22nd Bled eConference
eEnablement:
Facilitating an Open, Effective and Representative eSociety
June 14 - 17, 2009; Bled, Slovenia

A computational approach towards eliciting needs-driven bundles of healthcare services

Sybren de Kinderen[1], Jaap Gordijn[1], Rose-Marie Dröes[2], Franka Meiland[2]

[1] Department of Computer Science
VU University Amsterdam, the Netherlands
{sdkinde@cs.vu.nl, gordijn@few.vu.nl}

[2] Department of Psychiatry, EMGO institute for health and care research,
VU University medical center Amsterdam, the Netherlands
{rm.droes@vumc.nl, fj.meiland@vumc.nl}

Abstract

We propose a method, e^3-service, to reason about satisfying customer needs (problems stated by the consumer) by means of a bundle of multi-supplier service bundles (solutions from specific suppliers). The e^3-service method represents customer needs, desired consequences by the customer, and the services that realize those consequences in a multi-perspective service catalogue. This catalogue is used by a reasoner, which elicits customer needs and desired consequences, and matches these consequences with services offered by suppliers. The e^3-service method has been implemented in software to demonstrate its feasibility. In this paper, we show how e^3-service reasons about finding services for a consumer need using a case from the healthcare industry.

Keywords: commercial healthcare services, service bundling, computational reasoning, consumer needs

1 Introduction

With the liberalization of the Dutch healthcare industry, there is an increasing need for a mechanism that can translate consumer needs into (multi-party) bundles of healthcare services satisfying these needs. There are two main reasons for this. First, many private and (semi-)public organizations now offer variations of similar healthcare-services, with the result that healthcare practitioners - let alone end-consumers themselves - really have difficulties seeing the woods for the trees when trying to find bundles of services that satisfy care needs. Second, customers often have trouble expressing what they need in terms that are specific enough to find service bundles for.

As a response to the above, Dröes et al.[1] have developed a Digital Interactive Social Chart for informal carers of persons with DEMentia, called DEM-DISC. The key idea behind this social chart is that informal carers can express their care-needs (or the care-needs of the dementia-patient) through a question tree on a website and this website consequently responds with a list of service bundles that fits with the stated care needs. For example, an informal carer could state through the question tree that s/he is interested in social contacts for a person with dementia and state the postal code of that person, based upon which a ` meeting group for people with dementia' service would be found that is in the vicinity of where the person with dementia lives.

Currently, the social chart is able to successfully configure bundles of healthcare services based upon stated consumer needs. However, since there is a *configuration mechanism* underlying the social chart, finding service bundles currently is one-way traffic and as such, does not allow the consumer to adjust preferences based upon resulting service constraints such as fees to be paid, waiting lists and more. Allowing for such an adjustment is deemed important however because, as amongst others Woodruff [10] notes, consumer value always is a result of a trade-off between the benefits received from a service and the sacrifices made to acquire a service. Based upon supplier feedback, an informal carer therefore might be forced to adjust preferences or even to discard bundles of services altogether – even when they satisfy the needs initially stated.

In this paper we introduce a methodology, e^3-service, that uses *consumer-supplier interaction* to reason about finding bundles of services. e^3-service allows a consumer to attach importance scores to needs stated on a website, and then iterates a few times to gradually adjust solutions (service bundles) to customer needs (problems). The customer states his initial (incomplete) need, the supplier(s) proposes a service bundle, which is critiqued by the customer, and the supplier(s) then modifies the bundle, until the proposed bundle satisfies the need sufficiently. While critiquing, the customer has to rank the proposed bundles to state to what extent these bundles match the customer's need. This ranking is done by compensatory decision techniques used in marketing, weighting various customer-criteria [9], and a non-compensatory decision technique, MoSCoW from DSDM[3] (see section 2.1 for further elaboration on MoSCoW), to deal properly with `must-have' and 'won't have'requirements.

The contribution of this paper is two-fold: First, it discusses a *real-life* application of e^3-service. In earlier work [12] we introduced e^3-service using a simple ISP-example but did not validate in how far it could be used for reasoning about real, *complex* domain knowledge as found in –for instance- the healthcare domain. Second, this paper discusses mechanisms that could constitute a useful addition to the existing mechanisms in the existing social chart, more particularly (1) how to rank bundles according to how well they fit with consumer preferences and (2) how to adjust ranking of bundles and/or find alternatives based upon constraints imposed by found bundles.

In sections 2.1 and 2.2, we discuss the most important concepts e^3-service employs to model services. In section 2.3, we briefly reflect on the validation method used for the healthcare case study. In section 3, we show how e^3-service uses consumer-supplier interaction to reason about bundling healthcare-services. Section 4 provides a brief discussion of related work, while section 5 discusses reflections from domain experts and concludes.

2 e³-service modelling concepts

This section serves as background knowledge for section 3, where we frequently refer to the concepts below. In e³-service, as in established literature [13], we distinguish between two perspectives on services: A consumer and a supplier side perspective (see [2] for further elaboration). This section discusses the main concepts for modelling services with e³-service from both perspectives, using the healthcare case as a running example. Additionally, in section 2.3, we briefly discuss the validation method used in the healthcare case.

2.1 Concepts for modelling services from a customer perspective

The conceptual model for modelling a customer perspective service catalogue with e³-service can be found in figure 1. For guidelines on how to create a catalogue, see [2].

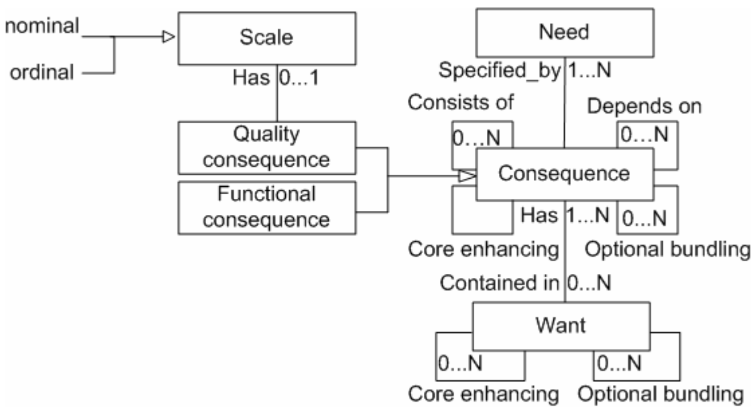

Figure 1: Conceptual model for creating a customer perspective service catalogue

Need. A need represents a problem statement or goal, independently from a solution direction [4].
EXAMPLE: A typical problem for informal carers of people with dementia is 'I cannot cope anymore, what can help?'. Note that this need does not yet include a notion of a solution.

Consequence. A consequence is anything that results from consuming (a combination of) valuable service properties [5].

- *Functional consequence.* A functional consequence represents the functional goal that can be achieved through consumption of a service that has a certain valuable property.
  EXAMPLE: Functional consequences from the service `Dining table' are `meal preparation', 'social contacts for person with dementia' and 'social contacts for informal carer' (see section 2.2 for further elaboration of the dining table service).
- *Quality consequence.* A quality consequence expresses qualitative properties of functional consequences in customer terminology.
  EXAMPLE: The consequences `Sugar free' and `Kosher' are quality consequences of the functional consequence `Meal preparation'.

Relations in this part of the ontology are:

- *Specified by.* A need is specified by zero or more consequences of the potentially consumed service. Thus, the consequences show how a need is satisfied.
  EXAMPLE: The consequences `practical support for persons with de dementia' and `social support for informal carer' are solution directions showing how the need `I cannot cope anymore, what can help? ' can be satisfied.
- *Core enhancing/Optional bundling.* Consequences can be value-enhancing with respect to a core consequence. To acquire the enhancing consequence, the core consequence must also be acquired. An optional bundling relation between two consequences A and B indicates that consequence B can add value to consequence A, but that consequence B can also be acquired separately from A.
  EXAMPLE: The consequence `Adjustments to home' can add value to the consequence `Loaning of medical equipment'. This is because when someone for example rents a wheelchair, home adjustments such as ramps or a lowered kitchen sink might also be needed.
- A consequence *depends on* one or more other consequences. This relation states that one consequence cannot be acquired without another consequence.
  EXAMPLE: `Kosher' and `Sugar free' are two consequences indicating dietary needs for the consequence 'Meal preparation'.
- A consequence may *consist of* one or more other consequences. Such consequence *laddering* [5] can be used to specify abstract consequences into more concrete consequences until a sufficiently detailed consequence is found for which solutions can be offered.
  EXAMPLE: `practical support for person with dementia' can be specified by the consequences 'Meal preparation' and 'Adjustments to home'.

Want. A want is a specific, supplier-independent solution that is commercially feasible to be provisioned on its own. However, as a want indicates a solution available in the market, at least one supplier should be willing to provide the solution.
Wants, interpreted as supplier-independent solutions, can be typically found in existing *service taxonomies* such as the NAPCS [8].
Relations in this part of the ontology are:

- A want *has* one or more consequences. These consequences are used to state how the want, being a service that can be provisioned commercially by a single supplier, satisfies a need or specifies a consequence.
  EXAMPLE: The want `Dining table' has the consequence `Meal preparation', which in turn specifies the consequence `practical support for person with dementia'.
- A want is in a *Core-Enhancing relationship* with zero or more other wants. The Core-Enhancing relationship indicates that for a certain want A, provided that want A is acquired, there exist wants B that could add value
  to A. This relationship exists also between consequences; the same relation is relevant on the level of wants, as these wants actually package sets of consequences available in the market. Analogously, a want may be in an *Optional Bundling* relationship with zero or more other wants.
  EXAMPLE: The value-enhancing want `Handyman' is in a Optional Bundling relationship with the basic want 'Loaning service' because, as mentioned under consequences, someone loaning medical equipment might also need adjustments to the home. Note here that the Optional bundling relationship between consequences referred to earlier indicates *why* a Handyman can add value to the Loaning service.

Scale. In e³-service we use scales [7] to cluster related quality consequences. We use two well-known types of scales:

- *Nominal.* A nominal scale indicates that a relationship exists between quality consequences, but introduces no ordering or ranking on these consequences.
  EXAMPLE: 'Sugar free' and `Kosher' are nominal categories that indicate dietary needs. The preference for either one depends on the customer.
- *Ordinal.* An ordinal scale introduces an ordering on consequences such that it is possible to state that consequence X is better than Y (but not *how much* it is better).
  EXAMPLE: Defining < 3 months as a `short waiting list' and ≥ 3 months as a `long waiting list' yields an ordinal scale for lengths of a waiting list.

Discussion: MoSCoW. MoSCoW is a prioritization mechanism used in DSDM software engineering projects to prioritize the implementation of software requirements. With the assumption that consequences are essentially requirements stated by the consumer, we use this prioritization mechanism also in e³-service.
MoSCoW distinguishes between the following four priority levels:

- Must-have requirements: Implementation of these requirements is *critical*.
- Should-have requirements: Implementation of these requirements is not critical, but should be done, if possible.
- Could-have requirements: These entail nice-to-have features
- Won't have requirements: These requirements should definitely not be implemented, at least not for the first release.

From MoSCoW, we adopt the notions of must-have and won't haves, so that the consumer can express that a consequence (1) definitely *must be* or (2) definitely *must not be* satisfied by a service bundle (solution).

2.2 Concepts for modelling services from a supplier perspective

The e³-service supplier ontology is depicted in Figure 2 using an example healthcare service. For a detailed discussion of this ontology, see [14].

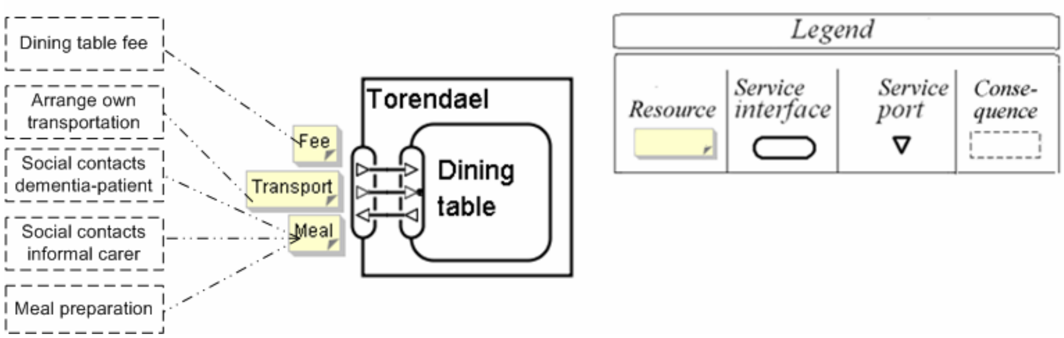

Figure 2: Example of individual supply side service

Consequence. A consequence is anything that results from consuming (a combination of) valuable service properties [5]. Similar to the customer perspective ontology, there exist several supply-side types of consequences and of relations between consequences. Since consequences are used both in the customer and supplier perspective of e³-service, they form the glue between both ontological perspectives. Hence, the concept *consequence* is the ontological as well as reasoning key in matching customer needs to supply-side services on offer.

Service property. A service property is a supplier-specific attribute.
Relations in this part of the ontology are:

- *Realized by:* A consequence is realized by one or more service properties. *EXAMPLE:* The consequences `social contacts informal carers' and 'social contacts for person with dementia' are realized by the service property `in group, with other persons with dementia and their informal carers' provided by the supplier (say) Torendael (a residential care home).
- A service property is always *part of* a supplier-specific *resource.* The distinction that we make between a service property and a resource is that a resource can be provisioned on its own commercially, while a property cannot. A property is therefore always part of a resource.
  EXAMPLE: consider that the service property `in group, with other persons with dementia and their informal carers' is part of the resource `Meal'.
- A resource is always *attached to* one or more service ports. In turn, each service port is always *part of* exactly one service interface.
  EXAMPLE: consider the Dining table service in Figure 2. Here, one sees that the resource 'Meal' is attached to a service port (the arrowhead), which in turn is part of the service interface of the Dining table bundle. The service interface indicates the actual bundling of resources from a (multi-)supplier perspective: all resources should be exchanged between the customer and supplier, or none at all. So, by considering the Dining table interface of our example, you know that you will also have to arrange your own transportation and give up a Dining table fee since these additional resources are attached to the other ports in this interface.

The important point here is that consequences also exist at the supply side. We furthermore assume that consequences ensuing from a supply-side service catalogue are expressed in a marketing vocabulary that represents the customer perspective. This is crucial for the match-making process, discussed in section 3.

2.3 Validation method

In earlier work [12], we developed a software tool that is capable of reasoning with instantiations of the conceptual models from e^3-service. Using an interactive dialogue, the tool allows a user to find services satisfying his need against acceptable sacrifices. Note here that the tool follows exactly the generic, *case-independent*, reasoning steps as described in [12]: No steps are added or removed.

For this case, we made instantiations of the conceptual models from e^3-service for the healthcare domain. These instantiations were made with information from the existing social chart, supplemented with information gathered from interviews with domain experts (on e.g. constraints such as pricing and waiting lists). These instantiations were used as tool input to create a dialogue that allows a user to find *health-care services* based upon his/her care needs.

Using the tool, we validated the usefulness of e^3-service with two 2+-hour tool demonstrations with the healthcare domain experts that were mainly responsible for creating the original social chart. The tool demonstrations consisted of scenario walkthroughs, where realistic consumer needs – such as a customer that tries to find a meal-preparation service – were taken as starting points to show how the tool interacts with the consumer to find bundles of healthcare services. For each scenario walkthrough, the domain expert then commented in how far the principles demonstrated could constitute a useful addition to the existing social chart. Since the tool conforms *exactly* to the reasoning steps from e^3-service and takes conceptual models from e^3-service as input, our assumption is that through such a demonstration, we can also validate the usefulness of principles embedded in e^3-service.

3 Reasoning about finding and ranking bundles of health-care services

In this section, we show how e^3-service reasons about needs-driven service bundling. We explain each of the high-level reasoning steps from figure 3. For illustration, we use a running example of an informal carer using the social chart to find a service that could aid with providing a meal.

3.1 Choose customer need and choose consequences

Starting from a customer need, we derive an initial set of consequences specifying this need. The tool asks the customer to choose a particular consequence (via prioritization) and then checks whether the selected consequence `consists of' other, more detailed, consequences. If so, the customer is again asked to make a choice after which,
for all chosen and implied consequences, the reasoning process again reviews whether considered consequences `consist of' other consequences. This continues until no more `consist of' relationships are found. This process is also called `laddering' and is a well-known practice from marketing theory [5].

Next, our e^3-service tool derives one or more wants that this consequence is a part of. Here, the assumption is that the experts that created the service catalogue used for the reasoning process have defined solutions upfront for detailed (i.e. leaf) consequences. Then, using this want as a starting point, the reasoning mechanism derives additional consequences that are also part of the want. Thus, this first step is a kind of bootstrapping process to find a highly ranked consequence, and it continues by evaluating wants (that includes this consequences) to ensure that needs elicitation is grounded in services that are in fact available on the market (i.e., wants).
Now we let the customer decide, per scale, on prioritization of quality consequences that are related to the functional consequences. When consequences are defined on a *nominal* scale, the customer is asked to assign to each of the consequences an importance value ranging from 1 (unimportant) to 10 (must-have, i.e. the offered service bundle should always include this consequence). Note that the process for prioritizing consequences defined on an *ordinal* scale differs from prioritizing on a nominal scale. However, since this case description only contains nominal scales, we choose to not elaborate further on this. Finally, by default, the selected functional consequences receive an importance ranking of `10' (must have).
We can now use the customer preferences expressed in terms of their consequences to compose the appropriate service bundles.

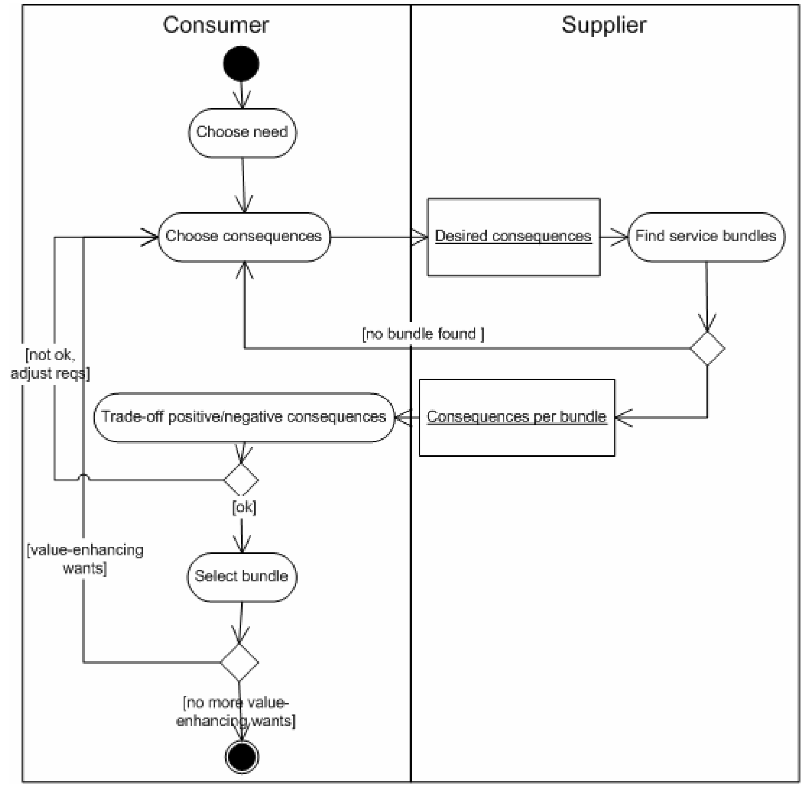

Figure 3: Generic reasoning structure of e[3]-service

Case: The consumer selects the need `cannot cope anymore, what can help?' From the four functional consequences specifying this need (see the consumer side service catalogue in figure 4) the consumer chooses `Practical support for person with dementia' as it comes closest to a meal preparation service. In turn, the consequence 'Practical support for person with dementia' is specified further by, amongst others, the consequences 'Loaning of supporting equipment' and 'Meal preparation'.

Next, the tool finds additional functional consequences for detailed, or leaf, consequences using wants as a bootstrap mechanism. As an example, consider that 'social contacts informal carer' and 'social contacts for person with dementia' are found for the leaf consequence 'meal preparation', using the want 'Dining table' as a bootstrap mechanism (see figure 4 for reference).

Then, per want, the tool presents the consumer with the main functional consequence that states *why* this want is of interest and with any additional functional consequences contained in the want. For instance, the want Dining table is presented as follows:

The want Dining table satisfies the consequence Practical support for person with dementia through the consequence
*Meal preparation
This want also has the following consequences:
*Social contacts person with dementia
*Social contacts for informal carer

Let us assume that an informal carer selects the want 'Dining table', because besides 'Meal preparation', it also offers the possibility of 'social contacts for person with dementia' and 'social contacts for informal carer'. Next, the tool asks the informal carer to assign preference scores to the *quality consequences* for the selected functional consequences. For the functional consequence 'Meal preparation', a scale for which an informal carer must express his/her preference scores is for instance 'Diet'. We assume that an informal carer scores the quality consequences ‘Kosher’ with 1 and 'Sugar free' with 2 to indicate that these consequence are of little value.

Finally the chosen functional consequences 'Meal preparation’, ‘Social contacts for person with dementia’ and ‘ Social contacts for informal carer’ will be given an importance score of 10 by default to indicate that these are must-have features.

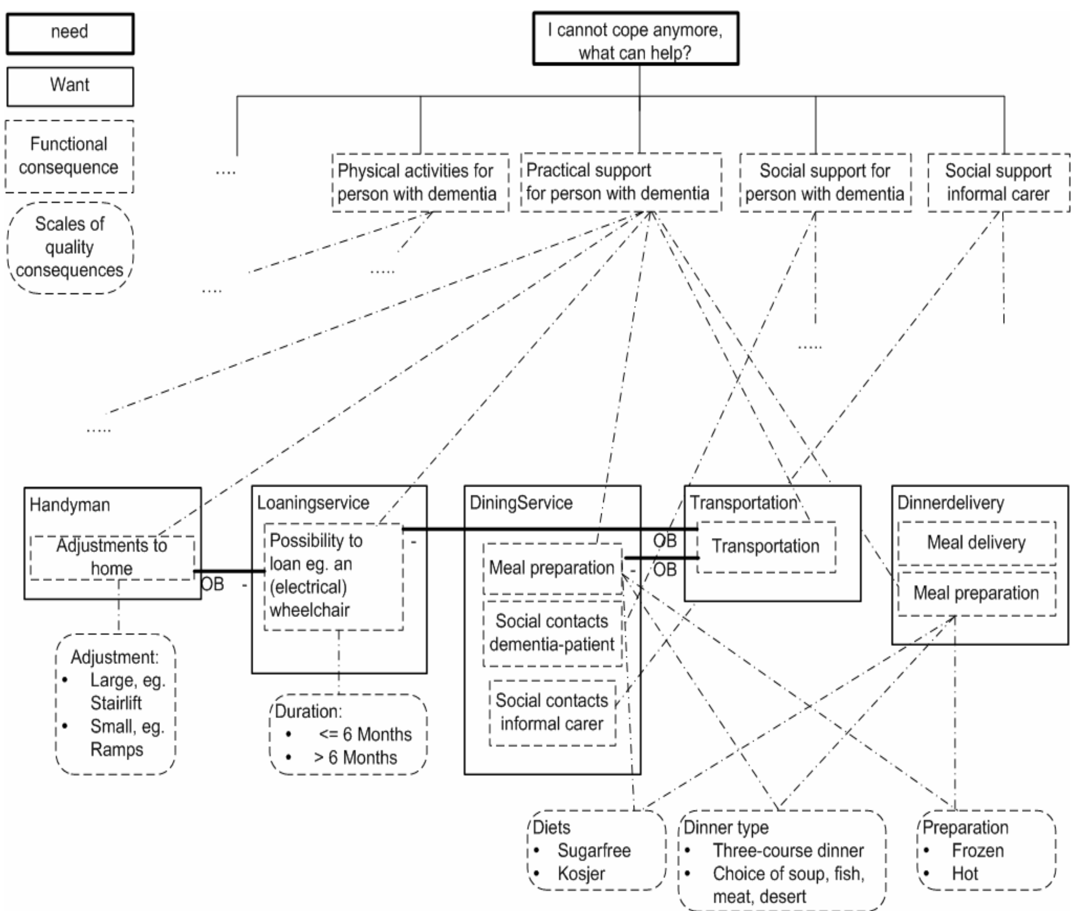

Figure 4: Consumer perspective catalogue for healthcare services

3.2 Composing Service Bundles

After this customer dialogue, we match the set of consequences desired by the customer to consequences defined from a supplier's perspective.

We can do this because (cf. section 3.2) the concept and theory of consequences provides the bridging connection between the customer and supply sides for the match-making process. The computational result of this consequence matching is a subset of supply-side consequences that, together with the prioritization scores provided by the customer, can be used to reason about (1) finding (composing) service bundles and (2) ranking service bundles according to prioritization scores.

We first search service bundles that can satisfy all `must-have' consequences. To find these bundles, we search for each must-have consequence:

(1) Supplier-specific *service properties* jointly satisfying the consequence. Next, we find the supplier-specific resources that contain these properties and finally bundles that contain these resources.
(2) Supplier-specific *resources* that jointly satisfy the consequence. We then find bundles containing these supplier-specific resources.

The result is a set of bundles for each must have-consequence. The set of bundles satisfying all must-have consequences then, is an intersection of all sets of bundles.
So, if must-have consequence A is satisfied by the set of bundles {X, Y, Z} and must-have consequence B is satisfied by the set of bundles {W,X}, the set of bundles containing all must-have consequences is {X}.

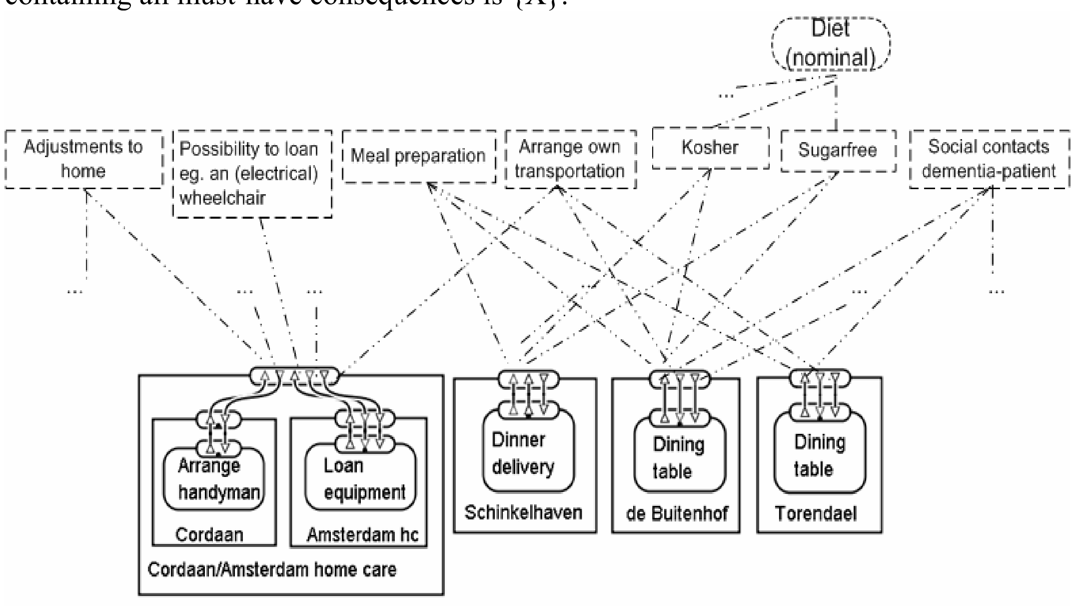

Figure 5: Sample of bundles of healthcare-services, generated cf. [Baidaphd]

Case: First, the tool matches all supply-side consequences to all consequences desired by the informal carer. In this scenario, the matching results in the following subset of supply-side functional and quality consequences and their respective importance scores: {social contacts for person with dementia (10), social contacts informal carer (10), Meal preparation (10), Sugar free (2), Kosher (1)}. For reference, see figure 5 (sample of the supplier catalogue).

The tool now finds the pre-generated service bundles that can satisfy all 'must-have' consequences.

(1) The tool finds the the set of bundles {'Dining table' Torendael, 'Dining table' de Buitenhof} for the must-haves 'social contacts informal carer' and 'social contacts for person for with dementia', because these bundles realize both consequences through the *service property* 'in group, with other persons with dementia and their informal carers' of the supplier-specific 'Meal' resources contained in these bundles.
(2) The tool finds the set of bundles {'Dining table' Torendael, 'Dining table' de Buitenhof, ' Dinner delivery' Schinkelshaven} for the consequence 'Meal

preparation', because all these bundles can satisfy this consequence through the *resource* 'Meal' (see figure 5 for reference).

When the tool now intersects above sets, it finds the bundles 'Dining table de Buitenhof' and 'Diningtable Torendael kosjer'.

3.3 Ranking Service Bundles

The next step is to rank the found relevant bundles. For this, we need to provide a score to indicate whether a consequence defined on a nominal scale is present in a service bundle. This score we provide in a binary way: if a consequence is present in a bundle it scores 1, else 0. Now that we have numerical values to express consequence scores as well as importance scores from the customer, we calculate a ranking score for each service bundle by using the multi-attribute scoring formula $SB_i = \sum_{j=1}^{n} \frac{w_j}{10} v_{ij}$, where SB_i is the ranking score for service bundle i, w_j is the importance ranking of consequence j as provided by the customer, and v_{ij} is the numerical value for the consequence j of service bundle i. After having calculated to what extent a service bundle fits with customer preferences, we find for each bundle additional consequences that a customer also must acquire by using the service interface of that bundle (see example).

Case: In this scenario, the tool thus calculates an importance score for each bundle based upon the consequences Sugar free, Kosher.

First, for each bundle, the consequences themselves are scored based upon whether or not they are satisfied by that bundle. Since in this scenario all consequences are defined on a nominal scale, this score is always provided in a binary way. For example, because 'Kosher' is present in the bundle Dining table Torendael kosher, this bundle will score a 1 for the consequence Kosher.

Now the tool calculates a score for each bundle by using the previously discussed multi-attribute scoring formula. For this scenario, this leads to the following scores:

- 0.2 for Dining table de Buitenhof ((0.2 * 1) for the consequence Sugar free)
- 0.1 for Dining table Torendael kosjer ((0.1 * 1) for the consequence kosher).

Now, for each bundle, we find its full set of consequences. For example, using the service interface of the bundle 'Dining table Torendael' depicted in figure 5, we find two additional service ports: one containing the resource 'fee' with a property of €7,- and one containing the resource 'transport'. From these, we then derive the consequences 'Dining table fee' and 'Arrange own transportation'.

3.4 Trade-off positive/negative consequences

We present the found bundles in a ranking to the consumer, where the ranking is based upon the multi-attribute score discussed in the previous paragraph. Also, for each bundle, the consumer is presented with a specification of the consequences s/he receives from a bundle and the consequences s/he has to give up to acquire a bundle. Furthermore, the tool shows a specification of any monetary objects in terms of a pricing model (for an elaboration of pricing models, see [11]).

The tool now presents the consumer with the choice to also score *negatively valued* consequences (of course, s/he can also choose to make the trade-off herself). When this option is selected, the consumer is again asked to attach a score from 1 to 10 to each consequence, with the difference that 1 is now used to indicate that the presence of a consequence in a service bundle is not negatively valued and 10 is used to indicate that a consequence must be *absolutely absent* from a bundle.

Based upon the scores attached to the negatively valued consequences, the tool again uses the scoring mechanism used to find and rank service bundles only now with the difference that (1) each bundle that satisfies a negatively valued consequence with a score of 10 (won't have) is automatically discarded, irrespective of other consequences satisfied by that bundle. As before, we rely on a non-compensatory decision rule, only now to discard a bundle *containing* a consequence (2) We compute again a score for each bundle remaining after (1) by using a multi-attribute scoring formula, only now while *subtracting* the scores of the negatively valued consequences satisfied by each bundle from the score already calculated for that bundle.

The consumer now has the option to either select a bundle from the ranking or, in case s/he finds the costs incurred for the bundles too high, to go back to the step 'choose consequences' and change his/her requirements (see the generic reasoning steps in figure 3 for reference). When the consumer returns to the step 'choose consequences', s/he can start again at any point in the consequence ladder.

When the consumer selects a bundle, we review whether there are any value-enhancing wants (either core/enhancing or optional bundled wants) for that bundle. As an example of a value-enhancing service for a Dining table, you can imagine an afternoon or evening with recreational activities for the person with dementia which, besides possibly keeping fit (mentally and/or physically), is an additional service to possibly further enhance social contacts. Since this paper focuses on how to (re-)rank bundles however, we will not discuss reasoning with these value-enhancing services further.

Case:
The consumer is presented with the ranked bundles:

(1) Dining table residential care home de Buitenhof. Score .2
(2) Dining table residential care home Torendael. Score: .1

For each bundle, the tool now presents the full set of consequences to the informal carer. For example, the tool specifies the bundle Dining table de Buitenhof as follows:

Please find below what the bundle Dining table de Buitenhof can provide you with

- Meal preparation
- Social contacts for informal carer
- Social contacts for person with dementia
- Sugarfree for the consequence Meal preparation

For the bundle Dining table de Buitenhof you have to give up the following:

- Arrange own transportation
- Dining table fee

Please note that the tool also specifies each bundle in terms of a pricing model so that it is clear to the informal carer what fee has to be paid to acquire a bundle. Due to lack of space however, we chose not to discuss this pricing model in this example.

Next, the informal carer chooses to also score *negatively valued* consequences. We assume she is content with the Dining table fees from both Dining table Torendael and Dining table de Buitenhof , but attaches a 10 to 'arrange own transportation' since s/he absolutely wants to avoid this.

Next, the tool finds that both Dining table services contain the consequence 'arrange own transportation'. Because the informal carer scored this consequence as an absolute won't have, the tool discards both bundles and informs the informal carer that none of the initially found bundles lacks the consequence ‘arrange own transportation’.
Since the informal carer now is aware that a type of dining table service *always* requires one to arrange ones own transportation and that it therefore would not make sense to adjust the requirements for the consequences attached to the want dining table, the informal carer chooses to start again with the consequence 'Practical support for person with dementia'. Here, s/he observes that the want 'Dinner delivery' also contains the consequence ‘Meal preparation’ and, since 'Meal preparation’ was the initial requirement the informal carer started with, we assume that s/he selects this want.

Next, upon eliciting the preferences regarding quality consequences for the functional consequence 'Meal preparation', the tool asks whether the informal carer would like to reuse the preferences initially entered for the nominal scale 'Diet'. We assume here that the dietary needs have not changed and thus, the tool now finds and ranks service bundles again for the quality consequences and their preferences scores as initially entered, only now supplemented with the functional consequence 'Meal preparation' only.
Based upon these consequences the tool now finds services that deliver a meal at home and Dining table services, but consequently discards the Dining table services because it knows that the informal carer absolutely wants to avoid the consequence `arrange own transportation'.
Finally, the informal carer is presented with a specification of the consequences from each dinner delivery service and, since s/he is content with having to give up a small fee each time a meal is delivered, the informal carer decides to acquire one of these services.

4 Related work

Quality Function Deployment [16] allows a company to translate a high-level need from the customer to required product features that together satisfy this need. The idea is that it brings together the marketing department and the more technically-oriented engineers to together discuss what implementation of a customer need would mean in terms of technical product features, including trade-offs to be made when implementing a feature. For example, a customer might have the need for a car-door to close more easily, based upon which engineers state that this can be realized through features such as actuators, or by using lighter materials. Additionally, what the engineer might also state through QFD is that using lighter materials either means that the car becomes more expensive or, in case of low-cost lighter material, that implementation of this feature would be to the detriment of the car’s isolation.
As in our work, QFD essentially distinguishes between a customer perspective (used by marketers) and supplier perspective (used by engineers) and performs a matching between these. However, it differs from our work in that it, amongst others: (1) is a paper-based toolkit to facilitate cross-departmental communication and as such, is not meant for computational reasoning and (2) does not distinguish between basic and value-enhancing features.

5 Reflections and conclusion

5.1 Reflections from domain experts

The domain experts' feedback most relevant for this paper was that the idea of consumer-supplier interaction is useful when someone is tied to a domain where services contain negative consequences such as a restricted personal budget or long waiting lists. They stated that constraints can indeed force someone to adjust his/her requirements and maybe even to reject a bundle altogether.

However, the domain experts did emphasize that consumer-supplier interaction was only found to be really useful when thinking about services where negative consequences have a *prominent* role. Otherwise, the current configuration mechanism also suffices. For instance, they pointed out there are also healthcare services in the Netherlands, such as admission into a care-home, that are *fully* covered by a general insurance fund (the AWBZ) and that as such, might not put such a heavy burden on someone when acquiring the service.

Thus, in a domain where services are not heavily constrained, explicitly weighing benefits against costs might not be necessary. This, the domain experts stated, is relevant since scoring all consequences – positive and negative - can also be quite time-consuming for the end-user.

Additionally, the domain experts pointed out that the supplier-independent service dependencies – to which we briefly referred in section 2 - could be a useful addition to the current social chart since these actually reflect the way in which healthcare consultants provide advice on healthcare services. For instance, with e^3-service you can relate the loaning of medical equipment to a handyman, which is not possible in the current version of the social chart.

Note here that ideally, of course, you would like to do *real user* testing in a decently sized population of, say, 50 users. However, at the time of this writing, we did not yet have the resources to perform this kind of large scale testing. Also, the tool is currently designed as a proof-of-concept: it provides a text-based dialogue instead of a graphical user interface and the entering domain knowledge is done through an ontology editor, which is a notoriously error-prone process. As such, the tool is not yet aimed at being directly usable by end-consumers.

5.2 Conclusion

In this paper we discussed how e^3-service uses consumer-supplier interaction to find and rank bundles of healthcare services. We discussed reflections of healthcare practitioners on e^3-service, which were based upon demonstrations of our software prototype that fully implements the principles from the e^3-service method. The feedback was promising, stating that the demonstrated principles, such as ranking of bundles, could indeed constitute a useful addition to the existing social chart.

Concerning future research, one point is that e^3-service should be more thoroughly validated. Although the first results are encouraging, ideally we would like further to further test our approach by means of (1) including demonstrated principles in the existing social chart (2) letting real users test the adjusted social chart.

References

[1] Dröes, RM. et al. (2005): A dynamic interactive social chart in dementia care, Studies in Health Technology and Informatics, Vol. 114, pp. 210--220, IOS Press

[2] de Kinderen S., Gordijn, J. (2008): e^3-service - an ontological approach for deriving multi-supplier IT-service bundles from consumer needs, Proceedings of the HICSS-41 (CD-ROM), Hawaii jan. 7-10 2008, Computer Society Press

[3] Stapleton J. (1997): “ Dynamic Systems Development Method”, Addison Wesley Longman, Reading, Massachusetts

[4] Arndt, J. (1978): How broad should the marketing concept be?, Journal of Marketing, Vol. 42, No. 1, pp. 101-103

[5] Gutman J., Reynolds T.J. (1988): Laddering theory-analysis and interpretation, Journal of Advertising Research, Vol. 28, No.1, pages:11.

[6] Elrod, T., Johnson, R.D, White, J.(2004): A new integrated model of non-compensatory and compensatory decision strategies, Organizational Behavior and Human Decision Processes, Vol. 95, No. 1, pp. 1-19

[7] García, F., et al. (2006): Towards a consistent terminology for software measurement ,Information & Software Technology, Vol. 48, No. 8, pp. 631-644

[8] Ambler, C., et al. (1998): Developing a Product Classification System for the United States, US Census Bureau

[9] Hutton Barron, F., Barrett B.E. (1996) Decision Quality Using Ranked Attribute Weights, Management Science, Vol. 42, No. 11, pp. 1515-1523

[10] Woodruff, R.B. (1997): Customer value: The next source for competitive advantage,
Journal of the Academy of Marketing Science, Vol. 25, No. 2, pp. 139-153

[11] de Miranda, B., Baida, Z., Gordijn, J. (2006) : Modeling Pricing for Configuring e-Service Bundles, Proceedings of The 19th Bled eCommerce Conference, june 5-7 2006

[12] de Kinderen, S. and Gordijn, J. and Akkkermans, H. (2009), Reasoning about customer needs in multi-supplier IT-service bundles using decision models , ICEIS 2009 - Proceedings of the Eleventh International Conference on Enterprise Information Systems, Volume ISAS, Milan, Italy

[13] Grönroos, C. (1990): Service Management and Marketing, Lexington Books

[14] Baida, Z.S. (2006): “Software-aided service bundling”, PhD-thesis Free University Amsterdam

[15] Griffin, A. and Hauser, J.R. (2003): The voice of the customer, Alfred P. Sloan School of Management, Massachusetts Institute of Technology