Semantic Inference on Heterogeneous E-Marketplace Activities

Jingzhi Guo, *Member, IEEE*, Lida Xu, Zhiguo Gong, *Associate Member, IEEE*, Chin-Pang Che, and Sohail S. Chaudhry

***Abstract*—An electronic marketplace (e-marketplace) is a common business information space populated with many entities of different system types. Each of them has its own context of how to process activities. This leads to heterogeneous e-marketplace activities, which are difficult to make interoperable and inferred from one entity to another. This study solves this problem by proposing a concept of separation strategy and implementing it through providing a semantic inference engine with a novel inference algorithm. The solution, called the RuleXPM approach, enables one to semantically infer a next e-marketplace activity across multiple contexts/domains. Experiments show that the cross-context/cross-domain semantic inference is achievable. This paper is an understanding of many aspects related to heterogeneous activity inference.**

***Index Terms*—Electronic marketplace (e-marketplace), heterogeneous systems, inference engine, semantic consistency maintenance, semantic inference.**

I. Introduction

An ELECTRONIC marketplace (e-marketplace) is a common business information space [27], [28], which is the infrastructure of electronic market in the pseudoform of market information systems for buyers and sellers to conduct business through electronic transactions. This is an extension and integration of various enterprise information systems and Internet computing technologies [65]. It has been shown that a business process for an electronic transaction across multiple enterprises comprises a conditional sequence of context-oriented activities in flow [7], [52], [72]. A major device reflecting such activities is the semantic inference, which is the e-marketplace phenomenon of reasoning from one action concept to another subsequent action concept between contextualized enterprises. An action concept represents an activity and could be considered as a verb being annotated by a concept definition, described by some rules, and implemented by an operation.

There is an extensive e-marketplace literature on semantic inference that includes several comprehensive studies [1], [4], [6], [13], [16], [54], [55]. In addition, there is ample research work on the classification of inference engine [34], [39], [49]. However, most of the heterogeneity studies on semantic inference in the e-marketplace have not been matched by success in algorithms. A heterogeneous e-marketplace by nature is an open world [67], where participating enterprise information systems are autonomous, distributed, interdependent, and emergent in participation for carrying out the business [27], [28], [62]. A semantic inference on such e-marketplace can span across multiple contexts/domains, i.e., multiple heterogeneous enterprise information systems. A context is an individual perspective on the meaning of concepts. A well-designed semantic inference algorithm must then maintain semantic consistency between heterogeneous activities of these systems and achieve meaningful correct reasoning across contexts/domains on underlying concepts. Cross-context/cross-domain semantic inference relies on an integrated set of common concepts. Such concepts are regarded as an important source of reaching meaningful understanding between heterogeneous e-marketplace activities in designing a cross-context/cross-domain semantic inference algorithm. Common concepts used in e-marketplace activities must also be independent of particular enterprise information systems and their pertaining inference engines. Only by this independence, the concept meanings of activities could then be correctly reasoned from one enterprise information system to another regardless of their heterogeneity. For example, by collaboratively yet independently referring to a set of common concepts, an offer from a seller can be 100% interpreted by an unknown buyer to infer the correct next acceptance offer that returns to the seller, without possible legal consequences [30], [32].

Nevertheless, among the few systematic methods that have been proposed for semantic inference [1], [2], [6], [23], [38], [42], [47], [58], most are conceptualized in well-specified frameworks where the syntactic forms and semantic concepts for inference systems have been defined or assumed semantically identical, for instance, specifications or system-wide standards in all reasoning phases. Common concepts are tightly coupled with all heterogeneous activities regardless of their context differences. Surely, well-specified or standardized information plays a central role in building the appropriate inference base for activity reasoning. Yet, it has long been recognized that a traditional well-specified approach, which tries to make inference under the closed world assumption, cannot

Manuscript received February 11, 2010; revised July 18, 2010, October 6, 2010, and January 3, 2011; accepted March 3, 2011. Date of publication August 30, 2011; date of current version February 17, 2012. This work was supported in part by the University of Macau under Research Grant RG055/08-09S/GJZ/FST. This paper is a substantially revised and extended version of the paper presented at the 2009 IEEE International Conference on Systems, Man, and Cybernetics. This paper was recommended by Associate Editor T.-M. Choi.

J. Guo, Z. Gong, and C.-P. Che are with the University of Macau, Macau, China (e-mail: jzguo@umac.mo; fstzgg@umac.mo).

L. Xu is with Old Dominion University, Norfolk, VA 23529 USA, with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the Institute of Systems Science and Engineering, Wuhan University of Technology, Wuhan 430070, China (e-mail: LXu@odu.edu).

S. S. Chaudhry is with Villanova University, Villanova, PA 19085 USA (e-mail: sohail.chaudhry@villanova.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2011.2162946

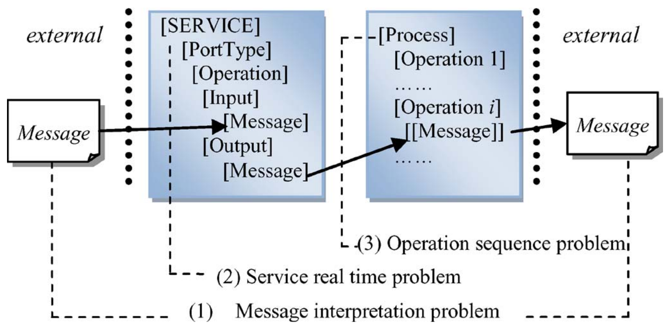


Fig. 1. Problems of existing key method.

cope with e-marketplace activity heterogeneity occurring in contextually different enterprises [30].

For example, in most existing approaches of business activity inference, the key method of reasoning is a combination of standard messaging format (e.g., Simple Object Access Protocol [61]), standard Web service description [17] (e.g., Web Services Description Language (WSDL) [68]), and standard business process execution (e.g., Business Process Execution Language (BPEL) [8], [40]). While we favor service-oriented architecture as a generic business integration approach in a rather stable business environment, this method has several uncertain aspects under the consideration of semantic activity execution, as shown in Fig. 1. First, there is a message interpretation problem. Existing approaches often assume that the message recipient will understand the message in both syntax and semantics. In fact, when messages are sent between e-marketplace players, i.e., users, firms, and e-marketplaces, each player may be unknown with each other and encode a same service using different concepts (e.g., using ecl@ss [20] or United Nations Standard Products and Services Code (UNSPSC) [66]). This implies a need of the commonly interpretable message concepts. Second, service operating on messages has service real-time problem. The intended messages could be received and sent only after the proper service has already been built between sender and recipient based on a service description document (e.g., WSDL). The reality is that an e-marketplace activity as a service can occur at any time among unknown parties. This implies a need of instant service creation and use. Third, there is an operation sequence problem. A business process is a conditional sequence of operations (or activities) and is often a heterogeneous application for different parties. Most existing methods assume a homogeneous business process that is applicable to all business partners (e.g., BPEL).

To alleviate the aforementioned problems, conceptualization technology has been developed to support the message understanding by providing standard vocabularies (e.g., UNSPSC [66] and ecl@ss [20]) and shared ontologies [24]. These two approaches require all messaging parties to adopt either a standard vocabulary or a domain-wide ontology. It is obvious that, when involved parties do not use the same standard vocabulary or span across multiple domains, both approaches have limitation to support cross-context/cross-domain semantic inference.

Semantic Web service (SWS) [5] is a solution to overcoming the first problem. It not only adopts a shared ontology to enable semantic definitions on terms involved in messaging, processes, and service modeling but also provides a mediation mechanism to allow heterogeneous ontology integration and interoperation. However, challenges for SWS still exist because mediation between heterogeneous ontologies is still a not well-solved problem in the ontology matching field [51], [63].

The aforementioned limitation exhibits challenging research issues of inferability [14] and accuracy [35] in cross-context/cross-domain semantic inference. These issues lead to noninferability or ambiguity like wrong inference (when a synonymous symbol has multiple heterogeneous meanings) or missing conclusion (when multiple heterogeneous symbols have a synonymous meaning).

To overcome the noninferable and ambiguous inferences, this paper addresses the issue by treating the semantic inference in a rule-based collaborative concept exchange (CONEX) framework, being presented as a RuleXPM approach that aims to support cross-context/cross-domain business processes, appropriate for e-business between buyers and sellers on e-marketplaces [31]. In this approach, an inference system involves five important aspects: inference logic (what reasoning approach is adopted), inference syntax (how to represent inference content in grammatical specifications), inference semantics (how to represent inference content in independent concept sets), inference rules (how to regulate the contextual inference activities), and inference operations (how to implement the contextual inference activities). Our approach provides a mechanism to enable a business process to be dynamically built across heterogeneous e-marketplace activities, where a concluded activity of one enterprise can be accurately derived from a heterogeneous antecedent activity of another enterprise. It also shows how the integration of heterogeneous e-marketplace activities as a cooperative integrated system provides a more suitable platform for e-business.

The remainder of this paper is organized as follows. Section II briefly reviews the related works of key inference issues and solutions. In Section III, a motivating example is described to explain the technical details that are discussed. An overview of the proposed semantic inference system is presented in Section IV. In Section V, the solution to a semantic inference engine is discussed. Section VI describes a RuleXPM inference algorithm (RIA) required by the inference engine. Section VII describes the experiments on the performance of the systems. Section VIII makes a brief discussion. Finally, in Section IX, a conclusion is made with summary, contribution, and future work.

II. Related Works

Semantic inference has been investigated in many areas, including logical systems [23], [47], [59], databases (DBs) [45], knowledge representation [10], [42], [46], semantic network [10], [38], [46], machine translation [53], semantic Web [1], [58], and e-business [4], [6]. The semantic inference on heterogeneous e-marketplace activities is in the scope of interdisciplinary study of the aforementioned areas, and its key research issues are inferability [14] and accuracy [35] caused by the noninteroperability of those concept sets used in the heterogeneous enterprise systems of the underlying open world. *Inferability* refers to an inferable business process in which an antecedent activity from one context/domain can always derive a concluded activity in another context/domain. For example, when a firm makes a product inquiry as an antecedent activity, it expects that those firms receiving the inquiry can infer an appropriate offer back to the inquiring firm. It is evident that inferability depends on the interoperability between the antecedent content and the concluded content of an inference chain in the form of a business process. Heterogeneous contents are a major obstacle for achieving correct inference between e-marketplace activities. Most approaches tend to place emphasis on concepts that can be shared across heterogeneous systems. Popular technologies include ontology engineering for domain-wide concept sharing [24] such as gene ontology [22] and Dublin Core for documents and publishing [19]; standardization for mandatory standard conformance such as UNSPSC [66] and ecl@ss [20]; and collaborative conceptualization for collaborative agreement on common concepts [32] such as XML Product Map (XPM) [71].

It is evident from the aforementioned approaches that ontology engineering is widely used in such situations. Ontology is either manually designed by ontology engineers or automatically generated by software systems. Ontology design determines that different ontologies created by different ontology engineers or systems are heterogeneous, since different ontology creators often have diverse background knowledge underlying their own contexts. The problem here is that, when a business process consists of a sequence of business activities across multiple enterprises and each enterprise adopts a different ontology, the activity inference from one enterprise to another then cannot be guaranteed correctly. This is because different enterprises may apply different ontologies in activity inference. For example, a seller may present a simple offer ontology instance including concepts of "offer, refrigerator, price, USD" for sending a message, while the buyer may present the same offer ontology instance including concepts of "quote, refrigerator, price, USD" for receiving the message. Ambiguous inference can thus occur on the buyer's side, as it cannot ascertain whether the "offer" semantically equals "quote" and whether "refrigerator" means the same as the seller. A more complex example can be that buyers and sellers adopt a totally different set of ontology models. These examples indicate that any business process across heterogeneous ontologies is generally not inferable or ambiguous. Currently, although many ontology integration approaches have been developed [50], however, the inferability and ambiguity issues have not been fully addressed, because ontology integration mainly focuses on intelligent ontology mapping that targets at higher similarity.

Standardization is another approach to inferability. A standard is a set of commonly accepted terms where these concepts can be shared between various entities. Standard adoption [48] is the precondition of inferability. The main problem is the difficulty of applying one sharable standard to all involved heterogeneous systems of sellers and buyers. This is due to the fact that the sellers and the buyers are often unknown to each other and there are no *a priori* mechanisms that require them to conform to a same standard.

Collaborative conceptualization approach [32] admits the diverse backgrounds of ontology engineers, which affect the interoperability between heterogeneous ontologies. It also recognizes the power of standards. It solves the interoperability problem between heterogeneous concept sets by providing a specified collaboration platform, where concept designers can make agreements on the equivalence of heterogeneous concepts created in different contexts. XPM [71] is a concept design specification and presently used in some research projects and prototypes [25], [41]. It permits heterogeneous enterprise systems to reserve their contextually different local concepts yet also enables them to map onto collaboratively maintained common concepts. This approach provides inferable concept sets between heterogeneous activities. However, its task is only limited to ensure the semantic consistency between heterogeneous activities. The issue of how to infer from one activity to the next activity has not yet developed in previous XPM research.

This paper adopts the collaborative conceptualization approach to maintain semantic consistency between heterogeneous concept sets used in different enterprise information systems on a CONEX network (ConexNet) [29], [32], [33]. Through this approach, it can avoid the problems that ontology engineering and standards cannot solve. Based on this approach, this paper analyzes the support of semantic inference between heterogeneous e-marketplace activities.

It is commonly agreed that inferability directly affects inference accuracy. The *accuracy* in this paper refers to the consistent representation of inference activities and contents and the exact conclusion from the antecedent across heterogeneous systems where inferences are made. The former belongs to the conceptualization technology study discussed earlier, while the latter is the study of inference engine design. In the literature, we have found that heterogeneous systems often have diverse representation approaches (e.g., heterogeneous representations for vocabularies, documents, processes, and rules) to inference activities and contents. When an e-marketplace is designed using diverse representation approaches, inference difficulties are increased, and inference accuracy is reduced. The reflection is that the representation languages cannot properly align the common concepts that are used in activity inference, such that a same concept is represented in different ways in different approaches. To avoid this problem, this paper presents a consistent e-marketplace platform using consistent XPM representation specification [29], [32] and defeasible logic [1], [3] to design concepts, activities, rules, and services.

In general, the design methods of inference engine mainly determine the inference accuracy. For example, the inference engine described in [69] builds the semantic data model on Resource Description Framework (RDF) Schema (RDFS)/Web Ontology Language (OWL), which represents the internal data in Oracle and adopts a forward-chaining inference strategy, while supporting the inference based on standard constructs and user-defined rules. Similarly, Jena [11] includes a rule-based inference engine built for RDF, RDFS, and OWL using description logic for both forward-chaining reasoning and backward-chaining reasoning. Euler [56] as another inference engine supports logic-based proofs in several types of representations, including RDFS and OWL. It is a backward-chaining reasoner that is enhanced with Euler path detection. The proof engine follows the Euler path to avoid endless deductions. All these methods have their merits in achieving accuracy within a shared domain. However, noninferability across heterogeneous domains/contexts is the common limitation for these inference engines. This becomes the motivation of this study.

Different design methods impact on the performance of an inference engine, although inferability and accuracy are the first priorities. Historically, two types of algorithms have been

FIRM A's Business Process:
Make inquiry(inquiry sheet template; <inquiry content>) → inquire(inquiry sheet) → [Messaging: Send(document) → Receive(document)] → match(inquiry sheet) → quote(quotation sheet) → [Messaging: Send(document)].
FIRM B' Business Process:
[Messaging: Receive(document) → match(inquiry sheet) → make offer sheet (offer sheet template；<offer content>) → offer (offer sheet), if the incoming inquiry sheet is satisfied.
Otherwise:
make inquiry rejection sheet (inquiry rejection sheet template; <inquiry rejection content>) → reject inquiry(inquiry rejection sheet), if the incoming inquiry sheet is not satisfied.

Fig. 2. Motivational example.

used for inference engines regarding the inference efficiency. They are the filter algorithm and the Rete algorithm [21], [73]. Nevertheless, the Rete algorithm has gained more popularity (e.g., SRI's new automated reasoning kit [37], TREAT [44], and official production system [9]) and has become the basis for many popular expert system shells such as C Language Integrated Production System [15], Jess [36], Drools [18], BizTalk, Rules Engine, and Soar [60]. The Rete algorithm supports forward chaining and provides a generalized logical description of an implementation of functionality responsible for matching facts against rules in a pattern-matching rule system. The use of a Rete network is also supposed to be much faster than the filtering technique. In this paper, we employ the forward-chaining method because a next activity as a conclusion in an inference rule is uncertain in heterogeneous enterprises.

III. Motivating Example

To illustrate the problem that we are going to solve, i.e., the inferability and accuracy of semantic inference across heterogeneous e-marketplace activities, we present a motivational example. In this example, we suppose that there are two enterprise information systems FIRM A and FIRM B, which are unknown with each other and semantically heterogeneous in their business processes as shown in Fig. 2.

In Fig. 2, in order to explicitly exemplify the semantic heterogeneity between FIRM A and FIRM B, we assume that FIRM A and FIRM B use English and Chinese, respectively. In the following sections, the semantic inference problem on heterogeneous systems is solved along with the illustration of this example.

IV. Overview of System Architecture

In most cases, sellers/buyers generate a rich variety of inferences using their self-constructed context models for the received information when comprehending a received business document (BD). For example, FIRM B may generate a semantic inference solution different from FIRM A when it perceives FIRM A's activity, shown in Fig. 2, based on its own knowledge. It is undisputed that both business knowledge and the information context facilitate the final interpretation of the received document by recipients to take subsequent actions. The recipients use their knowledge to capture the dependences between concepts and provide coherence to the representation of information. Context ties sellers/buyers to their known and

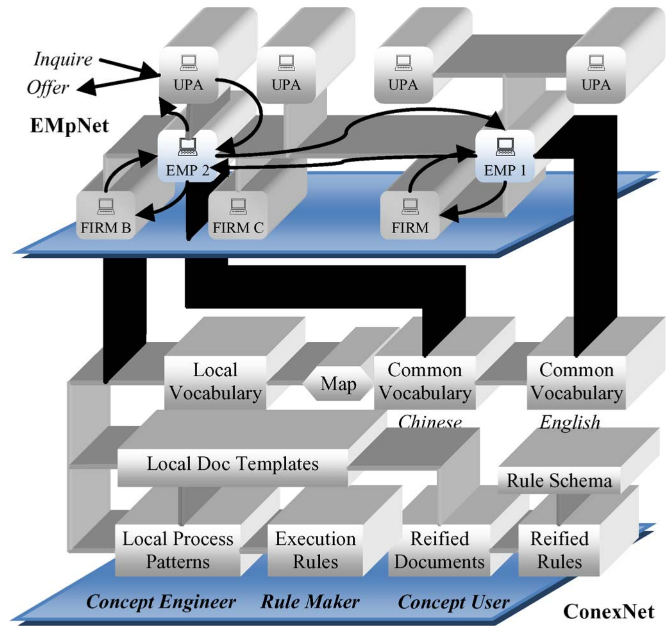


Fig. 3. System model.

familiar situations and rejects the information not belonging to them. A resolution of the received information, achieved primarily by relevant linking to the background knowledge of buyers and sellers, has to be made in order to create the information interpretation for subsequent activity inference. This seems, however, to be significantly difficult to achieve under the constraints imposed by the noninteroperable knowledge sources between buyers and sellers and the inconsistent information syntax and semantics that represent any concepts in exchange. Next, we explain how our approach can make use of the collaborative conceptualization theory [29], [32] to facilitate semantic inference.

A. Outline of the RuleXPM Approach

The RuleXPM approach is an integrated model that combines a set of representations of various types of concepts, some e-marketplace participating systems, and an inference process. The method consists of several major constituents that include a collaborative ConexNet, an e-marketplace network (EMpNet), and an inference engine. In this research, we concentrate on the development of EMpNet as well as the inference engine.

A schematic representation of the model is shown in Fig. 3. In this representation, ConexNet (as a semantic network) is first collaboratively formulated by the concept engineers, rule makers, and concept users who work together to create all types of semantic consistent concepts (abstract and reified) between contexts and domains. Each collaborative concept has a unique $iid \in$ IID. There are four resultant concept types: *common concepts* [of common vocabularies (CVs)] in different natural languages for different e-marketplace systems (EMP) (e.g., EMP 1 and EMP 2 shown in Fig. 3), *local concepts* [of local vocabularies (LVs), local BD templates, and local business process patterns (BPPs)] used in various enterprise systems (FIRM) (e.g., FIRM A and FIRM B shown in Fig. 3), *map concepts* that map local concepts onto common concepts, and *reified concepts* of abstract types as reified documents (each is a set of particular concepts) and reified rules for dynamically

TABLE I
SEVEN BASIC RELATIONS BETWEEN SIGNS

Name	Notation	Definition
atomicity	$\otimes$	For any sign A, A is *atomic* iff $\otimes$A.
independence	$\parallel$	For any two signs A and B, A is *independent of* B iff A $\parallel$ B.
interpretation	<<	For any sign A and interpretant $\Diamond$, A is *interpreted by* $\Diamond$ iff A << $\Diamond$.
conceptualization	$\Rightarrow$	For any two signs A and B, A *conceptualizes* B iff A$\Rightarrow$B.
contextualization	@	For any two signs A and B, A *is contextualized by* B iff A@B.
equivalence	$\equiv$	For any two signs A and B, A is *concept-equivalent with* B iff A $\equiv$ B.
reification	$\rightarrow$	For any two signs A & B, A *reifies* B iff A $\rightarrow$ B.

TABLE II
TWO COMPLEX RELATIONS IN $\sum$ Sign

Name	Notation	Definition
concealment	$\langle\,\rangle$	For any two signs A and B, A is *concealed from* B iff B$\langle$A$\rangle$, such that: (1) A is set of signs, and B is any sign such that B $\not\subset$ A; (2) For any $a \in$ A, a is an AISign; (3) P(A) $\parallel$ B, where P(A) is a power set of A.
interface	$\models$	For any three signs A, B and C, A is *interfaced by* B to C iff (1) A $\equiv$ B; (2) [B $\models$ C and] B $\equiv$ C.

defining and controlling things. ConexNet guarantees that all concepts created, communicated, and used in this network are accurate and semantically consistent without ambiguity on the CONEX chain of "reified concept riid $\Leftrightarrow$ local concept liid $\Leftrightarrow$ mapping concept (liid, ciid) $\Leftrightarrow$ common concept ciid $\Leftrightarrow$ mapping concept (ciid, liid) $\Leftrightarrow$ local concept liid $\Leftrightarrow$ reified concept riid." It contributes to a trichotomic view of design, implementation, and use of heterogeneous concepts for semantic consistency maintenance. With this view, *concept engineers* are responsible for collaborative concept design for common concepts and local concepts in a collaborative-concept-editing system (*see* demo [70]); *rule makers* implement all executable concepts as ruled concepts or control rules for verbs and adjectives in both common and local levels, and *concept users* automatically reify these concepts and simply use them (*see* demo [70]). ConexNet and its importance have already been described in the projects of CONEX [29], collaborative document exchange [26], and collaborative process exchange [30] and will not be elaborated in this paper. In the following section, we only briefly introduce the concept representation behind ConexNet with the illustration in Fig. 2.

B. Concept Representation in ConexNet

ConexNet [33] thinks that any terms, phrases, sentences, and even a document are a sign, which is a conveyor of a concept. A sign is a tuple of structure (S), concept (C), context (X), and interpretant ($\Diamond$) such that $\sum \mathrm{Sign} = (\mathrm{S}, \mathrm{C}, \mathrm{X}, \Diamond)$, forming ConexNet vocabularies. While any signs are *atomic and independent* (i.e., AISigns), they can be uniquely identified in any space and time and used to construct vocabularies, documents, and processes. Based on seven basic and two complex relations between signs, shown in Tables I and II, any subset of a *composite sign* CSign (i.e., a list of AISigns) and any subset of a *document sign* DSign (i.e., a tree of AISigns) can be independently represented, exchanged, and interpreted without meaning ambiguity between sign originator and sign user.

Formally, any sign A is *atomic* if and only if (1) $\mathrm{S(A)} = \mathrm{S(IID, T, AN, X, \{OP\})}$ and (2) $\mathrm{C(A)} = \mathrm{C(IID} \equiv \mathrm{T} \Leftarrow \mathrm{AN}\ @\ \mathrm{X)} \ll \Diamond$, where the structure S of sign A is composed of a set of elementary structures $\sum \mathrm{S}_i =$ (unique concept identifier IID, term T, concept definition AN, context X, optional extensions {OP}). The concept C of sign A is interpreted by an interpretant $\Diamond$ as an annotation AN at a context of X. The AN is again conceptualized as a term T that is equivalent to unique identifier IID. The uniqueness of IID cross-domains is guaranteed by an MD-IID scheme [33]. Aside from atomicity, any two signs A and B are *independent* if and only if A and B are uniquely identified as $\mathrm{MD\text{-}IID_A}$, $\mathrm{MD\text{-}IID_B}$, and $\mathrm{MD\text{-}IID_A} \neq \mathrm{MD\text{-}IID_B}$. Sign A is an *atomic and independent sign* (i.e., AISign) if and only if it is both atomic and independent. An AISign can be referenced by any other signs without any versioning problem, because anything happened has become a history and will be never changed.

Applying the relations of concealment and interface, we can achieve independence of any composite sign. Formally, for any sign A, A is said to be a *composite sign* (i.e., *CSign*) if and only if A can be expanded to a list of AISigns such that $\mathrm{A} \equiv (\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_n)$, where A is said to be interfaced by $(\mathrm{A}_1, \mathrm{A}_2, \ldots, \mathrm{A}_n)$ to B if and only if $\mathrm{B} \equiv (\mathrm{B}_1, \mathrm{B}_2, \ldots, \mathrm{B}_n)$ and $\mathrm{A}_1| = \mathrm{B}_1, \mathrm{A}_2| = \mathrm{B}_2, \ldots, \mathrm{A}_n| = \mathrm{B}_n$, where $\mathrm{A}_1 \equiv \mathrm{B}_1, \mathrm{A}_2 \equiv \mathrm{B}_2, \ldots, \mathrm{A}_n \equiv \mathrm{B}_n$. Furthermore, a sign D is a *document sign* (DSign or D) if and only if $\mathrm{D} = (\mathrm{D}_1^1, \mathrm{D}_i^2, \ldots, \mathrm{D}_i^k, \ldots, \mathrm{D}_i^n)$, such that D consists of a set of hierarchical AISigns $\mathrm{D}_1^1, \mathrm{D}_i^2, \ldots, \mathrm{D}_i^k, \ldots, \mathrm{D}_i^n$, where D_1^1 is the tree root, k is the tree level, and i is the sibling position. We can prove that any subdocument D' of D is a concealment of $\langle \mathrm{D}_1^1 \langle \mathrm{D}_i^2 \langle \ldots \langle \mathrm{D}_i^k \langle \ldots \langle \mathrm{D}_i^{m+1} \rangle\rangle\rangle\rangle\rangle\rangle$, and any subdocument D' of D is interfaced to A by RT such that $\mathrm{D}' \equiv \mathrm{RT}| = \mathrm{A}$, where $\mathrm{A} = (\mathrm{A}_1, \ldots, \mathrm{A}_n) \not\subset \mathrm{D}$ and $\mathrm{RT} = (\mathrm{RT}_1, \mathrm{RT}_2, \ldots, \mathrm{RT}_n) \subset \mathrm{D}'$.

For example, applying the ConexNet concept representation, we can collaboratively create terms/concepts of vocabularies for creating business processes of FIRM A and FIRM B (shown in Fig. 2) as shown in Fig. 4.

ConexNet concepts are collaboratively created following the collaborative conceptualization approach [32]. They are particularly developed to adapt to the cross-context/cross-domain vocabularies for creating/using heterogeneous documents, processes, and rules. They permit that any locally created composite and document concepts are semantically interoperable between contexts and domains when applying concealment and interface relations.

C. EMpNet

EMpNet, as shown in Fig. 3, is an activity network that depicts how the connected activities perform. The network, denoted as "$\sum \mathcal{S}$," is composed of a set of participating systems, to which each activity belongs. A particular participating system is a pair of node and line in a hierarchical graph such that

$$EMpNet = \sum \mathcal{S} \tag{1}$$

$$\mathcal{S} = (node, line) = (\mathcal{N}, \mathcal{L}) \tag{2}$$

(1) Build terms/concepts as AISigns in common vocabularies (CVs) by P2P collaboration.

- CV (English for EMP 1) = {0010≡(make, v), 0011≡ (inquire, v), 0012≡(match, v), 0013≡(offer, v), 0014≡(reject, v), 1300≡(is, v), 0020≡(inquiry, n), 0021≡(offer, n), 0022≡(sheet, n), 0023≡ (rejection, n), 0024≡(refrigerator, n), 2100(size, n), 2700≡(price, n), 4100≡ (property, n), 4200≡(age, n), 0025≡(orange, adj), 1600≡(cheapest, adj), 4400≡(largest, adj), 5100≡(less and equal, f), (0020 0022)≡ (inquiry sheet, f), (0021 0022)≡(offer sheet, f), (0020 0023 0022)≡ (inquiry rejection sheet, f)}, where "v" for verb, "n" for noun, "adj" for adjective and "f" for phrase. For simplicity, annotation (AN) is omitted.

- CV (Chinese for EMP 2) = {0010≡(zhizuo, v), 0011≡(xunjia, v), 0012≡(pipei, v), 0013≡(baojia, v), 0014≡(jujue, v), 0020≡(xunjia, n), 0021≡(baojia, n), 0022≡(dan, n), 0023≡(jujue, n), 0024≡(bingxiang, n), 0025≡(chengsede, adj), (0020 0022)≡(xunjiadan, f), (0021 0022)≡(baojiadan, f), (0020 0023 0022)≡(xunjiajujuedan, f)}.

(2) Build local vocabularies (LVs) by collaborative localization. By default, an LV mapped onto a CV is exactly the same as the part of the CV in terms and IIDs. It is changed when FIRM wants to use synonymous terms and IDs, for example:

- LV (FIRM A in EMP 1) = {…(default mapping)…, a550≡ (quote, v), a551≡(quotation, n), a552≡(fridge, n), a770≡ (quotation sheet, n)} through *map*(CV, LV A) = {…, (a550, 0013), (a551, 0021), (a552, 0024), (a770, (0021 0022))}.

- LV (FIRM B in EMP 2) = {…(default mapping)…, b310≡ (danju, n), b311≡(bingxiang, n), b312≡(chengsede, adj)} through *map*(CV, LV B) = {…, (b310, 0022), (b311, 0024), (b312, 0025)}.

Using mapped concepts of LVs, FIRM A and B can create documents in their own semantic context models.

(3) Build business documents (BD) as DSigns through concealment and interface, for example:

concept(iid = "xyz" t = "inquiry sheet" rt = "0020 0022"){
concept(iid = "xyz.1" t = "fridge" rt = "a552"){
concept(iid = "xyz.1.1" t = "orange" rt = "0025")}}

where iid ≡ t |= rt, such that iid uniquely identifies t and semantically references to rt, which is a local/common concept in LV/CV.

In this BD, the meaning of any concept is concealed, comprising the lower level concepts. Users can access to the definite meaning of the concept in hierarchy through interface RT and MAP(local IID, common IID). For example, remote users can semantically interpret "xyz" iid-ed concept through interface "rt" = {(0020 0022), a552, 0025} and MAP(a552, 0024). The concealment and interface relations make it possible for any subdocument of a BD to be reused as an independent concept.

Fig. 4. Examples of CV, LV, and BD.

TABLE III
EMpNet Configuration Method

Node ($\mathcal{N}$)	Node Type	Line Type ($\mathcal{L}$)
UPA	*Consumer Player*	*ConexNet networking*
FIRM	*Business Player*	*ConexNet networking*
EMP	*Mediation Provider*	*ConexNet networking*

where both node $\mathcal{N}$ and line $\mathcal{L}$ are typed to differentiate their functional behaviors, shown in Table III.

Table III shows the elements of how to construct EMpNet by configuring its node $\mathcal{N}$ and line types $\mathcal{L}$. The node type determines the particular participating systems that a node works. The line type determines which ConexNet that a particular node adopts for communication with other nodes. By default, EMpNet applies ConexNet as the networking method and concept semantic consistency maintenance method. This is because ConexNet provides an internal semantic communication standard developed by XPM [29], [32], [71].

EMpNet provides a topology of how participating systems of user program agents (UPAs), enterprise systems (FIRM), and e-marketplace systems (EMP) should be distributed and connected to perform as well as how each of them should take responsibility for e-marketplace activities. Each participating system has mappings onto ConexNet to edit and use the concepts in a semantic consistent way. Based on these concepts, further semantic inferences could be made between the participating systems through a proposed inference engine. The results of these inferences are the dynamically created business processes across heterogeneous participating systems.

D. Concept Separation Strategy

Reasoning on EMpNet (i.e., "→" in Fig. 3) plays a central role in e-marketplace activity inference. It is obvious that business processes of different UPAs, FIRMs, and EMPs are heterogeneous. Also, it is difficult to determine which activity should be followed one after another when activities are generated across different systems. Our approach is to employ a *concept separation strategy* to enable the semantic inference on heterogeneous e-marketplace activities. Applying this strategy, EMpNet is built for reasoning, which is able to interpret heterogeneous activities, where an activity is an action concept in the form of a linguistic verb represented as a triplet

$$Activity = (denotation, connotation, implementation) \tag{3}$$

$$\mathcal{A} = (\mathcal{D}, \mathcal{C}, \mathcal{I}) \tag{4}$$

$$\mathcal{A} = \mathcal{I}(\mathcal{D}, \mathcal{C}) \tag{5}$$

where the denotation $\mathcal{D}$ is a set of interfaced concepts in ConexNet, describing the interoperable meaning of the action, the connotation $\mathcal{C}$ is a set of concealed concepts defining the detailed document content and structure, and the implementation $\mathcal{I}$ is an executable rule document defining how the action should be executed. For example, an "inquire(inquiry sheet)" activity sent from FIRM A to FIRM B (in Fig. 2) can be represented as Activity(0011([0020 0022]("document body"))) by using the LV in Fig. 4(2) and the BD in Fig. 4(3). In this activity, the denotation is all RTs (interfaced IIDs) such as "0011" and "0020 0022." The connotation is all concealed IIDs (e.g., xyz, xyz.1, xyz.1.1) describing the document content and structure. The implementation is the execution rule document for processing "(0020 0022)." This document is implicit and unnecessary to be defined because, when FIRM B receives the activity, it will automatically search the document that matches the "0011" activity.

Following the aforementioned strategy and the activity definition, EMpNet can be developed under the following principles.

1) Any concept denotation $d \in \mathcal{D}$ is semantically equivalent between participating systems of EMpNet and implies that activities are universal. The globalization of denotation is achieved by collaborative concept design in ConexNet. For example, all concepts of CVs and LVs in Fig. 4 are all in $\mathcal{D}$. In this example, "refrigerator" and "orange" may mean differently in various LVs; ConexNet collaboratively ensures their semantic consistency across LVs.

"*Execute(0011: 詢價)* [IF *0011:(0020 0022 : 詢價單)* THEN *0012:匹配 (0020 0022);* IF *0012(0020 0022)* THEN *0010:製作(0021 0022 : 報價單) ELSE 0010(0020 0023 0022 : 詢價拒絕單)*]"

Fig. 5. Examples of an execution rule document of FIRM B.

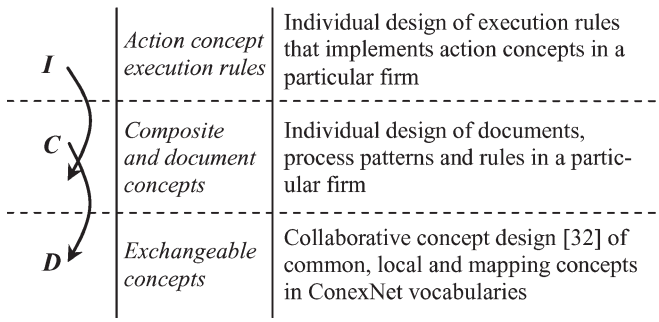


Fig. 6. Implementation of the concept separation strategy.

2) Any concept connotation $c \in \mathcal{C}$ is local to a particular EMpNet participating system. It refers to that composing a composite concept or document concept using independent and atomic concepts (i.e., AISigns) is a local matter, such as composing a document template, its reifications, and any BPP relevant to a participating system. Connotation is contextual and individual. The localization of connotation is achieved by the separation of local individual work from collaborative work. For example, the BD in Fig. 4(3) is in $\mathcal{C}$. Its composition is local by using the concepts in LV for FIRM A, which are denotations.
3) Any concept implementation $i \in \mathcal{I}$ is local to a particular EMpNet participating system. It refers to any execution rule document of an action concept. The executability of an action concept is achieved from the separation of the denotation of an action concept from its implementation as a local individual work. The illustration can be seen in Fig. 5.

Fig. 5 shows an execution rule document, which implements the action concept 0011. It is local to FIRM B that is described based on the business processes in Fig. 2 and the vocabularies in Fig. 4.

With these principles, EMpNet is able to reason between heterogeneous e-marketplace activities such that any activity is universally understandable by all participating systems yet personalized and controlled by local systems to determine the triggering, execution, and termination of any activity.

The concept separation strategy is implemented in each node as a triplet (D, C, I), as shown in Fig. 6, with the following divisions.

1) *Denotation division* D. Any concept denotation $d \in D$ of an activity is contextually understandable by all EMpNet participating systems, such that d is semantically exchangeable if and only if $d \in \mathcal{D}$ is also in ConexNet. In general, the denotation D is a set of vocabularies containing local concepts in LVs, common concepts in CVs, and mapping concepts in collaboratively mapped concepts, as defined in ConexNet. The D division guarantees the concept universality for semantic interoperability.
2) *Connotation division* C. Any concept connotation $c \in C$ of an activity is contextually understandable by all EMpNet participating systems, such that c is locally executable if and only if $c = \mathcal{P}(\mathcal{D}) \in \mathcal{N}$ and $\mathcal{D} \in$ ConexNet, where $\mathcal{P}(\mathcal{D})$ is a power set of $\mathcal{D}$ and is not empty. In general, the connotation C is a set of concepts in the forms of documents and process patterns made from LVs/CVs of ConexNet. A connotative composite concept can be contextually understandable between contexts because its name is a concept associated with an interoperable unique IID and its content is composed of denotative concepts that are semantically equivalent and contextually exchangeable in ConexNet. The C division guarantees the localizability of concept composition such that any composite concepts for an activity are locally composable and yet universally understandable without requiring the knowledge of external messaging parties.
3) *Implementation division* I. The concept denotation $d \in \mathcal{D}$ of any activity $a \in \mathcal{A}$ is implementable if and only if there exists only one denotation implementation $i \in I$, such that $i \in \mathcal{I}$ is locally executable on the connotation $c \in (a \cap \mathcal{C}) \in \mathcal{N}$. In general, the implementation division I is implemented in a set of rule documents, with each corresponding to an action concept. This guarantees that any action concept is not only universally exchangeable but also locally executable without the need of global knowledge, which often causes semantic conflicts.

This strategy implementation, as a whole, further guarantees that different nodes $\mathcal{N}$ can keep their heterogeneous BDs, processes, and rules as local artifacts but still ensure the semantically consistent activity interoperation without the need of sharing any standard BPPs, document templates, and executable programs. Comparing with the ontology engineering strategy, it enables semantic inference between heterogeneous domains and contexts as well as permitting participating systems (e.g., FIRM A and FIRM B) to achieve customization as needed. Noticeably, this is not available in most domain-wide-ontology-based approaches.

For example, the concept separation strategy can well be implemented for FIRM A and FIRM B if information of D, C, and I is given as follows:

1) sets of common atomic concepts of CVs, LVs, and MAPs [shown in Fig. 4(1) and (2)] $\subset D$;
2) sets of document templates and reifications (such as the example shown in Fig. 4(3) for FIRM A) $\subset C$;
3) sets of execution rules (such as the example shown in Fig. 5 in Section IV-D) $\subset I$.

Given the aforementioned information provided by ConexNet, when an activity "0011([0020 0022](a552(0025))))" is sent from FIRM A to FIRM B, FIRM B will first obtain "0011([0020 0022](b311(0025))))" through concept mapping. Then, it will apply the rules in Fig. 5 to determine the next series of activities of whether FIRM B should make an offer sheet or make an inquiry rejection sheet based on the match result.

V. Semantic Inference Engine

The purpose of the concept separation strategy is to support the design of a generic semantic inference engine that can work

[*ConexDocType*] **EmpnetEngine** (*ConexDocType*) (3.1)

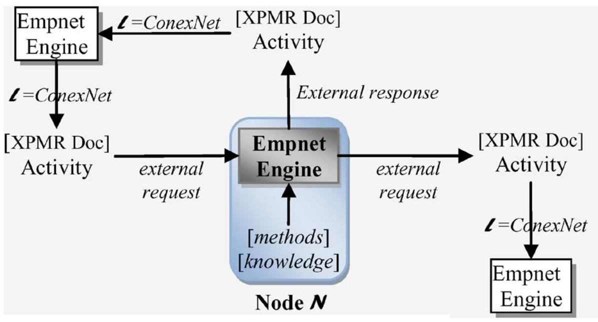


Fig. 7. Activities for the semantic inference engine.

in EMpNet for heterogeneous e-marketplace activity inference. A *semantic inference engine* can be regarded as a widget generically working on a participating system of EMpNet. It receives the incoming information of an activity, processes it, and returns the outgoing information of a next activity. In particular, the engine, called as "EmpnetEngine" and shown in Fig. 7, is a document-typed and service-oriented program defined in

$$[ConexDocType]\ \mathbf{EmpnetEngine}(ConexDocType). \quad (6)$$

The engine applies the forward-chaining inference method, which is frequently used in many systems [9], [21], [37], [44], [73] to make inference. This application is adequate because, for any activity rule of "IF A THEN B," the consequence is uncertain and can only be determined when antecedent A is true. For example, given an "inquire(inquiry sheet)" activity sent to three firms, we then have an activity rule "IF inquire(inquiry sheet) THEN X" as the incoming activity rule and three inquiry handling rules in the three firms as "IF inquire(inquiry sheet) THEN offer(offer sheet)," "IF inquire(inquiry sheet) THEN analyze(inquiry sheet)," and "IF inquire(inquiry sheet) THEN notAct(log sheet)," respectively. It is noticeable that the consequence of the incoming rule can only be known after the execution of the antecedent of the incoming rule by searching and comparing the local activity rule document of each firm. This inference model using the forward-chaining method can be described as follows:

$Concept1 \in Incoming\ ConexDoc$
$Concept2 \in Knowledge\ base$
IF $concept1\ [fact \mid rule]$ THEN
$Match(concept1, concept2)$
IF *Match* THEN $Action\ (concept2$ [fact | rule]$)$
ELSE *NonMatchAction*
Repeat.

In this model, the execution of concept match in each recipient system is strictly sequential starting from the concept that is going to match. Recursive rule sets are not permitted; otherwise, forward-chaining-based methods may fail. During the execution of concept match, if any concept mismatch is found, the engine signifies a nonmatch, which immediately triggers a nonmatch action. This model can increase the efficiency and is also the requirements for both semantic consistency maintenance and business process, particularly in e-business.

It should be evident that, when EmpnetEngine works on ConexNet, all its inputs and outputs can adopt the messaging form in the *reified XPM (XPMR) document* format [71], which is universal in both ConexNet and EMpNet. In this engine, the inputs come from both external EmpnetEngine and the local system. While the external input is an XPMR (as a set of atomic concepts in RT belonging to the denotation $\mathcal{D}$), the local input comes from the local knowledge base (which belongs to the connotation $\mathcal{C}$) either in XPM-based documents (reified rules, documents, and process patterns) or in relational DBs. Both types of inputs are handled by the execution rules, which belong to local implementation $\mathcal{I}$. The output of EmpnetEngine is also an XPMR and thus becomes the input of another EmpnetEngine. The strict separation of document use scopes from execution rule application domains can perfectly enable the heterogeneous e-marketplace activity inference.

It should be noticed that our approach also adopts a format hiding technique, i.e., all EMpNet users do not need to know any XPM document formats. Users simply use the XPM editors to create and use the publicly exchangeable vocabularies, documents, locally usable rules, and BPPs. The work of implementing various XPM editors is the research currently being conducted in our research group (*see* demo [70]). The XPM document visualization is a strategy to ease work for e-commerce practice and also a way of avoiding document format interoperability problem.

A. RuleXML-Based Processing for Incoming Activity

To guarantee that the incoming messages for the current activity and the outgoing messages for the next activity are interoperable in both syntax and semantics, an EMpNet-wide messaging standard must be established. In EMpNet construction, an XPM schema [71] is adopted. Since both incoming and outgoing messages of an EmpnetEngine are assumed as XPMRs, there are two alternatives of an *explicit XPM rule method* or an *implicit XPM rule method* for XPMR processing. The former refers to a method that converts any incoming XPMR into an explicit reified rule document stipulated by an XPM rule (RuleXPM) schema. The latter is a method that directly interprets any incoming XPMR as a set of implicit rules so that there is no need for additional explicit rules for XPMR processing. Both methods have their advantages and disadvantages. The explicit rule method is clear in reasoning, because any involved document is a set of rules, and easier for processing. The drawback is that it is indirect and requires a rule conversion. The implicit rule method is concise and direct because it omits the rule conversion. The drawback is its higher complexity of the design of the inference engine and the implementation of rules. Here, we adopt the explicit rule method such that any incoming XPMR must be converted into a RuleXPM document as input of EmpnetEngine for processing to finally derive another XPMR for output. Formally, we have

$$\begin{aligned}&\mathbf{External}: XPMR \rightarrow RuleXPM\\&\mathbf{Local}: \qquad RuleXPM\\&\qquad \rightarrow Match \rightarrow Action \rightarrow XPMR \quad (7)\end{aligned}$$

where any incoming XPMR document is converted into a RuleXPM document, which, together with local RuleXPM documents, executes query rules on local knowledge base and acts on execution rules to derive another XPMR document for the next activity. In the following, we briefly introduce the schemas of XPM template (XPMT)/XPMR and RuleXPM and their conversion rules, which govern how an incoming XPMR document is formed and converted into a RuleXPM document.

XPM Document Schema: Both XPM document template (called XPMT template) and its reification (called XPMR document) are governed by an XPMT/XPMR schema, which defines BDs as a set of abstract concepts or reified concepts as follows:

$$\mathbf{XPMT} := \text{concept}[iid, \text{cls}, \text{sel}, \text{op}, \text{gt}, \text{rt}, \text{fc}] \tag{8}$$

$$\mathbf{XPMR} := \text{concept}[iid, \text{cls}, \text{sel}, \text{op}, \text{gt}, \text{rt}, \text{fc}] \rightarrow \{value\}. \tag{9}$$

In this schema, any abstract concept comes from the collaboratively designed vocabularies of ConexNet [29], [32] and is recursive as concept[...](concept[...], ..., concept[...]) to derive a concept hierarchy and form a leveled connotation structure following DSign introduced in Section IV-B. It represents a semantic object that could be reified as any $\{value\}$. It denotes itself with a denotation structure [...] made by a set of elementary structures such as the following:

1) *iid*: unique concept identifier;
2) *cls*: classifier in a hierarchical placeholder;
3) *sel*: selection type of "choice," "sequence," and "preference";
4) *op*: numeric value representing the priority of the preference relation when sel = "preference" or an operator for $\{value\}$ if otherwise;
5) *gt*: concept grammar;
6) *rt*: referenced concept IID to an exchangeable concept;
7) *fc*: human-readable concept.

The "concept" has several variant notations for describing an XPMT/XPMR document structure such as ⟨word⟩, ⟨phrase⟩, ⟨sentence⟩, ⟨paragraph⟩, ⟨section⟩, ⟨table⟩, etc. The detailed specification of XPM/XPMR can be found in [71].

RuleXPM Schema: Any XPMR document can be written or converted into a RuleXPM document, governed by a set of conversion rules and/or a RuleXPM schema. The purpose of this schema is to build a logically inferable document, enabling to reason the result from XPMR documents. In our RuleXPM approach, the logic applied for rule design and conversion is the defeasible logic [1], [3], which handles both strict and defeasible rules, particularly the priority (i.e., a preference relation of XPMR). The structure of the RuleXPM schema is as follows:

$$\mathbf{rulexpm}(\text{rule}, \text{pref}, \text{reg}) \tag{10}$$

$$\mathbf{rule} := \text{rule}[\text{rid}, \text{sel}]\ (\text{concept}[\text{rid}, \text{rt}, \text{op}](\text{concept}^*)) \tag{11}$$

$$\mathbf{pref} := \text{preference}[\text{rid}, \text{op}]\ (\text{concept}[\text{rid}, \text{rt}, \text{op}](\text{concept}^*)) \tag{12}$$

$$\mathbf{reg} := \text{regulation}[\text{rid}]\ (\text{concept}[\text{rid}, \text{rt}, \text{at}](\text{concept}^*))\ . \tag{13}$$

In this schema, the "rule" is a strict rule identified by "rid" where "sel = (sequence|choice)" means "for all" or "there exist" and "op" refers to operator. "Preference" refers to a defeasible rule for priority relation identified by "rid" where "op" is to determine priority rating. Both "rule" and "preference" are XPMR concept representation rules. Within "rule" and "preferences," they are facts directly converted from the XPMR document. Differently, "regulation" defines strict concept execution rules. It is often used to determine whether the incoming rules should be associated with the knowledge base to execute queries and how a result should be generated to lead to a next XPMR document.

TABLE IV
XPMR-TO-RULEXPM TRANSLATION RULES

R1: concept[IID] ⇒ rule[RID](concept*) // *Abstract concept with IID to be a general strict rule.*
R2: concept[IID]→{value} ⇒ rule[RID]((concept→value)*) // *Reified concept with IID to be a particular strict rule.*
R3: concept[IID, sel]→{value} ⇒ rule[RID, sel]((concept→value)*) // *Sequence or choice reified concept when sel = "sequence" or "choice" to be a particular strict rule.*
R4: concept[IID, sel="preference", op]→{value} ⇒ pref[RID, op]((concept→value)*) // *Preferred reified concept when sel = "preference" to be defeasible rule for priority relation.*
R5: concept[IID_m, sel="preference", op_m], concept[IID_n, sel="preference", op_n], $op_m < op_n$ ⇒ preference[RID_m, op_m] > preference[RID_n, op_n] // *Concept with less op value has higher priority in superiority relation.*

XPMR to RuleXPM Translation Rules: The conversion from XPMR documents to RuleXPM documents needs additional rules to translate XPMR concepts to RuleXPM rules. The translation is governed by a set of rules, shown in Table IV, which is used to implement the conversion.

RuleXPM is designed to cope with heterogeneous environments. It simplifies the use of defeasible logic [3], Semantic Web Rule Language (SWRL) [64], and ConexNet concepts [71] to represent facts and strict and defeasible rules. For each rule, preference, and regulation, it derives a conclusion that is proved either true or false. Generally, the "rule" is strictly inferable while "preference" is defeasibly inferable, as follows:

$$rule ::= \text{“}Implies(\text{‘}[RID]antecedent\ consequent\text{’})\text{”}$$

$$preference ::= \text{“}Implies(\text{‘}[\text{RID}]antecedent\ consequent\text{’})\text{”}$$

$$antecedent ::= \text{“}Antecedent(\text{‘}IID - ed\ concepts\text{’})\text{"}$$

$$consequent ::= \text{“}Consequent(\text{‘}IID - ed\ concepts\text{’})\text{”}$$

The consequent or conclusion is proved if and only if Consequent("IID-ed concepts") is superior to or equal to Antecedent("IID-ed concepts"), notated as $C \geq A$. The proved result is used to determine the next rule that will be selected.

IF $C \geq A$ THEN C ELSE *other rule*.

The difference between rule and preference is the processing principles, such that the proved consequent of a rule is always included for use, i.e., Positiveness As Success (PAS), while the proved consequent of a preference is always excluded for use, i.e., Negation As Failure (NAF). The more preferable preference can defeat the less preferable preference.

For example, in the following rules applying the LV/CV in Fig. 4, "sequence" (i.e., "for all" ∀) indicates the processing of all facts (here are ⟨word⟩'s), rt-ed with "4100," "1300," and

"5100." If the query results of rt "4100," "1300," and "5100" exist and "5100" ≤ 20 exists, then it returns a query result as true; else, it returns a query result as false.

```
⟨xpm:rule xpm:rid = "r.5" xpm:sel = "sequence"⟩
   ⟨xpm:word xpm:rt = "41004200" xpm:rid = "r.5.1"/⟩
    ⟨!--property age--⟩
   ⟨xpm:word xpm:rt = "1300" xpm:rid = "r.5.2"/⟩
    ⟨!--is--⟩
   ⟨xpm:word xpm:rt = "5100" xpm:op =
     "LsAndEq" xpm:rid = "r.5.3"⟩20
   ⟨/xpm:word⟩⟨!--5100 ≡ less and equal--⟩
⟨/xpm:rule⟩
```

For all ⟨xpm:rule⟩, the query principle is PAS. The xpm:op in a fact likes a predicate associating an abstract concept and a reified concept to express a fact of statement (A op B). Similarly, "choice" (i.e., "there exist" $\exists$) indicates that only selected facts are processed by the rule.

For all ⟨xpm:preference⟩, the query principle is NAF, followed by superiority relation. For example, the following preferences mean that, for all not cheapest price and not largest size, they failed. If both exist, then the cheapest is selected.

```
⟨xpm:preference xpm:rid = "p.1" xpm:op = "1"⟩
   ⟨xpm:phrase xpm:sel = "sequence" xpm:rid = "p.1.1"⟩
      ⟨xpm:word xpm:rt = "1600" xpm:op =
        "min" xpm:rid = "p.1.1.1"/⟩
      ⟨!--cheapest--⟩
      ⟨xpm:word xpm:rt = "2700" xpm:rid =
        "p.1.1.2"/⟩⟨!--price--⟩
   ⟨/xpm:phrase⟩
⟨/xpm:preference⟩
⟨xpm:preference xpm:rid = "p.2" xpm:op = "2"⟩
   ⟨xpm:phrase xpm:sel = "sequence" xpm:rid = "p.2.1"⟩
      ⟨xpm:word xpm:rt = "4400" xpm:op =
        "max" xpm:rid = "p.2.1.1"/⟩
      ⟨!--largest--⟩
      ⟨xpm:word xpm:rt = "2100" xpm:rid =
        "p.2.1.2"/⟩⟨!--size--⟩
   ⟨/xpm:phrase⟩
   ⟨/xpm:preference⟩
```

RuleXPM is simple. This is because all facts within ⟨rule⟩ and ⟨preference⟩ are just RT-ed concepts in LVs/CVs of ConexNet no matter whether they are nouns, verbs, adjectives, adverbs, phrases, statements, or documents. The processing of RT and the processing of rules can be separated. This is highly flexible to convert any BD and business activity to a RuleXPM document to process.

B. *RuleXPM-Based Processing for Outgoing Activity*

When the incoming activity is converted to a RuleXPM document, it is ready to be processed to generate another XPMR document for a next activity. This research addresses how to generate the new XPMR document, which exactly matches the business need of a local participating system and has no semantic conflict with the processing capability of the participating system who will receive the XPMR document.

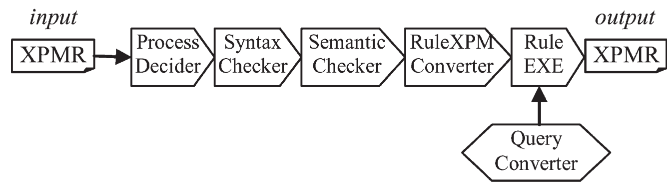


Fig. 8. RuleXPM inference procedure.

In the design of a RuleXPM inference engine, the aforementioned problems are solved by adopting the concept separation strategy. In particular, the outgoing activity (i.e., output) is semantically generated by separating the inputs of EmpnetEngine as a multiphase forward-chaining inference, where different inputs are separated into external inputs requiring universal semantic understanding and internal inputs independent of external environment. The match–act cycle is built in several phases in the entire processing of the outgoing activity, namely, the RuleXPM inference procedure as shown in Fig. 8.

This procedure is processed based on the following five types of resources:

1) incoming RuleXPM document converted from incoming XPMR document;
2) XPMT library;
3) RuleXPM-based stored execution rules (SERs), which stipulate action methods on XPMTs;
4) RuleXPM-based BPPs, which describe how a participating system stipulates the flow of the activity from one to another and which XPMT activity should be associated with;
5) knowledge base includes either relational DB [Structured Query Language (SQL)] or XPMR or both.

With these prepared sources, the RuleXPM inference procedure is designed with the following controls:

1) *process decider*: to determine which next activity will be executed based on the RuleXPM BPP, e.g., Fig. 5;
2) *syntax check*: to examine whether the incoming XPMR document is consistent with the XPM syntactic rules;
3) *semantic check*: to examine whether the incoming XPMR document is consistent with mutually understandable ConexNet concepts through RT;
4) *RuleXPM converter*: to convert the incoming XPMR document to RuleXPM document following the translation rules provided in Table IV;
5) *rule execution (EXE)*: to execute the incoming XPMR document based on BPP and SER to populate the reified data to XPMT to derive the next XPMR;
6) *query conversion*: to translate the RuleXPM documents and RuleXPM (SER), if necessary, into queries on SQL DB or XPMR documents for the needed information.

The RuleXPM inference procedure is to infer any incoming heterogeneous e-marketplace activity to a semantically consistent next activity.

Architecturally, the RuleXPM inference engine, shown in Fig. 9, is generically designed in the data part and the execution part. The data part consists of external data and internal data. The external data are the incoming messages from ConexNet and their defining XPM syntax and CONEX vocabularies (RT-ed). The internal data are composite concepts of a document template library (XPMT) and BPPs, SERs, and data sources

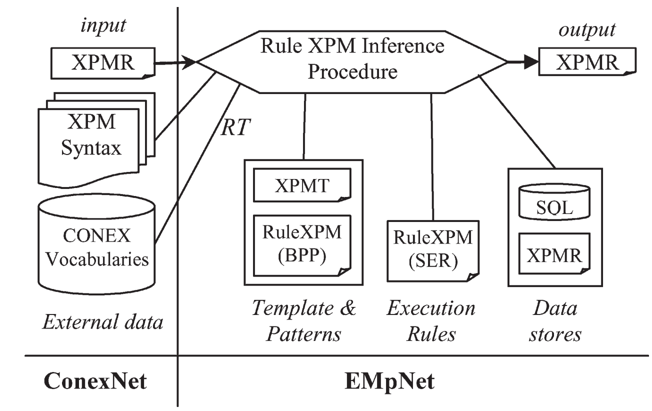


Fig. 9. RuleXPM inference engine.

```
IF XPMR THEN Execute(Decider)
IF h1 ∈ H ∈ XPMR THEN
   IF h1∈BPP ∧ h1→h2(XPMT) THEN
      Check(XPMR ← XPM)
      Check(r:R ∪ p:P ∈ VOC)
   ELSE Exit
Convert(XPMR) → RuleXPM
EXE(SER)
IF (r:R ∪ p:P ∈ RuleXPM) → DS ≠ ∅ THEN
   Execute(Query Conversion) → QueryResult
   IF r:R THEN QueryResult('r') → x
   IF p:P THEN QueryResult('r') → ¬y
   XPMR = Execute(XPMT, x, ¬y, yi > yj)
IF (r:R ∪ p:P ∈ RuleXPM) → DS = ∅ THEN
   IF r:R THEN r → x
   IF p:P THEN p → ¬y
   XPMR = Execute(XPMT, x, ¬y, ¬yi > ¬yj)
```

Fig. 10. RIA.

of XPMR and SQL relational DBs. The execution part is the RuleXPM inference procedure.

This architecture supports the concept separation strategy and makes the designed RuleXPM inference engine generic and suitable for use in all types of EMpNet participating systems. In this architecture, the inference engine is modular, i.e., each inference module is independent and reusable and the data in use can be dynamically generated, and is contextual.

VI. RuleXPM Inference Algorithm

In this section, we provide a generic *RIA*, which is used in RuleXPM inference engine, to more accurately describe the generation of the next activity. It is applicable to all participating systems of UPA, FIRM, and EMP. The general idea of this algorithm is that any action corresponding to the next activity is always dynamically triggered based on the RuleXPM BPP.

A. Preconditions of RIA

Based on Fig. 8, the RIA has preconditions, as follows:

1) XPM, an XPM schema for parsing and validating all XPM documents;
2) VOC, ConexNet LVs and CVs collaboratively designed and semantically consistent for all participating systems;
3) $XPMR \rightarrow RuleXPM = (H, R, P)$, where H is the document head containing the activity concept h, R is a set of normal concepts with attribute $sel \neq preference$, and P is a set of preference concepts with attribute $sel = preference$;
4) SER, a set of XPMR SERs;
5) BPP, a set of XPMR BPPs;
6) $XPMT$, a set of XPM document templates (XPMT);
7) DS, a data source either in relational DB SQL or in reified documents XPMR.

B. Postconditions of RIA

The postcondition of RIA is simply an XPMR document.

C. RIA Computation

To compute the postcondition from the preconditions, the algorithm is developed as shown in Fig. 10.

TABLE V
Kelvin's Inquiry Sheet

Basic Inquiry	
Size	800~1000 sqf
Price	HKD 6000 ~ 12000
Bedroom	2
Floor	>10
Property Age	<=20
Furnished	True
Cat Permit	True
Preference ranking	
Price – cheapest	1
Size – Largest	2
PropertyAge – newest	3

In this algorithm, the rule execution (EXE) machine is generic and does not associate with any particular activity (or action or operation). The behavior of the activity can be dynamically edited in particular RuleXPM execution rules (SER).

To validate the algorithm, we have implemented a prototype, which computes an example of Kelvin's inquiry (see Table V) for an offer. It shows the favorable result of what we have expected. The detailed information can be found in [57].

VII. Experiments and Results

A. Design of Experiment

The experiment evaluates the system performance by applying the RuleXPM prototype and measures the execution time of the RuleXPM inference engine. An experimental model was designed to record the execution time for both the RuleXPM inference engine and EMpNet. VMware Server 2.0 was employed to establish the experimental environment. VMware Server is quite powerful and can create, edit, and manage virtual machines. It can also consolidate many independently run virtual machines to create a complete testing environment. The experiment analyzes records through building a proper trend line with R-squared value in MS Excel [43]. This method of analysis utilizes minimum variance to minimize errors and thus to conclude the performance of the prototype.

The experiment executes a Kelvin example scenario of making an inquiry from a UPA and returns the best offer from all firms through EMP. The inquiry data are shown in Table V.

TABLE VI
EXPERIMENTAL RESULTS FOR $\mathcal{N} = 1$ WITH CHANGING RECORD NUMBER

Number of Records	**100**	**200**	**300**	**400**	**500**
Execution Time (ms) - SQL	45	73	96	123	154
Execution Time (ms) - XPMR	133	260	402	515	648
Trend line	**Linear**	**Log**	**Power**	**Exp**	
R-squared value (SQL)	0.9977	0.9355	0.9947	0.9725	
R-squared value (XPMR)	0.9991	0.9512	0.9994	0.9448	
Semantic consistency maintenance: 100% (in meaning comparison)					
Semantic inference correctness: Yes					

The particular experiment settings for the Kelvin example are as follows.

1) *Scenario setting.* A user Kelvin creates an inquiry of an apartment rental with the particulars detailed in Table V using a user editor by applying its LV and mapped CV. He sends the inquiry to his UPA in XPMR format. UPA sends this XPMR inquiry sheet to its e-marketplace system EMP 1, which again sends it to another e-marketplace EMP 2. EMP 1 and EMP 2 process the received inquiry and infer which firms the inquiry sheet should be sent based on the e-marketplace rules of "IF apartment rental THEN X." When firms' enterprise information systems (FIRM) receive this inquiry, each of them processes the received inquiry and makes an offer based on their offer rules and data sources. When the offer is made, it is sent to EMP 1 or EMP 2 respectively, where EMP 2 again sends the offer results to EMP 1. EMP 1 finally infers the best apartment rental offer to UPA using a set of business rules based on the set of received offer sheets.
2) *Data setting.* Each node (i.e., EMpNet participating system) includes a set of LVs/CVs $\subset$ ConexNet, a set of user rules, a set of XPMTs, a set XPMR data source, a relational DB, and a set of execution rules.
3) *Verification of semantic inference correctness and semantic consistency maintenance.* They are verified by a human using visual comparison between the incoming document and the outgoing document between inferences.

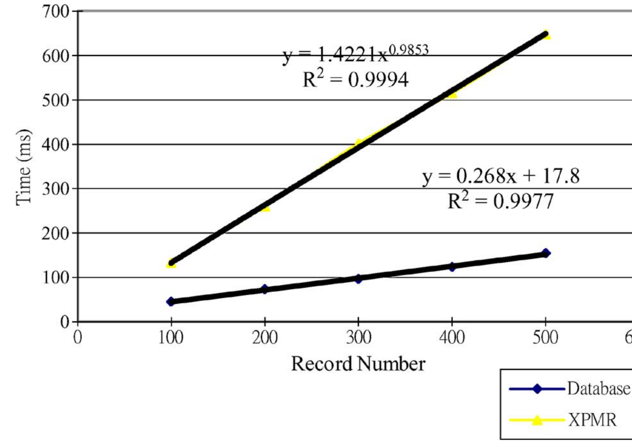


Fig. 11. Trend line for execution time when $\mathcal{N} = 1$.

TABLE VII
EXPERIMENTAL RESULTS FOR $\mathcal{N} = 3$ WITH CHANGING RECORD NUMBER

Number of Records	**100**	**200**	**300**	**400**	**500**
Execution Time (ms) - SQL	119	211	288	374	455
Execution Time (ms) - XPMR	209	391	573	766	908
Trend line	**Linear**	**Log**	**Power**	**Exp**	
R-squared value (SQL)	0.9994	0.9541	0.9993	0.9543	
R-squared value (XPMR)	0.9979	0.9568	0.9994	0.9462	
Semantic consistency maintenance: 100% (in meaning comparison)					
Semantic inference correctness: Yes					

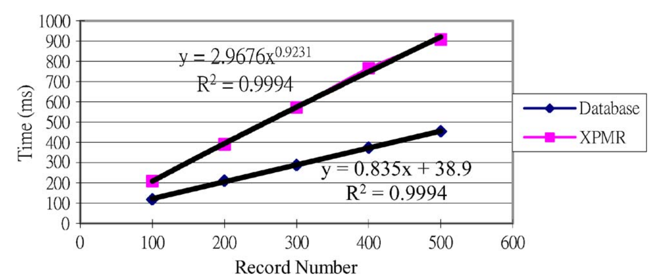


Fig. 12. Trend line for execution time when $\mathcal{N} = 3$.

B. Experimental Results

This section presents three groups of experimental results, which are performance tests in a single engine with changing record number, EMpNet with changing record number, and EMpNet with changing FIRM number.

Performance When Node $\mathcal{N} = 1$ With Changing Record Number: Table VI presents the execution time and R-squared values of a RuleXPM inference engine that processes an incoming XPMR document, needing to query data source DS $= DB$ or DS $= XPMR$ to derive a next activity.

By selecting the proper R-squared values, the trend lines have been selected to best describe the performance results when DS = SQL or DS = XPMR, shown in Fig. 11. The experiment shows that the system performance decreases as the record number of data sources increases, but the execution time for DS = XPMR is longer than that for DS = SQL.

Performance When Node $\mathcal{N} = 3$ With Changing Record Number: Table VII shows the execution time and R-squared values measured from issuing an inquiry for an offer through the path of one UPA $\rightarrow$ one EMP $\rightarrow$ one FIRM $\rightarrow$ one EMP $\rightarrow$ one UPA, where the data sources of each node have the alternatives of DS = SQL or DS = XPMR.

Fig. 12 shows a performance time increase when the record number increases. Similar to experiment 1, the XPMR query has worse performance than the SQL query.

Performance When Node $\mathcal{N} = 8$ With Fixed Record Number: Table VIII shows the system execution time and R-squared values measured from issuing an inquiry for an offer from a UPA through the path of UPA $\rightarrow$ (EMP 1, EMP 1 $\rightarrow$ EMP 2) $\rightarrow$ (FIRM 1, FIRM 2, FIRM 2, FIRM 4, FIRM 5) $\rightarrow$ (EMP 1, EMP 2 $\rightarrow$ EMP 1) $\rightarrow$ UPA, where each node has alternatives of SQL DB or XPMR as data sources. There are a total of 500 data records that are distributed in five FIRM data sources.

The result in Fig. 13 indicates that, as the FIRM number increases, the performance improves, and the improvement is slightly better for XPMR as data sources than SQL DB. This is an interesting result, implying that, when the heterogeneous e-marketplace becomes more distributed, i.e., more firms are

TABLE VIII
EXPERIMENTAL RESULTS FOR $\mathcal{N} = 8$ WITH FIXED RECORD NUMBER

Number of FIRMs	1	2	3	4	5
Execution Time (ms) - SQL	455	420	408	388	363
Execution Time (ms) - XPMR	908	822	746	698	660
Trend line	**Linear**	**Log**	**Power**	**Exp**	
R-squared value (SQL)	0.9779	0.9926	0.9844	0.9812	
R-squared value (XPMR)	0.9718	0.9616	0.948	0.9843	
Semantic consistency maintenance: 100% (in meaning comparison)					
Semantic inference correctness: Yes					

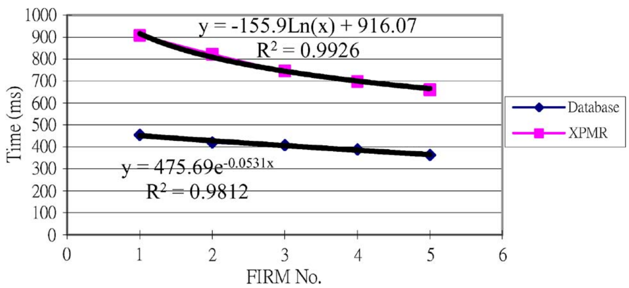


Fig. 13. Trend line for execution time when $\mathcal{N} = 8$.

added into the e-marketplaces, the inference performance will not reduce.

The experimental results show the following.

1) All the final results are correctly inferred, and semantic consistency between heterogeneous e-marketplace activities is maintained through human verification.
2) The execution time increases as the system records increase, assuming that the FIRM number remains unchanged.
3) The increasing speed of the execution time on DS = SQL is lower than that on DS = XPMR. This implies a limitation of applying XPMR as data store.
4) The execution time decreases while the FIRM number increases, assuming that the total system records remain unchanged. This is a good feature that is particularly useful for distributed EMpNet systems.

VIII. DISCUSSION

Semantic inference on activities generally involves three important components of facts, logic, and rules as existing research [1], [4], [16], [54]. Within the three components, how to compose facts and rules is the key to semantic inference.

Existing approaches popularly compose facts and rules in various types of ontology. However, an important feature of ontology is domain wide. This feature means that ontologies developed in different systems of sellers and buyers are autonomous and cannot interoperate with each other if they have different semantic contexts. Although much work on ontology alignment or matching has been done (e.g., the work described in [51]), it can only alleviate the ontology mismatch problem. The accuracy problem of ontology interpretation across domains/contexts is actually not solved.

While accurate interpretation is still a problem for using heterogeneous ontologies, an e-marketplace typically consists of two types of activity. The first type is those activities that do not require 100% accuracy of semantic interpretation by activity receivers. These activities include searching suppliers, recommending products, delivering advertisements, and making inquiries. The interpretation of these activities only requires higher similarity. The second type is those activities that require 100% accuracy of semantic interpretation by activity receivers. These activities often include offer, counteroffer, acceptance notice, order sheet, and contract. Any misinterpretation will lead to legal consequences because the BDs contained in these activities are legally binding to legal responsibilities. Any misinterpretation may cause wrong execution of legal responsibilities. Unfortunately, most existing inference methods for e-marketplaces are either applicable for only a single domain or designed for only achieving higher similarity.

The RuleXPM method, suggested in this paper, targets at achieving 100% accuracy for semantic interpretation across domains/contexts. It replaces domain-wide ontology by ConexNet concepts [29], [32], which are collaboratively created between heterogeneous domains. The collaborative concepts ensure the semantic consistency between heterogeneous domains and contexts and thus can be applied to compose cross-domain/cross-context inferable activities. These eventually make a next activity semantically inferable by separating denotation from connotation and implementation.

IX. CONCLUSION

This paper has discussed a semantic inference problem that requires reasoning between heterogeneous e-marketplace activities. A novel RuleXPM approach has been proposed to derive a correct next activity in a heterogeneous e-marketplace environment. It has introduced a concept separation strategy to separate an activity into concept denotation, concept connotation, and concept implementation. With this separation, any heterogeneous activity is interoperable utilizing ConexNet, which is related to the work researched in maintaining semantic consistency between heterogeneous concepts [29], [32]. To implement this strategy, a RuleXPM schema has been designed for governing the message handling using defeasible logic [3], SWRL [64], and ConexNet concept [71], and a semantic inference engine has been developed for deriving a next activity for the intended recipient of EMpNet. In this engine, a generic RIA has been introduced, which guarantees the correct semantic inference. The correctness of the approach is demonstrated in a prototype where experiments are made to test the performance.

This paper has the following contributions:

1) provided a new understanding of heterogeneous activity inference;
2) proposed a new concept separation strategy to solve the heterogeneous activity inference problem, which is extremely useful for heterogeneous business process integration and interoperation;
3) designed a new semantic inference engine on a multiphase forward-chaining algorithm, which has clarified the handling procedures of solving the heterogeneous activity problem.

In addition, this research can be applied in many practical applications. For example, it is used to design and implement electronic and virtual marketplace functionality, such as marketing, trading, payment, and logistics, semantic integration systems, multilingual systems, and interenterprise collaboration systems. In the future, we plan to implement an automated offering system and an automated negotiation system for an e-marketplace based on the result of this work.

REFERENCES

[1] C. Anagnostopoulos and S. Hadjiefthymiades, “Advanced inference in situation-aware computing,” *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 5, pp. 1108–1115, Sep. 2009.

[2] G. Antoniou and A. Bikakis, “DR-Prolog: A system for defeasible reasoning with rules and ontologies on the semantic Web,” *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 2, pp. 233–245, Feb. 2007.

[3] G. Antoniou, D. Billington, G. Governatori, and M. J. Maher, “Representation results for defeasible logic,” *ACM Trans. Comput. Logic*, vol. 2, no. 2, pp. 255–287, Apr. 2001.

[4] G. Antoniou, T. Skylogiannis, A. Bikakis, M. Doerr, and N. Bassiliades, “DR-BROKERING: A semantic brokering system,” *Knowl.-Based Syst.*, vol. 20, no. 1, pp. 61–72, Feb. 2007.

[5] S. Bhiri, W. Gaaloul, M. Rouached, and M. Hauswirth, “Semantic Web services for satisfying SOA requirements,” in *Advances in Web Semantics I*. Berlin, Germany: Springer-Verlag, 2008, pp. 374–395.

[6] Y. Blanco-Fernández, J. Pazos-Arias, A. Gil-Solla, M. Ramos-Cabrer, M. López-Nores, J. García-Duque, A. Fernández-Vilas, R. Díaz-Redondo, and J. Bermejo-Muñoz, “A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems,” *Knowl.-Based Syst.*, vol. 21, no. 4, pp. 305–320, May 2008.

[7] K. Boukadi, C. Ghedira, Z. Maamar, D. Benslimane, and L. Vincent, “Context-aware data and IT services collaboration in e-business,” in *Transactions on Large-Scale Data- and Knowledge-Centered Systems I*. Berlin, Germany: Springer-Verlag, 2009, pp. 91–115.

[8] BPEL. [Online]. Available: http://www.service-architecture.com/web-services/articles/business_process_execution_language_for_web_services_bpel4ws.html

[9] L. Brownston, R. Farrel, E. Kant, and N. Martin, *Programming Expert Systems in OPS5: An Introduction to Rule Base Programming*. Boston, MA: Addison-Wesley, 1985.

[10] J. Burns, W. Winstead, and D. Haworth, “Semantic nets as paradigms for both causal and judgmental knowledge representation,” *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 1, pp. 58–67, Jan/Feb. 1989.

[11] J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, “Jena: Implementing the semantic Web recommendations,” in *Proc. WWW*, 2004, pp. 74–83.

[12] C.-P. Che, J. Guo, and Z. Gong, “Inference on heterogeneous e-marketplace activities,” in *Proc. IEEE SMC*, 2009, pp. 3634–3639.

[13] C. Cheng, C. Chan, and K. Lin, “Intelligent agents for e-marketplace: Negotiation with issue trade-offs by fuzzy inference systems,” *Decision Support Syst.*, vol. 42, no. 2, pp. 626–638, Nov. 2006.

[14] P. Cholak, E. Kinber, R. Downey, M. Kummer, L. Fortnow, S. Kurtz, W. Gasarch, and T. Slaman, “Degrees of inferability,” in *Proc. COLT*, 1992, pp. 180–192.

[15] CLIPS. [Online]. Available: http://clipsrules.sourceforge.net/

[16] S. Colucci, T. Noia, A. Pinto, M. Ruta, A. Ragone, and E. Tinelli, “A nonmonotonic approach to semantic matchmaking and request refinement in e-marketplaces,” *Int. J. Electron. Commerce*, vol. 12, no. 2, pp. 127–154, Winter 2007.

[17] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, “Unraveling the Web services Web: An introduction to SOAP, WSDL, and UDDI,” *IEEE Internet Comput.*, vol. 6, no. 2, pp. 86–93, Mar./Apr. 2002.

[18] Drools. [Online]. Available: http://jboss.org/drools/

[19] Dublin Core. [Online]. Available: http://dublincore.org/

[20] ecl@ss. [Online]. Available: http://www.eclass.de/

[21] C. Forgy, “Rete: A fast algorithm for the many pattern/many object pattern match problem,” *Artif. Intell.*, vol. 19, no. 1, pp. 17–37, Sep. 1982.

[22] Gene Ontology. [Online]. Available: http://geneontology.org/

[23] L. Godo and R. Rodríguez, “Graded similarity-based semantics for nonmonotonic inferences,” *Ann. Math. Artif. Intell.*, vol. 34, no. 1–3, pp. 89–105, Mar. 2002.

[24] T. Gruber, “A translation approach to portable ontologies,” *Knowl. Acq.*, vol. 5, no. 2, pp. 199–220, Jun. 1993.

[25] X. Guan, “Context-based translation of constant concept values in e-business,” M.S. thesis, Univ. Macau, Macau, China, 2008.

[26] J. Guo, “Inter-enterprise business document exchange,” in *Proc. ICEC*, 2006, pp. 427–437.

[27] J. Guo, “A term in search of the infrastructure of electronic markets,” in *Research and Practical Issue of Enterprise Information System II*. Boston, MA: Springer-Verlag, 2007, pp. 831–840.

[28] J. Guo, “Business-to-business electronic market place selection,” *Enterprise Inf. Syst.*, vol. 1, no. 4, pp. 383–419, Nov. 2007.

[29] J. Guo, *Collaborative Concept Exchange*. Saarbrücken, Germany: VDM Verlag, 2008.

[30] J. Guo, Z. Hu, C.-K. Chan, Y. Luo, and C. Chan, “Document-oriented heterogeneous business process integration through collaborative e-marketplace,” in *Proc. ICEC,* 2008, article 39.

[31] J. Guo, Z. Hu, G. Antoniou, and C.-K. Chan, “Answering an inquiry from heterogeneous contexts,” in *Proc. ICEBE*, 2008, pp. 113–120.

[32] J. Guo, “Collaborative conceptualization: Towards a conceptual foundation of interoperable electronic product catalogue system design,” *Enterprise Inf. Syst.*, vol. 3, no. 1, pp. 59–94, Feb. 2009.

[33] J. Guo, ConexNet: A Collaborative Concept Exchange Network. [Online]. Available: http://www.sftw.umac.mo/~jzguo/pages/pub/ConexNet.pdf

[34] R. Hicks, “The no inference engine theory—Performing conflict resolution during development,” *Decision Support Syst.*, vol. 43, no. 2, pp. 435–444, Mar. 2007.

[35] C. M. Hoffmann and N. F. Stewart, “Accuracy and semantics in shape-interrogation applications,” *Graph. Models*, vol. 67, no. 5, pp. 373–389, Sep. 2005.

[36] Jess. [Online]. Available: http://jessrules.com/

[37] J. L. Laurière and M. Vialatte, “Snark: A language to represent declarative knowledge and an inference engine which uses heuristics,” in *Proc. IFIP Congr.*, 1986, pp. 811–816.

[38] C. Lee, C. Huang, L. Yang, and S. Rajasekaran, “Distributed path-based inference in semantic networks,” *J. Supercomput.*, vol. 29, no. 2, pp. 211–227, Aug. 2004.

[39] H. Lee and W. Moseley, “A parallel inference engine (PIE),” in *Proc. CSC*, 1987, p. 360.

[40] N. Lohmann, P. Massuthe, C. Stahl, and D. Weinberg, “Analyzing interacting WS-BPEL processes using flexible model generation,” *Data Knowl. Eng.*, vol. 64, no. 1, pp. 38–54, Jan. 2008.

[41] Y. Luo, “Semantic integration of heterogeneous online hotel information systems,” M.S. thesis, Univ. Macau, Macau, China, 2008.

[42] R. Martínez-Béjar, J. Cadenas, H. Shirazi, and P. Compton, “A semantics-driven, fuzzy logic-based approach to knowledge representation and inference,” *Expert Syst. Appl.*, vol. 36, no. 2, pp. 1940–1960, Mar. 2009.

[43] Microsoft, R-Squared Value in MS Excel. [Online]. Available: http://office.microsoft.com/en-us/help/HP052623211033.aspx?pid=CH101178621033

[44] D. P. Miranker, “TREAT: A better match algorithm for AI production systems,” in *Proc. AAAI*, 1987, pp. 42–47.

[45] P. Morizet-Mahoudeaux, “Maintaining consistency of database during monitoring of an evolving process by a knowledge-based system,” *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 1, pp. 47–60, Jan./Feb. 1991.

[46] P. Muro-Medrano, J. Banares, and J. Villarroel, “Knowledge representation-oriented nets for discrete event system applications,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 28, no. 2, pp. 183–198, Mar. 1998.

[47] N. Murray and E. Rosenthal, “Inference with path resolution and semantic graphs,” *J. Assoc. Comput. Mach.*, vol. 34, no. 2, pp. 225–254, Apr. 1987.

[48] M. Nelson, “The adoption and diffusion of interorganizational system standards and process innovations,” Ph.D. dissertation, Univ. Illinois Urbana-Champaign, Champaign, IL, 2003.

[49] J. Nigro and Y. Barloy, “The meta-inferences engine: A new tool to manipulate metaknowledge,” *Knowl.-Based Syst.*, vol. 21, no. 7, pp. 588–598, Oct. 2008.

[50] N. Noy, “Semantic integration: A survey of ontology-based approaches,” *ACM SIGMOD Rec.*, vol. 33, no. 4, pp. 65–70, Dec. 2004.

[51] Ontology Matching. [Online]. Available: www.ontologymatching.org/publications.html

[52] O. Perrin and C. Godart, “An approach to implement contracts as trusted intermediaries,” in *Proc. WEC*, 2004, pp. 71–78.

[53] V. Punyakanok, D. Roth, and W. Yih, “The importance of syntactic parsing and inference in semantic role labeling,” *Comput. Linguist.*, vol. 34, no. 2, pp. 257–287, Jun. 2008.

[54] A. Ragone, U. Straccia, T. Noia, E. Sciascio, and F. Donini, “Vague knowledge bases for matchmaking in P2P e-marketplaces,” in *Proc. ESWC*, vol. 4519, *LNCS*, 2007, pp. 414–428.

[55] A. Ragone, U. Straccia, T. Noia, E. Sciascio, and F. Donini, “Fuzzy matchmaking in e-marketplaces of peer entities using datalog,” *Fuzzy Sets Syst.*, vol. 160, no. 2, pp. 251–268, Jan. 2009.

[56] J. D. Roo, Euler Proof Mechanism. [Online]. Available: http://www.agfa.com/w3c/Euler/

[57] RuleXPM Demo. [Online]. Available: www.sftw.umac.mo/~jzguo/pages/RuleXPMDemo

[58] D. Russomanno, “A plausible inference prototype for the semantic Web,” *J. Intell. Inf. Syst.*, vol. 26, no. 3, pp. 227–246, May 2006.

[59] Z. Shan, T. Liu, Y. Qu, and F. Ren, “Modeling and inference of extended interval temporal logic for nondeterministic intervals,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 5, pp. 682–696, Sep. 2005.

[60] Soar. [Online]. Available: http://sitemaker.umich.edu/soar/home

[61] SOAP. [Online]. Available: http://www.w3.org/2000/xp/

[62] S. Sun, Q. Zeng, and H. Wang, “Process-mining-based workflow model fragmentation for distributed execution,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 2, pp. 294–310, Mar. 2011.

[63] H. Souza, A. Moura, and M. Cavalcanti, “Integrating ontologies based on P2P mappings,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 5, pp. 1071–1082, Sep. 2010.

[64] SWRL. [Online]. Available: http://www.w3.org/Submission/SWRL/

[65] W. Tan, Y. Xu, W. Xu, L. Xu, X. Zhao, L. Wang, and L. Fu, “A methodology toward manufacturing grid-based virtual enterprise operation platform,” *Enterprise Inf. Syst.*, vol. 4, no. 3, pp. 283–309, Aug. 2010.

[66] UNSPSC. [Online]. Available: http://unspsc.org/

[67] S. Wang, S. Zheng, L. Xu, D. Li, and H. Meng, “A literature review of electronic marketplace research: Themes, theories and an integrative framework,” *Inf. Syst. Frontiers*, vol. 10, no. 5, pp. 555–571, Nov. 2008.

[68] WSDL. [Online]. Available: http://www.w3.org/TR/wsdl.html

[69] Z. Wu, G. Eadon, S. Das, E. I. Chong, V. Kolovski, M. Annamalai, and J. Srinivasan, “Implementing an inference engine for RDFS/OWL constructs and user-defined rules in Oracle,” in *Proc. ICDE*, 2008, pp. 1239–1248.

[70] XPM Editor. [Online]. Available: http://www.sftw.umac.mo/~jzguo/pages/TexVDF_Demo/

[71] XPM Resource. [Online]. Available: http://www.sftw.umac.mo/~jzguo/pages/resource.html

[72] X. Zhao, C. Liu, Y. Yang, and W. Sadiq, “Aligning collaborative business processes—An organization-oriented perspective,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 6, pp. 1152–1164, Nov. 2009.

[73] D. Zhou, Y. Fu, S. Zhong, and R. Zhao, “The Rete algorithm improvement and implementation,” in *Proc. ICIII*, 2008, pp. 426–429.

Jingzhi Guo (M’05) received the B.Econ. degree in international business management from the University of International Business and Economics, Beijing, China, in 1988, the M.Sc. degree in computation from the University of Manchester, Manchester, U.K., and the Ph.D. degree in Internet computing and electronic commerce from Griffith University, Brisbane, Australia, in 2005.

He is currently an Assistant Professor in e-commerce technology with the Department of Computer and Information Science, University of Macau, Macau, China. His principal research is in the fields of concept representation, semantic integration, and collaboration systems, mainly applied to the fields of e-commerce, the electronic marketplace, e-banking, and virtual world.

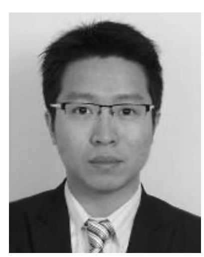
Lida Xu received the M.S. degree in information science from the University of Science and Technology of China, Heifei, China, in 1981, and the Ph.D. degree in systems sciences with emphasis on information systems from Portland State University, Portland, OR, in 1986.

He is an Eminent Professor of information technology with Old Dominion University, Norfolk, VA. He is also a Research Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and the Founding Director of the Institute of Systems Science and Engineering, Wuhan University of Technology, Wuhan, China. He is the Editor-in-Chief of multiple major publications on enterprise information systems launched by the world’s premier publishers. These publications include *Enterprise Information Systems* (Taylor & Francis), *Advances in Enterprise Information Systems Series* (Taylor & Francis), and *Advances in Systems Science and Engineering Series* (Taylor & Francis).

Dr. Xu serves as the Chair of the Enterprise Information Systems Technical Committee of the IEEE Systems, Man, and Cybernetics Society and the Chair of the International Federation for Information Processing Technical Committee 8 Workgroup 8.9.

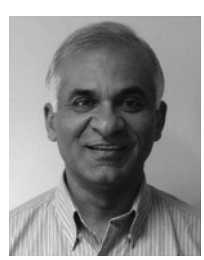
Zhiguo Gong (A’10) received the B.S. degree in mathematics from Hebei Normal University, Shijiazhuang, China, in 1983, the M.S. degree in mathematics from Peking University, Beijing, China, in 1998, and the Ph.D. degree from the Department of Computer Science, Chinese Academy of Sciences, Beijing, in 1998.

He is currently an Associate Professor and the Head of computer science with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include databases, digital libraries, Web information retrieval, and Web mining.



Chin-Pang Che received the B.S. degree in computer science from South China University of Technology, Guangzhou, China, in 2003 and the M.S. degree in e-commerce technology from the Department of Computer and Information Science, University of Macau, Macau, China, in 2009.

He is currently working in the real-estate industry. He is also currently with the University of Macau. His research interests include real-estate electronic marketplaces.



Sohail S. Chaudhry received the M.S. degree in industrial and management engineering and the M.Phil. and Ph.D. degrees in industrial engineering and operations research from Columbia University, New York, NY, in 1977, 1979, and 1985, respectively.

He is currently a Professor with the Department of Management and Operations/International Business, Villanova University, Villanova, PA. He serves on the editorial boards of several international journals in various capacities. His publications are mainly in the fields of location theory, production systems, management and control of quality, information systems, and human factors.