Value-Based Service Bundling: A Customer-Supplier Approach

Iván S. Razo-Zapata
Vrije Universiteit Amsterdam,
The Netherlands,
Email: i.s.razozapata@vu.nl

Pieter De Leenheer
Vrije Universiteit Amsterdam,
The Netherlands,
Collibra nv/sa,
Brussels, Belgium,
Email: p.g.m.de.leenheer@vu.nl

Jaap Gordijn
Vrije Universiteit Amsterdam,
The Netherlands,
Email: j.gordijn@vu.nl

***Abstract*—Due to future ubiquitous service environments, approaches for dynamic delivery of services face new challenges. Since the web 2.0 gave to customers the change to interact with the Internet, they are not any more static entities but can also contribute to the design or co-creation of new services. In this paper we show how customers can help to dynamically bundle services; our approach is based on customer and supplier perspectives. A broker is in charge of matching both perspectives, depicting not only customer desires but also supplier offerings. Moreover, our approach is illustrated by a case study. Finally, we provide some reflections as well as future lines of research.**

***Index Terms*—automatic service bundling; value modelling; business-oriented; service ontologies; educational services;**

I. Introduction

Service industry has been experiencing an enormous growing in the last years, they encapsulate over 70% of USA and Europe economies. Although the evident potential of this sector, there are still some misunderstandings in areas like service bundling, customer targeting and service provisioning [15]. Service bundling briefly states that given a specific customer need, at least one arrangement of services must be (automatically) composed to provide a service-based solution [10]. The assumption here is that due to the wideness of customer needs, it is necessary to combine the functionality of several services to cope with such customer needs.

Even though service networks represent a flexible and dynamic way for service delivering, yet some knowledge gaps exist, specially regarding strategic bundling of services and value (co)creation within the network. Such networks can provide specific service solutions in response to customer needs. Briefly, they include three components: customers, suppliers and enablers [4]. We argue that service bundling mainly involves relationships between the first two components, this is where different strategies might emerge.

In this sense, as a reponse to a given need, service bundling can also be seen as a way to organize(manage) Information Technology (IT) resources that are described by means of services. Moreover, since we aim at analysing the exchange of valuable service outcomes between customers and service suppliers, service bundles can provide new insights to business and IT alignment issues, which are of utmost importance in Enterprise Architecture (EA) [25].

To face these challenges, we model the customer and supplier perspectives. Whereas customers express their needs through a laddering process in which needs are refined into consequences, suppliers offer valuable service outcomes that can provide the desired consequences. In this paper, we focus on automating the bundling of services that fulfil customer needs in a multi-supplier setting.

In earlier work [12] we have proposed the e^3-*value* methodology to design and evaluate Service Value Networks (SVNs). In e^3-*value* , we model the actors part of the SVN, as well as what they exchange of economic value (e.g. service outcomes) and what they require in return for that (e.g. money). So, the e^3-*value* methodology helps in building a conceptual model for SVNs. As such, e^3-*value* is a consultancy instrument that helps to design SVNs and to articulate the business case. Furthermore, we have also demonstrated how, assuming an static customer need, SVNs can be semi-automatically generated with e^3-*value* support [21], [13].

In this paper we aim to bridge the gap in service bundling from a business-oriented perspective. We have removed the assumption of static customer needs and provided an interactive dialogue so customers can express their needs following a reasoning based on marketing theory. In addition, we also provide to service suppliers with capabilities to publish their offerings by means of an ontology-based catalog. Finally, since mass configuration of products is playing an important role nowadays, our long term ultimate goal is to automatically compose a SVN, including the required business processes and IT support in the form of web services. Such IT is then aligned with the business, since both are designed in an integrated way.

The rest of the paper is organized in the following way. Firstly, in Sect. II, we give a definition of the problem to be addressed in the paper. Secondly, in Sect. III we present some related work. Later on, Sect. IV presents our proposed framework for a service bundling. Afterwards, in Sect. V, we introduce a case study and a running example to illustrate our proposed framework. Finally, in Sect. VI, we present our conclusions and future research lines.

II. Problem Definition

Service has become a term loaded with different meanings in different circumstances, mostly depending on who uses it. Different terms that include the word service, e.g., e-services, network services, commercial services etc., are referred to as just services [3], [8]. Therefore we will concentrate only on a set of features that are relevant for our research. In this way we refer to a service as follows: a *service* is an *economic activity* possessing *intangible nature* and producing *valuable consequences* for which customers are willing to *exchange* their *valuable objects* [3], [8].

As an example, consider a customer need to be composed of valuable consequences such as downloading, sharing and uploading books and articles, in this case a service like *access to a digital library* can initially cope with that request. Moreover, this service can ask for an economic exchange to be made at the customer side, which implies an exchange of valuable objects between customer and supplier.

Although service markets can theoretically cope with any customer need, automatic mechanisms providing tailored and customer driven services are far from being a reality. Furthermore, sometimes customer needs can not be covered by a single service but combination of them *i.e.* a service bundle, which can involve one or more suppliers. A service bundle is a package of one or more services, these services can be provided by a single entity or by different enterprises, who each focus on their core competency [8].

Current efforts in service composition, which can be considered as a generic term for service bundling, mainly focus on work-flow issues, leaving completely out business-oriented aspects such as value representation [10]. More important, since current available web services might potentially conform an unlimited array of value-added enterprise applications, a clear picture about how the process of generating service bundles based on business-oriented aspects becomes a key issue [19].

There are several reasons for bundling services. First of all, from the point of view of suppliers, when they work together offering their benefits, the chances of covering complex customer needs are higher. In addition they can also reuse resources from other suppliers, which mainly saves costs of (re)implementation. For instance, a shop can reuse a paying service offered by another supplier. Secondly, from the customer point of view, service bundles sometimes are cheaper than getting services independently. Furthermore, customers have also access to more tailored services for covering their needs. As an example, if a user is really concerned about making backups of her/his files, (s)he can acquire an *on-line backup* service on top of *access to a digital library* service. Furthermore, bundling can improve service diversification, which is a key requirement in service industries where highly competitive markets can be found [18].

Dealing with highly variable, and sometimes vague, customer needs demands more than simple bundling strategies. Besides establishing relationships between service suppliers and final customers, such mechanisms must also establish relationships among the different suppliers, *i.e.* suppliers can jointly work to offer better options to the final customer. While the first ones are called Business to Customer (B2C) relationships, the latest are called called Business to Business (B2B) relationships. Even though, at this point we are more interested in facing the issue of dynamically establishing B2C relationships in a multi-supplier setting, we have already done some preliminary steps for solving B2B issues [13].

In the end, we aim at an automated bundling framework where customers and suppliers dynamically establish relationships with each other. To achieve this goal, we propose the use of customer and supplier ontologies where concepts at both sides can be automatically matched making possible the establishment of B2C relationships. Whereas the customer ontology expresses needs, the supplier ontology describes valuable offerings to match those needs.

III. Related Work

So far, the web service community has recognized several features in approaches aiming at automatic service composition. Dustdar and Schreiner [10], present a survey describing the main trends in this field, which are: *1) Static vs Dynamic composition*: Services can be composed either at design-time or at run-time. *2) Model driven composition*: service composition is based on either a meta-model or business rules. *3) Declarative service*: services are composed by description languages, which can be used to represent constraints, states among other properties. *4) Automated vs manual composition*: ontologies and XML-based representations are key tools for achieving automating composition of web services. *5) Context based (Discovery and Composition)*: services are differentiated depending their context, *i.e.* services can be available through different channels using different devices such as PCs, palmtops, cell-phones among others.

Moreover, according to our objectives, other features must be taken into account. *6) Business Orientation*: refers to whether the approach is based on economic relationships rather than on work flow decisions. It has been discussed in previous work why business modelling is different than process modelling [11]. Briefly, business models are centred around the notion of value, therefore it is relevant to determine who is offering what of value to whom and what expects of value in return. *7) Customer influence*: determines if the approach takes into account customer preferences in order to create the service bundles. *8) Visualization of bundles*: visualization of economical relationships has been recognized as a good way to overcome cognitive limitations and make structure, patterns and relationships apparent from the underlying data [5]. In our case it might lead to designing business rules, identifying new niches for suppliers among other findings.

In this section we briefly describe some approaches aiming at service composition. Making use of the previously described features, we provide descriptions for those approaches highlighting advantages and disadvantages. We start with approaches that have been inspired by the e^3-*value* framework [12]. Later on, we describe some semantic web approaches.

A. e^3-value *inspired*

Baida [3], proposes the Serviguration algorithm for bundling services, which relies on business dependencies to combine services. Nevertheless, the generated bundles do not always depict the concept of economic reciprocity, *i.e.* what they expect in return for an object of value delivered, which is important to understand the value exchanges within a SVN.

Nakamura *et al.* [20], introduce a framework based not only on value models but also on a value-meta model and an architecture of value-added service broker for dynamically composing services - Value-Based Composition (VBC). Nevertheless, due to the lack of a well detailed example, it seems to be still a work in progress. Consequently, this approach just provide ideas about how the bundling process might be automated in a dynamic environment.

GVP-Patterns [26], is an approach based not only on value patterns but also on process and goal patterns. A pattern is defined as a solution fragment that is recurrently observed in service bundles. At design-time, the goal patterns are linked with value and process patterns. Consequently, once a desired goal is defined, a goal pattern can generate a service bundle based on its associated value and process patterns. Unfortunately, GVP-Patterns requires a lot of hand-made tasks, therefore it is not suitable for automation.

e^3*service* [8] provides a framework for matching customer needs with service bundles offerings. The framework assumes service bundles have been already created at design-time. In this way, e^3*service* mainly automates the process for finding service bundles based on customer needs.

B. Semantic Web and Generic approaches

Traverso and Pistore, [23], propose a framework for service composition, which can be considered as a more generic term for service bundling [1]. The framework generates BPEL4WS plans based on a Model Based Planner (MBP). Even though the approach generates plans that can be delivered through services, it does not say anything about the interactions among customers and service suppliers.

Agarwal *et al.,* [1], propose the OntoMat-Service approach which generates plans for executing services based on customer choices. OntoMat-Service mainly offers a semi-automatic guideline for composing services by making use of user's intelligence. Consequently, it assumes a customer has a good understanding about the services s/he needs.

METEOR-S [22], is a composition framework based on Semantic Process Templates (SPTs). The idea is that a skilled designer can come up with a SPT for a desired service. Based on the STP, the required services are discovered and added to the data flow according to the required activities. Finally, an executable process is generated, validated, deployed and ready for invocation. As can be observed, METEOR-S is a static approach in which a designer builds a service that can be used for customers.

DynamiCoS [6], provides a framework for service composition. Services are composed at run-time following some customer requirements. One of the main drawbacks in this approach is the lack of a visual representation for the composed services as well as assuming that customers always have a clear idea about the services they need.

u-service [17], is another dynamic framework that bundles services based on customer context. Although u-service allows continuous interaction with customers, the bundling algorithm mainly deals with QoS aspects, ignoring the aspects about value exchanges. In addition, the bundling algorithm seems to work for a small number of services.

On the one hand, many e^3-*value* inspired approaches are considering business aspects for bundling services. However, these approaches have a more static flavour. On the other hand, many of the semantic approaches focus on process-oriented issues, that is why the composition process is usually modelled as a planning problem. Furthermore, the semantic approaches also lack from a visual representation for the bundled services.

Finally, we emphasize our interest in dynamic and business-oriented approaches allowing to customers and suppliers co-create service bundles as well as providing visual representation of economic relationships among them.

IV. Proposed Framework

A. Basic Concepts

An e^3-*value* model depicts a network of enterprises creating, distributing, and consuming objects of economic value [12]. The focus of the model is on **what** kind of objects enterprises must exchange to each other in order to cover customer needs [2]. Fig. 1 shows (at the bottom) the modelling constructs of e^3-*value* and (on top) a basic example.

The most important e^3-*value* constructs are as follows: *Actors*, such as a buyer and seller, are economically independent entities (Fig. 1). Actors transfer *value objects* (money, goods) by means of *value transfers*, which in turn connect *value ports*. For value objects, some actor should be willing to pay, which is shown by a *value interface*. A value interface models the *principle of economic reciprocity*: Only if you pay, you can obtain the goods and vice versa. Besides, actors perform *value activities*, which create something of economic value. An elaborated formalisation of e^3-*value* can be found in [12].

Figure 2 depicts our view about a business-oriented framework that allows customer and supplier interaction for dynamically bundling services. Such a framework finds solutions in the clash between two perspectives: the *customer perspective* and the *supplier perspective*. As can be observed, in order

[1]Service bundling is a more business oriented term while service composition is used in a more technical context

[2]and not in HOW, this is the focus of a business process

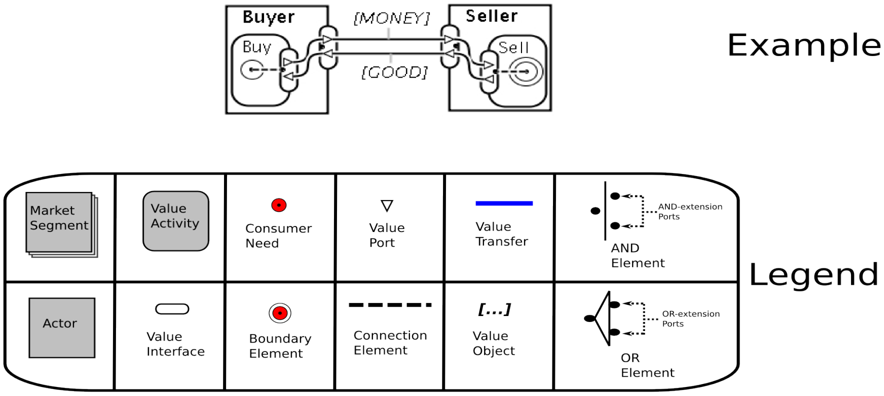

Fig. 1. *e^3-value* constructs for value modelling.

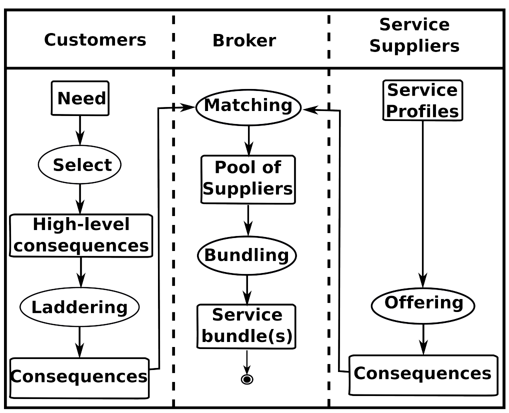

Fig. 2. Framework for Dynamic Service Bundling.

to cover the customer perspective *i.e.* a customer need, a broker matches this perspective with the supplier perspective. Later on, inside the broker an automatic composition must be started for matching and bundling services. Finally, the set of service bundles can be offered to the customer. As a basis we adopt *Serviguration* [3] and e^3*service* [7]. Both are ontology-based approaches that have been addressing the issue of service bundling based on customer and supplier perspectives. *Serviguration* introduces the notion of service dependencies that constrain service bundling while e^3*service* presents a customer-driven approach for matching a customer need with existing service bundles. In this work, we merge elements from both approaches with the e^3*-value* methodology, the latter offering several advantages such as a graphical modelling tool and analysis of sustainability [12]. The ontology merging is out of the scope of this paper. We highlight selected concepts in terms of examples.

B. Supplier Perspective

Fig. 3 depicts the result of aligning the e^3*service* supplier ontology with the e^3*-value* ontology. We only describe what is relevant for defining supplier profiles. Service suppliers are *actors* performing activities (*value activities*) to produce service outcomes (*value objects*) which can be offered to customers. In addition, a value object has *functional consequences* and *quality consequences* as a result of its use or consumption. Finally, a *service bundle* consists of *service elements* which are special types of value activities. For a full elaboration of these concepts we refer to [12] and [7].

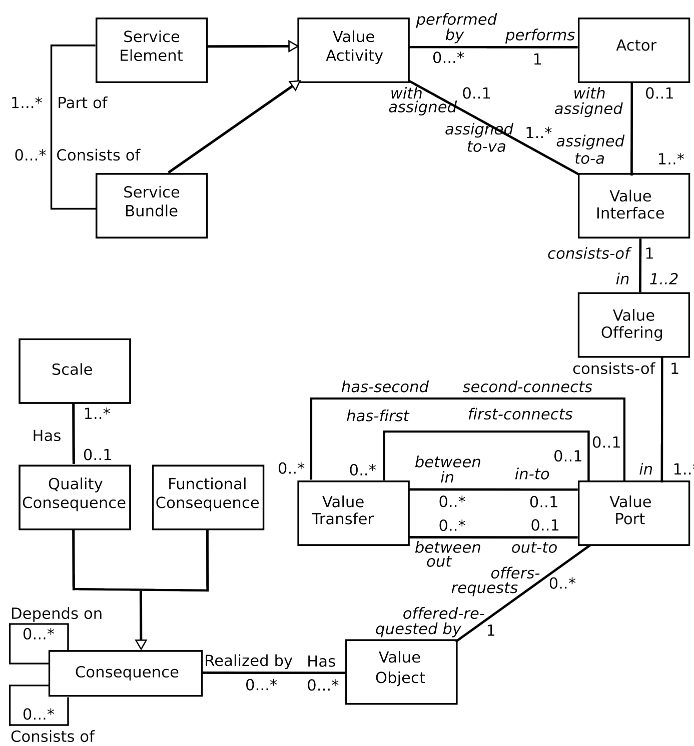

Fig. 3. The e^3*service* [7] supplier perspective ontology aligned to the e^3-*value* [12] ontology.

C. Customer Perspective

Figure 4 shows a UML rendering of the customer perspective. It is based on concepts from *established* customer needs literature [7].

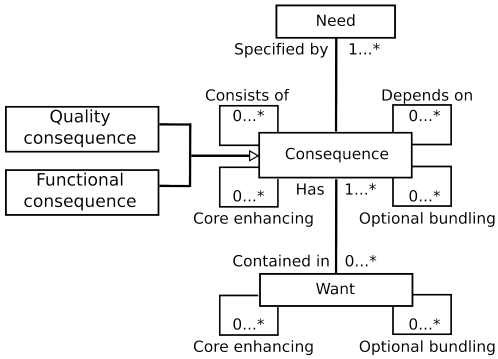

Fig. 4. The e^3*service* customer perspective ontology, based on [7].

- A `Need` represents a problem statement or goal, independently from a solution direction (see[2], [7]). *E.g.,* a *job candidate* has a need to fulfil a certain *job profile*.
- A `Consequence` is anything that results from consuming (a combination of) valuable service propertie(s) [14], [7]. We distinguish between two types.
  - `Functional Consequence` represents the functional goal that can be achieved through consumption of a service that has a certain valuable property [7]. *E.g.,* a `Functional Consequence` from the need to fulfil a certain `job profile` are all the involved *qualifications* or *competencies* that are required to perform such job. We devise four types of relationships with different semantics between consequences as shown in Fig. 4. *E.g.,* the consequence "read and write" consists of the consequences "read" and "write"; or the consequence "writing english" is core-enhancing for "programming", etc..
  - `Quality Consequence` A quality consequence expresses qualitative properties of other, *e.g.,* `Functional Consequences` in customer terminology [7]. Because it expresses the qualitative properties of another `Consequence`, a `Quality Consequence` cannot be acquired separately: It always depends on (a relation between `Consequences`) another type of `Consequence` . *E.g.,* `Writing English` has a required proficiency as a `Quality Consequence`.
- A `Want` is a specific, supplier-independent solution that is commercially feasible to be provisioned on its own [7]. 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 typically be found in service taxonomies such as UNSPC [3]. An example of a `Want` could be learning object for which multiple institutes are accredited to deliver them. Again, different types of relationship may hold: *e.g.,* all *learning objects* in a Bachelor's programme are core supporting for any learning objects in the Master's.

[3]United Nations Standard Products and Services Code

- A `Scale` groups `Quality Consequences` of the same type [7].

D. Combining perspectives

To *compose* a service bundle according to customer-expressed needs, a broker has to perform three steps: laddering, matching and bundling.

Algorithm 1 Bundling algorithm

```
procedure BUNDLING(r, p) ▷ r = required consequences, p = matching pool
    solutions ← {∅}
    SC ← {∅}
    for each rq ∈ r do                                  ▷ rq required consequence
        SC = suppliers in p providing rq
        if SC == ∅ then
            return solutions                            ▷ There is no solution
        else if solutions == ∅ then
            for each service ser ∈ SC do
                create a new (empty) service bundle sb
                sb.add(ser)
                solutions.add(sb)
            end for
        else
            for each sb ∈ solutions do
                copy sb as many times as SC.size()
                add these copies to solutions
                add a different ser from SC to each copy
                remove the original sb from solutions
            end for
        end if
    end for
    return solutions
end procedure
```

Laddering: is a marketing practice which uses a conceptual map to represent how a customer links specific product attributes to high-level values [7]. In our case we apply it to link service consequences to customer needs via high-level consequences (Fig. 4).

Matching: Matching determines a *matching pool* of service suppliers that plausibly provide part of the required consequences. Due to the variability of customer needs, single suppliers rarely provide all the required consequences on their own. Consequences are the key components for matching the two perspectives. For each functional consequence at the customer side, this matching process performs a comparison with all the functional consequences expressed at the supplier side and retrieves the services that offer the required consequence.

Bundling: Bundling finds combinations of suppliers in the matching pool that collectively cover the required consequences. Different principles or policies of interactions are key during bundling as they may constrain the possible collaboration between suppliers. At this stage, we have applied a modified version of Baida's algorithm [3]. One of the main differences is that, thanks to laddering and matching, our bundling algorithm is now suitable for a dynamic environment. Whereas the original Baida's algorithm bundles services according to the value objects they offer, we bundle services depending on the consequences they provide. Algorithm 1 depicts how such bundling is performed. As can be observed this algorithm generates service bundles by combining suppliers according to the consequences they offer.

Although it performs an exhaustive search, its performance under a is acceptable to make clear our point about how dynamic service bundling can be achieved through customer and supplier interaction.

To sum up our bundling process allows interaction between customers and suppliers. While suppliers offer their services by means of consequences, the customers ladder their high-level needs into specific consequences. Afterwards, a matching process retrieves the pool of all the service suppliers that can provide the required consequences. Later on, this pool is passed to the bundling algorithm, which generate alternative bundles by combining service suppliers according to consequences.

V. Case Study and Running Example

A. Basic Concepts

The European employment market is characterised by a contradictory situation: a very large number of candidates fail to find a job; and many employers are unsuccessful in locating appropriate candidates for their vocations. Given a Vocational Competency Ontology (VCO) (collaboratively developed in previous work [9]), skill gap analysis can overcome the semantic mismatch between candidate and market profiles, and capture a candidate's missing competencies. Stakeholders include educational institutes, public employment organisations, and industry partners from different European countries.

Assuming that candidate's needs has been already identified during skill gap analysis, our next step is automating the bundling of educational services in a multi-supplier setting *i.e.* the educational e-service web, which acts on publicly available instance data about related needs and services we found on the Web. This also illustrates that the necessary data is indeed available for a service web to emerge.

Central driver driving the evolution of the educational e-service web are the enterprise ecosystems. Once set out the goals and strategy of the company, different supporting business processes are lined out, involving the creation of (new) functions and tasks. Each of them require human performance, which in turn require certain competencies. From this feedback loop, relations between `Functions` and `Competencies` emerge. To describe competencies, there is a widely used HR-XML-standard called reusable competency definitions (RCDs). RCDs are yet underspecified so they retain their generative character.

To describe the relationship between functions and competencies enterprises define function profiles, which usually contain the following four essential parts [4]:

1) a competency map[5] (or tree) that references RCDs. *E.g.,* competency A is a parent of sibling competencies X, Y, and Z. Hence X, Y, and Z are needed to acquire A. From our experience [9], we learned that `Functions` tree roots are decomposed into `Job Tasks`, which are further decomposed into `Activities`, which on their turn are decomposed into `Partial Tasks`.
2) a set of RCDs for each of the involved `Partial Tasks`, the leafs of the competency tree;
3) proficiency levels for `Competencies` (default +1) (IEEE score or SCORM uses an interval of -1 to +1)
4) weight on edges in the competency map. *E.g.,* a weight of less than 0.5 between `Activity` X and `Job Tasks` A means mastering X is an optional requirement to acquire competencies to master A.

E.g., in the automotive industry functions are categorised along the car manufacturing process: going from press shop, to the body shop and paint shop, to finally end at the assembly shop[6]. In order to perform each of those functions, human operators with specialized competencies are required.

The candidate's search is equally driven by populating its CV by RCDs he collects through experience and education. If its current CV shows gaps to fulfil a certain function profile, a need emerges that has to be answered by the service web. The final stakeholder's cycle are the educational service suppliers. They monitor needs and function profiles and define services accordingly.

Currently the stakeholder community is simplified for the sake of illustration. In reality there are additional parties that are responsible for identifying large gaps in the candidate pools and predicting future needs in education, and finally organise this education. The main point we want to make here is twofold. On the one hand, all these educational parties can act independently and by doing so converge towards each other as long as their information (about functions, competencies and education service offerings) is published using (as shown below) open (IEEE and HR-XML) standards like RCD, SCORM, and LOM. On the other hand, candidates looking to fulfil some gaps by means of competencies can actually surf this service web to find service bundles providing the required competencies.

B. Dynamically Bundling Services

1) Supplier Perspective: For the demonstration, we surveyed a number of publicly available competency databases and picked out the National Database of Accredited Qualifications[7] (NDAQ), which contains details of Recognised Awarding Organisations and Regulated Qualifications in England, Wales and Northern Ireland.

[4] see *e.g.,* http://www.ostyn.com/standardswork/competency/ReusableCompMapProp.pdf

[5] Just like with RCDs, using the proper ontology governance methods, this may lead to standardisation of reusable competency maps (RCMs). See note on this here: http://www.ostyn.com/standardswork/competency

[6] cf. http://www.nedcar.nl/content/view/44/49/lang,en/

[7] http://www.accreditedqualifications.org.uk

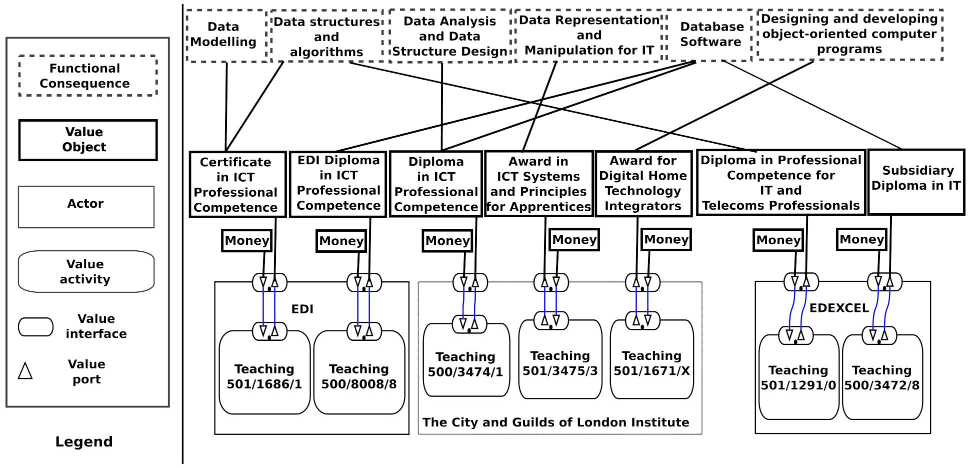

Fig. 5. Visual representation of the service catalog generated from Web data in NDAQ.

We harvested the NDAQ database and generated a catalog of service suppliers, of which an excerpt is depicted in Fig. 5 and annotated using the concepts in the supplier ontology (Fig. 3). Actors are educational institutes performing teaching activities, *i.e.* value activities [8]. These value activities offer several courses in form of value objects. Finally, value objects have functional consequences which we consider to be competencies (RCDs). *E.g.,* `The City and Guilds of London Institute` performs the value activity `Teaching 500/3474/1`, which produces the value object `Diploma in ICT Professional Competence` with the associated functional consequences (RCDs) `Data Analysis and Data Structure Design` and `Database Software`.

Furthermore, we have implemented a prototype in which we represent both catalogues by means of RDF files. The reasoning process is applied with Jena [9] a semantic framework for Java. At the end of the process we bring about a RDF file that can be visualized with the e^3*-value* editor.

2) Customer Perspective: We have also designed a customer catalog (Fig. 6) based on the NDAQ database, by grouping consequences according to possible courses in which they can be offered. Later on, these consequences are linked to customer needs via high-level consequences.

3) Laddering: Customers express their needs using an interactive dialogue system in which they can recursively refine vague needs in terms of functional consequences. To illustrate the laddering, we make use of the customer catalog in Fig. 6. As can be observed, the high-level customer need `How can I improve my programming skills?` is refined by two optional consequences: `Web Applications Development` and `Data Analysis and Design`. Recursively, these consequences are refined further. *E.g.,* if the consequence `Data Analysis and Design` is selected to cover the customer need, the laddering will determine that the functional consequence `Data structures and algorithms` is more concrete to fulfil such need, since the first one *consists of* the second one.

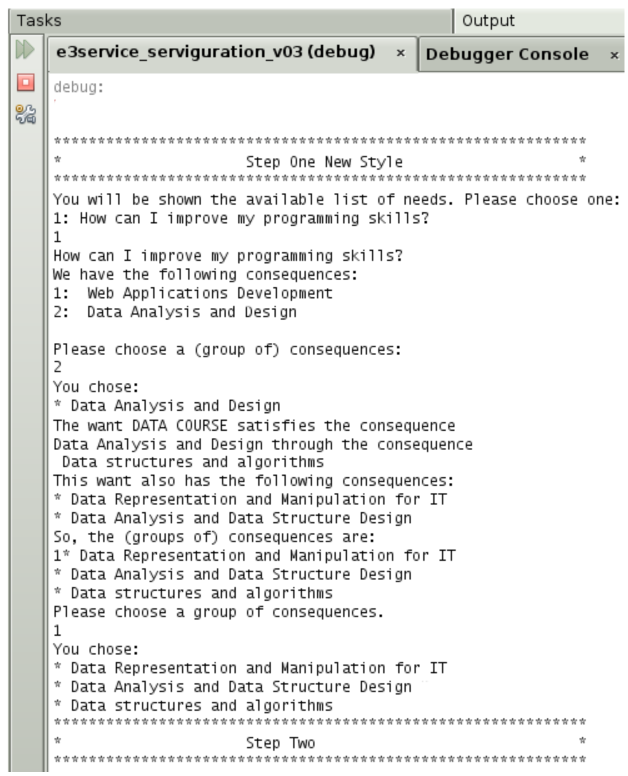

Fig. 7. Illustration of the Laddering process.

The next step involves discovering more consequences through the notion of wants explained earlier. In Fig. 6, the functional consequence `Data structures and algorithms` is contained in the `Data Course` want. By exploring this want, the functional consequences `Data Analysis and Data Structure Design` and `Data Representation and Manipulation for IT` are discovered. Fig. 7 depicts how the process was performed in our prototype.

[8]We assume that teaching is one type of educational service.
[9]http://jena.sourceforge.net/

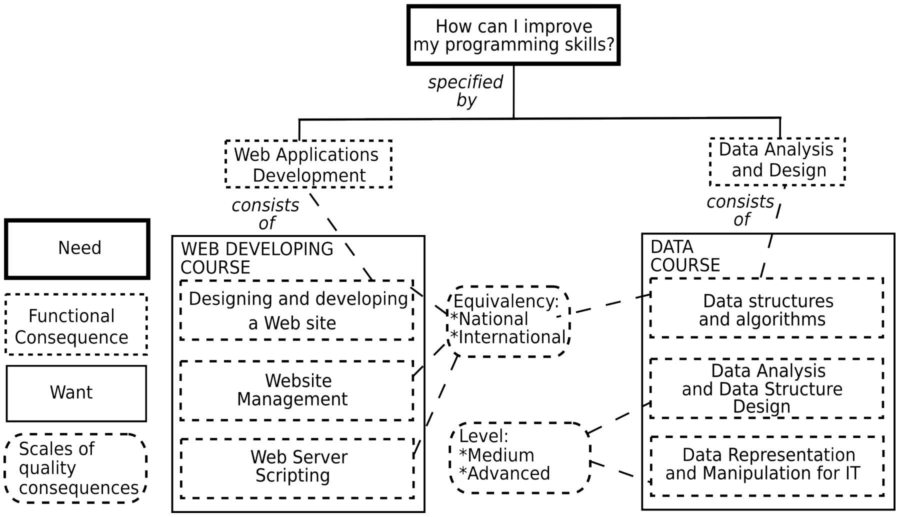

Fig. 6. Customer catalog

4) Matching: Once the required consequences have been specified in the customer side, the matching process retrieves all the possible service suppliers that offer the required consequences. The first part of this process is depicted in Fig. 8. As already explained, the next part and core of this process is a (semantic) comparison between functional consequences at both catalogues (supplier and customer).

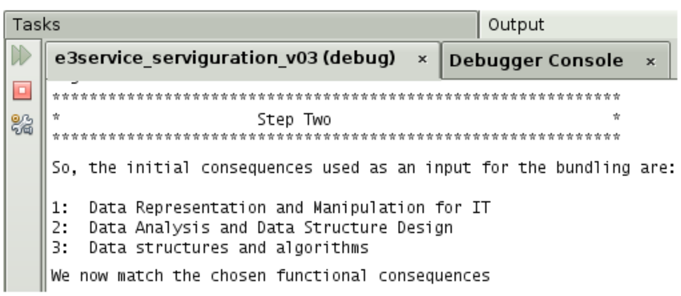

Fig. 8. Matching consequences.

5) Bundling: Fig. 9 depicts two of the alternative service bundles providing the required functional consequences for the customer need `How can I improve my programming skills?`. Since we have not implemented a selection process yet, the depicted bundles were selected by hand. As can be observed, our bundles include different suppliers offering different services, through a common interface which can be later offered to the final customer. Moreover, unlike *Serviguration* bundles, our bundles are represented according to the e^3*-value* ontology.

The bundles in Fig. 9 depict how service suppliers can interact together to provide service bundles that have been designed by interacting with a customer. More over, we can also observe that the supplier `The City and Guilds of London Institute` could communicate to the broker for which bundle is more willing to work. It can be done by exploring profitability analysis which is already supported by the e^3*-value* editor. For now, this option has not been implemented in our prototype, but will be considered as future work.

VI. Conclusions and Future Work

We have presented a framework for bundling services based on customer and supplier interaction. Such framework relies on a broker who performs three activities laddering, matching and bundling. While laddering allows transforming vague customer needs into concrete functional desires (consequences), matching and bundling dynamically bundle services for covering such functional desires. More important, we have also proposed how customers can work together with suppliers to co-create new valuable services.

Furthermore, to achieve a better understanding of the market forces driving service composition, we must focus on modelling the socio-economic content of services, *i.e.* the valuable outcome of services rather than only its interface specification. The e^3*-value* approach provides a theoretically grounded basis for this, as well as tools for modelling and analysing service bundles. Moreover, as illustrated by the case, pushed by social and knowledge connectivity, many sectors have available standards (like IEEE RCD/RCM, SCORM, and LOM) that provide an extension to this grounding and allow to disclose and link domain-specific knowledge as we demonstrated with NDAQ database.

Although at this stage we have described how service bundles can be composed to match customer needs (B2C relationships), a process for solving B2B dependencies is also needed. So far, the service bundles depict only the services that are needed for dealing with a specific customer need. Nevertheless, the services within the bundles might depend on additional services *i.e.* B2B relationships have to be solved. For instance, the educational services in Figure 9 might depend on pure technical services such as on-line payment or

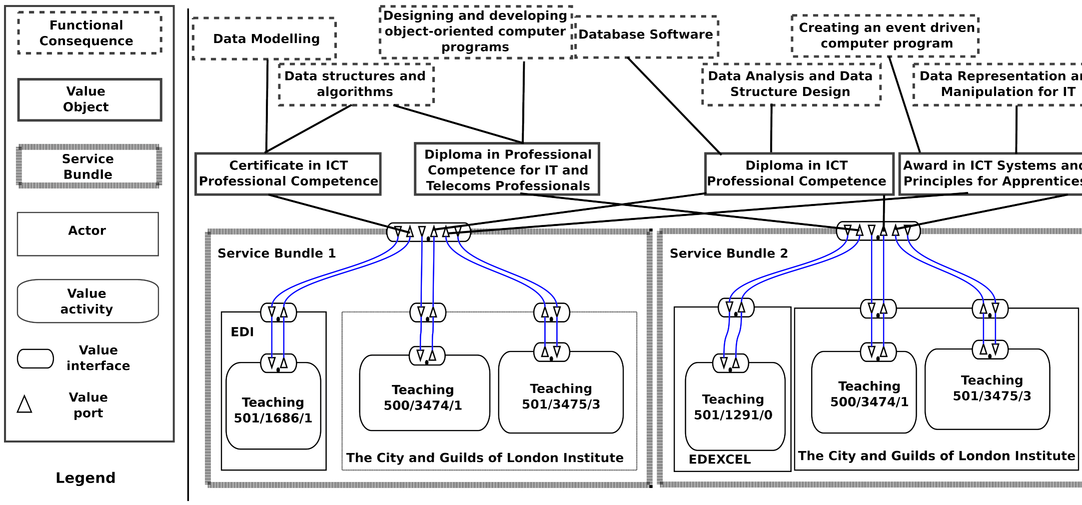

Fig. 9. Service bundles

internet connectivity, but also more strategic services like 24x7 helpdesk or blackboard facilities. These kind of dependencies have been already observed in several service sectors such as the health-care sector. In such a case, even though the customers only deal with hospitals or medical centres (main health-care suppliers), additional services like wholesalers, pharmacies and medical equipment providers are also needed to support the actioning of these main suppliers [4].

Moreover, once services have found each other in a matching pool, the freedom of interaction between services can be restricted by "rules of engagement". Interaction principles provide starting points for interaction, while rules/policies dictate different implementations of these principles depending on the business context. Constraints on the possible interactions limit the space of plausible service bundles.

Principles and policies can be enforced in different ways. For instance, by a governing authority to which a service is legally bound or a context provider that sets the legal business context in which the interaction takes place. In previous work we have explored how bundling can also be guided by *value skeletons* that are considered as visual representations of interaction principles and rules that apply for a specific sector. In [21], we illustrate this for IPR clearing in music industry.

In more recent work we have given the value skeletons in the configuration process the more humble role of *design pattern*. In [13], a skeleton provides the groundwork to develop service profiles, which specify not only their offerings but also relations with other services and information about service instances. Consequently, if designed according to the same value skeleton, all service profiles can be connected with each other. Moreover, service profiles are designed as building blocks that can be used to generate alternative bundles in a decentralized and semi-automatic way across sectors.

As a result of the generation of alternative service bundles, final customers must be able to select a bundle according to their preferences. In this sense, given a set of alternative bundles, a mechanism for scoring and selecting the best bundles has to be specified. Such mechanism must allow customers to *critique* alternative bundles by providing preferences about the desired/disliked properties or quality requirements. Moreover, customers might be also interested in *selecting* other consequences related to their needs. Currently, e^3*service* emulates such behaviour through a text-based B2C dialogue between a customer and suppliers in which a customer scores each alternative bundle by expressing whether its quality consequences are desirable or not.

In the same way, because the Web is continuously growing, more suppliers are able to offer services not only to customers but also to other suppliers. As a result of this B2B interaction, to *evolve* on the fly, customers and suppliers must consider marketing aspects such as value (co)-creation and cost sharing [16]. Furthermore, the perception of the customer community orchestrates the dynamics of the service bundles. We assume that the social Web already provides a plethora of interactive tools to codify this perception including bookmarking, tagging, blogging, and wikis, being developed and shared at little or no cost. Key problem that remains open, however, is the ability for customers to collectively decompose these facts in an interactive way and codify them in new needs or shifts in needs.

As future work we plan to address the uncovered processes in our framework, namely *B2B interaction* and *selection* of service bundles. To face each one of these issues we already have some guidelines such as the e^3*service* methodology and results from previous work [13], [7]. The first issue, *B2B interaction*, involves the process of solving inter&intra dependencies for each service bundle. This process produces service networks, which are the input for the selection process. In the end, if covering all their required consequences, customers can *select* one service bundle to be provided. Otherwise, customers have to *modify* their preferences (re)starting the bundling process, for which we have to consider also feedback from customers.

Moreover, because e^3-*value* models describe the required value exchanges to be performed, by representing service bundles with e^3-*value* concepts we aim not only at modelling business-relevant aspects but also at giving an input for specifying duties among suppliers, *e.g.*, value models can be translated into Service Level Agreements (SLAs). Wegman *et al.*, [24], have already explored some mechanisms for specifying SLAs based on what they call Supplier/Adopter Relationships (SAR). According to the authors, the SAR underlying principle is that suppliers are responsible for delivering features that bring value to the adopters. The VALUE-IT project currently addresses the issue of transforming e^3-*value* models into web services where ideas about defining SLAs have to be explored [10].

Even more, we forecast two main trends for achieving a business-oriented service bundling. A completely centralized approach assumes the existence of a broker, which carry on the whole process. One of the advantages of such approach is that the broker can be responsible not only for the bundling but also for the functioning and cost of the provided services. Furthermore, the broker has all the capabilities to adapt the services in case of changes coming either from the customer or the supplier side.

On the contrary, a full decentralized approach delegates responsibilities among all the participants. Nevertheless, such assumption brings about several issues like single services taking care of their own dependencies, trust and the certainty of providing complete solutions.

Despite these issues, decentralization also provides some benefits. First of all, monopolistic behaviour is avoided since services are individually operated by actors who voluntarily join and leave the environment. Secondly, services can be offered to lower prices by sharing resources. Moreover, it can provide failure resilience and self-organization in case of changes in the environment. Although these trends are opposite poles for service composition, a mixture of centralized approaches together with a distributed solutions might provide highly scalable and dependable service systems. Consequently, exploring this alternative is also part of our future research work.

To conclude, in our view, service composition must deal with dynamic B2C and B2B relationships. In this sense, service bundling mainly deals with B2C relationships, delegating to a second step the responsibility for solving B2B dependencies.

Acknowledgements. This paper has been partially funded by the NWO/Jacquard project VALUE-IT no. 630.001.205

References

[1] Sudhir Agarwal, Siegfried Handschuh, and Steffen Staab. Annotation, composition and invocation of semantic web services. *Journal of Web Semantics*, 33:1–24, 2004.

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

[3] Ziv Baida. *Software-aided service bundling*. PhD thesis, Free University Amsterdam, 2006.

[4] R. C. Basole and W. B. Rouse. Complexity of service value networks: conceptualization and empirical investigation. *IBM Syst. J.*, 47:53–70, January 2008.

[5] Rahul C Basole. Visualization of interfirm relations in a converging mobile ecosystem. *Journal of Information Technology*, 24(2):144159, 2009.

[6] Eduardo Gonçalves da Silva, Luís Ferreira Pires, and Marten van Sinderen. Towards runtime discovery, selection and composition of semantic services. *Comput. Commun.*, 34:159–168, February 2011.

[7] Sybren de Kinderen. *Needs-driven service bundling in a multi-supplier setting: The computational e3service approach.* PhD thesis, Vrije Universiteit Amsterdam, 2010.

[8] Sybren de Kinderen, Jaap Gordijn, and Hans Akkermans. Reasoning about customer needs in multi-supplier ict service bundles using decision models. In *11th International Conference on Enterprise Information Systems*, 2009.

[9] P. De Leenheer, S. Christiaens, and R. Meersman. Business semantics management: a case study for competency-centric HRM. *Computers In Industry*, 61(8):760–775, 2010.

[10] Schahram Dustdar and Wolfgang Schreiner. A survey on web services composition. *Int. J. Web Grid Serv.*, 1(1):1–30, 2005.

[11] Jaap Gordijn, Hans Akkermans, and Hans van Vliet. Business modelling is not process modelling. In *Proceedings of the Workshops on Conceptual Modeling Approaches for E-Business and The World Wide Web and Conceptual Modeling: Conceptual Modeling for E-Business and the Web*, ER '00, pages 40–51, London, UK, 2000. Springer-Verlag.

[12] Jaap Gordijn and J.M. Akkermans. e3-value: Design and evaluation of e-business models. *IEEE Intelligent Systems*, page 1117, 2001.

[13] Jaap Gordijn, Pieter Leenheer De, and Ivan S. Razo-Zapata. Generating service valuewebs by hierarchical configuration: An ipr case. In *HICSS 44*, 2011.

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

[15] John Hamilton. Service value networks: Value, performance and strategy for the services industry. *Journal of Systems Science and Systems Engineering*, 13(4):469–489, 2004.

[16] Shun Yin Lam, Venkatesh Shankar, M. Krishna Erramilli, and Bvsan Murthy. Customer value, satisfaction, loyalty, and switching costs: An illustration from a business-to-business service context. *Academy of Marketing Science*, 32:293–311, 2004.

[17] Nam Yeon Lee and Ohbyung Kwon. A complementary ubiquitous service bundling method using service complementarity index. *Expert Syst. Appl.*, 38:5727–5736, May 2011.

[18] Jose M Barrutia Legarreta and Carmen Echebarria Miguel. Collaborative relationship bundling: a new angle on services marketing. *International Journal of Service Industry Management*, 15:264–283(20), 1 March 2004.

[19] Man Sze Li. *Towards the Future Internet - Emerging Trends from European Research*, chapter The Economics of Utility Services in the Future Internet, pages 31–40. IOS Press, 2010.

[20] Kazuto Nakamura and Mikio Aoyama. Value-based dynamic composition of web services. *Asia-Pacific Software Engineering Conference*, 0:139–146, 2006.

[21] Ivan S. Razo-Zapata, Ania Chmielowiec, Jaap Gordijn, Maarten Van Steen, and Pieter De Leenheer. Generating value models using skeletal design techniques. In *5th international BUSITAL workshop*, 2010.

[22] Kaarthik Sivashanmugam, John A. Miller, Amit P. Sheth, and Kunal Verma. Framework for semantic web process composition. *Int. J. Electron. Commerce*, 9:71–106, January 2005.

[23] P. Traverso and M. Pistore. Automated composition of semantic web services into executable processes. pages 380–394. Springer-Verlag, 2004.

[24] Alain Wegmann, Gil Regev, Georges-Antoine Garret, and Francois Marechal. Specifying Services for ITIL Service Management. In *The International Workshop on Service-Oriented Computing Consequences for Engineering Requirements (SOCCER'08) in the 16th IEEE International Requirements Engineering Conference (RE08)*, 2008.

[25] Alain Wegmann, Gil Regev, Irina Rychkova, Lam-Son Lê, José Diego De La Cruz, and Philippe Julia. Business-IT Alignment with SEAM for Enterprise Architecture. In *The 11th IEEE International EDOC Conference (EDOC 2007)*. IEEE, 2007.

[26] Zlatko Zlatev. *Goal-oriented design of value and process models from patterns*. PhD thesis, University of Twente, 2007.

[10] http://www.e3value.com/projects/ourprojects/value-it/