

Service-oriented architecture for ontologies supporting multi-agent system negotiations in virtual enterprise

Xiaohuan Wang · T. N. Wong · Gong Wang

Received: 27 February 2010 / Accepted: 11 October 2010
© The Author(s) 2010. This article is published with open access at Springerlink.com

Abstract This paper offers a service-oriented architecture (SOA) for ontology-based multi-agent system (MAS) negotiations in the context of virtual enterprises (VEs). The objective of this paper is fourfold. First, it is to design a SOA which utilizes ontology and MAS to provide a distributed and interoperable environment for automated negotiations in VE. In this architecture, individual ontologies for both the VE initiator and its potential partners are constructed to describe and store resources and service knowledge. Second, a series of semantic ontology matching methods are developed to reach agents' interoperability during the negotiation process. Third, correspondence-based extended contract net protocol is presented, which provides basic guidelines for agents' reaching mutual understandings and service negotiation. Last, a fuzzy set theory based knowledge reuse approach is proposed to evaluate the current negotiation behaviors of the VE partners. A walkthrough example is presented to illustrate the methodologies and system architecture proposed in this paper.

Keywords Multi-agent systems · Virtual enterprises · Negotiation · Ontology · Service-oriented architecture

Introduction

In today's global economy, enterprises are inclined to focus on their core competences, and rely on other enterprises to provide complementary skills and services through cooperating and sharing knowledge with each other. Recently, it has

been a trend for companies to collaborate through the establishment of virtual enterprise. Virtual Enterprise (VE) is a temporary network of enterprises, whereas the VE initiator interacts with its partners to share skills, resources, costs and benefits to achieve market opportunities and gain more value for products and services (Pallot and Sandoval 1998).

Regarding the flexible and dynamic nature of the VE, appropriate information technology and enabling services are required to support the establishment and management of the VE and the integration and interoperation of business processes. An effective information infrastructure is required to coordinate and enable the services in the VE life cycle. A service refers to a well-defined, self-contained and distributed component, which supports interoperability and can be dynamically requested and responded among service providers and recipients. From the perspective of business, Service Oriented Architecture (SOA) is defined as a set of business, process, organizational, governance, and technical methods to enable an agile, business-driven IT environment for greater competitive advantage (Bieberstein et al. 2006).

Multi-agent systems (MASs) (Jennings 2000) as a technical method in SOA are largely applied to provide a system platform to enable distributed and dynamic negotiations among VE participants. With the MAS implementation, each individual enterprise is represented by an individual agent, enabled with certain capabilities to fulfill the negotiation objectives. Agents concede to each other by interchanging their service knowledge towards certain issues. To achieve a good negotiation performance, agents should mutually understand the terminologies they adopted, and then make decisions towards the counterpart's service knowledge.

From the perspective of Artificial Intelligence (AI), Gruber (1993) defines ontology as a specification of a conceptualization. Owing to the powerful and expressive characteristics, ontology models has been largely introduced

X. Wang (✉) · T. N. Wong · G. Wang
Department of Industrial and Manufacturing Systems Engineering,
The University of Hong Kong, Pokfulam Road, Hong Kong, China
e-mail: wangxh@hku.hk

in MAS to represent knowledge, and a lot of ontology-based applications are also developed to facilitate MAS negotiations. It is helpful in reaching semantic interoperability and reusing the existing knowledge.

This paper introduces a SOA for ontology-based MAS negotiations in VE from four perspectives: First, domain ontology is designed and constructed to describe and store service knowledge. Second, ontology matching methods are developed to reach agents' semantic interoperability for mutual understanding. Third, a correspondence-based extended contract net protocol (CNP) is presented, which is developed on the basis of the traditional CNP by considering the semantic interoperability among heterogeneous agents. Lastly, a fuzzy set theory based knowledge reuse method is introduced to support the evaluation of current negotiation behaviors of VE partners.

The remainder of this paper is structured as follows: Section "Literature review" reviews some related literatures. Section "Methodology" presents the SOA and the proposed methodologies. Section "System implementation and evaluation" presents the system implementation and the evaluation results. Section "Conclusions and future works" summarizes the conclusions and presents some future works.

Literature review

Agent enabled SOA in VE

A VE is a promising approach to form alliances of different enterprises in order to provide high-quality services in a dynamic and turbulent environment (Camarinha-Mahos and Afsarmanesh 1999).

MAS have many characteristics that enable themselves suitable in SOA in the VE negotiation context, such as *autonomous, responsive, flexible* and *pro-active, social* and *intelligent* (Dignum 2006; Lee and Kim 2008). Many researchers have engaged in providing solutions in MAS negotiations in VE applications. Xue et al. (2005) presented a multi-attribute negotiation and utility theory based MAS for supply chain coordination. In their application, agents were enabled to determine the planning parameters, tasks and initial resource allocations; and they could also coordinate protocols and strategies among themselves. Wang et al. (2007) proposed a MAS-based negotiation platform for small to medium enterprises to integrate resources to form a VE. In their application, a backward network algorithm was adopted for optimizing resource allocation. Moreover, facilitator agents were enabled to coordinate activities of resources in a parallel manner. Kaihara and Fujii (2008) proposed a MAS architecture focusing on negotiation process in VE formation. Firstly, VE business models were classified into three categories, i.e. vertically integrated business model, horizon-

tally specialized business model and hybrid business model. A contract net protocol (CNP) based negotiation protocol was then introduced as fundamental rules for agents' interactions. Finally, marketing science and cooperative game were applied to enrich the negotiation mechanism. Mahdavi et al. (2009) developed an e-based supply chain MAS for the design of mass-customized online negotiations. They utilized colored Petri nets to simulate the proposed system. Inventory policy and bullwhip effect were considered during agents' negotiation.

Ontology based MAS negotiations

Semantic interoperability

In real life, the VE initiator and its potential partners are equipped with heterogeneous techniques and knowledge structures. It is difficult for them to reach interoperability or mutual understanding in the agent enabled SOA automatically. Whitman and Panetto (2006) defined interoperability into four layers: technical, syntactic, semantic and pragmatic. In MAS, technical interoperability refers to message transfer among applications; syntactic interoperability means that agents should reach agreement on the format of data exchange; semantic interoperability refers to the understanding of message contents; while pragmatic interoperability means obtaining willingness, capabilities or policies to participate in the collaboration. With the rapid development of information technology, technical interoperability during message transferring can be easily achieved with advanced agent development platforms, for instance, Java Agent Development Platform (JADE), Trillian Agent Development Toolkit, Smart Python Multi-Agent Development Environment (SPADE), etc. The eXensible Markup Language (XML), Simple Object Access Protocol (SOAP), and Web Service Description Language (WSDL) etc can enable agents to obey the same format of data exchange to reach the syntactic interoperability. The pragmatic interoperability largely relies on participations of humans and organizations. Regarding semantic interoperability, it is still a research challenge and researchers have proposed knowledge representation languages and ontology sets to achieve semantic interoperability in restricted applications. It is actually the focus of this paper.

When service knowledge from heterogeneous enterprises needs to be transferred, it should firstly be expressed in independent structures, and then be compared or matched to reach mutual understanding. Ontology is introduced to provide an elaborate description and formal conceptualization of service knowledge. To date, many researchers and practitioners have introduced ontology to reach MAS semantic interoperability. Chira et al. (2006) proposed an

agent-based architecture to facilitate distributed design teams to cooperate with each other. Ontologies were utilized to represent knowledge to support dialogues among the design actors. Several ontologies were developed, namely detailed design ontology, quality standards ontology, material ontology, structure ontology and design artifact ontology. Functions to realize heterogeneous dialogues were implemented in the MAS. Malucelli et al. (2006) proposed methodologies to compare lexical and semantic similarities among heterogeneous ontologies defined in different agents. In their approach, ontology matching were done based on concept names, characteristics, relations etc. Lexical similarity identification, WordNet-based semantic similarity and N-grams for description were introduced in their study. Jiang et al. (2007) built an agent based virtual organization using ontology to represent their knowledge. A virtual sensor was developed to discover the semantic similarity between web documents from distributed collaborators using ontology oriented match making. The feature distance was evaluated by virtual distance. Many researchers like Tamma et al. (2005), Giovannucci et al. (2008), Garcia-Sanchez et al. (2009) designed negotiation ontology to describe terminologies appeared during the MAS negotiation process to reach semantic interoperability.

Ontology matching is essentially important for reaching semantic interoperability. Choi et al. (2006) presented a review on ontology matching. They classified ontology matching into three categories: ontology matching between an integrated global ontology and local ontologies; ontology matching between local ontologies; and ontology matching in ontology merging and alignment. Topics, merits, drawbacks and implementation tools towards each category were discussed. Orgun et al. (2005) introduced a dialogue based automation of semantic interoperability approach in MAS. In their approach, WordNet-based word similarity calculation method was proposed and latent semantic analysis method was adopted (Cognitive Science Laboratory, Princeton University 2006). Laera et al. (2007) proposed an argumentation based ontology matching approach for MAS. In their study, a correspondence repository was constructed, where matching candidates, ontology mismatches, matching preference and candidate thresholds were stored in. It also presented an argumentation protocol, enabling different agents to understand each other and transfer messages. Sahami and Heilman (2006), and Bollegala et al. (2007) engaged in conducting query-based semantic matching studies, where various types of query ontologies were constructed to enable agents' learning ontology terminologies and structures.

Knowledge reuse

In MAS negotiations, ontology is introduced to describe service knowledge while ontology matching is introduced to

generate term correspondence among heterogeneous service knowledge, aiming at reaching semantic interoperability in the VE. However, service knowledge sharing is not the ultimate objective, how to make good use of the existing service knowledge to improve the negotiation performance is of great importance.

Researchers have presented many methodologies to reuse the existing knowledge. Li et al. (2004), Xue et al. (2005) and Giovannucci et al. (2008) introduced utility theory to evaluate the on-going knowledge, such as negotiation bids, in order to benefit the most during the negotiation process. To ease the knowledge synthesis in a manufacturing environment, Qiu (2006) proposed an ontology-driven knowledge synthesis approach for heterogeneous information system. In this regard, heterogeneous disparate knowledge sources could be integrated and utilized for decision making. Formal models of domain ontology were developed in their study. Lo et al. (2008) established their ontology not only to describe knowledge for knowledge sharing, but also to provide a flexible semantic information access to databases and data warehouses. This ontology defined a set of vocabulary, semantic term relations, inference rules and logics. Tamma et al. (2005) and Garcia-Sanchez et al. (2009) constructed negotiation ontology for describing and supporting knowledge retrieval from MAS participants. Roles and rules of negotiation protocol were defined inside the ontology, which enabled negotiation participants to select proper negotiation protocols flexibly and dynamically. Rau et al. (2009) introduces fuzzy technology to evaluate the opponent's behaviors to obtain more benefit offers. Own preference degree and opponent's concession degree were discussed in their study.

To summarize, there are mainly two types of knowledge usages in MAS negotiations. One is the usage of the current service knowledge, such as utility theory based current bid evaluation; the other is the usage of the existing service knowledge, such as VE partner's past negotiation experiences. The existing service knowledge reveals negotiation strategies to some extent, such as concession degrees for certain issues to certain counterparts. How to utilize the existing service knowledge to improve the current negotiations are extremely important. Since fuzzy set theory excels in representing inaccurate and incomplete information in an accurate way (Zadeh 1965), this paper introduces a fuzzy-based approach to express counterparts' current behaviors towards different negotiation issues and classify the negotiation records into corresponding strategy patterns based on the membership functions of concession degree. Therefore, the VE initiator can utilize the existing service knowledge to evaluate the current trade-offs. During this process, ontology can not only store the service knowledge, but also it can provide ways to retrieve the right services.

Methodology

To facilitate establishment and management of the VE and the integration and interoperation of business processes, a SOA for ontology-based MAS is developed. The MAS contains agents that communicate with each other within a distributed and interoperable environment. The system is implemented in the agent platform JADE (Java Agent DEvelopment Framework). Protégé 3.3.1 is adopted to model the ontology; Ontoviz plug-in is adopted to visualize the ontology structure; Bean Generator plug-in is adopted to automatically generate JADE codes for implementing MAS; SWRL and Jess are introduced to reason the ontology consistency. JADE is fully implemented in Java language and it is conformed to the FIPA (Foundation of Intelligent Physical Agents) specifications. It is chosen as the development platform for three reasons. Firstly, JADE is an FIPA compatible platform, which utilizes the standard FIPA ACL (Agent Communication Language). It can simplify the implementation of MAS for distributed VE participants under a library of FIPA interaction protocols. Secondly, JADE is suitable to be implemented in the web-based service applications, such as developing individual agents to perform roles and functions of service providers and service recipients. In other words, JADE is appropriate in facilitating developing and implementing a SOA. Thirdly, ontology is introduced to represent domain knowledge using Protégé, which can automatically translate ontology structures into JADE code by the Bean Generator Plug-in. Consequently, JADE can not only facilitate the implementation of SOA-based MAS negotiations, but also can utilize the complex and complete semantic knowledge structure during the system implementation and operation processes.

SOA for ontology-based MAS

The SOA for ontology-based MAS is shown in Fig. 1. Under this architecture, each enterprise is represented by an agent. It is a generic architecture under which the VE initiator can

be either a *Buyer Agent (BA)* or a *Seller Agent (SA)*. In cases where the supplies of products or services are required, the VE initiator plays the role of *BA*. Conversely, the VE initiator plays the role of *SA*, if it is to offer distribution services to distributors or retailers. Actually, *BA* and *SA* in this architecture are negotiation counterparts enabled with the same functionalities. In a scenario that the VE initiator requests some services from other enterprises, the VE initiator is represented by a *BA* while other participants are represented by *SAs*. *BA* and *SA* negotiate with each other on certain issues, such as price, quantity, service, etc, to reach final agreement. The messages interchanged during the negotiation process can be regarded as individual service knowledge. Before real negotiation starts, two prerequisites should be satisfied: One is the proper description of individual service knowledge, which enables the accuracy and completeness of the messages being interchanged; the other is the semantic interoperability between individuals, which is an important factor for mutual understanding that will influence the negotiation performance. The above prerequisites are realized by a functional agent, namely, *Knowledge Manager Agent (KMA)*.

Besides *KMA*, other functional agents are developed: *Service Decomposer Agent (SDA)*, and *Coordinator Agent (CA)*. Besides, two types of knowledge bases are also developed i.e. *VE initiator's knowledge base* and the *potential partners' knowledge base*; as well as two types of libraries i.e. *correspondence library* and *correspondence candidate library*.

The components of the proposed architecture are explained as follows:

- *Buyer Agent (BA)*: In this study, *BA* represents the VE initiator, which can interact with *KMA* about the service knowledge and negotiate with *SAs* according to the service requirements.
- *Seller Agent (SA)*: It represents a potential VE partner, which is enabled with the same capabilities of the *BA*.
- *Knowledge Manager Agent (KMA)*: It receives service queries from *BA* and *SA*; matches their terminologies based on their individual ontologies; generates *correspondence library* and *correspondence candidate library*;

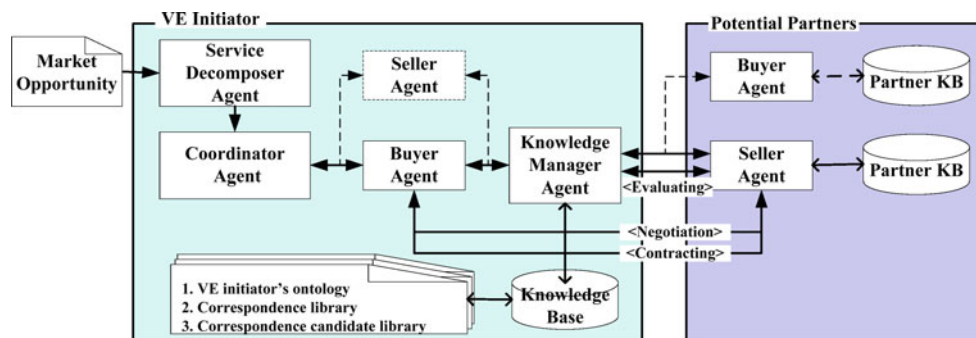


Fig. 1 Service-oriented architecture for ontology-based MAS

translates terminologies among *BA* and *SA* during the negotiations; retrieves the existing service knowledge from the *VE initiator's knowledge base*, and updates the *VE initiator's knowledge base* after each round of semantic matching and each round of negotiation.

- *Service Decomposer Agent (SDA)*: It handles the market opportunities, decomposes them into small sub-service requirements, and sends them to *CA*.
- *Coordinator Agent (CA)*: It receives sub-service requirements from *SDA*; initializes *BA* and *SA* to carry out negotiations; receives negotiation outcomes and makes final decision of who are the final winners and who will become the *VE* partners.
- *VE Initiator's Knowledge Base*: It stores the *VE initiator's ontology model*, *correspondence libraries*, and *correspondence candidate libraries*.
- *VE Partner's Knowledge Base*: It stores individual ontologies, *correspondence libraries* with different partners.
- *Correspondence library*: It records existing correspondences used in past negotiations between *BA* and other agents, which are supposed to be correct correspondences.
- *Correspondence candidate library*: It records the correspondence candidates obtained from semantic matching process, based on the *VE initiator's ontology* and ontologies of its potential partners, which have never cooperated with the *VE initiator* before.

Ontology modeling

Ontology modeling is the foundation of ontology-based applications. In this section, an introduction to ontology construction for the *VE initiator* is illustrated. To provide an elaborate and functional ontology structure, the ontology should be *explicit*, *functional*, and *flexible*.

Explicit is a basic characteristic of ontology. Before defining a real ontology, the objective of ontology construction should firstly be set clear, and then various knowledge sources related to the application domain should be gathered from both inside and outside of an enterprise. In this study, the objective of ontology construction is to describe service knowledge related to *MAS* negotiations among the *VE initiator* and its potential partners.

Ontology is *functional*. In this study, besides providing the explicit description of individual service knowledge, ontology has other functions, i.e. service knowledge storing and service knowledge retrieval for negotiation strategies, which are to be used in the knowledge reuse process. Therefore, concrete classes and properties should be constructed in the designed ontology structure in advance.

Ontology enables a *flexible* knowledge usage in nature. Since data stored as instances is independent from knowledge defined as classes or properties, it is easy to add, delete

or modify the ontology instances and attribute values. If the *VE initiator* is going to purchase a new product, what it needs to do is just to create a new instance of concept "*product*", and then input the attribute values of the new instance, without modifying the basic ontology structures.

The ontology model for the *VE initiator* is depicted in Fig. 2. In this graphic representation, the rectangles represent classes. The arcs linking two classes correspond to the properties which represent relationship between them. While the "*isa*" ("is a") labeled arrows denote the hierarchy relationships between two classes. For instance, the relationship between "*Agent Action*" and "*Initialize Negotiation*" represents that the latter is a subclass of the former.

To describe the scenario that the *VE initiator* selects proper negotiation strategies to initialize negotiations according to the issue preconditions, the following classes are defined in its ontology.

- *Agent action*: It is a super class of actions played by agents, including *Initialize Negotiation*, and *Initialize Strategy*. This is indicated by the "*isa*" hierarchical relation (i.e. subclass property) that connects the "*Agent Action*" class to the subclasses in Fig. 2. *Initialize Negotiation* provides basic terminologies appeared during the negotiation process, i.e. *has item*, *negotiation deadline*, *num of participant*, *num of issue*, *has initiator* and *has participant*, which are defined in the form of ontology properties and clarifies the negotiation variables. *Initialize Strategy* provides basic components of negotiation strategy, i.e. *has issue*, *has participant*, and *has initiator*, which are defined in form of ontology properties.
- *AID (Agent Identity)*: It describes the agent's identity in *MAS* by *name* and *address*, which are defined in the form of ontology attribute. It includes four types of agents, i.e. *BA*, *SA*, *KMA*, and *Partner Agent (PA)*.
- *Issue*: It is a basic concept when configuring the values of negotiation issues. Five types of issues are considered in this study, i.e. *Quantity*, *Due Date*, *Service*, *Price* and *Payment*.
- *Strategy Pattern*: It is designed for the purpose of storing the existing service knowledge. In this study, three issues are considered in the negotiation strategy, i.e. *Quantity*, *Service* and *Price*. *Quantity* indicates the amount of products the *VE initiator* would like to purchase; *Service* indicates the warranty or the post sale service level that the *VE initiator* requires towards the purchased products, the service level is assigned a factor value of 1 to 5, where 1 is the lowest level and 5 is the highest level; *Price* indicates that how much money the *VE initiator* would like to spend on the products. This class also provides strategy terminologies, i.e. *has first price*, *has final price*, *has first quantity*, *has final quantity*, *has first service*, *has final service*, *has price concession degree*, *has quantity*

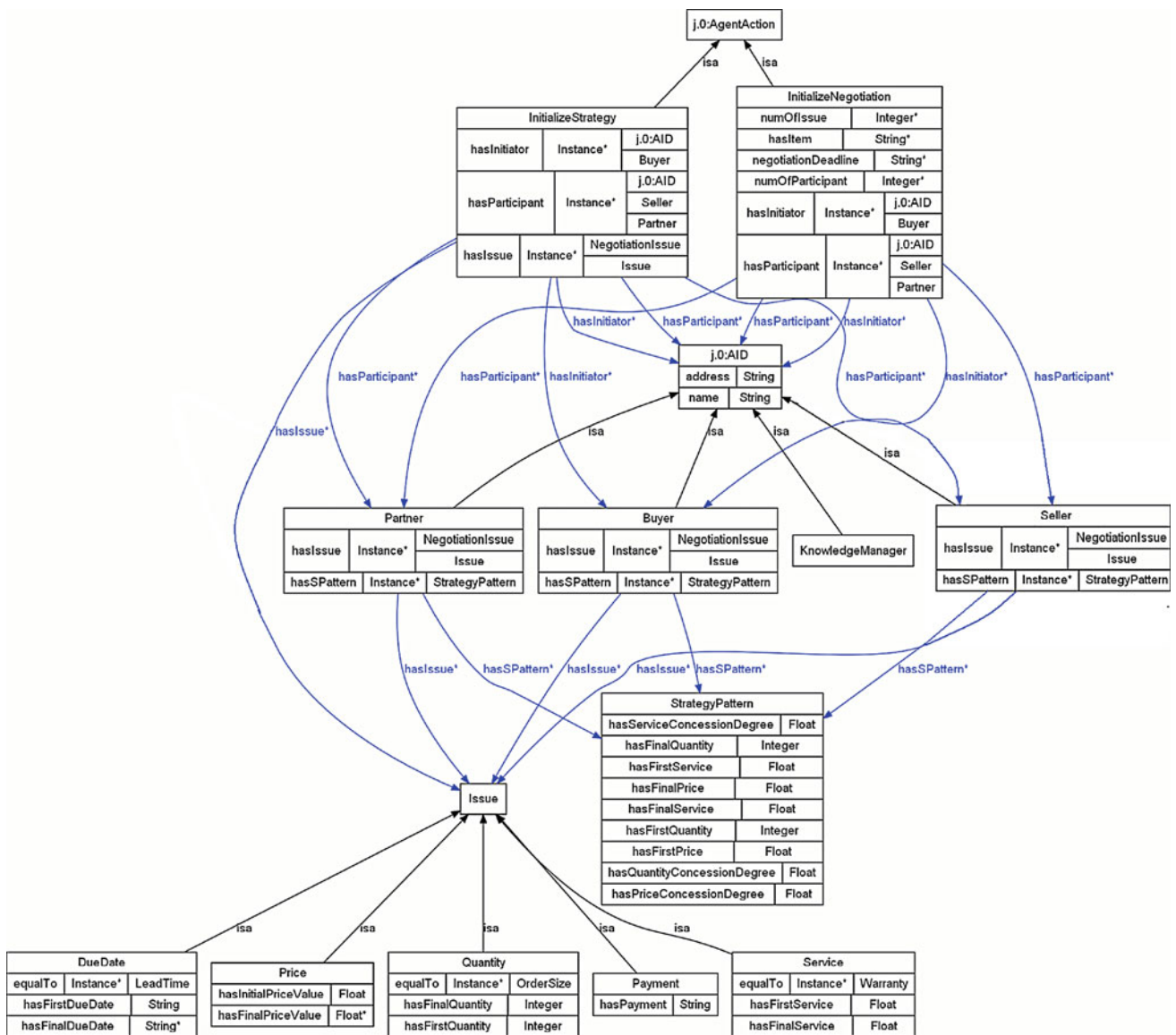


Fig. 2 Ontology model for the VE initiator, created by Ontoviz plug-in of Protégé 3.3.1

concession degree, *has service concession degree*, which are defined in form of ontology properties. *First* means the first proposed bid; *final* means the agreed bid; *concession degree* means that how much the buyer or seller would like to concede to the counterpart’s proposed bid. For each negotiation issue, there are three strategy categories for concession degree, i.e. Low, Medium and High.

Ontology-based MAS

Semantic ontology matching

The purpose of ontology matching is to find out correspondences between separate ontologies by semantic similarities. In doing so, semantic interoperability among participants in

MAS will be achieved. Many researchers introduce ontology matching system to demonstrate the ontology matching process (Calvanese et al. 2002; Haase and Motik 2005; Trojahn et al. 2008). From the perspective of MAS negotiations in VE, an ontology matching system for this study is introduced.

Definition 1 (*Ontology Matching System, OMS*) Suppose O and O' are two ontologies, which are defined as $O = \langle C, R, I \rangle$ and $O' = \langle C', R', I' \rangle$. Here, C and C' stand for the set of classes in ontologies O and O' respectively; R and R' stand for the set of properties in ontologies O and O' respectively; I and I' stand for the set of instances in ontologies O and O' respectively. An ontology matching system OMS is defined as a triple (O, O', M) . M is the set of correspondences between ontologies O and O' , which

is defined as a triple (e, e', σ) . Here, e and e' are classes or properties from C and C' , or from R and R' ; σ is similarity value between e and e' .

In this section, a series of ontology matching approaches are introduced according to the *OMS*, namely *Name based Term Matching (n-TM)*, *Structure and Constraint based Property Matching (sc-PM)*, and *Distance between Data Patterns of Instance based Property Matching (di-PM)*.

Many researchers have adopted *n-TM* as a fundamental method to calculate semantic similarity of two separate terms (Rahm and Bernstein 2001; Chen and Chen 2009; Wu and Wang 2007). In this approach, class and property of heterogeneous ontologies should be stemmed and pre-processed into atomic terms (Xu and Croft 1998). A public dictionary is then required to calculate the semantic similarities between different terms. In this paper, WordNet dictionary is introduced. The *n-TM* is introduced as follows:

Name-Based Term Matching (n-TM)

Input: Term e from ontology O , Term e' from ontology O'

Output: $\langle e, e', \sigma \rangle$, correspondence between e and e' and its similarity σ

- (1) Initialize similarity value $\sigma(e, e') = 0$;
- (2) Get all the senses and their hypernyms of e and e' respectively, i.e., $e(*)$ and $e'(*)$;
- (3) Calculate the lengths of all paths between e and e' , and get the shortest path. Suppose L is the length of the shortest path between e and e' , then $\sigma_1(e, e')$ can be calculated by Eq. (1):

$$\sigma_1(e, e') = e^{-\alpha L} \tag{1}$$

- (4) Calculate the depths of all terms in set of $e(*)$ and $e'(*)$. Suppose H is the biggest depth between e and e' , then $\sigma_2(e, e')$ can be calculated by Eq. (2):

$$\sigma_2(T, T') = \frac{e^{\beta H} - e^{-\beta H}}{e^{\beta H} + e^{-\beta H}} \tag{2}$$

- (5) Final term similarity can be calculated by Eq. (3):

$$\sigma(e, e') = \omega_1 \sigma_1(e, e') + \omega_2 \sigma_2(e, e') \quad \text{where } \omega_1 + \omega_2 = 1 \tag{3}$$

On the basis of *n-TM*, semantic similarity between any terms can be calculated. During MAS negotiation process, properties are major components that make up of the negotiation messages, such as *quantity*, *price*, *service* etc. Therefore, semantic similarity between properties from different ontologies is the main focus of this study.

From the perspective of ontology definition, ontology structures and constraints defined within each element have significant effect on elements' semantics for two reasons.

Firstly, properties in different hierarchies of classes represent different semantics. Secondly, properties with different data types express different semantic as well. Therefore, *sc-PM* is proposed to calculate property similarity between different ontologies from the perspective of ontology structure and constraint. The *sc-PM* is introduced as follows:

Structure and Constraint Based Property Matching (sc-PM)

Input: Properties r and related classes c in ontology O ; Property r'_0 and related class c'_0 in ontology O' .

Output: $\langle r, r'_0, \sigma_{\max} \rangle$, the highest similarity between properties in ontology O and property r'_0 .

- (1) Find all properties r in ontology O of the same data type with r'_0 and their related classes c ;
- (2) Based on *n-TM*, calculate similarities between qualified properties r and r'_0 using Eq. (4):

$$\sigma(r, r'_0) = \alpha_1 \sigma_1(c, c'_0) + \alpha_2 \sigma_1(c, r'_0) + \alpha_3 \sigma_1(c'_0, r) + \alpha_4 \sigma_4(r, r'_0), \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1 \tag{4}$$

- (3) Then the highest similarity σ_{\max} can be obtained by Eq. (5):

$$\sigma_{\max} = \max(\sigma(r, r'_0)) \tag{5}$$

Example Suppose “Quantity” is a class from Ontology O , it has an integer property as “quantity value”; “Sell” is a class from Ontology O' , it has an integer property as “amount”. According to *sc-PM*, the property similarity between “quantity value” and “amount” is calculated as Eq. (6). Thereinto, the term similarity is calculated according to *n-TM*, whereas $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 1/4$.

$$\begin{aligned} \sigma(\text{quantity value, amount}) &= 1/4 * [\sigma_1(\text{Quantity, Sell}) \\ &\quad + \sigma_2(\text{Quantity, amount}) + \sigma_3(\text{Sell, quantity value}) \\ &\quad + \sigma_4(\text{quantity value, amount})] \\ &= 1/4 * (0.3933 + 0.8332 + 0.3836 + 0.7809) \\ &= 0.5978 \end{aligned} \tag{6}$$

Besides structures and constraints defined within ontologies, contents of instances stored in different ontologies express different semantics to some extent. For example, the value of property “product” in one instance is “watch”, while the value of property “quantity” in another instance is “1000”; they are obviously sharing different data patterns. Therefore, *di-PM* is proposed to calculate property similarities in different ontologies from the perspective of ontology instances. The *di-PM* is introduced as follows:

Distance Between Data Pattern of Instance Based Property Matching (*di-PM*)

1) Contents of instances are per-processed into corresponding data strings according to the rule set:

*Turn all numerals into symbol “0”; Turn all alphabets into symbol “1”; Turn all “,” into symbol “X”; Turn all white space into symbol “Y”; Turn all “http://” into “*****”; Turn “@”, “-”, “:”, “/”, “.” into symbol “#”.*

2) Concrete symbols are defined for data strings: For buyers, *SB* stands for a string from the buyer; $|SB|$ stands for the length of *SB*; SB_i stands for the *i*th character of *SB*, for an integer $i \in \{1 \dots |SB|\}$; and $SB_{i\dots j} = SB_i^S B_{i+1} \dots SB_j$ stands for a partial string from *SB*, where $i > j$. For sellers, *SS* stands for a string from the seller; $|SS|$ stands for the length of *SS*; SS_i stands for the *i*th character of *SS*, for an integer $i \in \{1 \dots |SS|\}$; and $SS_{i\dots j} = SS_i^S S_{i+1} \dots SS_j$ stands for a partial string from *SS*, where $i > j$;

3. According to Navarro (2001)’s string distance theory, distance between strings *SB* and *SS* is defined as the minimal cost of operations that transform *SB* into *SS*, indicated by $d(SB, SS)$. Here, the operation refers to delete, insert or substitute a character. Therefore, the *di-PM* is described as below:

Input: Instance strings *SB* and *SS*

Output: Distance between input strings, $d(SB, SS)$

Define a matrix $C_{|SB|,|SS|}$, where $C_{i,j}$ represents the minimum number of operations needed to transform $SB_{1\dots i}$ into $SS_{1\dots j}$, here, $C_{i,0} = i$ $C_{0,j} = j$. The computation is shown as below:

$$C_{i,j} = \begin{cases} C_{i-1,j-1}, & \text{if}(SB_i = SS_j) \\ 1 + \min(C_{i-1,j}, C_{i,j-1}, C_{i-1,j-1}), & \text{if}(SB_i \neq SS_j) \end{cases} \quad (7)$$

$$d(SB, SS) = C_{|SB|, |SS|} \quad (8)$$

Example Suppose that “*Product*” is an instance of class “*Product*” in Ontology *O*, it has a property as “*name*” with content “*watch band*”; “*Article*” is an instance of class “*Article*” in Ontology *O'*, it has a property as “*model number*” as “*JEB762*”. The semantic distance is calculated as follows:

- Transform property content into strings: “*watch band*” → “11111Y1111”; “*JEB762*” → “111000”;
- According to Equations (7–8), distance between “11111Y1111” and “111000” is 7.

Correspondence-based extended contract net protocol

Correspondences are similar property pairs generated through ontology matching methods proposed in Section

“Semantic ontology matching”. To realize MAS communication and service knowledge negotiation under circumstances of mutual understanding, a correspondence-based extended contract net protocol is presented. It is developed on the basis of the traditional Contract Net Protocol (CNP), which is a prevalent agent interaction protocol that defines basic message types in accordance with three consecutive phases, i.e. inviting, bidding and awarding. The traditional CNP confines the interaction thread to the one-to-one bidding scenario, without considering the semantic interoperability between agents. The proposed correspondence-based extended CNP is contributing in two aspects. Firstly, it introduces a third party agent namely Knowledge Manager Agent (*KMA*), which is responsible to solve semantic interoperability problems between *BA* and *SAs*. Secondly, two more scenarios as correspondence retrieval and correspondence candidate translating scenarios are considered comparing to the traditional bid negotiation scenario. It introduces other partner agents to provide additional aids in reaching semantic interoperability, which will increase the feasibilities of reaching successful mutual understanding.

The developed correspondence-based extended CNP is as shown in Fig. 3. *BT* is a term in buyer’s terminology; *ST_j* is a term in the target seller’s terminology; *Sagent* is an agent participating in negotiations using (*BT*, *ST_j*) as a correspondence; *TT* is a term in the target seller’s terminology; *translated* indicates that (*BT*, *TT*) is a correspondence candidate, not a correspondence; *RT_i* is a term in form of (*BT*, *RT_i*) in other partner’s *correspondence libraries*. Interpretation for the protocol is provided in Table 1.

As shown in Fig. 3, this protocol comprises three scenarios, i.e. correspondence retrieval, correspondence candidate translation and bid negotiation.

- Correspondence Retrieval: For a certain term *BT* in the service query, there is another term *ST_j* appearing in the buyer’s *correspondence library* as a correspondence pair (*BT*, *ST_j*). If this correspondence pair also exists in the target seller’s *correspondence library*, this pair will be directly retrieved for later negotiation usage. If not, the target seller’s *correspondence library* should be updated accordingly.
- Correspondence candidate translation: If *BT* does not exist in the buyer’s *correspondence library* but in the *correspondence candidate library*, then pair (*BT*, *TT*) will be retrieved and sent to *SA*. If it appears in the seller’s *correspondence library*, *SA* will inform *BA* and *BA* will update its *correspondence library* with pair (*BT*, *TT*). If not, *SA* will query other partners. If it appears in other partners’ *correspondence library* (in other words, other partners have used (*BT*, *TT*) as a correspondence during past negotiations), *SA* will inform the *BA* and *BA* will update its *correspondence library* as well. However, if no

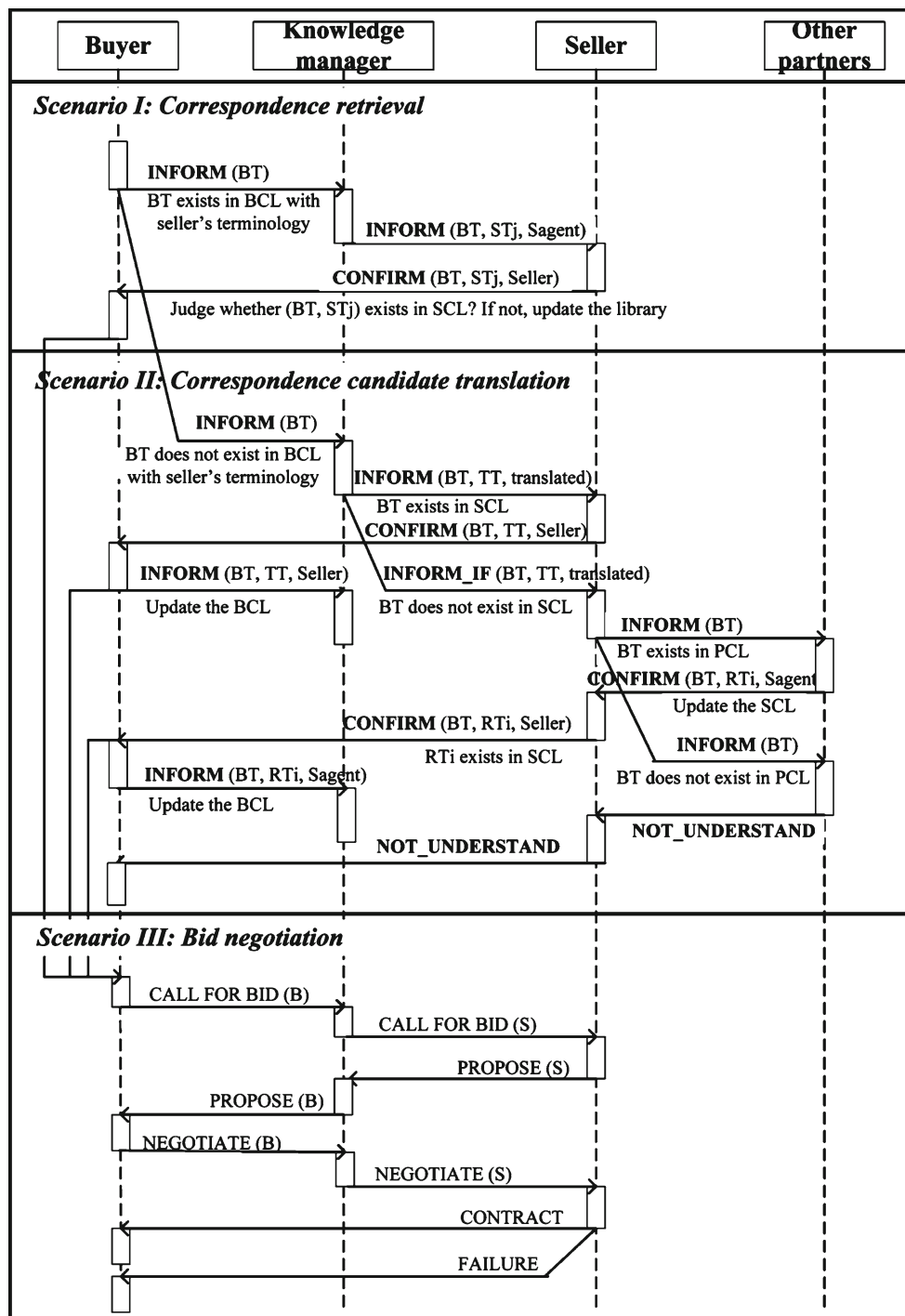


Fig. 3 Correspondence based negotiation protocol

other partners have used this pair before, SA will send a “Not Understand” message to BA, and the negotiation process terminates.

- Bid negotiation: After all terms can be understood by all participants, the bid negotiation process will be started.

Ontology-based knowledge reuse using fuzzy set theory

Knowledge management is becoming a core competitiveness of enterprises in today’s service-dominant business world. In the study of MAS negotiations in VE, the existing service knowledge reveals the concession strategies that

Table 1 Interpretation of negotiation messages

Message name	Functionality
INFORM	Send a term to the receiver agent Ask it to check its correspondence library using the received information
CONFIRM	Confirm the correspondence with the initiator
NOT_UNDERSTAND	No correspondence is in the partner's library
CALL FOR BID	Initialize the negotiation with a preliminary bid
PROPOSE	Respond to CALL FOR BID message
NEGOTIATE	Negotiate with each other
CONTRACT	Reach agreement and contract
FAILURE	No agreement is made

counterparts adopted in previous trade-offs. For different issues, different counterparts will adopt different strategies to concede. It's therefore necessary for the VE initiator to study the existing strategies and reuse them to evaluate the current trade-offs.

Different enterprises have different focuses towards different negotiation issues. Some may be sensitive to price fluctuation, some may be serious about the lead time, quantity, or service offered. For issues with different importance, enterprises will concede differently. This paper introduces the concept of *average concession degree* as an evaluator, indicating that: The buyer would accept the seller's concession degree if it is equal to or higher than the *average concession degree* of the homogenous negotiations; otherwise, the buyer might reject the current bid and negotiate further.

In this study, the existing service knowledge for its partners is stored as instances of the VE initiator's ontology. The VE initiator can utilize this knowledge to evaluate the behaviors of its negotiation counterparts. Since fuzzy set theory is good at dealing with inaccurate and incomplete information (Zadeh 1965), it is introduced in this paper to utilize the existing knowledge and provide suggestions on the current negotiations.

There are two types of service knowledge sources in the negotiation process. One is the existing service knowledge, stored in the VE initiator's ontology; the other is the current service knowledge. To illustrate the basic idea of the fuzzy set theory based knowledge reuse approach, a case study involving negotiations between one BA and three SAs is introduced. In this study, three negotiation issues, i.e. *price*, *quantity* and *service*, are considered in the formation of a VE. Among them, *price* and *quantity* are cost-oriented issues while *service* is a benefit-oriented issue. For the buyer's self interest, the value of a cost-oriented issue

should be lower while that of a benefit-oriented issue should be higher.

To present the calculation flow of the proposed approach, the buyer and seller 1's negotiation process is studied in Fig. 4. Each step in Fig. 4 is explained as follows:

- Step 1: Both buyer and sellers' proposed bids at every round ($r = 1, 2, \dots, final$) are recorded;
- Step 2–3: When the negotiation comes to the $r = 10$ round, according to the current concession degree given by Eq. (9) and the membership function given in Fig. (4) and Eq. (10), the membership degree at the $r = 10$ round can be calculated; the result is shown in Step 3 in Fig. 4.

$$\begin{aligned}
 & con_degree(I_i, S_j, r) \\
 &= \frac{bid(I_i, S_j, 1) - bid(I_i, S_j, r)}{bid(I_i, S_j, 1) - bid(I_i, B, 1)} \tag{9}
 \end{aligned}$$

whereas, $bid(I_i, S_j, 1)$ indicates the first bid of seller S_j towards issue I_i ; $bid(I_i, S_j, r)$ indicates the r th bid of seller S_j towards issue I_i ; $bid(I_i, B, 1)$ indicates the first bid of buyer towards issue I_i .

$$\begin{aligned}
 & membership(con_degree_{min}, con_degree_{middle}, \\
 & con_degree_{max}) \\
 &= \begin{cases} (0, 0.17, 0.48), & \text{the } con_degree(I_i, S_j) \text{ is low} \\ (0.3, 0.5, 0.7), & \text{the } con_degree(I_i, S_j) \text{ is medium} \\ (0.52, 0.83, 1), & \text{the } con_degree(I_i, S_j) \text{ is high} \end{cases} \tag{10}
 \end{aligned}$$

- Step 4: The records of historical negotiation cases at different rounds r ($r = 1, 2, \dots, final$) are classified into different strategy patterns according to Eq. (11). In each strategy pattern (low, medium, high), the actual concession degrees for each case are recorded, and the average final concession degree is calculated based on Eq. (12) for different issues.

$$\begin{aligned}
 & con_degree_pattern(I_i, r) \\
 &= \begin{cases} low, & \text{if } 0 \leq con_degree(I_i, r) < 0.33 \\ medium, & \text{if } 0.33 \leq con_degree(I_i, r) < 0.67 \\ high, & \text{if } 0.67 \leq con_degree(I_i, r) \leq 1 \end{cases} \tag{11}
 \end{aligned}$$

$$\begin{aligned}
 & average_con_degree(I_i, SP_k, final) \\
 &= \frac{\sum_{n=1}^{n=n_k} con_degree(I_i, final)}{n_k} \tag{12}
 \end{aligned}$$

whereas, SP_k is the k th strategy pattern, n_k is the amount of the existing records of service knowledge stored in SP_k .

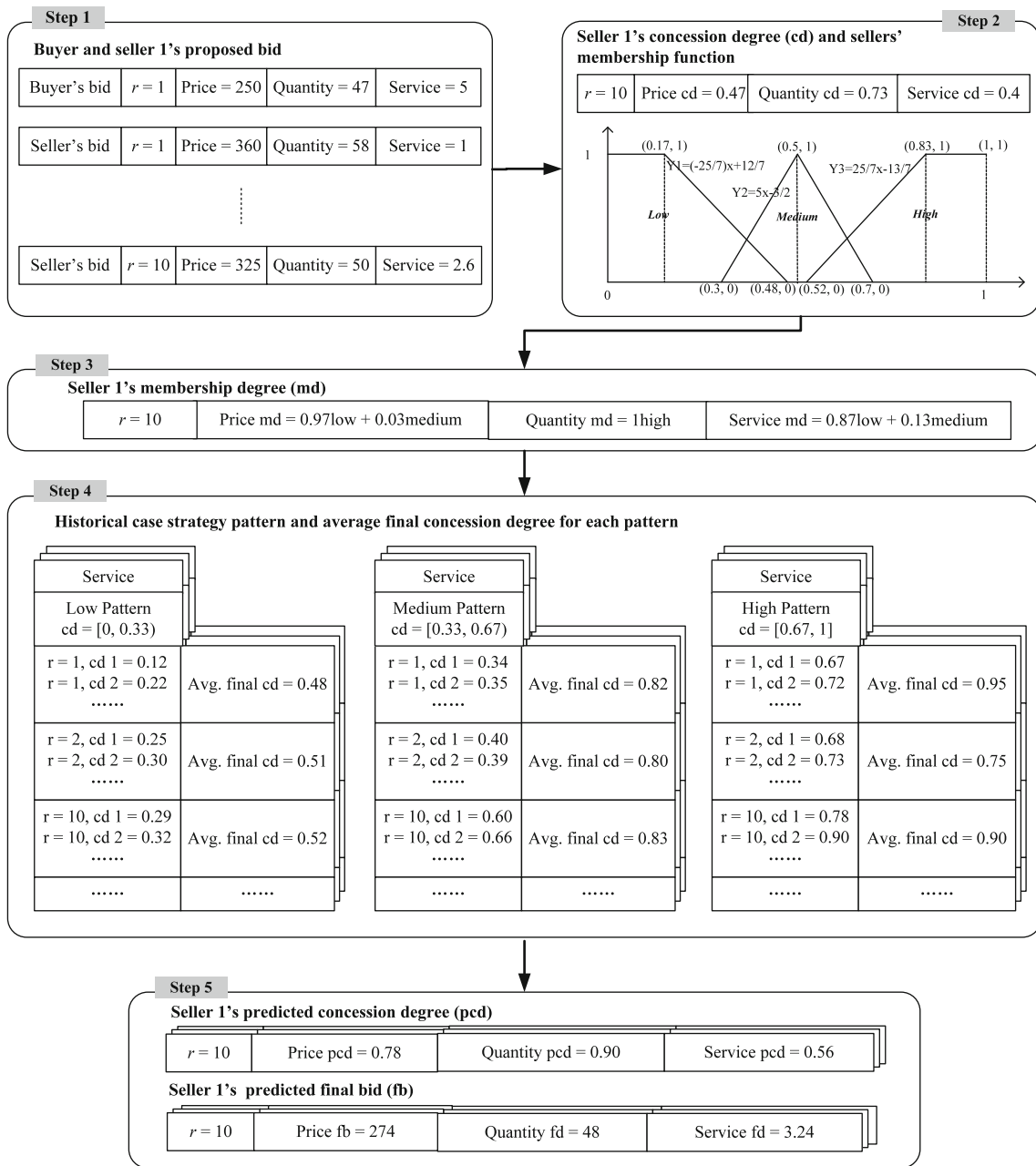


Fig. 4 Basic calculation flow of fuzzy set theory based knowledge reuse approach

- Step 5: As shown in Step 3 in Fig. 4, for seller's service membership degree at the r th round, the seller's concession degree constitutes 0.87 of the low service strategy pattern, and 0.13 of the medium service strategy pattern. Based on the average final concession degrees of service as shown in Step 4 in Fig. 4, i.e. average final concession degree for low service strategy is 0.52, and average final concession degree for medium service is 0.83. Therefore, the seller's predicted concession degrees can be calculated by Eq. (13); and the seller's predicted bids can be calculated by Eq. (14). The results are shown in Step 5 in Fig. 4.

$$\begin{aligned}
 & \text{predicted_con_degree}(I_i, S_j, \text{final}) \\
 &= v_a \times \text{average_con_degree}(I_i, SP_a, \text{final}) \\
 &+ v_b \times \text{average_con_degree}(I_i, SP_b, \text{final}) \quad (13)
 \end{aligned}$$

whereas, v_a is the membership degree of S_j belonging to strategy SP_a , while v_b is the membership degree of S_j belonging to strategy SP_b .

$$\begin{aligned}
 & \text{predicted_bid}(I_i, S_j, \text{final}) \\
 &= \text{bid}(I_i, S_j, 1) \pm \text{predicted_con_degree} \\
 & (I_i, S_j, \text{final}) \times |\text{bid}(I_i, S_j, 1) - \text{bid}(I_i, B, 1)| \quad (14)
 \end{aligned}$$

Table 2 Negotiation bids for buyer and sellers

	BA's initial bids			SAs' first bids			SAs' current bids ($r=10$)		
	Price	Quantity	Service	Price	Quantity	Service	Price	Quantity	Service
SA1	250	47	5	360	58	1	325	50	2.6
SA2	675	100	5	800	130	1	738	119	2.6
SA3	1000	100	5	1300	125	2	1198	105	4

Table 3 Current concession degree and membership degree for sellers, $r = 10$

	Current concession degree			Membership degree		
	Price	Quantity	Service	Price	Quantity	Service
SA1	0.32	0.73	0.40	0.97 low+0.03 medium	High	0.87 low+0.13 medium
SA2	0.50	0.55	0.40	Medium	0.98 medium+0.20 high	0.87 low+0.13 medium
SA3	0.34	0.80	0.67	0.97 low+0.03 medium	High	0.87 medium+0.13 high

where, “+” indicates that it is a benefit-oriented problem; “-” indicates that it is a cost-oriented problem.

In this study, the sellers are assumed to be restricted to follow a same concession strategy throughout the negotiation. Therefore, suggestions are given to the buyer under this assumption: The buyer should accept the final bid if it is equal to or lower than the predicted final bids in a cost-oriented problem; or accept the final bid if it is equal to or higher than the predicted final bids in a benefit-oriented problem. Otherwise, it should make further negotiations with the sellers.

Experiment results for all sellers are displayed in Tables 2, 3, 4, 5. To evaluate the performance of the proposed approach, Table 6 depicts the actual bids and comparisons between actual and predicted bids. In this regard, for cost-oriented issues Price and Quantity, it is the buyer's self-interest to reduce the values of the actual bids. Thus, if a comparison bid (actual bid minus predicted bid) is a positive value, the buyer would prefer to use the predicted bid. For instance, the buyer predicts the SA1's price bid as 274 but a seller's actual bid price is 278, the comparison bid is then $278 - 274 = 4$ and the buyer would reject the seller's bid. For a benefit-oriented issue, in contrast, a higher actual bid value incurs better performance. Therefore, the buyer would prefer to use the predicted bid if the comparison bid is a negative value. For instance, SA2's predicted service level is 3.24 and a seller's actual bid value on service is 3.2, the comparison bid is then $3.2 - 3.24 = -0.04$ and the seller's bid would not be accepted by the buyer as the actual service level is lower (worse) than the predicted bid.

Table 6 presents the comparison of the actual and predicted bids from the three sellers. Regarding the price issue, all actual bids of SA1, SA2 and SA3 are, respectively, 4, 10 and 34 dollars higher than the predicted bids. For the

Table 4 Average final concession degree of each strategy pattern

	Price	Quantity	Service
Low	0.79	0.50	0.52
Medium	0.59	0.80	0.83
High	0.59	0.90	0.90

quantity issue, the actual bid from SA3 is 2 units more than the predicted one. For the service issue, the service level of SA1 is 0.76 higher than the prediction while that of SA2 is 0.04 lower. Thus, performance of the actual bids does not yet meet the predictions such that actual bids of SA1's price, SA2's price and service, and SA3's price and quantity, are worse than the predicted ones. Therefore, as indicated in the last column of Table 6, the number of issues that can be improved for SA1, SA2, and SA3 are 1, 2, and 2 respectively. The values of these issues are to be rejected and replaced by the predicted bids by the buyer, and the buyer would like to use the predicted bids to negotiate with the sellers for further improvement. The buyer and sellers will reach agreement on new bids afterwards. Overall, with the adoption of the fuzzy set theory based knowledge reuse method, 55.56% negotiation bids might be rejected by BA. Namely, after evaluating the current bid by reusing the historical negotiation knowledge, the buyer can decide whether or not it can benefit the most from the current bid, and then the buyer's negotiation performance can be improved accordingly.

System implementation and evaluation

Performance of the fuzzy set theory based knowledge reuse method was evaluated in Section “Ontology-based knowledge reuse using fuzzy set theory”. In this section, a walk-

Table 5 Predicted concession degree and predicted final bid for sellers

	Predicted concession degree			Predicted bid		
	Price	Quantity	Service	Price	Quantity	Service
SA1	0.78	0.90	0.56	274	48	3.24
SA2	0.59	0.80	0.56	726	106	3.24
SA3	0.78	0.90	0.84	1066	103	4

Table 6 Results for SAs 1-3 (*Comparison bid = Actual bid – Predicted bid*)

	Actual bid			Comparison bid			Number of issue to be improved
	Price	Quantity	Service	Price	Quantity	Service	
SA1	278	48	4	4	0	0.76	1
SA2	736	106	3.2	10	0	-0.04	2
SA3	1100	105	4	34	2	0	2
Sum of issues				9			5
Issues might be rejected (percentage)				5 / 9 = 55.56%			

Numeric in bold and italic indicates that it can be improved by adopting the fuzzy set theory based knowledge reuse method

through example is introduced to test the performance of the ontology matching methods and the feasibility of correspondence-based contract net protocol.

The case scenario is that a watch company acting as the VE initiator wants to purchase some components from its upper tier supply chain partners. Suppose that there are three potential partners which would like to join the VE. In this way, the MAS is a system with one BA and three SAs. The set of buyer ontology was illustrated in Fig. 2. Definition of the seller ontology is shown in Fig. 5 which presents the classes and properties of the seller ontology model and the “isa” (“is a”) labeled arrows denote the hierarchy relationships between two classes. To simplify for illustration, it is assumed that three sellers share the same definitions of classes and properties but different instances. In other words, the definition of seller ontology shown in Fig. 5 is shared by all sellers. However, content of the data type property values, such as contents of *has product*, *has payment pattern* and *has Company Name* etc. are specific for different sellers.

As a simple example, Table 7 lists part of the property contents for BA, SA1, SA2 and SA3, and the corresponding property contents distance results. Three pairs of “String” type properties are listed. For BA, the properties are “*has Item*”, “*has Payment*” and “*has Name*”. For SAs, the corresponding properties are “*has Product*”, “*has Payment Pattern*” and “*has Company Name*”.

During the ontology matching process, to generate a more accurate ontology matching result, different property contents can be utilized to identify the differences of the heterogeneous terminologies, using the *di-PM* presented in Section “Semantic ontology matching”. As shown in Table 7, contents of the different properties for different sellers are totally

different. These “String” type properties are then evaluated and compared. Numerics in Column “Distance (*BA, SAs*)” in Table 7 indicate the distances between contents of BA and SAs’ properties. Numerics in Column “Distance rank” indicate the actual ranks of distances of the appointed content pairs in the distance result list. For instance, the content of BA’s property “*has Item*” is “Watch band”. It is compared to the respective contents of three properties (“*has Product*”, “*has Payment Pattern*” and “*has Company Name*”) of SA1. Based on *di-PM*, the corresponding distances are calculated to be 1, 9 and 21, respectively. Therefore, the content “Watch strap” has the closest distance with the content “Watch band”. Consequently, among all SA1’s properties, “*has Product*” is the most similar property, to BA’s property “*has Item*”.

Section “Performance of semantic matching algorithm” is to evaluate the performance of the semantic matching methods proposed in Section “Semantic ontology matching”. Based on the correspondences obtained from Section “Performance of semantic matching algorithm”, Section “Protocol implementation within JADE-based MAS Platform” tests the proposed correspondence-based CNP in four scenarios.

Performance of semantic matching algorithm

From Figs. 2 and 5, ontologies have heterogeneous definitions for class, property and instance. To mutually understand each other, semantic ontology matching is required. As specified in Section “Semantic ontology matching”, semantic similarity between properties can be calculated by *sc-PM* and *di-PM*. In this section, a third method named *combinational property matching (c-PM)* is proposed to leverage the performances of *sc-PM* and *di-PM*. The result value of *c-PM*

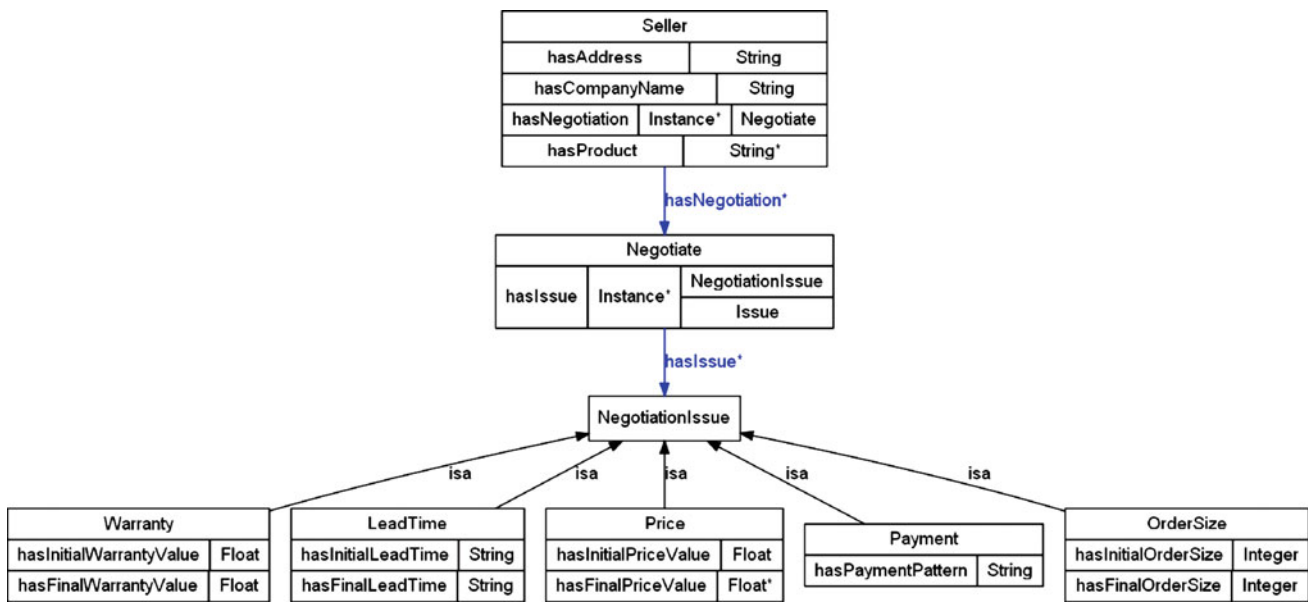


Fig. 5 Ontology model for the Seller, created by Ontoviz plug-in of Protégé 3.3.1

Table 7 Contents and distances between BA and SAs, using di-PM to calculate the distances

BA's property name	BA's property content	SAs	SAs' property name	SAs' property content	Distance (BA, SAs)	Distance rank
Has item	Watch band	SA1	Has product	Watch strap	1	1
			Has payment pattern	T/T, D/P	9	2
			Has company name	AAA Watch component corporation	21	3
Has payment	T/T, D/P	SA2	Has product	Watch band	9	2
			Has payment pattern	T/T	4	1
			Has company name	BBB company of Watch industry	26	3
Has name	No.1 Watch manufacturing factory	SA3	Has product	Wrist band	22	2
			Has payment pattern	T/T, D/P, L/C	25	3
			Has company name	Watch CCC Co. Ltd	16	1

is given on basis of the results of *sc-PM* and *di-PM* as shown in Eq. (15).

$$Value_{c-PM} = w_1 \times Value_{sc-PM} + w_2 \times Value_{di-PM} \tag{15}$$

To select the correspondence candidate, proper threshold of semantic similarity results should be determined. The setting of threshold will limit the output of correspondence candidates, and accordingly reduce the algorithm complexity. This paper introduces two approaches to set threshold: One is using the value of semantic similarity as the threshold; the other is using rank No.1 of similarity value or instance distance or combinational result as the threshold.

To evaluate the performance of the proposed ontology matching methods, four typical performance evaluators in the domain of information retrieval are introduced (Islam and Inkpen 2008):

$$\text{Precision (P)} : P = TP / (TP + FP) \tag{16}$$

$$\text{Recall (R)} : R = TP / (TP + FN) \tag{17}$$

$$\text{F - Measure (F)} : F = 2PR / (P + R) \tag{18}$$

$$\text{Accuracy (A)} : A = (TP + FP) / (TP + FP + FN) \tag{19}$$

Here, TP stands for True Positive (namely how many correspondences were selected with right meanings); FP stands for False Positive (namely how many correspondences were not selected, which are actually with right meanings); FN stands for False Negative (namely how many correspondences were

selected with wrong meanings). Regarding the performance evaluation, a better ontology matching performance requires a bigger TP, a smaller FP and FN. The values of TP, FP and FN and the performance of ontology matching methods are determined by the threshold setting method, which can determine the output of semantic similarity calculation results.

The objective of this study is to find out the most appropriate property matching method with the most appropriate threshold setting method. Therefore, four experiments are conducted:

- *sc-PM* using the value of semantic similarity as threshold
- *sc-PM* using rank No.1 of similarity value as threshold (*sc-PM* rank)
- *di-PM* using rank No.1 of instance distance as threshold (*di-PM* rank)
- *c-PM* using rank No.1 of combinational result as threshold (*c-PM* rank)

Based on the *sc-PM* and Equations (16–19), the performance of *sc-PM* using semantic similarity as the threshold is shown in Table 8 and Fig. 6. Take the threshold $T = 0.1$ as an example, all the term similarity results that are higher than 0.1 are considered as correspondences. In this regard, the human ontologists identify that there are 16 correspondences that are selected with right meanings ($TP = 16$); there is 0 correspondence with right meanings but are not selected ($FP = 0$); and there is 0 correspondence with wrong meanings but are selected ($FN = 0$). Therefore, the calculated values of precision, recall, F-measure and accuracy are equal to be 1.

Table 8 and Fig. 6 indicate that the performance result is very sensitive with the threshold (T) changing, especially when T is set to be higher than 0.5. Consequently, it is not proper to adopt the value of semantic similarity as threshold to decide the selection of correspondence candidates.

Tables 9 and 10 display partial correspondence properties used in this experiment, and the similarities, ranks and

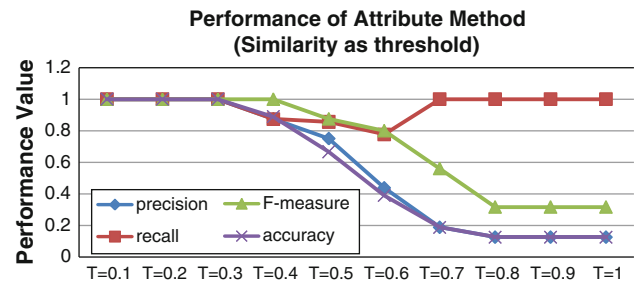


Fig. 6 Performance of *sc-PM* on precision/recall/F-measure/accuracy for all sellers

the performance of *sc-PM* method (Table 9), *di-PM* method and *c-PM* method (Table 10) using rank No.1 as threshold. Here, *sc-PM* rank, *di-PM* rank and *c-PM* rank are actual ranks of the appointed correspondences ranking in the result lists. The *sc-PM* rank is the rank of semantic similarity results calculated by *sc-PM* method; the *di-PM* rank is the rank of instance distance results calculated by *di-PM* method; the *c-PM* rank is the rank of combinational results calculated by *c-PM* method using Eq. (15). Rank No.1 method means that only the correspondence ranking at the first place of the result lists is considered as the right correspondence. The precision, recall, F-measure and accuracy are calculated based on the Rank No.1 method. As specified at the beginning of Section “System implementation and evaluation”, it is assumed that three sellers share the same definitions of classes and properties but different instances. In other words, for *sc-PM* method, three sellers share the same knowledge structures of the seller’s ontology, it’s therefore that the similarity and rank results are shared by three sellers, as shown in Table 9; for *di-PM* and *c-PM* methods, the contents of classes and properties of the seller’s ontology are different from each other, it’s therefore that the similarity and rank results are different as well, as shown in Table 10.

Regarding the result value of *c-PM* given in Eq. (15), w_1 and w_2 are set to be 0.5 in this case.

Table 8 Evaluating results of *sc-PM* using semantic similarity as the threshold

Threshold	TP	FP	FN	Precision	Recall	F-measure	Accuracy
$T = 0.1$	16	0	0	1	1	1	1
$T = 0.2$	16	0	0	1	1	1	1
$T = 0.3$	16	0	0	1	1	1	1
$T = 0.4$	14	2	2	0.875	0.875	1	0.8889
$T = 0.5$	12	4	2	0.75	0.8571	0.875	0.6667
$T = 0.6$	7	9	2	0.4375	0.7778	0.799981	0.3889
$T = 0.7$	3	13	0	0.1875	1	0.560006	0.1875
$T = 0.8$	2	14	0	0.125	1	0.315789	0.125
$T = 0.9$	2	14	0	0.125	1	0.315789	0.125
$T = 1$	2	14	0	0.125	1	0.315789	0.125

Table 9 Partial *sc-PM* similarity and rank results for correspondences between buyer and sellers

Buyer		Sellers	
		<i>sc-PM</i> Similarity	<i>sc-PM</i> Rank
Correspondence property (Right correspondence in reality)			
Issue	Issue	0.3938	5
Item	Product	0.4414	3
Name	Name	1	1
Address	Address	1	1
Payment	Payment pattern	0.5429	1
Due date	Lead time	0.6195	2
Service	Warranty	0.3618	4
Quantity	Order size	0.6389	3
Price	Price	1	1
Precision			0.44
Recall			0.44
F-measure			0.44
Accuracy			0.64

The values of *sc-PM* rank, *di-PM* rank and *c-PM* rank in Tables 9 and 10 are calculated in two steps. Take buyer’s property “issue” as an example. Firstly, “issue” is compared with all of the seller’s properties (i.e. issue, product, name, address, payment pattern, lead time, warranty, order size, price) based on *sc-PM*, *di-PM* and *c-PM* methods. Secondly, according to the calculation results, the pairs of buyer and seller’s correspondence properties are ranked, i.e. *sc-PM* rank, *di-PM* rank and *c-PM* rank.

In Table 9, according to the *sc-PM* method, the *sc-PM* similarities between “issue” and all of the seller’s properties (i.e. issue, product, name, address, payment pattern, lead time, warranty, order size, price) are 0.3938, 0.4145, 0.3505, 0.3443, 0.4548, 0.4215, 0.2486, 0.4477, and 0.3387, respectively. The similarity between “issue” and “issue” ranks the 5th place in the descending ranking list. Therefore, the *sc-PM* rank for correspondence pair “issue, issue” is 5. In the same way, *sc-PM* ranks for the rest correspondence pairs can be obtained as well.

In Table 10, according to the *di-PM* method, the *di-PM* instance distances between “issue” and all of the seller 1’s properties (i.e. issue, product, name, address, payment pattern, lead time, warranty, order size, price) are 0, 8, 6, 43, 4, 5, 2, 5, and 6, respectively. The distances between “issue” and “issue” ranks the 1st place in the ascending ranking list. Therefore, the *di-PM* rank for correspondence pair “issue, issue” is 1. In the same way, *di-PM* ranks for the rest correspondence pairs can be obtained as well.

According to the *c-PM* method shown in Eq. (15), the *c-PM* results between “issue” and all of the seller 1’s properties (i.e. issue, product, name, address, payment pattern, lead time, warranty, order size, price) are 4.5, 6, 5.5, 7.5, 2, 3.5, 5.5, 3, and 6.5, respectively. The combinational result between “issue” and “issue” ranks the 4th place in the ascending ranking list. Therefore, the *c-PM* rank for correspondence pair “issue” and “issue” is 4. In the same way, *c-PM* ranks for the rest correspondence pairs can be obtained as well.

The result indicates that when T is set to be the rank No. 1 correspondence candidate, the performance of *sc-PM* is the worst and that of *di-PM* is the best. However, *di-PM* is less restrained and contents of instances can be readily modified

Table 10 Partial *di-PM* rank and *c-PM* rank for correspondences between buyer and sellers 1–3

Buyer	Sellers	Seller 1		Seller 2		Seller 3	
		<i>di-PM</i> rank	<i>c-PM</i> rank	<i>di-PM</i> rank	<i>c-PM</i> rank	<i>di-PM</i> rank	<i>c-PM</i> rank
Correspondence property (Right correspondence in reality)							
Issue	Issue	1	4	1	1	1	1
Item	Product	1	1	1	1	3	2
Name	Name	1	1	1	1	1	1
Address	Address	1	1	1	1	1	1
Payment	Payment pattern	4	4	1	1	2	1
Due date	Lead time	1	1	1	1	2	2
Service	Warranty	3	3	1	3	1	3
Quantity	Prder size	2	2	1	1	1	2
Price	Price	1	1	1	1	1	1
Precision		0.67	0.56	1	0.89	0.67	0.56
Recall		0.67	0.56	1	0.89	0.67	0.56
F-measure		0.67	0.56	1	0.89	0.67	0.56
Accuracy		0.75	0.69	1	0.90	0.75	0.69

manually, the performance cannot be very stable. For this reason, *combination -PM* using rank No.1 of combination ranks as threshold is selected to leverage *sc-PM* and *di-PM* in a well-performed way, and generate correspondence properties for VE partners in MAS negotiations.

Protocol implementation within JADE-based MAS platform

Section “Correspondence-based extended contract net protocol” presents a correspondence-based extended CNP, which is aiming at providing a mutual understanding environment to facilitate agents’ communication and negotiation based on the traditional CNP. Figure 3 illustrates that the proposed protocol is developed into three scenarios, whereas correspondence retrieval and correspondence candidate translation are aiming at reaching semantic interoperability among the VE participants; while bid negotiation is developed on the basis of the traditional CNP for VE participants to reach agreement on the cooperation issues to form a VE. In this section, based on the correspondences obtained in Section “Performance of semantic matching algorithm”, four experiments are conducted to evaluate the feasibility of the former two scenarios. The results are shown in Fig. 7a–d, which are tracing diagrams of agents’ interactions, created by the sniffer agent of the JADE platform.

In this study, *BA* wants to purchase “a minimum quantity = 47” pieces of products from *SA1*; while *SA1* wants to sell “a minimum order size = 48” pieces of products. The terminologies of *BA* and *SA1* are displayed in Tables 9 and 10. When *BA*’s request is sent to *SA1*, it cannot understand the word “quantity”. Therefore, *KMA* is required to do the translation. The following experiment is to test different behaviors of *SA1*, when the correspondence pair “quantity, order size” exists in different areas, i.e. in the buyer’s *correspondence library*, or in the buyer’s *correspondence candidate library* and seller’s *correspondence library*, or in the partner’s *correspondence library*.

- Directly Confirm: Fig. 7a illustrates a direct confirmation between *BA* and *SA1*. After *KMA* receives “quantity” from *BA*, it finds out that “quantity, order size” exists in the buyer’s *correspondence library*. Therefore, when *SA1* receives “quantity, order size” from *KMA*, it confirms the usage of the correspondence with the *BA* directly.
- Directly Agree: Fig. 7b illustrates a direct agreement between *BA* and *SA1*. After *KMA* receives “quantity” from *BA*, it finds out that there is no correspondence pair using both buyer’s and seller’s terminologies, but there is a correspondence candidate “quantity, order size”. *KMA* informs the seller about the correspondence candidate, and *SA1* finds out that it is in its *correspondence library*.

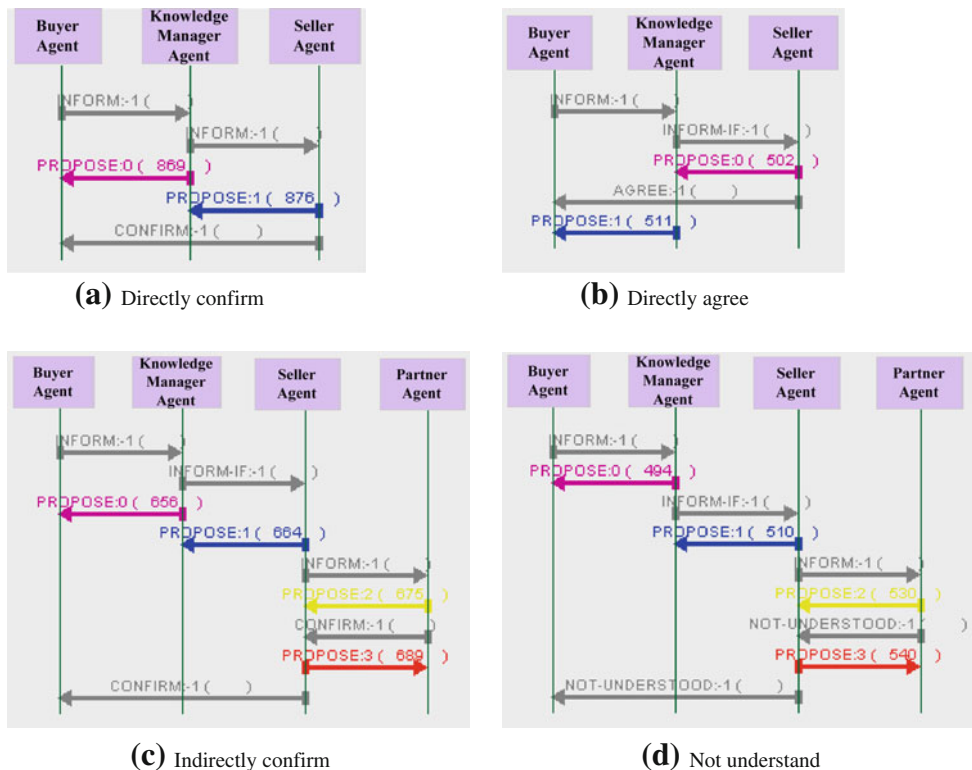


Fig. 7 a Directly confirm. b Directly agree. c Indirectly confirm. d Not understand

Therefore, *SA1* sends an *AGREE* message to *BA* to allow the usage of the correspondence.

- Indirectly Confirm: Fig. 7c illustrates an indirect confirmation between *BA* and *SA1*, with indirect help of *PA*. After *KMA* receives “quantity” from *BA*, it informs a correspondence candidate “quantity, order size” to *SA1*, and *SA1* finds out that it does not exist in its *correspondence library*. Afterwards, *SA1* queries *PA* about “quantity”, and *PA* replies *SA1* with a correspondence “quantity, order size”. In other words, *PA* confirms this correspondence with *SA1*. Then *SA1* confirms it with *BA*.
- Not Understand: Fig. 7d illustrates a not understand message between *BA* and *SA1*. After *KMA* receives “quantity” from *BA*, it informs a correspondence candidate “quantity, order size” to *SA1*, and *SA1* finds out that it does not exist in its *correspondence library*. Afterwards, *SA1* queries *PA* about “quantity”, but no correspondence pair exists in the partner’s *correspondence library* that uses both buyer’s and seller’s terminologies. Consequently, *PA* sends a *NOT UNDERSTAND* message to *SA1*, and *SA1* sends it to *BA*.

In doing so, *BA* and *SA1* can reach agreement on the correspondences to be used in the later bid negotiation scenario, where they can negotiate with each other under a mutual understanding circumstance.

Conclusions and future works

This paper has presented a SOA for ontology supporting MAS negotiations in VE. In this study, enterprises are delegated by agents. To coordinate and enable the services in the VE life cycle, agents need to communicate and negotiate with each other. However, in most cases, different enterprises adopt heterogeneous knowledge; it is therefore difficult for agents to understand each other automatically. For this reason, ontology is adopted to reach semantic interoperability to realize MAS negotiations under a mutual understanding environment. To achieve this objective, this paper is contributing in the following aspects:

- Providing an elaborate and explicit service knowledge description for individual enterprises. Knowledge related to negotiations and self enterprise knowledge are described and stored in individual ontologies. Elaborate and explicit knowledge structure enables individual enterprises to communicate and negotiate in a high efficient way.
- To reach semantic interoperability, three property matching methods are introduced, i.e. Structure and Constraint based Property Matching (*sc-PM*), Distance between Data Patterns of Instance based Property Matching

(*di-PM*), and Combinational Property Matching (*c-PM*). These methods provide solutions to find out property correspondences between any two of individual ontologies in MAS. The property matching methods are the source impetus that allows VE participants to understand each other.

- A correspondence-based extended contract net protocol is developed to regulate the interactions of MAS participants. This protocol comprises three stages: correspondence retrieval, correspondence candidate translation and bid negotiation. The former two stages are the basis of the latter stage in reaching semantic interoperability before real bid negotiation starts.

Besides reaching semantic interoperability, a fuzzy set theory based knowledge reuse method has been established to store and reuse the existing negotiation knowledge. Experiments justify that this method can improve the negotiation performance to some extent.

In the practice of real negotiations in VE, the major difficulty relies in that not all the enterprises adopt ontologies to represent their knowledge, and not all of them would like to share their terminologies during the negotiation process. However, our study is under the assumption that all the enterprises that would like join in the VE utilize ontology to represent their service knowledge, and all of them are willing to provide the terminologies they use.

Another limitation of this paper is the security and privacy of agent enabled SOA in VE. Since different enterprises play different roles, i.e. either buyer agents or seller agents, different roles of agents should be assigned different levels of authority to access the MAS platform. Therefore, future work will be focusing on developing security mechanism for service-oriented MAS architecture in VE.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Bieberstein, N., Bose, S., Fiammante, M., Jones, K., & Shah, R. (2006). *Service-Oriented Architecture (SOA) Compass: Business value, planning, and enterprise roadmap*. Prentice Hall: IBM Press.
- Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). *Measuring semantic similarity between words using web search engines* (pp. 757–766). WWW2007, May 8–12, 2007, Banff, Alberta, Canada.
- Calvanese, D., Giacomo, G. D., & Lenzerini, M. (2002). A framework for ontology integration. *The emerging semantic web: Selected papers from the first semantic working symposium* (pp. 201–214).
- Camarinha-Mahos, L. M., & Afsarmanesh, H. (1999). *The virtual enterprises concept: Infrastructures for virtual enterprise*. Boston, MA: Kluwer.

- Chen, Y. J., & Chen, Y. M. (2009). Development of a mechanism for ontology-based product lifecycle knowledge integration. *Expert Systems with Applications*, 36(2), 2759–2779.
- Chira, O., Chira, C., Roche, T., Tormey, D., & Brennan, A. (2006). An agent-based approach to knowledge management in distributed design. *Journal of Intelligent Manufacturing*, 17(6), 737–750.
- Choi, N., Song, I. Y., & Han, H. (2006). A survey on ontology matching. *ACM SIGMOD Record*, 35(3), 34–41.
- Cognitive Science Laboratory, Princeton University. (2006). WordNet a lexical database for the English language. <http://wordnet.princeton.edu>. Accessed 10 May 2009.
- Dignum, V. (2006). An overview of agents in knowledge management. In Y. Q. Shi (Ed.), *LNCS 4369* (pp. 175–189). Heidelberg: Springer.
- FIPA ACL Message Structure Specification. <http://www.fipa.org/specs/fipa00061/SC00061G.html>.
- Garcia-Sanchez, F., Valencia-Garcia, R., Martinez-Bejar, R., & Fernandez-Breis, J. T. (2009). An ontology, intelligent agent-based framework for the provision of semantic web services. *Expert Systems with Applications*, 36(2), 3167–3187.
- Giovannucci, A., Rodriguez-Aguilar, J. A., Reyes, A., Noria, F. X., & Cerquides, J. (2008). Enacting agent-based services for automated procurement. *Engineering Applications of Artificial Intelligence*, 21(2), 183–199.
- Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2), 199–220.
- Haase, P., & Motik, B. (2005). A mapping system for the integration of OWL-DL ontologies. *IHIS'05*, November 4, 2005, Bremen, Germany.
- Islam, A., & Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, 2(2), Article 10. JADE, <http://jade.tilab.com/>.
- Jennings, N. (2000). On agent-based software engineering. *Artificial Intelligence*, 117(2), 277–296.
- Jiang, P., Mair, Q., & Feng, Z. R. (2007). Agent alliance formation using ART-networks as agent belief models. *Journal of Intelligent Manufacturing*, 18(3), 433–448.
- Kaihara, T., & Fujii, S. (2008). Game theoretic enterprise management in industrial collaborative networks with multi-agent systems. *International Journal of Production Research*, 46(1), 1297–1313.
- Laera, L., Blacoe, L., & Tamma, V. (2007). Argumentation over ontology correspondences in MAS. In *Proceedings of the sixth international joint conference on autonomous agents and multi-agent systems (AAMAS 2007)*, Honolulu, Hawaii, USA (pp. 1293–1300). International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Hawaii, USA. Honolulu:acm
- Lee, J. H., & Kim, C. O. (2008). Multi-agent systems applications in manufacturing systems and supply chain management: a review paper. *International Journal of Production Research*, 46(1), 233–265.
- Li, L., Wu, B. L., & Yang, Y. (2004). An ontology-oriented approach for virtual enterprises. In J.X. Yu et al. (Eds.), *LNCS 3007* (pp. 834–843). Heidelberg: Springer.
- Lo, W. S., Hong, T. P., & Jeng, R. (2008). A framework of E-SCM multi-agent systems in the fashion industry. *International Journal of Production Economics*, 114(2), 594–614.
- Mahdavi, I., Mohebbi, S., Zandakbari, M., Cho, N., & Mahdavi-Amiri, N. (2009). Agent-based web service for the design of a dynamic coordination mechanism in supply networks. *Journal of Intelligent Manufacturing*, 20(6), 727–749.
- Malucelli, A., Palzer, D., & Oliveira, E. (2006). Ontology-based services to help solving the heterogeneity problem in e-commerce negotiations. *Electronic Commerce Research and Applications*, 5(1), 29–43.
- Navarro, G. (2001). A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1), 31–88.
- Orgun, B., Dras, M., Cassidy, S., & Nayak, A. (2005). DASMAS-dialogue based automation of semantic interoperability in multi-agent systems. In T. Meyer & M. A. Orgun (Eds.) *Proceedings of Australasian ontology workshop*, Sydney, Australia. *CRPIT*, 58 (pp. 75–82). ACS.
- Pallot, M., & Sandoval, V. (1998). *Concurrent enterprising*. Boston, Kluwer: Academic Publishers.
- Protégé, Stanford Center for Biomedical Informatics Research, <http://protege.stanford.edu>.
- Qiu, R. G. (2006). Towards ontology-driven knowledge synthesis for heterogeneous information systems. *Journal of Intelligent Manufacturing*, 17(1), 99–109.
- Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4), 334–350.
- Rau, H., Chen, T. F., & Chen, C. W. (2009). Develop a negotiation framework for automating B2B processes in the RosettaNet environment using fuzzy technology. *Computer and Industrial Engineering*, 56(2), 736–753.
- Sahami, M., & Heilman, T. D. (2006). *A web-based kernel function for measuring the similarity of short text snippets* (pp. 377–386). WWW2006, May 23–26, 2006, Edinburgh, Scotland
- Tamma, V., Phelps, S., Dickinson, I., & Wooldridge, M. (2005). Ontologies for supporting negotiation in e-commerce. *Engineering Applications of Artificial Intelligence*, 18(2), 223–236.
- Trojahn, C., Quaresma, P., & Vieira, R. (2008). *Conjunctive queries for ontology based agent communication in MAS*. AAMAS 2008, May 12–16, Estoril, Portugal.
- Wang, D. S., Nagalingam, S. V., & Lin, G. C. I. (2007). Development of an agent-based Virtual CIM architecture for small to medium manufacturers. *Robotics and Computer-Integrated Manufacturing*, 23(1), 1–16.
- Whitman, L. E., & Panetto, H. (2006). The missing link: Culture and language barriers to interoperability. *Annual Reviews in Control*, 30(2), 233–241.
- Wu, J. N., & Wang, Y. G. (2007). *IAOM: An integrated automatic ontology mapping approach towards knowledge integration*. LSMS 2007, Lecture Notes on Computer Science (LNCS), vol. 4688, pp. 502–509.
- Xu, J. X., & Croft, W. B. (1998). Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems (TOIS)*, 16(1), 61–81.
- Xue, X. L., Li, X. D., Shen, Q. P., & Wang, Y. W. (2005). An agent-based framework for supply chain coordination in construction. *Automation in Construction*, 14(3), 413–430.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338–353.