conflicts. In cases concerning conflicts between two local production rules involving the same demand (but with different service properties; that is the second situation described above):

- No major conflicts were identified
- Minor conflicts turned to be divergent: either the conflict could be ignored (i.e., the POS relation was stronger than the NEG relation), or the conflict was unsolvable (and hence the resource at hand could not be part of a solution).
- In all cases of the third type of conflict, there was no solution for the conflict (and hence no service bundle could satisfy the demands).

Based on these findings and on Baida, de Bruin & Gordijn (2003b), we determine rules for conflict resolution. These are described in Table 2. As can be seen from the table, mainly minor conflicts require human intervention to understand the nature of the conflict, and to assess how the conflict should be handled.

Table 2 Conflict resolution in case of conflicting production rules

Conflict severeness	**Conflict resolution**
Major conflict	No solution exists (no service bundle can satisfy the demands)
Minor conflict	Business experts should decide whether the conflict can be solved or not. If the conflict is declared as solvable, the resource at hand may (but need not necessarily) be included in a bundle (i.e., consider only the POS relation; disregard the NEG relation). If the conflict is declared as unsolvable, the resource at hand may not be part of a bundle. Yet, because the resource didn't necessarily have to be part of a bundle (as it did not have the selection relation), a solution is yet possible.
Third type of conflict	Our experience shows that it would be safe to say that no solution exists (no service bundle can satisfy the demands). Yet, business experts may still wish to analyze every such case independently to see whether there are some exceptional cases where a solution may exist nevertheless.

The third conflict situation that we described in the previous section occurs when a demand and a resource have a global production rule (that applies independently of any service property), as well as local production rules (that apply only when specific service properties are specified).

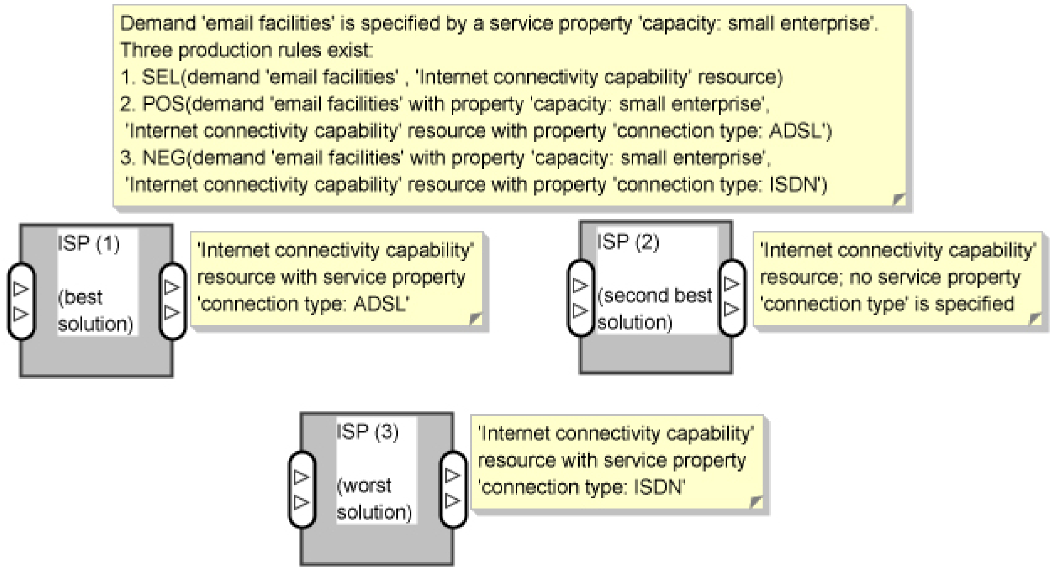
The need for local production rules next to global production rules stems from differing levels of reasoning. In reality, most demands and resources are specified by some service properties. Various similar demands and resources may exist, that differ only in some property, or in the value of a property. For example, two Internet connectivity capability resources may exist, both with the service property 'download speed' and 'upload speed', but yet every resource will have different values for these properties (i.e., different download/upload speed). Reasoning with global production rules, we may say that a demand for email facilities may be satisfied by an Internet connectivity capability resource without specifying any properties (i.e., without requiring certain download speed or upload speed). This is a global production rule. However, if the same demand is set with a quality descriptor 'capacity: household', we will set requirements also on the download/upload speed, resulting in local production rules. Note that the service ontology allows us to describe whether the values of resources in production rules specify a minimum value, a maximum value or an exact value. A production rule may then define that when a demand for 'email facilities' is set with the service property 'capacity: household', a resource 'Internet connectivity capability' must be selected with service property 'download speed' with a value of *at least* 800 Kbit/s.

Imagine we have a global production rule between demand D1 and resource R1 (without specifying service properties), as well as a local production rule between demand D1 with property Q1 (further referred to as D1Q1) and resource R1 with property Q2 (further referred to as R1Q2, it does not matter whether Q1 and Q2 are the same property or not). We need to define the relations between the global production rule between D1 and R1 ('parents') and the local production rule between D1Q1 and R1Q2 ('siblings'). Different relations may apply for different combinations of parents and siblings (any of them may be one of four types: SEL, REJ, POS, NEG). We analyzed every possible combination of production rules (parent, sibling), resulting in the following guidelines for conflict resolution:

- A global production rule between demand D1 and resource R1 applies whenever demand D1 is set, unless it is overridden by a local production rule. The global production rule holds for resource R1, independent of its quality descriptors.
- A local production rule between D1 and R1Q2 restricts R1 only when R1 is specified by a property Q2, whenever demand D1 is set without being specified by any service properties.
- As a rule of the thumb, a production rule of siblings (local) is more specific than the production rule of the parents (global), and therefore it overrides the parent's production rule. Yet, a production rule of siblings may override a production rule of parents only if the siblings' production rule is *strong*. A local *weak* production rule (POS or NEG) may only add selection criteria for selecting resources, but it may not override a global *strong* relation (SEL or REJ).
- Seemingly contradicting production rules may co-exist, if one is global, and the other is local. For example, the global production rule SEL(D1, R1) and the local production rule REJ(D1, R1Q2) should be interpreted as follows: when demand D1 is set, any solution must include a resource R1, and this resource must not have service property Q2.
- If the parents have a REJ relation, a service that provides resource R1 *mustn't* be part of a service bundle. In this case there is no logic behind modeling any other relation (SEL, POS, NEG) on the siblings level, because the strong REJ relation cannot be overridden. Modeling a REJ relation on the siblings level is possible but redundant, so it can be ignored.

Cases where weak relations are involved (POS or NEG) can be used to define an ordering among solutions, as we show in Figure 7. For example, a solution that involves a POS relation is better than a solution that does not involve such a relation; a solution that involves a NEG relation is worse than a solution that does not involve such a relation. Our experience in modeling real-world situations shows that when a NEG relation is involved, and there exist solution bundles that do not involve this relation (i.e., solutions that do not provide a resource as specified by a NEG relation), in fact there is no need to offer those bundles that provide the resource for which a NEG relation exist, because customers would not choose for it (in Figure 7, small enterprises seeking for 'email facilities' will not select service ISP (3), when the other options exist). Therefore, if there are solutions that do not provide a resource specified by a NEG relation, we do not generate solutions that *do* include this relation (in Figure 7 this means that service ISP (3) will not be offered as a solution).

Figure 7 Ordering among solutions



Context: How One Customer Differs from Another

The service value perspective of our service ontology – including the concepts *needs, wants, demands, sacrifice* and *service property* – reflects a customer view on services. As such it is by definition context-sensitive: every customer or customer type may have a different viewpoint on a service, based on his/her situation (time, location, role), on different expectations and on past experiences (Zeithaml et al., 1990). In this section we show how the context of a customer can be taken into consideration in the design of customer-tailored service bundles.

A customer's context may either relate to his personal situation or to his belonging to a target group. For example, when we offer medical services to patients, we take into consideration their *personal* medical dossiers, with knowledge about their health state. On the other hand, when we offer services to customers without knowledge of them as individuals, we base our offering on more general customer characteristics, e.g., the customer's age group or customer type (industrial versus household). Customers who share similar needs/demands in similar contexts (e.g., the demand for energy supply for industrial customers within a geographic region) are said to belong to the same *market segment* (Kotler, 1988): "a market segment is defined as a concept that breaks a market, consisting of actors, into segments that share common properties".

We model this information in the service ontology using the concept *context*, reflecting the physical and social situation of (in our case) customers of services that we model. The concept 'context' in the service ontology has the attributes name and value, for example 'name: age, value: 65' or 'name: customer type, value: industrial'. Multiple contexts may be valid simultaneously.

Two customers may have the same demand, and yet require different services to satisfy this demand because of their different ages or customer types. Hence, the transformation (captured by production rules) between needs/wants/demands and available resources (that are provided by services), and the choice of services to be included in a bundle, depend on the context of a given customer, or a customer group. Service bundles are to be designed for customers who have certain needs, and are in a certain context. Throughout this chapter we refer to Figure 1, presenting a simplification of the whole *serviguration* process. Context information is taken into consideration explicitly twice in the process:

1. Some context information describes the conditions under which a whole service element qualifies (or does not qualify) as a solution (we refer to this as “context on the service level”). This is supported by the relation “service element has context” in the service offering perspective. Services that require another context than the one specified by the current customer are not valid candidates to be included in a service bundle by the service configuration task.
   Example:
   A service for hot water (for room heating) is provided only to customers who live in a certain region. We model this geographical restriction as context information, related to the hot water service.
2. Some context information describes the conditions under which a benefit (resource) can satisfy a demand (we refer to this as “context on the resource level”). We model such relations by defining that production rules depend on a customer’s context.
   Example (from the health care domain):
   Demand D1: Discussion group concerning how to cope with the changing behavior of a dementia patient
   Resource R1: Coping advice for informal carers of dementia patients
   Production rule: SEL(D1, R1)
   Context: Customer type: informal carer
   Explanation: The SEL relation will be triggered only in queries where the customer type is ‘informal carer’. Consequently, when an informal carer asks for ‘discussion group concerning how to cope with the changing behavior of a dementia patient’, we will search for a service that offers ‘coping advice for informal carers of dementia patients’. When a different customer (e.g., a patient) has the same demand, the SEL production rule will not be triggered, and therefore we will not offer a service that provides coping advice for informal carers. Different resources exist for different customer types because patients and informal carers require different advice and support (yet, a single service may provide resources for both).

Some context information can be considered as global assumptions that narrow the scope of the information we model and of information systems that can use this model. For example, when developing an information system for service offerings within a specific geographic region, the location is assumed to be a global assumption, and it is not necessary to explicitly constrain all service offerings to that region. Global assumptions of a model (of services and customer needs) are considered to be known by all the users of the model, and are not made explicit in the *serviguration* process.

STEP 4: CREATE SERVICE BUNDLES

The process of ensuring customer value of service offerings is termed *serviguration* (Baida, 2006; Baida et al., 2003a) and sketched in Figure 1. Customer demands and acceptable sacrifices are mapped to possible service benefits (referred to as resources). These describe available services. They are then used as a trigger for the service bundling (configuration) process, resulting in zero or more sets of services that provide the required customer benefits, within the limitations of the acceptable sacrifice. Customer benefits, therefore, are criteria (or requirements) for the service configuration process. Each benefit can be related to some higher-level demand, want or need of a customer. The process of creating service bundles, based on a given set of available services and on a set of requirements expressed in resources, is discussed at length in Akkermans et al. (2004) and Baida et al. (2004b) and is beyond the scope of this chapter. For the current discussion, it suffices to say that we use research on configuration theory from the field of knowledge engineering. By describing services in a way that corresponds with existing configuration ontologies, we simplify the bundling process to a configuration task, for which a wealth of research exists (Borst, 1997; Borst *et al.*, 1997; Gruber et al., 1996; Schreiber et al., 2000; Stefik, 1995).

To automate the process of service bundling (or configuration), we combine our service tool with a configuration software tool developed by our partner Fundacion LABEIN in Bilbao, Spain. The configuration tool uses service models created by the service modeling tool (see step 2) to create service bundles based on a given set of requirements. The created service bundles are then imported back to the service tool, where they are visualized to enhance user friendliness.

LESSONS LEARNED FROM THE ELECTRICITY SUPPLY CASE STUDY

Needs Can Be Expressed Using Goal Hierarchies

This lesson falls into two parts. First, we can use goal-hierarchies to represent needs. This is important because we then can utilize existing knowledge about goal-reasoning.

Second, need-hierarchies are of use during elicitation of other needs. Business experts provided us with an initial list of identified customer needs. By asking the question *why*, requirements engineers elicit more abstract goals than those first identified in order to find out other important sub-goals of the more abstract goal that were overlooked in the first place (Van Lamsweerde, 2000). Our eventual need hierarchy (see Table 1) evolved from the initial one by applying two methods: asking the question *why* about existing needs, and also asking the question *why* about existing solutions (available services and their provided resources or results). We found that both techniques help identify new needs as well as concretize vaguely defined customer needs. Furthermore, asking the question *why* about existing needs helps understand the granularity of needs; it helps define whether a need should indeed be seen as a need, or actually is a more concrete want or demand.

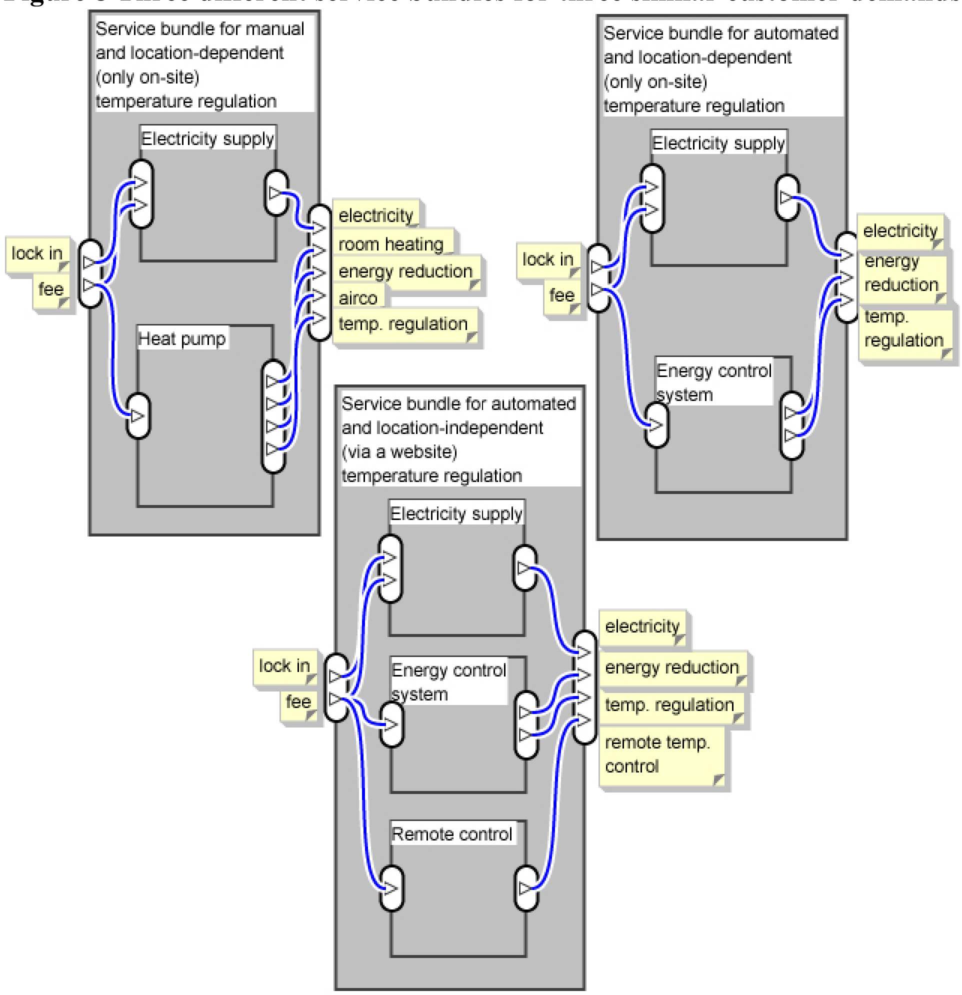
Service Ontology Allows for Reasoning on Inconsistencies and Bundles

Relations, as specified by the FS-graph, can cause inconsistencies, for example, in a situation in which a customer specifies conflicting quality criteria for a demand (e.g., a top quality, low-budget service). Handling such inconsistencies (referred to as *conflict management*) must be performed during the reasoning process. We defined guidelines for conflict management, based on this and earlier case studies (Baida, 2006; Baida et al., 2003b; De Bruin et al., 2002).

From a business perspective, reasoning on potential service bundles is of most interest. For example, (1) some services require other supporting services; (2) other services may have substitutes that also provide a good solution for a customer; (3) suppliers may prefer to bundle specific services for better utilization of existing infrastructure, and so forth. All these business rules can be expressed in a computer-interpretable way, so that software can implement them. We have built a prototype software tool that does exactly this kind of reasoning. Now that we have a set of required resources, we have to create bundles of services that offer these resources. Any of the required resources may be offered by multiple services, so typically more than one service or service bundle will include these resources and, hence, fulfill the customer's demands. This last process – service configuration – is discussed thoroughly in Baida et al. (2004b) and in

Akkermans et al. (2004) and includes the already mentioned business rules. The service configuration process implements production rules of the type *add service Y to every bundle that includes service X*, *services X and Y may not be part of the same bundle*, and so forth.

Figure 8 Three different service bundles for three similar customer demands



Reflecting Back on the Case Study Domain

We modeled a variety of services in the energy case study, including electricity supply, broadband Internet access, hot water supply, energy control and more (Baida et al., 2004b), analyzed relations between services and customer demands, and created service bundles to satisfy customer demands. As a result of the modeling of service elements and the automated generation of service bundles, the energy utility at hand succeeded in defining service bundles for specific groups of customers in such a way that these bundles fit the requirements of their respective customers. Furthermore, our analysis helped understand which service bundles should not be offered to specific groups of customers, because they do not satisfy the requirements of these customers well enough, or because other bundles can satisfy the same requirements better.

An important advantage of ontologies is that they help reason with domain knowledge. Our ontological approach, summarized in Figure 1, enables automated reasoning. For example, the customer need for ’indoor comfort’ is reduced to three wants, including ’temperature regulation’.

We found three service bundles that satisfy this want (see Figure 8). All of them include electricity supply plus extra services, supplied by different suppliers. In other words, these service bundles compete with each other. An electricity supplier can then decide whether to offer all of these bundles or just a subset thereof. The choice of a bundle to offer implies also a choice of a business partner to work with, since the extra services are offered by other suppliers. The same want is further specified by several demands. Reasoning on the demand level, we see that the competing bundles provide differing quality levels, so in fact they may address different market segments. It is then up to the supplier(s) to decide which service level(s) to offer.

RELATED RESEARCH

When our method is used by marketing personnel for developing (e-)service offerings, the use of our service ontology can be complemented by the means-end theory, which provides an even more abstract view on service offerings. The means-end theory (Gutman, 1982; Zeithaml, 1988) uses relations between customer values and product/service attributes and benefits in order to explain customer behavior and his or her preference for one product/service or another. A means-end chain is "a model that seeks to explain how a product or service selection facilitates the achievement of desired states" (Gutman, 1982); customers seek means to achieve their ends (goals). The means-end theory uses a hierarchical model to describe this customer goal-oriented behavior. The model consists of three related concepts: values, benefits/consequences, and (product/service) attributes. The hierarchy is created by relating values to underlying benefits and attributes. In their studies, Gutman (1982), Herrmann *et al.* (2000), and Mentzer et al. (1997) present examples of means-end chain models in different sectors: the railway sector, the automobile industry, and the beverages industry. Examples are provided for values, benefits, and attributes (the three elements of a means-end chain model ordered in a decreasing level of abstraction). We have presented in this chapter a need hierarchy with needs, wants, and demands (ordered in a decreasing level of abstraction). Comparing these three studies with ours, we can make the following observations about relations between the means-end theory and the service ontology:

1. Values in the means-end theory either can be terminal or instrumental. Terminal values are more abstract than any concept in the service ontology; instrumental values correspond to needs in the service ontology.
2. Benefits/consequences in the means-end theory correspond to wants and needs in the service ontology.
3. Attributes in the means-end theory either can be abstract or concrete. Abstract attributes correspond to wants in the service ontology; concrete attributes correspond to demands in the service ontology.

The existence of a similar and equivalent structure (hierarchy) and concepts makes it possible to incorporate the use of our method and ontology with means-end chain models by marketing departments. The added value that our method presents in this context is twofold:

1. Value hierarchies, as in the means-end theory, define relations between values, benefits, and attributes. By adding AND/OR/XOR refinements to hierarchies, we enable a much more detailed and useful analysis of these relations. For example, an OR refinement implies that any low-level element (e.g., demand or attribute) can satisfy a higher-level element (e.g., want or benefit). Consequently, it may not be necessary for a service provider to implement all lower-level attributes. Such knowledge cannot be inferred from means-end hierarchies in their

traditional form.

2. The means-end theory does not consider the possible solutions (actual service offerings of suppliers) for a customer's demands. Customer needs are refined to the degree of desired product attributes, but these are not related further to any elements that provide these attributes. The service ontology, on the other hand, includes both customer needs and available solutions. By using our method and ontology, it becomes possible to relate not only product attributes, but also possible solutions (available or future service offerings) to a customer's needs and values. This can be used for marketing analyses, but it is of greatest importance for e-service offerings, because they require that all elements of the process (from customer needs to actual solutions) be linked so that information systems can reason about the process and provide a customer with a suitable solution for his or her needs.

In addition, Herrmann et al. (2000) argued that the means-end theory needs to be complemented with a means to transform customer needs to more concrete, implementation-related measurements. They suggested combining the means-end theory with quality function deployment (QFD). Their approach is similar to ours in that both approaches facilitate a transformation process from vaguely defined customer needs to concrete measurements. While Herrmann et al. (2000) focus on how to understand customer behavior as a key to design new services, our work assumes that knowledge exists about customer behavior, and we focus on how an software can use this knowledge to design service bundles out of available services.

An interesting observation is that we perform conflict resolution in the *relations* (production rules) between features (demands) and solutions (resources). This is opposed to conflict resolution in software engineering (Van Lamsweerde *et al.*, 1998) and software architecture, where conflicts are managed on the feature side: goals and requirements. A possible explanation for this difference is the fact that in software design all requirements and goals refer to the same single artifact: the system to be developed. In the case of service bundling, on the other hand, customer demands need not depend on each other, and the solution may comprise of totally independent services (artifacts) that can be consumed at different times. For example, a customer may have a demand for home entertainment as well as entertainment outside home. These two demands do not conflict, because a solution service bundle may include a service that delivers home entertainment (e.g., a TV subscription) and a service that delivers entertainment outside home (e.g., a subscription for the National Ballet), and the two may be consumed independently, at different times and locations.

CONCLUSION AND FUTURE WORK

We proposed an ontology for understanding customer needs for e-services. Using this ontology, it is possible to reason about possible service bundles that satisfy needs. The bottom line of the energy study was that the analysis performed made it possible for the energy utility involved to define service bundles for specific groups of customers in such a way that bundles fit the demands of their respective customers. Furthermore, it helped understand which service bundles should not be offered to specific groups of customers, because they do not satisfy the demands of these customers well enough, or because other bundles can satisfy the same demands better. For example: to satisfy a customer demand for energy supply, a bundle theoretically may include combinations of the following services: electricity supply, heat pumping, and hot water. However, customers would prefer bundling electricity supply with hot water to bundling electricity supply with heat pumping due to a lower price. If there had not been a geographical limitation on the supply of hot water, the bundle electricity supply and heat pumping would not have been of interest. Another example is the customer demand of temperature regulation for indoor comfort. The following service elements result in benefits (resources) that satisfy this demand for industrial customers: heat pumping, energy control system, and remote control. However, given the desired quality criteria for this demand (automated vs. manual, location-independent vs. on-site), different combinations of these (and other) services need to be offered.

Knowledge and expertise from business research, information science, and computer science have been intertwined in our research to solve the problem at hand. We spilt the process into a customer perspective (step one of our method), a supplier perspective (step two of our method), and a transformation process between the two (step three of our method). By expressing both perspectives using a formal ontology, also expressible in a machine-interpretable language (RDF), we facilitate checking business knowledge for consistency, using it for reasoning by software, and performing a systematic analysis of the domain.

Business research literature concerning customer needs acknowledges the existence of (need) hierarchies. However, it lacks a few elements, necessary for making business knowledge machine-interpretable: (1) a definition of hierarchical decompositions (e.g. AND/OR/XOR structures) of customer needs; (2) a well-defined (in computational terms) description of services; (3) a definition of possible relations (links) between needs and solutions; and (4) an understanding of how demands (functional requirements) differ from desired service quality (non-functional requirements). As we have shown in this chapter, we use existing requirements engineering practices to add the necessary formalism to business concepts: we use goal hierarchies and production rules to relate features (needs, demands) to solutions (services, described by resources). By embedding these constructs and business concepts in a service ontology, expressible in a machine-interpretable language, we create a framework for software-based reasoning: first customer demands trigger the selection of resources (benefits), and then a configuration process creates bundles of services that provide these customer benefits.

The method presented in this chapter uses conceptual modeling and formalizing techniques, widely accepted in computer science and information science, and applies them to concepts from business research. In spite of the elusive nature of important business concepts such as quality, benefit, and value, it is possible to derive concrete parameters out of more abstract ones by using several layers between the two. Abstract notions can be transformed to somewhat less abstract notions; these can then be transformed or mapped to even more concrete notions. The QFD approach uses this technique, and so do means-end hierarchies and requirements engineering goal hierarchies. However, both QFD and means-end models have a limited perspective: supplier's solution and customer needs, respectively. Our method, on the other hand, connects both perspectives using FS-graphs. The two perspectives must be related in order to allow an automated process that finds a solution for a specific high-level need.

By applying our approach for the Norwegian Energy sector, we managed to elicit business knowledge and to formalize it in such a way that it can be expressed in computer-interpretable terms. The service offering perspective was implemented in a software tool. Using our service ontology as its fundaments, the tool is capable of creating bundles of services, when requirements are specified in terms of resources. In the present chapter, we have shown how we derive such requirements: by (1) adding an earlier step in which we formalize customer demands, and (2) mapping them onto available resources. Our service ontology includes a perspective dedicated to these demands: the service value perspective.

The service ontology includes the notion of quality criteria to describe customer demands. Demands, however, are subjective and context-sensitive. A wealth of knowledge exists within business research about service quality (Grönroos, 2000; Zeithaml et al., 1990). The service ontology includes constructs for modeling the quality related to demands and to services. So far, the available constructs have proven to be suitable and sufficient. Future research can be directed at incorporating existing service quality models (e.g., SERVQUAL) (Zeithaml et al., 1990) into the service ontology.

We considered in this chapter the complexity problem, caused by the large number of pairs (demand, resource) for which business experts have to consider whether a production rule must be modeled. We observed that clusters of demands and resources can be identified, such that the vast majority of production rules will be between pairs of clusters. Only limited effort needs to be put into modeling production rules outside these pairs of clusters, so the modeling effort is reasonable. At the same time we acknowledge that more empirical studies are required to make a sound statement about the complexity problem in modeling production rules.

Also, we investigated production rules involving only one demand and one resource. These were enough to model realistic and complex domains. Yet, more empirical studies are required to investigate the necessity for production rules involving multiple resources (e.g., IF demand X THEN resource Y *or* Z). Complexity can also be reduced by prioritizing customer needs. This will also be a topic of future research.

ACKNOWLEDGEMENTS

This work has been partially supported by the European Commission, as project No. IST-2001-33144 OBELIX and by the Dutch Ministry of Economic Affairs, as the FrUX project (Freeband User eXperience). This chapter is an updated and extended version of an article titled ‘Finding e-service offerings by computer-supported customer need reasoning’, published in the *International Journal of E-Business Research (IJEBR)* **1**(3), pp. 91–112.

REFERENCES

Akkermans, H., Baida, Z., Gordijn, J., Peña, N., & Laresgoiti, I. (2004). Value webs: Using ontologies to bundle real-world services. *IEEE Intelligent Systems, 19*(4), 57-66.

Aschmoneit, P., & Heitmann, M. (2002). Customer centred community applica¬tion design: Introduction of the means-end chain framework for product design of community applications. *The International Journal on Media Management, 4*(1), 13-21.

Baida, Z. (2006). *Software-aided service bundling; intelligent methods & tools for graphical service modeling.* Vrije Universiteit, Amsterdam. PhD thesis.

Baida, Z., Akkermans, H., & Gordijn, J. (2003a). *Serviguration: Towards online configurability of real-world services.* Proceedings of the Fifth International Conference on Electronic Commerce (ICEC03), Pittsburgh, Pennsylvania. ACM Press.

Baida, Z., de Bruin, H., & Gordijn, J. (2003b). E-business cases assessment: From business value to system feasibility. *International Journal of Web Engineering and Technology, 1*(1), 127-144.

Baida, Z., Gordijn, J., Morch, A. Z., Sæle, H., & Akkermans, H. (2004a). *Ontology-based analysis of e-service bundles for networked enterprises.* Proceedings of the 17th Bled e-commerce Conference, Bled, Slovenia.

Baida, Z., Gordijn, J., Sæle, H., Morch, A. Z., & Akkermans, H. (2004b). *Energy services: A case study in real-world service configuration.* Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAiSE 2004), Riga, Latvia. Springer-Verlag.

Berry, L., & Parasuraman, A. (1991). *Marketing services: Competing through quality*. New York: The Free Press.

Bigné, E., Martínez, C., & Miquel, M. J. (1997). The influence of motivation, experience and satisfaction on the quality of service of travel agencies. In *Managing service quality (volume iii)* (pp. 53-70). London: Paul Chapman Publishing.

Borst, P. (1997). *Construction of engineering ontologies for knowledge sharing and reuse.* Universiteit Twente, Enschede, The Netherlands. PhD thesis.

Borst, P., Akkermans, H., & Top, J. (1997). Engineering ontologies. *International Journal of Human-Computer Studies, 46*, 365-406.

Centeno, C. (2003). *Adoption of internet services in the enlarged european union: Lessons from the internet banking case (http://fiste.Jrc.Es/download/eur20822en.Pdf, retreived august 2004)* (No. EUR 20822 EN).

De Bruin, H., Van Vliet, H., & Baida, Z. (2002). *Documenting and analyzing a context-sensitive design space.* Proceedings of the 3rd Working IFIP/ IEEE Conference on Software Architecture (WICSA-02), Montreal, Canada.

De Bruin, H., & Van Vliet, H. (2001). *Scenario-based generation and evaluation of software architectures.* Proceedings of the Third Symposium on Generative and Component-Based Software Engineering (GCSE2001), Erfurt, Germany.

De Bruin, H., & Van Vliet, H. (2002). *Top-down composition of software architectures.* Proceedings of the 9th International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS2002), Lund, Sweden.

De Bruin, H., Van Vliet, H., & Baida, Z. (2002). *Documenting and analyzing a context-sensitive design space.* Proceedings of the 3rd Working IFIP/ IEEE Conference on Software Architecture (WICSA-02), Montreal, Canada.

Donzelli, P. (2004). A goal-driven and agent-based requirements engineering framework. *Requirements Engineering, 9*(1), 16–39.

Flæte, A., & Ottesen, G. (2001, 13/14 October). Telefiasko for viken*. Dagens Næringsliv (Norwegian Newspaper)*.

Ford, K., & Bradshaw, J. (Eds.). (1993). *Knowledge acquisition as modeling*. New York: John Wiley & Sons.

Fuxman, A., Liu, L., Pistore, M., Roveri, M., & Mylopoulos, J. (2003). *Specifying and analyzing early requirements: Some experimental results.* Proceedings of the 11th IEEE International Requirements Engineering Conference (RE'03), Monterey Bay, California. IEEE Computer Society.

Grönroos, C. (2000). *Service management and marketing: A customer relationship management approach*. Chichester, UK: John Wiley & Sons.

Gruber, T., Olsen, G., & Runkel, J. (1996). The configuration design ontologies and the vt elevator domain theory. *International Journal of Human-Computer Studies, 44*(3/4), 569-598.

Gutman, J. (1982). A means-end chain model based on consumer categorization processes. *Journal of Marketing, 46*(2), 60-72.

Herrmann, A., Huber, F., & Braunstein, C. (2000). Market-driven product and service design: Bridging the gap between customer needs, quality management, and customer satisfaction. *International Journal of Production Economics, 66*(1), 77-

96.
Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 24*(1), 75-105.
Holbrook, M. B. (1999). *Consumer value: A framework for analysis and research*. New York: Routledge.
Kärkkäinen, H., & Elfvengren, K. (2002). Role of careful customer need assessment in product innovation management empirical analysis. *International Journal Production Economics, 80*(1), 85-103.
Kasper, H., van Helsdingen, P., & de Vries Jr., W. (1999). *Service marketing management: An international perspective*. Chichester, UK: John Wiley & Sons.
Keck, D. O., & Kuehn, P. J. (1998). The feature and service interaction problem in telecommunications systems: A survey. *IEEE Transactions on Software Engineering, 24*(10), 779–796.
Kotler, P. (1988). *Marketing management: Analysis, planning, implementation and control*. Englewood Cliffs, NJ: Prentice Hall.
Lancaster, K. J. (1966). A new approach to consumer theory. *Journal of Political Economy, 74*(2), 132–157.
Liu, L., & Yu, R. (2001). *Requirements to architectural design – using goals and scenarios.* Proceedings of the First International Workshop “From Software Requirements to Architectures” (SREAW-01), Toronto, Canada.
Löckenhoff, C., & Messer, T. (1994). Configuration (chapter 9). In J. Breuker & W. V. d. Velde (Eds.), *Commonkads library for expertise modelling – reusable problem solving components* (pp. 197–212). Amsterdam, The Netherlands: IOS Press.
Lovelock, C. (2001). *Services marketing, people, technology, strategy* (4th ed.). Englewood Cliffs, NJ: Prentice Hall.
Martinussen, K. F. (2002). Kankan som kunne. (presentation at the itenergi 2002 conference). Retrieved December, 2005, from http://www.itenergi.com/kari_martinussen.ppt
McCullough, D. (2002). A user’s guide to conjoint analysis. *Marketing Research, 14*(2), 18-23.
Mentzer, J. T., Rutner, S. M., & Matsuno, K. (1997). Application of the means-end value hierarchy model to understanding logistics service value. *International Journal of Physical Distribution and Logistics, 27*(9/10), 630-643.
Mittal, S., & Frayman, F. (1989). *Towards a generic model of configuration tasks.* Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89), San Francisco, CA. Morgan Kaufmann.
Murthi, B., & Sarkar, S. (2003). The role of the management sciences in research on personalization. *Management Science, 49*(10), 1344-1362.
Mylopoulos, J., Chung, L., Liao, S., Wang, H., & Yu, E. (2001). Exploring alternatives during requirements analysis. *IEEE Software, 18*(1), 92-96.
Mylopoulos, J., Chung, L., & Yu, E. (1999). From object-oriented to goal-oriented requirements analysis. *Communications of the ACM, 42*(1), 31-37.
Quine, W. V. O. (1961). *From a logical point of view, nine logico-philosophical essays*. Cambridge, MA: Harvard University Press.
Reynolds, T. J., & Gutman, J. (1988). Laddering theory, method, analysis and

interpretation. *Journal of Advertising Research, 28*(1), 11-29.

Schreiber, A. T., Akkermans, J. M., Anjewierden, A. A., de Hoog, R., Shadbolt, N., Van der Velde, W., et al. (2000). *Knowledge engineering and management*. Cambridge, MA: MIT Press.

Sharp, H., & Galal, A. F. G. (1999). *Stakeholder identification in the requirements engineering process.* Proceedings of the 10th International Conference and Workshop on Database and Expert Systems Applications (DEXA 99), Florence, Italy.

Stefik, M. (1995). *Introduction to knowledge systems*. San Francisco, CA: Morgan Kaufmann.

Stephens, S. A., & Tripp, L. L. (1978). *Requirements expression and verification aid.* Proceedings of the 3rd international conference on Software engineering (ICSE 1978), Atlanta, Georgia.

Teare, R. E. (1998). Interpreting and responding to customer needs. *Journal of Workplace Learning, 10*(2), 76-94.

Van Lamsweerde, A. (2000). *Requirements engineering in the year 00: A research perspective.* Proceedings of the 22nd International Conference on Software Engineering, Limerick, Ireland.

Van Lamsweerde, A. (2001). *Goal-oriented requirements engineering: A guided tour, invited minitutorial.* Proceedings of the International Joint Conference on Requirements Engineering (RE'01), Toronto, Canada.

Van Lamsweerde, A., Darimont, R., & Letier, E. (1998). Managing conflicts in goal-driven requirements engineering. *IEEE Transactions on Software Engineering, 24*(11), 908–926.

Vasarhelyi, M., & Greenstein, M. (2003). Underlying principles of the electronization of business: A research agenda. *International Journal of Accounting Information Systems, 4*(1), 1-25.

Xue, M., Harker, P. T., & Heim, G. R. (2003). *Incorporating the dual customer roles in e-service design (working paper no. 03-04, retrieved august 2004 from http://fic.Wharton.Upenn.Edu/fic/papers/03/0304.Pdf)*.

Zeithaml, V. A. (1988). Consumer perceptions of price, quality, and value: A means-end model and synthesis of evidence. *Journal of Marketing, 52*(3), 2-22.

Zeithaml, V. A., & Bitner, M. J. (1996). *Services marketing*. New York, NY: McGraw-Hill.

Zeithaml, V. A., Parasuraman, A., & Berry, L. (1990). *Delivering quality service: Balancing customer perceptions and expectations*. New York, NY: The Free Press.