

An Ontological Approach for Eliciting and Understanding Needs in e-Services*

Ziv Baida¹, Jaap Gordijn¹, Hanne Sæle², Hans Akkermans¹,
and Andrei Z. Morch²

¹ Free University, FEW/Business Informatics,
De Boelelaan 1083a, 1081 HV Amsterdam, The Netherlands
{ziv, gordijn, elly}@cs.vu.nl

² Dep. of Energy Systems, SINTEF Energy Research,
7465 Trondheim, Norway
{Hanne.Saele, azm}@sintef.no

Abstract. The lack of a good understanding of customer needs within e-service initiatives caused severe financial losses in the Norwegian energy sector, resulting in the failure of *e-service* initiatives offering packages of independent services. One of the causes was a poor elicitation and understanding of the e-services at hand. In this paper, we propose an ontologically founded approach (1) to describe customer needs, and the necessary e-services that satisfy such needs, and (2) to bundle elementary e-services into needs-satisfying e-service *bundles*. The ontology as well as the associated reasoning mechanisms are codified in RDFS to enable software support for need elicitation and service bundling. A case study from the Norwegian energy sector is used to demonstrate how we put our theory into practice.

1 Introduction

Today, e-Business still focuses on supporting the production, sale, and consumption of *physical* products. However, in real-life, many products are actually *services*, in contrast to goods that you can drop onto the floor. Consequently, a new paradigm in e-Business is emerging: *e-services* [20].

E-services are a web-based version of traditional services, defined as business activities, deeds and performances of a mostly intangible nature [16, 18, 25, 17]. In the rest of this paper we refer to this definition when we use the term ‘e-service’. Note that e-services are not the same as web-services. E-services are *commercial* services, provisioned over the Internet, whereas web-services are a paradigm to arrive at truly distributed computing (more information on the differences between e-services and web-services can be found in [6]).

* This work has been partially supported by the European Commission, as project No. IST-2001-33144 OBELIX (Ontology-Based ELeCTronic Integration of complex products and value chains) and by the Dutch Ministry of Economic Affairs, as the FrUX project (Freeband User eXperience).

In e-Business scenarios, it is important that all participants have a *shared* understanding of the goods and services offered and requested. For instance, many efficiency gains in supply chain management rely on information integration along this chain; to arrive at such an integration each party in the chain should have the same understanding of the good or service to be delivered. Moreover, since e-Business scenarios use software components extensively almost by definition, it is also important to have a *formal*, or at least machine interpretable understanding of the services and goods. As an example, consider a software component that proposes a bundle of services or goods (potentially delivered by multiple suppliers) that as a *bundle* satisfies a specific customer need. Such a bundling component should be able to reason about meaningful combinations of products¹. This can only be done if the service and good descriptions are machine interpretable. Making a shared, formal conceptualization of, in this case, goods and services is the field of ontologies [10]. Existing product ontologies have a strong emphasis on physical goods (see e.g. [2, 1]) and not yet on services.

To address this shortcoming, we developed and tested an ontology to represent e-services [3], and implemented a prototype software tool. This ontology can be used, first to build a catalogue of e-services, and second to compose *bundles* of e-services, which as a bundle satisfy a customer need. Bundling is a well known economic principle in building service offerings [16]. Elementary services in a bundle can be offered by different enterprises that form a partnership. An ontological approach is really needed here; all businesses in such a partnership should have a shared and formal understanding of their elementary e-services to facilitate automated reasoning about commercial viable e-service bundles.

So far, our ontology has been stated in supplier-oriented terminology. Customers have to formulate their customer needs (or requirements) in *supplier-oriented* terminology to arrive at meaningful bundles of e-services. Obviously, a customer wants to express his needs in his own terminology, which can be different from the terminology used by the supplier. This paper presents an extension to our ontology that allows a customer to do so.

This work is unique in a number of ways. First, it recognizes the commercial nature of services, in contrast to web-service ontological approaches, which are strong at facilitating distributed computing but poor in the commercial interpretation of 'service'. Second, our ontology is capable of doing automated bundling of elementary e-services into e-service bundles using a configuration problem solving method borrowed from AI. And now, our ontology can be used to derive, based on customer needs, bundled e-service offerings.

Our work is not limited only to e-services, but applies to any service offerings. Nevertheless, our work is of much greater importance for e-services, since they require automating the processes that may otherwise be performed in the minds of service personnel. Consequently, for e-services realization it is absolutely necessary that domain knowledge is conceptualized, formalized and made machine-interpretable. This is what we aim to achieve in our work.

¹ Both *services* and *goods* are subclasses of *product*.

This paper starts with an introduction of our case study in Section 2. Then we present a summary of earlier work on our e-services ontology (Section 3). In Section 4 the ontological extension to cover customer needs, wants, and demands is introduced. Then, in Sections 5 and 6, we show how this extension uses known theory from Requirements Engineering to come from customer needs to service-outcomes, to be produced by suppliers. Section 7 discusses existing and envisioned tool support. Finally, in Section 8 we present our conclusions and identify directions for future research.

2 Case Study: Service Bundles in the Norwegian Energy Sector

This paper uses a case study, based on a real-life e-service elicitation project we carried out in Norway. The study at hand is about electricity supply. Electricity is an anonymous product. Due to a fierce competition in generation and supply of electricity, the difference in electricity retail prices per kWh between suppliers is diminishing. Electricity fits ideally to definition of “perfect substitute”, since electricity which is sold from one supplier has exactly the same physical characteristics as electricity sold by another supplier, and also prices are very similar. Consequently, many suppliers are seeking for ways to differentiate their offerings from competitors, so that customers are able to distinguish the individual offerings from suppliers. One way to do so is to add complementary and additional services such as Internet access and home comfort management to the electricity supply offering. In our study for an electricity supplier in Trondheim, Norway, we analyzed possible service bundles that can be offered via the Web to customers who wish to buy electricity. Service bundles had to be designed so that (1) they are commercially interesting, and (2) they meet customers’ demands. We focus on the first question in [5]. The present article focuses on the second question.

3 Ontological Framework: A Service Ontology

On a high level of abstraction, our service ontology [3] embodies three interrelated top-level perspectives:

The *service value* perspective captures knowledge about adding economic value from a customer viewpoint. It expresses 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). As the service value perspective is the main contribution of this paper, we elaborate more on this perspective in Section 4.

The *service offering* perspective represents the supply-side viewpoint: it provides a hierarchy of service components (e.g. 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 process* perspective encapsulates knowledge about putting the service offering into operation in terms of business processes. In contrast to the usual production process of physical goods, customers often take active part in the service production process.

Our earlier work focused on the service offering perspective (see [7] for a detailed discussion). For the current discussion it suffices to understand that the provisioning of a service requires a set of *service inputs* and results in the availability of a set of *service outcomes*. Very often the service outcomes reflect the customer benefits from a service, whereas the customer sacrifice is mapped into service inputs (e.g. payment). Service inputs and outcomes are referred to as *resources*. Hence a service is described by its resources: its requires inputs and its generated outcomes. Section 4 presents the service value perspective. Sections 5 and 6 describe how both perspectives can be related in order to support reasoning on needs-driven service bundling.

4 Service Value Perspective

The service offering perspective of our service ontology describes service elements including their input- and outcome-resources, as well as customer requirements in *supplier-oriented* terms. The motivation for doing so is that the service offering perspective aims at configuring the various e-services elements of different suppliers in a more comprehensive bundle, and for doing so we need the actual service elements that can be provisioned by these suppliers.

Customers however do not articulate their needs in terms of supplier-oriented requirements but employ their own, subjective terminology for expressing *demands*. To deal with these demands, we extend our ontology with *needs, wants, demands*, and miscellaneous constructs. In brief, customers state their *demands*, which can (partly) be satisfied by a series of bundled *resources*, which in turn are provisioned by *service-elements* (see Figure 1).

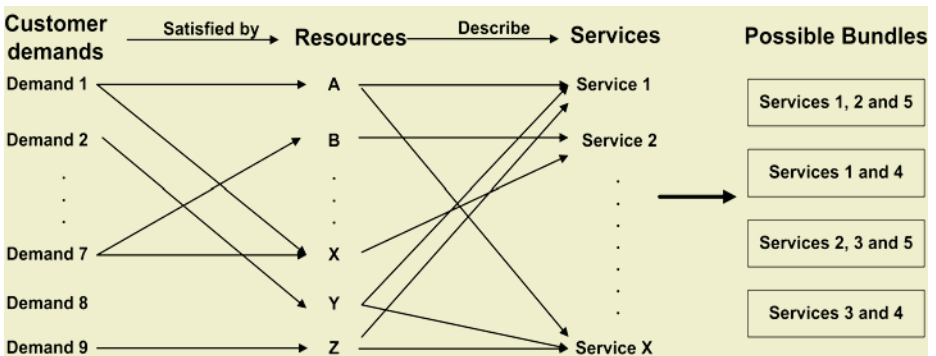


Fig. 1. Configuring service bundles based on customer demands

The *service value* perspective is sketched in Figure 2 and is explained below.

Needs, wants and demands. The starting point for the discipline of marketing lies in the human needs and wants [18]. The term *need* refers to what humans need and want (to buy). Kotler [18] distinguishes between 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.

Needs are often vague; the need for “financial security” can be interpreted in many ways. Customers concretize their needs by transforming them into wants and demands, for example based on exposure to existing services and to marketing campaigns. Often when a customer is interested in a service, he has already transformed his needs into wants and demands. He has, in fact, already found a solution for his problem (need). *Example:* indoor comfort (need); lighting (want); energy supply (demand). An exploration of customer needs, wants and demands for the energy utility we investigated is provided in Table 1.

Service quality is the degree and direction of the discrepancy between a customer’s expectations and the perception of the service [9]. Customer expect-

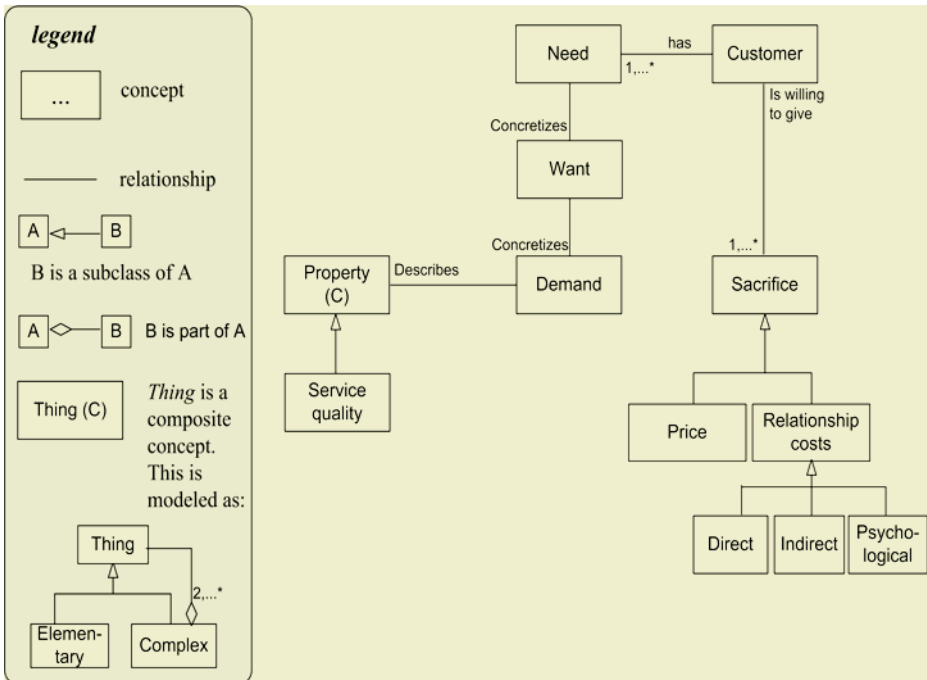


Fig. 2. Service sub-ontology representing the service (customer) value perspective

tations embrace several different elements, including desired service, predicted service and a zone of tolerance that falls between the desired and adequate service levels [8]. Expectations are based on word of mouth communications, personal needs, past experience and external communications from service providers [25]. At least two widely accepted generic methods for defining service quality are used in business science: that of the Nordic school [16] and that of the North American school (SERVQUAL, see [25]). *Example:* In electricity supply, quality can be seen as the allowed deviations in voltage, frequency, the allowed occurrences of outages, etc. Customers, for instance with respect to power outages, have different views on this (some customers have high-availability requirements, while others have not).

Next to quality, also other criteria may play a role, e.g. location and time (*where* and *when* the service should be provided). For this reason we introduced the concept **Property** in our ontology, as a super class of ‘service quality’. In the rest of this paper whenever we use the term ‘desired quality’ we refer also to other properties.

Sacrifice. The customer sacrifice includes the price of the service as well as relationship costs: direct (e.g. investment in office space), indirect (e.g. time that the customer has to devote to maintaining the relationship) or psychological costs (e.g. lack of trust in a service provider; unpleasant sensory experiences such as noise) [16]. *Example:* time spent waiting to be served; travel costs; switching costs (from one supplier to another).

5 From Service Value Perspective to Service Offering Perspective

5.1 Need, Want and Demand Hierarchies

The notions *goals* and *dependencies* have been acknowledged in the field of Goal Oriented Requirements Engineering (GORE) as suitable for transforming high-level organizational needs to concrete system requirements [13]. We utilize this GORE terminology to relate the service-value perspective to the service-offering perspective.

Needs, Wants, and Demands (further collectively referred to as “needs”) capture the answer for the question why a service bundle is offered. Hence *needs* are equivalent to *goals* in system/software design: needs represent *why* a service bundle is needed; goals represent why a system/software is needed.

Goals, at different levels of abstraction, capture the various objectives that the system under consideration should achieve [22, 23]. Just like goal hierarchies exist in GORE [15], also the marketing literature acknowledges hierarchies of needs [18]. As shown in Section 4, a *need* is a state of felt deprivation of some basic satisfaction; it can be concretized by *wants*, and further by *demands*: wants for specific products that are backed up by an ability and willingness to buy them.

Table 1 presents our hierarchy of needs, wants and demands (further referred to as ‘need hierarchy’) for the energy case study at hand. The notations H/I refer

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

<i>Customer's Needs</i>	<i>Customer's Wants</i>	<i>Customer's Demands</i>
Indoor comfort (H,I)	Lighting (H,I); Home services (cooking, washing) (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 bud- get control (H,I)	Energy regulation for budget control (H,I), with different characteristics (manual / au- tomated; on-site regulation / location-independent)
	Temperature regulation for increased comfort (H,I)	Temperature regulation (H,I) with different characteris- tics (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)
Safety (H,I)	Increased security (H,I); Reduced insurance pre- mium (H)	Safety check of electrical instal- lation (H); Internal control of electrical in- stallation (I)
IT support for business (I)	IT-services (I)	ASP-services (I); Hardware (I); Software (I)

to the customer type: Household or Industrial. As can be seen from the table, demands may either indeed refer to concrete services (e.g. a mobile phone line), or be more abstract, when a customer does not necessarily know which service can satisfy his need, or when a diversity of solutions exists (e.g. the demand for temperature regulation).

5.2 A Need Goal Tree

Needs, wants, and demands can be arranged in an AND/OR goal tree as we know from GORE. In GORE, links between goals are aimed at capturing situations where goals positively or negatively support other goals [23]. Hard-goals can be refined through AND/OR graph structures. Soft-goal refinement uses the same AND/OR structures, as well as weaker links: *A contributes positively to B* and *A contributes negatively to B*. By using links to refine goals it becomes possible

to reason about goals. We therefore suggest to introduce the same AND/OR refinements used in GORE also for need hierarchies. In such a need hierarchy, needs are concretized in wants, which in turn are concretized in demands. Demands are the leafs of the tree. An example of such a need-hierarchy is given in the left-most part of Figure 3. Note that the marketing literature does not make a distinction between types of needs or refinements in need hierarchies.

Our case study showed that the use of above refinement structures requires adding a *context* dimension, since customer needs differ per customer type, and thus the refinement changes per customer type. This needs-differentiation relates to the notion of *market segment* in marketing literature [18]: “a market segment is defined as a concept that breaks a market, consisting of actors, into segments that share common properties”. These common properties depend on the specific context.

For example, the customer want for ‘communication’ can be refined to three demands: (regular) telephone line, mobile phone line and Internet access. Whereas one customer may require only a regular phone line, another may want Internet access and a mobile phone line, and no regular phone line. This illustrates precisely why supplier stated requirements are not sufficient for e-service bundling: suppliers present services in terms of what they can offer, while customers initially think in vague terms such as ‘communication’.

Quality attributes are stated as properties of demands in Table 1. For instance, consider temperature regulation with quality properties *manual*, *automated*, *on-site*, etc.

By following the AND/OR relations, the knowledge required for reasoning about potential service outcomes satisfying a need is thus available on the demand level, rather than on the more abstract want or need levels. Since demands are satisfied by outcomes (resources) provisioned by e-services, demands are a good starting point for finding service bundles that may satisfy higher needs.

6 Demands and Resources Are Features and Solutions

6.1 Demands Are Satisfied by a Service That Provides Certain Resources

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, found demands should be satisfied by service-outcomes (resources) provisioned by suppliers. We employ Feature-Solution graphs [11, 12] to relate demands and resources.

6.2 Linking Demands to Resources

A link between customer demands and resources provisioned by services 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

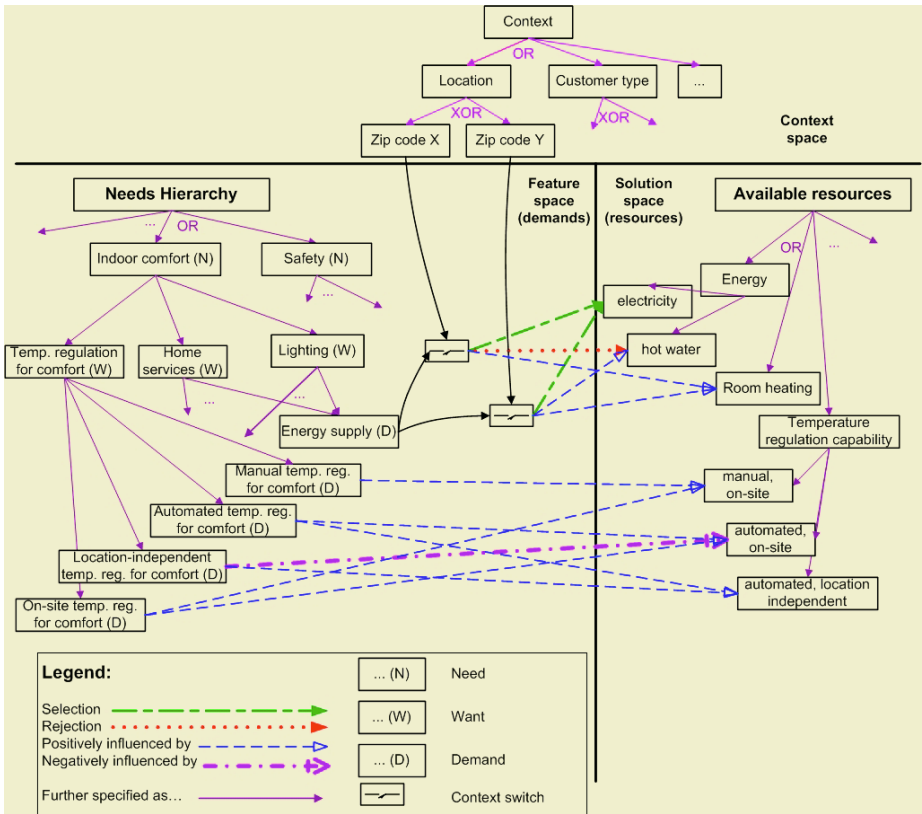


Fig. 3. Partial FS-graph of the energy case study

situation X is encountered then select solution Y. De Bruin et al. [11, 12] suggested the use of a context-aware *Feature-Solution graph* (FS-graph) to model these production rules. FS-graphs capture and document context-sensitive domain knowledge, so that it becomes possible to reason about feasible solutions and the requirements they support. An FS-graph includes three spaces, organized in hierarchies of AND-(EX)OR 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 produced by available services.
3. **Context space:** contextual information relevant for the domain (e.g. customer types, geographic restrictions).

Links between elements of the Feature-space (demands) and elements of the Solution-space (resources) may have the semantics of *selection* (if demand A then resource B), *rejection* (if demand A then not resource B) or weaker relations:

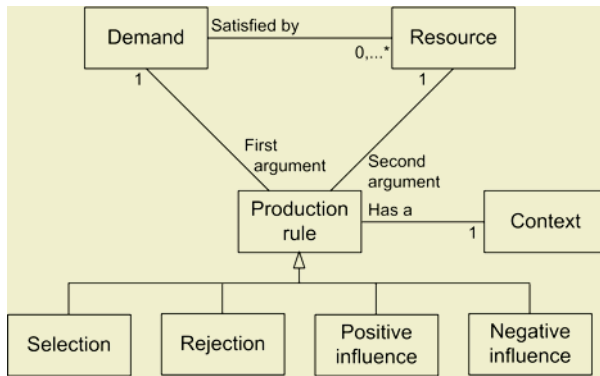


Fig. 4. FS-graph constructs in the service ontology

positively influenced by or *negatively influenced by*. The FS-graph offers levels of flexibility as a result of the different decomposition possibilities of features and solutions.

An example FS-graph, adapted for our case, can be found in Figure 3. For visualization reasons we present only a fraction of the need hierarchy tree, and we mention the type of hierarchy (AND/OR/XOR) explicitly only in a few of the places. As can be seen, contextual information can change the behavior of a production rule. This is modeled by a *context switch*. If a switch node is selected, the switch is closed and establishes context-dependent relations between features and solutions [12]. Context may include location, but also customer type (see the discussion on market segments in Section 5.2).

While the FS-graph can be used to visually communicate above production rules, its constructs need to be added to the earlier presented service ontology in order to facilitate the automated support of linking demands and resources. Figure 4 shows how we incorporate FS-graph structures in the service ontology.

The service value perspective of our service ontology – including the concepts *needs*, *wants*, *demands*, *sacrifice* and *quality* – reflects a customer view on services. As such it is by definition context-sensitive: every 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.

6.3 Relevance of Relations in the FS-Graph

We applied the FS-graph approach to the energy case study, considering customer demands as features, and available resources as solutions. This resulted in lessons learned regarding the four types of relations between features and solutions.

1. A *selection* relation hardly exists. There is not really a *single* resource for a possible demand, but alternatives exist.

2. A *rejection* relation may be required, but does not occur often. An example is the demand for energy supply with quality descriptor ‘green energy’ and the resource ‘energy’ with quality descriptor ‘nuclear’.
3. A *positively influenced by* relation is the basis for our model. It denotes that a customer demand *can* be fulfilled by certain resources, and hence by certain services.
4. A *negatively influenced by* relation may occur, but not often. For example, when a demand is specified by quality descriptors implying that the customer is interested in a cheap service, whereas a suggested resource is specified by the quality descriptor ‘high’.

As can be seen, the *positively influenced by* relation plays the main role in our case. One could thus question the use of the FS-graph constructs, which include four relations. There are two answers for this question. First, a conceptualization and formalization of domain knowledge is an absolute necessity for automated reasoning and solving problems about that knowledge. The FS-graphs approach, using goal hierarchies, has shown to be an effective approach for making this knowledge explicit, and suitable for systematic automated reasoning using production rules. Second, when adding also acceptable sacrifices (i.e. price) to the graph, we will receive a richer FS-graph, in which the *rejection* relation and the *negatively influenced by* relation will occur much more often (a low acceptable sacrifice will disqualify expensive solutions).

6.4 Reflecting Back on the Case Study

We modeled a variety of services in the energy case study, including *electricity supply*, *broadband Internet access*, *hot water supply* and *energy control* and more (see [7]), analyzed links 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, enabled reasonings as the following. The customer *need* for ‘indoor comfort’ is reduced to three wants, including ‘temperature regulation’. We found three service bundles that satisfy this want. 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 whereas the competing bundles provide

a solution to the same want, they target different demands, that encapsulate differing quality levels. A supplier can then decide whether it wants to provide all quality levels (and thus sell all generated bundles), or address only a specific target group (high-end, low-end).

7 Tool Support

The service offering perspective of our ontology was implemented in a CASE tool (a prototype is available at www.cs.vu.nl/~ziv/tool). The tool presents an easy-to-use GUI, with which business analysts and domain experts can model services from a supplier perspective. Subsequently, the tool is capable of transforming the visually-modeled services into a computer-interpretable RDF representation, based on our service ontology, and to generate also an RDF representation of *supplier-oriented* bundling requirements (i.e. in terms of resources: service inputs and service outcomes). We then use a *configuration tool* (based on a configuration ontology [4]) to generate service bundles based on the service descriptions and on these requirements. The tools communicate by exchanging RDF files: the service modeling tool provides service descriptions and bundling (configuration) requirements. The configuration tool provides solutions: service bundles that meet the requirements. Using our service ontology as its fundamentals, our software is capable of configuring bundles of services, when requirements are specified in terms of resources.

In the present paper we have shown how we derive such requirements: by adding an earlier step in which we formalize customer demands, and map them into available resources. We are currently involved in a project where the emphasis is put on this earlier step, for which software support will be implemented as well. Two main issues should be solved for effective software support: (1) understanding the nature of contextual information that influences production rules, and (2) conflict resolution concerning weak production rules (the *positively/negatively influenced by* relations). These issues are still ongoing work. Conflict resolution may be performed in advance, and not in real-time, because it is often desired to know in advance which bundles may be generated; in a business environment it may turn not to be desirable to enable customers to generate **any** valid service bundle, because some bundles may be valid, but yet not financially interesting for service suppliers. Consequently, a needs-driven analysis can be performed as part of a business analysis, thereby identifying and solving conflicts *offline*, and then the space of possible solutions (service bundles) can be limited to those bundles that were examined and found feasible.

8 Conclusions and Future Work

Only a few years ago, when the 'dotcom' wave was still rolling, it was almost impossible to find an electric utility in Norway, which did not offer so-called *bundles* via its website. One could observe a great variety of products and ser-

vices, which were offered together with electricity retail contracts. Despite costly marketing campaigns, these offerings were mostly not appreciated by customers and failed. Experience shows that the bundling of services without sound logical fundamentals of the bundles-configuration process (as applied in [7]) and without reasoning about customer needs and demands may cause severe financial losses [19, 14]. The need for such fundamentals, combined with an online process, was the driving force behind our case study.

In Sections 6.3 and 6.4 we give examples of results we obtained in this case study from modeling customer demands and available services with an ontology. Domain experts declared to have gained insight into their domain by the use of an ontology to model domain knowledge. We investigated and modeled service bundles for two market segments of energy consumers: households and industrial customers. Our analysis resulted in identifying sets of services to be offered to these market segments (or: customers), to meet specific needs and demands. In other words, whereas multiple services could be offered to a customer, reasoning about his needs is required in order to satisfy the client. If this knowledge is conceptualized and formalized, software can perform this reasoning instead of humans. This conceptualization and formalization takes place in what we refer to as *Needs Engineering*.

Knowledge and expertise from business science, information science and computer science have been intertwined in our research to solve the problem at hand. We split the process into a customer perspective, a supplier perspective and a transformation process between the two. By expressing both perspectives using a formal ontology, also expressible in a machine-interpretable language (RDFS), we ensure that domain knowledge is formalized in such a way that it can be checked for consistency and used for reasoning by software.

Business science 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 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 article, we use existing requirements engineering practices to add the necessary formalism to business concepts: we use goal hierarchies, goal links 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 with which a reasoning can be performed: first customer demands trigger the selection of resources (benefits), and then a configuration process creates bundles of services that provide these customer benefits.

Our work aims at facilitating automated reasoning processes through conceptualizing and formalizing domain knowledge. Automating reasoning processes is a necessity in order to facilitate complex e-service scenarios. In this paper we show how research from various disciplines can be combined to achieve this goal.

We showed how our approach helped domain experts analyze a complex case study.

So far our research has dealt with demands. We modeled also a set of service quality criteria, but we have not explored how their analysis should differ from that of demands. As we have already seen positive results from the use of the i^* modeling framework [24] in value-oriented modeling [21], adopting i^* for our current research as well seems promising.

Another future research direction concerning service quality is incorporating the SERVQUAL model [25] (which is broadly-used in business science) in the service ontology to describe service quality from a customer perspective using SERVQUAL's generic dimensions that customers use for evaluating service quality.

We have demonstrated how the mapping between the customer perspective and the supplier perspective can be performed by production rules, as modeled with FS-graphs. Future work should further investigate the nature of these production rules. We have noticed so far that some of them relate a demand – specified by (e.g. quality) properties – to a resource itself – disregarding its (quality) properties – whereas others relate a demand to a combination of a resource with (quality) properties.

References

1. *eCl@ss website*. 2005. <http://www.eclass.de>.
2. *UNSPSC website*. 2005. <http://www.unspsc.org>.
3. Hans Akkermans, Ziv Baida, Jaap Gordijn, Nieves Peña, Ander Altuna, and Iñaki Laresgoiti. Value webs: Using ontologies to bundle real-world services. *IEEE Intelligent Systems - Semantic Web Services*, 19(4):57–66, 2004.
4. Ander Altuna, Alvaro Cabrerizo, Iñaki Laresgoiti, Nieves Peña, and Daniel Sastre. Co-operative and distributed configuration. In *Net.ObjectDays (NODE) 2004*, pages 69–80, Erfurt, Germany, 2004.
5. Ziv Baida, Jaap Gordijn, Andrei Z. Morch, Hanne Sæle, and Hans Akkermans. Ontology-based analysis of e-service bundles for networked enterprises. In *Proceedings of The 17th Bled eCommerce Conference (Bled 2004)*, Bled, Slovenia, 2004.
6. Ziv Baida, Jaap Gordijn, Borys Omelayenko, and Hans Akkermans. A shared service terminology for online service provisioning. In *Proceedings of the Sixth International Conference on Electronic Commerce (ICEC04)*, Delft, NL, 2004.
7. Ziv Baida, Jaap Gordijn, Hanne Sæle, Andrei Z. Morch, and Hans Akkermans. Energy services: A case study in real-world service configuration. In *Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAiSE 2004)*, pages 36–50, Riga, Latvia, 2004. Springer-Verlag.
8. L.L. Berry and A. Parasuraman. *Marketing Services: Competing through Quality*. The Free Press, New York, NY, 1991.
9. E. Bigné, C. Martínez, and Maria José Miquel. The influence of motivation, experience and satisfaction on the quality of service of travel agencies. In Paul Kunst and Jos Lemmink, editors, *Managing Service Quality (Volume III)*, pages 53–70, London, UK, 1997. Paul Chapman Publishing Ltd.

10. Pim Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, Universiteit Twente, Enschede, NL, 1997.
11. Hans de Bruin and Hans van Vliet. Top-down composition of software architectures. In Per Runeson, editor, *Proceedings of 9th International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS2002)*, pages 1–10, Lund, Sweden, April 2002. IEEE Computer Society.
12. Hans de Bruin, Hans van Vliet, and Ziv Baida. Documenting and analyzing a context-sensitive design space. In J. Bosch, M. Gentleman, C. Hofmeister, and J. Kuusela, editors, *Software Architecture: System Design, Development and Maintenance; Proceedings of the 3rd Working IFIP/IEEE Conference on Software Architecture (WICSA-02)*, pages 127–141, Montreal, Canada, August 2002. Kluwer Academic Publishers.
13. Paolo Donzelli. A goal-driven and agent-based requirements engineering framework. *Requirements Engineering*, 9(1):16–39, 2004.
14. Anne Flæte and Gregers Ottesen. Telefiasko for viken. *Dagens Næringsliv* (Norwegian newspaper), 13/14 October 2001.
15. Ariel Fuxman, Lin Liu, Marco Pistore, Marco Roveri, and John Mylopoulos. Specifying and analyzing early requirements: Some experimental results. In *Proceedings of the 11th IEEE International Requirements Engineering Conference (RE'03)*, pages 105–114, Monterey Bay, California, 2003. IEEE Computer Society.
16. C. Grönroos. *Service Management and Marketing: A Customer Relationship Management Approach, 2nd edition*. John Wiley & Sons, Chichester, UK, 2000.
17. H. Kasper, P. van Helsdingen, and W. de Vries jr. *Service Marketing Management: An International Perspective*. John Wiley & Sons, Chichester, UK, 1999.
18. P. Kotler. *Marketing Management: Analysis, Planning, Implementation and Control, 6th edition*. Prentice Hall, Englewood Cliffs, NJ, 1988.
19. K. F. Marthinussen. KANKAN som kunne... (presentation). In ITEnergi 2002 Conference, 2002. Available at www.itenergi.com/kari_martinussen.ppt, last visited March 2005.
20. Roland T. Rust and P.K. Kannan. E-service: a new paradigm for business in the electronic environment. *Communications of the ACM*, 46(6):36–42, 2003.
21. Bas van der Raadt. Business-oriented exploration of web service ideas: Combining goal-oriented and value-based approaches. Master's thesis, Vrije Universiteit, Amsterdam, NL, 2004.
22. Axel van Lamsweerde. Requirements engineering in the year 00: a research perspective. In *Proceedings of the 22nd international conference on Software engineering*, pages 5–19, Limerick, Ireland, 2000. ACM Press.
23. Axel van Lamsweerde. Goal-oriented requirements engineering: A guided tour, invited minitutorial. In *Proceedings of RE'01 - International Joint Conference on Requirements Engineering*, pages 249–263, Toronto, Canada, 2001. IEEE.
24. Eric S. K. Yu. *Modelling Strategic Relationships for Process Reengineering*. PhD thesis, University of Toronto, Toronto, Canada, 1995. Also appears as Technical Report DKBSTR-94-6.
25. V.A. Zeithaml, A. Parasuraman, and L.L. Berry. *Delivering Quality Service: Balancing Customer Perceptions and Expectations*. The Free Press, New York, NY, 2001.