VE architect-driven partner selection: ontology-based approach

Alireza Khoshkbarforoushha*

Department of IT Engineering,
Tarbiat Modares University,
P.O. Box 14115-111, Tehran, Iran
E-mail: a_khoshkbarforoushha@sbu.ac.ir
*Corresponding author

Masoumeh Tajvidi and Mehrnoush Shamsfard

Department of Electrical and Computer Engineering,
Shahid Beheshti University,
1983963113, Tehran, Iran
E-mail: m.tajvidi@webmail.sbu.ac.ir
E-mail: m-shams@sbu.ac.ir

Mohammad Aghdasi

Department of IT Engineering,
Tarbiat Modares University,
P.O. Box 14115-111, Tehran, Iran
E-mail: aghdasim@modares.ac.ir

Abstract: An underlying formation of instant virtual enterprises (VE) is the matter of flexible and efficient partner selection. In rapidly changing business environment and turbulent market conditions, selecting the partners for (re)configuration of VE must be realised regarding many factors such as skill, cost, resource, availability, goal, etc. However, these attributes primarily are not constant and may not be appropriate for some networks or conditions; hence they should be instantly replaced with some new ones. This imposes a demand that the VE architect selects partners in terms of changing criterion. Thus, we have to establish an adaptable and flexible partner selection mechanism that is driven by the VE architect's requirements. To achieve this, the paper proposes an ontology-based partner selection algorithm to effectively calculate the semantic similarity between VE architect's requirement ontology and partners' ontologies. The algorithm comprises diverse metrics including gravitation of resources (GoR), path similarity, path weight, and definition similarity which make it more close to VE architect. The paper also discusses implementation and evaluation of it using different approaches which confirm the quality, efficiency, and generality of algorithm.

Keywords: ontology-based partner selection; instant virtual enterprise; architect-driven semantic partner selection; semantic matchmaking.

Reference to this paper should be made as follows: Khoshkbarforoushha, A., Tajvidi, M., Shamsfard, M. and Aghdasi, M. (2013) 'VE architect-driven partner selection: ontology-based approach', *Int. J. Networking and Virtual Organisations*, Vol. 12, No. 4, pp.283–309.

Biographical notes: Alireza Khoshkbarforoushha received his MSc in Information Technology Engineering from Tarbiat Modares University, Tehran, Iran. He is a member of Automated Software Engineering Research (ASER) Group and Service-Oriented Enterprise Architecture (SOEA) Lab at Shahid Beheshti University. His research interests include cloud computing, big data analytics, and service-oriented computing.

Masoumeh Tajvidi received her MSc in Computer Software Engineering from Shahid Beheshti University. She is a member of Service-Oriented Enterprise Architecture (SOEA) Lab at Shahid Beheshti University. Her research interests include cloud and service-oriented computing.

Mehrnoush Shamsfard received her BSc and MSc both on Computer Software Engineering from Sharif University of Technology, Tehran, Iran. She received her PhD in Computer Engineering-Artificial Intelligence from AmirKabir University of Technology in 2003. She has been an Assistant Professor at Shahid Beheshti University from 2004. She is the Head of NLP Research Laboratory of Electrical and Computer Engineering Faculty. Her main fields of interest are natural language processing, ontology engineering, text mining, and semantic web.

Mohammad Aghdasi is an Associate Professor of the Department of Information Technology Engineering at Tarbiat Modares University, Tehran, Iran. He received his BSc in Engineering from Sharif University of Technology, Tehran, Iran, in 1981, MSc in Management Engineering from University of Electro-Communication, Japan, in 1986, and his PhD in Management Science Engineering from University of Tsukuba Institute of Socio-Economic Planning, Japan, in 1989. His current research interests include business process management and business process reengineering.

1 Introduction

An underlying formation of instant virtual enterprises (VE) is the matter of efficient, quick, and flexible partner selection. In rapidly changing business environment and turbulent market condition, brokerage and partners search is an important activity in the creation phase of a virtual enterprise, where the most adequate consortium of enterprises should be selected to respond to a given business opportunity.

Partner search can be based on a number of different information sources, being private, public, or independent (Camarinha-Matos et al., 2003). But even when this information source is available, for some business opportunities it might be necessary to look for new partners when new skills or more resources are necessary (Camarinha-Matos et al., 2000). Moreover, during the operation of a VE it might be necessary to replace some partners or change the roles of some others (Camarinha-Matos et al., 2003). In dynamic or instant VE, partner selection is the responsibility of a leader who is one of the enterprise members while forming a VE (Chen, 2008). We call him VE architect.

In selecting the partners for the (re)configuration of virtual enterprise, scholars (Petersen and Divitini, 2002; Rocha and Oliveira, 1999; Wu and Su, 2005) enumerate many factors such as skill, cost, resource, availability, goal, etc., that should be taken into account. However, these attributes primarily are not constant and may not be appropriate for some networks or conditions; hence they should be instantly replaced with some new factors. Plisson et al. (2007) emphasise that one of the most important characteristics that changes from one network to another is a set of partners' competencies, representing the ability of the network (or its subset) to participate in particular types of projects and to perform specific tasks. This imposes a requirement that the VE architect selects partners in terms of changing criterion. Thus, we have to establish an adaptable and flexible partner selection mechanism that driven by the VE architect requirements. In this regard, Grefen et al. (2009) stress the importance of automated tools that help VE architect to determine appropriate members for creation of an instant VE. In a similar way, some years later authors of Camarinha-Matos and Afsarmanesh (2003) emphasised that lack of appropriate support tools for partner search and selection as one of the obstacles in the early phase of instant VE planning and creation.

To meet the above-mentioned demand, this article proposes an ontology-based partner selection algorithm and related tools and techniques to effectively select the most appropriate partners within a network. The proposed algorithm is, indeed, a semantic matchmaking method, since it can play a vital and effective role in partner selection in VE (Camarinha-Matos and Afsarmanesh, 2003; Huang et al., 2007; Wang et al., 2008).

The proposed partner selection algorithm identifies the best partner through semantic similarity measurement between VE architect's requirement ontology and partners' ontologies. Requirement ontology is the representation of a request using ontology languages that capture consensual knowledge of requirements in a formal way. Indeed, it specifies the expected competencies of desired partner.

To be more specific, after taking VE architect's requirements (using business rule language) and translating them into corresponding ontology (the translation techniques and mechanism has thoroughly been discussed in Khoshkbarforoushha and Aghdasi (2009), the algorithm tries to find the partner that its ontology match with the expressed requirement. Therefore, there is a key assumption in the proposed approach that is every partner in the network defines and organises relevant knowledge about activities, processes, organisations, skills, competencies, etc., using OWL-DL (W3C: OWL Web Ontology Language Overview, 2009) ontology language. In reality, such an assumption is trivial, since in the last decades many projects aimed at creating ontologies concerning the domain of VE including Collaborative Network Organization (The CNO Ontology, 2009) ontology, TOronto Virtual Enterprise ontology (TOVE) (Fox, 1992).

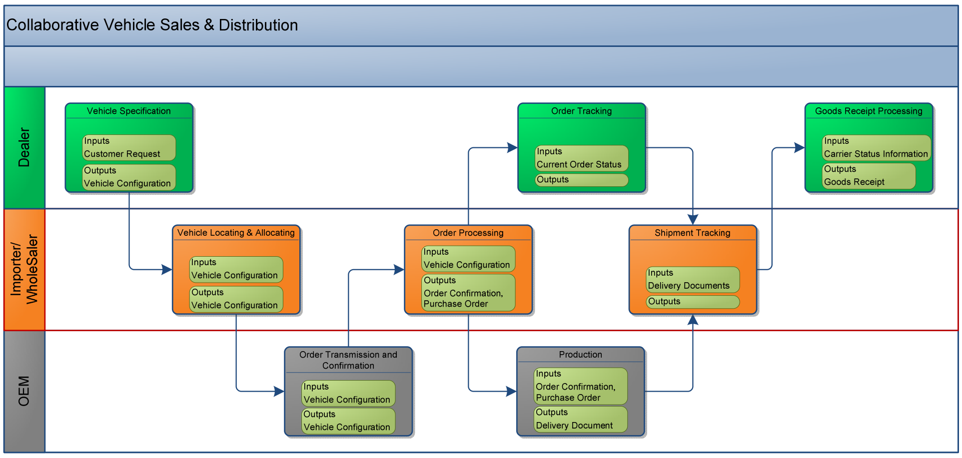
The rest of the paper is organised as follows. Section 2 presents a practical scenario which exhibit the motivation of our work. Section 3 explores the related work in both semantic matchmaking and ontology-based partner selection algorithms. We present the algorithm framework and its related phases and metrics in Section 4. We also put the developed algorithm to the preliminary test by applying it to a case from the automotive industry. In Section 5, firstly we discuss the proof-of-concept prototype system that supports the approach. Secondly, the algorithm is evaluated using two different approaches, and finally, we investigate its time complexity and performance. We next sum up the discussion in Section 6 and provide some conclusions in Section 7.

2 Motivating scenario

Sales and distribution processes in the automotive industry are highly flexible and complex. They are subject to change due to new regulations and the increasing need to improve customer satisfaction in the vehicle buying experience (SAP Solution Composer, 2006).

As process mapping (i.e., Figure 1) indicates three different companies – an OEM, an importer, and a dealer – must have seamless collaboration to respond to a customer's request for a new vehicle.

Figure 1 Collaborative vehicle sales and distribution business process (see online version for colours)



In reality, this business process must be supported with various importers, dealers as partners within a network of organisations. Meanwhile, highly flexible and changing environment causes the combination of partners to execute such a business process alters over and over. Therefore, the VE architect requires a flexible process for identification and selection of best partners to meet a new business opportunity.

3 Related work

Since our algorithm contributes and connects two research areas including semantic-partner selection and ontology-based matchmaking, this section reviews the studies related to them to clarify how much the developed algorithm advances these research areas.

3.1 Recent trends in VE partner selection

For selecting partners Hsieh and Lin (2012) proposed an architecture based on combinatorial reverse auction mechanism. To minimise the cost of virtual enterprise, they develop algorithms to find a near-optimal solution, and implement a prototype system based on this algorithms and web service technologies to verify the effectiveness of the proposed methodology. The partner selection problem is formulated based on combinatorial reverse auction and the Lagrangian relaxation technique for solving the problem.

The partner selection problem is investigated with the consideration of environmental protection and two 'green criteria', carbon emission and lead content in manufacturing production by Zhang et al. (2012). In this work, the partner selection problem formulated with green criteria, and a new algorithm, Pareto genetic algorithm (Pareto-PSGA), is proposed. Compare to other intelligent algorithms, the Pareto-PSGA shows high performance in solving the problem.

For solving the partner selection problem (Niu et al., 2012) an enhanced ant colony optimiser (ACO) which has better results in search accuracy and computing time was proposed. Another novel cultural algorithm which is based on PSO (CPSO) was proposed that its objective is minimising total project cost and time. The simulation results show that this novel algorithm is better than regular PSO (Wei and Bu, 2012).

A business correlation model including both quality correlation and selection correlation is proposed by Wu et al. (2013). They also presented an approach for correlation-driven QoS aware optimal service selection which is based on genetic algorithms and by the empirical study; the efficiency and effectiveness of the proposed approach are demonstrated.

In formation of a dynamic cloud collaboration platform, the cloud partner selection is investigated by Hassan and Huh (2013). They proposed a multi-objective optimisation model while considering individual information and past relationship information with collaboration cost optimisation among cloud providers. Also a framework, called MOGA-IC is implemented in their work.

3.2 Related work in ontology-based matchmaking

A body of work has been reported under the theme of ontology-based semantic matchmaking. In this area, there are many research papers that investigate similarity measures within one ontology. Apart from the papers in this category, scholars between the years from 2000 to 2005 developed many mapping and matching system such as PROMPT (Noy and Musen, 2000), Cupid (Madhavan et al., 2001), COMA (Do and Rahm, 2002), GLUE (Doan et al., 2003), S-Match (Giunchiglia et al. 2005). Based on the pros and cons of these systems that have been studied and compared in Huang et al. (2007), none of them could meet the requirements of the partner selection application which are cited in Sections 1 and 2. Therefore, this section reviews the closest and newest work in ontology-based matchmaking.

In Paolucci et al. (2002), authors showed how service capabilities are presented in the profile section of a DAML-S description and how a semantic match between advertisements and requests is performed. In their work the degree of match is determined by the minimal distance between concepts in the taxonomy tree, and differentiates between four degrees of matching: exact, plug-in, subsumes and fail. In Lei and Ian (2003), authors introduce DAML + OIL based matchmaking, which uses a DL reasoner to compare ontology-based service descriptions. They also extend the four degrees: exact, plug-in, subsumes intersection and disjoint. Both of these methods primarily consider the simple subsumption between the concepts in the ontology, and ignore their detailed semantic difference.

In Shu et al. (2007) proposed a semantic matchmaking algorithm that contemplates both subsumption and definition distances for the purpose of matching service requesters and providers. Even though their work presents an interesting algorithm for semantic matchmaking, but it has two shortcomings. Firstly, it utilises simple subsumption distance approach which solely relies on subtraction of level of concepts in a hierarchy. Secondly, their definition similarity metric is appropriate for measuring similarity of classes within a same ontology. This point will be discussed in more detail in 4.3.5.

In Billig et al. (2007), authors proposed a framework for semantic matching based on enterprise ontologies. Enterprise ontologies are used as a basis for determining the relevance of information with respect to the enterprise. Their approach is the integration of point set distance measures with a modified semantic distance measure for pair-wise concept distance calculation. They also combine measures to determine the intra-ontological distance between sub-ontologies. However, their work ignores definition similarity, path similarity, and path weight measurement between resources.

3.3 Related work in semantic-based partner selection

There are limited research papers in semantic-based partner selection. In Wang et al. (2008), authors presented a comprehensive semantic-based resource allocation framework to enhance the matchmaking process. Their framework employs semantic reasoning techniques select the eligible resource candidates. They also provided a bidding to further optimise the resource selection according to runtime conditions such as duration, cost and, etc. However, the main metric for conceptual matchmaking is the distance between concepts of ontologies. In other words, some parameters such as concept definition, path type between resources, etc., have been neglected. Moreover, the measures have not been empirically evaluated.

In Huang et al. (2007), authors introduce a compatibility vector system, created upon a schema-based ontology-merging algorithm, to determine and maintain ontology compatibility, which can be used as a basis for businesses to select candidate partners with which to interoperate. Their algorithm just provides a basis for e-business partner selection through considering concepts distance in taxonomic relationships. Thus, their work neglects path similarity, concept definition, etc., since their algorithm aim to figure out heterogeneity of independently designed ontologies.

3.4 Concluding remarks

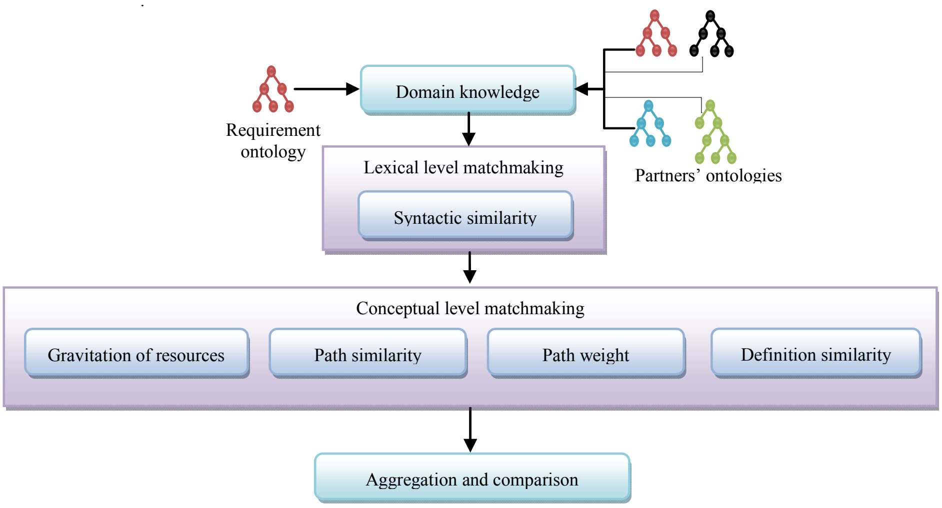
Exploration of ontology matchmaking and semantic partner selection studies provide some salient conclusions for us. Every semantic matchmaking algorithm utilises some metrics on the basis of its target demand or application. This lead to that some of them neglect the important metrics such as defenition similarity, path similarity, etc., in their caculations. On the other hand, the limited proposed semantic partner selection approaches are not close to the end users (e.g., VE architects) and their requirements. Thus, these algorithms could not take and reflect their priorites and preferences. Moreover, they do not incorporate a holistic view in which many metrics can be utilised and make the selection process more accurate and precise.

4 Ontology-based algorithm for partner selection

4.1 The algorithm framework

Matchmaking can be divided into two categories including syntactic/lexical level and semantic/conceptual level matchmaking. In the context of this paper, matchmaking is defined as the process of searching possible matches between requirement ontology and partner ontology. The proposed ontology-based partner selection algorithm consists of three phases including Lexical level matchmaking, conceptual level matchmaking, and aggregation and comparison, Figure 2.

Figure 2 Ontology-based partner selection framework (see online version for colours)



First of all, both requirement ontology and partners' enterprise ontology, as inputs are inserted into the framework. In fact, it is expected to find the best partner who satisfies the requirement as much as possible. At the first phase, the framework measures syntactic similarity of resources (i.e., concepts or concept instances) between two ontologies. Thereafter, at the second phase, the resulted sets, which are the outputs of syntactic similarity analysis, examined via semantic-based techniques including gravitation of resources (GoR), path similarity, path weight, and definition similarity. At the third phase, Conceptual similarity values are compared in order to identify the qualified partner. Following, subsections reveal details of each phase.

4.2 Lexical level matchmaking

First phase of the proposed algorithm involves finding similar resources between requirement ontology and partners' enterprise ontology via syntactic similarity measurement. Syntactic matchmaking is the process of finding similar resources between ontologies through syntactic similarity measurement of their labels. Since similarity is the inverse of distance (Rhee et al., 2007), syntactic similarity measurement is the inverse of string distance measurement. There are a number of string distance measurement metrics such as Jaro (1995), and Winkler (1999), etc. Our framework calculates syntactic similarity of resource (i.e., concept or concept instance) labels between requirement ontology and partners' enterprise ontology using Jaro-Winkler (Cohen et al., 2003) metric.

The Jaro-Winkler measure τ: $S \times S \rightarrow [0\ 1]$ is as follows:

Let s and t refer to strings.

$$\tau(s,t) = \tau_{Jaro}(s,t) + P \times Q \times \frac{\left(1-\tau_{Jaro}(s,t)\right)}{10}$$

Such that P is the length of common prefix and Q is a constant.

The reason of utilising this metric is due to good results that have been recorded for the metric in the literature. Besides, Jaro-Winkler metric seems to be intended primarily for short strings (Cohen et al., 2003). Since resource (i.e., concept or concept instance) labels are almost short strings, Jaro-Winkler metric is an appropriate choice for the framework purposes.

Regarding every partner could be the one who satisfies requirement, syntactic and semantic matchmaking analysis must be done for each of them independently. Hence, after lexical level matchmaking between requirement and partners' ontologies, there is an output set for each partner. The output set incorporates the syntactically similar resources between requirement and partner ontology. We will elaborate more about this issue in 4.6.1.

4.3 Conceptual level matchmaking

Second phase of thealgorithm framework deals with conceptual or semantic matchmaking between ontologies. The developedalgorithm comprises different metrics including GoR, path similarity, path weight, and definition similarity.

4.3.1 GoR measurement

Definition 1: GoR: GoR between two resources (class or class instance) say x, y is defined as the reverse of shortest path length (SPL) between x and y.

$$GoR(x\ ,y) = \frac{1}{SPL(x,y)}$$

The GoR is defined on the basis that in ontologies the semantic relevance between resources decreases while the distance between them increases. The GoR term is inspired by the gravitation force between planets, in which when they are further from each other, they are less attracted towards each other. The GoR concept is inspired by the work of (Paolucci et al., 2002), in which the authors emphasise the degree of match is determined by the minimal distance between concepts in superclass-subclass hierarchy, which is also known as a taxonomy (Horridge et al., 2004). As discussed in related work, most of the work in this area are focused on the acquisition of taxonomical relationships and often neglect the importance of inter-linkage between concepts.

However, we generalise this notion for the whole ontology including the relationship between classes and instances and also inter-linkage between concepts. This is why there is huge number of relationships in partners' ontologies which are defined using *Object properties*. This kind of relationships is known as non-taxonomic relationships.

GoR concept for measuring semantic relevance between resources may not be true all the time. In other words, there may be some exceptions in which GoR does not guarantee the semantic relevance between resources. Therefore, GoR concept for measuring semantic relevance may reduce the exactness of matchmaking, but our algorithm enjoys path similarity to eliminate such exceptions as much as possible.

For GoR calculation,

Let,

- R_i, R_j: refers to two resources
- d_{ij}: refers to SPL between R_i and R_j on the condition that i is less than j.

$$SPL\left(R_i, R_j\right) = \begin{cases} d_{ij} & \text{If there is at least a path between } R_i \text{ and } R_j. \\ \infty & \text{If there is no path between } R_i \text{ and } R_j. \end{cases} \quad (1)$$

$$GoR\left(R_i, R_j\right) = \frac{1}{SPL\left(R_i, R_j\right)} (\quad (2)$$

In equation (1), if there are some paths between R_i and R_j, the SPL is considered, but if there is no path between these resources, infinity is considered as SPL value. Based on GoR definition, equation (2) calculates the degree to which two resources are relevant to each other. In case of more than one shortest path between two resources, our algorithm picks one of them randomly.

4.3.2 Path similarity measurement

Definition 2: Path similarity: path similarity between two paths say $P_{x,y}$ and $P_{z,w}$ is defined as the number of similar relations (edges) between these paths divided by the maximum length of the paths. Two relations are similar when they have same relation type and properties.

Relation type (t) in OWL is categorised as follows:

- t_1: relation between classes that is subClassOf
- t_2: relation between class and its instances that is type
- t_3: inter-relation between concepts that is objectProperty.

It should be noted that some important relation types such as *hasPart* or *isPartOf* is in the third category so that is defined in the form of *objectProperty* relation type.

Object properties in OWL may have four kinds of properties or characteristics (c) as follows:

- c_1: functional
- c_2: inverse functional
- c_3: symmetric
- c_4: transitive.

Now, for path similarity calculation,

Let,

- P_{R_i,R_j}: Refers to the shortest path between resources R_i and R_j on the condition that i is less than j. Similarly, P_{R_z,R_w} is a shortest path between R_z and R_w.
- $L(P_{R_i,R_j})$: Returns the length of P_{R_i,R_j}.
- r_k: Refers to k^{th} relation on a path between two resources. r_k is itself a path with the length of 1.
- t_m: Refers to relation type. t_m could be t_1, t_2 or t_3.
- c_n: Refers to relation properties including c_1, c_2, c_3, and c_4.
- $r(t_m, c_n)$: Refers to a relation r with the type t_m and also property set c_n.

$$P_{R_i,R_j} = \bigcup_{k=i,i+1,\ldots,j} \left\{ r_k\left(t_{m_k}, c_{n_k}\right)\right\} \tag{3}$$

Such that $m_k \in \{1, 2, 3\}$, $n_k \in \{1, 2, 3, 4\}$,

$$\left(P_{R_i,R_j} \cap P_{R_z,R_w}\right) = \bigcup_{f,q\in I} P_{R_f,R_q} \quad \text{for some} \quad I \subset \{i, i+1, \ldots, j\} \tag{4}$$

$$\bigcup_{k\in\{f,f+1,\ldots,q\}} \left\{r_k\left(t_m, c_n\right)\right\} \subset \left\{ \bigcup_{k\in\{i,i+1,\ldots,j\}} \left\{r_k\left(t_m, c_n\right)\right\}\right\} \cap \left\{ \bigcup_{k\in\{z,z+1,\ldots,w\}} \left\{r_k\left(t_m, c_n\right)\right\}\right\} \tag{5}$$

$$L\left(P_{R_i,R_j} \cap P_{R_z,R_w}\right) = \max\left(\bigcup_{f,q\in I} \left\{ L\left(P_{R_f,R_q}\right)\right\}\right) \tag{6}$$

$$\text{Path Sim}\left(P_{R_i,R_j}, P_{R_z,R_w}\right) = \frac{L\left(P_{R_i,R_j} \cap P_{R_z,R_w}\right)}{\max\left(L\left(P_{R_i,R_j}\right), L\left(P_{R_z,R_w}\right)\right)} \tag{7}$$

A path is composed of a sequence of relations; equation (3). The result of intersection of two paths is a set of paths with different length, equation (4). The members of this set are the paths which exist in both input paths. These members have the same type and properties in both input paths, equation (5). For calculation of type similarity, our framework utilises syntactic similarity of relation labels. Moreover, for similarity analysis of relation properties, the algorithm just contemplate transitive one, since other properties do not impose any flaws in semantic similarity of two paths.

Finally, in equation (7), path similarity is calculated through dividing maximum length of the similar paths by maximum length of input paths.

4.3.3 Path weight measurement

Definition 3: Path weight: Path weight w is defined as the maximum weight of relations in a path between two resources say x and y in requirement ontology. This concept is based on the notion that the relations between resources in requirement ontology do not have the same degree of importance. In other word, in accordance with VE architect's requirements, a certain relation may be more significant comparing the other one. The proposed algorithm considers such a notion in terms of relation weights. Therefore, VE architects can emphasise that a given requirement, which is declared as a relation between two resources, is more or less important in comparison with the other parts of his/her priorities. In fact, this novel metric makes our algorithm more close to the VE architect.

In Semantic Web Languages, such as RDF and OWL, a property is a binary relation: it is used to link two individuals or an individual and a value. However, in some cases, the natural and convenient way to represent certain concepts is to use relations to link an individual to more than just one individual or value. These relations are called n-ary relations. For example, one may want to represent properties of a relation, such as our certainty about it, severity or strength of a relation, relevance of a relation, and so on. In response to this issue some patterns have been introduced in W3C Working Group (2006), and our approach uses *Pattern 1: Introducing a new class for a relation,* to add weights to requirement ontology.

Generally, authors categorised the level of importance as follows:

- important
- very important
- critical.

Therefore, VE architect could explicitly stress whether or not a given relation between these resources are important, very important or critical. In case of omitting such a declaration, the algorithm set the weight of the relation as important, by default.

Now, for path weight calculation,

Let,

- P_{R_i,R_j} : Refers to the shortest path between resources R_i and R_j on the condition that i is less than j.
- r_k: Refers to k^{th} relation on a path between two resources. r_k is itself a path with the length of 1.

$$P_{R_i,R_j} = \bigcup_{k=i,i+1,\ldots,j} \{r_k\} \tag{8}$$

$$w(r_k) = \alpha_k, \quad \alpha_k \geq 1 \tag{9}$$

$$w\left(P_{R_i,R_j}\right) = \max\left\{w(r_i), w(r_{i+1}), \ldots, w(r_j)\right\} \tag{10}$$

Regarding to the definition of path weight, the maximum relations weight in a given path is considered as path weight, equation (10). This should be noted that we obtain the value of 1 for important, 3 for very important, and 5 for critical based on experimental results. However, every VE architect could define their weights regarding its requirements and needs.

4.3.4 Definition similarity measurement

Definition 4: Definition similarity: definition similarity is defined as the degree of similarity between restrictions of classes in requirement and partner ontology. Every concept (i.e., class) in ontology can be defined via a set of conditions and restrictions that are listed in Table 1. These are the semantic description of classes.

Table 1 OWL restriction elements

No.	*Restriction type*	*Symbol*
1	Owl: all values from	$\forall$
2	Owl: some values from	$\exists$
3	Owl: has value	$\ni$
4	Owl: min cardinality	$\geq$
5	Owl: max cardinality	$\leq$
6	Owl: cardinality	$=$

Now, for definition similarity calculation,

Let,

- rs_{C_i} : Refers to restriction set *rs* for the class C_i.
- $rs_h(rt_k, rp_l, v_m)$: Refers to a restriction from *rs* set, with the restriction type of rt_k, restricted property rp_l, and the value of v_m. It should be noted that *v* could refer to cardinality, or an individuals (i.e., concept instances) in terms of restriction type. To be more specific, in allValuesFrom or someValuesFrom restriction type *v* refers to an individual while in minCardinality or maxCardinality, *v* refers to cardinality.
- $n(rs)$: Returns total number of *rs* set members.
- m, k, l: Refer to the indexes of the sets.
- I_a: Refers to individuals.
- x: Refers to total number of a concept restrictions.

$$rs_{C_i} = \bigcup_{h=0,\ldots,x} \left\{ rs_h \left(rt_{k_h}, rp_{l_h}, v_{m_h} \right) \right\} \tag{11}$$

Such that $k_h \in \{1, 2, 3, 4, 5, 6\}$, $l_h \in \mathbb{N}$, $(m_h \in \mathbb{N}) \vee (m_h \in I_a)$,

$$\left(rs_{C_i} \cap rs_{C_j} \right) = \bigcup_{h=0,\ldots,g \leq x} \left\{ rs_h \left(rt_{k_h}, rp_{l_h}, v_{m_h} \right) \right\} \tag{12}$$

Such that $k_h \in \{1, 2, 3, 4, 5, 6\}$, $l_h \in \mathbb{N}$, $(m_h \in \mathbb{N}) \vee (m_h \in I_a)$,

$$\text{Definition Sim}\left(rs_{C_i}, rs_{C_j} \right) = \frac{n\left(rs_{C_i} \cap rs_{C_j} \right)}{\max\left(n\left(rs_{C_i} \right), n\left(rs_{C_j} \right) \right)} \tag{13}$$

The intersection of two restriction sets are a restriction set [equation (12)], in which the members of output set have the same type, restricted property, and value in two input sets. For similarity analysis of restriction type, again the algorithm uses syntactic similarity. Finally, for definition similarity measurement, the number of similar restrictions is divided by the maximum number of input restriction sets, equation (13).

It has to be emphasised that in definition similarity measurement, we calculates all the states in which one condition may embody another one, some conditions are more general than the others or in similar way one condition is special form of another one.

4.3.5 Conceptual similarity value measurement

In previous subsections GoR, path similarity, path weight, and definition similarity have been defined and calculated. Based on these metrics, we are able to compute conceptual similarity value (CSV) for each of the inputs of this phase including requirement resource set (i.e., requirement ontology) and partners' resource set. CSV indicates conceptual level matching degree. There are two points in CSV calculation. Firstly, ontology is a directed graph, hence the path set from R_i to R_j may be quite different from the path set from R_j to R_j, equation (14). Thus, GoR, path weight, and path similarity of former path set is dissimilar with the latter one. For CSV calculation we consider both of these cases and summations in equation (15) satisfy this issue.

Secondly, CSV calculation for every two resources within a set must be calculated with respect to the corresponding ones, which have been identified through syntactic similarity, in requirement resource set. For example, if there is no path from R_i to R_j in requirement ontology, our algorithm does not take existence or non-existence of such a relation in partners' ontology into account. This is why such a relation does not have any importance and VE architect does not need such a relation in requirement ontology, hence we do not need to care about it.

However, these two points are not true for definition similarity calculation. The reason of that is definition similarity measurement unlike the GoR, path similarity or path weights are not done within the members of the same resource set. Indeed, definition similarity of every concept in partners' ontology directly compare to the similar one in requirement ontology. This lead to compare two ontologies based on their restrictions that impose for object properties.

Classes in OWL ontology divided into two kinds: primitive class that only has necessary conditions and defined classes that has at least one set of necessary and sufficient conditions. If an individual is a member of primitive class it must satisfy the conditions. However, if some individual satisfy these conditions we cannot say that it is the member of that class (Horridge et al., 2004).

On the contrary, if an individual is a member of defined class it must satisfy the conditions, however if some individual satisfies the conditions then the individual must be a member of the class. Unlike Shu et al. (2007), that considers only defined classes, our algorithm contemplate both primitive and defined classes. This is why, in Shu et al. (2007) authors utilise description reasoning to find similar classes within a same ontology, while our algorithm aim to find similar classes between two ontologies including requirement and partner ones, equation (16).

Regarding the above-mentioned points, CSV is calculated as follows:

Let,

- S: Refers to resource set of requirement ontology.
- T: Refers to resource set of partner ontology.
- n: Refers to total number of T set members.
- $S[i]$, $S[j]$, $T[i]$, $T[j]$: Refers to the members of S and T sets.

- *CSV*(*S*, *T*): Refers to conceptual similarity measurement of *T* with respect to *S*.

$$\text{Path Set}\left(R_1, R_2\right) \neq \text{Path Set}\left(R_2, R_1\right) \tag{14}$$

$$\varphi = \sum_{i=1}^{n}\sum_{j=1}^{n}\left[\left(GoR\left(T[i], T[j]\right) \times w\left(P_{S[i],S[j]}\right) \times \text{Path Sim}\left(P_{S[i],S[j]}, P_{T[i],T[j]}\right)\right)\right] \tag{15}$$

$$\psi = \sum_{k=1}^{n} \text{Defenition Sim}\left(r_{T[k]}, r_{S[k]}\right) \tag{16}$$

$$CSV(S,T) = \begin{cases} \varphi + \psi & if \quad GoR\left(S[i], S[j]\right) \neq 0 \\ \psi & if \quad GoR\left(S\ [i], S[j]\right) = 0 \end{cases} \tag{17}$$

In equation (17), it is assumed that the corresponding members in *S* and *T* have the most syntactic similarity value. For instance, *S* [1] and *T* [1] have the most syntactic similarity value.

4.4 Aggregation and comparison

In phase one and two, lexical and conceptual level matchmaking for each of the requirement and partners' ontologies have been done. Third phase of the algorithm is aggregation and comparison. In this phase, the CSV values of each partners' ontology are compared with CSV value requirement ontology. The qualified partner is the one whose CSV value is closer to CSV value of requirement. This means, difference between requirement CSV value and qualified partner CSV value is minimum comparing to the others. Therefore, qualified partners in which its ontology satisfies the requirement more than the others, is identified.

4.5 Algorithm pseudo-code

In order to elaborate more on how our proposed ontology-based algorithm for partner selection works, this subsection provides the algorithm pseudo-code, Figure 3.

4.6 Applying algorithm to the case

To illustrate the procedure of partner selection, authors apply the algorithm to the given case that is collaborative vehicle sales and distribution. Suppose that there are two WholesalerX and WholesalerY in the network and the VE architect wants to identify and select the best one who meets his requirement.

Therefore, it is aimed to identify the qualified wholesaler with respect to specified requirement. In order to keep calculations simple and understandable, only a fragment of requirement and partners' ontologies has been taken into account. Figure 4(a), Figure 4(b), and Figure 4(c) are requirement ontology, wholesalerX, and wholesalerY ontology, respectively.

Figure 3 Ontology-based partner selection algorithm pseudo-code

Input Ontologies R, O_i;
Output Selected Ontology O_f;
Given S_R,S_i ***Set***;
Given CSV_i, CSV_R, temp, min ***Double Datatype***;
Given O_f= null ***Ontology;***
Begin
    for each O_i (Partners' ontology) ***do***
        /* SyntacticSim is a function for calculating Syntactic similarity between resource labels with the aim of Jaro-Winkler metric */
        S_i = SyntacticSim (R, O_i);
        // S_R,S_i are resource sets of R, $O_i$ontologies, respectively.
    end for
    //Calculate the CSV value for requirement ontology.
    CSV_R = CSV (S_R, S_R);
    min = CSV_R;
    for each S_i (ontology resource set) ***do***
        // Calculate CSV value for Partners' ontologies with respect to the requirement.
        CSV_i= CSV (S_R, S_i);
        temp = CSV_R - CSV_i;
        //Comparison and selection of qualified partner ontology.
        if temp***is less than*** min ***then***
            min = temp;
            O_f= O_i;
        end if
    end for
    returnO_f;
end

Figure 4(a) Sample requirement ontology for given scenario (see online version for colours)

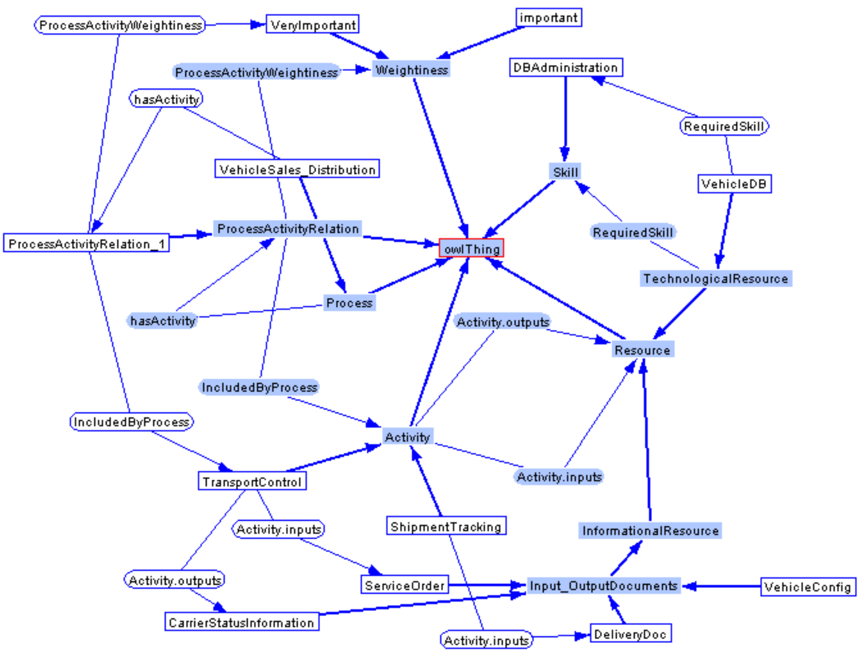


Figure 4(b) Sample ontology for WholesalerX of given scenario (see online version for colours)

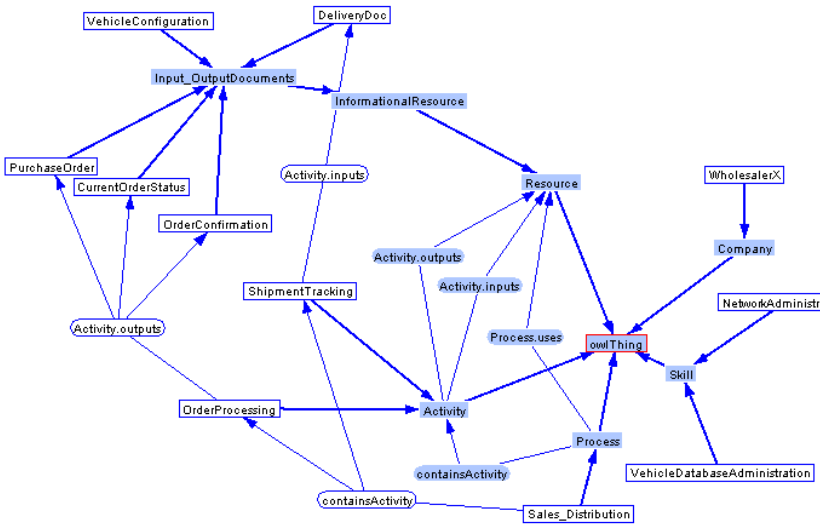
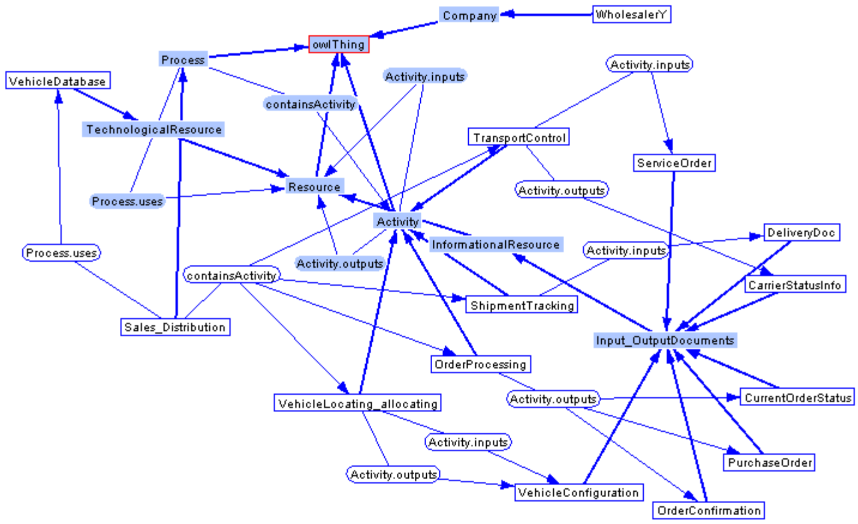


Figure 4(c) Sample ontology for WholesalerY of given scenario (see online version for colours)



4.6.1 Lexical level matchmaking

As discussed earlier, in the first phase we have to apply Jaro-Winkler metric to the ontologies. Table 2 denotes syntactic similarity values for some of resources in the example.

Table 2 Syntactic similarity values for some of resources in the example

Resource labels		*Similarity values*
Requirement ontology	*Partner ontology (wholesalers)*	
Process	*Process*	1.0
VehicleDB	*VehicleDatabase*	0.92
OrderConfirmation	*VehicleConfiguration*	0.65

Any partners within a network could be the one who satisfy the requirement. Thus, our similarity measure must deal with each of the partners' ontologies independently. Therefore, after syntactic matching of resource labels between requirement and partners' enterprise ontologies, there exists an output set for each partner. In our example, we compute syntactic similarity of partners' enterprise ontologies against the S_R set which contains the requirement ontology resources.

> Requirement ontology:
> S_R={"VehicleConfig","Process","important","ProcessActivityRelation","ProcessActivityRelation_1","DBAdministration","VehicleDB","DeliveryDoc", "Activity","Skill","Resource","VeryImportant","InformationalResource", "CarrierStatusInformation","ServiceOrder","Weightiness","TransportControl", "ShipmentTracking","TechnologicalResource","VehicleSales_Distribution", "Input_OutputDocuments"}

The results of syntactic matchmaking are the subsequent sets (i.e., S_X for WholesalerX and S_Y for WholesalerY). That is, these sets have the highest syntactic similarity with the previous set.

> Partner's ontology (WholesalerX Company):
> S_x={"VehicleConfiguration","Process","null","null","null","null", "VehicleDatabaseAdministration","DeliveryDoc","Activity","Skill", "Resource","null","InformationalResource","null","null","null","null", "ShipmentTracking","null","null", "Input_OutputDocuments"}

> Partner's ontology (WholesalerY Company):
> S_y={"VehicleConfiguration","Process","null","null","null","null", "VehicleDatabase","DeliveryDoc","Activity","null","Resource","null", "InformationalResource","CarrierStatusInfo","ServiceOrder","null", "TransportControl","ShipmentTracking","TechnologicalResource","null", "Input_OutputDocuments"}

Some of the members in S_x and S_y are null value. In case that a resource in S_R does not have any equivalent resource in S_x and S_y with syntactic similarity more than threshold, the algorithm set the value of null instead. For instance, TechnologicalResource in S_R set does not have syntactic similarity value of more than threshold with any resources in WholesalerX. With respect to our test results, we ensured that the threshold must be set to the value of 80%.

4.6.2 Conceptual level matchmaking

Regarding to the example, in the following, the result of GoR, path weight, and path similarity measurement for some sample resources is presented in Tables 3, 4, and 5, respectively.

Table 3 GoR values between some of resources in given example

Ontology	*Resource Pairs* (R_i, R_j)	*GoR* (R_i, R_j)
Requirement	*(VehicleDB, Skill)*	0.5
WholesalerX	*(Sales_Distribution, Input_OutputDocuments)*	0.33
WholesalerY	*(PurchaseOrder, Company)*	0.0

Table 4 Path weight of some paths in given example

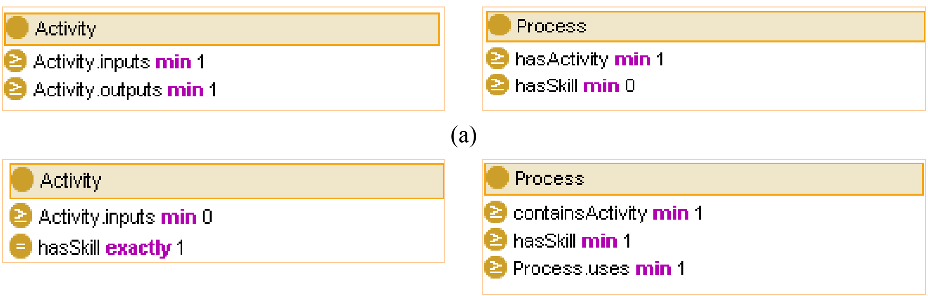
Ontology	*Path between* R_i *and* R_j	$w(P_{Ri,\,Rj})$
Requirement	*(VehicleSales_Distribution, TransportControl)*	3.0
WholesalerX	*(Sales_Distribution, OrderProcessing)*	1.0

Table 5 Path similarity values between some paths

Ontology	$P_{R_i,R_j} \in requirement$ $P_{R_f,R_j} \in WholesalerX \quad or \quad WholesalerY$	*PathSim* $(P_{Ri,Rj}, P_{Rf,Rq})$
Requirement, WholesalerX	$(P_{VehicleSales\_Distribution,\ Input\_OutputDocuments}, P_{Sales\_Distribution,\ DeliveryDoc})$	0.25
Requirement, WholesalerY	$(P_{TransportControl,Input\_OutputDocuments}, P_{TransportControl,\ CarrierStatusInfo})$	0.5

In the continuation of applying the algorithm to given example, definition similarity is computed for some of the classes in requirement and WholesalerX ontologies. Two classes that are 'activity' and 'process' with their corresponding restrictions are depicted in Figure 5(a) and Figure 5(b).

Figure 5 'Activity' and 'process' classes in, (a) requirement ontology (b) WholesalerX ontology (see online version for colours)



Following, the result of definition similarity calculation for 'activity' and 'process' classes in both requirement and WholesalerX ontologies are presented in Table 6.

Table 6 Definition similarity values between defined classes

Classes (C_i, C_j) $Ci \in WholesalerX,\ Cj \in requirement$	*DefinitionSim* (r_{Ci}, r_{Cj})
(*Activity, Activity*)	0.50
(*Process, Process*)	0.33

Finally, Table 7 encompasses the result of CSV calculation for S_R, S_x, S_y sets.

Table 7 CSV values for S_R, S_x, S_y sets

Resource sets	*Conceptual similarity value*
$CSV(S_R, S_R)$	47.58
$CSV(S_R, S_x)$	11.91
$CSV(S_R, S_y)$	19.50

4.6.3 Aggregation and comparison of CSVs

After computing CSV values for each set, we have to compare them with requirement CSV value to identify qualified partner. In this regard, firstly, we have to compute CSV value for requirement resource set because in the best case a certain partner can be exactly similar to the requirement ontology. This means, in our algorithm the first input for CSV value calculation is requirement itself in order to estimate the possible upper bound of CSV. Based on the obtained results, WholesalerY is the qualified partner, since CSV value of WholesalerY (i.e., 19.50) is closer to the CSV value of requirement (i.e., 47.58) in comparison with the WholesalerX.

However, this question may arise 'Why WholesalerY is the best partner regarding the specified requirement?'. If we concentrate on these three ontologies it can be observed that in requirement ontology 'TransportControl' specified as a very important activity which must be satisfied by partners. In this matter, WholesalerY does have this activity, while WholesalerX lacks this activity. Thus, one of the important reasons of selecting WholesalerY is due to more coverage of requirement ontology.

5 Experimental results

5.1 Implementation and evaluation

In order to evaluate the algorithm, authors implemented it using existing technologies. For syntactic similarity, we used Secondstring Java package (Cohen et al., 2009) and for processing ontologies and semantic-based similarity computations we utilised Jena package (HP Labs, 2002). Moreover, to capture and model the knowledge of a network and its partners we utilised CNO ontology. Figure 6 displays the screenshot of the implemented tool.

We then conduct two experiments using two different approaches. In the first one, we run an experiment using 56 dissimilar scenarios as requirement ontologies against instantiated CNO ontology for two different business processes including Collaborative Vehicles Sales and Distribution from automotive industry, and Collaborative Online Brokerage in banking industry. Regarding the test scenarios is based on banking and automotive industry, the CNO ontology has been customised in order to include/exclude some of the concepts on the basis of required scenario scope, Figure 7.

Figure 6 Screenshot of implemented tool for ontology-based partner selection (see online version for colours)

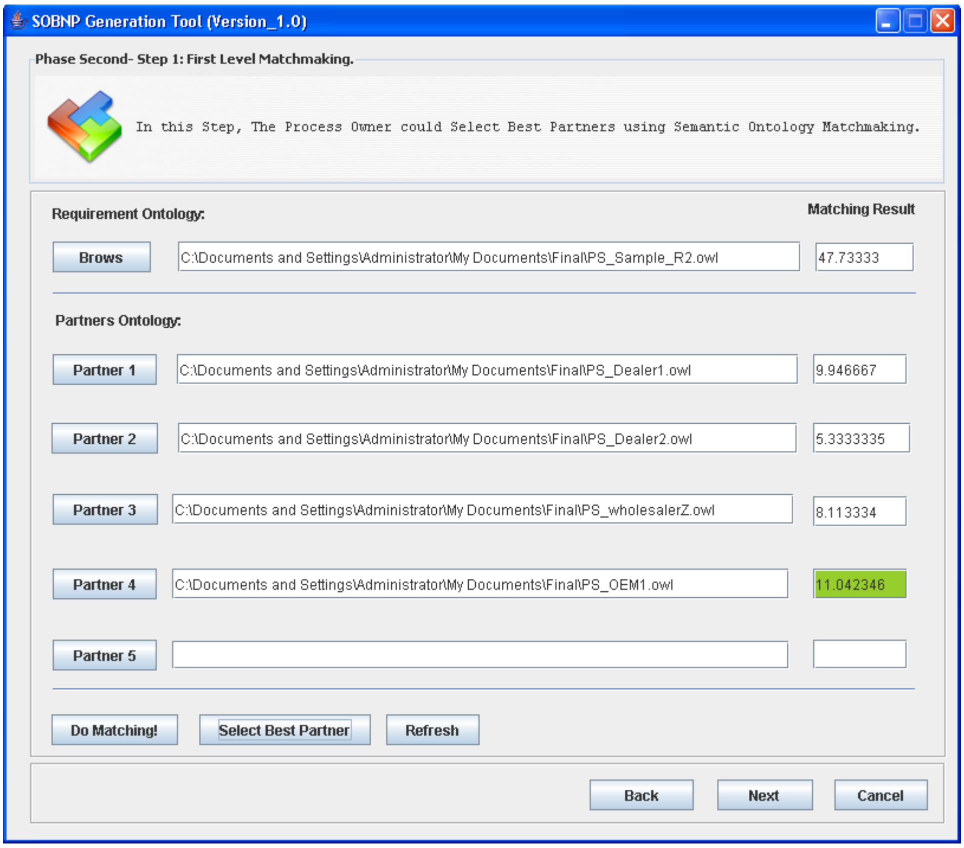
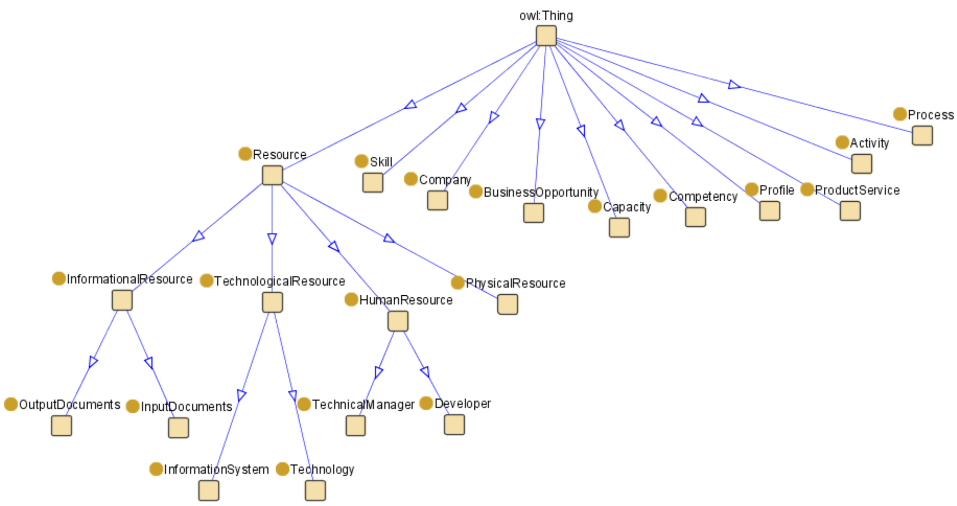


Figure 7 Customised CNO ontology fragment (see online version for colours)



Thereafter, authors compare the results from the algorithm with those from a manual matching by ontology expert. Table 8 denotes the accuracy results of the algorithm.

Table 8 The results of algorithm evaluation using the first approach

Business process	*Number of partners*	*Number of requirements scenarios*	*Number of correct matches*	*Number of wrong matches*	*Accuracy*
Collaborative vehicles sales and distribution	6	22	14	8	63%
Collaborative online brokerage	5	30	23	7	76%

In the second approach, the authors collected 49 real-world ontologies created and maintained by professionals, as our test ontologies. Most of these ontologies are in Business or e-business domain and all of them are specified by OWL. Thereafter, the ontologies classified into eight categories in terms of their theme. Table 9 denotes the classified ontologies within each category.

Table 9 Forty nine test ontologies and their categories

Categories	*Ontologies*
People	• Conference.owl • factbook-ont.owl • foaf.owl • People.owl • Person.owl
Process	• Action.owl • BravoAirProcess.owl • Grounding.owl • Process.owl • Profile.owl • Service.owl
Software	• Bom.owl • jOWL.owl • Soft-Onto.owl • som.owl • vom.owl • System-ont.owl

Table 9 Forty nine test ontologies and their categories

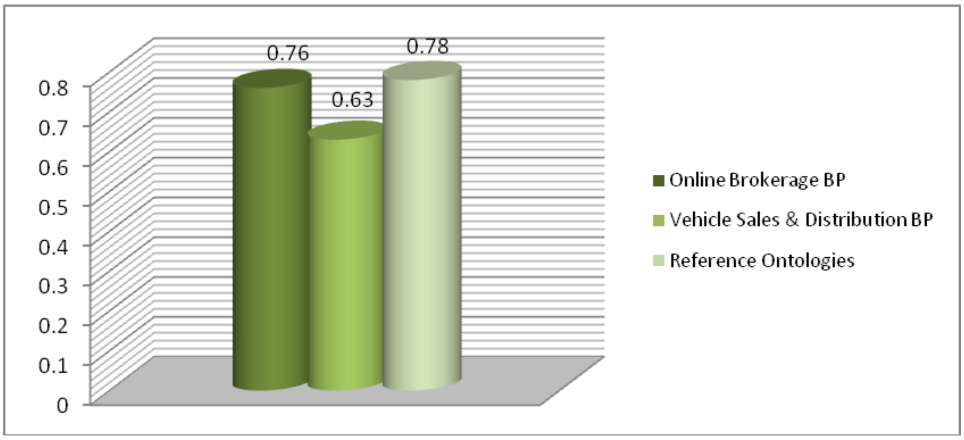
Categories	*Ontologies*
Travel	• Countries.owl • Restaurant.owl • Space.owl • Terrorism.owl • TourismA.owl • TourismB • Travel.owl
Virtual enterprise	• CNO.owl • ContractOntology.owl • DagstuhlVirtualOrganization.owl • DagstuhlVirtualOrganization.owl • SOBNP.owl
Russia	• russia1.owl • russia2.owl • russiaA.owl • russiaB.owl • russiaB.owl
Animal	• animalsA.owl • animalsB.owl • koala.owl
Mutual exclusive	• Biblio.owl • BibTex.owl • camera.owl • camera2.owl • sportEvent.owl • sportEvent.owl • time.owl • time-entry.owl

After that, we randomly pick up one ontology from a category as requirement and match it against the rest ontologies within that category. Table 10 plots the accuracy results of the test.

Table 10 The results of algorithm evaluation using the second approach

Number of categories	*Number of ontologies*	*Number of correct matches*	*Number of wrong matches*	*Accuracy*
8	49	11	3	78%

Figure 8 The proposed algorithm evaluation results (see online version for colours)



At last, Figure 8 indicates the results of experiments using two mentioned approaches.

5.2 Time complexity analysis

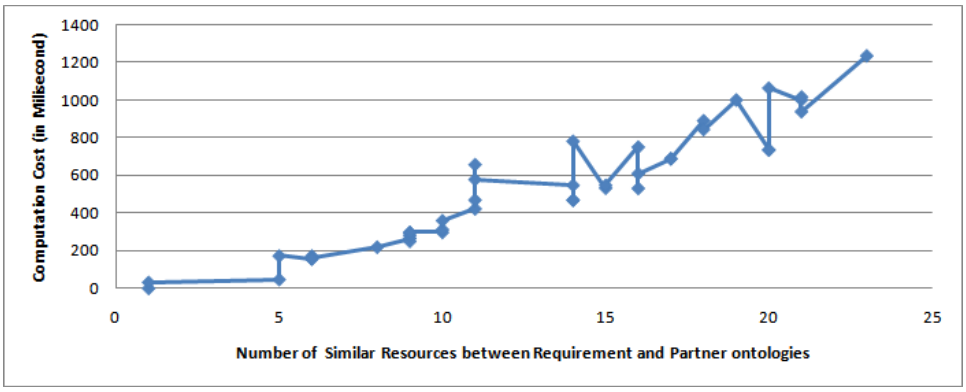
In the process of matchmaking, there is often a conflict between quality and efficiency (Shu et al., 2007). Although our algorithm aims to get a good quality of match, but its efficiency is tolerable. The time complexity of the proposed semantic-based partner selection algorithm is in the order of $O(n^2)$, with n is the number of resources in output set for partners after syntactic matchmaking. To be more specific, CSV is dependent upon calculating GoR with a time complexity of $O(n^2)$, path weight measurement with constant time complexity, path similarity with $O(m^2)$ where m is the maximum length of two paths, and finally definition similarity with time complexity of $O(n)$. As these metrics must be computed for every single resource twice, the total time complexity of CSV become $O(n^4m^2)$. However, since it is possible to pre-compute the four metrics particularly GoR and Path similarity, the total time complexity of the proposed algorithm become $O(n^2)$ which is an acceptable time complexity order. In this regard, the time complexity of ontology-based partner selection algorithm in Huang et al. (2007) is $O(n^2)$ too.

5.2.1 Performance analysis

To perform performance analysis of the proposed algorithm, we run it using different ontologies. As expected, the computation cost increases when the number of similar resources between requirement and partner ontologies increases, Figure 9.

There are some contradictions, for example when the number of resources is the value of 11, the execution time varies. In a similar way, execution time with the number of 20 resources decreases. This is why the number of resources in requirement and partners' ontologies also influence the execution time of matchmaking, since the algorithm deals with SPL calculation, path similarity, etc. In performance tests, we utilise requirement ontologies and partners' ontologies in which the number of their resources varies from 8 to 51 and 10 to 165, respectively. Regarding the size of tested ontologies it is deduced that the matchmaking process is done within a reasonable response time.

Figure 9 Performance analysis of the algorithm (see online version for colours)



6 Discussion

6.1 Dealing with dynamic environment

As discussed in introduction section, changing business environment and turbulent market conditions requires an efficient and adjustable partner selection algorithm. In this regard, this article proposed an ontology-based partner selection algorithm that could be tuned according to the VE architect requirements. A key assumption in the proposed algorithm is that every partner within a network has to capture, define, and organise relevant knowledge about its activities, processes, organisations, skills, competencies, etc., using OWL-DL ontology language. However, the partners' resources such as business process, skills, core competencies, etc., are not permanent and evolve during the time. Therefore, their ontologies will be invalidated after a period of time. Consequently, their chance for participating in new business opportunity will be lost. To tackle this problem, partners have to evolve their ontologies using real-time enterprise ontology evolution techniques such as those that introduced in Wong et al. (2008). As our algorithm is close to the VE architect and his requirement, it could effectively be adjusted with dynamic conditions too.

6.2 Dealing with heterogeneity

By introducing a business process from the automotive industry, we implicitly claim that our algorithm could be used for identifying best sub-processes and activities from partners in order to realise the expected business process. However, one could question that how the algorithm could handle that, while the sub-processes or activities held by different business systems are heterogeneous because each has a different conditional sequence of activities or actions, respectively. There are two distinct solutions. In the first solution, partners have to transform their business processes into OWL in terms of high-level Petri Nets concepts such as Transition, Place, From Place, To Place, and so on (Ehrig et al., 2007). As our algorithm is a general application-oriented semantic matchmaking algorithm, it could measure similarity between the transformed business process models and select the appropriate one.

The second solution is another form of the first one in which partners have to specify the concerned workflow patterns between activities or actions in their ontologies. With the aim of these patterns, we could determine if the specified sub-process in requirement ontology is equivalent with those that the partners' ontologies demonstrate.

6.3 Generality of the algorithm

Based on the results of experiments we obtained indications of positive evaluation of the algorithm in terms of its quality. However, there are two important points that must be elucidated. Firstly, the evaluation of the algorithm using the first approach confirms that the algorithm is an appropriate method for partner selection within a network of organisations. Secondly, the evaluation of the algorithm using the second approach assured us that the algorithm can be used in generalised form of ontology matchmaking. As ontology-based semantic matchmaking methods play a vital role in most of research paradigms including semantic-based service discovery, semantic-based resource allocation for workflow management systems, etc., hence a generalised semantic matchmaking algorithm could advances many research areas.

7 Conclusions

This paper proposed an ontology-based partner selection algorithm that identifies best partner through calculating semantic similarity between VE architect's requirement ontology and partners' ontologies. The algorithm introduced and utilised diverse techniques including GoR, path similarity, path weight, and definition similarity. These metrics then were adjusted and combined on the basis of VE architect's requirements. The experimental results concluded that the algorithm is an effective and efficient solution for semantic-based partner selection in instant VEs.

References

Billig, A., Blomqvist, E. and Lin, F. (2007) 'Semantic matching based on enterprise ontologies', *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, pp.1161–1168, Springer, Berlin Heidelberg.

Camarinha-Matos, L.M. and Afsarmanesh, H. (2003) 'Elements of a base VE infrastructure', *Computers in Industry*, Vol. 51, No. 2, pp.139–163.

Camarinha-Matos, L.M., Afsarmanesh, H. and Rabelo, R.J. (2000) 'Supporting agility in virtual enterprises', *E-Business and Virtual Enterprises*, October, pp.89–104.

Camarinha-Matos, L.M., Afsarmanesh, H. and Rabelo, R.J. (2003) 'Infrastructure developments for agile virtual enterprises', *International Journal of Computer Integrated Manufacturing*, July, Vol. 16, Nos. 4/5, pp.235–254, doi: 10.1080/0951192031000089156.

Chen, T.Y. (2008) 'Knowledge sharing in virtual enterprises via an ontology-based access control approach', *Computers in Industry*, Vol. 59, No. 5, pp.502–519.

Cohen, W. et al. (2003) 'A comparison of string distance metrics for name-matching tasks', *Proc. of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03)*, 9–10 August, Acapulco, Mexico.

Cohen, W. et al. (2009) *Second String Project Page* [online] http://secondstring.sourceforge.net/ (accessed October 2011).

Do, H.H. and Rahm, E. (2002) 'COMA: a system for flexible combination of schema matching approaches', *Proceedings of the 28th international conference on Very Large Data Bases*, VLDB Endowment, August, pp.610–621.

Doan, A., Madhavan, J., Dhamankar, R., Domingos, P. and Halevy, A. (2003) 'Learning to match ontologies on the semantic web', *The VLDB Journal*, Vol. 12, No. 4, pp.303–319, Springer-Verlag, New York, NY.

Ehrig, M., Koschmider, A. and Oberweis, A. (2007) 'Measuring similarity between semantic business process models', *Proceedings of the fourth Asia-Pacific conference on Conceptual modelling*, Australian Computer Society, Inc., January, Vol. 67, pp.71–80.

Fox, M.S. (1992) 'The TOVE project: a common-sense model of the enterprise', *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Lecture Notes in Artificial Intelligence*, No. 604, pp.25–34, Springer-Verlag, Berlin, Germany.

Giunchiglia, F., Shvaiko, P. and Yatskevich, M. (2005) 'Semantic schema matching', *Proceedings of the Thirteenth International Conference on Cooperative Information Systems (CoopIS 05)*, Agia Napa, Cyprus, November.

Grefen, P., Mehandjiev, N., Kouvas, G., Weichhart, G. and Eshuis, R. (2009) 'Dynamic business network process management in instant virtual enterprises', *Computers in Industry*, Vol. 60, No. 2, pp.86–103.

Hassan, M.M. and Huh, E.N. (2013) 'Multi-objective optimization model and algorithms for partner selection', *Dynamic Cloud Collaboration Platform*, pp.37–45, Springer, New York.

Horridge, M., Knublauch, H., Rector, A., Stevens, R. and Wroe, C. (2004) *A Practical Guide to Building OWL Ontologies Using The Protégé´-OWL Plugin and CO-ODE Tools Edition 1.0* [online] http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf.

HP Labs (2002) *Jena: A Semantic Web Framework for Java* [online] http://jena.sourceforge.net/ (accessed May 2011).

Hsieh, F-S. and Lin, J-B. (2012) 'Virtual enterprises partner selection based on reverse auctions', *The International Journal of Advanced Manufacturing Technology*, Vol. 62, Nos. 5–8, pp.847–859.

Huang, J., Dang, J. and Huhns, M.N. (2007) 'Ontology-based partner selection in business interaction', in Rittgen, P. (Ed.): *Handbook of Ontologies for Business Interaction*, pp.364–380, IGI Global.

Jaro, M.A. (1995) 'Probabilistic linkage of large public health data files (disc: P687–689)', *Statistics in Medicine*, Vol. 14, Nos. 5–7, pp.491–498.

Khoshkbarforoushha, A. and Aghdasi, M. (2009) 'Rule-based service oriented business network process realization in dynamic virtual enterprises', *E-Business Engineering, 2009: ICEBE'09: IEEE International Conference on*, IEEE, October, pp.111–118.

Lei, L. and Ian, H. (2003) 'A software framework for matchmaking based on semantic web technology', *Proceedings of the Twelfth International World Wide Web Conference (WWW 2003)*.

Madhavan, J., Bernstein, P. and Rahm, E. (2001) 'Generic schema matching with cupid', *Proceedings of the Twenty-Seventh VLDB Conference*, Roma, Italy.

Niu, S.H., Ong, S.K. and Nee, A.Y.C. (2012) 'An enhanced ant colony optimiser for multi-attribute partner selection in virtual enterprises', *International Journal of Production Research*, Vol. 50, No. 8, pp.2286–2303.

Noy, N. and Musen, M. (2000) 'Prompt: algorithm and tool for automated ontology merging and alignment', *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI 00)*, AAAI Press, Menlo Park, CA.

Paolucci, M., Kawamura, T., Payne, T.R. and Sycara, K. (2002) 'Semantic matching of web services capabilities', *Proceedings of the First International Semantic Web Conference (ISWC)*, Sardinia, Italy, 9–12 June.

Petersen, S.A. and Divitini, M. (2002) 'Using agents to support the selection of virtual enterprise teams', *Proceedings of 4th International Bi-Conference Workshop on Agent-Oriented Information Systems*.

Plisson, J. et al. (2007) 'An ontology for virtual organization breeding environments', *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 37, No. 6, pp.1327–1341.

Rhee, S.K., Lee, J. and Park, M. (2007) 'Ontology-based semantic relevance measure', *Proceedings of the First International Workshop on Semantic Web and Web 2.0 in Architectural, Product and Engineering Design*.

Rocha, A.P. and Oliveira, E. (1999) 'An electronic market architecture for the formation of virtual enterprises', *Proceedings of the IFIP Tc5 Wg5.3 / PRODNET Working Conference on infrastructures For Virtual Enterprises: Networking industrial Enterprises*, 27–28 October.

SAP Solution Composer (2006) [online] http://www.sap.com/solutions/businessmaps/composer/index.epx (accessed November 2010).

Shu, G. et al. (2007) 'Ontology-based semantic matchmaking approach', *Advances in Engineering Software*, Vol. 38, No. 1, pp.59–67.

The CNO Ontology (2009) [online] http://kt.ijs.si/software/CNOntology/onto.zip (accessed June 2010).

W3C Working Group (2006) *Defining N-ary Relations on the Semantic Web* [online] http://www.w3.org/TR/swbp-n-aryRelations/ (accessed November 2011).

W3C: OWL Web Ontology Language Overview (2009) [online] http://www.w3.org/TR/owl-features/ (accessed June 2010).

Wang, C. et al. (2008) 'A comprehensive semantic-based resource allocation framework for workflow management systems', *IEEE Network Operations and Management Symposium*, 7–11 April, doi: 10.1109/NOMS.2008.4575225.

Wei, Z. and Bu, Y-P. (2012) 'Cultural algorithm based on particle swarm optimization for partner selection of virtual enterprise', *Control Conference (CCC), 2012 31st Chinese*, IEEE, July, pp.2238–2241.

Winkler, W.E. (1999) *The State of Record Linkage and Current Research Problems*, Statistical Research Division, US Census Bureau.

Wong, J.H. et al. (2008) 'Real-time enterprise ontology evolution to aid effective clinical telemedicine with text mining and automatic semantic aliasing support', *Lecture Notes in Computer Science*, Vol. 5332, pp.1200–1214, Springer-Verlag, Berlin, Heidelberg.

Wu, N. and Su, P. (2005) 'Selection of partners in virtual enterprise paradigm', *Robotics and Computer-Integrated Manufacturing*, April, Vol. 21, No. 2, pp.119–131, ISSN 0736-5845.

Wu, Q., Zhu, Q. and Zhou, M. (2013) 'A correlation-driven optimal service selection approach for virtual enterprise establishment', *Journal of Intelligent Manufacturing*, pp.1–13.

Zhang, Y., Tao, F., Laili, Y., Hou, B., Lv, L. and Zhang, L. (2012) 'Green partner selection in virtual enterprise based on Pareto genetic algorithms', *The International Journal of Advanced Manufacturing Technology*, Vol. 67, Nos. 9–12, pp.1–17.