# A Consumer Needs-Driven Approach for Finding IT-Service Bundles

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

*Today, IT-support for enterprises is increasingly delivered as a commercial service. Consequently, IT-service provisioning requires at least two enterprises, a seller and a buyer. Moreover, the seller itself can often be considered as a network of enterprises, jointly satisfying an IT-need of a buyer, thereby utilizing each other strengths. In this paper, we contribute an ontologically well-founded methodology called* e$^3$service *for creating multi-supplier IT-service bundles that satisfy customer IT-needs. Such IT-needs, actually being high level problem statements that often require multiple IT-services to cover (hence the term bundle), are often too ill-defined to find specific IT-services for, thus requiring a systematic approach to finding these solutions. The main idea behind* e$^3$service *is to conceptualize established ideas from service marketing literature, with the ultimate aim to generate a bundle of IT-services semi-automatically starting from a vague customer need. This paper focuses itself on a systematic analysis of a customer need, or: how to gradually arrive from a vague problem statement to specific IT-services covering this problem statement. We illustrate this process by using findings from a case study performed with an industry partner.*

## 1. Introduction

In recent years, the notion of customizable bundles of IT-services to satisfy complex needs from specific customers, has gained interest. Consider a daily-life example of obtaining Internet access as provided by Internet Service Providers (ISPs). Here, the proposition of an ISP is actually a bundle consisting of more elementary service outcomes, such as IP-based access, an email box, space to host a website, and access to newsgroups. However, depending on customer needs, other bundles are possible, for instance only IP-based access plus email plus IP-telephony. In addition to creating various bundles for customization of IT-services to provide a closer match to customer needs, such IT-services are increasingly offered by a *networked value constellation*, rather than just a *single enterprise* [17]. By doing so, suppliers can utilize their core competencies, while still offering (jointly) a satisfying, complex, service to customers. In the ISP-example, Internet access can be a multi supplier bundle where IP-access is 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.

Guidelines on creating customized service bundles have already been studied in business literature, most notably by [11],[14],[15] . However, these guidelines are fairly generic (the focus is on just services and not specifically on IT-services). Moreover, they lack a conceptualization and formalization so it is difficult to systematically and (semi-) automatically reason about service bundles. Such reasoning is important, because IT-services, as mentioned 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 by supported by information technology as much as possible. To put it differently, on the long term we are aiming to contribute computer-aided composing and provisioning of commercial IT-service bundles satisfying a complex need in a network of enterprises.

It is important at this point to understand that we consider IT-services really as *commercial* services: economic activities, deeds and performances of a mostly intangible nature [15], 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 BPSS [1], BPEL4WS [2], WSCI [3], and WS-Coordination [5], to name only a few: these are mainly intended to arrive at a cross-organizational computing platform to facilitate interoperability on a more *technical* level. Obviously, web services potentially can serve as an implementation platform

for the commercial IT-services we have in mind. In previous work (see Section 2), we have proposed (1) an ontology to represent a supplier perspective on service outcomes (basically a service-catalogue of suppliers, including constraints that can be used to reason about potential service-bundles), (2) a basic ontology to represent needs, and (3) a reasoner that bridges needs and services using a feature solution graph ([8], mapping needs on alternative service outcomes).

The contribution of this paper is then two-fold. First, it proposes a customer-oriented lightweight ontology to represent and reason about customer needs. Second, we show how this ontology can be used to (1) elicit needs of consumers, and to (2) find service outcomes, as represented by using our supply-side ontology on services. The customer-need ontology, as well as its use for need-elicitation and service-bundling reasoning is explained in Section 3. In Section 4, we apply the ontologies to a case study we performed in cooperation with MalieNet; a real-life networked value constellation. Finally, in Sections 5 and 6, we present a discussion and our conclusions.

## 2 Previous work: Serviguration

This paper builds upon previous research we did on *serviguration* [4]. Serviguration (service configuration) provides computer supported reasoning about general service bundles, by combining concepts from business literature (e.g. various kinds of dependencies between services, which can be seen as constraints for the bundling reasoner) with concepts from computer science (e.g. RDF ontologies, and production rules on top of these, as used in the field of knowledge engineering). Case studies in the realm of electricity supply and healthcare we have performed, have shown that by using this methodology, meaningful bundles of services can generated semi-automatically [4] . 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 the 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. For instance, a service such as email, occurring in a catalogue of an ISP, would be related as enhancing service to the service IP-access. From a supply-side perspective, this relation can be motivated by the technical consideration that modems required for IP-access provisioning, and servers required for email account hosting, are for example co-located at the same site, and therefore (as a bundle) cost less, than if these services were located at different physical locations (due to the fact that broadband connections between two sites can be avoided). As a consequence, to satisfy a need to surf on the web the bundling reasoner would find a bundle [IP-access], and a bundle [IP-access, email]. It is important to understand that the [IP-access, email] bundle is found, due to technical, supplier-oriented, reasons (co-location as a result of which the bundle can be offered cheaper), rather than that is directly based on a customer requirement who may need in addition to IP-access also email. Ideally we would like to generate only those bundles that contain precisely those features the customer is interested in, and not bundles that contain unnecessary features that have to be paid for non-the-less. To this end, serviguration employs a rudimentary customer need ontology, which uses another terminology then is done for the supply side of services. The idea that customers may have a different perspective on services than suppliers do is amongst others also stated by [18]. In order to improve the generation of bundles by avoiding unnecessary extra services, we detail in this paper (Section 3) a customer-oriented need/want/demand ontology, that is capable of representing relationships (and constraints) between needs/wants/demands that are grounded in *consumer needs*, and not in *supplier-oriented* motivations.

A second and related issue is that serviguration makes a clear, sharp-cut, distinction between a supplier perspective, and a customer perspective. Using a feature-solution graph [8], needs of customers (features) are then mapped onto service-outcomes of suppliers (solutions). As a result, serviguration defines the customer perspective independently of the supply side perspective. However, to improve serviguration, we propose to make a more gradual movement from needs to service-outcomes. We consider it as hard to define customer needs independently from what kind of IT-services are offered by suppliers. To put it differently, it is difficult to state problems (needs), without having some form of solution (service-outcomes) already in mind. In general, this observation has been made already quite a while ago in Computer Science [7], and in the design literature by [6]. Also, the observation is backed up by the means-end chaining theory [12], where laddering is employed to abstract away from specific product attributes that are typically defined by the supplier (e.g. a product brand like Levis) to the consequences derived from those attributes that are typically defined on the customer side (e.g. Levis provide for more social prestige). This position also closely relates to discussions in Requirements Engineering. In [19], with respect to requirements, a distinction is made between problem descriptions (e.g. goals, problem frames, value models) and solution specifications (e.g. cf. IEEE 830). A key question is then how to move from a prob-

lem description to a solution specification. In this paper, we explore a gradual move, during which more and more solution aspects (services) are included to arrive at a satisfying solution for an initial problem statement (customer need). Gradually moving from problem to solution has the added advantage of being able to find substitutes for service outcomes more easily, since considering solutions at different levels also allows for exploration of substitutes at different levels; in a sense, this is a useful to suppliers who are reviewing their competition. Later in this paper, we explain more precisely how we see this.

Finally, serviguration concentrates on commercial services as a rather generic construct. The aim of this work is to focus on IT-services, to ultimately generate and build networked IT-value constellations.

In sum, to address these fore mentioned issues, our *e*$^3$*service* methodology, building upon serviguration, has two main purposes: 1. it intends to elaborate on the customer perspective on services, and to take the interplay between the customer and suppply side more into account. 2. it specializes itself towards reasoning about IT-service bundling, more specifically those that can be delivered online. Focusing on this domain also allows for standardizing certain service attributes, such as a standard QoS-measurements as quality attributes for IT-services. This is however beyond the scope of this paper.

# 3 *e*$^3$*service* : Needs-driven IT-service bundling

In this section, an explanation will be provided of *e*$^3$*service* . First, we will discuss how *e*$^3$*service* employs value modeling to elicit the service outcomes of the service outcomes on which a bundling analysis will be performed. After this, we will show how *e*$^3$*service* analyzes customer needs, taking the existing service outcomes as a starting point.

Although it is counterintuitive to elicit customer needs in this way, there are reasons to do so. The most important reason is that we often operate in an already up-and-running business context (in practice, uncharted territories seldom occur), with already available service outcomes available. To our experience, a key issue is then to understand which benefits a customer can derive from certain service outcomes. These benefits would be more difficult to find when reasoning in a top-down fashion - i.e. when starting from customer needs (meaning; the problems that are to be solved by employing the services) - since specific service outcomes and the ends to which these can be employed are then left out of the elicitation process.

## 3.1 Step 1: Employing value modeling to elicit service outcomes

The first step in *e*$^3$*service* is to elicit the services on which the bundling analysis will be performed, where we consider the services to be economically valuable services produced and consumed in a network of enterprises. For this purpose, we employ a value perspective. This is necessary because we need to be familiar with the available services and the benefits thereof first if we want to reason about the specific customer needs they satisfy. Also, a conceptualization of the service outcome allows us to focus on analyzing needs that are important to scrutinize further. This is because familiarity with the main benefits of the service - as gained through value modeling - enables us to ask the question of why these benefits are important to the customer, thus providing a good starting point for arriving at customer needs. In marketing literature, a similar approach is towards needs elicitation is the laddering-technique [12] mentioned earlier.

In this paper, we employ *e*$^3$*value* [10] to elicit the benefits provided by the services, followed by a bundling analysis. The *e*$^3$*value* methodology helps in understanding a network of enterprises in terms of the enterprises themselves, the value adding activities they perform, and the valuable objects they transfer with each other. Here, the valuable objects should be seen as IT-service outcomes. In addition, *e*$^3$*value* allows to reason about economic reciprocity and about the Discounted Net Present Value Flow (DNPVF) on a per actor basis, to evaluate economic feasibility for each actor participating in a value constellation.

## 3.2 Step 2: Elicit and further analyze customer needs

The next step in the *e*$^3$*service* methodology is to elicit and analyze customer needs, using the found service outcomes as an input. As in *serviguration*, *e*$^3$*service* employs a need/want/demand hierarchy [13] as a basis for specifying customer needs into demands that are specific enough for matching against service outcomes, but adds to these need/want/demand concepts typed relationships, to facility a gradual move from need (problem) to service-outcome (solution). In the next section we will explain the concepts used in our ontology, with a focus on the need/want/demand hierarchy. Next we show how to stepwise arrive at a customer need specification, using the service outcome as a starting point.

### 3.2.1 The *e*$^3$*service* customer need ontology

The basic thought behind using a need/want/demand point of view, is that it enables the detailing of a *need* - being

a problem of the customer that exists independently of the service - into *demands* using the concept of a customer *want* as an intermediary level. The difference between the want and demand is that a want is typically a *generic satisfier* for a need, and most importantly: *that a want does not contain the notion of the customer being able and willing to pay for it*.

As an example, consider a person in London, who 'wants to talk to a person in Paris' (a need) and that (s)he has chosen to travel to Paris to satisfy this need. Then 'transportation' is a want, since it is a generic satisfier for being able to talk to one another and moreover, it does not contain the notion of whether the customer is willing and able to pay for it; other wants are of course also possible, such as using some form of 'electronic communication'. Demands are in this example the modes of transportation, e.g. by 'boat', 'train' or 'airplane'; each of them containing a fairly good idea on how much the customer has to pay for getting from London to Paris.

The notion of using need/want/demand for specialization of customer needs is depicted in Figure 1. Through the relationship between sacrifice and demand depicted in the model, it is again stressed that the notion of the customer being willing and able to pay for a service outcome is important, since a favorable attitude towards acquiring a service does not automatically imply actual acquisition of that service (e.g. the person mentioned earlier might prefer to fly business class, but also might not be able to afford it). In marketing theory this is further backed up by [9], who states in the Theory of Reasoned Action (TRA) that it is crucial to analyze the customers' willingness to actually buy a service, and not to merely focus on measuring how favorable the attitude towards a service is.

Using different levels of abstractions (need/want/demand) while reasoning about customer needs has the advantage of identifying substitutes for a certain service more easily. For instance, recall that 'transportation' in our running example is only *a* way in which the need could be satisfied; an alternative could be some form of 'electronic communication'. This shows that a gradual approach toward needs-elicitation allows for a broader view of solutions that can be used to substitute a certain service in a way that is similar to Porters' Five Forces model [16]. In this model, one of the considerations is to also review substitutes for a product, and not to merely review direct competition that offers a product with similar features. This is similar to the difference between a want and demand, where direct competition can satisfy the same demand and on a higher level of abstraction, different substitute satisfiers compete for satisfying the same need.

From the model, it can also be observed that a typed relationship between demands can exist. This is derived from the fact that some types of service dependencies, such as the enhancing service of a spam filter to complement a basic e-mail service, typically have a customer component and therefore needs to be reviewed from a customer perspective as well. Such service dependencies are derived from marketing literature [11][14][15], and define in what ways two services can be related to one another. The example of the spam filter for instance, is typically a *value-enhancing* service that can optionally be delivered next to a basic e-mail service. However, there are also other types of service dependencies that can exist such as a *core/supporting* relationship between two services, where the core service - to use our running example: e-mail - cannot be delivered without some type of supporting services, such as administration.

So far, we have found two service dependencies that have a customer component . These are:

- *A core/enhancing relationship* - fulfilling the enhancing demand positively contributes to satisfying the need belonging to the core demand. Our perception of this dependency differs somewhat from the original interpretation found in service literature (and also in *serviguration*). Originally, this relationship is defined between two *fixed* service outcomes, provisioned by *fixed* suppliers. However, as we try to reason about IT-service bundling in a *networked* business setting, we try to abstract away from *fixed* suppliers and therefore define the core/enhancing dependency between demands (without having in mind a specific supplier already). Moreover, placing this dependency on the customer side allows us to express this dependency in customer terminology, thus allowing the customer to make the decision of whether or not a certain enhancing service should be included in a service bundle. An example of this relationship would be the e-mail - spam filter mentioned earlier.

- *Optional bundling* - which are more loosely related demands, the most important difference with the core/enhancing service being that these demands do not share the same need as a root. This also means that when fulfilled, optionally bundled services do not directly add value to the core service. An example of optional bundling could in the case of e-mail be streaming video, which is something very different from the core service but could be something that adds value to the customer as well.

The usefulness of analyzing these dependencies from a customer perspective is that it allows us to define more customized service bundles: this is due to the fact that modeling these dependencies on the customer side allows for asking customers directly whether they are interested in extra services based on the wants and demands fulfilled by this extra service, rather than first deriving bundles of service outcomes. For instance, say that the want beloning to the

demand of a spam-filter is *decrease unwanted mail*, then the question can be asked to the customer: would you also like a spam-filter, in order to decrease the number of unwanted e-mails? When answered positively, this demand can be taken into account in the bundling process; otherwise, it is left out.

We also added the concept of quality attributes to demands (not shown in Figure 1). These aim at describing the non-functional properties of a service, such as some reliability-metric, and play a key-role in customer satisfaction. Quality attributes have a place on the customer perspective since in marketing literature, service quality usually is perceived to be the difference between what the customer *expects* from a certain service (derived from e.g. past experiences or promises made) and the *actual* service which *can really be* delivered [20] For example, imagine that an ISP-customer expects a uptime -which can be presumed as a metric of reliability - of 99% of the time spent online, while the ISP might only be able to guarantee 95% uptime. This would result in a negative gap between expected quality and that which can be delivered, which provides us with an indication that the customer will probably be unsatisfied with this service if (s)he were to leave the situation unchanged (where changes include going to another provider, or trying to lower customer expectations concerning the level of quality). Due to lack of space however, we will not address this notion of quality further.

#### 3.2.2 Finding needs using service outcomes as a starting point

So far, we reviewed the concepts from $e^3service$ , now *how* do we use these concepts in practice to elicit and further analyze customer needs. To this end, we take the following steps:

1. *Derive demands from service outcomes.* Transforming service outcomes into customer demands means that we should abstract away from attributes specific to a certain supplier and only focus on the main benefit provided by the service in question. In our transportation example, a specific carrier like Easyjet can abstract away from his/her 'transportation' service, with specific properties such as no 'on-flight meals', to a more generic demand: 'flight' Note that it is important to abstract away from the *individual* organizations providing the service outcomes, unless certain service outcomes have such a unique character they are only provisioned by just one supplier. This notion of abstraction is important since in principle a demand can be satisfied by multiple suppliers.

2. *Cluster demands into wants.* This can be done by reviewing which demands must be satisfied together, and by asking the question what instrumental goal is fulfilled by the satisfaction of these demands. We perform this type of intermediary clustering - intermediary meaning; without directly moving to a problem description - in order to find a labeled, common denominator for related demands, so that it is clear *why* certain demands must be satisfied together. How this clustering works, is illustrated in section 4.

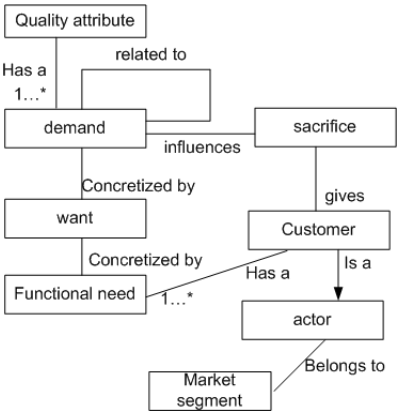


**Figure 1. needs ontology**

3. *Derive needs from wants.* In order to elicit a customer need, we try to find the terminal goal that is (partially) achieved by fulfilling the instrumental goal of a want. In other words; when eliciting needs, one tries to find a problem that is (partially) solved by fulfilling a want.

4. *Define dependencies between demands.* Having clarified the needs, it is now possible to define relationships between demands using the dependencies discussed earlier. Finding them can be done by using mentioned dependencies as guidelines; for instance, when one finds that 1) fulfilling one demand contributes positively to satisfying the need of another demand and that 2) this demand does not necessarily have to be fulfilled to satisfy that need, then a core/enhancing relationship exists. Reviewing if an optionally bundled relationship exists can be done by asking a slightly different question; given a certain demand X, would the customer be interested in satisfying another demand Y as well, *independently* of whether satisfying that demand contributes to satisfying the need corresponding to X?

Using this approach, we are able to construct a gradual transition from problem statement to solution, instead of defining a clear-cut distinction between them. Next, we show how this transition works in practice.

## 4 A real-life case study with *e*$^3$*service*

In this section, we will illustrate the principles discussed using a case study carried out with MalieNet.

### 4.1 The MalieNet case study

#### 4.1.1 MalieNet; an IT-service provider for industry-platforms

Malienet is an (IT-) service provider aimed at the Branch Organizations (BO's) in the Netherlands. These BO's are organizations that represent companies working within the same industry, or branch (e.g. the branch organization for all companies that sell cars). The companies represented by the BO's are the members of a BO, and these members pay contribution in return for e.g. extensive lobbying (the results of which could be beneficial to a whole industry sector). The BO's themselves are also a member of one overarching organization; VNO/NCW (the organization of the Dutch employers). An overview of the relationships between the mentioned parties can be seen in figure 2.

Malienet has been initiated because the BO's were loosing members (thus generating less revenue due to the contribution paid by these members), the underlying cause being that the BO's members paid contribution but were not seeing much that is of added value for them in return. The BO's, are for the large part rather traditionally thinking organizations that - besides lobbying and providing industry-specific consultation regarding rules and regulations - do not provide that much that is of added value for their members. When also taking into account the trend of member organizations organizing matters themselves (e.g. checking rules and regulations online), there is an indication that something should change for the BO's to prevent them from getting into trouble by loosing members and thus the contribution paid by them.

The aim of Malienet is to transform BO's into innovative organizations, by means of IT-service provisioning and consultancy regarding its business-wise deployment. The idea is that in that way, BO's also offer IT-enabled services to their members, thereby giving them more of their money's worth (e.g. by providing members with online community services that would enable them to communicate with one another online).

For this purpose, MalieNet has an extensive catalogue of IT-services that can be provided to the BO, ranging from stand-alone content management services to an integrated ERP-package covering all aspects of an organization. However, MalieNet does not actually realize any of the IT-services it provides; this is done by specialized partners. MalieNets role is in bringing together the relevant parties that are necessary to realize certain innovative service outcomes. In this respect, MalieNet is a typical networked organization; it consists of a number of companies that each focuses on his/her core competency, where the core competency of MalieNet lies in knowing who to bring together. A concrete example of this is that MalieNet brought a number of BO's and a CRM-provider together, so that a CRM-package could be developed that was tailored to the requirements of a BO. In this case study, we will focus on one of MalieNets' IT-services; providing the ability of creating mailings online.

#### 4.1.2 The Direct Mail service outcome from MalieNet

Our cooperation with MalieNet started with an initial interview session with MalieNet's sales representative and its consultants. After this session, we constructed an *e*$^3$*value* model for the MalieNet constellation, which revealed amongst others a Direct Mail service. The goal of MaileNets Direct Mail (DM) service is to provide the members of a BO with the possibility of setting up a DM initiative through the website of their BO. The service is offered in cooperation with a logistic service provider, here the biggest postal company in The Netherlands: TNT. The idea behind this service is that a member of a BO could go to the BO's website, design its own mailing using a wizard, and having the designed cards delivered physically a few days later. This way, the BO's members do not have to go through the hassle that is usually associated with DM themselves (e.g. employees having to prepare mailings themselves, thus wasting valuable time). However, this service turns out to be less popular than expected, one of the main reasons being the simplicity of the service. Currently, the DM-service only allows the customer to order customized A5-sized cards (meaning; that the customer is able to define the front and back of the cards and nothing more).

After a second session with MalieNet's sales representative and a TNT representative, it was felt that more value can be added to this service in order to make it more attractive to the BO, since the organization that implements this IT-service - TNT - also offers related IT-services (e.g. an IT-service that allows the customer to design customized stamps online). These could, when offered together with the basic card-service, make the DM-service more attractive. The current problem however is that the IT-services offered by TNT are presented in a fragmented manner (e.g. the customized stamp is offered separately from the DM-service) the implication being that the BO's might not even be aware of these value-added services. The question from MalieNet therefore becomes: how can we make the DM-service outcome more attractive by bundling and thus better employing existing IT-services? Ideally, we envision that a member of a BO could go their BO's website and, on the basis of filling in a short multiple-choice questionnaire receives a bundle of IT-services that best fits its DM-needs

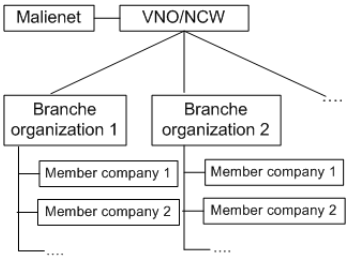


**Figure 2. MalieNet; providing IT-services for industry platforms who in turn are enabled to add more value to their members**

instead of the basic card design service.

### 4.2 Step 1: Using value modeling to elicit the TNT Direct Mail service outcome

As stated, we review the services on which a bundling analysis will be performed first in order to take them into account when reasoning about needs in the next step. The results of this analysis can be seen in figure 3, which shows the individual services of TNT as they exist currently, as well as the way in which they are provisioned to the members of the BO. We also chose to model some value activities in dashed lines. This is to indicate that the provisioning of the online DM-service, even though they might appear as being provisioned by the BO itself, is actually realized by TNT; the only value added by the BO is that they provide for a central website on which this functionality is offered (which could act as a portal for their members) and that they provide for some branch-specific card designs, but the service in itself and the basic benefits provided by it (a set of customized DM-cards) stay the same. Because of this - the IT-service provided by the BO being an "'instantiation"' of an IT-service provided by TNT- the idea is to model these IT-services differently, with the aim of not including them in the bundling analysis since this same instantiation can of course also be performed after a bundle has been generated.

### 4.3 Step 2: Elicit and further analyze customer needs from DM services

As explained, we first derive customer demands from the available service outcomes (the value objects as elicited by performing step 1). A comprehensive picture of all steps (and thus how we gradually move from need to service outcome) can be found in figure 4. The translation of service outcomes into customer demands can be thought of as a process of instantiation, where a modeled service outcome is a class and a demand an instantiation of that class. In our

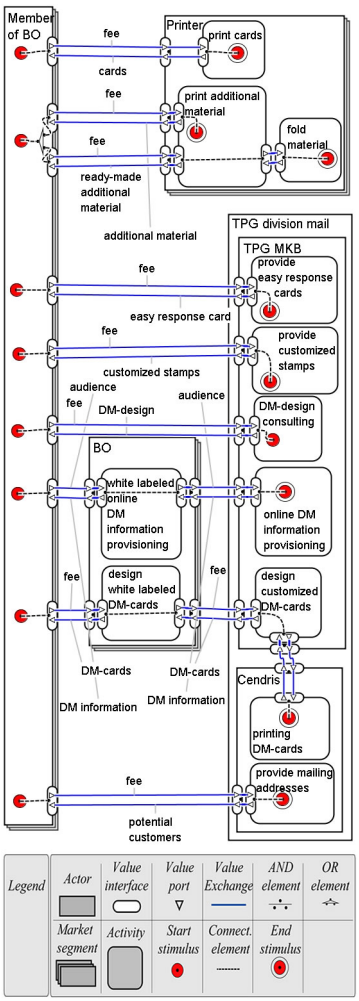


**Figure 3. MalieNet; Individual IT-services**

example, we can imagine that the service outcome 'customized DM-card' can be instantiated as a card containing introductory information but we can also imagine instantiating this service outcome as an invitation to an event. It is worth to note that we do not model a separate instantiation for every service outcome. In the case study, we illustrate this by letting both the DM-card and easy response card point toward the same demand; this is because both service outcomes provide the same benefit and as such can act as a substitute for one another. It has to be pointed out, however, that both services can have differences regarding specific properties; in this case, the specific property that distinguishes the *easy response card* from the ordinary *customized DM card* is the response medium, where the *easy*

*response card* allows the customer to react directly by sending back the card itself, while the ordinary *DM card* does not provide that option.

After having defined the demands based upon the specific service outcomes, we cluster the demands into higher level wants. In our case study, this is illustrated by the clustering of the demand *'prospect information'* with *'send introductory information using direct mail'* in the want *'create awareness through sending information'*. The usefulness of clustering demands into wants is that it shows which demands must be satisfied together. In this case, sending information to prospects cannot be performed without having some basic information on these prospects (e.g. addresses), which is a dependency that exists independently of how this prospect information is going to be delivered.

In order to complete the need/want/demand hierarchy, wants are finally clustered into needs by reviewing what type of problem a want (partially) satisfies. In this case, the want *'create awareness through sending information'* is used to satisfy the basic problem of *'attracting prospects'*. The need of attracting prospects can also be be satisfied through other means. For instance, the need for attracting prospects can also be fulfilled by investing more in existing customers, with the assumption that positive word-of-mouth helps when acquiring new customers. This backs up the statement that by using different levels of abstraction when reasoning about needs, it is also easier to find substitutes.

Finally, we add dependencies between the demands. In the graph, we can see the two types of dependencies we identified so far;

- The core/enhancing relationship which in this case relates the core service to services that further personalize the mailing. When satisfied, this demand allows the customers' mailing to stand out, which in the end positively contributes to satisfying the need belonging to the core demand

- The optional bundling relationship, which points to a demand that has a separate need as a root. Note that it was worthwhile to model this dependency in this case study, since many of the members of the BO are small companies that typically have little experience in using direct mail. As such, they could use some type of support, especially since it is expected that the real difficulty with direct mail does not reside in realizing the mailing using some online wizard, but rather in thinking up such points as: what do I want to accomplish with my mail? and derived from that: what should the mailing look like and what message should I get across? When a member of a BO is left in the dark with respect to answering these questions - as is currently the case - it is also expected that it is less likely that (s)he will acquire the service. Rather, when the question concerning support is directly asked to the member of the BO in some type of wizard then at least (s)he knows that it is possible to receive aid in setting up a DM initiative.

## 5 Lessons Learned

During the development of $e^3$*service* and its application in practice we learned some important lessons, some of which served as a starting point for improving our methodology. In this section we will discuss some of these lessons, together with a few observations that did not lead to adjustment in $e^3$*service* , but are important for further research.

**Lesson 1: Service dependencies that posses a customer component should be reviewed from such a perspective as well.** The methodology that we used as a basis for $e^3$*service* , serviguration, reviewed service-dependencies from a supply side perspective only. During the execution of this case study however, we found that it is useful to review these dependencies between customer demands also, since it allowed us to take a more customer-oriented approach toward service bundling. For instance, it allows us to include only those value-enhancing services in a bundle that are of interest to the customer. This is because the customer is allowed to make a choice on the demand him/herself, by trading off the costs inherent do the demand (recall that a demand is typified by an idea of how much it costs) with the way in which satisfying the demand contributes to satisfying the need belonging to the core demand. Beforehand, this was not possible since serviguration generated all possible service bundles and as such also came up with bundles containing extra services in which the customer might not be not interested.

**Lesson 2: It is useful to take the existing service outcomes as a starting point.** When reasoning about customer needs, it was found to be useful to take the existing service outcomes as a starting point. This is because it allows for brainstorming about the ways in which the customer can benefit from the service outcomes provided, such as the fact that an easy response card can be used not only for sending introductory information, but also for sending around an invitation to an event to existing customers. It is more difficult to find such specific applications when we reason in a top-down fashion (i.e. starting from a need) since in that case, the specific service outcomes and also the possibilities provided by them are not taken into account.

**Lesson 3: Trying to reason more formally about customer needs is possible, albeit to a limited extent.** Finding specific, systematic guidelines toward eliciting IT-needs

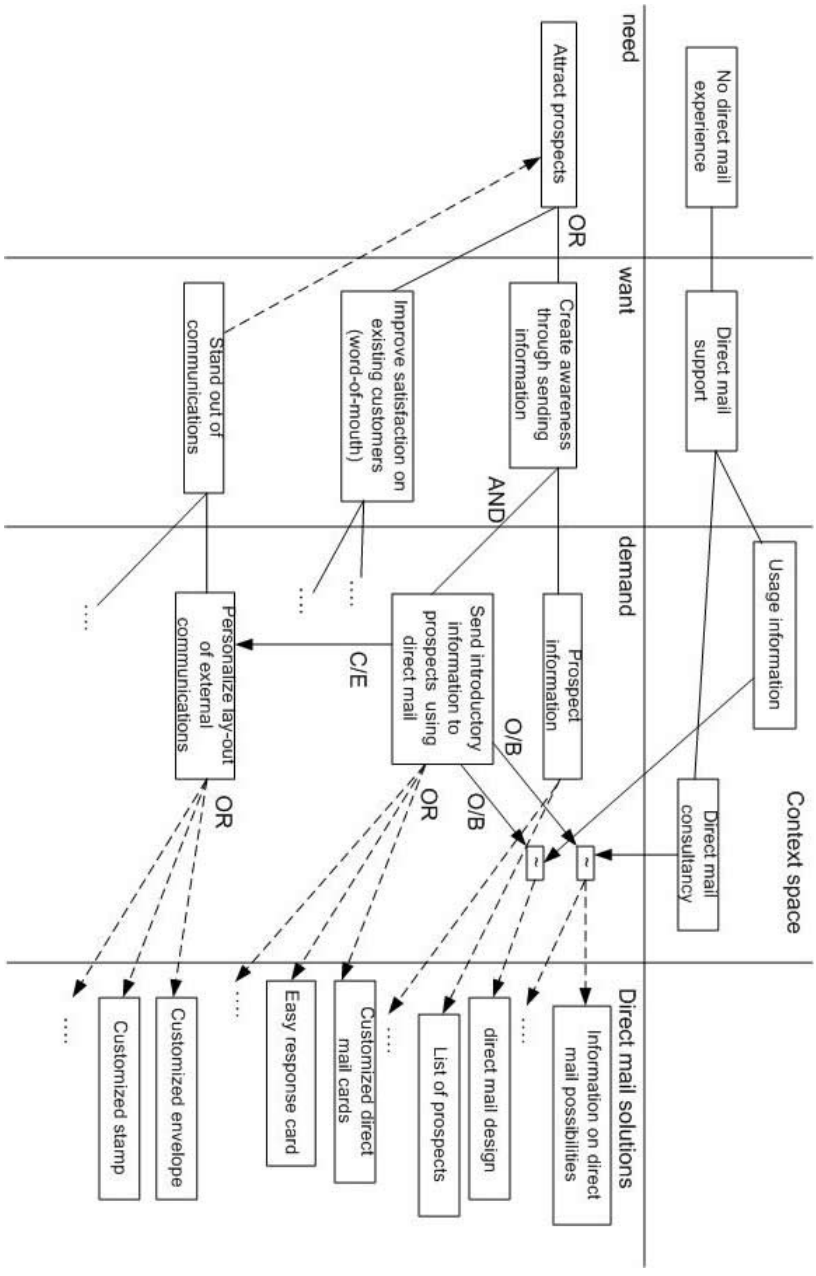


**Figure 4. A gradual transition from need (problem) to service-outcome (solution)**

has proven to be difficult. This is illustrated by our attempt of finding guidelines for grouping demands into wants; we were able to define only few and concluded that there is no such thing as a fixed recipe to finding a want, besides thinking of it as a generic category that can be used to cluster demands together on the one hand, and being an instrumental goal toward satisfying a terminal goal on the other.

## 6 Conclusions and further research

This paper presented a first step towards creating a methodology that can be used to create bundles of IT-services tailored to a customer need. We showed that a gradual transition exists between problems as specified by the customer and IT-services that satisfy these problems, and that taking existing service outcomes as a starting point when eliciting customer these problems -as a variation on the laddering-methodology - can be a useful approach.

Further research includes building standard IT-service taxonomies using existing commercially oriented service libraries such as ITIL as a starting point, with the aim of 1) standardizing reasoning on quality attributes, and 2) to be able to find demands more quickly and less arbitrary, based upon the descriptions of generic service classes rather than specific services delivered by a single provider.

Also, we would like to investigate the impact a business model has on finding services that match needs. In the current case study, there is a central organization -MalieNet - that has a catalogue of services, but we wonder what would happen if the customer itself searches for the IT-services necessary to cover its need. Because there is no such thing as a centralized catalogue in that case, the question arises of how to find IT-services that cover a need sufficiently in a more decentralized business environment.

**Acknowledgements.** We are grateful for the input of Pieter van der Ploeg from MalieNet, and Hans Burg and Frank Verlaan from Boer&Croon. They provided us with useful discussions concerning the case study and - from their practical background- gave indirect indications that a strict separation between customer and supply side can not exist. This research has been partly funded by NWO/STW/Jacquard as the project VITAL.

## References

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

[2] T. Andrews, F. Curbera, H. Dholakia, J. K. Y. Goland, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawanara. Business process execution language for web services version 1.1, http://www.ebxml.org/specs/ebBPSS.pdf. Technical report, 2003.

[3] A. Arkin, S. Askary, S. Fordin, W. Jekeli, K. Kawaguchi, D. Orchard, S. Pogliani, K. Riener, S. Struble, P. Takasci-Nagy, I. Trickovic, and S. Zimek. Web service choreography interface (WSCI) 1.0, http://www.w3.org/TR/wsci/. Technical report, 2002.

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

[5] L. Cabrera, G. Copeland, W. Cox, M. Feingold, T. Freund, J. Johnson, C. Kaler, J. Klein, D. Langworthy, A. Nadalin, D. Orchard, I. Robinson, J. Shewchuk, and T. Storey. Web services coordination (WS-Coordination), ftp://www6.software.ibm.com/software/developer/library/ws-coordination.pdf. Technical report, 2003.

[6] 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.

[7] O.-J. Dahl, E. Dijkstra, and C. Hoare. *Structured Programming*, volume 8 of *A.P.I.C. Studies in Data Processing*. Academic Press, 1972.

[8] H. de Bruin and H. van Vliet. Quality-driven software architecture composition. *Journal of Systems and Software*, 269-284(3), 2003.

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

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

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

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

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

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

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

[16] M. Porter. *competitive strategy - techniques for analyzing industries and competitors*. Free Press, 1998. 1st Free Press edition.

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

[18] M. VasarHelyi and M. Greenstein. Underlying principles of the electronization of business; a research agenda. *International Journal of Accounting Information Systems*, 4(1):1–25, December 2003.

[19] R. Wieringa. Requirements researchers: Are we really doing research? *Requirements Engineering*, 10:304–306, 2005.

[20] V. Zeithaml and M. Bitner. *Services Marketing*. McGraw-Hill, 1996.