e^3*service* : An ontological approach for deriving multi-supplier IT-service bundles from consumer needs

Sybren de Kinderen
VU University Amsterdam
De Boelelaan 1081
1081 HV, Amsterdam, The Netherlands
sdkinde@few.vu.nl

Jaap Gordijn
VU University Amsterdam
De Boelelaan 1081
1081 HV Amsterdam, The Netherlands
gordijn@cs.vu.nl

Abstract

IT-services should not only be considered from a technical perspective, but should also be seen as commercial *services that satisfy a* consumer need*. Examples include well-known services such as Internet access or content provisioning services. Typically, to satisfy a consumer need, a* bundle *of elementary services is required. In such a bundle, each elementary service can be offered by a different supplier. A key problem is then how to actually find service-bundles that satify consumer needs as close as possible. Because IT-service bundles can be automatically provisioned online immediately after ordering, finding a service bundle satisfying a need should preferably also happen automatically. To this end, we propose the* e^{3}service *ontology, which offers constructs from service marketing, but in a computational way, such that automated reasoning support can be developed to match consumer needs with IT-services. This paper presents the* e^{3}service *ontology and explains it by a case from the telecom industry.*

1 Introduction

In recent years, the notion of customizable bundles of IT-services to satisfy complex needs from specific consumers has gained interest. Consider a daily-life example of obtaining internet services from Internet Service Providers (ISPs). Often, the proposition of an ISP is actually a bundle consisting of more elementary services such as IP-based access, an email box, space to host a website, telephony, and access to newsgroups. However, the consumer may prefer a different, perhaps a smaller, bundle; for instance *only* IP-based access plus email plus IP-telephony. Such a bundle then more closely matches the consumer need compared to the original -fits for all- full-service bundle.

Additionally, IT-services are increasingly offered by a networked value constellation, rather than just a single enterprise [22]. By doing so, suppliers can utilize their core competencies, while still satisfying a consumer need. In the ISP-example, the offered bundle can be a multi-supplier bundle: IP-access is then provided by a telecom operator, an email box is offered by a commercial enterprise utilizing economies of scale, as can hold for website hosting, which may be offered by yet another enterprise.

We perceive automatically composing and provisioning such a needs-driven, multi-supplier IT-service bundle as a key problem. In a future scenario we foresee, a consumer would ideally state to the web his IT-needs, and the web (or some intermediate party) responds with a list of candidate multi-supplier IT-service bundles. After selection of a specific bundle by the consumer, the IT-services in the bundle should be provisioned automatically.

Guidelines on creating customized service bundles have already been studied in business literature, most notably by [14],[19],[20] . However, these guidelines are fairly generic (the focus is on services in general and not specifically on IT-services). More importantly, they lack conceptualization and formalization so it is difficult to systematically and (semi-) automatically reason about service bundles. Such reasoning is important, because IT-services, as illustrated by the ISP example, are bought and provisioned online, *enabled by* information technology. To adequately facilitate this buying and provisioning process, the elicitation of IT-needs, as well as the selection of *commercial* IT-services that can be provisioned to satisfy such needs, should be supported by information technology as much as possible.

The contribution of this paper to such a reasoning process is an ontology about *consumer* needs, called *e*^{3}*service* . The ontology relates a need to available IT-services, which are in a service catalog. Moreover, we propose a *gradual* process from need elicitation and statement (essentially the 'problem statement') to a bundle of IT-services (the 'solution'), recognizing that consumers often have already parts

1530-1605/08 $25.00 © 2008 IEEE

of a service bundle in mind, if they articulate their needs. In other words: need (and problem) statement does usually not happen context-free, but already includes knowledge about the kind of available services (solutions) in the market. This phenonema is also known in design and problem solving theory (see e.g. [11]).

It is important to know that we understand IT-services really as *commercial* services: economic activities, deeds and performances of a mostly intangible nature [20], with a focus on those services that can be ordered and provisioned (nearly) online. This is in contrast with web services and related standards such as WSDL [1], BPSS [2], BPEL4WS [4], WSCI [5], and WS-Coordination [9], to name only a few: these services are mainly intended to arrive at a cross-organizational computing platform to facilitate interoperability on a more *technical* level.

This paper is structured as follows. In section 2, we present a comprehensive overview of the bundling reasoning process. As this paper focuses only on a part of the reasoning process, namely consumer-oriented reasoning, we present in section 3 an ontology to represent consumer needs, wants, demand and benefits. In section 4 we apply this ontology to a real-life case study, to reason about potential service bundles. In section 5, we discuss related work on IT-service bundling. Finally, in section 6 we present our conclusions.

2 Reasoning about IT-service bundles

To develop a reasoning process that is capable of generating bundles of IT-services based on a consumer need *and* doing so in a multi-supplier setting, we propose e^3*service* . The idea of e^3*service* is to generate these bundles (semi-) automatically, or in other words; to formalize the reasoning process of needs-driven IT-service bundling as much as possible. To achieve formalized reasoning on service bundling, we take notions from established services and marketing literature and combine and conceptualize them by using computer science techniques. Most notably amongst the formalization techniques we employ is the notion of a formal ontology, which in [8] is defined as an *'explicit formal specification of a shared conceptualization'*. This definition highlights two aspects of an ontology that are of interest to our research: (1) a formal specification, since we want to reason about bundling services (semi-)automatically and (2) the shared conceptualization; we want to reason about service bundling in a networked enterprise, meaning that it is imperative that every organization in this network talks about the same concepts to avoid mismatches. In other words, a *shared understanding* must be present amongst the participating companies for a network of enterprises to succeed. This is important since without such a shared understanding, an organization in a networked enterprise could, for example, interpret the concept of a service as a purely technical web-service, while others could interpret it as a purely commercial service. Finally, we want to ensure that this shared understanding has its ground in concepts that have proven themselves to exist in practice. That is why we base our ontology on *established theory* from service marketing literature.

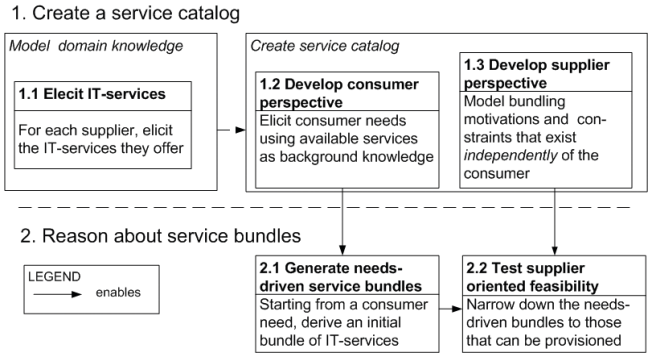

Figure 1. Reasoning about needs-driven service bundling

The high-level steps of e^3*service* are presented in figure 1. We make an explicit distinction between two processes needed for the bundling reasoning: the *creation* of service catalogs (that should happen on beforehand), and reasoning *with* these catalogs about feasible *bundles* on a per consumer-need basis.

1. *Create a service catalog*. This has to be done *before* we can actually reason about service bundles themselves. The aim is to build per-supplier catalogs that describe the services and the needs which can be satisified by these services. Obviously, the catalogs will be used by the service bundling reasoning itself (see step 2). Building this catalog requires three steps:

   1.1 Eliciting the suppliers, and for each supplier, eliciting the IT-services they offer. The outcome of this task is a -per supplier- list of commercial IT-services. These IT-services should as fine-grained as possible, nevertheless it should still be *commercially* feasible to provision each service in its own right.

   1.2 Formalizing each IT-service from a *consumer* perspective. As we will see later on, this comprises an understanding of the consumer need that is satisfied by the IT-service, ultimately in terms of benefits (being features of an IT-service, such as a mail box size in case of an e-mail service). Also, this step explores consumer-side

constraints: these constraints represent which supplementary benefits can (not) be provided given a stated consumer need, by offering to the consumer additional services.

1.3 Formalizing each IT-service from a *supplier* perspective. In this step we reason about bundles of services by reviewing supplier oriented motivations and constraints for bundling. For instance, according to literature on mixed bundling [10], suppliers want to offer a bundle of services for a lower price than for the sum of the prices of the individual services the bundle consists of, because the total revenue for all sold bundles is higher than the total revenue for all individual services. Also, we elicit supplier-driven constraints between services. For instance, a VoIP service puts certain constraints on the QoS-attributes of an internet access service (a technical constraint but important for the allowed bundles). As an example of a commercial constraint, Apple only allows downloaded songs to be played within their own iTunes environment. The important reason to distinguish supplier-driven constraints from consumer-driven constraints is that supplier-driven constraints exist *independently* from the needs of an end-consumer.

2. *Generate bundles of services, using the service catalogs*. As can be seen in figure 1, we create service-bundles in two steps:

2.1 Deriving an initial set of service bundles that covers a consumer need. This step is enabled by step 1.2. The *e*^{3}*service* ontology we propose in section 3 has been designed such that reasoning about this set of bundles becomes possible.

2.2 Narrowing down the set of service bundles to those that are feasible and desirable from a supply side perspective. This step is enabled by step 1.3.

In this paper, we will focus on the *consumer-oriented* steps. This means that we will elaborate upon steps 1.1 and 1.2 to illustrate the creation of a consumer-oriented service catalog, and that we walk through step 2.1 to illustrate how this catalog can be used to generate bundles of services covering a consumer need. Narrowing down the candidate bundles from a supply side perspective (step 2.2), as well as formalizing service catalogs for that purpose, has been extensively discussed in earlier research (see e.g. [3]).

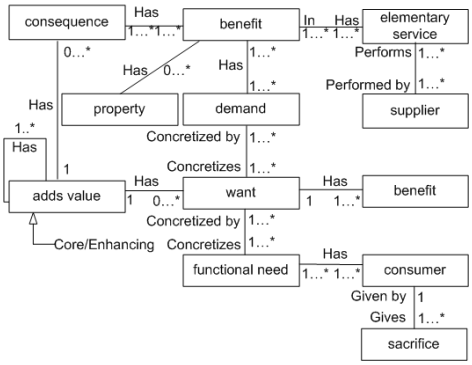

Figure 2. Needs ontology

3 *e*^{3}*service* : An ontology for configuring IT-services based on consumer needs

In this section, we present our ontology (see figure 2) that enables the reasoning process of matching consumer needs with available IT-services. We will discuss the concepts depicted in this ontology individually, and also touch upon the relationships that exist between these concepts.

To illustrate the discussed concepts, we will use the running example of a basic e-mail hosting service. In brief, the example includes a person that has a need to communicate with family abroad (the problem). To do so, he can use email or instant messaging (parts of solutions). Examples of concrete email services are *g-mail* or *hotmail* (being more specific solutions). Such services may have specific benefits (e.g. the allowed size of an email box). A consumer may derive economic value of these benefits (e.g. a certain retention item of their email).

3.1 Elementary service.

An elementary service is the entity which is of economic value to the end-consumer, and which is provisioned by a supplier. It is the smallest unit that, from a commercial point of view, can be meaningfully obtained from a supplier. Typically, elementary services are listed in a service catalog of a supplier. The notion of 'elementary service' allows for connecting the consumer-oriented *e*^{3}*service* ontology to supplier-oriented ontologies (see e.g. [3]).

EXAMPLE: 'e-mail hosting'. Note that a specific property of e-mail hosting, 'mailbox size', is *not* an elementary service , since it can not be provisioned in its own right, but is always connected to the e-mail service it belongs to.

3.2 The need/want/demand-hierarchy

The need/want/demand-hierarchy is a view on marketing as proposed by [6] and later by [18], that emphasizes a gradual transition from a need - a high level problem statement - to a set of elementary services that together provide a solution for that need. In this section, we discuss the concepts from this hierarchy as used in our ontology. Additionally, we will touch upon the concept of a sacrifice, since this concept denotes the trade-off that is present between consumer demand and the cost for satisfying that demand.

Functional need. A functional need represents a problem statement or goal, *independently* of considering a solution direction. We stress a separation of problem and solution here, since a need can usually be covered by multiple solutions [6]. Because of this we need to avoid a bias towards one of these solutions when defining a need, so that promising other solutions are not overlooked.

EXAMPLE: 'communicating with family abroad'. This need statement does not include a notion of a solution yet; in other words, nothing is said about *how* the communication will be done.

Want. In the e^3*service* ontology, a want is something that can be offered by a *single* supplier, with the constraint that this is commercially feasible. We emphasize this notion of a single supplier because ideally, we want to enable *each* organization in the networked enterprise to focus on his/her core competencies. The bundling reason is then about meaningfully combining these single-supplier elementary services into one multi-supplier service bundle, to satisfy a consumer need.

RELATIONS: *Concretizes*: a want concretizes a functional need by specifying an initial solution direction for a problem statement. A need can be concretized into multiple alternative wants since a certain problem can be resolved though multiple solutions.

EXAMPLE: A want satisfying the functional need of 'communicating with family abroad' is 'e-mail hosting'. An alternative want is 'instant messaging'.

Demand. A demand represents a want as provisioned by a *specific supplier*. As such, a demand differs from a want because a demand provides *supplier-specific* values to the properties of a want (see example). We use a strict distinction between wants and demands, because they actually refer to two different steps in the reasoning process about bundles. Since the notion of a want is defined *independently* from a specific supplier, it enables to reason about *the means* we employ to satisfy the need, without having a specific supplier in mind already. In a second step, we reason about *which supplier* is satisfying a want by taking the demands into account. Without explicitly separating these steps, a consumer would be forced to choose *directly* between suppliers, with the danger that the consumer loses sight of what the commonalities and differences between their offered solutions are. Now, the consumer first focuses on choosing certain features (e.g. web-based access in the case of an e-mailing service) that he is interested in, *independently* of who actually provisions these features.

RELATIONS: *Concretizes*: A demand concretizes a want if it specifies the generic want, for instance e-mail hosting, for a specific supplier. A want has generally one or more demands, meaning that one or more suppliers can satisfy a want.

EXAMPLE: 'g-mail' (from Google) is a demand that specifies the want 'e-mail hosting'. For example, 'g-mail' may have a distinguishing property 'mail-box size' that would be different from the 'mail-box size' offered by 'hotmail'.

Sacrifice. A sacrifice represents something valuable to the consumer that has to be given up in order to acquire a service. When making buying decisions, consumers usually trade off their demand for services with the sacrifice(s) needed for acquisition. Note that due to lack of space, the concept of a sacrifice is not discussed in the rest of this paper

RELATIONS: *Given by*: A sacrifice is given by one consumer

EXAMPLE: A consumer sacrifices a monthly fee for a paid e-mail service. Note that, even though money is the most common sacrifice, other types of sacrifices such as time are also possible.

Discussion. The need/want/demand hierarchy provides us with a useful starting point in finding bundles of IT-services based upon consumer needs. The reasoning that can be performed with it is still rather limited however, since the need/want/demand concepts do not take into account the specific *benefits* of a elementary service. This inclusion of benefits in the reasoning process is however important, since they could discriminate two elementary services from one another that at first seem similar - similarity of elementary service offerings being something that is often the case when talking about a network of enterprises. For instance, hotmail and g-mail are similar in the sense that they are both e-mail hosting services, yet they differ with respect to the specific features they offer. In this example, g-mail differentiates itself on the property "'mailbox size"'.

How we intend to make the distinction on the level of benefits, is discussed in the next section.

3.3 Benefits, consequences and value derivations

Benefit. Benefits describe properties that are of economic value to the consumer in the sense of *value in use* [21]. In other words, benefits provide an increase of economic utility to the consumer, through something functional, social (e.g. status) or otherwise. This is in contrast to the notion of *value in exchange* [21], which indicates the amount of revenue generated by *selling* such a property. In sum, to understand the value of a elementary service for a particular entity, the notion of *value in exchange* is convenient for the *supplier*, whereas *value in use* is more appropriate for the *consumer*.

RELATIONS: A want (and a demand also) *has* one ore more benefits.

- A single want has one or more benefits. Benefits of a want do however not have specific values, as benefits exist independently of a specific supplier. For instance, the specific size (e.g. 1 GB) of a mailbox is not specified, it is only specified *that* a mailbox has a size.
- A single demand has one or more benefits. Since a demand is specific for a supplier, benefits of a demand *do* have specific values. For instance, in the case of the size of a mailbox, the size (e.g. 1 GB) would be specified for the specific supplier.

EXAMPLE: In case of an e-mailing service, a specific benefit could be 'customized domain'. A customized domain allows for customizing an e-mail address, so instead of art.vandelay@someunchangebledomain.com a customized e-mail address would be art@vandelay.com. A customized domain is a benefit because an e-mail service with a customized domain gives the consumer more status, heightened stature being a measure of more value in use.

Discussion. As mentioned, benefits are important when discriminating between two apparently similar services. However, we also incorporated the notion of benefits in our ontology to enable a closer match between consumer needs and the services providing a solution for these needs. Consumers namely, as is also stated in [12], are also interested in the benefits provided by a certain service, such as a large mail box size or web-based e-mail access. Suppliers on the other hand provide services with *generic* sets of benefits, with the result that a consumer might acquire a service containing benefits in which he/she is not interested. Or even worse: that a service insufficiently performs on a benefit in which the consumer is interested (such as web-based access with an interface that is not user-friendly) or that the deisred benefit is not contained in a service at all. The problem then is to find as close a match as possible between the benefits the consumer is interested in and the benefits contained within the services offered. The incorporation of the benefit-concept in our ontology addresses this issue, additionally to making a distinction between two apparently similar services.

Consequence. This concept represents the *subjective* added value for the end-consumer that is gained directly through obtaining a benefit provided by a service. Deriving consequences from benefits is based on means-end chaining, in which a technique called *laddering* [16] is employed. Laddering shows how a specific benefit ultimately contributes to satisfying a consumer need. Laddering shows this contribution of a benefit to a need by using the intermediary step of a *consequence*.

Deriving a consequence from a benefit is done by asking the question 'what happens when we consume a elementary service in which benefit X is contained?'. In the next paragraph, we will discuss the concept of 'value derivation', which provides basic guidelines that aid in reasoning on *how* value is derived from benefit.

RELATIONS: A benefit *has* one or more consequences. Multiple benefits can point to the same consequence.

A consequence *contributes to* a functional need.

EXAMPLE: The benefit 'web-based e-mailing access' allows for the consequence of 'cost-effective communication'. Additionally, it also allows for the consequence of 'accessing mail at any site with internet access'. 'cost-effective communication' ultimately contributes to satisfying the need of 'communicating with family abroad'.

Value derivation. In our ontology, we reason about value derivation as a result of consuming a certain benefit, by using a consumer value framework presented by Holbrook et al [17]. This framework, which originates from the field of axiology, is used to explain *how* end-consumers derive value while consuming a product/service. It consists of three main categories, each with two opposite dimensions. By means of choosing a combination of three dimensions - a single dimension from each category - one ultimately arrives at a singular value derivation; this combination then also explains how the value derivation should be interpreted.

As an example of the opposing dimensions, consider active vs reactive use: active use means that a consumer does something to a product/service as part of the consumption process (eg. a nicely decorated cup used by a roman emperor to drink wine) while reactive use means that the product/service acts upon the consumer instead (eg. the decorated cup on display in a museum). Examples of sigular value derivations that arise from a combination of the dimenions are status and efficiency. For a more detailed discussion on these dimensions and value derivations, see [17].

EXAMPLE: The benefit 'customized domain' from an e-mail service, can be annotated with the value derivation 'status', resulting in the consequence of 'enhancing status through personalized e-mail address'.

3.4 Incorporating dependencies between wants

The notion of dependencies, as discussed in [7], indicates that two services are related to one another. This relation can exist from a *supplier* perspective; for instance a paid e-mail service cannot be delivered without some basic administrative services such as billing. On the other hand, such a dependency can exist from a *consumer* perspective; for instance a spam filter adds value for the consumer when bundled with an e-mail hosting service. In accordance with the scope of this paper, we will only discuss consumer-oriented dependencies.

Adds value. As benefits have consequences (in terms of economic value for the consumer), the wants, that actually aggregate benefits from several demands, implicitly also have consequences. We will illustrate this aggregation of benefits on a want-level further when we discuss the case study in section 4. For the discussion of this concept, it is sufficient to know that wants have benefits and as such, also consequences. In figure 2, a relationship between a want and a consequence has been made explicit.

This relationship indicates that when a consumer has a specific want, other wants and belonging consequences may also become relevant. The consequences inherent to these other wants then explain *why* this relation is present.

In our ontology, we incorporated one such relationship (adapted from [7]): Core/Enhancing(C/E). This relationship indicates that a service B is able to provide added value when bundled with a certain core service A. A constraint in this dependency is that service B cannot be acquired independently from service A. Note that there are more types of relationships between wants that need to be reviewed from a consumer-perspective; due to lack of space however, we cannot elaborate on them in this paper.

RELATIONS: An Adds value relationship *contains* a single want and a single consequence. This pair represents a commercially feasible offering, plus part of the subjective value gained from consuming a benefit contained within this offering.

has Adds value has a relationship with one or more other adds value relationships. By this, we mean that a relationship exists between two or more pairs of wants and consequences. The basic idea behind this relationship is to show that a relationship exists between two services by using the concept of a want, and to show to the consumer *why* this relationship exists by using the concept of a consequence.

EXAMPLE: The want 'email' can be supplemented with the want 'spam filter'. Since a spam-filter is only relevant in combination with an e-mailing service, a C/E relationship exists between these two wants. The relationship can then be explained by the consequence "increase number of relevant mails' from the benefit 'spam reduction', the latter being contained in the 'spam filter' want.

4 A real-life case study with *e*^{3}*service* on telecommuncation services

We now illustrate how the *e*^{3}*service* ontology works in a real-life case study about telecommunication services, by following the earlier presented steps.

4.1 Step 1.1 - Elicit service suppliers and the offered elementary services.

The aim of this step is to get a list of IT-services of the service supplier(s), in this case KPN.

We do this by eliciting the IT-services that KPN offers to its end-consumers. From KPN, we found four IT-services: (1) VoIP, (2) IP-connectivity, (3) homepage capability, (4) e-mail, and (5) spam filter. In this case study, we performed this elicition by constructing an initial service catalog based upon documentation received from KPN. For validation of the initial service catalog, we then held a feedbacksession with an enterprise architect from KPN.

However, the found services from KPN could have equally well been provided by separate enterprises. Each enterprise then focuses on his/her core competencies. In the case study, we also considered seperate organizations that provide services similar to KPN. We did this to illustrate how we make a distinction between two wants that at first appear similar. Due to lack of space however, we chose not to show these alternative services in this step. Instead, we included the services and the benefits we found, while considering a multi-enterprise setting, directly in the service catalog as discussed in step 1.2.

4.2 Step 1.2 - Formalize each IT-service from a consumer perspective.

We build up the service catalog in a *bottom-up fashion*; we take the services found in step 1.1 as a starting point, and on the basis of them, elicit the needs, wants and demands they cover, as well as the benefits.

4.2.1 Populate the service catalog with demands and find the benefits contained within them

The first step is to populate the service catalog with the services modeled in step 1.1. These services are actually sim-

ilar to demands, since they are the services as provisioned by the specific suppliers. Next, we add the benefits as contained in the demands. Benefits are elicited by reviewing the specific properties of a service that provide the end-consumer with more value in use. For instance, the property of web-based e-mail access is a benefit because it allows a consumer to access mail at any site, without having to install a seperate mail-client.

However we did not yet model these benefits in the first step. So, we should now make the benefits in the populated catalog explicit for each of the demands modeled in step 1.1. For example, 'call to fixed lines, 'call to mobile', 'number recognition' and 'voice mail' are all benefits from KPN's VoIP service. These benefits again illustrate that a demand is not the same as a benefit, since 'voice mail' for instance cannot be viably delivered on its own.

The resulting explication of benefits and the demands that contain them, can be found in figure 3.

4.2.2 Derive wants

On the basis of the demands, we elicit wants. We first abstract away from the *specific values* that the suppliers give to their benefits. So, for instance, in the case of a customized domain we abstract away from the supplier-specific property of providing you with a personalized e-mail address that ends with '.nl'. After having made this abstraction, the services that contain these benefits become the wants. If there are multiple similar services available from multiple suppliers, there will be a merge of these multiple services into a single want. For instance, a demand 'VoIP' as provisioned by the specific suppliers KPN and Abel becomes a want 'VoIP', *independently* of these suppliers. This single want will then also inherit the benefits from these different service objects. To illustrate this, consider the want 'VoIP' in figure 3. This want contains not only the benefits from KPN's VoIP service, but also the benefits from a different supplier, namely Abel.

4.2.3 Using consequences to show how benefits contribute to satisfying a functional need

Next, we derive the consequences from the benefits by asking the question: 'What happens when we consume a service in which this benefit is contained?'. As mentioned in section 3.3, we use the value derivation framework from Holbrook to aid us in this process. To illustrate, take two benefits from our VoIP service modeled in figure 3: 'call to mobile' and 'call to fixed lines'. Now, by using the value derivation 'efficiency' from Holbrook, we can state that both of these calling options - to mobile and fixed lines - are more cost-effective when employed through a VoIP calling-service than they would be from another calling-service (eg. a POTS or mobile calling-service). However, when applying the value derivation 'status' to these two benefits we can find yet another, at first sight less obvious consequence namely that - when calling via VoIP - one can use it as a gadget, since it a relatively new phenomenon.

By using the consequences we can now derive a set of needs. We do this by considering the goal that is achieved through a consequence. For example, in the case of the consequence of employing VoIP as a gadget, we can ultimately derive the need for more social stature.

However, there can also be cases in which we are not able to define a new need on basis of a consequence. In the service catalog from this case study (figure 3), examples of such consequences are an 'increase in relevant calls' and an 'increase in availability' neither of which achieve new goals on their own terms. However, we *can* usually show that these consequences positively *contribute* to satisfying a need. For instance, an 'increase in relevant calls' aids in making the direct communication easier in case of a VoIP-service. Of course, this positively-contributes-to relation is only valid in case the want in which the benefit is contained is actually acquired.

4.2.4 Define relationships between wants

Now that we have defined both the wants and the consequences, we can define the 'adds value' relationships that exist between wants. In the case study, this relationship can be illustrated by the C/E dependency that exists between the wants 'e-mail hosting' and 'spam filter'. This dependency has already been discussed in our ontology; therefore, we suffice with a short summary. A spam filter could, from a consumer perspective, add value to an e-mail hosting service, where the consequence of one of the benefits in the spam filter service -'increase in relevant e-mails' - indicates *why* it could add value to the e-mail hosting service.

4.3 Step 2.1: Generate needs-driven service bundles using the service catalog

In this section, we will briefly illustrate how to use the catalog (see figure 3) for finding service bundles. For this purpose, we imagine that there are two different end-consumers that are both interested in increasing their communication possibilities:

- A *student*, who is seeking an alternative for the relatively expensive mobile phone and wants a private e-mail address besides the one provided by his college.
- A *yup* that, besides increasing his communication abilities, mainly wants to show that he is a front-runner when it comes to employing technology in everyday life.

Now the student would start at (1) the need of communicating directly to search for an alternative for the mobile phone and (2) at the need of communicating indirectly where he expects to find an e-mailing service. The yup on the other hand, besides increasing his communication capabilities, mainly wants to employ technology to increase his social stature. In his case, we consider this to be his main concern.

For the student and the yup, we have now defined two different set of needs: 'increasing (in-)direct communication' for the student and 'increasing social stature' for the yup.

The next step is looking at the consequences belonging to these needs; in case of the student, these would be 'cost-effective communication' and 'cost-effective calling'. For the yup, these would be 'personalization of e-mailing' and 'possessing a gadget', *either one of which* (indicated by the OR-annotation in figure 3) provides for a possible increase in stature.

Based upon the services that contain these consequences, we are now able to derive an initial set of wants: 'e-mail hosting' and 'call via VoIP' for the student and customized domain' and 'call via VoIP' for the yup.

Next, we look whether there is a core/enhancing relationship present between the wants. According to our catalog, one such dependency is present between a 'customized domain' and 'e-mail hosting' and since 'customized domain' is the enhancing service object, it can in this case only be acquired in combination with 'e-mail hosting'. Also, concerning this dependency we can see that an 'e-mail service' can be bought in combination with 'a spam filter'

Thus, so far we have defined already a fairly elaborate set of possible services that satisfy the functional needs. For the student, these are 'call via VoIP' and 'e-mail hosting', with the optional service 'spam filter' For the yup, these are 'e-mail hosting' in combination with 'customized domain' (one cannot be provisioned without the other in this case) with an optional 'spam filter' or 'call via VoIP'. Note that we did not include the 'customized domain' in a possible bundle for the student, even though there is a C/E-relationship present between e-mail hosting and customized domain. This is because he was never interested in satisfying the need of increasing social stature, which, as we have seen, is linked to the want of a customized domain.

Next, we look at the specific benefits present in the wants and whether these are of interest to the consumer. We need to do this because we have not yet reviewed all benefits from the wants and as such, also not how these benefits contribute to satisfying a need. In our example, we assume that the student is interested in a VoIP-service with the benefit 'number recognition' since it increases the number of relevant calls and as such contributes to satisfying the need communicating directly.

We now have a list of services, and the benefits contained within them that could be of interest to the consumer. These benefits are those that directly satisfy a need - e.g. the benefit of calling to fixed lines enables direct communication - and those that benefits that positively contribute to satisfying a need -e.g. number recognition. The last step is to review which actual service offerings from specific suppliers provision these benefits. In case of the student, we can see that he is best off with the [VoIP by KPN] service when it concerns VoIP, since this provides him with number recognition whilst the other provider does not. Concerning e-mail, he can choose for both [e-mail hosting by KPN, spam filter from KPN], [e-mail hosting by sylconia, spam filter by K&K]. In the case of the yup , [VoIP by Abel] and [VoIP by KPN] are valid bundles for VoIP, since he does not have a preference concerning a specific benefit. Concerning a customized domain, [e-mail-hosting by sylconia/KPN , customized domain by sylconia] and [e-mail-hosting by Sylconia/KPN, customized domain by sylconia, spam filter by K&K] are valid bundles for the yup.

5 Related work

The *Business Motivation Model* (BMM) [15] is a model representing *ends* (goals, objectives) that are to be achieved by *means*. It abstracts away from implementation issues such as the business processes necessary to provide for the means. In comparison to our work, BMM does not explicitly assist in deriving consumer needs from a set of IT-services. Also, it does not take a multi-supplier perspective.

Concerning the usage of an ontology, e^3*service* is similar to the e^3*value* -ontology from requirements engineering [13]. As is the case with BMM however, e^3*value* is not specifically aimed at IT-services. Additionally, the concepts from e^3*value* are not specifically aimed at analyzing consumer needs.

Serviguration [7] (service configuration) provides computer supported reasoning about general service bundles. Case studies in the realm of electricity supply and healthcare have shown that by using this methodology, meaningful bundles of services can generated semi-automatically [7]. Moreover, given the -per case study- supplier-oriented service catalogue started with, in principle a significant amount of different bundles are possible (millions), which *serviguration* reduced by its reasoning process to a few relevant bundles (tenths), based on stated consumer needs, and supplier-oriented relationships (and constraints) between elementary services. So, serviguration is a good first attempt to arrive at automated configuration of a networked value constellation, in which a series of suppliers satisfy an need by bundling services. However, serviguration concentrates on conceptualizing services mainly from a *supplier* perspective and while it does have a ontology for taking con-

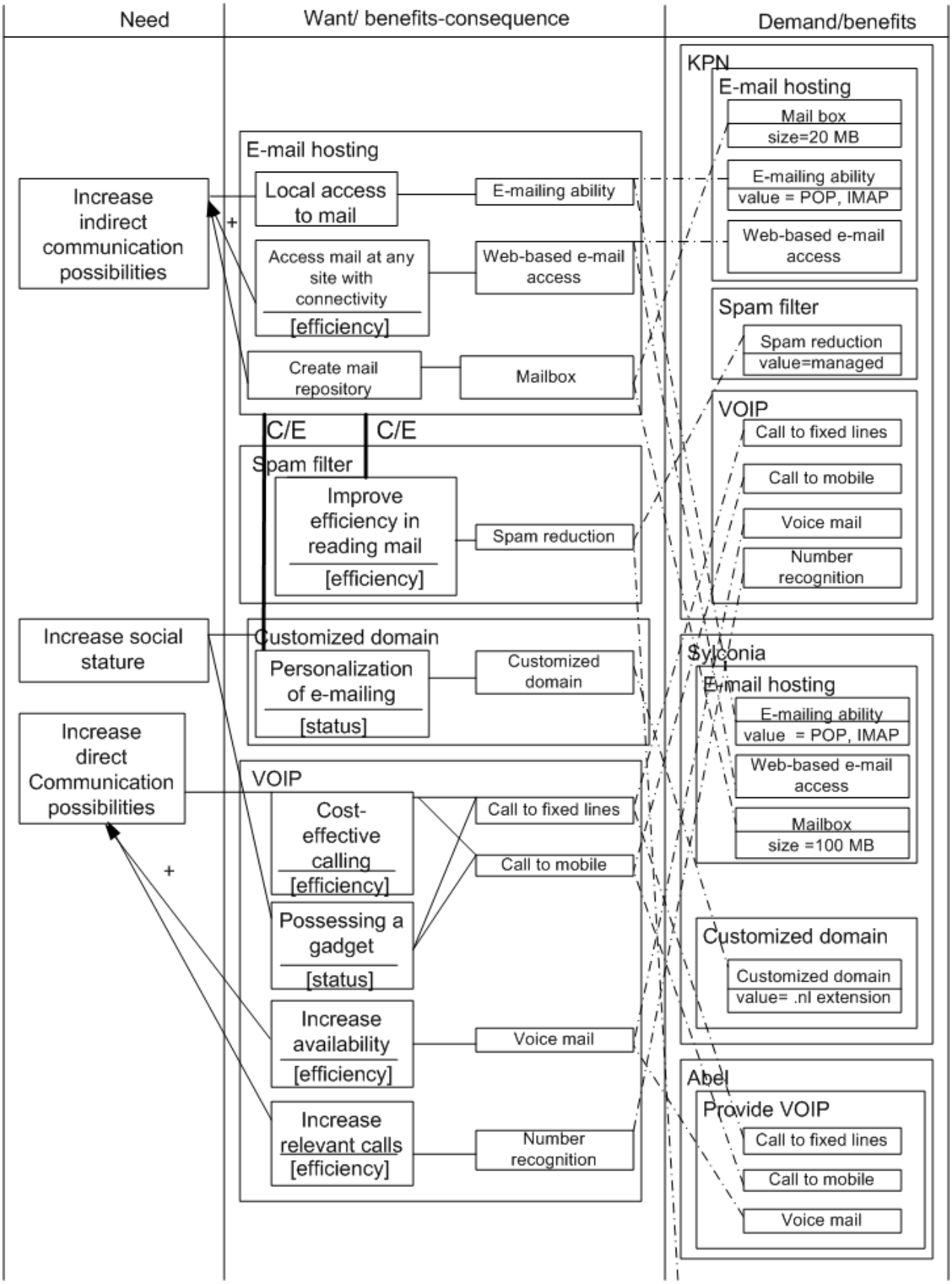

Figure 3. The commercial services catalog in the VoIP case study

sumer needs into account, this needs ontology is rudimentary. Most importantly, the needs ontology from serviguration does not include the concept of a benefit, while this inclusion is important to differentiate between two apparently similar service offerings.

Finally, [23] presents a methodology for evaluating how a business problem can be tackled by combining IT-services from multiple actors in a networked business setting. Yet this methodology differs from *e*^{3}*service* because (1) it does not try to formalize the bundling process as much as possible and (2) it is not specifically aimed at the end-consumer.

6 Conclusions and further research

In this paper, we have shown how a catalog of IT-services can be created in a structured manner by applying a formal ontology. Also, we have presented how we can reason about creating IT-service bundles on the basis of such a catalog. Additionally, this paper clarified that there is a difference between what is offered to the consumer, being the the services, and the features the consumer is interested in, being the demands. Usually, there is a mismatch between the set of benefits contained by a *service*, and the benefits contained by a *demand*.

We would like to point out that the *e*^{3}*service* -ontology has a somewhat idealized vision of discovering needs. This is because it assumes that all consumers are able to articulate their needs precisely and use the same phrasing for this articulation. In reality however, many consumers might not be that rational. For instance, consumers might phrase their needs differently from those presented to them in the consumer-oriented service catalog. However, we explicitly chose to exclude such irrational consumer behavior from *e*^{3}*service* , even though it is sometimes inherent to the discovery of customer needs. We did this because in the end irrational behavior is hard to capture formally, which is in conflict with our research goal of automating the process of needs-driven IT-service bundling as much as possible.

Currently, we are working on software support for the *e*^{3}*service* ontology.

Acknowledgements. The authors thank Leo Stout (KPN) and Peter Hermans for their input on the case study. This research has been partly funded by NWO/STW/Jacquard as the project VITAL.

References

[1] Web services description language (wsdl) version 2.0, http://www.w3.org/TR/2007/PR-wsdl20-primer-20070523/.

[2] ebXML business process specification schema version 1.01, http://www.ebxml.org/specs/ebBPSS.pdf, 2001.

[3] H. Akkermans, Z. Baida, and J. Gordijn. Value webs: Ontology-based bundling of real-world services. *IEEE Intelligent Systems*, 19(44):2332, July 2004.

[4] T. Andrews et al. Business process execution language for web services version 1.1, http://www.ebxml.org/specs/ebBPSS.pdf. Technical report, 2003.

[5] A. Arkin et al. Web service choreography interface (WSCI) 1.0, http://www.w3.org/TR/wsci/. Technical report, 2002.

[6] J. Arndt. How broad should the marketing concept be? *Journal of Marketing*, 42(1):101–103, January 1978.

[7] Z. Baida. *Software-aided service bundling*. PhD thesis, Free University Amsterdam, May 2006.

[8] P. Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, Universiteit Twente, Sept. 1997.

[9] L. Cabrera et al. Web services coordination (WS-Coordination), ftp://www6.software.ibm.com/software/developer/library/ws-coordination.pdf. Technical report, 2003.

[10] S.-Y. Choi, D. O. Stahl, and A. B. Whinston. *The Economics of Doing Business in the Electronic Marketplace*. MACMillan Technical Publishing, Indianapolis, IN, 1997.

[11] N. Cross. Creative cognition in design: processes of exceptional designers. In *C&C '02: Proceedings of the 4th conference on Creativity & cognition*, pages 14–19, New York, NY, USA, 2002. ACM Press.

[12] M. Fishbein. *Belief, attitude, intention and behavior : an introduction to theory and research*. Addison-Wesley, 1978. third print.

[13] J. Gordijn and H. Akkermans. Value based requirements engineering: Exploring innovative e-commerce ideas. *Requirements Engineering Journal*, 8(2):114–134, 2003.

[14] C. Grönroos. *Service Management and Marketing*. Lexington Books, 1990.

[15] B. R. Group. The business motivation model - business governance in a volatile world. Technical report, 2005.

[16] J. Gutman and T. Reynolds. Laddering theory-analysis and interpretation. *Journal of Advertising Research*, 28(1):11, Febraury/March 1988.

[17] M. Holbrook"'. *Consumer value; a framework for analysis and research*. Routledge, 1999. 1st edition.

[18] P. Kotler. *Marketing Management*. Prentice Hall, 2000.

[19] C. Lovelock. *Service Marketing - People, Technology, Strategy*. Prentice Hall, 2001. Fourth edition.

[20] R. Normann. *Service Management - strategy and leadership in service business*. Wiley, 2000. third edition.

[21] J. Ramsay. The real maening of value in trading relationships. *International Journal of Operations and Production Management*, pages 549–565, 2005.

[22] D. Tapscott, D. Ticoll, and A. Lowy. *Digital Capital - Harnessing the Power of Business Webs*. Nicholas Brealy Publishing, 2000.

[23] B. van der Raadt', J. Gordijn, and E. Yu. Exploring web services from a business value perspective. *Proceedings of the 13th IEEE International Conference on Requirements Engineering*, 25(6):53–62, 2005.