Collaboration behavior enhancement in co-development networks

Shadi, M.

Publication date
2017

Document Version
Other version

License
Other

Link to publication

Chapter 5

Behavior-based trust evaluation of VO partners

The material of this chapter is partially covered in the following papers:

- Afsarmanesh, H., Sargolzaei, M. and Shadi, M. (2015). Semi-automated software service integration in virtual organisations. *Enterprise Information Systems, Taylor&Francis*, 9(5-6), pp.528-555.

- Afsarmanesh, H., Sargolzaei, M. and Shadi, M. (2012). A framework for automated service composition in collaborative networks. In *Collaborative Networks in the Internet of Services,* 13^{th} *IFIP WG 5.5 Working Conference on Virtual Enterprises* (pp. 63-73). Springer.

5.1 Introduction

To support the success of collaboration in VOs, it is necessary to monitor, supervise, and coordinate the planned activities, in order to discover and/or predict potential risks of failure in their fulfillment. Our approach contributes to developing novel methods and mechanisms for monitoring both individual behavior and collective behavior of partners against some norms defined in VOs (see Chapters 3 and 4). Based on the results of norm monitoring, it is possible to evaluate the partners' trust level, which is considered in this thesis as an important factor for risk prediction in VOs. The factors that are considered for potential risk in VOs in our research is low trustworthiness, insufficient communication, and heavy workload (see Chapter 6). Therefore, one advantage of partners' trust evaluation is to assist the coordinator with dynamically identifying which one of the currently running tasks might involve risks.

The other advantage of trust evaluation is to enhance the service selection in Service Oriented Architecture (SOA)-based VOs, in order to create integrated value-added services. To enhance their business opportunities, organizations involved in many service industries are increasingly active in pursuit of both on-line provision of their business services and collaborating with others in joint initiatives addressed in VOs. However, Collaborative Networks (CNs) in service industry sector face many challenges related to sharing and integrating their collection of provided business services as well as their corresponding software services. At present, due to the lack of uniform unambiguous formal definitions for the implemented business services at the VBE, and ignorance the role of partners' behavior, SMEs involved in VOs cannot be properly supported with the discovery of existing services or facilitation of their composition. In other words, currently the discovery of most-fitting services can only perform a search/match based on the service interface, e.g. service names and operation names, and in some rare cases, if they are available, also searching on some limited semantics related to the functionality of the requested service, e.g. modelled in terms of preconditions, assumptions, post conditions and effects [99].

This chapter concentrates on measuring the trust level of each VO partner, acting as an agent, within a VBE. Evaluating the partners' trustworthiness during the VO operation phase is an important question that has been ignored in the literature. To the best of our knowledge, the previous works, such as [69], [70] and [61] focus on trust evaluation at VO creation phase. However, we define trust as a fuzzy norm, which is monitored during the VO operation phase. As mentioned in Chapter 3, for monitoring the partners' behavior, we propose a VO supervisory framework in which four kinds of norms, i.e. socio-regulatory norms, co-working norms, committing norms and controlling norms are introduced. Based on the results of monitoring the socio-regulatory norms, co-working norms, and committing norms, as well as the score of individual collaborative behavior (ICB) of partners in the VBE (see Chapter 2), the trust level of each partner is calculated. Here, the trust level of an agent shows how much it can be trusted for performing its assigned tasks in the VO. Trust is considered to have multi-dimensional aspects, however the focus of our concern here and thus our proposed approach is on the work behavior of agents. For the purpose of reasoning on the trust level of agents, a fuzzy norm is defined in our framework, which is also monitored by Norm Monitoring Component (NMC) introduced in Chapter 4. Considering that in our addressed environment, some of the criteria influencing the trust measurement are uncertain, the trust requirements are specified as fuzzy norms. It means that VO coordinator can define a minimum acceptable level for agent's trustworthiness, based on the type of the VO.

As a proof of concept for an approach to trust measurement of VO partners, this chapter presents a software service-developing VO, in which a new competency model is introduced for VBE organization members, to support the selection of best-fit services to be potentially composed for creation of new value-added services. This competency-model encompasses (i) the novel meta-data introduced in [6] addressing formal uniform definition of services provided by VBE organizations, (ii) the quality aspects of the provided service, and (iii) the service providers' trustworthiness. Here, a key issue, rarely taken into account in service industry is the role of service providers' behavior-based trustworthiness in service selection, which plays a major role in our service discovery. The conducted research in [98] shows that if a service provider is trustworthy, the likelihood of service delivery with high quality increases. To the best of our knowledge, our introduced competency model for VBE members in service industry is novel and beyond the state of the art in this area.

This chapter is structured as follows. Some related works for trust in VOs is addressed in Section 2. How to evaluate the trustworthiness of VO partners is addressed in Section 3. Section 4 discusses the role of agents' behavior in SOA-based Virtual Organizations. Finally in Section 5, some conclusive remarks are addressed.

5.2 Related Work

Large number of research addresses trust measurement in different disciplines. The focus of our research is on trust evaluation of partners in virtual organizations, and there are some research more closely related to our approach. These mostly address trust in multi-agent systems, and peer-to-peer systems.

The work done in [88] measures the trustworthiness of an agent based on a fuzzy model for which the aggregated recommendations (called weighted trustworthiness value), the agent credibility, and the opinion weights are fed into the system as inputs. After completion of the business interaction with the trusted agent, the credibility value of the recommending agents that deliver their opinions about the trusted agent is adjusted by applying the CCCI (Correlation, Commitment, Clarity, and Influence) methodology [88]. In [50], the trustworthiness of an agent is represented by a probability-certainty density function and then the trusting agent updates the trustworthiness of referrers comparing their claims about trusted agents with actual experiences of the trusting agent. In [13], the notion of promise is used to interpret trust between agents. As such if a promise is expected to be kept in future then its promiser is trustworthy. Then, an experimental test is repeated and other agents' experiences are used to determine the truth of a hypothesis of an expected trustworthiness, using Bayesian interpretation. To establish this expectation, the outcomes "promise kept" and "promise not kept" are considered. They however consider just two states of interaction/promises in their approaches. Our approach is a combination of both objective evaluation and subjective evaluation, and it uses many more states of promises to calculate agents' norm obedience degrees.

In [58], a data set is built of agent interactions and trust values, and divided into training and test data sets. Then the update value of trust is learned based on the identified commitment operations. In the work presented in [58], similar to our work, a number of different states for commitments are considered for trust estimation, but the approach needs to define and run many different tests in order to learn before it can assess trustworthiness of agents.

A fuzzy trust model for peer-to-peer systems in which peers stand for computers is also implemented in [48]. Each peer uses its information of the interactions that it has had with another peer, and tracks the number of successful and unsuccessful interactions for each dimension, in order to model the trust of that peer. Related to virtual organizations, mostly there are two factors fed into the proposed trust models, i.e. organization's capability, as well as its reputation and credibility in the networked virtual organization [70] and [61]. In [70], to evaluate the VO partner's trustworthiness, a fuzzy system is designed in which rules are defined based on the VO goals as well as factors of concern such as processing time and the average of delays in deadlines. This approach is not appropriate for virtual organizations that have several goals and sub-goals and it also needs large involvement of VO decision maker to design the fuzzy inference.

In our proposed approach, we use fuzzy comprehensive evaluation method to combine a number of our introduced hierarchical factors, in order to measure the trust levels for the VO partners. The factors used in our approach to evaluate trust of agents are different from the factors considered in [64] and [63] and to the best of our knowledge, they are novel for assessing the VO partners.

5.3 Trust Evaluation

Establishing trust among partners is primarily rooted in the individual behavior of the involved partners. Therefore, monitoring the behavior of organizations involved in the VO provides good indications for their trust measurement. As mentioned in Chapter 3, the behavior of organizations involved in the VO is constrained by socio-regulatory norms, committing norms, co-working norms and controlling norms. Trust requirements of VO partners are specified as fuzzy norms, called trust-related norms, which are a kind of controlling norm specified by the VO coordinator (see Chapter 3). In a normative environment like a VO, norms can have different levels; norms at level zero are triggered by external events, whereas norms at other levels are triggered in case of certain violations of agents' norms inside the VO [94]. In our framework, if one of the agents' committing norms or socio-regulatory norms are violated then that agent's trust-related norm is triggered. In other words, the trust-related norm is placed at the upper level of the socio-regulatory norms and committing norms. To monitor the violation of trust-related norm, it is necessary to evaluate the current agent's trust level by the Trust Evaluating Component.

To evaluate the overall trust of each agent in a VO, as the main factors, both its Individual Collaborative Behavior (ICB) recorded at the VBE and its work behavior in the current VO are taken into account (see Figure 5.1).

The first factor is ICB of the agent in VBE, as discussed in details in Chapter 2. For measuring it, the following four specific quality-behavioral dimensions are considered: integrity, courage, agreeableness, and openness. A quantitative causal approach is applied that inter-relates some known factors about trust assessment from the environment with the traits related to these four behavioral dimensions. The Formulas derived from the causal relationships are then used to measure ICB for each organization in the VBE, in comparison to all other VBE members.

The second factor is related to the work behavior of an agent in the current VO, which has a vital role in its trust evaluation. On one hand, in [13], the notion of promise is used to interpret trust between agents. It states that if a promise is expected to be kept in future, then its promiser is trustworthy. On the other hand, in [88] the trustworthiness of an agent is measured based on a fuzzy model for which the aggregated recommendations, the agent credibility, and the opinion weights are fed into the system as inputs. However, in our approach, we evaluate the agent's work behavior as the second factor in trust evaluation based on both the norm obedience degrees and the recommendations of other agents about the behavior of that agent in its joint responsibilities. It means we consider three sub-factors, the so-called middle-level factors, i.e. the agent's Committing Norm Obedience Degree (CNOD), the agent's Socio-regulatory Norm Obedience Degree (SNOD), and the agent' s Cooperative Traits (CT) to evaluate this factor. Furthermore, Interaction Rate (IR), Co-work Quality (CoQ), and Not Being Opportunistic (NBO) are the three sub-factors considered for the CT. These are called low-level sub-factors, which are evaluated based on the aggregation of the recommendations received from other agents that are involved in joint-promises with the agent in the current VO. To the best of our knowledge, an approach to agent's trust evaluation is novel.

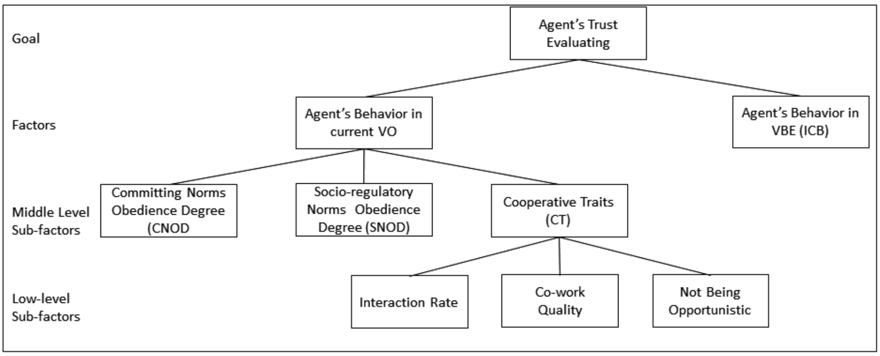

Figure 5.1: Factor hierarchy for the agent's trust evaluating.

There are a number of reasons that justify why we apply the AHP Fuzzy Comprehensive Evaluation method for our trust assessment approach. Firstly, applying AHP, both the quantitative and qualitative factors can be taken into account. Secondly there are some factors/sub-factors, which can themselves be evaluated through the hierarchical structure of the AHP (Analytic Hierarchy Process) [84]. Finally, several characteristics, such as the implicitness in specifying the belief, and the willingness of the trusted agent, result in fuzziness and dynamism of trust [29].

The AHP method allows us to utilize fuzzy mathematical principles to synthesize various factors influencing a certain element, in order to evaluate the merits and demerits of that element [63]. At first, this method determines the factor set of the evaluating element, and the evaluation set according to which the membership degree of each factor is calculated to build a fuzzy evaluation matrix. Then a weight vector is established for evaluation factors (applying AHP) and finally a comprehensive evaluation to various factors is made. The detailed steps of AHP Fuzzy Comprehensive Evaluation method for evaluating the trust level of agents in the current VO include the following:

(1) Determination of factor set affecting the evaluation of the agents' trust level.

Figure 5.1 illustrates the factor hierarchy for evaluation of the trust level of an agent. The value of ICB is placed in an interval [0,1], while the value of CNOD, and CNOD are placed in [-1,1]. As mentioned before, to measure low-level subfactors for an agent, we use the recommendations of other agents who were involved in the joint-promises with the agent, which are aggregated and normalized in an interval [0,1].

(2) Determination of fuzzy evaluation set, named E.

Our proposed fuzzy evaluation set for trustworthiness is defined based on Marsh's view [65]. Marsh defined three levels for trust: distrust, untrust and trust, as Figure 5.2(a) shows. In this point of view, trust of an agent in another agent represents the beliefs of the first agent that the second agent will cooperate with other agents even in situations where the second agent has the opportunity to defect. Distrust is the beliefs of the first agent that the second agent will act against the best interests of others. The distance from distrust to trust corresponds to untrust showing that the second agent is not reliable to cooperate with, in spite of being trusted. As Figure 5.2 (a) illustrates, the cooperation threshold indicates the starting point of trust. Considering Marsh's view, the proposed evaluation set of trustworthiness is defined, as follows:

$$E = \{e_1 = HighDistrust, e_2 = MediumDistrust,\\ e_3 = LowDistrust, e_4 = LowTrust, e_5 = Medium\\ Trust, e_6 = HighTrust\}.$$

In our evaluation set, shown in Figure 5.2 (b), the space between trust and distrust is named Low Trust. We use trapezoidal fuzzy membership function, which is defined below:

$$Trapezoid(x;a,b,c,d) = \left\{ \begin{array}{cc} (\frac{x-a}{b-a}) & if\ a \leq x \leq b, \\ 1 & if\ b \leq x \leq c, \\ (\frac{d-x}{d-c}) & if\ c \leq x \leq d, \\ 0 & \text{otherwise} \end{array} \right\} \tag{5.1}$$

where, x is a real value, and a, b, c, d represent left end point, left center point, right center point, and right end point of the trapezoidal fuzzy membership functions, respectively. For example, $e_2(x)$= Trapezoid (x;-0.8,-0.6,-0.4,-0.2), where e_2 is the second membership function shown in Figure 5.2 (b). It should be noticed that e_1 is a special case of trapezoidal function, which is called R-functions with parameters $a = b = -\infty$, and also e_6 is a special case of trapezoidal function, which is called L-functions with parameters $c = d = +\infty$ [81].

(3) Determination of weights for factors and sub-factors. To determine the weights of factors and sub-factors, we have applied Analytic Hierarchy Process (AHP) [84]. As initialization stage in Algorithm 2 shows, to determine the criteria weights vector, a paired comparison matrix (matrix A) is initiated. The matrix A is an $n \times n$ matrix, where n is the count of evaluation criteria considered to be compared. Assume that each a_{jk} indicates the relative importance of the j^{th} factor to k^{th} factor, so there are three possibilities:

- $a_{jk} > 1$, if the j^{th} factor is more important than the k^{th} factor
- $a_{jk} < 1$, if the j^{th} factor is less important than the k^{th} factor
- $a_{jk} = 1$, if the two criteria have the same importance

There are two constraints: $a_{jk} \times a_{kj} = 1$ and $a_{jj} = 1$ for all j.

Table 5.1 indicates the suggestive numerical scales for relative importance between any two factors [84], based on which the comparison matrix A is established. After building the matrix A, a normalized matrix is derived from A, by making the sum of the entries on each column equal to 1. Therefore, each entry in column i is divided by the sum of the entries of that column. Ultimately, the criteria weight vector $W_{1\times n}$ is constructed by averaging the entries on each row of the normalized matrix (see normalization stage in Algorithm 2).

Once several paired comparisons are done, some inconsistencies may appear. To illustrate it with an example, assume that a decision maker wants to evaluate 3 criteria. In his evaluation the second criterion is more important than first criterion, and the third one is also more important than the second one. There is an inconsistency in his evaluation if he also judges the third criterion as equally or slightly important than the first one. An effective technique is embedded in AHP which is responsible to check the consistency of the judgments made by the expert. As consistency checking stage in Algorithm 2 shows, at first, to compute the Consistency Index (CI), it is needed to calculate the Principal Eigen value

Algorithm 2: AHP for assigning weights to the criteria

```
Input:
  • n, the number of criteria
  • RI, random consistency index
Output:
  • W, a weight vector
/* Initialization                                                    */
Get matrix A_{n×n}   // The entry of the matrix a_ij shows the
   comparison between criterion i and criterion j
/* Normalization                                                     */
for j := 1 to n do
    SC[j] := sum elements in column j of matrix A    // SC is an
       array
end
for j := 1 to n do
    for i := 1 to n do
        a_ij := a_ij / SC[j]    // This is the normalized matrix A.
    end
end
/* Weight Calculation                                                */
for i := 1 to n do
    SR[i] := sum elements in row i of normalized Matrix A    // SR is
       an array
    W[i] := SR[i] / n
end
/* Consistency Checking                                              */
λ_max := Σ_{i=1}^{n} W[i] × SC[i]
CI := (λ_max − n) / (n − 1)
if CI/RI < 0.1 then
    return W
else
    Get new comparison matrix A and start again    // Notice that A
       was inconsistent and a new A is needed.
end
```

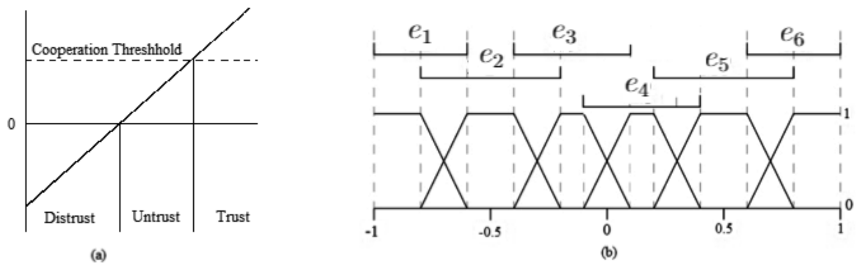

Figure 5.2: (a) Different concepts of trust based on Marsh's view [65]. The negative interval refers to distrust, while the positive interval started at the cooperation threshold refers to trust. The space between these two sections is undistrust. (b) The proposed Fuzzy Evaluation Set to evaluate trust level of agents in VOs. It is established based on the Marsh's view. We chose a new name, i.e. Low Trust, for the space between distrust and trust. $E = \{e_1 = HighDistrust, e_2 = MediumDistrust, e_3 = LowDistrust, e_4 = LowTrust, e_5 = MediumTrust, e_6 = HighTrust\}$.

[84], λ_{max}, which equals to the sum of products between each element of weight vector obtained by the mentioned approach and the sum of columns of the matrix A before normalization. After that, CI is computed as follows:

$$CI = \frac{\lambda_{max} - n}{n - 1}$$

where, n is the number of criteria in Matrix A. Random consistency Index (RI) is proposed by Saaty in [84] which is calculated by finding the average of random consistency indexes of 500 matrices. If $\frac{CT}{RI}$ is equal or smaller than 10% then the inconsistencies can be ignored.

Status	**Value**
Equally important	1
Slightly more important	3
More important	5
Greatly more important	7
Fully more important	9
For comparison between margins mentioned above	2,4,6,8

Table 5.1: Relative importance where comparing every two factors [84].

Generally, the AHP computations are directed by the decision makers' experience, hence AHP can be considered as a means to translate the decision makers' evaluations into a multi-criteria ranking, as Algorithm 3 shows. The following paragraphs explain step by step what the algorithm does.

The examples of relative importance for comparing three sub-factors CNOD, SNOD, and CT, and also for comparing three sub-factors of CT (i.e. IR, CoQ, and NOB) are shown in Figure 5.3 (a) and (b). If the VO coordinator has decided that the weight of CNOD, is 6 times more important than CT, consequently the weight for the CT in contrast to CNOD is $\frac{1}{6}$. Applying AHP approach, the weight of factors/sub-factors are calculated, as shown in Figure 5.3(c). We assume that, the weights for two main factors, i.e. agent's behavior in current VO and agent's ICB, are respectively 0.8, and 0.2.

(4) Determination of evaluation matrix and comprehensive evaluation.

Considering the first *for* statement in Algorithm 3, the evaluation matrix for CT is established as follows:

$$R^{''} = \begin{bmatrix} r^{''}_{1,1} & r^{''}_{1,2} & r^{''}_{1,3} & r^{''}_{1,4} & r^{''}_{1,5} & r^{''}_{1,6} \\ r^{''}_{2,1} & r^{''}_{2,2} & r^{''}_{2,3} & r^{''}_{2,4} & r^{''}_{2,5} & r^{''}_{2,6} \\ r^{''}_{3,1} & r^{''}_{3,2} & r^{''}_{3,3} & r^{''}_{3,4} & r^{''}_{3,5} & r^{''}_{3,6} \end{bmatrix}$$

where, $r^{''}_{i,j}$ represents the membership degree of the i^{th} sub-factor of CT, in e_j (i.e. j^{th} membership function in evaluation set E). As mentioned in the second step, evaluation set E includes six trapezoidal function, specified by the formula 5.1. In other words, the rows of matrix $R^{''}$ represent the membership degrees of IR, CoQ, and NBO in set E, respectively.

As mentioned in the previous step, a weight vector is found for three sub-factors of CT (applying Algorithm 2), which is called $W^{''}$. Note that $W^{''}$ is a vector consisting of three entities representing IR, CoQ and NBO. Therefore, we can measure the comprehensive evaluation of CT called $B^{''}$, as follows (see Algorithm 3):

$$B^{''} = W^{''} \times R^{''} = [b^{''}_1, b^{''}_2, b^{''}_3, b^{''}_4, b^{''}_5, b^{''}_6].$$

Then the evaluation matrix for agent's behavior in current VO, is established as follows (see the second *for* statement in Algorithm 3):

$$R^{'} = \begin{bmatrix} r^{'}_{1,1} & r^{'}_{1,2} & r^{'}_{1,3} & r^{'}_{1,4} & r^{'}_{1,5} & r^{'}_{1,6} \\ r^{'}_{2,1} & r^{'}_{2,2} & r^{'}_{2,3} & r^{'}_{2,4} & r^{'}_{2,5} & r^{'}_{2,6} \\ b^{''}_1 & b^{''}_2 & b^{''}_3 & b^{''}_4 & b^{''}_5 & b^{''}_6 \end{bmatrix}$$

where, $r^{'}_{i,j}$ represents the membership degree of the i^{th} sub-factor of agent's behavior in current VO in e_j (i.e. j^{th} the membership function in evaluation set E). In other words, the first two rows of matrix $R^{'}$ represent the membership degrees of CNOD, and SNOD in set E, respectively, and the third row of $R^{'}$ is $B^{''}$, obtained in previous step. Considering the weight vector found for sub-factors of

Algorithm 3: Fuzzy Comprehensive Evaluation Method for Trust

Input:
- x, the agent for which the trust level is evaluated
- IR, a single value denoting Interaction rate of agent x
- CoQ, a single value denoting Co-working Quality of agent x
- NBO, a single value denoting Not being opportunistic for agent x
- CNOD,a single value denoting Committing Norm Obedience Degree of agent x
- SNOD, a single value denoting Socio-regulatory Norm Obedience Degree of agent x
- ICB, a single value denoting Individual Collaborative Behavior of agent x
- $W_{1\times2}$, a weight vector for two main factors
- $W'_{1\times3}$, a weight vector for CNOD, SNOD, and CT
- $W''_{1\times3}$, weight vector for IR, CoQ, NBO
- $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, Fuzzy evaluation set. `//` $\mu_{e_i}(z)$ `is specified by Formula 5.1.`

Output:
- B, the comprehensive evaluation vector

for i := *1 to 6* **do**
  $R''[1, i] := e_i(IR)$
  $R''[2, i] := e_i(CoQ)$
  $R''[3, i] := e_i(NBO)$
end
$B''_{1\times6} := W''_{1\times3} \times R''_{3\times6}$;
for *i:=1 to 6* **do**
  $R'[1, i] := e_i(CNOD)$
  $R'[2, i] := e_i(SNOD)$
  $R'[3, i] := B''[i]$
end
$B'_{1\times6} := W'_{1\times3} \times R'_{3\times6}$;
for *i:=1 to 6* **do**
  $R[1, i] := B'[i]$
  $R[2, i] := e_i(ICB)$
end
$B_{1\times6} := W_{1\times2} \times R_{2\times6}$;
return B

	CNOD	SNOD	CT
CNOD	1	4	6
SNOD	$\frac{1}{4}$	1	3
CT	$\frac{1}{6}$	$\frac{1}{3}$	1

(a)

	IR	CoQ	NOB
IR	1	$\frac{1}{3}$	$\frac{1}{2}$
CoQ	3	1	3
NOB	2	$\frac{1}{3}$	1

(b)

Factors	Weights	Medium-level factors	Weights	Low-level factors	weights
Behavior in current VO	0.8	CNOD	0.7		
		SNOD	0.21		
		CT	0.09	IR	0.16
				CoQ	0.59
				NOB	0.25
ICB	0.2				

(c)

Figure 5.3: An example of relative importance among trust factors and sub-factors as well as weights calculated applying AHP.

agent's behavior in current VO, W', we measure the comprehensive evaluation B' for it, as follows:

$$B' = W' \times R' = [b'_1, b'_2, b'_3, b'_4, b'_5, b'_6].$$

Finally, the evaluation matrix R for trust is established as follows (see the third *for* statement in Algorithm 3):

$$R = \begin{bmatrix} b'_1 & b'_2 & b'_3 & b'_4 & b'_5 & b'_6 \\ r_1 & r_2 & r_3 & r_4 & r_5 & r_6 \end{bmatrix}$$

where, r_j represents the membership degree of agent's ICB in e_j (i.e. j^{th} the membership function in evaluation set E). Considering the weight vector found for main factor of trust evaluations, W, we measure the comprehensive evaluation for Trust, B, as follows (see Algorithm 3):

$$B = W * R = [b_1, b_2, b_3, b_4, b_5, b_6]$$

(5) Analysis of the fuzzy comprehensive evaluation. The final results, i.e. vector B, can be analyzed through defuzzification to be transformed into a single number. Mostly, the maximum degree of membership, which regards the maximum value of vector B as the overall value of the evaluating element, is used for defuzzification. The work done in [37] shows that weighted average grade of membership produces more reasonable evaluation results, while using the maximum degree may lead to the loss of a lot of information. The weighted average method is the ratio of the sum of the average point in each membership function multiplied to its membership degree (the corresponding value in vector B) to the sum of membership degrees for all entries of vector B, as formula below shows [81]:

$$T = \frac{\sum_{i=1}^{6} B(i) * z(i)}{\sum_{i=1}^{6} B(i)} \tag{5.2}$$

where, $z(i)$ shows the average point of the i^{th} membership function in set E. T is placed in [-1,1], because the interval in which the evaluation set is defined, is [-1,1]. The negative values of T show distrust area and its positive values show trust area in Figure 5.2 (b). In our approach, trust level, which is necessary for an agent, can be treated differently based on the type of the VO. In other words, in each VO, the VO coordinator defines the minimum acceptable trust level. For example, in one VO it may be sufficient to have a medium trust level for agents, while in another one it may be necessary to fulfill the high trust level for all agents.

Example

Assume that at time t_1 the value of factors and sub-factors for an agent are as follows: Interaction Rate (IR) = 0.3, Co-work Quality (CoQ) = 0.7, Not Being Opportunistic (NBO) = 0.2, CNOD= -0.2, SNOD = 0.1, ICB= 0.8. The rows of matrix R'' are the membership degrees of IR in set E, membership degrees of CoQ in set E, and membership degrees of NBO in set E, respectively. Then, considering the weights shown in Figure 5.3 (c) and applying Algorithm 2, the comprehensive fuzzy evaluation for CT is as follows:

$$B'' = W'' \times R'' = \begin{bmatrix} 0.16 & 0.59 & 0.25 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} =$$
$$\begin{bmatrix} 0 & 0 & 0 & 0.33 & 0.375 & 0.295 \end{bmatrix}$$

The rows of R' are membership degrees of CNOD in set E, membership degrees of SNOD in set E, and B'', respectively. The comprehensive fuzzy evaluation for agent's behavior in current VO is calculated at the next step, as follows:

$$B' = W' \times R' = \begin{bmatrix} 0.7 & 0.21 & 0.09 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.33 & 0.375 & 0.295 \end{bmatrix} =$$
$$\begin{bmatrix} 0 & 0 & 0.7 & 0.239 & 0.033 & 0.026 \end{bmatrix}$$

The rows of matrix R are B' and membership degrees of ICB in set E, respectively. Finally, the comprehensive fuzzy evaluation for trust is:

$$B = W \times R = \begin{bmatrix} 0.8 & 0.2 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0.7 & 0.239 & 0.033 & 0.026 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.56 & 0.19 & 0.026 & 0.22 \end{bmatrix}$$

It shows the trust level of the agent is Low Trust, because applying the weighted average method as a kind of defuzzification [81] method, the crisp value of agent's trust level is 0.134. The average points of six membership functions $e_1, e_2, e_3, e_4, e_5, e_6$ in evaluation set is z=(-0.8, -0.5, -0.15, 0.15, 0.5, 0.8). Therefore, applying the weighted average method on the vector B above, we have (considering Formula 5.2):

$$T = \frac{0.56 * (-0.15) + 0.19 * 0.15 + 0.026 * 0.5 + 0.22 * 0.8}{0.56 + 0.19 + 0.026 + 0.22} = 0.134$$

Assume that, it is required for agents to have at least Medium Trust level. In this situation, the violation of trust-related norm for the agent in this example is 1, because the membership degree of point 0.134 in Medium Trust and High Trust membership functions is zero.

5.4 Agent Behavior in SOA-based Virtual Organizations

In service industries, VOs are usually established to fulfill one of the following two purposes. The first purpose is to target a specific emerged opportunity within the market or society, which requires either the combination of different capabilities and resources provided by a number of different organizations, the simple accumulation of their resource capacities, or both. The candidate SMEs are then usually selected by the VO planner and invited to accept the joint responsibility of fulfilling the tasks needed to achieve the common goals of the VO.

A second purpose for the VO formation is to support innovation. For instance, one or more SMEs together foresee the potential of investing into the development of some new services, and act as the VO planner, targeting the merge of abilities, resources, capacities, etc. from a number of SMEs that can then together fulfill the development of the planned innovation. Consider the following application case. Suppose that there is a VBE in which the partner SME-1 plans to simply create a new integrated tourism package that includes the reservation of flight-tickets, hotel-rooms, day-trips, and dinners at restaurants in a specific touristic region of a country. Furthermore, assume that all of the above mentioned individual business services are already implemented and provided as web-services by several different

SMEs within the VBE. Hence, when SME-1 is able to discover these services and identify the most-fit SMEs with which to work and share services for creating this new package as an integrated service, it may act as the VO planner and start the formation of the newly required VO together with those other SMEs. But at present, due to the lack of such uniform unambiguous formal definitions of the implemented business services at the VBE, and unawareness about the service provider's behavior, the SME-1 can be properly supported with neither the discovery of such existing services nor facilitation of their composition.

5.4.1 SOA-based Virtual Organizations

At present, creating an integrated web service, out of existing services provided by different VO partners, is challenging. This is partially due to the lack of concise definitions of service properties and quality, level of trust in service provider for their developments, as well as the lack of supporting tools and infrastructure for this purpose. Before discussing a new framework for service composition, a new layered view is introduced on Service Composition in VOs, including (a) the Business Specifying Layer, (b) the Service Oriented Computing Layer and (c) the Collaboration Layer, as shown in Figure 5.4. There are two kinds of business

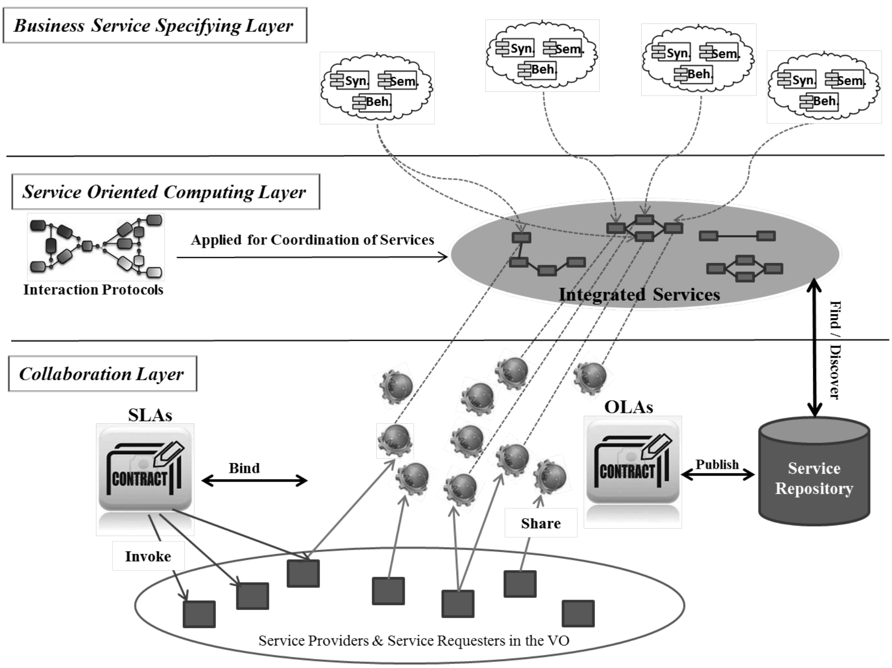

Figure 5.4: A Layered View on Service Composition in VOs [6].

services considered in industry, including the manual services and the software services. In the proposed model, only software services have been taken into consideration, and a basic simple software service is assumed to accompany every manual service consisting of a start and end notifications for that service.

(a) Business Service Specifying Layer

To support business service integration for VOs, the software services need to be both discoverable and integrable. For this purpose, the abstract description of the software services should be extended with supplementary properties, which are captured within a concrete meta-data, presented in "Business Service Specifying Layer", where the meta-data content has been presented as a means to accurately formalize every atomic software service. Every composite business service is then represented by a set of meta-data, each member of which providing the concrete formalization of an atomic service. The introduced meta-data [6] also constitutes a main element within the business service competency, which is defined later, and facilitates service-oriented computing (i.e. service discovery and integration), addressed in the lower layer. To support machine-to-machine service inter-operation in a VO, the meta-data captures the three characteristics of each service, namely its syntax, semantics, and behavior. This meta-data targets the comprehensive capture of all characteristics of the business services, through which their equivalent software services (web services), can be unambiguously developed.

Typically, the syntactic properties of a service are represented by XML-based standards and languages, such as the Web Service Description Language (WSDL) and Simple Object Access Protocol (SOAP) [32]. Conceptual properties of software services, here referred to as semantics are typically defined with an ontology, as an explicit specification of a conceptualization of the knowledge about the service. The service ontology definition, also in the VO context, encompasses a group of vocabularies that specify semantic attributes of services (e.g. goals and category) and their relationships, which together present a meaningful concept about the service [23]. Moreover, it is also needed to formalize the externally observable behavior of each service, which shall represent the proper invocation order of its operations. The selected formal notation to represent this aspect is the Constraint Automaton [11], adopted in order to formalize the execution plan of the business services, when aimed to develop its equivalent software service. The formal notation provided by Constraint Automaton supports the definition of a desired sequence for operations' invocation, needed for automatic execution of services.

(b) Service Oriented Computing Layer

The "Service Oriented Computing Layer" as the second layer, provides mechanisms for service discovery, execution, and integration to support service designers with offering new value-added services, through composition of the existing shared services in the VO. There are multiple approaches developed by the research community in the area of services discovery, but only a few addresses the comprehensive design of a framework encompassing all aspects needed for efficient semi-automated Service Oriented Computing (SOC). Below four categories of approaches, that are more closely related to some aspects of the proposed approach are addressed and discussed in details.

Keyword-based Search. These methods crawl through the syntactical properties published for web services, such as the service name, to conduct matchmaking of relevant web services to the user service requirements. Most of these methods are based on textual descriptions represented in WSDL documents [32]. In order to improve this keyword-based browsing method, some categorization, based on the syntactic descriptions of web services, are proposed. As an example of these methods, in [30] web services are categorized by their service types in order to compose services at the abstract type level, instead of at the instance level. That platform assumes a data type hierarchy, for each domain, defined by domain experts, where the hierarchy only captures the main data types used in that domain. This method is simple and proves to be efficient in certain cases, but it is not comprehensive, because the hierarchy does not reflect the other attributes of services for its categorization.

Semantic-driven Search. These approaches consider as the base the semantics of software service and represent ontology based methodologies used for improving service discovery and composition. Several languages and standards have been proposed to specify semantics of web services in order to assist services matchmaking. The Web Service Modeling Ontology (WSMO) is one of the main proposed approaches to describe web services semantically, and including their functional and non-functional properties, as well as other aspects that are relevant for interoperating with them [72]. The other main ontology-based approach in the context of web services is the OWL-S, which is a service oriented language, and built on top of the Web Ontology Language (OWL). Based on the OWL-S, several research works try to find a match between the requested service and the set of advertised services. For instance, Benaboud et al. have proposed an agent-based approach for Web services discovery, which uses OWL-S to describe both the Web services and the customer requirements [78]. Furthermore, several semantics-enabled extensions of WSDL (Web Services Description Language) have been proposed, such as WSDL-S [99], which annotate WSDL documents with semantic concepts in an intuitive manner.

QoS-aware Search. They apply the definition of Quality of Service (QoS) to the service selection; in order to ensure the optimal service execution plan among the set of discovered candidates. The QoS attributes investigated in this group of research works are oriented toward measuring the software quality aspects such as the execution time, availability, and reliability. In [101] a QoS-aware service evaluation method is presented to select a qualified web service among the set of candidate services matching against certain request, while taking into account the QoS characteristics of these services. During the last years, plenty of research investigations are conducted on QoS measuring approaches as well as on service selection methods based on the QoS measures. For instance, in [87] an egalitarian-based negotiation model is proposed that aims to select a required service by achieving the egalitarian principle between the user's point of view on one side, and the service provider's point of view and the provided features, on the other side.

The proposed approach goes beyond that of the previous works and several challenging aspects of SOC, such as the QoS parameters (our so-called Quality Specification Criteria in Chapter 4), and the service provider's trustworthiness are also addressed. The main concern neglected in the literature is the service provider's trustworthiness. Moreover, in our approach, the complexity of software services is considered as the service behavior, and is defined through the introduced meta-data for services within the proposed framework.

Automation and successful application of service composition at this layer requires not only the rich meta-data provided in the upper layer, but also the coordination of required interaction protocols among the composed web services. Moreover, the behavioral properties of service providers, and the recommending partners are then also deployed within the service discovery approach. Later on, when defining the service competency model of C3Q (including: Capability, Cost, Conspicuity, and the Quality specification criteria) for addressing the VO services, the above meta-data, the trust level of service provider, and QoS of the service, respectively constitutes the Capability, Conspicuity, and Quality specification criteria aspects of the VO service competency.

(c) Collaboration Layer. Finally, the third layer, "Collaboration Layer", includes a pool of software services that are offered by different members/stakeholders of the VO, to support the intended services defined in the upper layer. The shared services are published in a repository or directory according to the specific Operational Level Agreement (OLA) [49]. An OLA is agreed among the VO stakeholders and the composite service provider, to describe the responsibilities of each VO member/stakeholder (service provider) towards the specified composite services. OLAs are also supported by Service Level Agreements (SLAs) among web services [62]. A SLA reflects an agreement between the service provider and the clients of a service to create assurances on the service level at the binding time. As such, the expected performance of a deployed software service is defined at the service level. In VOSAT, expected performance of a service is translated into some Quality Specification Criterion (QSC). After delivering the service, QSCs are also evaluated accompanied by the Committing

Norm Obedience Degree (CNOD), which are applied in trust evaluation of the promiser/service provider. Response time, supported throughput, and service availability are some examples of performance metrics contained in its SLA. In fact, SLAs provide some needed information for developing the new VO service competency model defined in this chapter. The approach adopted for service quality assessment in the proposed model borrows ideas from [92] that monitors the behavior of VO members against some norms for identifying their level of trustworthiness. According to this approach, all agreements in OLA and SLA among the involved partners in the VO are translated into committing and socio-regulatory norms, as mentioned in Chapter 3. With this, at any point of time the trust level of the VO member (constitute by a service provider or a service client) would be reflected upon its claims concerning different characteristics of its provided services, as well as its feedback about others' services. This information is used in the C3Q service competency model defined later in Section 5.4.3.

It should be noticed that service providers share the specifications of their services at the level of VBE, and for creation of each value-added composed service, a VO is formed in which some norms are defined for its partners. Then, the behavior of service providers is monitored against their promises and the related Quality Specification Criteria of services are also evaluated. It means that the CNOD is measured for service providers involved in all VOs formed to compose services. All socio-regulatory norms defined for the service provider in these VOs, are also monitored to measure SNOD for that service provider. The degree of Cooperative Traits (CT) for each service provider is also measured through aggregation of all recommendations of the clients about that service provider's cooperative traits, including Interaction Rate, Co-work Quality, and Not Being Opportunistic in these VOs. The individual collaborative behavior (ICB) of the service provider as a member in VBE is also evaluated based on the causal relationships between four behavioral dimensions and their related traits, as explained in Chapter 2. Consequently, the trustworthiness of the service provider can be evaluated, applying the AHP fuzzy comprehensive evaluation method on criteria, including CNOD, SNOD, CT, and ICB. If a service provider does not participate in past VOs for service composition, the default value 0.5 is assigned to its trustworthiness.

5.4.2 Service Composition Architecture

Monitoring the behavior of organizations involved in the VO and the way they perform their commitments can provide good indications about their claims either related to their own services or recommendations that they make about other agents. For instance, if the VO coordinator is notified in advance about the weaknesses and risks associated with providing some services, then the coordinator can take timely and appropriate strategic actions. For this purpose, we have developed a framework, which enables formalization of agents' responsibilities and commitments to perform their sub-tasks in the VOs, as addressed in Chapter 3.

For SOA-based VOs, every agreement made between a service provider and service requester to deliver a particular service is also translated into a promise in this framework. In other words, the terms in OLA (Operational Level Agreement) are considered as promises in VOSAT. The details of an agreement to deliver a service, which is mentioned in SLA (Service Level Agreement) are considered as Quality Specification Criteria of a promise, which are discussed in Chapter 4 to measure CNOD of each partner. VOSAT evaluates the overall trust value of each agent applying the AHP comprehensive fuzzy evaluation method on some criteria, such as the agent's Individual Collaborative Behavior (ICB) in the VBE, CNOD, Socio-regulatory Norm Degree (SNOD), etc.

Our proposed competency-model, named C3Q, captures the following four aspects of the business service competency: (i) Capability, (ii) Cost, (iii) Conspicuity, and (iv) Quality specification criteria. The service capability is captured through the three elements of its corresponding business service meta-data, namely the syntax, semantics, and behavior aspects. We use the level of agent's trustworthiness in our new service-competency model to support the effective service composition. The trustworthiness of a service provider is considered in service selection as Conspicuity in the proposed competency-based model, while the trustworthiness of recommending agents is considered as a credibility factor in their recommendations about the QoS of a service.

Figure 5.5 shows the framework, in which VOSAT functionality is used for more effective service composition in VOs. Besides VOSAT, this framework consists of six sub-spaces, i.e. Service Modeler, Enhanced Query Editor, Query Processor, Service Search Engine, Service Selector, and Service Integrator, which are discussed in the following paragraphs. Service modeler provides three specifications of syntax, semantics and behavior of a business service. Enhanced query editor represents a query in the format of soft constraint automata [86], which specifies the properties of intended services. Query processor is responsible to process the query aiming at service discovery. Service search engine is responsible for match making the functional properties of service, i.e. syntax, semantics, and behavior. The Service selector chooses one of the services as the results of search engine considering quality criteria of service. Finally, service integrator is responsible for composition of some services to form a new value-added composite service. The main contribution of this research is on service modeler and service selector.

Service Modeler supports the modeling and provision of atomic business services. The specification in this subspace is achieved through using a set of concrete formalisms and standards. From the business service analysis point of view, all services that are intended to be shared within the CN are required to be unambiguously defined to their corresponding meta-data. Obviously, the syntactical meta-data for services are captured and stored in a syntax registry, where most

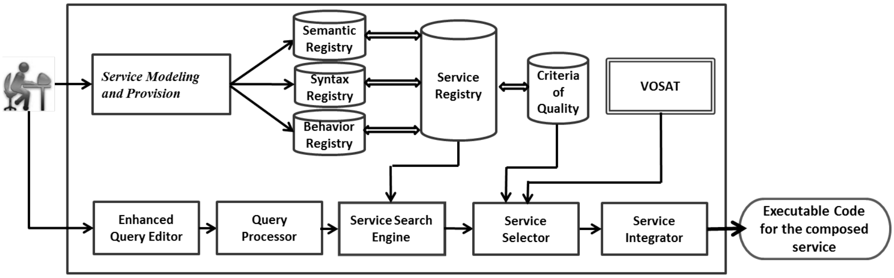

Figure 5.5: The service composition framework in collaboration with VOSAT

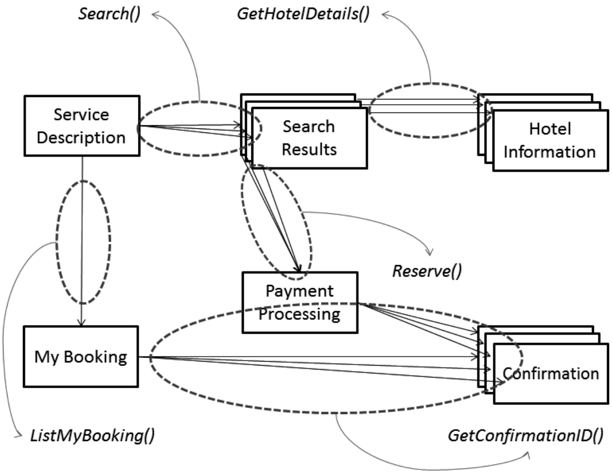

Figure 5.6: The Hotel booking service[86].

activities in the framework, e.g. discovery, adaptation, and execution of services, refer to this registry. Service semantics represented by OWL-S [78] is similarly captured and stored within an ontological registry labeled as "Semantic Registry".

The behavior of the software services is formalized by a constraint automaton. It is, therefore, necessary to generate a constraint automaton for each service, which should be captured and stored within the behavior registry. The behavior registry is needed for services to serve two main purposes: (1) the need to match behavioral aspects of services, for the purpose of service discovery, and (2) to unambiguously generate final executable code for every integrated service. An example of hotel booking service is illustrated in Figure 5.6, in which *getHotelDetails* operation is invoked only after a *search* operation. The behavior of this service is shown by a constraint automaton in Figure 5.7. Figure 5.8 also shows an example of the WSDL extension with the behavior description represented in

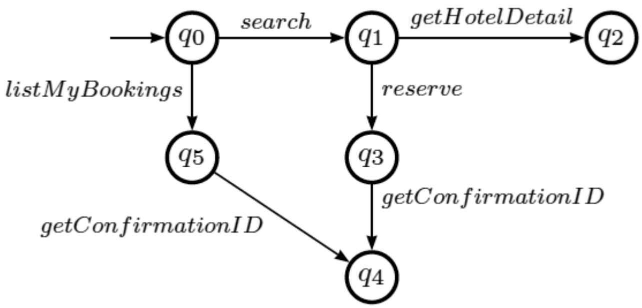

Figure 5.7: The constraint automaton modeling the behavior of the Hotel booking service [86].

```
…
<BWSDL: Behaviour>
        <transition: Name= “Search”, CurrentState= “q0”, NextState= “q1”>
        <transition: Name= “getHotelDetails”, CurrentState= “q1”, NextState= “q2”>
        <transition: Name= “reserve”, CurrentState= “q1”, NextState= “q3”>
        <transition: Name= “getConfirmationID”, CurrentState= “q0”, NextState= “q5”>
        <transition: Name= “getConfirmationID”, CurrentState= “q3”, NextState= “q4”>
</BWSDL: Behaviour>
…
```

Figure 5.8: Example of the WSDL extension by the behavior description [86].

Figure 5.7.

Consider that a member of the VO intends to create a new integrated service composed of a number of services shared by other enterprises, for instance creating a travel package as a new business service developed through the composition of a flight reservation service, an hotel reservation service, etc. As the first step, this user needs to specify his/her planned integrated service, its components (e.g. all constituent business services such as flight reservation, etc.), and the interaction protocols among these components. As the second step, the user needs to discover if the needed atomic services exist in the VO as shared services. Therefore, it needs to search for its required services and then select the best-fit service based on functional and non-functional criteria. Functional criteria are the triple meta-data, while non-functional ones constitute the QoS and the service providers' trustworthiness provided by VOSAT. In other words, service discovery needs a *Service Search Engine* and a *Service Selector*.

The approach to service integration pivots on the coordination of discovered software services. Separation of coordination concerns from computation concerns in exogenous languages like Reo [68] results in easier integration of independent components, such as the shared services in a CN. Reo in comparison with the other current exogenous coordination languages is more mature, and benefits from having several related formal semantics, such as the constraint automaton, and provides tools for analyzing the coordinated services and model checking of the generated Reo systems. Therefore, Reo is selected to model the coordination of the needed interactions among the software services. The details of this implementation architecture is shown in [86].

5.4.3 Competency Model Supporting Discovery

We propose the C3Q model of service competency to facilitate the selection of the best-fit service(s), based on the parameters in the user's query. In [41], the 4C-model of organizations' competency is introduced to enhance the participation of the VBE member organizations in future VOs. The C3Q model of service competency is rooted in the 4C-model of organization competency. In the proposed C3Q service competency model, besides Capability, Cost, and Conspicuity, there is a fourth fundamental characteristic of services named the Quality specification criteria. The definitions of these four aspects of services are provided below:

- **Capability**: it represents the service characteristics that are related to its corresponding meta-data, i.e. the Syntax, Semantics and Behavior of the service.
- **Cost**: it represents the cost of providing the service per user request.
- **Conspicuity**: it means for identifying the validity of information related to a service, as claimed by its provider. In the proposed framework, the trustworthiness of each VO partner is reflected, as calculated by the VOSAT.
- **Quality specification criteria of service**: some of the QoS metrics agreed in the SLA are selected here in the proposed framework, which address the CN concerns in C3Q service competency model. These consist of the response time, availability, throughput, and reliability.

To identify the most-fit software service(s), among the potential set of discovered candidate services, and to rank the resulted services, the C3Q model of service competency is applied. This most-fit service selection process involves two stages of functional and non-functional ranking, as described in the following sub-sections. The proposed service competency model is used for service discovery in which the capability is considered as a functional aspect, while other three elements are used in non-functional ranking.

First Stage- Functional Ranking

In this stage, the capability of services (syntax, semantics and behavior aspects) is only considered to compare against the search parameters specified by the user. Based on the comparison results, certain similarity scores are assigned to each of the discovered software services, indicating its similarity to the user request. These scores express the proximity (match) between capabilities of an intended service (asked by client) and a discovered service registered in the shared service directory. To measure these similarity scores, user specifies the demanded behavior of the service (sequence of operation invocations), as well as the required semantic and syntactic descriptions (e.g., the name of service operation, the data types of the input/output arguments of the operation, etc.) of his/her desired services, within a query. The box labeled as *Enhanced Query Editor* in Figure 5.5 represents the module that allows users to specify their queries based on desired functional (syntactical, semantic, and behavioral) and non-functional properties. These queries are processed by the module represented as *Query Processor* in the planned framework of Figure 5.5, which obtains the user preferences for service discovery purposes. In the next step the query processing is performed by the *service search engine*, which starts to match as much as possible the description of the demanded component services defined within the user's query, against the shared services that are registered within the service directory. This process will then provide to the user a set of discovered services, which are ranked in matching the user query, above certain threshold level. As such, the role of the service search engine is the discovery of the set of potential matching services, among the existing shared services in the VBE, based on their capabilities. For this matching process, the service search engine has to search the syntax, semantics, and behavior registries, as well as the service directory simultaneously, against the user query. For match-making among the requested and provided services, the discovery problem is treated as a soft constraint satisfaction problem, and AI techniques, e.g. Depth First Search, are then applied to solve it. It should be noticed that the implementation details and experiment results of this stage are addressed in [86]. The similarity score obtained in this stage for the software service s_i is called $Score_{Capability}$.

Second Stage- Non-functional Ranking

In this stage, a utility function is defined, which gives a new score to the software services that are leveled in functional ranking. In SLA, each service provider claims certain characteristics about the quality metrics of its provided software services. Following paragraphs address in details how to deal with these metrics and other criteria involved in second stage ranking.

The VO partners that get to use the shared software services, provided by other partners, also participate in evaluating/rating the performance of those services, through rating their quality metrics. As mentioned before, four most relevant quality specification criteria identified for assessment of software services that are shared in VBEs, namely: the response time, availability, throughput, and reliability. Nevertheless, if in one VBE, other quality specification criteria become interesting and needed to be considered, they can be similarly added to the approach defined below:

- Response Time (RT): is the time that a service takes to respond to the invocation requests from the clients. Response time depends on some factors such as the load intensity, and the average response time is usually measured in the scale of milliseconds.

- Availability (A): is defined as the probability that the service can respond to consumer requests. The value of availability is a number in the range [0,1], and can be measured as the percentage of time that a service is operating.

- Throughput (T): represents the rate at which a service can process requests, and is measured as the total number of invocations that can be supported in a given period of time.

- Reliability (R): shows the likelihood of successful invocation of the service. It can be computed as the ratio of the number of error messages to the total number of received messages, both error and successful completion.

The values of four quality specification criteria, which organization k as a VBE member assigns to the shared service s_i, is as follows:

$$(QSC_{k,1}(s_i), QSC_{k,2}(s_i), QSC_{k,3}(s_i), QSC_{k,4}(s_i))$$

where, 1 is for Response Time, 2 is for Availability, and 3 is for Throughput, and 4 for Reliability.

Here two specific aspects related to the QSCs are addressed, first how different values of a QSC are aggregated for each service, and second, how the calculated result is normalized. Assume that the set $S = \{s_1, s_2, \ldots, s_m\}$ denotes the software services discovered in the first stage addressed above in this section. Each software service contains some rating values for its quality specification criteria (i.e. for its: RT, A, T, R). These values are partially collected directly from the service properties who published the services, partially computed through the execution monitoring performed by the VO supervision, and partially gathered from the clients' feedback on the use of the services. But to be sure about the rating values (RT, A, T, R) that are claimed by the recommending organizations, the collected rating values are then multiplied by the trust level of the recommending organization. The trust level of each partner is calculated in reference to its own performance in the VO. This calculation is done by VOSAT, during the VO operation phase. If the trust level of the partner k is in distrust area, zero is assigned to $Trust(k)$, otherwise it equals to the crisp value of its evaluated trustworthiness, applying the weighted average defuzzification method, as discussed in this chapter (see Formula 5.2).

The aggregated value of quality specification criteria for software service s_i is as follows:

$$q_{ij} = 1/n \sum_{k=1}^{n} Trust(k) * QSC_{k,j}(s_i) \qquad\qquad j = 1 \ldots 4 \tag{5.3}$$

where, n is the number of s_i's clients, and q_{i1} to q_{i4} shows the values of RT, A, T, and R for software service s_i, respectively. It should be noticed that the trustworthiness of each recommending agent is updated based on the actual QSC specified after delivering of the service for which this member recommends.

For the m discovered services in the first stage $S = \{s_1, s_2, \ldots, s_m\}$, the following matrix is considered, where the i^{th} row in the matrix denotes the discovered software service i, which consists of the four quality properties, i.e. its RT, A, T, R, which are represented here by q_{i1} to q_{i4}.

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ \cdots & \cdots & \cdots & \cdots \\ q_{m1} & q_{m2} & q_{m3} & q_{m4} \end{bmatrix} \tag{5.4}$$

Before defining a utility function, the matrix Q needs to be normalized. The main reason for normalization is that different dimensions, scales and value ranges are considered for different quality attributes, and they are not uniform. For example, the unit of measurement for throughput is typically invocation/second, for response time is millisecond and for reliability is a percentage. Therefore, to develop the ranking formula, it is needed to first make a uniform measurement for all these service qualities, independent of their measurement units. Furthermore, generating a uniform index for all service qualities also provides an equal weight for all considered criteria, as the starting point for the ranking process. Therefore, all quality attributes are normalized into the same value range of [0,1]. In this approach, the Max-Min normalization approach is applied, which is one of the most widely used approaches, introduced in [28]. Below, in Formula 5.5, the $q^{'}$ normalizes the values for those quality attributes that have positive connotations, such as: the Availability, Throughput, and Reliability, whose values will be scaled as follows:

$$q^{'}_{kj} = \left\{ \begin{matrix} \dfrac{q_{kj} - {}_{i=1}^{m}min(q_{ij})}{{}_{i=1}^{m}max(q_{ij}) - {}_{i=1}^{m}min(q_{ij})} & if\ {}_{i=1}^{m}max(q_{ij}) \neq {}_{i=1}^{m}min(q_{ij}) \\ & 1 \qquad otherwise \end{matrix} \right\} \tag{5.5}$$

where, m is the number of services selected according to the similarity functions explained in the first stage, i.e. functional ranking, and ${}_{i=1}^{m}max(q_{ij})$ and ${}_{i=1}^{m}min(q_{ij})$ respectively show the maximum and minimum values in the entire column j in matrix Q. For example, if three software services are resulted by the first stage ranking, and the value for the reliability attribute of these three selected software services are 10, 13, and 17 percent respectively, then these values are respectively normalized to 0, 0.43, and 1. To normalize the attributes with negative connotation, such as the Response time the formula below is used:

$$q_{kj}^{'} = \begin{cases} \dfrac{{}^{m}_{i=1}max(q_{ij}) - q_{kj}}{{}^{m}_{i=1}max(q_{ij}) - {}^{m}_{i=1}min(q_{ij})} & if\ {}^{m}_{i=1}max(q_{ij}) \neq {}^{m}_{i=1}min(q_{ij}) \\ 1 & otherwise \end{cases} \tag{5.6}$$

For example, if the response time of three software services selected in the first stage, are 400, 450, and 510 milliseconds respectively, then these values are respectively normalized to 1, 0.54, and 0. Finally, after the normalization, the transformation matrix $Q^{'}$ is defined as follows:

$$Q^{'} = \begin{bmatrix} q_{11}^{'} & q_{12}^{'} & q_{13}^{'} & q_{14}^{'} \\ q_{21}^{'} & q_{22}^{'} & q_{23}^{'} & q_{24}^{'} \\ ... & ... & ... & ... \\ q_{m1}^{'} & q_{m2}^{'} & q_{m3}^{'} & q_{m4}^{'} \end{bmatrix} \tag{5.7}$$

Furthermore, when a VBE member requests a software service, it can also specify the preference requirements for the quality metrics. These preferences are considered to be specified for RT, A, T, R attributes of the service, and they form a weight vector, such as $W = \{w_1, w_2, w_3, w_4\}$, where $\sum_{j=1}^{4} w_j = 1$. So, the comprehensive quality value for each software service s_i, considering that user specifies the preference vector, is as follows:

$$Score_{Quality}(s_i) = \sum_{j=1}^{4}(w_j * q_{ij}^{\prime}),\ 1 <= i <= m \tag{5.8}$$

where, m is the number of services selected according to the similarity functions explained in the first stage. In the system implementation, and considering the defined C3Q service competency model, for identifying the most-fit service(s) among those that are discovered, and labeled as partially matching the user request, four different scores are generated for each service to define a utility function for new ranking of selected services. The first score, which is the similarity score calculated in functional ranking stage, is called $Score_{Capability}(s_i)$. The second score is focused on the non-functional ranking, is calculated as explained in the second stage above in this section, and is represented as the $Score_{Quality}(s_i)$. The third score, $Score_{Conspicuity}(s_i)$, is related to the provider's trustworthiness which is calculated based on the ratio of the provider's trustworthiness to the maximum trust level among other service providers whose services are considered in the ranking. In Section 5.3, the trust level of each VO partner is measured, applying AHP fuzzy comprehensive evaluation method. If the trust level of a partner is in distrust area, zero is assigned to $Trust(provider s_i)$, otherwise it equals to the crisp value of trustworthiness, as discussed in this chapter. The fourth score is focused on the cost associated with using each software services, and is represented as $Cost(s_i)$. Cost should also in turn be normalized. Considering that Cost has also a negative connotation, Formula 5.6 is used to normalize it.

Finally, the overall score as a kind of utility function, used for ranking the set of matched software services against the user request, is calculated as follows, applying weighted sum model:

$$\begin{aligned} OverallScore(s_i) = \alpha \times Score_{Capability}(s_i) + \beta \times Score_{Quality}(s_i) + \\ \gamma \times Score_{Conspicuity}(s_i) + \delta \times Cost(s_i) \end{aligned} \tag{5.9}$$

Here $\alpha, \beta, \gamma, \delta$ are respectively the weight coefficients, which can be specified by the users, in order to emphasize/de-emphasize the importance of certain criteria/parameter within the formula for overall score calculation, and where $\alpha + \beta + \gamma + \delta = 1$. It should be noticed that this function can be optimal if we find appropriate values for weights, for example through applying machine learning approaches.

Assume that we have a query that searches for software services that return the "weather" forecast for a location indicated by the name of a "city". The details of this example is discussed in [86]. Table 5.2 shows top-five ranked experiment results, where the other web services obtained a similarity score, $Score_{Capability}$, less than 0.3. The columns of Table 5.2 respectively show the name of web service, capability score calculated based on comparing the meta-data of the web service with the query, quality score, conspicuity score, overall score and finally ranking of the services. The last column of Table 5.2, represents the final ranking of these services, for α=0.6, β =0.2, γ = 0.1, and δ = 0.1. It is also assumed that all services are shared free in the VBE.

Table 5.2: The ranking of the matched Software services[86]

Name of WS	Capability Score	Quality Score	Conspicuity Score	Overall Score	Ranking
globalweather	0.69	0.92	0.5	0.648	1
usweather	0.54	0.34	0.625	0.454	4
Weather	0.5	0.88	0.125	0.488	2
WeatherWS	0.48	0.50	1	0.488	3
usweather	0.44	0.34	0	0.332	5

5.5 Conclusion

This chapter addresses how AHP fuzzy comprehensive evaluation method is applied to evaluate the trust level of VO partners. This method combines a number of relevant factors into the modeling of an element, and applies fuzzy mathematical principles to evaluate them. It should be noticed that due to the fuzzy nature of trust, the trust requirements related to the VO partners, are implemented as fuzzy norms. For this purpose, the VO coordinator defines a minimum required and/or tolerated trust level for partners in the VO. This means, if for example the trust level of a partner is lower than the minimum level tolerated in the VO, that partner's trust-related norm is then violated, indicating the lack of required trustworthiness of that partner.

This chapter also addresses a new approach to improve software service selection as a part of service discovery, applying our new service competency-model for CNs. The proposed approach is aimed to rank the identified registered services, according to their similarity/fitting score with the requirements specified in the query provided by the user. For identifying the most-fit service(s) among those matched for their functional similarity, the user's preferences on the criteria of service quality, such as the availability, reliability, etc. are considered, as well as the cost of using the service. The trustworthiness of the service provider calculated by VOSAT, is also considered as important input in the automated selection of most-fit services. The organization's trustworthiness is used as the "weight" multiplied by any quality rating for the services that are shared in the VBE, provided by its members, and is specified as an element in the competency-model of VBE members' services to be used for selection by others.