Client Provider Collaboration for Service Bundling

Ioan Alfred LETIA, Anca MARGINEAN
Technical University of Cluj-Napoca
Baritiu 28, RO-400391 Cluj-Napoca
{letia, anca}@cs-gw.utcluj.ro

***Abstract*—The key requirement for a service industry organization to reach competitive advantages through product diversification is the existence of a well defined method for building service bundles. Based on the idea that the quality of a service or its value is given by the difference between expectations and perceptions, we draw the main components of a frame that aims to support the client and the provider agent in an active collaboration meant to co-create service bundles. Following e3-value model, we structure the supporting knowledge around the relation between needs and satisfying services. We deal with different perspectives about quality through an ontological extension of Value Based Argumentation. The dialog between the client and the provider takes the form of a persuasion whose dynamic object is the current best configuration. Our approach for building service packages is a demand driven approach, allowing progressive disclosure of private knowledge.**

***Index Terms*—Argumentation over preferences, Customer needs, Service configuration, Service offering, Web services**

I.INTRODUCTION

Reality is not black and white; therefore classical logic fails in dealing with all the aspects of knowledge. Aiming to persuade rather than to prove, demonstrate or refute, argumentation proved to be more appropriate for solving disagreement, so common in real life, and for reaching an acceptable solution in situations with incomplete information, different perspectives or different preferences.

The Dung's abstract framework [7] laid the foundation of argumentation frameworks. [3] extends it by adding values to arguments and filtering the defeat relation relative to these values – Value Based Argumentation Framework (VAF) with Audiences. Audiences empower different parties to decide the defeat of an argument. The structure of the argument does not count in VAFs, from this raising the simplicity of finding the admissible sets. [14] combines BDI paradigms with argumentation aiming at a solution for practical reasoning. An argumentation semantics has also been described for Defeasible Logic in [8] and [15] modeled hard and soft business rules for negotiation purpose using this non-monotonic logic. Attack and admissibility have been studied in assumptions based framework [4] that refines the use of abduction for non-monotonic reasoning.

Lately, a new direction is evident in argumentation: identification of new forms of attack according to the internal structure of the argument. Such an enhanced semantics for attack is given in [19], while the Argumentation Interchange Format [5] aims to provide an interchange frame for representing the argument and the possible attacks.

Representation, composition and management of web services have been thoroughly studied, but no common solution was yet identified [9], [11], [17], [12]. The subjective and imprecise nature of service quality requires objective criteria. A common direction in qualitative composition of services is the use of the utility. [11] introduces utility function policy and multi attribute optimization. Functional and non-functional requirements of web services are handled by Web Service Offering Language (WSOL) [18]. [18] introduces an XML, WSDL compatible notation that allows description of classes of services associated to a single web service. Each class is determined by functional and non-functional QoS factors. All along the work in the field, evaluation of the quality seems to be the main issue raising problems when dealing with quality.

Differently to approaches where web services are seen as auto descriptive, reusable software components, composible through flow techniques, [1] and [2] introduce a new perspective over the services, oriented on the relation between needs and satisfying services. The quality model in these papers lays its foundation on the difference between the expectations and the perceptions of the client about the services.

Following this model, we define a frame for agents capable to manage services in an intelligent and active manner. Both the client and the provider become active components in building and selecting the required configuration. An agent is better in managing the services if it succeeds in "selling" well the service, earns good reputation and satisfies the client. In order to do that it needs good knowledge models and efficient strategies for building proposals and for persuading the client.

II.SERVICE REPRESENTATION

In a perfect situation, the creation of value through services requires perfect providers and perfect clients, omniscient, intensively enhanced with knowledge, able to objectively assess their offers, respectively their requirements based on complex models of the dependencies between the elements involved. In the real situations, competitive providers need to rely on their ability to build appropriate models of their clients - needs and demands with the acceptable sacrifice associated. The reality and the management/marketing theories proved that without a thorough understanding of the clients it is hard to offer high quality services. The problem does not change when switching to the client's view, but it is unfeasible to expect from the client to have a comprehensive model for the means of satisfying his needs.

Digital Object Identifier 10.4316/AECE.2008.01007

We adopt the e3 value approach [1], by separating the knowledge in client perspective and provider perspective. The basic elements of the requirements engineering - needs, wants, and demands - are modeled in an ontology of needs representing the provider's view about the client's needs - the what component. The means for satisfying the needs are modeled through service offering ontology encapsulating the how component.

As a difference to the e3-value model, we consider services as added value not only to the client, but also to the provider. Therefore, the business knowledge of a provider incorporates (1) common accepted mappings between needs and satisfying services and (2) mappings between needs and services with a high level of novelty. The second one is aimed to give a strategic advantage to the provider in face of the competitors. When building a solution to a requirement, both mapping types are used, but the use of second type mappings needs further attention, their adequacy being established according to the current client in order to maintain their promotion in a non-invasive limits.

The representation AND\OR\XOR proved to be expressive enough for capturing the relations between needs, respectively services. The context aware feasible solution graphs from [1] is maintained as a representation of the mappings existing between the needs and the services descriptions (resources).

A. NEEDS ONTOLOGY

Creating a framework for describing a dialog between a client and a provider, the needs ontology encapsulates the clients' knowledge about creating value through the satisfaction of the needs. In the absence of needs, there is no reason for existence of services, therefore the needs are considered to be the core of the service theory.

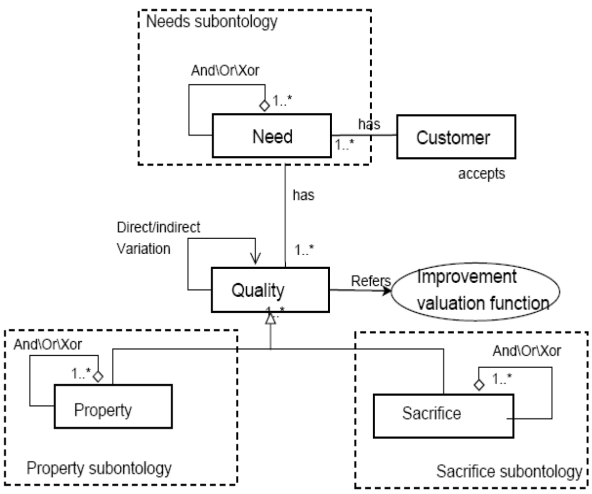

Figure 1. Overall structure of Needs Ontology.

As it can be seen in Fig.1., the customer can have more needs and for each need there can be more quality factors. The quality aims to capture the expectations of the client in his needs satisfaction, and the zone of tolerance that falls between the desired and adequate service levels.

We consider two important subclassifications for the quality factors: properties and sacrifice. The properties allow the customer to express, more or less vaguely, its requirements in the needs satisfaction, like quality factors or certain properties of the potential services. The sacrifice stands for the acceptable sacrifice for the client; it is characterized by a deeper subjective nature than the properties, its valuation being client dependent. Even if we can talk about subjective values for properties from the client's viewpoint, it is necessary to observe that they become more objective when they are associated to certain services.

All the three subontologies - Needs, Properties and Sacrifice - are domain dependent. Both the client and the provider are using the ontology of Needs. Even so, it is realistic to accept that the ontology used by the client can differ from the one used by the provider. In the current paper we don't address this issue, but we consider that the direction chosen for extending the classical VAF facilitates creation of a support for a coherent dialog between parties using different ontologies. In order to avoid failure in reaching the agreement, it is imperative to include in the provider's knowledge a general, commonly accepted model for needs and their associated quality factors.

Besides aggregation through AND\OR\XOR structure, we model also horizontal relations between quality factors, regardless they are properties or sacrifice. This is also the reason to bring the two perspectives of properties and sacrifice together, as a difference to the e3 model. For now we define two relations: direct and indirect variations. There is a direct variation between two factors if one's improvement implies the improvement of the other one too. Otherwise, if the improvement of one factor determines the declination of the other factor, then there is an indirect variation relation. These relations aim to represent the tacit knowledge about dependencies between properties or sacrifice subconcepts.

The focus of the current work being on quality of needs satisfaction, we do not insist on needs ontology useful especially for verifying the functional constraints. Instead, we insist on quality modeling. Similar to quality grupoids from [20], a model of quality factor is included in the Needs Ontology:

- **Improvement valuation function.** For each QoS factor *q*, there is a function f(q) defined on the values of the factor, comparing two values of q, and returning 1 if the first one is better than the second one. Based on these functions, a total order relation » on $\bigcup_{q\subset O(Q)} Dom(q)\times Dom(q)$ is defined in the following way: $\forall q\in O(Q)$, $\forall val_1$, $val_2 \in Dom(q) \bullet val_1 » \mathrm{val}_2 \quad \Leftrightarrow \; f(q)(val_1, val_2) > 1$ where *O(Q)* stands for the reunion between properties and sacrifice. For example, we consider the better cost to be the one that is smaller, therefore the f(cost)($cost_1$, $cost_2$) will be $cost_2/cost_1$. Between two different values of reliability, the greater is the best so we use a direct divide between the corresponding values.
- **Aggregation function**. For each quality factor, aggregation function assesses an overall value of a *q* factor in a quality vector - the total time, the overall reliability, the estimated risk. It can be sum, min, or max, or any other domain dependent function. For a quality factor with a discrete domain, the overall function and the comparing one can be defined extensionally. It may be the case to be unable to define a strict, mathematical overall function for some quality factors; in this situation, an approximate evaluation function is required.

B. SERVICE OFFERING ONTOLOGY

The business world evidenced that providers' and customers' viewpoints on services are different. If the client perspective on value creation is needs satisfaction driven, the provider perspective, represented through service offering ontology, encapsulates the services and their components as they are actually delivered in order to satisfy the customers' needs [2].

The service offering ontology embraces the provider knowledge about the available services, their interdependencies and provided quality. We use a simplified version of the e3 value model, avoiding the use of resources. Therefore, in the service offering ontology, the quality is directly tied to the service, not to resource. As in the cited model, the business essence of the services is described through construct form configuration theory. Dependencies relations create the frame for service bundling; a service bundle aggregates services based on relations like enhancing, supporting, and substitute or excluding (Fig. 2).

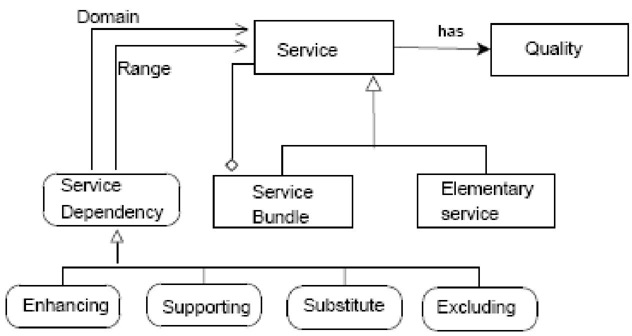

Figure 2. Service offering ontology.

C. RELATING NEEDS TO SERVICES

The two different perspectives - client and provider - are bound together through the relations between needs and services. Taken from the same configuration theory, context aware feature-solution graphs are used to map services to the needs that they satisfy. The feature space is given by the needs, while the solution one is given by the services from service offering ontology. We extend the context space to specific business rules and client typologies; the context determines the applicability of the feature-solution rules. Specific business rules and accumulated information about the client influence the level of satisfaction of a need by a service. Taking a modal view on needs - services relations, more types of relations are identified: selection, rejection, positively influenced by, and negatively influenced by.

D. SPECIFIC BUSINESS RULES

Besides a good model of needs, services and relation between them, a competitive organization usually practices a private marketing policy, customized according to the current client and to the context. This policy, sometimes modeled as soft business rules [15], is different from the knowledge encapsulated in the above mentioned ontologies – hard business rules, due to its high specificity level; it can be seen as the exceptions met in knowledge modeling. For example, when an organization wants to promote a new service, for replacing or adding value to an existing one, or for offering a solution for a yet unsatisfied need, it must have a good promoting policy for persuading the clients. A strong requirement for such a policy is the identification of the target zone of clients. Moreover, the relation between the new service and the satisfiable needs must be clearly stated, in order to be able to give strong arguments to the clients.

The marketing policy adds value not only to the client, but they are oriented to give competitive advantages to the provider too. We propose the same context aware feature-solution graph representation, and we give a light classification to the possible rules:

- completely instantiated: associate a new service, different from the conventional one to a need. In this case, the rules will relate a service to a need or to a need having a certain value for some quality factors (properties or sacrifice)
- based on ontological relations: for all the needs, wants or demands respecting an ontological constraint customize the satisfying service (the service itself or its properties).

According to the lifecycle phase of the service from the marketing viewpoint, rules from here can pass into the common ontologies or vice versa. Therefore the provider has to be careful and avoid being too aggressive in his promotion, selecting the target; it can also be the case that the client will not understand the arguments given for the proposal and useless overhead in the communication is reached.

E. CLIENT TYPOLOGY

One of the premises for an efficient negotiation is for each involved party to build models of its opponent. Given the abilities to correctly observe the behavior of the opponent and to update the model for it, the negotiator is better in choosing proper actions [16].

Aiming at a negotiation over appropriate configurations of services, we consider that the world model incrementally built by the provider is expressed through a client profile that allows its classification in a client typology. Client typologies influence the adequacy of some services as a solution for certain needs, therefore they should determine the provider's behavior. We propose an ontology for supporting typologies, and for the current work we propose a simple And\Or\Xor representation. For real situation this representation needs further extensions in order to support the classification of the client based not only on needs and properties, but also on a world model continuously updated during the dialog.

Classification of a client in one typology is done (1) objectively, based on the concrete data about the client, like small, medium, or big organization, business value, and geographical situation and so on, and (2) subjectively, based on the provider's view about the client, extracted from the arguments exchanged in the dialog or from past collaborations.

III. KNOWLEDGE ENHANCED PROPOSALS

We consider that the valuation of the proposed service is directly dependent to the accuracy reached by the proposed service in matching the expectation of the client: needs and demands, their properties and acceptable sacrifice.

The request for a service is defined in terms of needs, and associated quality expressed through properties and acceptable sacrifice. The form of the request is the following $[(N_1, Q_1), ..., (N_n, Q_n), Q]$, where N_i is a need, Q_i is a vector of associated required properties and acceptable sacrifices, and Q is the quality vector for the request as a whole. The representation for a possible service bundling is similar, with the important difference that instead of needs we have services: [(S_1, Q_1), ..., (S_n, Q_n), Q]. Another difference between request and possible service bundle is that the first one is more subjective - including more or less strict quality constraints, while the latter one is objective, expressing the properties and the required sacrifices for the contained services. Constraints on properties and sacrifices can be of two types: (1) existence of a property - for example the monitoring property in satisfying logistic needs, or rollover abilities for a banking transaction demand, or excluding a certain type of service (without off-line paying) and (2) a quantified constraint.

We assume that the service bundles meeting the functional constraints exist; a configuration solution is proposed in [2]. The qualitative evaluation of the possible solutions, considering different opinions for client and provider, followed by a selection of the most appropriate one, constitutes the object of our present work.

The solution is built step by step, through argumentation based persuasion, aiming agreement above different valuations between the client and the provider. We are talking about persuasion, because it is almost impossible to quantify over quality factors in order to establish a total order relation over the solutions, therefore the best solution will be the one that the persuaded client will consider acceptable. By reaching an acceptable solution we understand that the configuration was proposed by the provider and the client was persuaded that this is the best solution comparing to the others. We need to underline that this solution is not necessarily the best possible one. The proposals have the form of (i) a new configuration or (ii) a proposed improvement on a reference solution together with the arguments justifying that. Therefore the object of persuasion changes between certain configurations and improve relation between two service bundles satisfying the quality constraints of both the client and the provider.

The main reasons for using a step by step approach, with improvement as main persuasion object are: (1) the solution is co-created by the involved parties assuring more transparency, (2) it facilitates generation of personalized configuration, (3) it handles the lack of an agreed quality measure bringing together all the mentioned dimensions, through a one by one comparison, and (4) it allows integration of different opinions. Even though the overall quality of a solution can not be measured, it is always possible to say that one solution is better that the other in regard to a certain factor.

Given existence of more opinions on the importance of different quality factors, we chose VAF framework with audiences [3] as the starting point in argumentation. The VAF framework has the advantages of integrating different preference relations over the arguments through audiences. In this argumentation framework, the argument structure is not important; therefore there can be more argued objects in the same time. Depending on the credulous or skeptical approach followed, the VAF results in admissible sets of arguments from which the supported argued objects are identified.

A. MULTI-CLASS VALUE BASED ARGUMENTATION FRAMEWORK WITH AUDIENCES

We propose an extension of VAF with classes of value instead of values, and relate these classes to concepts in an ontology. The basic idea for the extension is introducing classes of values and allowing preferences not only inside a class, but also between classes. This way, a new intermediate level is introduced over the values from standard VAFs. Starting from the classical definitions for VAF with audiences, we define the extended framework.

Definition 1. A multi-class VAF (χ, A , V, η) is defined by <$H(\chi, A, V, \eta)$>, where $H(\chi, A)$ is an argument system, V is a set of classes {C_1, C_2, ..., C_n}, each class C_i having m_i values $\{ v_1^i, v_2^i, ..., v_{m_i}^i \}$, and $h : c \rightarrow \bigcup_{C_i \in V} \{v_1^i, v_2^i, ..., v_{m_i}^i\}$ a mapping that associates a value η(x) with each argument $x \in c$.

Definition 2. An audience for a VAF (χ, A , V, η), is a binary relation $R \rightarrow (V \cup q) \times (V \cup q)$, where $q = \bigcup_{C_i \in V} \{v_1^i, ..., v_{m_i}^i\}$, whose transitive closure R* is asymmetric, i.e. at most one of the (v, v'), (v', v) are members of the R* for any distinct $v, v' \in V \cup q$. We say that v_i is preferred to v_j in the audience R iff:

- $\exists C_1, C_2 \in V$ s.t. $v_i = C_1, v_j = C_2$ and $(C_1, C_2) \in R^*$ (preference between classes) or
- $\exists C \in V$ s.t. $v_i, v_j \in C$ and $(v_i, v_j) \in R^*$ (preference inside a class) or
- $\exists C_i, C_j \in V, C_i \neq C_j$ s.t. $v_i \in C_i$ and $v_j \in C_j$ and $(C_i, C_j) \in R$ (preference between elements of two different classes).

An audience expresses preference relation between classes of values and also between values in these classes. As it can be seen, the preference can be defined between classes or between two values inside the same class (the first two cases), while in the case of values in different classes, the preference is given by the preference between their classes. The other definitions for VAF remain the same, given that the audience is defined as above.

For our quality driven search of service bundle, we have two audiences, one for the provider and one for the client. We consider that a client or a provider would valuate the quality of a configuration along three dimensions: needs satisfaction, quality constraints fulfillment and conflict resolution. Therefore, in each audience, classes correspond to these three dimensions, V= {needs, quality, conflict}. Both the client and the provider can define preference between these classes, specifying for example that arguments about the quality fulfillment are the strongest. For the reason exposed, we don't insist on the first class of values. As for the second one, the values are different properties and sacrifices as identified in the Property Subontology *O(P)* and Sacrifice Subontology *O(S)* of the Needs ontology. Having both sacrifice factors and properties in a single ontology allows the client and the provider to express preferences between quality factors regardless they are properties or sacrifice.

Existence of more dimensions of the attacks expressed through classes of values is similar to the use of different types of attacks from approaches based on internal structure of arguments [19].

B. ARGUMENT STRUCTURE

The structure of the arguments does not count for the argumentation based self-deliberating process of involved parties, but it is very important for the dialog between them. The structure of the argument should be self-explanatory and sufficient for one to decide acceptation of the argument or to build a good counter-argument. Through counterattack we understand an argument that attacks one argument given by the other party.

The arguments are generated according to an instantiation schema that we will describe later. The knowledge carried by the exchanged arguments includes not only the attack relation and the promoted values, but also some ground facts expressing the rational of that argument. The detailing level of these ground facts should be directly dependent on the level of intimacy between agents, as defined in [16]. For our case, in a simplified version of negotiation, it is indirectly determined by the updated world model encapsulated in the identified client typology.

An argument includes the proposed improvement of the proposed configuration, a sign + or - signifying a pro or a counter argument and the reason - the facts supporting this. Therefore the argument structure is a tuple <*sign, obj , Ground_fact*>. If the argument is for a configuration, then the *obj* stands for that configuration. If the argument argues about an improvement, then the *obj* is $value(C_1,C)>value(C_2, C)$, where C_1, C_2 are two different possible service bundles, while C is the request, and $value(C_i, C)$ is the value of the service as perceived by the party given the argument. The increase in value does not necessarily mean an overall improvement along all three dimensions (needs, qual, and confl); it stands for an improvement along at least one dimension. When proposing or attacking a configuration, the agents (customer and provider) have to say why a configuration is better suited for the client request than the other one. During the dialog initiated by the client through its request, the provider will propose and the client will attack, counter attack, or approve. A possible classification of the ground facts of the arguments includes:

- matched functional properties, eventually relations between services
- more or less matched functional properties combined with a positive variation of some quality factor for certain properties, included in the request,
- a positive variation of the overall quality of the configuration
- the relative variation of two quality elements related by an indirect variation

C. INSTANTIATION RULES

The provider will consider for the persuasion phase only service bundles supported by admissible arguments regarding his own audience. According to the identified client typology or other context issues, on each step, the provider uses its instantiation schema to derive new arguments and only after that the admissible set is determined. The admissible arguments are used both for (1) selecting an acceptable offer to propose and (2) to persuade the customer. The provider reasons with arguments above quality factor regardless they were or were not included in the client request. However, it may be the case that the client will not be able to reason on other quality factors than the requested ones, therefore the provider should choose the arguments for persuasion from those referencing quality factors mentioned by the client.

We are interested in the strategy used by the provider for generation of its proposals. We will use the notations $Req=[(N_1, Rq_1), ...(N_n, Rq_n), Req_Q]$, for requests, where Rq_i are vectors of constraints on quality elements, and proposals of the form $C=[(S_1, Sq_1), ...(S_n, Sq_n), Prov_Q]$, where Sq_i is a vector of objective values for the quality elements associated to the available services, S_i includes the need satisfied by the service and $Prov_Q$ stands for overall quality values.

Table 1 shows some potential rules from instantiation schema. For each element of this schema there must be defined the conditions, the structure of the resulting argument and the promoted value.

The $qual(N_i, q_i, sq_j, C)$ relates a property q_i of a need N_i included in the request to the quality property sq_j of the configuration C including service that satisfies the need. Formally, we could define it in the following way:

$qual(N_i, q_i, sq_j, C)\ \hat{U}$

$\$Rq_i \cdot constraint(q_i)\ \hat{I}\ Rq_i\ \grave{U}\ (N_i, Rq_i)\ \grave{I}\ Req\ \grave{U}$

$\$S_j\ satisfying\ N_i \cdot (S_j, Sq_j)\ \grave{I}\ C\ \grave{U}\ sq_j\ \grave{I}\ Sq_j.$

The first three rules determine arguments for a single configuration. The rules 1 and 2 simply expresses that a configuration is good or not according to the satisfaction of a certain quality constraint. As for the rule 3, it argues for rejecting a configuration that concomitantly satisfies two needs that have associated some quality factors found in an indirect variation relation, and the relative ratio between the corresponding quality factors is greater than a certain threshold.

The rules for analyzing the improvement involve two configurations. The rules 2 and 3 express improvement over a certain quality factor, while the last one expresses a good relative variation between two quality factors related through an indirect variation.

The examples in the table do not include rules based on relation between services. It is clear that such rules bring valid arguments for the provider, useful in its self-deliberating process. But the usability of such arguments on persuading the client is not straightforward, considering that the client reasons only above needs, and not directly above services. Therefore, their integration in the persuasion dialog requires further processing resulting in the change of the references to the services into references to the needs.

As the number of quality factor increases, the number of arguments instantiated from the schema increases too. On each step of the persuasion, the most adequate argument is needed, carrying the most relevant knowledge for the moment. This is not quite straightforward especially due to

	Conditions	*Argument*	*The value*
Examples of instantiation rules for arguments over one single configuration			
1	$\exists q_i, N_i, sq_j \bullet \text{qual}(N_i, q_i, sq_j, C) \wedge \neg \text{ constraint}(q_i)(sq_j)$	$(-, C, \neg \text{ valid\_constraint}(q_i)(sq_j))$	q_i
2	$\exists q_i, N_i, sq_j \bullet \text{qual}(N_i, q_i, sq_j, C) \wedge \text{constraint}(q_i)(sq_j)$	$(+, C, \text{ valid\_constraint}(q_i)(sq_j))$	q_i
3	$\exists q_i, N_1, sq_j \bullet \text{qual}(N_1, q_i, sq_i, C), i = 1..2 \wedge$ $(q_1 > q_2 \in A_{client} \wedge \text{indirect\_variation}(q_1, q_2) \vee$ $sq_1 > sq_2 \in A_{provider} \wedge \text{indirect\_variation}(sq_1, sq_2) \wedge$ $\Delta q * \Delta sq^{-1} < k$	$(-, C, \text{relative\_variation}(q_1, q_2, sq_1, sq_2))$ A good value of a quality factor involves a bad value for the other quality factor. When the ratio between them is in some user defined limits, the configuration is acceptable; otherwise is rejected.	*confl*
Examples of instantiation rules for arguments over comparison of two different configuration			
1	$\exists C_1 \bullet \text{ qual}(C_1) > \text{qual}(C_2)$	$(+, obj, \text{qual}(C_1) > \text{qual}(C_2))$	*qual*
2	$\exists C_1 \bullet \exists N_i, q_i, sq_j^1, sq_j^2 \bullet$ $\text{qual}(N_i, q_i, sq_j^1, C1) \wedge \text{qual}(N_i, q_i, sq_j^2, C_2)$ $\wedge\, sq_j^1 >> sq_j^2$	$(+, obj, \text{better}(q_i, q_j^1, q_j^2))$ One configuration assures a better quality factor for satisfying the same need.	q_i
3	$\exists C_1 \bullet \exists N_i, q_i, sq_j^1, sq_j^2 \bullet$ $\text{qual}(N_i, q_i, sq_j^1, C1) \wedge \text{qual}(N_i, q_i, sq_j^2, C_2)$ $\wedge\, sq_j^2 >> sq_j^1$	$(-, obj, \text{better}(q_i, q_j^1, q_j^2))$	q_i
4	$\exists C_1 \bullet \exists q_i, N_{i,}, sq_i^1, sq_i^2, i = 1..2 \bullet$ $\text{qual}(N_i, q_i, sq_i^j, C_j), j = 1..2 \wedge$ $(q_1 > q_2 \in A_{client} \wedge \text{indirect\_variation}(q_1, q_2) \vee$ $sq_1^1 > sq_2^1 \in A_{provider} \wedge \text{indirect\_variation}(sq_1^1, sq_2^1) \wedge$ $sq_1^2 > sq_2^2 \in A_{provider} \wedge \text{indirect\_variation}(sq_1^2, sq_2^2)) \wedge$ $\Delta q_1 > 1 \wedge \Delta q_2 < 1 \wedge \Delta sq_1^1 > 1 \wedge \Delta sq_2^2 < 1 \wedge \Delta sq_1^2 > 1 \wedge \Delta sq_2^2 < 1 \wedge$ $\Delta q_1 * \Delta q_2^{-1} > 1 \wedge \Delta sq_1^j * \Delta sq_2^j > 1, j = 1..2 \wedge$ $\Delta sq_1^1 * \Delta sq_2^{1^{-1}} > \Delta sq_1^2 * \Delta sq_2^{2^{-1}}$	$(+, obj, \text{better\_rel\_variation}($ $q_1, q_2, sq_1^1, sq_2^1, sq_1^2, sq_2^2))$ The ratio between quality improvement of a factor and the declination of another factor is greater for the configuration C1 than for C2.	*confl*

TABLE 1. INSTANTIATION SCHEMA

the existence of too parties, with different knowledge, and interests, willing to disclose only as much knowledge as needed. As underline in [13], existence of a selection method imposes, either restricting the conditions of instantiation rules, either expressing criteria to pick up the right argument. In both cases, the selection must handle the changes met in the context.

III.NEGOTIATION

The goal of the negotiation is to increase the global utility in a limited time, using quality factors modeled in quality ontology. In order to have a finite sequence of exchanged messages resulting in at least one acceptable solution for the client and the provider, the latter should follow a strategy based on following rationals [10]:

- Is there a reason to believe that the offer will be accepted by the other agent?
- Is there a reason to believe that the offer is the best suited for him?
- Is there a reason to believe that for now, there is no better offer suited to its own interest and also to what he believes to be the other's interest?
- Is there a reason to comply with a cooperative policy or to a self-interest policy?

In the absence of the world model including the observation about the opponent, in case the value system of the client and of the provider are quite different, it is very probable, for intensively constraint request, that the dialog does not results in a solution. The client would attack all the proposals, and the provider could even skip to choose a solution that would be high valuated by the client. Therefore the context awareness imposes itself as a strong requirement.

The client is not interested in the service itself, but in the satisfaction of its needs, the guaranteed properties and the requested sacrifice. Therefore, the client agent reasons only

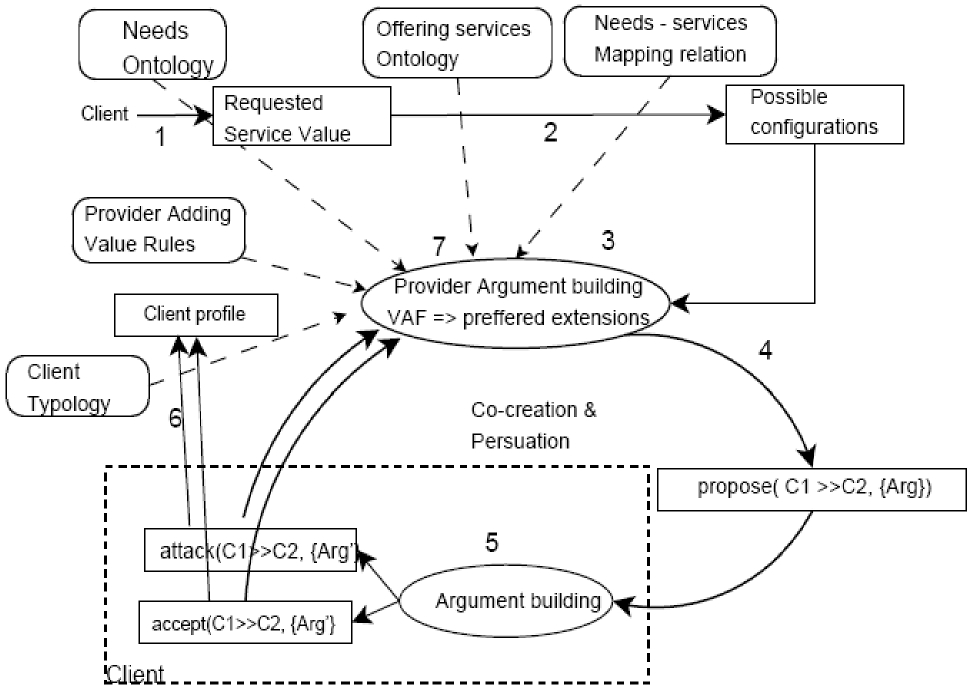

Figure 3. Overview on the service configuration building.

over the needs, without having direct access to the knowledge about services and their interdependencies.

The disagreement between parties is solved through persuasion; therefore the exchanged arguments will always attack each other or they will support independent and non conflicting configurations, but there will never be the case of two consecutive arguments supporting the same configuration. Each party can agree to the configuration proposed by the other or it can (i) attack the configuration itself or (ii) attack by proposing a new configuration that improves some quality factor.

The main steps involved in achieving the service bundle are drawn in Fig 3:

1. the client makes a request utterance, including the description of the request in terms of needs with quality constraints. The expectation model of the client includes also some preferences between the basic dimensions and also between elements of these dimensions of the quality, but they are not communicated to the service provider.
2. the provider generates possible service bundles based only on functional aspects - this is not deeply discussed here due to our focus on quality issues
3. the provider uses its instantiation schema for generating arguments and then it finds admissible sets that reflect only its audience, The instantiation schema is applied in the current context (no information about the client is available yet)
4. the provider chooses a possible solution from the solutions supported by arguments in the admissible set and makes a proposal utterance to the client, initiating the persuasion dialog
5. the client searches for pro and counter arguments for the proposal, issuing the corresponding utterances
6. the provider uses the ground facts from the arguments uttered by the client in order to augment the client profile
7. a new instantiation of the VAF for the current context results in new admissible sets for the provider, this time influenced not only by its own preferences but also by the perceived preferences of the client. In the case of an attack, the provider tries first to counter-attack (attack the client's argument) and if this is not possible, it tries to change the proposal. The step from one configuration to another can be controlled by certain marketing policies, the service provider trying to solve the client's complains.
8. the dialog ends when (1) the client can not attack anymore, in which case the last proposal is considered acceptable, and after a human client validation, it is accepted and the service bundle is actually issued, or (2) the provider can not counter-attack or improve its proposed solutions, in which case there is no acceptable service bundle.

We state that the client agent should submit requests in a certain order, starting with the most wanted. The business game is played by implicit rules of self interest - if the client is happy with a non-perfect solution, generally there is no reason to give him the perfect one. Therefore, the client starts by stating the highest expectations and in case these prove to be non achievable, he gives up to some of them, relaxing the request, until he gets a satisfactory solution or cancels the service request.

When searching for a new configuration to propose, the provider can use three strategies: (1) propose all the time the best offer from its point of view, (2) propose one of the acceptable configuration for the request, (3) propose the worst. Considering that the preference of the provider are different than that of the client, the second solution seems to be good enough, the other one needing the building, on the provider's side, of an order of the offered configurations. Due to the different possible requests, this phase can not be done a priori by the provider, therefore it is better to avoid the complexity involved in building all the acceptable proposals followed by their ordering. We underline again that the provider aims to find a solution mutual agreed, not necessarily the best one.

IV. RELATED WORK

Persuasion viewed as a communication intended to induce belief through arguments is frequently used in the Multi Agent System field, especially for the cases of agents with different opinions. A persuasion dialog protocol between agents seeking and granting authorization to access some information sources is presented in [13] and [6]. Each party gives all the counter arguments for the received argument. If one agent can not counter attack anymore, the other agent is considered to be successful. The role of the context is increased in our approach, the provider aiming not simply to attack the client's argument, but to increase his knowledge by understanding the reasons of the client's behavior. In the cited paper, the scope of the persuasion is to convince the other agent about the right of accessing some information, meaning that the object of persuasion is static. In our approach, the object of persuasion is dynamic, the configuration in discussion being build along the dialog. Even more, the preferences of both parties can themselves evolve as the information about the context changes.

Enhancement of the attack semantic is discussed in [19]. According to the internal structure of argument, the attack can be on premises, reasoning rule, or conclusion. In our approach, each party initiates on each step of the dialog a self deliberating process that sees arguments only based on a simple attack relation (based on different signs for the same object) and proposed classes of audiences. This way, we benefit from the simplicity of attack's semantics from VAF, while still differentiate between types of attacks according to the corresponding dimensions (quality, conflict, needs). Moreover, the ordering relation between values rises from ontology, liberating the client and the provider from the burden of specifying it in an explicit way. As future work, we aim to benefit more from ontological knowledge about the values that are promoted by arguments. We consider that in this way, we can facilitate integration of different and partially known opinions.

As for the service bundling, [1], [2] lays the foundation of the e3-value model, stating and structuring the needed knowledge. They offer a new perspective about service, different from that of web services; the classic composition of web services is replaced by the bundling of services responding to complex needs. We take the same approach, but we focused on creation of a flexible framework allowing the client and the provider to collaborate in order to build the appropriate bundling.

V. CONCLUSION

The present paper aims to offer guidelines for building a framework where a real manager of services can collaborate with clients in order to build high quality services packages satisfying some requests. The main aspects addressed are related to (i) the static knowledge modeling, (ii) the instantiation schema for arguments, (iii) the reasoning above these arguments, and (iv) the dialog between the client and the provider. All of them should be context aware and the whole framework should support an accurate understanding of the context.

Further work includes refining all the aspects outlined in this paper. Moreover, a special attention is required for identification of other argument instantiation rules and their exact context dependency. Also, the selection of the argument with the greatest persuasion power needs to be defined, similar to identification of critical questions. We will also analyze the joint between argumentation and ontological knowledge aiming integration between parties using different ways of valuating the quality of a service.

REFERENCES

[1] H. Akkermans, Z. Baida, J. Gordijn, N. Pena, I. Laresgoiti, A.Altuna: "Value webs: Ontology-based bundling of real-world services". *IEEE Intelligent Systems* 19, 2004

[2] Z. Baida, J. Gordijn, H. Akkermans, H. Saele, A. Morch: "Finding e-service offerings by computer-supported customer needs reasoning". *International Journal of E-Business Research 1*, 2005

[3] T. Bench-Capon, S. Doutre, P. Dunne: "Audiences in argumentation frameworks". *Artificial Intelligence* 171, 2007

[4] A. Bondarenko, F. Toni, R. A. Kowalski: "An Assumption-Based Framework for Non-Monotonic Reasoning". *Logic Programming and Non-monotonic Reasoning, Proceedings of the Second International Workshop,* Portugal, 1993

[5] C. Chesnevar, J. Mcginnis, S. Modgil, I. Rahwan, C. Reed, G.S. Simari, M. Vreeswijk, G. Willmott: "Towards an argument interchange format". *Knowledge engineering review, 21(4),* 2006

[6] S. Doutre, P. McBurney, L. Perroussel, J.M. Thevenin: "Arguing for gaining access to information". *6th International Joint Conference on Autonomous Agents and Multiagent Systems,* USA, 2007

[7] P. M. Dung. "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games". *Artificial Intelligence,* 77(2), 1995.

[8] G. Governatori, M. Maher, D. Billington, G. Antoniou: "Argumentation semantics for defeasible logics". *Journal of Logic and Computation,*, 14(5), 2004

[9] D. Greenwood, M. Lyell, A. Mallya, H. Suguri: "The IEEE FIPA approach to integrating software agents and web services". *6th International Joint Conference on Autonomous Agents and Multiagent Systems,* USA, 2007

[10] R. Lin, S. Kraus, J. Wilkenfeld, J. Barry: "An automated agent for bilateral negotiation with bounded rational agents with incomplete information". *17th European Conference on Artificial Intelligence,* Italy, 2006.

[11] S. Lamparter, A. Ankolekar, R. Studer, S. Grimm: "Preference-based selection of highly configurable web services". *16th International World Wide Web Conference,* Canada, 2007

[12] A. Malucelli, D. Palzer, E. Oliveira: "Ontology-based services to help solving the heterogeneity problem in e-commerce negotiations". *Electronic Commerce Research and Applications,* 5, 2006

[13] L. Perrussel, S. Doutre, J. Thevenin, P. McBurney: "A persuasion dialog for gaining access to information". *International Workshop on Argumentation in Multi-Agent Systems,* USA, 2007

[14] A. Rahwan: "An Argumentation based approach for practical reasoning". *5th International Joint Conference on Autonomous Agents and Multiagent Systems,* Japan, 2006

[15] T. Skylogiannis, G. Antoniou, N. Bassiliades, G. Governatori, A. Bikakis: "DR-NEGOTIATE - A system for automated agent negotiation with defeasible logic-based strategies". *Data and Knowledge Engineering,* 63(2), 2007

[16] C. Sierra, J. Debenham: "The LOGIC negotiatin model". *6th International Joint Conference on Autonomous Agents and Multiagent Systems,* USA, 2007

[17] S. Stein, T. Payne, N. Jennings: "An effective strategy for the flexible provisioning of service workflows" *Huang, J., Kowalczyk, R., Maamar, Z., Martin, D., Mueller, I., Stoutenberg, S., Sycara, K., eds.: Service Oriented Computing: Agents, Semantics and Engineering. LNCS* 4504. Springer-Verlag, 2007

[18] V. Tosic, B. Pagurek, K. Patel: "WSOL - A Language for the Formal Specification of Classes of Service for Web Services". *International Conference on Web Services,* USA, 2003

[19] A. Z. Wyner, T. J. M. Bench-Capon: "Towards an Extensible Argumentation System". *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 9th European Conference,* Tunisia, 2007

[20] M. Aiello, F. Rosenberg, C. Platzer, A. Ciabattoni, S. Dustdar: "Service QoS composition at the level of part names". *Bravetti, M., Nunes, M., Zavattaro, G., eds.: Web Services and Formal Methods. LNCS 4184. Springer-Verlag,* 2006