Dynamic Cluster-based Service Bundling: A value-oriented framework

Iván S. Razo-Zapata, Jaap Gordijn, Pieter De Leenheer and Hans Akkermans
VU University Amsterdam
De Boelelaan 1081
1081 HV, Amsterdam, The Netherlands
$\{$*i.s.razozapata,j.gordijn,pieter.de.leenheer*$\}$@*vu.nl*

***Abstract*—In this paper we present a framework for dynamic service bundling, which focuses on the exchange of valuable outcomes between customers and service suppliers. The approach is based on three components: A customer, a broker and a pool of suppliers. The broker is in charge of matching the customer and supplier perspectives and performing a cluster-based bundling process. The applicability of our approach is proven by means of a case study in the educational service industry.**

***Keywords*-service bundling, value orientation, dynamic framework, clustering.**

I. INTRODUCTION

Although the service industry has been widely recognized as a strategic point in the future internet-based economy, issues such as service *bundling* still present knowledge gaps [14]. Briefly, service bundling is the process of combining services to provide a suitable service-based solution for a given need [2]. In general terms, it is needed to analyse the kind of required reasoning to allow this bundling. More specifically, it is important to understand how customers and suppliers can interact with each other to dynamically co-create valuable service bundles.

Based on the concept of *economic reciprocity*, we consider a service as an economic activity that offers and requests valuable outcomes to and from its environment [8]. Moreover, we are interested in the bundling of commercial services that can be provided and consumed by using information and communication technology (ICT). For instance, Netflix and Spotify are common examples of commercial ICT services, they offer valuable outcomes in exchange for a fee [1]. In the rest of the paper when mentioning services, we actually refer to ICT commercial services, unless something else is specified.

The advantages of service bundling can be perceived by customers as well as suppliers. Firstly, from the customer point of view, service bundles are generally cheaper and more suitable for their needs. Secondly, from the point of view of suppliers, when they work together offering their outcomes, the chances of covering complex customer needs are higher. Furthermore, they can also reuse services from other suppliers, which not only saves costs of (re)implementation but also allows to focus on and improve their own service outcomes, *i.e.* offering better services. For example, if a customer concerns about media entertainment, a service bundle including Netflix and Spotify might be highly valuable to this customer.

In order to understand the interaction between customers and suppliers within the bundling process, we aim at analysing the exchange of valuable outcomes between suppliers and customers, *i.e.* how value-oriented business to customer (B2C) relationships are created. We propose a value-oriented framework by addressing the following two parts:

1) Customer-Supplier Interaction (CSI): To allow a suitable matching between what a customer needs and what the service suppliers offer, it is required to provide a mechanism to facilitate the interaction between these two perspectives. In this way tailored and more valuable service bundles can be created. Customers not only can take part in the reasoning but also provide valuable inputs such as information about their preferences. Service suppliers can be bundled to offer valuable outcomes for covering customer needs. At the end, such interaction leads to creating B2C relationships.

2) Dynamic Bundling (DyB): Once a customer need has been defined, service suppliers must be on-the-fly bundled to provide a solution for such need. In this sense, starting with single descriptions of services, the service bundles can be found by iterating about possible combinations. Moreover, since the service environment is always changing (customer needs and suppliers' offerings evolve over time), a dynamic framework can gracefully adapt itself to those changes.

To sum up, the contribution of this paper is twofold. First, we highlight the need for researching service bundling issues from a value-oriented perspective. Second, we present a dynamic framework for service bundling. Our framework takes into account the customer and supplier perspectives while performing service bundling. Besides, it focuses on the exchange of valuable service outcomes between customer and service suppliers.

The rest of the paper is organized as follows. A discussion about related work is given in Sect. II. Our bundling framework is presented in Sect. III. Later on, Sect. IV describes a case study as well as the application of our framework. Finally, we provide conclusions and future work in Sect. V.

[1]Netflix (https://www.netflix.com/) is a service for watching movies. Spotify (http://www.spotify.com/) is a service for listening to music.

II. Related Work

Although service bundling presents some common points with service composition as studied by the web service community [6], there are also some differences [2]. First of all, the current efforts in the web service community mainly focus on two aspects. *1) Technical descriptions:* web services are described by means of so-called IOPEs properties [3]. Moreover, the web services are merely considered as RPC-based [4] components that can be executed at any point in time [7]. *2) Process-oriented composition:* Since web services mainly describe procedural properties, *i.e.* preconditions and effects, the composition of web services is most of the time modelled as a planning problem. In this way, the goal of a composition problem is to generate a plan for web service invocation that obeys a set of given constraints and requirements [6].

Traverso and Pistore, [16], propose a framework for web service composition. 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.

METEOR-S [15], 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 combination of web services. Based on the STP, the required web 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 composite web service that can be used for customers.

DynamiCoS [3], provides a framework for web service composition. Web 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 web services as well as assuming that customers always have a clear idea about the web services they need.

u-service [12], is another dynamic framework that bundles web 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.

Moving away from the web service community, the following approaches aim at bundling commercial services. Serviguration, [2], presents a service bundling framework which is based on the customer and supplier perspectives. Even though the framework allows bundling services, it presents two main drawbacks. First of all, the framework does not provide any interaction between customers and suppliers. Second, the algorithm applies an exhaustive search, which makes it unsuitable for large-scale environments.

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

[3]IOPE: Input, Output, Pre-conditions and Effects.

[4]Remote Procedural Call (RPC)

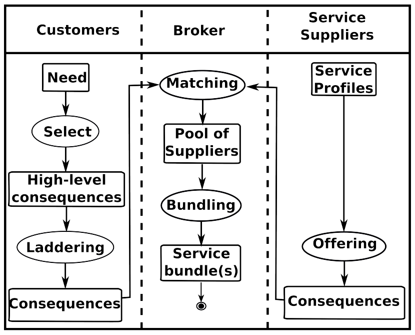


Figure 1: Framework for Dynamic Service Bundling.

e^3*service* [4] 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.

Letia *et.al.*, [13], propose a framework in which customers and suppliers establish an interactive dialogue for bundling services. The idea is that a supplier proposes some bundle to cover a given need. Later on, the bundle can be modified based on pro and counter arguments coming from the customer. Although the framework provides a dialogue between customer and suppliers, it lacks from a tangible example to verify its applicability.

Kohlborn *et.al.* [10], present a framework for service bundling that relies on relationships. Although the framework describes some interesting ideas, it does not present anything neither about customer-supplier interaction nor a tangible example.

As can be observed, on the one hand, the web service frameworks focus on process-oriented issues while completely overseeing more strategic issues such as the economic nature of services [16], [15], [3], [12]. On the other hand, although some frameworks focus on business issues *i.e.* economic reciprocity, what a service offer and what expects in return, yet they cannot completely deal with CSI and DyB [2], [13], [4]. To achieve our goals, *i.e. a value-oriented framework for dynamic service bundling*, we will mainly reuse concepts from the Serviguration and *e*^{3}*service* approaches [2], [4]. We elaborate on that by describing our dynamic framework.

III. Proposed Framework

To address the CSI and DyB parts, we propose a framework, which finds solutions in the clash between two perspectives: the *customer perspective* and the *supplier perspective* (Fig. 1). The framework is composed of three

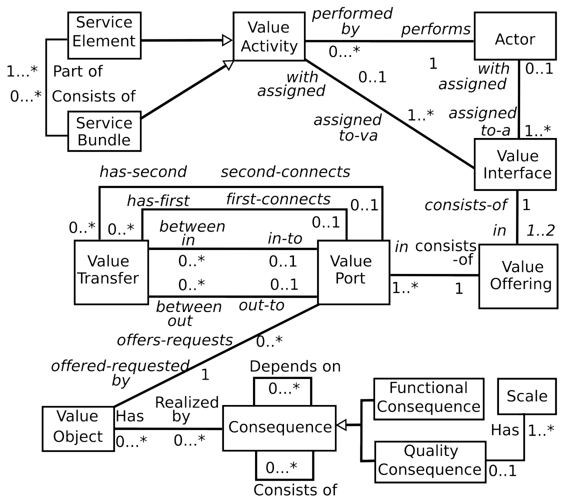


Figure 2: The e^3*service* [4] supplier ontology aligned to the e^3*value* [8] ontology.

main components: A customer, a broker and a pool of service suppliers. Whereas the service suppliers express their offerings by means of service profiles, the customers express their requests in terms of needs. Once a customer need is specified, a broker carries on an automatic process for matching both perspectives and bundling services. Finally, the broker can offer a pool of service bundles to the customer.

A. Customer and Supplier Perspectives

In order to deal with the CSI part, our framework reuses and adapts ontologies that have been previously researched by the Serviguration and e^3*service* approaches [2], [4]. The supplier perspective mainly describes the set of service outcomes that can be provided by service suppliers. The customer perspective depicts how the customer needs can be covered by means of wants and consequences. Fig. 2 and Fig. 3 show the main concepts of both ontologies. We explain them below.

1) Supplier perspective: Service suppliers are *actors* performing activities (*value activities*) to produce service outcomes (*value objects*) which can be offered to customers. In addition, as a result of its use or consumption, a value object has *functional consequences* and *quality consequences*. 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 [8] and [4].

2) Customer perspective: The customer ontology is based on concepts from established customer needs and marketing literature [4], [11], the main concepts are as follows:

- A `Need` represents a problem statement or goal [1], [11], [4] [5].

[5]Needs are the basic human requirements. Wants are specific objects that might satisfy the need.

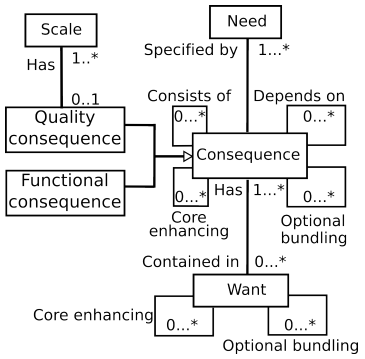


Figure 3: The e^3*service* customer ontology [4].

- A `Consequence` is anything that results from consuming (a combination of) valuable service outcomes [9], [4]. We distinguish between two types.
  - `Functional Consequence` [4] represents the functional goal that can be achieved through consumption of a service outcome that has a certain valuable property.
  - `Quality Consequence` [4] expresses qualitative properties of other. 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`.
- A `Want` [4] is a solution that is commercially feasible to be provisioned on its own. As a `Want` indicates a solution available in the market, at least one supplier should be willing to provide the solution.
- A `Scale` [4] groups `Quality Consequences` of the same type.

B. Dynamic bundling

To bundle services based on the customer and supplier perspectives, three steps have to be performed: laddering, matching and the bundling algorithm itself.

1) Laddering: Is a marketing practice which uses a conceptual map to represent how a customer links specific product attributes to high-level values [4]. In our case we apply it to link service consequences to customer needs via high-level consequences (Fig. 3). Although this step is not the core of the paper, we illustrate how it is performed by means of our case study in Sect. IV.

2) Matching: Matching determines a *matching pool* (MP) of services that plausibly provide part of the required functional consequences (FCs). Due to the variability of customer needs, single services rarely provide all the FCs on their own. FCs are the key components for matching the two perspectives. For each FC within a customer need, this matching process performs a comparison with all the FCs expressed at the supplier side and retrieves the services that

	β			
Service ID	FC_1	FC_2	FC_3	FC_4
S_{17}	1	1	1	0
S_{16}	1	1	0	1
S_{15}	1	0	1	1
S_{14}	1	0	1	1
S_{13}	1	0	0	1
S_{12}	1	0	0	0
S_{11}	1	0	0	0
S_{10}	0	1	1	1
S_9	0	1	0	1
S_8	0	1	0	1
S_7	0	1	0	0
S_6	0	0	1	1
S_5	0	0	1	0
S_4	0	0	1	0
S_3	0	0	0	1
S_2	0	0	0	1
S_1	0	0	0	1

Table I: Matrix representation. S_{15} and S_{14} are services offering the same FCs, which can be provided by different suppliers. The same holds for S_{11-12}, S_{8-9}, S_{4-5}, S_{1-3}.

offer the required FC. At the end, the retrieved services are stored in MP.

3) Bundling Algorithm: Since FCs are the matching point between the customer and supplier ontologies, it is meaningful to represent a service by means of its FCs. For instance, if the required consequences are: FC_1, FC_2, FC_3 and FC_4, a service providing FC_1, FC_2 and FC_3 can be represented through a binary vector $\beta = [1110]$, where the first three 1s stand for the FCs offered by the service, and the last 0 means that the service does not provide FC_4. Moreover, we highlight that this sorting does not imply any preference about FCs, *i.e.* all the FCs can be equally important.

In this way, MP can be described as depicted in Table I. Moreover, based on its β vector, services can be clustered, *i.e.* services are assigned to the same cluster if they offer the same FCs. The purpose of a cluster is not only to group services offering the same FCs but also to focus the searching of possible bundles on the clusters, *i.e.* explore combinations of clusters rather than combinations of a huge number of services.

Furthermore, we identify two types of clusters, upper and lower clusters. The Table II depicts the set of upper clusters. As can be observed, because all the upper clusters provide FC_1, *i.e.* they are overlapped in FC_1, they cannot be combined with each other. The name upper cluster comes from FC_1 being the most significant bit (msb) in the vector β, *i.e.* they have the highest β values.

Contrary to the upper clusters, some of the lower clusters can be combined with each other. Two lower clusters can

Cluster ID	Elements	Cluster.β
C_8	$\{S_{11}, S_{12}\}$	[1000]
C_9	$\{S_{13}\}$	[1001]
C_{11}	$\{S_{14}, S_{15}\}$	[1011]
C_{13}	$\{S_{16}\}$	[1101]
C_{14}	$\{S_{17}\}$	[1110]

Table II: Upper Clusters.

Cluster ID	Elements	Cluster.β
C_1	$\{S_1, S_2, S_3\}$	[0001]
C_2	$\{S_4, S_5\}$	[0010]
C_3	$\{S_6, C_1 \oplus C_2\}$	[0011]
C_4	$\{S_7\}$	[0100]
C_5	$\{S_8, S_9, C_1 \oplus C_4\}$	[0101]
C_6	$\{C_2 \oplus C_4\}$	[0110]
C_7	$\{S_{10}, C_3 \oplus C_4, C_2 \oplus C_5, C_1 \oplus C_6\}$	[0111]

Table III: Lower Clusters.

Solution Clusters	Solution Bundles
$C_{14} \oplus C_1$	$\{S_{17}, S_1\}, \{S_{17}, S_2\}, \{S_{17}, S_3\}$
$C_{13} \oplus C_2$	$\{S_{16}, S_4\}, \{S_{16}, S_5\}$
$C_{11} \oplus C_4$	$\{S_{14}, S_7\}, \{S_{15}, S_7\}$
$C_9 \oplus C_6$	$\{S_{13}, S_7, S_4\}, \{S_{13}, S_7, S_5\}$
$C_8 \oplus C_7$	$\{S_{12}, S_{10}\}, \{S_{12}, S_7, S_6\}, \{S_{12}, S_7, S_4, S_1\}, \{S_{12}, S_7, S_4, S_2\}, \{S_{12}, S_7, S_4, S_3\}, \{S_{12}, S_7, S_5, S_1\}, \{S_{12}, S_7, S_5, S_2\}, \{S_{12}, S_7, S_5, S_3\}, \{S_{12}, S_5, S_8\}, \{S_{12}, S_5, S_9\}, \{S_{12}, S_4, S_8\}, \{S_{12}, S_4, S_9\}, \{S_{11}, S_{10}\}, \{S_{11}, S_7, S_6\}, \{S_{11}, S_7, S_4, S_1\}, \{S_{11}, S_7, S_4, S_2\}, \{S_{11}, S_7, S_4, S_3\}, \{S_{11}, S_7, S_5, S_1\}, \{S_{11}, S_7, S_5, S_2\}, \{S_{11}, S_7, S_5, S_3\}, \{S_{11}, S_5, S_8\}, \{S_{11}, S_5, S_9\}, \{S_{11}, S_4, S_8\}, \{S_{11}, S_4, S_9\}$

Table IV: The pools of solution clusters and bundles.

be combined if, and only if, their FCs do not overlap. *E.g.*, since $C_1.\beta = [0001]$ and $C_2.\beta = [0010]$ do not overlap, C_1 and C_2 can be combined and added to C_3's elements. Moreover, when combined, the lower clusters can give birth to new clusters. *E.g.*, in Table III , the cluster C_6 does not contain any single service but a combination of two clusters.

We use the $C_1 \oplus C_2$ expression to denote that the services inside C_1 can be combined with the services within C_2. In simple words, it means that three steps are performed. 1) take an element from C_1, 2) take an element from C_2, and 3) combine the two elements. Consequently, based on Table III, $C_1 \oplus C_2$ produces the following combinations: $\{S_1, S_4\}$, $\{S_1, S_5\}$, $\{S_2, S_4\}$, $\{S_2, S_5\}$, $\{S_3, S_4\}$, $\{S_3, S_5\}$.

Once the upper and lower clusters are known, the next step is to combine both of them for generating the pool of solution clusters. A *solution cluster* is a combination of one upper cluster and one lower cluster such that their FCs: 1) do not overlap, and 2) match all the required consequences. *E.g.*, if we combine C_{14} with C_1, we can observe that $C_{14}.\beta$ does not overlap with $C_1.\beta$, besides $C_{14} \oplus C_1$ provide all the required consequences, *i.e.* $C_{14}.\beta = [1110]$ together with $C_1.\beta = [0001]$ provide $, FC_1, FC_2, FC_3$ and FC_4. Therefore, $C_{14} \oplus C_1$ is a solution cluster. In this way, after expanding all the solution clusters, we get the pool of solution bundles. The Table IV depicts the pool of solution clusters together with the pool of solution bundles.

The Algorithm 1 depicts the pseudo code of our proposed bundling algorithm. As can be observed, there are four main steps. Firstly, the services are clustered based on a binary vector β which represents the FCs offered by each one of the services within the cluster. Secondly, the so-called lower clusters are combined with each other. As already mentioned, the clusters can be combined if, and only if, their FCs do not overlap. In this way we assure that a cluster keeps the property of being a pool of non-overlapping FCs. Thirdly, it generates the solution clusters by combining the upper clusters with the lower clusters. Finally, the algorithm expands the pool of solution clusters and generates the pool of solution bundles.

Algorithm 1 Bundling algorithm - General view.

```
1: procedure BUNDLING($r, p$)    ▷ $r$ = required consequences, $p = MP$
2:     $n \leftarrow |p|$    ▷ number of services
3:     $m \leftarrow |r|$    ▷ number of required consequences
4:     $C \leftarrow Generate\_clusters(n, m)$
5:     $Combine\_lower\_clusters(C, m)$
6:     $SC \leftarrow Generate\_solution\_clusters(C, m)$
7:     $solutions \leftarrow Expand\_solution\_clusters(SC)$
8:     return $solutions$
9: end procedure
```

C. Discussion

Since service bundling requires combining services to fulfil a given need, a huge number of services can seriously impact the performance of any bundling algorithm, *i.e.* the higher the number of services, the bigger the solution space. In this sense, our proposed algorithm aims at avoiding that issue by focusing on the required FCs. Consequently, the solution space only depends on the number of FCs, which for a given customer need are rarely numerous.

Besides, we can also cluster services based on their FCs. In this sense, to find the pool of possible bundles we must only combine those clusters. Furthermore, we have made a distinction between upper and lower clusters. The heuristics behind this distinction is as follows; since the upper clusters cannot be combined with each other, the core of the bundling process only relies on finding the combinations within the lower clusters. In this way, this process only takes into account half of the search space, i.e. search space $= \frac{2^m-1}{2}$, where m is the number of required FCs. Once the lower clusters have captured all the possible combinations, the upper clusters can be combined with the lower clusters if, and only if, their β values do not overlap. In this way we are sure that services inside a bundle do not offer the same FCs.

Finally, as can be observed in Fig. 2, there are two types of consequences. Although quality consequences (QCs) have not been considered in the bundling process, they will be used to express preferences among the pool of possible bundles. We illustrate this statement as well as the whole bundling process by means of a case study.

IV. Case Study

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 [5]), skill gap analysis can overcome the 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 have been already identified during skill gap analysis, the 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.

Central actor 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).

To describe the relationship between functions and competencies enterprises define function profiles, which usually contain the following essential parts [6]: a competency map (or tree) that references RCDs and proficiency levels for `Competencies`. We consider that the required `Competencies` can be treated as the FCs to be performed by a job candidate.

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[7]. In order to perform each of those functions, human operators with specialized competencies are required.

[6]http://www.ostyn.com/standardswork/competency/ReusableCompMapProp.pdf
[7]cf. http://www.nedcar.nl/content/view/44/49/lang,en/

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 for education emerges that has to be answered by the e-service web. Summarizing, whereas service suppliers can provide competencies by means of FCs, the candidates search for service bundles offering the competencies they lack for a given job.

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. The main point we want to make here is twofold. On the one hand, all these educational parties can independently offer their services to a service web. On the other hand, candidates looking to fulfil some gaps by means of competencies can actually surf this e-service web to find service bundles providing the required competencies.

A. *Dynamically Bundling Services*

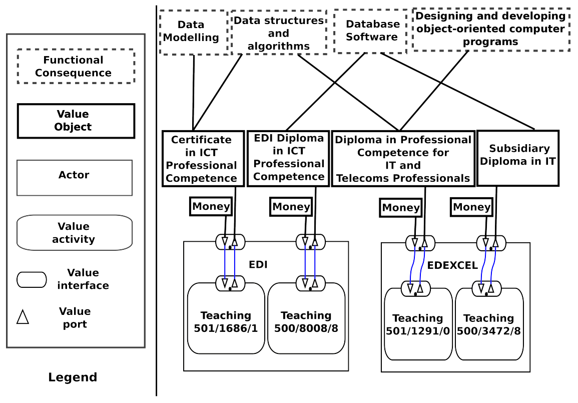
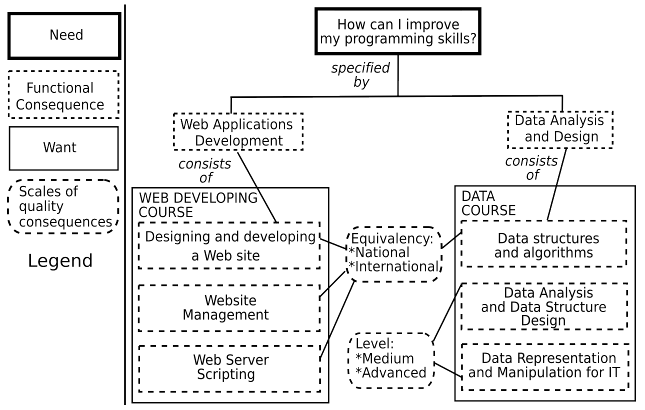


Figure 4: Visual representation of the service catalogue generated from Web data in NDAQ.

1) Supplier Perspective: For the demonstration, we surveyed a number of publicly available competency databases and picked out the National Database of Accredited Qualifications[8] (NDAQ), which contains details of Recognised Awarding Organisations and Regulated Qualifications in England, Wales and Northern Ireland. We harvested the NDAQ database and generated a catalogue of service suppliers, of which an excerpt is depicted in Fig. 4 and annotated using the concepts in the supplier ontology (Fig. 2). Actors are educational institutes performing teaching activities, *i.e.* value activities [9]. 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.,* Education Development International plc (`EDI`) performs the value activity `Teaching 501/1686/1`, which produces the value object `Certificate in ICT Professional Competence` with the associated functional consequences (RCDs) `Data Modelling` and `Data structures and algorithms`. Our final catalogue is composed of 58 services provided by four service suppliers.

2) Customer Perspective: We have also designed a customer catalogue (Fig. 5) 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.



Need
Functional Consequence
Want
Scales of quality consequences
Legend
How can I improve my programming skills?
specified by
Web Applications Development
Data Analysis and Design
consists of
consists of
WEB DEVELOPING COURSE
Designing and developing a Web site
Website Management
Web Server Scripting
Equivalency: *National *International
Level: *Medium *Advanced
DATA COURSE
Data structures and algorithms
Data Analysis and Data Structure Design
Data Representation and Manipulation for IT

Figure 5: Customer catalogue

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 catalogue in Fig. 5. 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.

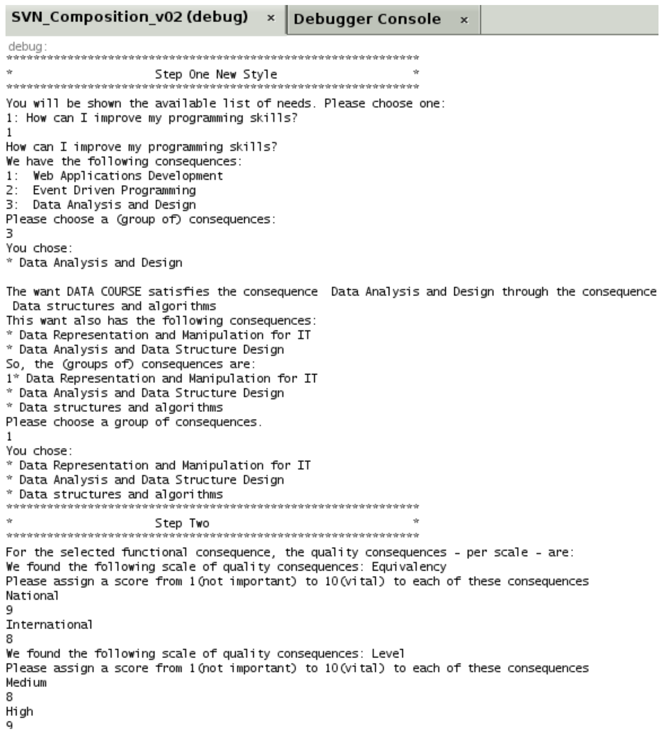
The next step involves discovering more consequences through the notion of wants explained earlier. In this case the wants are courses through which FCs (competencies) can be obtained. In Fig. 5, the FC `Data structures and algorithms` is contained in the `Data Course` want. By exploring this want, the FCs `Data Analysis and Data Structure Design` and `Data Representation and Manipulation for IT` are discovered.

Finally, as part of the interactive dialogue, customers are asked to indirectly evaluate the relevance of FCs by scoring their associated QCs. *E.g.,* for the FC `Data Representation and Manipulation for IT`, a customer is requested to score the relevance of the QCs `Medium` and

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

[9]We assume that teaching is one type of educational service.

`Advanced`. Later on, service bundles can be sorted based on these preferences. Fig. 6 depicts how the process was performed in our prototype.



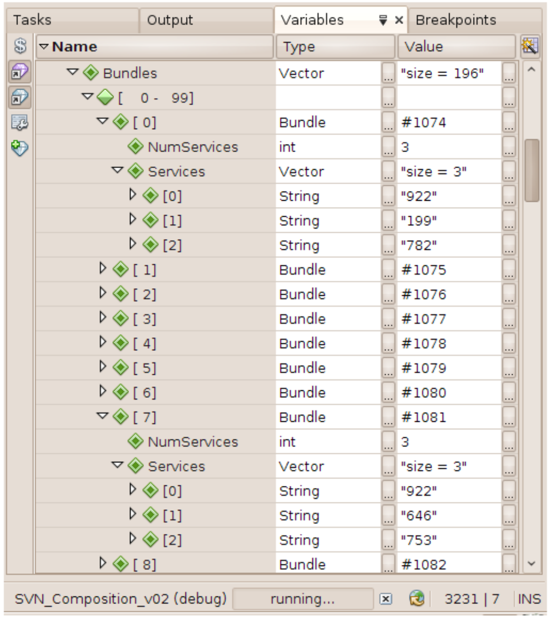
```
debug:
*************************************************************
*                    Step One New Style                     *
*************************************************************
You will be shown the available list of needs. Please choose one:
1: How can I improve my programming skills?
1
How can I improve my programming skills?
We have the following consequences:
1:  Web Applications Development
2:  Event Driven Programming
3:  Data Analysis and Design
Please choose a (group of) consequences:
3
You chose:
* Data Analysis and Design

The want DATA COURSE satisfies the consequence  Data Analysis and Design through the consequence
 Data structures and algorithms
This want also has the following consequences:
* Data Representation and Manipulation for IT
* Data Analysis and Data Structure Design
So, the (groups of) consequences are:
1* Data Representation and Manipulation for IT
* Data Analysis and Data Structure Design
* Data structures and algorithms
Please choose a group of consequences.
1
You chose:
* Data Representation and Manipulation for IT
* Data Analysis and Data Structure Design
* Data structures and algorithms
*************************************************************
*                    Step Two                               *
*************************************************************
For the selected functional consequence, the quality consequences - per scale - are:
We found the following scale of quality consequences: Equivalency
Please assign a score from 1(not important) to 10(vital) to each of these consequences
National
9
International
8
We found the following scale of quality consequences: Level
Please assign a score from 1(not important) to 10(vital) to each of these consequences
Medium
8
High
9
```

Figure 6: Illustration of the Laddering process.

4) Matching: Once the required consequences have been specified in the customer side, the matching process retrieves all the possible services that partially or completely offer the required consequences, *i.e.* generating MP.

5) Bundling: At this step the Algorithm 1 is applied with the following input parameters: $r = \{$`Data Representation and Manipulation for IT`, `Data Analysis and Data Structure Design`, `Data structures and algorithms`$\}$ and $p = MP$. Fig. 7 shows a sample of the computed bundles, which in total are 196. *E.g.*, the bundle$_{1074}$ is composed of the services S_{922}, S_{199} and S_{782} while the bundle$_{1081}$ is composed of the services S_{922}, S_{646} and S_{753} [10]. Once the solution bundles have been computed, we can generate visual representations for them.

Fig. 8 depicts two of the service bundles providing the required FCs 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. 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 [2], our bundles are represented according to the e^3*value* ontology, *i.e.* the bundles can be visualized and used within e^3*value* models [8].



Name	Type	Value
▽ Bundles	Vector	"size = 196"
▽ [0 - 99]		
▽ [0]	Bundle	#1074
NumServices	int	3
▽ Services	Vector	"size = 3"
▷ [0]	String	"922"
▷ [1]	String	"199"
▷ [2]	String	"782"
▷ [1]	Bundle	#1075
▷ [2]	Bundle	#1076
▷ [3]	Bundle	#1077
▷ [4]	Bundle	#1078
▷ [5]	Bundle	#1079
▷ [6]	Bundle	#1080
▽ [7]	Bundle	#1081
NumServices	int	3
▽ Services	Vector	"size = 3"
▷ [0]	String	"922"
▷ [1]	String	"646"
▷ [2]	String	"753"
▷ [8]	Bundle	#1082

Figure 7: Solution Bundles.

The bundles in Fig. 8 depict how service suppliers can interact together to provide service bundles that have been designed by interacting with a customer. Moreover, 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 it will be considered as future work.

Furthermore, we have implemented a prototype in which we represent both catalogues by means of RDF files. The bundling process is applied with Jena [11], 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 [12].

V. Conclusions and Future Work

We have presented a framework for service bundling that focuses on the exchange of valuable service outcomes between service suppliers and customers. The approach deals with two parts: CSI and DyB. CSI allows value (co)-creation by bundling services that are more suitable and then valuable for final customers. DyB provides a flexible way for combining single suppliers into service bundles.

In our case study we have demonstrated the applicability of our approach as well as the importance of analysing value-

[10] 1074 and 1081 are semi-arbitrary IDs.

[11] http://jena.sourceforge.net/

[12] http://www.e3value.com

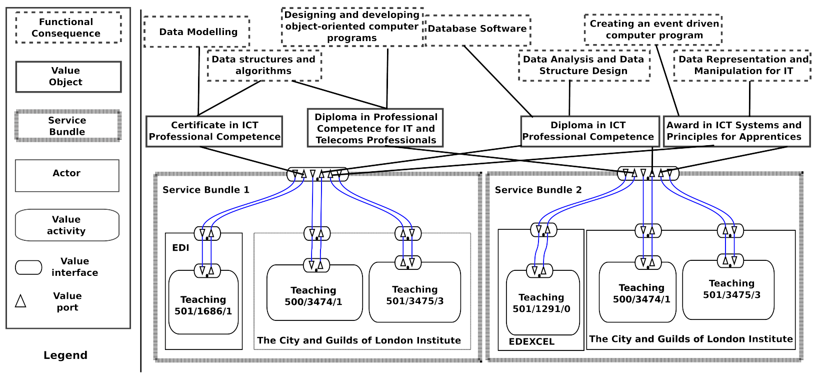

Functional Consequence
Value Object
Service Bundle
Actor
Value activity
Value interface
Value port
Legend
Data Modelling
Designing and developing object-oriented computer programs
Database Software
Creating an event driven computer program
Data structures and algorithms
Data Analysis and Data Structure Design
Data Representation and Manipulation for IT
Certificate in ICT Professional Competence
Diploma in Professional Competence for IT and Telecoms Professionals
Diploma in ICT Professional Competence
Award in ICT Systems and Principles for Apprentices
Service Bundle 1
EDI
Teaching 501/1686/1
Teaching 500/3474/1
Teaching 501/3475/3
The City and Guilds of London Institute
Service Bundle 2
Teaching 501/1291/0
EDEXCEL
Teaching 500/3474/1
Teaching 501/3475/3
The City and Guilds of London Institute

Figure 8: Service bundles

oriented aspects within a service environment, in this case an educational service environment.

Moreover, we consider that the concept of a cluster can be further explored to solve issues such as adaptability and business to business (B2B) relationships. In the sense of adaptation we foresee two settings. 1) New services come: Once a new service is available, it can be clustered according to its FCs and incorporated to new solution bundles by expanding the cluster in which it was assigned . 2) Services within a solution bundle leave: In this case the complete bundle can be replaced by other bundle or the leaving services can be replaced by services within the cluster where they are assigned. For the second issue, by describing services based on their FCs, we can also address issues regarding B2B relationships. For instance, if two services are connected by a B2B rule requiring to merge both services, i.e. a compulsory bundling, we can simply merge their FCs and threat them as a single service.

Finally, as part of the future work, we will perform a more exhaustive analysis to determine the real complexity of our cluster-based bundling algorithm.

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

References

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

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

[3] 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.

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

[5] 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.

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

[7] George Baryannis et. al. *Service Composition*, volume 6500 of *Service Research Challenges and Solutions for the Future Internet.*, beitrag in buch Service Composition, pages 55–84. Springer, October 2010.

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

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

[10] T. Kohlborn, C. Luebeck, A. Korthaus, E. Fielt, M. Rosemann, C. Riedl, and H. Krcmar. Conceptualizing a bottom-up approach to service bundling. In *22nd International Conference on Advanced Information Systems Engineering (CAiSE'10)*, 2010.

[11] Philip Kotler and Kevin Keller. *Marketing Management*. Prentice Hall, 2006.

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

[13] I. A. Letia and A. Marginean. Client provider collaboration for service bundling. *Advances in Electrical and Computer Engineering*, 8:36–43, 2008.

[14] 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.

[15] 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.

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