

Applying Semantic Web Rules to Business-to-Business Negotiation

Hsun-Ming Lee

Texas State University - San Marcos

E-Mail: sl20@txstate.edu

Yu-Liang Chi

Chung Yuan Christian University

E-Mail: maxchi@mis.cycu.edu.tw

Mayur R. Mehta

Texas State University - San Marcos

E-Mail: mm07@txstate.edu

ABSTRACT

Negotiation automation deals with business processes that frequently require the input of human knowledge to achieve their goal – that of agreement in negotiations. However, traditional business-to-business (B2B) vocabularies on the Web are limited to data exchange, and do not extend to the sharing and production of knowledge. In this paper, we explore the future of the Web: the semantic Web. This study investigates the Electronic Business using Extensible Markup Language (ebXML) platform and introduces a rule base by which negotiation parties can dynamically communicate the logic of their negotiation strategies through the definition of semantic Web rules. Using standardized semantic Web technologies for all of the negotiation parties, we present the application of negotiation automation to agendas and explanations. In particular, explanations take advantage of the technology that allows rules (or human logic) to operate in remote systems and to interface between human beings and machines.

Keywords: Semantic Web, Rule Base, Negotiation Automation, Electronic Business

INTRODUCTION

The information industry has been integrating business-to-business (B2B) processes using standardized vocabularies such as Electronic Business using Extensible Markup Language (ebXML) and RosettaNet (www.rosettanet.org) for many years. These standards provide an interoperable foundation for the B2B e-marketplace using Extensible Markup Language (XML) technology. For example, a system has been successfully deployed to send invoices and statements electronically between a buyer and a seller in Hong Kong using ebXML, through which the buyer can access procurement and reconciliation data from its own database for operational planning and statistical activities in a timely and efficient fashion (ebXML, 2003).

The ebXML standards specify e-business transactions on the Web by defining common terms for data components, business processes, and collaboration protocol agreements. Negotiation automation represents a special type of business process, in which the capturing of the full requirements is a challenge, as it usually involves dynamic human decisions. The semantic Web, which is the future of the Web, provides advanced XML tools such as OWL (Web Ontology Language) to represent human concepts (W3C, 2004b). In addition, the Semantic Web Rule Language (SWRL) has been proposed to integrate these concepts with rules (W3C, 2004a).

Recently, the SweetDeal project used dynamically editable knowledge representation with semantic Web rules to facilitate e-contracting processes, and suggested that there are significant advantages (Grosf and Poon, 2004). Semantic Web rules provide a high level of conceptual abstraction, and it is possible to communicate automatically through them as they are interoperable between heterogeneous rule/database systems and the rule-based agents that use those systems. In addition, there is a relatively high degree of automated reusability as a result of the declarative, modular, and interoperable nature of the knowledge representation. Specifically, the approach of the SweetDeal project has demonstrated the ability of semantic Web rules to handle business process exceptions, such as late delivery, in e-commerce.

To declare, operate, and exchange the semantic Web rules in B2B negotiation, negotiation agents in e-commerce systems need to communicate based on protocols that are provided by industry standards. However, the negotiation protocols in ebXML currently focus on the creation and exchange of negotiation issues, which cannot be changed once the negotiation process is started. More importantly, the protocols only represent limited human knowledge and do not directly support a standard rule

expression, which is needed for implementing advanced negotiation models such as negotiation strategies. In a negotiation, strategies are the rationale for choosing among various actions at a certain stage (Rosenschein and Zlotkin, 1994).

This study conducts applied research to develop a novel approach to setting up negotiation strategies that are carried out by software agents in B2B negotiation systems. The semantic Web rules, which represent human logic, allow the negotiation agents to exchange documents and make strategic decisions with rules. The first strategy is that of the agenda, which specifies the order in which issues are negotiated in multi-issue negotiation. In particular, an endogenous agenda allows negotiation parties to decide on the order of the negotiation process (Fatima et al., 2004; Fatima et al., 2003). The second strategy is that of explanations, which comprise additional information that explains why an offer is made or rejected in the argumentation-based negotiation model. Explanations are a form of persuasive argument for quickly reaching an agreement (Parsons et al., 1998; Rahwan et al., 2003). The use of rules allows XML-formatted agenda and explanation information to be communicated effectively to facilitate the B2B negotiation process.

SEMANTIC WEB RULE LANGUAGE

SWRL is based on a combination of OWL and Rule Markup Language (RuleML) (RuleML.org, 2001; W3C, 2004a). OWL, which was originally proposed for the development of a powerful knowledge-sharing semantic Web, specifies information using many kinds of human concepts, including classes (general items), the relationships that may exist between items, and the properties that those items may have. OWL-based applications or services are considered to be “intelligent” in the sense that they work more accurately at the human conceptual level (W3C, 2004c). We can also add another aspect of intelligence by constructing a rule base that reasons about the concepts. RuleML integrated with OWL has been chosen because they share the same underlying format of XML.

Briefly, an SWRL rule formally expresses an if-then rule using the OWL syntax. A rule expresses the implication relationship between the classes of body (the “if” clause) and head (the “then” clause), both of which consist of kinds of atoms. Figure 1 shows a simple rule which defines that parent and brother properties imply an uncle property. More complete syntactical information can be found in the SWRL specification.

For the sake of readability, a rule is often written in the following form:

antecedent \Rightarrow *consequent*,

where both *antecedent* and *consequent* are conjunctions of the atoms that are written as $a_1 \wedge \dots \wedge a_n$. Variables are indicated using the standard convention of prefixing them with a question mark (e.g., ?x). Using this syntax, the sample rule in Figure 1 would be written as follows.

hasParent(?personA,?personB) \wedge hasBrother(?personB,?personC) \Rightarrow hasUncle(?personA,?personC).

```
<ruleml:imp>
  <ruleml:_rlab ruleml:href="#example1"/>
  <ruleml:_body>
    <swrlx:individualPropertyAtom swrlx:property="hasParent">
      <ruleml:var>personA</ruleml:var>
      <ruleml:var>personB</ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom swrlx:property="hasBrother">
      <ruleml:var>personB</ruleml:var>
      <ruleml:var>personC</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom swrlx:property="hasUncle">
      <ruleml:var>personA</ruleml:var>
      <ruleml:var>personC</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>
```

Figure 1. A simple semantic Web rule

NEGOTIATION DOCUMENTS

The ebXML specification defines negotiation automation as a process that delivers Collaboration Protocol Agreement (CPA) documents to the two parties in a negotiation (OASIS, 2003). A party can initiate the process by submitting an offer as a CPA template (draft CPA). Both parties then exchange and modify the CPA template until an agreeable CPA is constructed. In general, the CPA template contains all of the necessary information that will be delivered in the final contract, with some items left open for negotiation. The Negotiation Description Document (NDD), which is used to describe the open items, defines the lists of choices and the negotiable ranges of the numeric values.

An NDD includes the following typical types of negotiable items.

- Value: for negotiating the value of the item.
- Unordered value: for negotiating the presence or absence of a member of a set of unordered values.
- Ordered value: for negotiating to choose between the members of a set of ordered values, where the preference is of a simple kind.
- Integer value: for expressing whether an integer is present or not and then the choice between different integer values using simple preference measures.
- Present or not: for negotiating the presence or absence of a value.
- Preference: for expressing preference among values of a similar nature

As should be obvious, the NDD provides a minimum set of functions to capture the parameters of the negotiation process. It meets the basic need, but is limited in its ability to support requirement changes during the process. For example, it is impossible for an NDD to specify the strategy by which a contractor wants to increase the minimum price to 3,500 dollars if HTTPS is chosen as the transport protocol. It is also static and cannot be updated after the process starts according to the ebXML negotiation protocol.

BUILDING A SEMANTIC WEB RULE BASE

A key issue in the development of this approach to setting up negotiation strategies is the building of a semantic Web rule base. In the system designed by this approach, trading partners dynamically update semantic Web rules as the B2B negotiation process is in progress. To simplify our discussion, we assume that the parties start the negotiation with a single CPA template. For the rules to have a common understanding of the structure of the negotiation items, all of the trading partners are required to have a copy of OWL document that represents all of the NDD data elements. Noy et al. (2001) suggested the use of an OWL modeling tool, such as Protégé, to establish a system ontology, which includes classes, properties, and individuals (Gennari et al., 2003; Noy et al., 2001). We develop negotiation ontology classes based on ebXML NDD Schema (see the Appendix). For example, the “NegotiableInformationItem” class has the “hasCPAValue” property, which indicates the value that is entered onto the CPA template for a negotiable item, and a “isPresentOrNot” property that describes whether the value must be fulfilled by the trading partners.

The OWL classes that are designed to support negotiation processes are considered to be ebXML extensions, and their individuals (or objects) are easily converted into the ebXML environment. To use our proposed negotiation support environment, a trading partner who requests offers would acquire the Protégé tool, import the classes into Protégé, and create a knowledge base comprising the individuals for defining the choices and negotiable ranges (see Figure 2). Once the classes and individuals are installed in the negotiation systems of all parties involved in the negotiation, such definitions become endogenous, and the individuals are programmatically updated by negotiation agents after the negotiation processes begin. The definitions must be satisfied by offers or counter offers.

The agents that run in the B2B environment execute the rules to adjust the values that are stored in the individuals. The rules are defined separately by trading partners (or users) to perform inferences. For example, Rule A is used to specify value dependency, in which the minimum price is increased to 3,500 dollars if the HTTPS transport protocol is requested in a software development contract. This rule is defined as follows.

Rule A

$$\begin{aligned} & \text{sameIndividual}(?itemA, \text{TransportProtocol}) \wedge \\ & \text{hasCPAValue}(\text{Transport_TransportProtocol}, \text{"HTTPS"}) \\ & \Rightarrow \text{hasSmallestValue}(\text{EndPointPrice}, 3500) \end{aligned}$$

The agent executes the rule after a CPA template (or an offer) is received, where “TransportProtocol” is an individual of negotiable items and “EndPointPrice” is an individual of end points. “EndPointPrice” is also a property of a range information individual that is defined for the “Price” negotiable item. As a result, the lower range of the “Price” negotiation item is updated.

Due to constant change of business environments, rules should be updateable after a negotiation begins. For example, an engineer in the company that negotiates the software contract in the above example discovers a simple procedure to configure the HTTPS transport protocol and the cost to support such software development is reduced than the original estimate. Therefore, the company can decide to decrease the minimum price to 3,200 dollars to win the contract. To make this change, the users simply update the Rule A for an agent in the negotiation system.

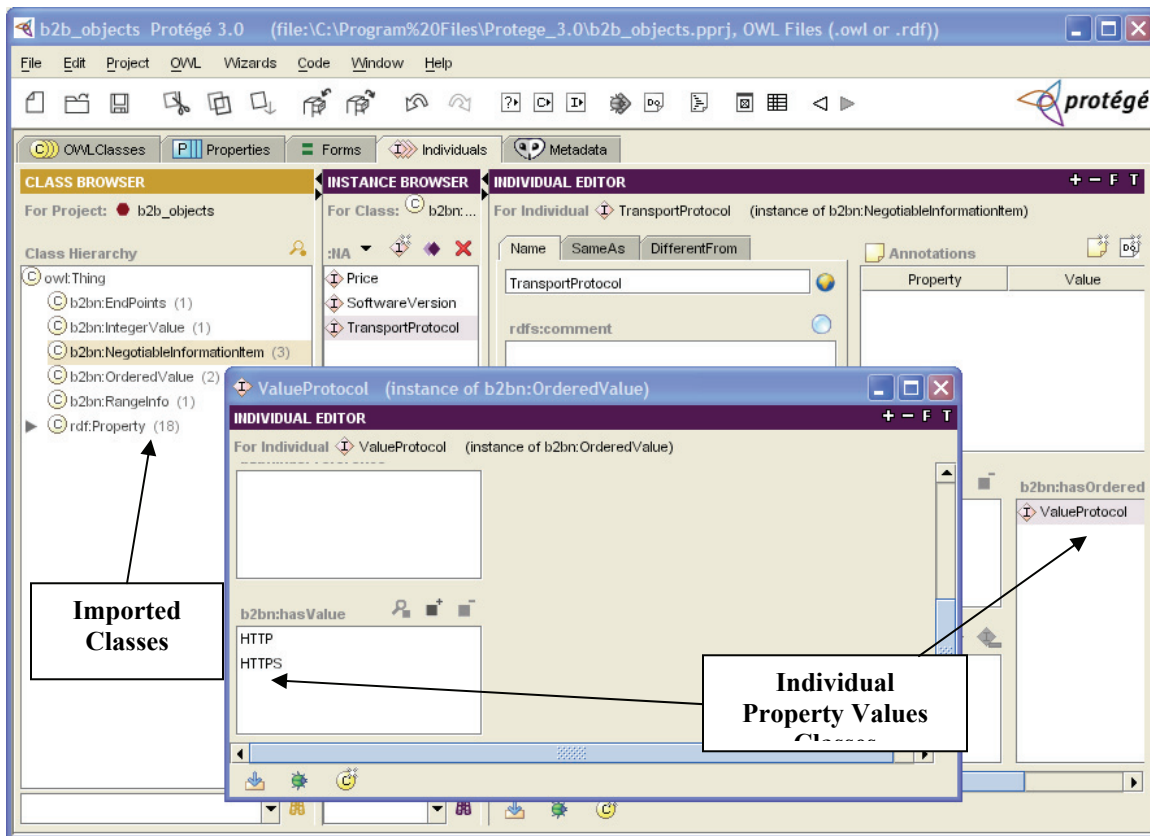


Figure 2. Creating and updating OWL individuals using Protégé

STRATEGIES: AGENDA AND EXPLANATION

The Rule A demonstrates a simple negotiation strategy. In this approach, the dynamic rule bases are utilized to build model-based negotiation strategies such as an agenda and an explanation. An agenda suggests the next action and an explanation is an action taken to accelerate the negotiation process. The agenda definition requires the order dependency of the negotiation items. For example, in Rule B, if the value of the transport protocol is present, then the software version must also be present in the offer. This rule that forces the order of two negotiable items is defined as follows.

Rule B

$$\begin{aligned} & \text{sameIndividual}(\text{?itemA}, \text{TransportProtocol}) \wedge \text{IsPresentOrNot}(\text{TransportProtocol}, \text{"Yes"}) \wedge \\ & \text{sameIndividual}(\text{?itemB}, \text{SoftwareVersion}) \\ & \Rightarrow \text{IsPresentOrNot}(\text{?itemB}, \text{"Yes"}). \end{aligned}$$

An explanation is implemented by including rules in negotiation documents. The following steps demonstrate how the negotiation explanation works.

1. Alice, who provides a software package, sets up Rule A to increase the software price for the package to support the HTTPS protocol during a negotiation.
2. An offer from Bob is rejected by Alice's agent because the offered price is less than 3,500 dollars and it requires HTTPS features.
3. The rule, accompanied with a counter offer, is sent to Bob's agent by Alice's agent.
4. Bob receives the counter offer from his agent and views the rule as an explanation.
5. Bob accepts the explanation and saves the rule in the rule base for his agent.
6. Bob prepares another offer and the rule is executed by his agent before the offer is sent to Alice.

The negotiation explanation is one of the best examples that take advantage of the mobility of the semantic Web rules, which can be easily delivered on the Web and can also be integrated into remote systems that are connected to the Web.

CONCLUSION AND DISCUSSION

XML is the de facto standard for developing B2B applications on the Web. Semantic Web technologies, which extend the XML standards, envision the Web in the future as a platform for the easy gathering and use of information and the submission of feedback about that information. Semantic Web rules add a high level of automation (or intelligence) to the processing of business documents across companies.

In this paper, we have argued that B2B negotiation represents the type of processes that require both technology standards and flexibility for communicating human logic on the Web. We have therefore applied semantic Web rules to the development of strategies for enhancing negotiation systems based on the ebXML standards. A key advantage of this application is the mobility of semantic Web rules in the B2B environment, where a rule that is declared by one trading partner can be effectively passed to and executed by the systems of all trading partners that share the same negotiation ontology. Moreover, we believe that negotiation systems developed in accordance with the approach presented in this paper are likely to operate more effectively than those based on the traditional ebXML standards, because such systems provide the following features.

- (1) A standardized rule base that can be modified during the negotiation process.
- (2) A traceable strategy because the rules that define the strategy are executed for all offers and counter offers in the negotiation process until an agreement is reached.
- (3) A reusable strategy that can be easily configured to restart a negotiation with a different trading partner.

Although we have limited our study to B2B negotiation processes, it is quite evident that semantic Web rules are very promising. Additional work is being pursued, including the development of B2B applications to adopt the concepts of semantic Web enabled Web services (Bussler et al., 2002). Such applications are to connect intelligent Web services using semantic Web rules to support complex business collaboration processes crossing enterprise boundaries.

REFERENCES

- Bussler, C., Maedche, A., & Fensel, D. (2002). A Conceptual Architecture for Semantic Web Enabled Web Services, *ACM SIGMOD Record*, 31 (4), 24-29.
- ebXML (2003). ebXML Case Study - ebMS Streamlines Invoicing and Reconciliation Process. Retrieved from http://www.ebxml.org/case_studies/documents/mtrc_saggiocase-v4-mm1-110203.doc.
- Fatima , S., Wooldridge, M., & Jennings, N. R. (2004). An Agenda Based Framework for Multi-issues Negotiation, *Artificial Intelligence*, 152 (1), 1-45.
- Fatima , S., Wooldridge, M., & Jennings, N. R. (2003). Optimal Agendas for Multi-issue Negotiation. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, 129-136.
- Gennari , J. H., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy , M., Eriksson, H., Noy, N. F., & Tu, S. W. (2003). The Evolution of Protégé: An Environment for Knowledge-Based Systems Development, *International Journal of Human-Computer Studies*, 58 (1), 89-123.
- Grosz, B. N. & Poon, T. C. (2004). SweetDeal: Representing Agent Contracts with Exceptions Using Semantic Web Rules, Ontologies, and Process Description, *International Journal of Electronic Commerce*, 8 (4), 61-98.
- Noy, N. F., Sintek, M., Decker, S., Crubézy, M., Fergerson, R. W., & Musen, M. A. (2001). Creating Semantic Web Contents with Protégé-2000, *IEEE Intelligent*

- Systems, 16 (2), pp. 60-71.
- OASIS (2003). ebXML Automated Negotiation of Collaboration Protocol Agreements Specification Version 0.0.4. Retrieved from <http://www.oasis-open.org/committees/ebxml-cppa/negotiation/negotiation.subteam.documents/>.
- Parsons, S., Sierra, C., & Jennings, N. R. (1998). Agents that Reason and Negotiate by Arguing, *Journal of Logic and Computation*, 8 (3), 261-292.
- Rahwan, I., Ramchurn, S. D., Jennings, N. R., McBurney, P., Parsons, S., & Sonenberg, L. (2003). Argumentation-Based Negotiation, *The Knowledge Engineering Review*, 18 (4), pp. 343-375.
- Rosenschein, J. S. & Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. The MIT Press, Cambridge, MA.
- RuleML.org (2001). RuleML DTDs Version 0.7. Retrieved from <http://www.ruleml.org/0.7/>.
- W3C (2004a). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Retrieved from <http://www.w3.org/Submission/SWRL>.
- W3C (2004b). OWL Web Ontology Language Reference. Retrieved from <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>.
- W3C (2004c). OWL Web Ontology Language Use Cases and Requirements. Retrieved from <http://www.w3.org/TR/webont-req/>.

APPENDIX

Sample NDD OWL classes

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:j.0="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://www.txstate.edu/cis/b2bnegotiation.owl#"
  xml:base="http://www.txstate.edu/cis/b2bnegotiation.owl">
  <owl:Ontology rdf:about="">
    <owl:imports
rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
  </owl:Ontology>
  <owl:Class rdf:ID="RangeInfo"/>
  <owl:Class rdf:ID="EndPoints"/>
  <owl:Class rdf:ID="NegotiableInformationItem"/>
  <owl:Class rdf:ID="IntegerValue"/>
  <owl:Class rdf:ID="OrderedValue"/>
  <owl:ObjectProperty rdf:ID="hasIntegerValue">
    <rdfs:domain rdf:resource="#NegotiableInformationItem"/>
    <rdfs:range rdf:resource="#IntegerValue"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasOrderedValue">
    <rdfs:domain rdf:resource="#NegotiableInformationItem"/>
    <rdfs:range rdf:resource="#OrderedValue"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasRangeInfo">
    <rdfs:domain rdf:resource="#IntegerValue"/>
    <rdfs:range rdf:resource="#RangeInfo"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="hasEndPoints">
    <rdfs:domain rdf:resource="#RangeInfo"/>
    <rdfs:range rdf:resource="#EndPoints"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="hasPresentOrNot">
    <rdfs:domain rdf:resource="#NegotiableInformationItem"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasCPAValue">
    <rdfs:domain rdf:resource="#NegotiableInformationItem"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="hasPreference">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#RangeInfo"/>

```

```
        <owl:Class rdf:about="#OrderedValue"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasLatestValue">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Endpoints"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasValue">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#OrderedValue"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasCPAPath">
  <rdfs:domain rdf:resource="#NegotiableInformationItem"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasSmallestValue">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Endpoints"/>
</owl:DatatypeProperty>
</rdf:RDF>
```

```
<!-- Created with Protege (with OWL Plugin 1.3, Build 225.4)
      http://protege.stanford.edu -->
```