REASONING ABOUT CUSTOMER NEEDS IN MULTI-SUPPLIER ICT SERVICE BUNDLES USING DECISION MODELS

Sybren de Kinderen, Jaap Gordijn and Hans Akkermans
The Network Institute, VU University Amsterdam, The Netherlands

Keywords: Commercial services, Service bundling, Customer dialogue, Computational reasoning, Ontologies.

Abstract: We propose a method, e^3*service* , to reason about satisfying customer needs in the context of a wide choice of multi-supplier ICT service bundles. Our method represents customer needs, their ensuing consequences, and the services that realize those consequences in a service catalogue. This catalogue is then used by a reasoner, which elicits customer needs, computes their consequences, and automatically matches these consequences with services offered by suppliers. The e^3*service* method has been implemented and tested in software to demonstrate its feasibility.

1 INTRODUCTION

Commercial ICT services, such as VoIP, bandwidth, email, web hosting, etc., are electronic, i.e. they can be ordered by the customer online and provisioned by the suppliers online. Usually, services never come alone: customers typically need an ICT service *bundle* to satisfy their needs. Moreover, these services can be obtained from multiple competing suppliers. So, the customer has to decide online about the best selection — both in terms of the offered services and of their supplier. In this paper, we show how this customer-driven decision process can be facilitated by automatic reasoning and match-making tools that bridge the gap between general customer needs and actual ICT services on offer. The key idea to enable this is to employ a Propose/Critique/Modify (PCM) problem-solving method (Motta, 1999) that iterates a number of times through a knowledge-based dialogue with the customer, so as to gradually and mutually adjust solutions (services) to problems (needs).

2 THE e^3*service* ONTOLOGY

We use a running motivating example of a customer who wants to communicate with family abroad, and employs hosted email as a solution to do so. We then show how this example can be represented by using our e^3*service* ontology for service need and service bundle modeling. This ontology considers two different perspectives: the customer perspective (section 2.1), and the supplier perspective (section 2.2).

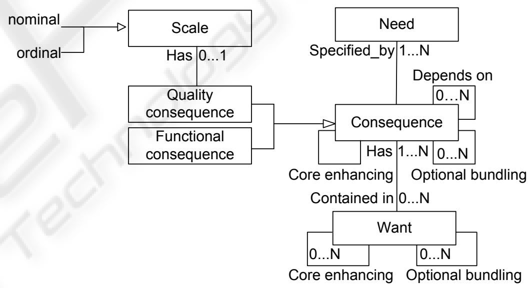

Figure 1: Ontology used for reasoning about service bundling from the customer perspective.

2.1 Customer Perspective Ontology

The customer perspective in the e^3*service* ontology is presented in Figure 1; for an overview of its background ideas, see (de Kinderen and Gordijn, 2008). Below, we summarize the key concepts and relationships.

Need. A need represents a problem statement or goal, independently from a solution direction (Arndt, 1978). *EXAMPLE:* A customer may have a need 'communicate with family abroad'. 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 (Gutman and Reynolds, 1988).

de Kinderen S., Gordijn J. and Akkermans H. (2009).
REASONING ABOUT CUSTOMER NEEDS IN MULTI-SUPPLIER ICT SERVICE BUNDLES USING DECISION MODELS.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 131-136
DOI: 10.5220/0001985001310136

• *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:* A functional consequence from the want 'email' is 'send and receive text'.
• *Quality consequence.* A quality consequence expresses qualitive properties of functional consequences in customer terminology. *EXAMPLE:* The consequences 'small mailbox' and 'large mailbox' are quality consequences of the functional consequence 'send and receive text'.
Relations in this part of the ontology are:
• *Specified by.* A need is specified by zero or more consequences. The consequences - as results of a service - show how a need is satisfied. *EXAMPLE:* The consequences 'send and receive text' and 'hear and speak voice' are ways to satisfy the need 'communicate with family abroad'.
• *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:* As an example of core/enhancing consequences, the consequence 'reduce reception of unwanted text messages' adds value to the consequence 'send and receive text'.
• A consequence *depends on* one or more other consequences. This relation states that one consequence cannot be acquired without another consequence. *EXAMPLE:* 'Use at any site with connectivity' and 'local' (e.g. indicating a pre-configured PC) are two consequences indicating the location from where 'send and receive text' can be used.
• A consequence may *consist of* one or more other consequences. Such consequence *laddering* (Gutman and Reynolds, 1988) 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:* 'send and receive text'can be specified by the consequences 'send text' and 'receive text'.

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. *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. *EXAMPLE:* the want 'email' has the consequence 'send and receive text', which in turn is a way to satisfy the need 'communicate with family abroad'.
• 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 'Spam filtering' is in a Core-Enhancing relationship with the basic want 'email', since there is no business logic in acquiring a spam filter without an email service. The Core Enhancing relationship between consequences indicates *why* a spam filter want adds value to an email want.
Scale. In e^3*service* we use scales 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:* 'use at any site with connectivity' and 'local' are nominal categories that indicate email access options. 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 $\leq$ 0.2 GB as a 'small mailbox' and $>$ 2 GB as a 'large mailbox' yields an ordinal scale for the size of an email box.

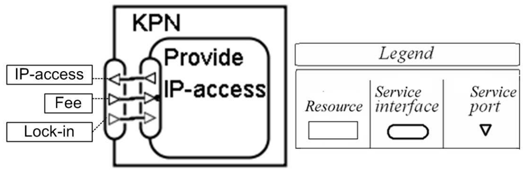

Figure 2: Example for supply-side concepts: resources, service ports and service interfaces.

2.2 Supplier Perspective Ontology

The e^3*service* supplier ontology is depicted in Figure 2 using an example ICT service. For a detailed discussion of this ontology, see (Baida, 2006).
Consequence. A consequence is anything that results

from consuming (a combination of) valuable service properties (Gutman and Reynolds, 1988). 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 e^3*service* , they form the glue between both ontological perspectives. Hence, the concept of *consequences* 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 consequence 'send and receive text' is realized by the service properties 'protocol=POP3' and 'upspeed = 256 kbps' and 'downspeed= 512 kbps', all provided by the supplier (say) KPN.

• 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 properties 'upspeed = 256 kbps' and 'downspeed= 512 kbps' from KPN are part of the resource 'KPN IP access'. Here, up- and downspeed from KPN can only be delivered in combination with IP access from KPN.

• 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 KPN IP access service in Figure 2. Here, one sees that the resource IP access is attached to a service port (the arrowhead), which in turn is part of the service interface of the IP access 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 KPN-IP-access interface of our example, you know that you will also have to give up customer lock-in and a certain 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 below.

3 FROM CUSTOMER NEEDS TO ICT SERVICE BUNDLES

How to derive a service bundle, given a customer need? The high-level reasoning process is shown in Figure 3. We have implemented this reasoning process in a software tool for demonstration and validation purposes.

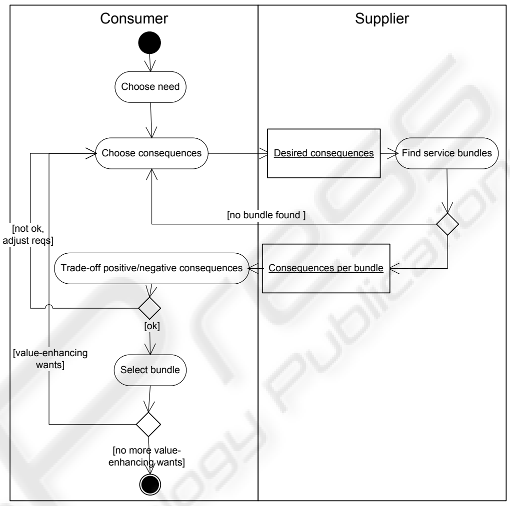

Figure 3: The generic reasoning structure of the e^3*service* method.

3.1 Needs & Functional Consequences

Starting from a customer need, we derive an initial set of consequences specifying this need. In the customer perspective ontology (section 2.1), we do this by expanding the relation 'specified by' from a single need to one or more consequences specifying this need. First we focus on functional consequences only, since a functional consequence shows (partly) what goals can be achieved by a service and so shows *how* a service can satisfy a need. The reasoning process first 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 (Gutman and Reynolds, 1988).

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).

Case Example. We start with the need: 'communicate with someone over a distance'. By following the relation 'specified by', we derive the functional consequences 'send and receive text' and 'hear and speak voice'. Let's assume that the customer chooses 'send and receive text'. Subsequently, we look whether this consequence is specified further by reviewing whether a 'consists of' relationship exists. When we look at the service catalogue used for this specific case (Figure 4) we see that 'send and receive text' is not specified further. Next, we derive the want that contains 'send and receive text'. For this simple case, the only want containing 'send and receive text' is 'email access'. Then, our tool derives all possible functional consequences from the want 'email access'; in this case we derive, next to 'send and receive text', also the consequence 'use for newsletter'. Thus, the consumer is presented with the group of consequences 'send and receive text', which the tool indicates as a solution result satisfying the need 'communicate over a distance', but also the additional possibility 'use for newsletter'.

3.2 Choose Additional Consequences

When a particular functional consequence is selected, we derive its quality consequences by following the relation *depends on* in our customer ontology. These quality consequences are then grouped by scale by (1) deriving, for each quality consequence, its scale by following the relation 'has' between quality consequence and scale and (2) grouping together quality consequences that are defined on the same scale.

Now that the conseqeunces are grouped per scale, we let the customer decide, per scale, on consequence prioritization. Depending on the *type* of scale (recall that different types of scales exist, e.g., nominal or ordinal) the customer is presented with two different prioritization tasks: (1) 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); (2) When consequences are defined on

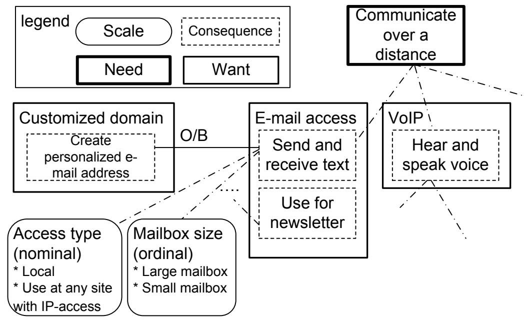

Figure 4: Customer catalogue for the email example.

an *ordinal* scale, the customer is asked to assign a preference ordering to the *scale only* and not to the consequences in the scale. For the scoring of consequences, we use the ranking from best to worse which is inherent to an ordinal scale. How we convert this best-to-worse ranking of consequences into a score, is discussed in the next step (compose and rank service bundles, see below). Finally, by default, the selected functional consequence receives 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.

Case Example. By following the relation *depends on*, we infer that there are four quality consequences that depend on the functional consequence 'send and receive text': 'small mailbox', 'large mailbox', 'local' and 'use at any site with connectivity' (cf. Figure 4). These quality consequences are grouped per scale, resulting in two scales of two quality consequences each: (1) the ordinal scale mailbox size with the quality consequences 'small mailbox' and 'large mailbox' and (2) the nominal scale 'access type' with the quality consequences 'local' and 'use at any site with connectivity'. Now, the customer is asked to prioritize the consequences from these scales. For the nominal scale 'access type', our tool presents the consequences 'local' and 'use at any site with connectivity'. Say, the customer gives the score '1' to 'local' and the score '8' to 'use at any site with connectivity', since s/he wants to be able to communicate from anywhere. Next, the customer is asked to assign an importance ranking to the ordinal scale 'mailbox size'. Suppose s/he attaches an importance ranking of '3' to this scale, since the size of a mailbox is of little importance for satisfying his/her need: 'communicate with family abroad'. Finally, by default, the importance ranking '10' is attached to the consequence 'send and receive text' since this is the selected functional consequence.

3.3 Composing Service Bundles

After this customer dialogue, we first match the set of consequences desired by the customer to consequences defined from a supplier's perspective. We can do this because (cf. section 2.2) the concept and theory of consequences provides the bridging connection between the customer and supply sides for the matchmaking 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 for those service bundles that can satisfy all 'must-have' consequences. To find these bundles, we search for: (1) the supplier-specific service properties jointly satisfying the consequence; (2) supplier-specific resources that contain these properties; and finally (3) bundles that contain these resources. Depending on the scale type, we evaluate on the basis of the consequences in hand whether a bundle can satisfy all consequences from nominal scales that are marked as must-haves, and contains at least one consequence from each ordinal scale marked as a must-have.

Case Example. From the previous step, we have the functional consequence 'send and receive text' with importance '10' and four quality consequences: 'small mailbox' and 'large mailbox' from the scale 'mailbox size' (the latter with an importance ranking of 3) and 'local' (importance 1) and 'use at any site with connectivity' (importance 8) from the scale 'access type'. These are matched to all supply- side consequences, see Figure 5. Next, we find the bundles that satisfy the must-have consequence 'send and receive text'. For this, we first find all possible sets of supplier-specific service properties satisfying this consequence. An example in this case could be 'POP3' or 'upspeed= 128 kbps'. These service properties belong to the resources 'email access (KPN)' and 'IP access KPN' which are attached to two service ports of the bundle 'KPN email bundle' (Figure 5). 'KPN email bundle' is therefore satisfying all must-have consequences and hence can be considered further, as is the case with all other bundles shown in this figure since they can all provide the consequence 'send and receive text'.

3.4 Ranking Service Bundles

The next step is to rank the found relevant bundles. For this, we convert the best-to-worse ordinal ranking of consequences to a numerical ranking, by using the Rank-Order Centroid method (ROC, (Barron and Barrett, 1996)). Additionally, 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 and 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. For each service bundle we then obtain service ports additional to those already found by reviewing their service interface. Taking these additional service ports as a starting point we then derive consequences by using the process of finding service ports based upon consequences described before, only then in reverse.

Case Example. To rank the bundles, we calculate a score for the consequences defined on the *ordinal* 'mailbox size' scale. Using ROC, we derive the score 0.75 for the consequence 'large mailbox' and 0.25 for the consequence 'small mailbox'. Now, we calculate a ranking score for each of the possible bundles by using the above multi-attribute formula, and then compute its full set of consequences. For example, in the case of the KPN g-mail bundle, we find two additional service ports: one containing the resource 'fee' with a service property of 'EUR 20', the other containing the resource 'lock-in' with a service property of '12 months'. From this, we derive the consequences 'IP-access fee' and 'lock-in'. With the consequences already found, we finally arrive at the following full set of consequences for the KPN g-mail bundle: send and receive text, large mailbox, access mail at any site with connectivity, IP access fee, lock-in.

3.5 Trade-off Decision Making

Next, we present the found bundles in a ranking to the customer, together with a specification of the consequences received from a bundle and the consequences s/he has to give up to acquire a bundle. Furthermore, the tool shows a specification in terms of a pricing model (cf. (de Miranda, 2006)). The customer 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'

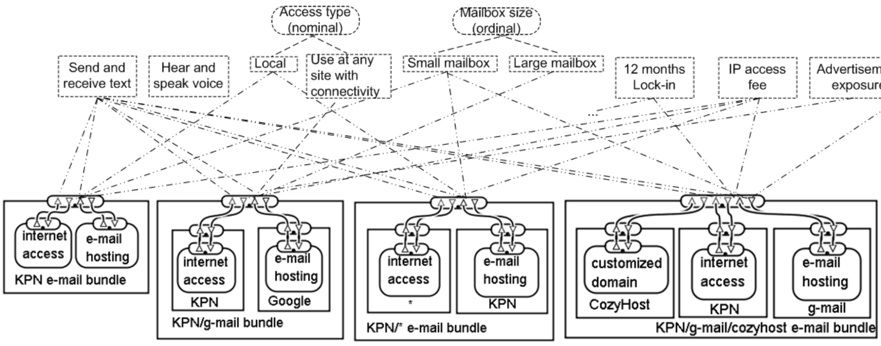

Figure 5: Supplier catalogue for email example.

and change his/her requirements.

For the selected bundle, we consider possible value-enhancing wants for that bundle. As in the step ‘choose consequence’, the customer makes a choice based on a combination of a want (the solution) and a consequence (why the solution is valuable). Then, we iterate again through the steps described above to derive a set of possible service bundles, only now for the value-enhancing wants.

4 CONCLUSIONS

Our general claim and contribution is that it is possible to have automated support that helps bridging the gap between the customer and supplier perspectives on complex service bundles. We have practically demonstrated this for a case of ICT service bundles. Although these two perspectives are fundamentally different, and are therefore associated with differing ontologies and vocabularies, match-making is possible through the introduced marketing theory concept of *consequences* (Gutman and Reynolds, 1988). On this basis, a Propose/Critique/Modify problem-solving method (Motta, 1999) intertwined with a customer-driven dialogue system employing existing decision theory, is helpful in selecting and composing ICT service bundles on the supply side that match expressed general needs on the customer side. More specifically, when ranking service bundles, we use a combination of the compensatory decision technique of (Barron and Barrett, 1996) and the non-compensatory decision technique MoSCoW from DSDM (Stapleton, 1997). We have implemented the presented reasoning processes in our e^3*service* software tool. The services and needs are ontologically described as an RDF dataset that complies with the e^3*service* RDF schema. The reasoning process has been implemented as a Java program using an RDF service catalogue that is the basis of the structured knowledge-driven interactive dialogue with the customer.

REFERENCES

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

Baida, Z. (2006). *Software-aided service bundling*. PhD thesis. VU University Amsterdam.

Barron, F. H. and Barrett, B. E. (1996). Decision quality using ranked attribute weights. *Management Science*, 42(11):1515–1523.

de Kinderen, S. and Gordijn, J. (2008). e^3*service* - an ontological approach for deriving multi-supplier IT-service bundles from consumer needs. In *Proceedings 41st Hawaii International Conference on System Sciences (HICSS-41)*. IEEE Computer Society.

de Miranda, B. (2006). An ontological approach for the use of pricing models to sell services. Technical report. VU University Amsterdam.

Elrod, T., Johnson, R., and White, J. (2004). A new integrated model of noncompensatory and compensatory decision strategies. *Organizational Behavior and Human Decision Processes*, 95(1):1–19.

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

Motta, E. (1999). *Reusable Components for Knowledge Modelling: Case Studies in Parametric Design Problem Solving*. IOS Press. Amsterdam.

Stapleton, J. (1997). *Dynamic Systems Development Method*. Addison Wesley Longman. Reading, MA.