Using Semantic Web Technology to Design Agent-to-Agent Argumentation Mechanism in an E-Marketplace

Abstract

In existing e-marketplaces, buyers can use search engines to find products that exactly match their demands, but some products those are potentially interesting to them cannot be found out. This research aims to design a multi-agent e-marketplace in which buyers and sellers can delegate their agents to argue over product attributes via an agent-to-agent argumentation mechanism. To make the idea possible, this research adopts the Semantic Web technology to express agents’ ontologies and uses an abstract argumentation framework with information gathering approach to support defeasible reasoning. A laboratory experiment is conducted to assess the performance of the argumentation mechanism. The experimental results show that the proposed system can help buyers to search both exactly and potentially interesting products, and e-marketplaces are supposed to help buyers to search potentially interesting products. The proposed architecture and approaches can inspire existing and initiative e-marketplaces to design their product searching and recommendation mechanisms.

Keywords: multi-agent e-marketplace, argumentation mechanism, Semantic Web, ontology, abstract argumentation framework, defeasible reasoning.
Using Semantic Web Technology to Design Agent-to-Agent 
Argumentation Mechanism in an E-Marketplace

1. Introduction
1.1. Research background and motivation

Persuasive presentation and negotiation are fundamental tasks in a selling process (Oberhaus, Ratliffe, and Stauble, 1993; Anderson, 1995). A salesperson introduces potentially interesting products to the prospect and promotes these products. After that, the salesperson deals with prospect resistance and objections, and arranges the terms of an agreement with the prospect in the negotiation stage. For online selling, many negotiation agents have been researched (Matwin, Szapiro, and Haigh, 1991; Oliver, 1997; Wasfy and Hosni, 1998; Zeng and Sycara, 1998; Lin and Chang, 2001; Dumas et al., 2002; Huang and Lin, 2007). However, how to use agent technologies to facilitate persuasion for online selling is not well addressed yet. Huang and Lin (2007) proposed a sales-agent, called Isa, to handle online persuasion and negotiation dialogues with human buyers. Isa can stand for a seller to persuade a buyer into increasing his/her product evaluation but it only focuses on agent-to-human argumentation. How to design an agent-to-agent argumentation mechanism is needed to be researched for reducing both sellers and buyers’ load and facilitating online selling.

Two obstacles must be broken through for designing an agent-to-agent argumentation mechanism. The first obstacle is how to enable agents in an e-marketplace to understand other agents’ arguments. Semantic Web technologies help Web information to be machine-understandable (Berners-Lee, Hendler, and Lassila, 2001). These technologies enable agents to understand arguments and transcend this obstacle. The second obstacle is how to prove whose arguments are true or false. In a cell phone e-marketplace, for example, a buyer delegates a buyer agent to search good-feature cell phones. The buyer and the sellers in this e-marketplace probably have different definitions of the concept “good-feature cell phone.” Therefore, we need a well-developed argumentation framework to describe relations among arguments and prove their status. Argumentation in a multi-agent context is a process by which one agent attempts to convince another agent of the truth (or falsity) of state of affairs. This process involves agents putting forward arguments for and against propositions, together with justifications for the acceptability of these arguments (Wooldridge, 2002). In an argumentation process, a truth can be defeated when new information appears. Through argumentation, following aforementioned example, a seller agent can persuade a buyer agent to believe its cell phone is good at features even their masters (the seller and buyer) have different definitions of a good-feature cell phone. Dung (1995) developed an argumentation framework for defeasible reasoning. The advantage of this framework is that it pays no special attention on the internal structure of the arguments and therefore this framework can be applied to every domain. The suitability of this framework motivates this research to adjust it to design an agent-to-agent argumentation mechanism in an e-marketplace.

1.2. Research objectives

In existing e-marketplaces, buyers can set conditions and find out products exactly matching these conditions using a search engine. The products those are potentially interesting to the buyer but do not exactly match the conditions cannot be found out. Many chances to deal are missed. This research aims to design a multi-agent e-marketplace, in which buyers and sellers
can delegate their agents to argue over product attributes via an agent-to-agent argumentation mechanism. This research adopts Semantic Web technologies and refers to Dung’s abstract argumentation framework to design this mechanism. This mechanism can help buyers to find out not only exactly interesting but also potentially interesting products. Moreover, it gives sellers a chance to persuade the buyer agents as well as buyers into considering or even buying their products.

2. Related Works

2.1. Semantic Web

Semantic Web inherits some concepts of WWW and adds “meaning” to the Web that enables machine to comprehend semantic documents and data (Berners-Lee, 2001). In fact, people use software agents to search information and deal with some time-consuming or complex tasks is more and more popular, however agents cannot understand all data on the Web like people do. To make agents understand what Web documents mean, in February 2004, World Wide Web Consortium (W3C) released the Resource Description Framework (RDF) and the Web Ontology Language (OWL) as W3C Recommendations for the Semantic Web structure. RDF is used to express information and to exchange knowledge on the Web. OWL is used to publish and share ontologies, which support advanced Web search, software agents and knowledge management (www.w3.org/2001/sw/).

2.1.1. OWL

As mentioned, the most recent development in standard ontology languages is Web Ontology Language (OWL) from W3C. OWL evolves from DAML+OIL that is a combination of OIL and DAML. The Ontology Inference Layer (OIL) is the first ontology language integrating feature from frame-based systems and description logics (DLs), and it is based on RDF and XML to express semantics. The DARPA Agent Markup Language (DAML) is used to develop a language and tools to facilitate the concept of the Semantic Web. For the same purpose, the joining of OIL and DAML bring a powerful language for defining and instantiating Web ontology. W3C slightly revised DAML+OIL to form OWL that builds on RDF and RDF Schema and adds more vocabulary for describing properties and classes (Herman and Hendler, 2006). OWL is developed based on Description Logic which makes it possible for concepts to be defined as well as described. Furthermore, OWL allows the use of a reasoner to check consistency (whether or not one class is possible to have any instances) and subsumption (whether or not one class is a subclass of another class).

There have been many scholars defining what an ontology is, in brief, ontologies are used to capture domain knowledge. An ontology describes the concepts (classes) in a domain and also the relationships (properties) between those concepts. Properties of each concept describe various characteristics and attributes (slots or roles) of the concept, and restrictions (facets or role restrictions) on slots. A knowledge base is composed of an ontology involving a set of individual instances of classes (Natalya and Deborah, 2001).

OWL Ontologies can be categorized into three species according to its expressiveness: OWL-Lite, OWL-DL and OWL-Full. Readers can refer to the OWL Web Ontology Language Overview (Herman and Hendler 2006) for a more detailed synopsis of these species. This research will use OWL-DL to express agents’ ontologies because it supports automatic reasoning based on DLs. DLs are a family of logic formalisms for knowledge representation (Baader, et al. 2002). The DL syntax and corresponding OWL elements are listed in Horrocks, Patet-Schneider,
and van Harmelen (2003). Ontologies using the DLs can be easily described by OWL-DL for the Semantic Web. In addition to OWL, another language called SWRL is needed to specify rules in ontologies.

2.1.2. SWRL

SWRL (Semantic Web Rule Language) is a language to describe rules for the Semantic Web. The SWRL syntax is a combination of OWL and RuleML. RuleML is a XML-based rule language that adopts a kind of standardization and webizing form to present rules (Grosof 2004). SWRL also adopts OWL syntax because RuleML can make structure standardizing but cannot make content do, and rule usage cannot be stipulated either. OWL helps to define vocabulary and attributes used in the rules. We can use common inference engine, such as Jess rule engine (http://herzberg.ca.sandia.gov/jess/), to reason a domain knowledge described by SWRL.

In common with many other rule languages, SWRL rules are written as antecedent-consequent pairs. In SWRL terminology, the antecedent corresponds to the rule body and the consequent corresponds to the rule head. The head and body consist of a conjunction of one or more atoms. SWRL rules reason about OWL individuals, primarily in terms of OWL classes and properties and also can refer explicitly to OWL individuals and support the common same-as and different-from concepts. Similarly, the “differentFrom” atom can be used to express that two OWL individuals are different. Moreover, SWRL has an atom to determine if an individual, property, or variable is of a particular type. The type specified must be an XML Schema data type. Besides, SWRL supports a range of built-in predicates, which greatly expand its expressive power. SWRL built-ins are predicates that accept several arguments. They are described in detail in the SWRL Built-in Specification. The simplest built-ins are comparison operations. All built-ins in SWRL must be preceded by the namespace qualifier “swrlb:”.

2.2. Argumentation Theory

2.2.1. Toulmin Argument Structure

Toulmin Argument Structure gives us a tool for both evaluating and making arguments. The main parts of Toulmin's model are the data, claim, backing, warrant, rebuttal, and qualifier. A data is a fact that describes present situation. A claim is supported by data and by a warrant, which is a general rule or principle supporting the step from data to a claim. The backing is a justification for the warrant, and the rebuttal is a condition where a warrant does not hold. A qualifier expresses the applicability of the warrant (Toulmin, 1958). Figure 1 illustrates an argument based on Toulmin's model.

![Figure 1: Toulmin Argument Structure](image-url)

Toulmin Argument Structure is useful to organize arguments and knowledge but loosely specifies how arguments relate to each other and provides no guidance as to how to evaluate the arguments or prove their statuses.
2.2.2. Abstract Argumentation Framework

An abstract approach to non-monotonic logic developed in several articles by Bondarenko, Dung, Toni and Kowalski. The major innovation of the approach is that it provides a framework and vocabulary for investigating the general features of argumentation systems, and also for non-monotonic logics that are not argument-based. This section presents Dung’s formulation (1995) because in Bondarenko et al. (1997) the basic notion is not for arguments but for a set of what they call “assumptions”. They treat an argument as a set of assumptions.

Dung’s abstract argumentation framework completely abstracts from both the internal structure of an argument and the origin of the set of arguments. The argumentation framework (AF) denoted as \( \text{AF} = \langle \text{AR}, \text{attacks} \rangle \), where \( \text{AR} \) is a set of arguments, and an attack is a binary relation on \( \text{AR} \). Here, an argument is an abstract entity whose role is solely determined by its relations to other arguments. The notation ‘←’ is an attack relation between two arguments. The relation \( \text{arg}_1 \leftarrow \text{arg}_2 \) denotes that \( \text{arg}_1 \) is attacked by \( \text{arg}_2 \). Dung also defined various notions of so-called argument extensions, which are intended to capture various types of defeasible consequence. These notions are declarative, just declaring sets of arguments as having a certain status. The basic formal notions are as follows.

- An argument \( a \) is attacked by a set of arguments \( B \) if \( B \) contains an attacker of \( a \) (not all members of \( B \) need attack \( a \)).
- An argument \( a \) is acceptable with respect to a set of arguments \( C \), if every attacker of \( a \) is attacked by a member of \( C \). For example, if \( a \leftarrow b \) then \( b \leftarrow c \) for some \( c \in C \). In that case we say \( c \) defends \( a \), and also that \( C \) defends \( a \).
- A set of arguments \( S \) is conflict-free if no argument in \( S \) attacks an argument in \( S \).
- A conflict-free set \( S \) of arguments is admissible if each argument in \( S \) is acceptable with respect to \( S \).
- A set of arguments is a preferred extension if it is a \( \subseteq \)-maximal admissible set.
- A conflict-free set of arguments is a stable extension if it attacks every argument outside it.

Dung showed that many existing nonmonotonic logics can be reformulated as instances of the abstract framework.

2.2.3. Defeasible Argumentation Systems

An argumentation framework also needs a ‘proof-theory’ to compute that a particular argument has a certain status. One approach is assigning priority ordering to arguments and an argument with lower priority cannot defeat a higher-priority argument. Vreeswijk and Prakken (2000) proposed a dialectical form of an argumentation game between a proponent and an opponent as a natural form of a proof theory. The initial argument is acceptable if its proponent has a winning strategy; that is, if a proponent can make the opponent run out of moves against his/her any possible counter-arguments. Figure 2 illustrates two argumentation games, where a node means a move. A proponent’s moves are denoted as black nodes and an opponent’s moves are denoted as white nodes. The relation \( \text{P1} \leftarrow \text{O1} \) denotes that \( \text{P1} \) is attacked by \( \text{O1} \). Prakken (2001) defined the disputational status of a dispute move that a move \( M \) of a dispute \( D \) is in if and only if all moves in \( D \) that reply to it are out; otherwise \( M \) is out. The status of a move is in means that the argument of this move is acceptable. We can find that a leaf node in a dialogue tree must be in because it has no attackers. This approach is very easy to calculate the status of each argument. In game (a), for instance, \( \text{P1} \) is acceptable and included in the admissible set \( \{\text{P1, P3, P4}\} \). In game (b), however, \( \text{P1} \) is unacceptable.
3. System Architecture

This research aims to design a multi-agent e-marketplace equipped with an agent-to-agent argumentation mechanism. In this e-marketplace, buyers can delegate their buyer agents to search products matching their needs and sellers can delegate their seller agents to persuade buyer agents to believe their products can match the buyers’ needs. A buyer agent communicates with each seller agent and initiates an argumentation dialogue. This research designs the argumentation mechanism referring to Dung’s argumentation framework and Vreeswijk and Prakken’s dialectical game approach but makes some adjustments. Two assumptions are proposed to make an argumentation dialogue more simple and suitable for e-marketplaces.

Assumption 1: Some beliefs are not changeable in a buyer’s mind. If a seller agent’s claim has a conflict with the buyer agent’s unchangeable beliefs the dialogue is not need to be continued and the buyer agent cannot be persuaded to accept the seller agent’s proposal. This assumption makes sense because persuading a buyer to buy a product that s/he definitely dislikes is not necessary. Buyers’ arguments should have higher priority than sellers’ because purchase decisions are made by buyers.

Assumption 2: Agents’ ontologies are incomplete. Asking buyers or sellers to talk all their beliefs to their agents is a difficulty. Additionally, buyers may not sure their needs. Therefore, an argumentation dialogue in an e-marketplace should be an information gathering process instead of a dispute. In this system, when a seller agent’s argument is conflict with a buyer agent’s changeable beliefs or the buyer agent has no idea whether a seller agent’s argument is true or false the buyer agent will query the seller agent about the reasons for the argument rather than start a sequence of attack actions. If the seller’s reasons are not conflict with the buyer agent’s unchangeable beliefs the buyer agent can be persuaded to accept the seller agent’s proposal.
Figure 3 illustrates the architecture of the buyer and seller agents. Each agent has its own ontology to represent its mental state and shares the e-marketplace ontology. An agent’s mental state ontology describes concepts, relations, and rules about products defined by its master. The e-marketplace ontology defines the common vocabulary used in this e-marketplace and constitutes undefeated rules that are supported by the most buyers and sellers. Ontologies are described in OWL and SWRL formats. Once a dialogue starts, an agent’s argumentation mechanism is responsible for choosing arguments from its ontology to utter and these arguments are formed in Agent Communication Language (ACL) based on the communicative acts specified by Foundation for Intelligent Physical Agents (FIPA). The reasoner and rule engine help the agent to check the consistency between the opposite agent’s arguments and its own mental state ontology.

A product searching process is executed by a buyer agent according to the following steps:

1. Declaring demand: A buyer defines the products s/he needs using an interface without any technical jargon and then these definitions are automatically transformed into SWRL and added into the buyer agent’s ontology. After that, the buyer can send his/her agent to find exactly and potentially interesting products by communicating with seller agents.

2. Find the products that exactly match the buyer’s demand: A buyer agent finds the products that exactly match the buyer’s demand by the following procedure –
   (1) Perform monotonic reasoning on the e-marketplace and the buyer agent’s ontologies to reason out which products are exactly interesting (exactly compliant with the buyer’s definitions about a good product).
   (2) Add these products that exactly match the buyer’s demand into the Option List of Exactly Interesting Products.
   (3) Add the seller agents whose products cannot be prove to be exactly matching into the Talk List.

3. Find potentially matching products: For each seller agent in the Talk List -
   (1) Call for proposal.
   (2) Receive the seller agent’s proposal.
   (3) Request for the claim and its supporting premises about the proposal.
   (4) Receive the seller agent’s claim and its premises.
   (5) Agree this claim and add the proposal into Option List of Potentially Interesting Products if all premises can be prove to be true, otherwise refute this claim and reject the proposal.

4. Updating the buyer agent’s ontology: A dialogue history about each potentially interesting product is shown in the option list. The buyer can check the seller agent’s arguments that the buyer agent cannot disagree and modify his/her beliefs and the buyer agent’s ontology.

\[
\text{prove}(c) \\
\text{Query the reason that supports the claim } c. \\
\text{Receive the reason } R \text{ that supports the claim } c. \\
\]

For each premise \( p \) in \( R \)
- Believe \( p \) is TRUE if \( p \) is consistent with the e-marketplace and buyer agent’s ontologies.
- Believe \( p \) is FALSE if \( p \) is conflict with the e-marketplace or buyer agent’s ontology.
- Otherwise \( p \) is TRUE if prove(\( p \)) return TRUE, or \( p \) is FALSE if prove(\( p \)) return FALSE.

Return TRUE if all \( p \) in \( R \) are TRUE, or return FALSE if one \( p \) in \( R \) is FALSE.

Figure 4. The Algorithm for Proving a Seller Agent’s Claim.
The algorithm for a buyer agent to prove a seller agent’s claim \( c \) is stated in Figure 4. The buyer agent firstly queries the reason that supports the claim (a claim is also a premise supporting former claim). The buyer agent checks each premise after receiving the argument from the seller agent. If a premise cannot be proved true or false according to the buyer agent’s ontology, the additional reasons are continually queried. Finally, the claim is proved true if no premise is false.

A seller agent tries to persuade a buyer agent to recommend the product to the buyer using the following procedure:

1. Declaring supply: A seller defines his/her product using an interface without any technical jargon and then these definitions are automatically transformed into SWRL and added into the seller agent’s ontology.
2. Persuade buyer agents: The seller agent persuade a buyer agent by following steps –
   (1) Propose proposal when receiving a buyer agent's call-for-proposal message.
   (2) Inform the claim and supporting premises about the proposal when the buyer agent request for it.
   (3) Inform the premises of the queried claim.
   (4) Terminate the dialogue when receiving the message of either accept or reject proposal.

Briefly speaking, this research treats an argumentation process in an e-marketplace as an information gathering process in which a buyer agent queries a seller agent about related information. If the information provided by a seller agent is not conflict with the buyer’s unchangeable beliefs the seller agent can persuade the buyer agent otherwise the buyer agent cannot be persuaded. This argumentation process does not depend on internal structure of arguments and the feature of abstract remains.

3.1. Demonstration of Agent-to-Agent Dialogues

This research adopts a cell phone trading marketplace for demonstration. A buyer defines the conditions of a good or a bad cell phone and what conditions are non-negotiable according to his/her beliefs via a template (see Figure 5). These definitions will be automatically transformed into rules and added into the buyer agent’s ontology. A seller also uses a similar template to define his/her own rules and to input product information.

![Figure 5. The Belief Acquisition Template.](image-url)
The following scenario with some cases of dialogue demonstrates how an argumentation proceeds using the argumentation mechanism.

Ariel wants to buy a cell phone and she thinks feature and price are important criteria. She believes if a cell phone has slider type, the cell phone’s feature is good. Ariel’s budget is smaller than NT$ 5000, therefore Ariel does not consider the cell phones with prices higher than NT$ 5000. Besides, she also thinks that a cell phone’s battery time requires at least 250 hrs, otherwise the battery time is not good. As to the cell phone’s brand, she views NBrand and MBrand as good brands.

Ariel’s need can be represented by the following rules in her agent B’s ontology:

```
B: GoodFeatureCellphone(x) ∧ GoodPriceCellphone(x) → GoodCellphone(x)
    hasFeature(x, Slider) → GoodFeatureCellphone(x)
    hasPrice(x, ≤5000) → GoodPriceCellphone(x)
    hasPrice(x, >5000) → BadPriceCellphone(x)
    BadPriceCellphone(x) → BadCellphone(x)
    hasBatteryTime(x, ≥250) → GoodBatteryTimeCellphone(x)
    hasBatteryTime(x, <250) → BadBatteryTimeCellphone(x)
    hasBrand(x, NBrand) → GoodBrandCellphone(x)
    hasBrand(x, MBrand) → GoodBrandCellphone(x)
```

There are three unchangeable beliefs in this ontology, the price of a good price cell phone must be lower than or equal to NT$ 5000, a bad price cell phone cannot be a good cell phone, and a good battery time cell phone must have a battery time that exceeds or equals 250 hours.

**Case 1:**

The seller agent S1 sells the Cell phone 1 and believes it is a good cell phone. A cell phone has good function means it has the functions GPS and Email tool. A cell phone has good battery time means its battery time exceed 300 hours. A cell phone has good brand means its brand is PBrand. A cell phone has good price means its price is not exceed NT$ 4999. A cell phone has good feature means it has bar type feature. A cell phone has good presented date means its presented date is not earlier than 2005/12/10. All these beliefs can be represented as the following rules:

```
S1: GoodBrandCellphone(C001) ∧ GoodPriceCellphone(C001) ∧
    GoodPresentedDateCellphone(C001) ∧ GoodFunctionCellphone(C001) ∧
    GoodBatteryTimeCellphone(C001) ∧ GoodFeatureCellphone(C001) →
    GoodCellphone(C001)
    hasBrand(C001, PBrand) → GoodBrandCellphone(C001)
    hasPrice(C001, ≤4999) → GoodPriceCellphone(C001)
    hasPresentedDate(C001, ≥2005/12/10) → GoodPresentedDateCellphone(C001)
    hasFunction(C001, GPS) ∧ hasFunction(C001, Email tool) →
    GoodFunctionCellphone(C001)
    hasBatteryTime(C001, ≥300) → GoodBatteryTimeCellphone(C001)
    hasFeature(C001, Bartype) → GoodFeatureCellphone(C001)
```
Table 1. The Specification of Cell Phone C001.

<table>
<thead>
<tr>
<th>Model</th>
<th>j44</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brand</td>
<td>PBrand</td>
</tr>
<tr>
<td>Battery Time</td>
<td>300 hrs</td>
</tr>
<tr>
<td>Presented Date</td>
<td>2005/12/10</td>
</tr>
<tr>
<td>Price</td>
<td>NT$ 4999</td>
</tr>
<tr>
<td>Feature</td>
<td>Bartype</td>
</tr>
<tr>
<td>Function</td>
<td>GPS, Email tool</td>
</tr>
</tbody>
</table>

Argumentation between the buyer agent B and the seller agent S1 includes the following sequence of arguments:

B: Please recommend a good cell phone for me.

S1: I think C001 is a good cell phone.

B: Please tell me why.

S3: GoodBrandCellphone(C001) \(\land\) GoodPriceCellphone(C001) \(\land\)

\(\land\) GoodPresentedDateCellphone(C001) \(\land\) GoodFunctionCellphone(C001) \(\land\)

\(\land\) GoodBatteryTimeCellphone(C001) \(\land\) GoodFeatureCellphone(C001) \(\rightarrow\)

GoodCellphone(C001)

B: I agree that the cell phone C001 is a good cell phone.
In case 1, a monotonic reasoning cannot infer that the cell phone C001 is a good cell phone or a bad cell phone, that makes S1 be taken into the Talk List and then an argumentation with S1 is started. In dialogue, agent S1 informs agent B that the cell phone C001 has good brand, price, presented date, function, battery time and feature, which makes S1 believes it is a good cell phone. The buyer agent checks whether the premises of GoodCellphone(C001) can be proved true or false according to its ontology and finds that the premises GoodPriceCellphone(C001) and GoodBatteryTimeCellphone(C001) can be proved true but the premises GoodBrandCellphone(C001), GoodPresentedDateCellphone(C001), GoodFunctionCellphone(C001), and GoodFeatureCellphone(C001) cannot be proved true or false. Therefore the agent B further queries the premises of the claims GoodBrandCellphone(C001), GoodPresentedDateCellphone(C001), GoodFunctionCellphone(C001), and GoodFeatureCellphone(C001). Finally, all premises can be proved true and the buyer agent accept S1’s proposal. The agent S1 persuades agent B into believing the cell phone C001 is a good cell phone and this cell phone can be added into the List of Potentially Good Cell Phones.

Case 2:

The seller agent S2 sells the Cell phone C002 and believes it is a good cell phone. A cell phone has good function means it has the functions GPS and Email tool. A cell phone has good battery time means its battery time exceed 150 hours. A cell phone has good brand means its brand is SBrand. A cell phone has good price means its price is not exceed NT$ 3999. A cell phone has good feature means it has flip feature. A cell phone has good presented date means its presented date is not earlier than 2005/12/10. All the beliefs can be represented as the following rules:

\[
\text{S2: GoodBrandCellphone(C002) } \land \text{ GoodPriceCellphone(C002) } \land \text{ GoodPresentedDateCellphone(C002) } \land \text{ GoodBatteryTimeCellphone(C002) } \land \text{ GoodFeatureCellphone(C002) } \rightarrow \text{ GoodCellphone(C002)}
\]

\[
\text{hasBrand(C002, SBrand) } \rightarrow \text{ GoodBrandCellphone(C002)}
\]

\[
\text{hasPrice(C002, } \leq 3999) \rightarrow \text{ GoodPriceCellphone(C002)}
\]

\[
\text{hasPresentedDate(C002, } \geq 2005/12/10) \rightarrow \text{ GoodPresentedDateCellphone(C002)}
\]

\[
\text{hasFunction(C002, GPS) } \land \text{ hasFunction(C002, Email tool) } \rightarrow \text{ GoodFunctionCellphone(C002)}
\]

\[
\text{hasBatteryTime(C002, } \geq 150) \rightarrow \text{ GoodBatteryTimeCellphone(C002)}
\]

\[
\text{hasFeature(C002, Flip) } \rightarrow \text{ GoodFeatureCellphone(C002)}
\]

Table 2. The Specification of Cell Phone C002.

<table>
<thead>
<tr>
<th>Cell phone C002</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Uu8</td>
</tr>
<tr>
<td>Brand</td>
<td>SBrand</td>
</tr>
<tr>
<td>Battery Time</td>
<td>150 hrs</td>
</tr>
<tr>
<td>Presented Date</td>
<td>2005/12/10</td>
</tr>
<tr>
<td>Price</td>
<td>NT$ 3999</td>
</tr>
<tr>
<td>Feature</td>
<td>Flip</td>
</tr>
<tr>
<td>Function</td>
<td>GPS, Email tool</td>
</tr>
</tbody>
</table>
Argumentation between the buyer agent B and the seller agent S2 includes the following sequence of arguments:

B: Please recommend a good CellPhone for me.

S2: I think Cell phone C002 is a good cell phone.

B: Please tell me why.

S2: GoodBrandCellphone(C002) ∧ GoodPriceCellphone(C002) ∧
    GoodPrensentedDateCellphone(C002) ∧ GoodFunctionCellphone(C002) ∧
    GoodBatteryTimeCellphone(C002) ∧ GoodFeatureCellphone(C002) →
    GoodCellphone(C002)

In case 2, a monotonic reasoning cannot infer that the cell phone C002 is a good or bad cell phone so that the agent S2 is taken into Talk List and then an argumentation dialog starts. Since the claim cell phone C002 is a good-price cell phone can be proved true according to agent B’s ontology the agent B does not query the reason. Finally, agent S2 claims that C002 is a good battery time cell phone because its battery time exceeds 150 hours. This claim is conflict with agent B’s unchangeable belief that a good battery time cell phone must have a battery time exceeding or equaling 250 hours. In this situation, the seller agent’s argument is attacked by the buyer agent’s argument and the buyer agent rejects the proposal to finish this argumentation. The cell phone C002 cannot be added into the List of Potentially Good Cell Phones. The agent S2 cannot persuade agent B into believing the cell phone C002 is a good cell phone.

4. System Evaluation and Results
4.1. Experimental Procedure

This research implemented the e-marketplace using Java programming language, JADE platform (jade.tilab.com) and Jess rule engine (herzberg.ca.sandia.gov) based on the proposed architecture and approach. We conducted a laboratory experiment to evaluate the agent-to-agent argumentation mechanism. We built 50 seller agents to sell cell phones in advance based on the famous cell phone Web sites in Taiwan (www.sogi.com.tw and www.eprice.com.tw). Each seller agent sold one cell phone and these cell phones had no duplication. 36 undergraduate students who majored in Information Management joined this experiment. These subjects had experience of searching and purchasing cell phones and they were willing to buy cell phones in the future. This experiment was held in a computer classroom, every computer was set with related programs and the executable environment in advance. In the experiment, an instructor firstly told the subjects the experimental purpose, procedure, and a cover story that let them put himself/herself in the scenario of buying a cell phone in a multi-agent e-marketplace. In the next phase, each subject logon the system and then defined his/her need via a belief acquisition interface. Then, each subject can delegate his/her buyer agent to communicate with the seller...
agents to search appropriate products that exactly or potentially match his/her need. After the buyer agent communicated with all seller agents in Talk List, a list of exactly good cell phones and a list of potentially good cell phones were recommended to the buyer. The former list included the items exactly matching the buyer’s need and these items were searched by a monotonic reasoning. The later list included the potentially interesting items that searched by argumentations (non-monotonic reasoning). The subjects were asked to give a rating about how interested they feel for every exactly and potentially matching item using a 7-point Likert scale and the score ranges from -3 to 3. In this way, this experiment can assess if the argumentation mechanism is able to recommend potentially interesting items effectively. Besides, the system automatically recorded the time cost and dialogue history during each argumentation dialogue. In the end of experiment, the subjects were asked to fill up a short questionnaire for acquiring their feedbacks, including the user backgrounds, their feelings about system use, comprehensions of the argumentation process, and the satisfactions of the lists of exactly good cell phones and potentially good cell phones. Each feedback to a question is measured by a 7-point Likert scale and the score ranges from -3 to 3.

When a subject used the system, s/he chose the import condition that a good cell phone must have using the condition setting panel. After choosing, the condition definition panel appeared to make the subject define his/her detailed demands. For instance, s/he thought a good cell phone must have a good price and a good feature, and then s/he defined that a good price is less than NT$ 5000 and viewed the good-price attribute as a non-negotiable attribute, that is s/he did not consider a cell phone with a price higher than NT$ 5000. Therefore, s/he checked the box of non-negotiable attribute and the frame of the condition will turn to red. S/he also thought a good feature means the feature of a cell phone is slider. These two definitions were conducive to produce the rules about what is a good cell phone. As to the conditions the subject did not choose in the first panel, those will show in the other condition definition panel. The subject can determine whether s/he wanted to use the panel to define his/her demands for the rest of the conditions. S/he also can set the non-negotiable attribute for each condition. The list panel will present a tree of conditions that the subject had defined.

The subject made the forms out step by step and submitted the information, then it is stored in SWRL format in an OWL file and the Jess rule engine is started for proceeding monotonic inferring. Jess can infer the individuals of the good cell phones and the bad cell phones. The seller agents whose cell phones cannot be inferred good or bad were added into the Talk List. In the List of Exactly Good Cell Phones, an exactly good cell phone will be shown. The product photo, number, brand, model, battery time, presented date, feature, price and function are also on the table. The first column of the table is the score column designed to get the subject’s feedback about how interested the subject feel about the product. After the subject scored each product, the buyer agent began to communicate with seller agents. The target agents the buyer agent wanted to communicate with were the seller agents in the Talk List and their arguments were generated based on their ontologies and the algorithm described in Section 3. Finally, List of Potentially Good Cell Phones including the cell phones their seller agents made successful persuasions. The table in the list is added two columns, the reason to accept the proposal and the detailed dialogue content of each product. In the reason column which recorded the defeated arguments of the buyer agent. The dialogue column can help the subject to see the complete argumentation history.

4.2. Results of the Experiment

Figure 6 illustrates the subjects’ profiles.
Figure 6. Subjects’ Profiles.

The average number of potentially good items recommended by each buyer agent is 23, the average number of dialogues is 24, and the average number of messages in a dialogue is 11. There are 14 buyer agents found none of item that exactly match the buyer’s need and 6 buyer agents found none of item that potentially match the buyer’s need. Additionally, the average time cost in a dialogue is shorter than 1 second.

We measure a buyer’s interest in a list of recommended items by averaging scores of items in the list. There were 22 subjects received nonempty list of exactly good items. Their average score of interest in the list is 0.636 and the standard deviation is 0.889. 30 subjects got nonempty list of potentially good items. Their average score of interest in the list is 0.711 and the standard deviation is 0.414. There were 19 subjects whose lists of exactly and potentially good items were not empty. This research uses pair-sample $t$ test to compare the 19 subjects’ interests in the two lists. The result is depicted in Table 3 and shows that there is no significant difference between the interest in the list of exactly good items and the interest in the list of potentially good items. The average interest in the list of potentially good items is positive and is not lower than the average interest in the list of exactly good items that means the argumentation mechanism is able to find out potentially interesting items for buyers.

Table 3. Interests in the Lists of Exactly and Potentially Good Items.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>SD</th>
<th>$t$-value ($p$-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interest in the list of exactly good items.</td>
<td>0.519</td>
<td>0.913</td>
<td>-1.113 (0.280)</td>
</tr>
<tr>
<td>Interest in the list of potentially good items.</td>
<td>0.786</td>
<td>0.333</td>
<td></td>
</tr>
</tbody>
</table>

Table 4 reveals average scores of the questions in the questionnaire. We can find that subjects had positive attitudes toward the system and agreed that this system can help them to search potentially interesting products.

Table 4. The Average Scores of the Questions.

<table>
<thead>
<tr>
<th>Question</th>
<th>Average score</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Do you feel the system is easy to use?</td>
<td>1.583333</td>
<td>0.953794</td>
</tr>
<tr>
<td>2. Can you understand the system manipulation process?</td>
<td>1.805556</td>
<td>0.966651</td>
</tr>
<tr>
<td>3. Do you feel the system can help you to search interesting products?</td>
<td>1.527778</td>
<td>0.83287</td>
</tr>
<tr>
<td>4. Are you satisfied with the recommended items in the list of exactly good cell phones?</td>
<td>1.083333</td>
<td>1.037492</td>
</tr>
<tr>
<td>5. Are you satisfied with the recommended items in the list of potentially good cell phones?</td>
<td>1.083333</td>
<td>0.829156</td>
</tr>
<tr>
<td>6. Can you understand the dialogue contents provided in the list of potentially good cell phones?</td>
<td>1.027778</td>
<td>0.86558</td>
</tr>
<tr>
<td>7. Do you feel the dialogue contents can help you to understand why the agent recommends these items to you?</td>
<td>0.944444</td>
<td>0.664348</td>
</tr>
<tr>
<td>8. Do you agree that an e-store should help you to search not only the exactly interesting products but also potentially interesting products?</td>
<td>1.861111</td>
<td>1.031525</td>
</tr>
<tr>
<td>9. Do you agree that this system can help you to search potentially interesting products?</td>
<td>1.666667</td>
<td>0.881917</td>
</tr>
</tbody>
</table>
We further compare the 19 subjects’ satisfactions with the two lists. The result is depicted in Table 5 and shows that the satisfaction with the list of exactly good items and the satisfaction with the list of potentially good items are identical and positive. The potentially interesting items searched by the argumentation mechanism can satisfy buyers. We also find that even the items that exactly match the conditions set by the buyers cannot fully satisfy the buyers. The possible reason is that buyers usually cannot fully know their needs or cannot fully understand the products they search for. Therefore, e-marketplaces should help buyers search not only exactly but also potentially interesting items.

Table 5. Satisfactions with the Lists of Exactly and Potentially Good Items.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>SD</th>
<th>t-value (p-value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satisfaction with the list of exactly good items.</td>
<td>1.160</td>
<td>0.834</td>
<td>0.000 (1.000)</td>
</tr>
<tr>
<td>Satisfaction with the list of potentially good items.</td>
<td>1.160</td>
<td>0.834</td>
<td></td>
</tr>
</tbody>
</table>

5. Conclusions

This research designs a multi-agent e-marketplace with an agent-to-agent argumentation mechanism. Using this mechanism, buyers can find out potentially interesting products through their agents. Moreover, sellers can delegate their agents to make buyer agents change beliefs and recommend their products to the buyers. To make agent-to-agent argumentation possible, this research adopts OWL and SWRL to clearly express agents’ ontologies and uses an abstract argumentation framework with information gathering approach to support defeasible reasoning. A prototype system based on the proposed architecture and approaches was developed for trading cell phones and a laboratory experiment was conducted to evaluate it. The experimental results show that the proposed system is able to help buyers to search not only exactly but also potentially interesting products, and e-marketplaces are supposed to help buyers to search potentially interesting products.

This research indicates two innovation directions for electronic commerce. First, argumentation mechanism is useful for online matchmaking and recommending potentially interesting items. How to acquire users’ beliefs easily and how to present dialogue history comprehensibly are also important. Therefore, more user-friendly argumentation-based agents for searching various products should be developed. Second, the Semantic Web technology is getting mature to express complex rules and information. Developing smarter agents using Semantic Web technology is worthy to be researched. We believe that the proposed architecture and approaches can help existing and initiative e-marketplaces to design their argumentation mechanisms and facilitate the evolution of modern applications for electronic commerce.

6. References


