Reasoning About Substitute Choices And Preference Ordering in e-Services

Sybren de Kinderen¹ and Jaap Gordijn¹

VU University Amsterdam De Boelelaan 1081 1081 HV, Amsterdam, The Netherlands {sdkinde|gordijn}@few.vu.nl

Abstract. e-Services are just like normal services, but can be ordered and provisioned via the Internet completely. Increasingly, these e-services are offered as a multi-supplier bundle of elementary services. How to automatically compose and prioritize these multi-supplier e-service bundles is considered as a key problem. In this paper, we present the e^3 service ontology to represent a multi-supplier e-service catalogue from a consumer need perspective. Then, we use this ontology to reason about alternative e-service bundles satisfying a particular need, and to prioritize the found bundles using the consumer benefits they provide. The ontology and the reasoning process are illustrated by a case study in the Dutch telecommunication industry.

Keywords: e-services, ontology, service bundling, consumer needs

1 Introduction

In recent years, *customizable* e-service bundles, satisfying complex consumer needs have gained interest. We understand e-services as *commercial* services: economic activities, deeds and performances of a mostly intangible nature. Web services, such as WSDL [6] and others, are a useful *technical* implementation platform for e-services but do not really recognize the commercial perspective on services. Consider e.g. the daily-life example of a *specific* consumer need for Internet access and email. Often, the proposition of an ISP is then a *general purpose* e-service bundle, consisting of more elementary e-services such as IP-based access, an email box, space to host a website, telephony, and access to newsgroups. However, the original, individual, consumer need *only* requires the provisioning of an IP-based access/email e-service. The latter bundle more closely matches the consumer need compared to the -fits for all- full-service bundle.

Additionally, these e-services are increasingly offered by a *networked value* constellation, rather than just a single enterprise [19]. 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 telecommunication 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 the automatic composition and provisioning of such a customized, needs-driven, multi-supplier e-service bundle as a key problem. In a future scenario we foresee that a consumer would ideally state to the web his needs, and the web (or some intermediate party) responds with a list of candidate multi-supplier e-service bundles, which are sorted according to how well they fit to the stated consumer preferences. After selection of a specific bundle by the consumer, the e-services in the bundle should be provisioned automatically.

Guidelines on creating customized service bundles have already been studied in business literature, most notably by [12],[16],[17]. However, these guidelines are fairly generic (the focus is on services in general and not specifically on eservices). 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 e-services, as illustrated by the ISP example, are bought and provisioned on line, *enabled by* information systems. To adequately facilitate this buying and provisioning process, the elicitation of needs, as well as the selection of *commercial* e-services that can be provisioned to satisfy such needs, should be supported by information systems as much as possible.

In earlier work [7], we have presented the $e^3 service$ ontology that allows for the structured creation of service bundles based upon consumer preferences. However, since often multiple, and alternative, service bundles are possible, the next question is then how to rank the bundles according to the consumer need. Therefore, in this paper we show how to (1) reason about substitute services, and (2) assign a *preference ordering* to found service-bundles, based on a consumergiven prioritization of the benefits (s)he wants to obtain. Additionally, we involve pricing of the service bundle in the reasoning process about preference ordering. The contribution of this paper therefore is that we provide a framework allowing *semi-automated* reasoning about multi-supplier, commercial-service bundles.

This paper is structured as follows. In section 2, we provide a comprehensive overview of the e^3 service ontology. In section 3 we apply this ontology to a case study, to create a consumer-oriented catalogue of e-services. Based upon this catalogue, we then show *how* we reason about preference ordering when creating e-service bundles. Finally, in section 4 we provide a discussion, and in section 5 we present our conclusions.

2 The e^3 service ontology

To make this paper self-contained, we summarize the e^3 service ontology (see figure 1, and [7] for more details). This summary is organized by clustering the concepts in the ontology as follows: (1) the need/demand/want hierarchy, (2) benefits, consequences, and value derivations, (3) dependencies between want/consequence pairs, and (4) services. The ontology is based on established service marketing literature (e.g. [12], [16] and [17]).



Fig. 1. The e^3 service ontology

2.1 The need/want/demand-hierarchy

The need/want/demand-hierarchy emphasizes a gradual transition from a *need* - a problem statement - to a set of services that together provide a solution for that need, called a *demand* (see [2] and [15]). For the e^3 service ontology, this results in the following concepts.

Functional need. A functional need represents a problem statement or goal, *independently* from a solution direction [2].

Example. A consumer may have a need to 'communicate with family abroad'. This need does not include a notion of a solution yet, as nothing is stated about how the communication will be done.

Want. A need (problem) can usually be covered by multiple *alternative* wants (solutions) [2]. Also, a need may require *multiple* wants that each satisfy a need partly, but together do so completely. The separation of problem and solutions is important to avoid that we overlook alternative wants (solutions) for needs (problems) during the elicitation process. In doing so, a want does not indicate a specific (named) supplier satisfying the stated need yet. Thereby, we explicitly separate the finding of a *general* solution for a need (a want), from finding a *specific* supplier doing so (a demand). However, as a want indicates a solution available in the market, at least *one* supplier should be willing to provide the solution.

Relations. A want *has* one or more benefit ranges (to be detailed later), which, in short, are properties of a provided service, which are economically valued by a consumer. For a want, these properties are defined independently from a specific supplier.

Example. A want satisfying the need 'communicating with family abroad' is for example 'e-mail hosting'. A benefit for 'e-mail hosting' is a certain mail-box size range (eg. 500 MB to 1 GB). An alternative want is 'instant messaging'.

Problem/solution pair. A want is a (partial) *solution* for a functional need, which is stated by a problem/solution pair. Problem/solution pairs can be related to each other to state that (1) they are *alternatives* for each other, or (2) together they form a *complete* solution for a problem.

Example. 'e-mail hosting' and 'instant messaging' are alternative wants for satisfying the need 'communicating with family abroad'. 'E-mail hosting' plus 'IP-connectivity' exemplify partial problem/solution pairs, which together satisfy the need 'communicating with family abroad'.

Demand. A 'want' is provisioned by a specific supplier as a demand. A demand differs from a want, as a demand provides supplier-specific values to the properties for benefits of a want. We use a strict distinction between wants and demands, because they refer to two different steps in the automated reasoning process about substitutes and preference ordering. In the first step, we reason about the required *generic* benefits, as contained by a want, to satisfy a need, independently from a supplier. In the second step, we reason about the *specific* suppliers who can satisfy a need in terms of a demand, with specific benefits. This simplifies the reasoning process as the customer first only focuses on choosing the required benefits (e.g. a mailbox with a small/big size) without a supplier in mind, and thereafter chooses specific properties benefits as offered by a specific supplier (eg. a 1MB sized box).

Relations. A demand *concretizes* a want if: (1) each benefit of the want matches with a benefit of the demand, meaning that (2) for each benefit, the supplier-specific benefit of the demand is in the specified range of the corresponding benefit for the want. Usually, a want has one or more demands, meaning that one or more suppliers can satisfy a want.

Example. 'Gmail' (from Google) is a demand that specifies the want 'email hosting'. For example, 'Gmail' may have a distinguishing benefit 'mail-box size=1 GB' that would be different from the 'mail-box size=0.5 GB' as offered by 'Hotmail'. However, both benefits fall into the benefit range '500 MB to 1 GB' as specified for the want.

2.2 Benefits, consequences and value derivations

Benefit range and benefits. Benefits describe properties that are of economic value to the customer in the sense of value-in-use [18]. In other words, benefits provide an increase of economic utility to the customer, through something functional, social (e.g. status) or otherwise. A benefit is also used to connect *demands*, as needed by the customer, to *services*, as provided by the supplier. Often, there is a mismatch between the *set* of benefits as contained by a customer demand, and the *set* of benefits as contained by a supplied service. In our work, we assume that the customer and the supplier use the *same* terminology to represent the customer/supplier benefit itself, so ontologically, these benefits are the same (although specific values may differ obviously). Reasoning about a match between a found consumer demand and available supplier e-services, is then about finding a multi-supplier e-service bundle with a set of benefits, that comes closest the required set of benefits as contained by the customer demand.

A benefit *range* is a more general construct, which specifies a range of values a benefit may have.

Relations. First, a want has one or more benefit ranges. Since a want exists independently of a specific supplier, benefits on the want-level do not possess supplier-specific values. Instead, benefits on the want level have a range of possible values, within which a supplier-specific benefit could fall. For instance, in the case of the size of a mailbox, a range could be 500 MB-3 GB. Second, a single demand has one or more benefits. Since a demand is specific for a supplier, benefits of a demand have supplier-specific properties. In the case of the size of a mailbox, the size could for instance be 2.6 GB for a specific supplier. Third, a specific benefit is for one benefit range, and a range can have multiple benefits that fall-in the benefit range.

Consequence. A consequence represents the subjective added value for the endcustomer if he consumes a benefit (falling into a certain benefit range). In the reasoning process as presented in section 3, deriving consequences from benefits is based upon the laddering-technique from means-end chaining [13]. In brief, this is done by asking the question 'what happens if we consume service X in which benefit Y is contained? '.

Relations. A benefit range *has* one or more consequences. Multiple benefit ranges can point to the same consequence. A consequence indirectly contributes to satisfying a need, via the benefit range, demand, and want of that need.

Example. The benefit 'web-based e-mail access' has the consequence 'costeffective communication'. 'Cost-effective communication' contributes to satisfying the need 'communicating with family abroad'. Considering an example of a benefit with a range of values, we can define the consequence 'have a large mail box' based upon the range 1GB-3GB for the benefit 'mail box size'.

Value derivation. We reify the relation between 'benefit range' and 'consequence' by introducing the concept of *value derivation*. While *eliciting* a service catalogue, we reason about value derivation as a result of consuming a certain benefit, by using a consumer value framework presented by Holbrook et al [14]. This framework, which originates from the field of axiology, is used to explain *how* end-consumers derive value while consuming a product/service. Note that a framework as proposed by Holbrook serves as a 'plug-in'. In case of business-tobusiness services, value derivation will be done entirely differently, and so other frameworks should be used. Since the focus of this paper is on finding appropriate e-service bundles given a certain need, and not on eliciting the e-service catalogue itself, we do not elaborate further on value derivation.

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'.

2.3 Dependencies between want/consequence pairs

The notion of service-dependencies (see [3]) indicates that services may depend on each other. For instance, a service can serve as an option for another service, or a service may exclude meaningful consumption of another service. In [3], this relation has only been investigated from a *supplier* perspective; e.g. a paid e-mail service cannot be delivered without a billing service. We have found that such dependencies can also exist from a *consumer* perspective; e.g. a spam filter adds value for the customer if it is bundled with an e-mail hosting service.

Adds value and dependency. As benefits of wants have economic value consequences for the customer, the wants themselves also have consequences. In $e^3 service$, this is represented as a reified 'adds-value' relationship between one want and one consequence. Obviously, each want and consequence may be in many of these relationships.

We have found two specific kinds of *dependencies*, which may exist between two or more 'adds value' relations (so between want/consequence pairs). In a 'Core/Enhancing' (C/E) dependency, a want/consequence pair B provides added value if bundled with a want/consequence pair A. Pair B cannot be acquired independently from A. In a 'Optional Bundling' (OB) dependency, a want/consequence pair B adds value to a want / consequence A. Yet, in case of an OB relation, A and B can also be acquired separately.

These dependencies may exist between multiple want/consequence pairs ('adds value'), as shown by the concept *dependency* in the e^3 service ontology.

Relations. First, 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. Second, 'adds value' has a relationship with one or more other adds value relationships, via the 'dependency' concept.

Example. The pair 'e-mail' (want)/'local access to mail' (consequence) is in a Core/Enhancing dependency with pair 'spam-filter' (want)/'reduction in number of unwanted e-mails' (consequence). So, the want 'e-mail' is related to the consequence 'reduction in number of unwanted e-mails' from the want 'spam filter', where the consequence from latter want indicates *why* this relationship exists. Note that a Core/Enhancing relationship is present, because an acquisition of a spam-filter only makes sense in combination with an e-mail service.

2.4 Service

Service. A service is of economic value to the end customer, and is provisioned by a supplier. It is the smallest unit that, from a commercial point of view, can be obtained from a supplier. Services are listed in a service catalogue of a supplier. The notion of service allows for connecting the customer-oriented e^3 service ontology to supplier-oriented ontologies (see e.g. [1]).

Relations. First, a service is *supplied by* precisely one supplier, since a service is supplier specific. Obviously, a supplier can *supply* multiple services. Second, a service *has* one or more benefits. These benefits are the source for matching supplier-services with benefits that belong to wants.

Example. An 'e-mail hosting' service is an is example of a service. **Sacrifice.** A sacrifice represents something valuable to the consumer and supplier that has to be given in return, in order to acquire a service. *Relations.* A service *requires* one or more sacrifices. This models that a consumer is not willing to obtain a service against any price, but rather is confronted with a budget-constraint, and therefore is limited in demands (s)he can have.

Example. Based upon a monthly fee (e.g. $40 \in$) (sacrifice) that has to be paid for the service '4 Mb/s Internet access', as well a contract-duration of minimally one year (also a sacrifice), a consumer may decide to revise his/her demand, such that a '1 Mb/s Internet access' service for a monthly fee of 10 \in will be selected.

3 A case study on e-service substitution and preference ordering in the TelCo industry

3.1 An e-service catalogue

We now show, for the need 'communicating with family abroad', how $e^3 service$ can be used to (1) make a choice between *substitute* services (different services that satisfy a similar need (e.g. 'instant messaging' and 'VoIP')), and (2) make a choice between *similar* services (e.g. two similar 'VoIP' services but of different suppliers). To this end, we fist need an e-service catalogue, which is shown in figure 2. In brief, we create such a catalogue by considering the e-services as available in the market, and then by deriving the needs these services could satisfy. For a detailed description of how to create such a catalogue, see [7]. The catalogue for this case study, which due of lack of space we can only show partly (see figure 2), has been created by studying service documentation as provided by our industry partner KPN (the largest Dutch TelCo operator), and by interviewing KPN representatives. Effectively, the catalogue is an instantiation of the e^3 service ontology. We have evaluated the catalogue afterwards with a domain expert from KPN, who is actively involved in realizing service bundles for 'VoIP', for descriptive validity. The catalogue itself is further explained as part of the following description of the substitution and preference ordering reasoning process.

3.2 Reasoning about substitution and preference ordering of e-services

We now illustrate how to derive telecom e-services from consumer needs, by considering an average 2.4 household consumer, who wants to communicate with family abroad but finds that using a traditional phone is too expensive, as a prototypical example.

Step 1: Select a consumer need, and derive an initial set of wants satisfying this need.

In the e-service catalogue, we first traverse from a need to (alternative) wants that can (partly) satisfy this need.

Case: We assume that the consumer starts at the need 'communicate directly' since this need-definition comes closest to the need 'communicating with family abroad'. By traversing from this need to the want-level, we thus find the initial,



Fig. 2. A partial e-service catalogue for the need 'communicating with family abroad'

alternative, set of wants [VoIP, instant messaging], because both can satisfy the need 'communicate directly'.

Step 2: Consider dependencies between want/consequence pairs, to elicit additional wants. The next step is to consider for each 'want' as found in step 1, any additional wants, by checking the reified 'adds-value' relationships between want/consequence pairs. If, for a considered want A (with its consequences), we can find related wants B, C, ... (with their consequences), we ask the consumer to decide upon inclusion of wants B, C, ..., by presenting him/her the consequence for wants B, C, ... As such, the consequences provide the consumer with a rationale to choose.

Case: In our catalogue (figure 2), there is a Core/Enhancing(C/E) reified 'adds-value' relationship between 'VoIP' and 'number portation'. Therefore, the consumer is presented the consequence 'keep current phone number' for the 'number portation' service. Due to all the hassle involved when changing a phone number, we assume that the consumer indicates that (s)he would want to keep her current number. Therefore the want 'number portation', which includes the consequence 'keep current phone number', is included as an additional want. So, the set of wants that is now used for our further reasoning is: [{VoIP, number portation}, instant messaging] (note that 'number portation' is only relevant in combination with 'VoIP', and not with 'instant messaging').

Step 3: Ask the consumer to assign a preference ordering to the consequences.

In the third step, we ask the consumer to prioritize the consequences for the benefits of wants found so far. We do so, because consumers are not really interested in a service itself, but in the *specific benefits* that a service possesses and in how these benefits help them to fulfill their needs. Yet, the consumer might not always be able to assign a preference ordering to a benefit because the benefit is *by definition* always stated in objective (and often technical) terms (recall that a benefit is e.g. the size of mailbox, the bandwidth of a data-connection, etc.). For this reason, we ask the consumer to assign a preference ordering to a *consequence*, which is then propagated to the underlying benefit. In other words; we present the consumers with the value-in-use or *goal* that can be achieved through a benefit, rather than the benefit itself.

The consumer should have a concrete way to express his/her preference. For this purpose, we use a four-point importance scale which we based on the MoSCoW-list. The latter is often used in Rapid Application Development (RAD) software engineering projects [5] to prioritize software requirements. Our scale consists of the following categories:

- Must have. A consumer can assign a 'must-have' priority to one or more consequences. For a service bundle to be relevant for the consumer, all 'musthave' consequences must be satisfied by the bundle.
- Should have. A 'should-have' consequence should be realized by a benefit from a supplier-specific service, but as opposed to a 'must-have' consequence', realization is not a necessity.

- Could have. A 'could-have' consequence is something that the consumer perceives as a nice-to-have feature. A 'could-have' consequence is perceived to be less important than a 'should-have' consequence.
- Does not matter. As implied by name, this last category can be used by the consumer to indicate that a consequence does not have to be taken into consideration in the bundling process.

Case: The consumer is presented with the consequences for the wants {VoIP, number portation}, and {instant messaging}. Examples include 'hear (natural) voice during conversation' and 'keep current phone number', each of which can be assigned an importance ranking. We assume that our consumer assigns the importance 'must-have' to 'hear (natural) voice during conversation' because it gives more character to the conversation. Considering 'should-have' preferences, our consumer assigns such a preference to 'avoid unwanted conversations' and 'keep current phone number'. Lastly, we assume that our consumer assigns a 'could-have' preference to the consequence 'make multiple calls simultaneously', because there are multiple members in the household.

Step 4: Present consumer with the set of underlying benefits.

Next, the consumer is presented with the set of benefits that belong to the consequences for which an importance ranking was done, and with an opportunity to change these benefits. Of course, these benefits are automatically assigned the same importance ranking as their consequences have. This step is in particular relevant if a consumer wants to consider the *objective* benefits of a service. This often the case for the more technically oriented consumers who want to specify exactly the benefits they expect from a service, such as the exact download speed they desire from an internet connection.

Case: Our consumer has selected the consequence 'make multiple calls simultaneously' and would therefore be presented with the benefits: '#phone numbers: $\langle = 2$ ' and '#phone numbers: $\rangle = 1$ '. As such, the consumer from our average 2.4 household has the option of specifying how many phone numbers are required. In our scenario, we assume that the consumer chooses '#phone numbers: $\rangle = 1$ ' since there are more than 2 members in the household who each, at some point, might require their own phone number.

Step 5: Generate alternative service bundles and assign preference ordering.

After all benefits have been prioritized and possibly changed by the consumer, the next step is to generate the actual e-service bundle that provides the benefits. To do so, a table is built with the prioritized benefits and the corresponding consequences on the one hand, and the e-services, taken from the catalogue, which can provide the requested benefits (see table 1) on the other hand. The 'consequences' are important in the process, since multiple benefits may have a same consequence. This allows for realizing a particular consequence in alternative ways, more specifically by providing alternative benefits. As a result, the service bundle composition problem becomes less constrained, thus increasing the chance of finding a valid solution. Alternative benefits for a same

10

		-					
Consequence	Benefit	VoIP	VoIP	VoIP	Instant	Number	Number
		(KPN)	(XS4all)	(skype)	messaging	portation	portation
					(Microsoft)	(KPN)	(XS4all)
Must have							
Voice during	communication:	X	X	Х			
conversation	oral						
Should have							
Keep current	number portation					Х	Х
phone number							
Avoid unwanted	number recognition	Х	Х				
conversations	caller-ID			Х			
	Microsoft-ID				Х		
Could have							
Multiple	#Phone numbers:		Х				
calls simult.	>= 1						

Table 1. A preference ordering as done by the average 2.4 household

consequence are in the catalogue (figure 2) shown by the 'OR' label between a consequence and its benefits.

So, the found table is used to generate the relevant e-service bundles. The bundles are evaluated against the benefits and their preference ordering, as assigned by the consumer, in the following order.

- 1. Service bundles that do not satisfy all 'must-have' benefits, are rejected.
- 2. For the remaining service bundles, the total number of satisfied 'should-have' benefits per bundle are used to rank the bundles. If a bundle A satisfies more 'should-have' benefits than bundle B, bundle A will be preferred over B, *independently* of the amount of 'could have' benefits. This corresponds to how MoSCoW is used in RAD software engineering projects, where first all 'should-have'requirements are implemented before, *and independently of*, the 'could have' requirements.
- 3. In case two or more service are ranked equally, the number of 'could-have' benefits per bundle are used for ranking.

Case: For our average 2.4 household, consider the generated table 1. Here, the 'must-have' benefits are clustered at the top rows, followed by the 'should-haves' and thereafter the 'could-haves'.

'Must-have' benefits. The 'must-have' benefits lead to the following set of *alternative* services [VoIP (KPN), VoIP(Skype), VoIP(Xs4all)], because all these services each include the benefit 'communication: oral'. Since all these services result in the same number benefits (namely one), we can not indicate a preference ordering yet.

'Should-have' benefits. We now consider the 'should-have' benefits. In our table, we have four of such benefits:

- (1) 'number portation', which inherited its 'should-have' preference from the consequence 'keep current phone number', and

- (2) 'number recognition', (3) 'caller ID', and (4) 'Microsoft ID'.

The last three benefits inherited their 'should-have' preferences from the 'avoid unwanted conversations' consequence; therefore these benefits are alternatives for the 'avoid unwanted conversations' consequence. In figure 2, this is represented by an OR-dependency between 'avoid unwanted conversations' (consequence) and the benefits realizing this consequence.

These benefits result in the services [Instant messaging (Microsoft), Number Portation (KPN), Number portation (XS4All))], to be added to the already found services as a result of the 'must-haves'. Note that the other 'should-have' benefits (Number recognition (KPN) (XS4All), Caller-ID (Skype)) are already included in the services found while considering the 'must-haves'.

'Could-have' benefits. We finally consider the 'could-have' benefit '#phone numbers: >= 1'. In our consumer specific catalogue, we find that the 'VoIP' service of XS4all is the only supplier that can provide this benefit. This service was already included as a candidate in the service bundle.

Alternative services. We now consider *alternative* services satisfying the same need. In other words: we must avoid awkward bundles, such as the bundle [VoIP (KPN), VoIP (XS4all)] because both services in this bundle act as an alternative in satisfying the same need: 'communicate directly'. However, we may find [VoIP (KPN)] and [VoIP (XS4all)] as *alternative* bundles.

Preference ordering. After checking for alternative services, we arrive at the following possible bundles, sorted according to their correspondence to the preference ordering of benefits.

- [VoIP (XS4all), number portation(XS4all)], [VoIP(XS4all), number portation(KPN)].
- [VoIP (skype), number portation(XS4all)], [VoIP(Skype), number portation(KPN)], [VoIP(KPN), number portation(XS4all)], [VoIP(KPN), number portation(KPN)].
- 3. [XS4all].
- 4. [VoIP(skype)], [VoIP(KPN)].

Preference ordering is done by comparing each possible bundle using the actual benefits desired by the consumer. So, for instance, the combination [VoIP (XS4all), number portation(XS4all)] is ranked higher than [VoIP(KPN), number portation(KPN)] because, supposing that the satisfaction of all other benefits is equal, [VoIP (XS4all), number portation(XS4all)] satisfies an extra 'should-have' benefit: '#Phone numbers: ≥ 1 '. Additionally, any 'VoIP' service combined with 'number portation' is ranked higher than an individual service. This is because the number of 'should-have' benefits satisfied by such a combination is higher than the number of 'should-have' benefits satisfied by any individual 'VoIP' service, since it includes the benefit 'number portation'. Also, no bundles are generated that contain an 'instant messaging' service. This is because 'instant messaging' does not contain the benefit 'communication:oral'. Moreover, an 'instant messaging' service will not be offered in combination with a 'VoIP' service,

because it acts as an alternative in satisfying the same need: 'communicating with family abroad'.

Finally, note that we only discuss *consumer-oriented* reasoning about service bundling in this paper. Therefore, bundles such as [VoIP(Skype), number portation(XS4all)] are taken into consideration, even though they are likely not possible from a *supplier-oriented* view. Such a bundle would be rejected by supplier-oriented bundling analysis. For a more elaborate discussion on the supplier-view on service bundling, we refer to [3].

Step 6: Present the consumer with the sacrifices of services and an opportunity to change the importance rankings.

Before the consumer actually chooses one of the possible bundles, (s)he considers the sacrifices of each alternative bundle. Based on the sacrifices, the consumer may change the preference orderings assigned to the benefits. We do so because the price of a service plays a significant role in a consumers' decision to actually acquire a service [9]. In our ontology, this issue is represented with the concept 'sacrifice' (To represent the pricing of a service itself, we use the pricing models of [8]).

If the consumer is not satisfied with the price that has to be paid to acquire a certain service bundle, (s)he has an opportunity of changing the preference ordering of consequences and therefore benefits. Changing the preference ordering of a benefit entails going back to step 3. If the consumer eventually finds that there is a balance between benefits received from a bundle and the price that has to be paid for it, (s)he selects the bundle for provisioning.

Case: Our consumer zooms in on both bundles on top of the preference ordering, and finds that acquiring the benefit 'number portation' costs $10 \in$ in both cases. Now, for the sake of argument, say our consumer finds this too expensive and wants to change the preference of the benefit 'number portation'. For this, (s)he is presented with an opportunity to go back to step 3, where all the values already filled in are still present (such as '#Phone numbers: $\geq = 1$ '.). Now the consumer changes the preference ordering on 'number portation' from 'should-have' to 'does-not-matter' and generates the alternative bundles again. A new set of service bundles is generated, this time with [XS4all] at the top of the list.

4 Discussion

Practical usefulness. The domain expert of KPN considers $e^3 service$ to be a useful tool for facilitating communication between marketeers and IT-personnel. This precedes our own goal, namely automated consumer-oriented e-service bundling. Marketeers, responsible for designing these bundles, do not always know whether an e-service bundle is *technically* feasible. Since $e^3 service$ relates benefits of e-services as experienced by consumers to a supplier-oriented catalogue of services (see also [1]), $e^3 service$ contributes to closing this gap.

The domain expert from KPN also pointed out that on-the-fly e-service bundling as envisioned, brings about problems that need to be considered before this idea can be *realistically* implemented. Below, we provide a selection of a few mentioned problems.

Planning of e-service provisioning. Some e-services take days of preparation before they can be provisioned to the consumer, often due to contractual and technical arrangements to be made. Therefore, the provisioning of a bundle has to be carefully planned. This calls for inclusion of (skeleton) planning techniques in the reasoning process (see e.g. [10]).

Single-point-of-contact. Often consumers want to have a single-point-of-contact in case there are problems with the provisioned service bundle (e.g. a helpdesk). A dynamic, and on-the-fly generated multi-supplier e-service bundle should have mechanisms to mitigate these single-point-of-contact services. One solution is to consider such services as e-services *themselves*, which therefore should be part of the e-service composition process. Moreover, the need for single-point-of-contact services (e.g. to repair service-failures) can be reduced by allowing for automated reconfiguration of provisioned service-bundles, e.g. facilitated by platforms for adaptable compositions of web-services [4].

e-Service pricing. In the telecommunication industry, discounts are a frequently used mechanism to attract consumers. With *single*-enterprise bundles, pricing these bundles and deciding on discounts is relatively straightforward. If however consumers create their own *multi*-supplier bundles, deciding on discounts is more difficult, due to supplier-specific pricing schemes and discount-policies.

5 Conclusions

In this paper, we showed how to reason about bundling of e-services, based upon consumer needs, in a structured and semi-automatic way. Additionally, we showed how to derive a preference ordering for the found e-service bundles, and how the pricing of a supplier-specific service - through an influence upon consumer preferences -can influence this preference ordering of bundles. We have also illustrated how $e^3 service$ works in practice, for a case study in the telecommunication industry. Currently, we are working on software support for the $e^3 service$ methodology.

For future research directions, we will integrate a supplier perspective on eservice bundling (specifically *e-serviguation*, see [1]), to generate e-services bundles that are not only valid from a consumer perspective, but also from a supplier perspective. Additionally, we will address the quality (or: non-functional) attributes of a service more in-depth. This is because quality also plays an important role in acquiring e-services, especially when considering B-to-B-environments where such quality aspects usually have to be strictly agreed upon by means of a SLA (see e.g. [11]).

Acknowledgements. We want to thank Leo Stout and Ron van der Kwaak from KPN for useful comments on the case presented in this paper. This research has been partly funded by NWO/STW/Jacquard as the project VITAL.

References

- 1. Hans Akkermans, Ziv Baida, and Jaap Gordijn. Value webs: Ontology-based bundling of real-world services. *IEEE Intelligent Systems*, 19(44):2332, July 2004.
- J. Arndt. How broad should the marketing concept be? Journal of Marketing, 42(1):101–103, January 1978.
- Z.S. Baida. Software-aided service bundling. PhD thesis, Free University Amsterdam, May 2006.
- Luciano Baresi, Elisabetta Di Nitto, Carlo Ghezzi, and Sam Guinea1. A framework for the deployment of adaptable web service compositions. Service Oriented Computing and Applications, volume=1, number=1, pages=75-91, year=2007.
- Beynon-Davies, Carne, Mackay, and Tudhope. Rapid application development (rad): an empirical review. *European Journal of Information Systems*, 8(3):211– 223, 1999.
- David Booth and Canyang Kevin Liu. Web services description language (wsdl) version 2.0, http://www.w3.org/TR/2007/PR-wsdl20-primer-20070523/, 2007.
- S. de Kinderen and J. Gordijn. e³ service an ontological approach for deriving multi-supplier it-service bundles from consumer needs. In Proceedings of the Forty-first Hawai'i International Conference on System Sciences (HICSS-41) (CD-ROM). Computer Society Press, jan 7-10 2007.
- B. de Miranda, Z. Baida, and J. Gordijn. Modeling pricing for configuring e-service bundles. In Proceedings of The 19th Bled eCommerce Conference, june 5-7 2006.
- 9. M. Fishbein. Belief, attitude, intention and behavior : an introduction to theory and research. Addison-Wesley, 1978. third print.
- P.E. Friedland and Y. Iwasaki. The concept and implementation of skeletal plans. Journal of Automated Reasoning, (1):161–208, 1985.
- 11. U. Greiner. Quality-Oriented execution and optimization of cooperative processes: Model and algorithm. PhD thesis, Univ. Leipzig, 2006.
- 12. C. Grönroos. Service Management and Marketing. Lexington Books, 1990.
- J. Gutman and T.J. Reynolds. Laddering theory-analysis and interpretation. Journal of Advertising Research, 28(1):11, Febraury/March 1988.
- 14. M.B. Holbrook. *Consumer value; a framework for analysis and research*. Routledge, 1999. 1st edition.
- 15. P. Kotler. Marketing Management. Prentice Hall, 2000.
- 16. C. Lovelock. Service Marketing People, Technology, Strategy. Prentice Hall, 2001. Fourth edition.
- 17. R. Normann. Service Management strategy and leadership in service business. Wiley, 2000. third edition.
- J. Ramsay. The real maening of value in trading relationships. International Journal of Operations and Production Management, 25(6):549–565, 2005.
- D. Tapscott, D. Ticoll, and A. Lowy. Digital Capital Harnessing the Power of Business Webs. Nicholas Brealy Publishing, 2000.