# A MODEL OF AGENT ONTOLOGIES FOR B2C E-COMMERCE

Domenico Rosaci
*DIMET Department - University of Reggio Calabria*
*Via Graziella - Loc. Feo di Vito 89060 Reggio Calabria (Italy)*

Keywords: Intelligent Agents, B2C E-Commerce, Ontologies

Abstract: This paper proposes a formal model of agent ontologies, suitable to represent the realities of both customers and sellers in a B2C electronic commerce scenario. This model is capable of describing the entities involved in the above realities (products, product features, product categories) as well as the behaviour of customers and sellers in performing their activities. A system architecture, based on the presented ontology model, is also briefly described.

## 1 INTRODUCTION

In the last few years, many agent-based systems for supporting business-to-customer (B2C) e-commerce activities have been proposed. In this context, agents can be seen as *mediators* between the actual involved subjects, i.e. customers and businesses. Traditional marketing research has developed many descriptive theories and models that attempt to capture the Consumer Buyer Behaviour (CBB) and that differ in various aspects; however, we can highlight the following five relevant steps:

1. *Need Identification*. In this stage the consumer is stimulated to become aware of some unmet need. For instance, consider the case of a customer interested in a certain category of books. Agents can continuously monitor the Web and advert the customer when a new book of that category is available.
2. *Product Brokering*. Once a consumer has identified a need to satisfy, he has to find *what* to buy through a careful evaluation of the various products possibly satisfying that need. This requires a comparison of product alternatives based on some consumer-provided criteria. At the end of this step, a set of products, usually called the *consideration set*, capable of satisfying the consumer desires, has been identified.
3. *Buyer Formation Coalition*. The customers, after having chosen the product to buy in the product brokering stage, before choosing the most suitable merchant in the merchant brokering stage, may interact with other (similar) buyers to form a *buyer coalition*, in order to approach the merchant with a large order and thus obtain a leverage.
4. *Merchant Brokering*. In this step, the consideration set is combined with merchant-specific alternatives based on consumer-selected criteria (e.g, availability, price, delivery time, warranty, reputation, etc.) for helping the consumer to determine *whom* to buy from.
5. *Negotiation*. During this step, the various terms of the transaction as, for instance, the price, are determined. The benefit of negotiating the price of a product, instead of fixing it, is that negotiation relieves the merchant from needing to determine the value of the good a priori (Maes et al., 1999). Rather, the price is dynamically determined by the marketplace.

Several agent-based systems have been proposed for mediating the above activities. For a detailed survey of these systems, see (He et al., 2003). However, none of the existing approaches supports in a unified manner all the described stages, allowing customers and sellers to interact in a virtual marketplace by using a unique, integrated, framework. This paper aims to propose such a unified approach, by defining a formal ontology model which both the customer and seller realities are represented in, and that can be fruitfully exploited in all the CBB stages. Such an



Rosaci D. (2004).
A MODEL OF AGENT ONTOLOGIES FOR B2C E-COMMERCE.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 3-9
DOI: 10.5220/0002619000030009

ontology provides a flexible way to access customer and seller information, since it contains both an intensional representation of the above realities (i.e., a meta-schema), as well as an extensional representation that allows, when it is necessary, to handle the actual data. Furthermore, the ontology represents customer and seller behaviour (e.g., negotiation strategy) by using logic propositions able of dynamically activating some pre-defined agent's actions.

The paper is structured as follows: in the Section 2, we deal with some related work; in the Section 3, we give a formal description of our ontology model. As an example of application of this model, in the Section 4, an architecture of a system supporting a virtual marketplace is also briefly described. Finally, in the Section 5, we draw our conclusions.

## 2 RELATED WORK

The necessity of representing, in the profile of a customer, not only concepts of his interest but also his behaviour in accessing those concepts, has been considered in several works in the Information Systems field as, for instance, in (Bergamaschi et al., 2001; Terracina and Ursino, 2000; Buccafurri et al., 2002a). As far as the e-commerce research is concerned, MOMIS (Mediator envirOnment for Multiple Information Sources) (Bergamaschi et al., 2001) handles both integration and querying of multiple, heterogeneous information sources, storing both structured and semi-structured data. Data source integration is carried out by following a semantic approach based on Description Logics, clustering and a common data model capable of representing all involved data sources. In (Buccafurri et al., 2002b; Rosaci et al., 2002), a multi-agent system for representing and handling e-commerce activities is proposed. In such an approach, an agent is present in each e-commerce site, handling the information stored therein. Furthermore, an agent is associated with each customer, handling his profile. The information associated with both sites and customer profiles is represented and handled using a particular conceptual model called *B-SDR network*. This latter allows to uniformly manage heterogeneous data sources and to construct and maintain a profile storing information about the visits the customer carries out to the various e-commerce sites. MORPHEUS (Yang et al., 2001) is a comparison-shopping agent that automatically collects product descriptions from a group of on-line stores on user's behalf. Since the Web stores are heterogeneous, a wrapper must be built and maintained for each store. Ontologies, i.e. intensional descriptions of product characteristics and customer and seller behaviour, have been already exploited in B2C e-commerce context (Omelayenko, 2001) and also many ontology-based approaches have been proposed in multi-agent systems field (OAS 2002, 2002).

All the previously described approaches try to solve the heterogeneity by adopting techniques based on the integration of the various sources or on the use of wrappers. They are very interesting from the viewpoint of a user that want to consider the whole Web for searching goods of his interest. However, the price to pay for obtaining the integration of a potentially overwhelming amount of Web sources is often high. Indeed, integration techniques are onerous in terms of time to spend for constructing the integrated global representation of the various sources. Moreover, such a global representation is often very difficult to handle, since it has large dimensions and needs a continuous pruning of the less important concepts. On the contrary, the approach we propose in this paper is based on the construction of virtual marketplaces whose actors (sellers and customers) are represented in a uniform manner, due to the use of agents that operate as a personal assistant and translate the user interests and preferences in a pre-defined standard, represented by the agent ontology model that we here introduce.

## 3 AGENT ONTOLOGIES

The ontology of an agent is a representation of the reality of the agent's owner. In a B2C e-commerce context, such a owner can be either a customer or a seller. In the first case, the agent ontology has to store the customer interests and preferences w.r.t. the virtual marketplace and the customer behaviour in purchasing goods (e.g., the way of visiting e-commerce sites, the strategies in negotiating, etc.). In the second case, the agent ontology has to represent the product categories of the seller, the product characteristics (price, availability, etc.) and the service features (warranties, time delivery, etc.).

In this section, we propose a formal definition of an agent ontology. In order to better explain the various concepts we introduce below, we propose a simple situation of a customer and a seller that interact in a virtual marketplace. This situation will serve as a leading example along this paper.

**Example 1** Let $John$ be a customer interested in purchasing $books$ and $CDs$. His interest in books mainly concerns with narrative and poetry. In the past, he purchased on the Internet the books *Anna Karenina*, *The Buddenbrok*, *Les Fleurs du mal* and *La Divina Commedia*. He also purchased the CD *The Ghost of Tom Joad*. In the case of a non negotiable price, $John$ usually behaves as follows: (i) when he purchases a book, he also purchases a CD; (ii) moreover, he considers a book (resp., a CD) as an **interesting book** (resp., an **interesting CD**) if the **price** is smaller than 20 US$ and the **delivery time** is smaller than 3 days. In the case of a negotiable price, suppose $John$ negotiates with a seller for a good that has a base **proposed price** equal to $p(0)$ at the step 0 of the negotiation. The $John$'s behaviour is as follows. At the step 0, $John$ makes an **offer** $o(0) = 0.8 * p(0)$. At the step $i$ of the negotiation ($i = 1, 2...$), in response to a new proposed price $p(i)$ of the seller, $John$ offers a value $o(i) = (p(i) + o(i-1))/2$.

Let $Word\&Music$ a seller of books and CDs. It deals with three categories of books, namely *narrative*,*essay* and *poetry*. This seller, in a bilateral negotiation with a customer, behaves as follows. Let $p(0)$ be the base **proposed price** of a book, that it proposes at the step 0 of a negotiation with a customer. Suppose it receives an **offer** $o(i)$, at the step $i$ of the negotiation ($i = 0, 1, ....$), by the customer. If $o(i)$ is smaller than $0.7 * p(i)$, the seller aborts the negotiation. Otherwise, it proposes a new price equal to $p(i+1) = (o(i) + p(i))/2$.

From this simple situation, we can observe that different kinds of knowledge need to be represented in the ontologies of the customer and the seller. Namely:

- **Entities**. In both the cases of the customer and the seller in the example above, the agent ontology has to represent some products relating to the agent's owner. Each product must have an *ID* that identifies it, and a set of associated *features* that gives some information about it. For instance, a book may have a feature *title*, a feature *author*, a feature *delivery_time* and a feature *price*. Such a representation of books is an intensional entity (a metadata) and each actual book can be viewed as an instance of this entity (a data). For example, the book *Anna Karenina* may be represented by an entity instance with ID=1, *title* =*"Anna Karenina"*, *author* =*"Tolstoj"*, *delivery time* =2 and *price* =17. More formally:

  **Definition 1** Let $T$ be a set {String,Integer,Real...} of data types. We define the *feature set*, denoted by $F$, as a set of variable names, each variable having a type that belongs to $T$. A *feature* is an element $f \in F$. Let $D$ be a function from $F$ to $T$, that associates to each feature $f \in F$ its data type $D(f) \in T$. An *instance* of the feature $f$ is a value $v \in D(f)$.

  **Definition 2** Let $f, g \in F$ be two features with $D(f) = D(g)$ and let $k \in D(f)$ be a constant. With the notation $f = g$ we mean that the value of $f$ is set equal to the value of $g$, and with the notation $f = k$ we mean that the value $f$ is set equal to the value $k$.

  This definition can be applied, besides to product characteristics, also to each object belonging to the customer's (resp., the seller's) world as, for instance, **proposed price** and **offer** that may be represented by the features $proposed\_price$ and $offer$, respectively. In sum, features can be exploited both in the definition of products than in the definition of more general entities. For instance, an entity $negotiation$ may represent the behaviour of both $John$ and $Word\&Music$ in negotiating, and $proposed\_price$ and $offer$ may be the features of this entity. We give below a formal definition of entity.

  **Definition 3** The *entity domain* $\mathcal{E}$ is the set of all the tuples $\langle ID, f_1, f_2, ..., f_n \rangle$, where $ID$ is an Integer variable and $f_1, f_2, ..., f_n \in F$. An *entity* is an element of $\mathcal{E}$.

  **Definition 4** Let $e = \langle ID, f_1, f_2, ..., f_n \rangle$ be an entity. An *instance* of $e$ is a tuple $i = \langle idv, fv_1, fv_2, ..., fv_n \rangle$, where $idv \in Integer$ and $fv_1 \in D(f_1), fv_2 \in D(f_2), ..., fv_n \in D(f_n)$.

- **Categories**. A category is a collection of entities. For example, in the ontology of the seller $Word\&Music$, the category $BOOKS$ may group all the books available to be purchased. Moreover, a category can be organized in some subcategories. For instance, in our leading example, the category $BOOKS$ of $John$ contains the subcategories $POETRY$ and $NARRATIVE$. Thus, generally, if we consider a set of entities as a limit-case of category (i.e., a category that does not contains any sub-categories), we can inductively say that a category is either a set of entities or a set of sub-categories. More formally:

  **Definition 5** The *category domain* $\mathcal{C}$ is the set of all the tuples
  $\langle ID, e_1, e_2, ..., e_n \rangle$, where $ID$ is an Integer variable and $e_1, e_2, ..., e_n \in \mathcal{E}$, and of all the tuples $\langle ID, c_1, c_2, ..., c_n \rangle$, where $ID$ is an integer and $c1, c2, ..., c_n \in \mathcal{C}$. A *category* is an element of $\mathcal{C}$.

  In the Figure 1, all the categories, instances and features involved in the $John's$ reality are described. Note that we have inserted the entity $Negotiation$ in a particular category $BEHAVIOUR$.

  Categories are intensional information. An instance of a category is a set of instances of all the entities belonging to it. We can inductively define a category instance as follows:

  **Definition 6** Let $c = \langle ID, e_1, e_2, ..., e_n \rangle$ be a category, where $ID$ is an Integer variable and $e_1, e_2, ..., e_n \in \mathcal{E}$. An instance of $c$ is a tuple $i = \langle idv, ei_1, ei_2, ..., ei_n \rangle$, where $idv \in Integer$ and $ei_1, ei_2, ..., ei_n \in Integer$ are identifiers of entity instances.

**JOHN**=$\langle ID, \mathbf{BOOKS}, \mathbf{CDS}, \mathbf{BEHAVIOUR}\rangle$;
**BOOKS**=$\langle ID, \mathbf{NARRATIVE}, \mathbf{POETRY}\rangle$;
**NARRATIVE**=$\langle ID, \mathbf{Book}, \mathbf{Book}\rangle$;
**POETRY**=$\langle ID, \mathbf{Book}, \mathbf{Book}\rangle$;
**CDS**=$\langle ID, \mathbf{Cd}\rangle$;
**BEHAVIOUR**=$\langle ID, \mathbf{Negotiation}\rangle$;
**Negotiation**=$\langle ID, \mathbf{proposed\_price}, \mathbf{offer}\rangle$;
**Book**=$\langle ID, \mathbf{title}, \mathbf{author}, \mathbf{delivery\_time}, \mathbf{price}\rangle$;
**Cd**=$\langle ID, \mathbf{title}, \mathbf{author}, \mathbf{delivery\_time}, \mathbf{price}\rangle$;
**title, author**=String;

**delivery_time, price, proposed_price,offer**=Integer;

Figure 1: The categories, entities and features of the customer $John$

Let $c = \langle ID, c_1, c_2, ..., c_n\rangle$, where $ID$ is an integer and $c1, c2, ..., c_n \in \mathcal{C}$. An instance of $c$ is a tuple $i = \langle idv, ci_1, ci_2, ..., ci_n\rangle$, where $idv \in Integer$ and $ci_1, ci_2, ..., ci_n \in Integer$ are identifiers of category instances.

In the Figure 2, all the category instances involved in the $John's$ reality are described.

This figure represents the following information:

The category instance JOHN1 (representing all the interests of the customer $John$) has ID=5 and contains the category instance with ID=4 (i.e., BOOKS1), the category instance with ID=1 (i.e., CDS1) and the category instance with ID=6 (i.e., BEHAVIOUR1). In its turn, the category instance BOOKS1 has ID=4 and contains the category instance with ID=3 (i.e., NARRATIVE1) and the category instance with ID=2 (i.e., POETRY1). The category instance CDS1 has ID=1 and does not contain any category instances since, as specified by the schema of its category CDS, it is composed by an entity instance, namely that one having ID=5 (i.e., Cd1). Similarly, the category instance BEHAVIOUR1 has ID=6 and is composed by the entity instance with ID=6 (i.e., Negotiation1). The category instance NARRATIVE1 has ID=3 and is composed by the entity instance with ID=1 (i.e, Book1) and by the entity instance with ID=2 (i.e., Book2). The category instance POETRY1 has ID=2 and is composed by the entity instance with ID=3 (i.e, Book3) and by the entity instance with ID=4 (i.e., Book4).

The entity instances with IDs=1..4 represent books. The entity instance with ID=5 represents a CD and the entity instance with ID=6 represents a step of a negotiation.

- **Knowledge patterns**. An ontology has to store, besides information about the involved entities, also other information relative to the behaviour of the customer (resp.,the seller) in purchasing (resp., in selling) the products. In our leading example, the negotiation behaviour of the customer $John$ and the seller $Word\&Music$ are some examples of this kind of knowledge that have to be represented.

Such a knowledge can be stored into an ontology by considering a set of *events*, that represents actions belonging to the activity of the customer (resp., the seller), that may happen or not. For examples, in the leading example's situations, the concepts of **interesting book, interestingCD, buy a book, buy a CD, make an offer, propose a new price** can be modeled as events and represented by boolean variables as $interestingBook$, $interestingCD$, $buyBook$, $buyCD$, $makeO$, $proposeP$, respectively. Moreover, also a relational expression involving features is an event as, for instance, $(deliveryTime < t)$ or $(price < p)$. An event can be thus represented by a boolean expression.

**Definition 7** An *event* is either: (*i* ) a boolean variable or (*ii* ) an expression of the form $a\theta b$, where $a, b \in F$, $D(a) = D(b)$ and $\theta \in \{=, <, \leq, >, \geq\}$ is a relational operator or (*iii* ) an expression of the form $a\theta c$, where $a \in F$ and $c \in D(a)$ and $\theta$ is a relational operator. The events of the type (*ii* ) and (*iii* ), that involve only features and do not involve any boolean variable, are called *feature-events*.

The existing relationships between events can be represented by *propositional rules* as, for instance, relative to the situations described in our leading example for the $John$ customer:

$$k_1 : buyBook \Rightarrow buyCD$$

$$k_2 : (price < 20), (time < 3) \Rightarrow interestingBook$$

$$k_3 : (price < 20), (time < 3) \Rightarrow interestingCD$$

$$k_4 : interestingBook, proposeP \Rightarrow makeO$$

or, for the $Word\&Music$ seller:

$$k_5 : makeO \Rightarrow proposeP$$

**JOHN1**=$\langle 5, 4, 1, 6\rangle$;
**BOOKS1**=$\langle 4, 3, 2\rangle$;
**NARRATIVE1**=$\langle 3, 1, 2\rangle$;
**POETRY1**=$\langle 2, 3, 4\rangle$;
**CDS1**=$\langle 1, 5\rangle$;
**BEHAVIOUR1**=$\langle 6, 6\rangle$;
**Negotiation1**=$\langle 6, 5, 4\rangle$;
**Book1**=$\langle 1, “AnnaKarenina'', “Tolstoj'', 17, 2\rangle$;
**Book2**=$\langle 2, “TheBuddenbrook'', “Mann'', 18, 2\rangle$;
**Book3**=$\langle 3, “Lesfleursdumal'', “Beaudelaire'', 15, 2\rangle$;
**Book4**=$\langle 4, “LaDivinaCommedia'', “Alighieri'', 25, 2\rangle$;
**Cd1**=$\langle 5, “TheGhostofTomJoad'', “Springsteen'', 18, 2\rangle$;

Figure 2: The category instances and the entity instances of the customer $John$

These rules, that we call *knowledge patterns*, affirm that some events happen when other related events happen. For instance, the rule $k_1$ affirms that if the event $buyBook$ has the value $true$, also the event $BuyCD$ has the value $true$.

More formally:

**Definition 8** A *knowledge pattern* $k$ is a propositional rule of the form $a_1, a_2, ..., a_n, \bar{a}_{n+1}, \bar{a}_{n+2}, ..., \bar{a}_m \Rightarrow b$ , where $a_1$, $a_2$, $a_n$, $a_{n+1}$, $a_{n+2}$, ..., $a_m$, $b$ are events. This notation means that, if both $a_1, a_2, ..., a_n$ assume at the same time the value $true$, and $\bar{a}_{n+1}, \bar{a}_{n+2}, ..., \bar{a}_m$ assume at the same time the value $false$, then $b$ assume the value $true$. Let $fs = \{f_1, f_2, ..., f_o\}$ be a set of features. We say that $k$ is a knowledge pattern on $fs$, if each feature of $fs$ appears in almost one of the events $a_1, a_2, ..., a_m$ contained in $k$, and all the events $a_1, a_2, ...a_n$ are feature-events.

- **Actions**. Often, when an event happens in the world of a customer or a seller, an action is consequently produced. For instance, in our leading example relative to the customer $John$, when $John$ decides to make an offer in a negotiation phase, the value of the offer is equal to the mean between his previous offer and the price proposed at the present by the seller. We can thus say that, when the event $makeO$ becomes true, a program, that we denote by $of$, is called that sets the value of the feature $offer$ equal to $(proposed\_price + offer)/2$, where $proposed\_price$ is the feature representing the price proposed by the seller. Similarly, in the case of the $Word\&Music$ seller, we can say that, when the event $proposeP$ becomes true, a program $pf$ is activated that behaves as follows: if $offer < 0.7 * proposed\_price$, it sets the event $end$ to the value $true$ and then terminates, otherwise it sets the value of $proposed\_price$ equal to $(proposed\_price + offer)/2$. In this latter case, we observe that the program modifies both a feature value and an event value.

  Whe call *action* a 5-tuple composed by an event, as $makeO$, a program, as $co$, that is activated by the event, a set of features, as $\{proposed\_price\}$, that are the arguments of the program, and another two sets of features and events, as $\{offer\}$ and $\{end\}$, respectively, whose value is modified by the program. More formally:

  **Definition 9** An *action* is a tuple $\langle e, P, fs1, es, fs2\rangle$ where $e \in \mathcal{E}$, $P$ is a program, $fs1, fs2 \in 2^F$, $es \in 2^{\mathcal{E}}$, such that the program $P$ is activated if $e = true$ by passing it as input arguments the features belonging to the set $fs1$ and $P$ modifies both the value of the features belonging to the set $fs2$ and the events belonging to the set $es$.

  In our leading example, we can define the actions
  $a_{John} = \langle makeO, of, \{proposed\_price\}, \{\},$
  $, \{offer\}\rangle$
  $a_{Word\&Music} = \langle proposeP, pf, \{offer\}, \{end\},$
  $, \{proposed\_price\}\rangle$.

Now, we can give our definition of agent ontology. This is a collection of four sets: a set of entities, representing all the products which the customer is interested in (resp., which are in the seller's catalog) and all the other entities belonging to the customer (resp., the seller) activities; a set of categories, describing the hierarchical structure of the entities; a set of knowledge patterns, describing the rules followed by the customer in purchasing (resp., followed by the seller in selling) products and a set of actions, specifying what actions the customer (resp., the seller) performs when the environment changes. Since all these sets contain only intensional information, such an ontology is purely intensional and can be viewed as a schema. We also define the *instance* of an ontology, that contains, for each entity (resp., category) of the ontology, also a set of instances of that entity (resp., category). More formally:

**Definition 10** An *agent ontology schema* is a 4-tuple $\langle E, C, K, A\rangle$, where: (*i*) $E$ is a set of entities; (*ii*) $C$ is a set of categories; (*iii*) $K$ is a set of knowledge patterns; (*iv*) $A$ is a set of actions.

**Definition 11** An *ontology instance* of an ontology $O$=$\langle E, C, K, A\rangle$ is a 4-tuple $\langle EP, CP, K, A\rangle$, where:

(*i*) $EP$ is a set of of pairs $(e, ei)$, such that $e \in E$ is an entity and $ei$ is a set of instances of $e$; (*ii*) $CP$ is a set of of pairs $(c, ci)$, where $c \in C$ is a category and $ci$ is a set of instances of $c$; (*iii*) $K$ is a set of knowledge patterns; (*iv*) $A$ is a set of actions.

As an example, the ontology of the customer $John$ is shown in the Figure 3. An instance of this ontology is shown in the Figure 4.

# 4 THE OBA-B2C SYSTEM'S ARCHITECTURE

The OBA-B2C system allows to realize a virtual marketplace. It is composed by a central unit, called *agency*, and several agents connected to the agency. There are two agent typologies in the system, namely the *customer agents* and the *seller agents*. On the one hand, each customer agent is associated with a real customer, and operates on his behalf in order to automatically carry out the various stages of the e-commerce. On the other hand, each seller agent is associated to a real seller and performs e-commerce activities on the seller's behalf.

Each agent has to be registered on the agency in order to be enabled to participate to the virtual marketplace. The agency is a Web site that provides several services to the actors of the virtual marketplace. The registration of an agent consists in the assignment of two access codes (an agent name and a password) to the agent. Each registered agent can enter into the marketplace by accessing the agency site and authenticating itself by means of the above codes. The agency maintains a list of all the registered agents, and is able to provide information about them. By means of the agency, an agent of the virtual marketplace can find those agents that have similar interests, in order to form a coalition of buyers. It is worth pointing out that the agency does not make public the information about the registered agents, in order to protect customer privacy. Thus, the agency behaves as a mediator that supports the agent communication. As an example, an agent $a$ that desires to know the agents that have similar preferences, send to the agency those information about itself (personal preferences, buying behaviour, etc.) it wants to make public. Automatically, the agency contacts all the agents of the marketplace, sends them the above information, receives the answers by the contacted agents and finally transmits these answers to $a$. Agency also mediates virtual auctions, by allowing the seller agents of the marketplace to create auctions for selling goods and the customer agents to bid offers for desired goods.

In the client-agent architecture of the OBA-B2C system, the customer agent is a client that a real customer exploits for participating to the virtual marketplace. This agent (see Figure 5) is composed by two knowledge bases called *ontology* and *address book*, and by two main programs, called *ontology manager* and *communication manager*, respectively. The core of the agent architecture is the *ontology*, that has the structure defined in the Section 3. The ontology is handled by the ontology manager, that exploits the information contained in it for supporting the various stages of the e-commerce. For this reason, the ontology manager is composed by five program modules that mediate, respectively, need identification, product brokering, buyer coalition formation, merchant brokering and negotiation. Each module performs, on the behalf of the customer, the associated activity. For instance, the merchant brokering manager exploits the information contained in the customer ontology relative to the products of interest, and finds in the virtual marketplace the best merchants for those products. Each program module also monitors the customer behaviour in the associated activity. All the program modules of the ontology manager need to contact the agents of the marketplace for performing their tasks. For instance, the buyer coalition formation has to contact all the agents of the marketplace in order to find possible collaborators. The communication among the agents is handled by a component called *communicator manager* . This module exploits a database of agent addresses, called *address book*, for performing its tasks.

The seller agent is a client software that a real seller can exploit for participating in the virtual marketplace. This agent is composed by an *ontology*, an *address book* and tree main programs, called *ontology manager*, *site builder* and *communication manager*, respectively. Similarly to the customer agent, the core of the seller agent architecture is the ontology that, in this case, is not built by observing the owner behaviour but, obviously, the seller itself is able to build it by specifying the categories of interests and the products to sell. This activity is performed by a dedicated component called *site builder*. The site builder operates as a seller assistant, allowing the seller to build his Web store by using a uniform representation w.r.t. the virtual marketplace. For instance, if the seller tries to insert in his ontology a product (resp., a product category) that has the same characteristics of a product (resp., a product category) always present in the ontology of another seller of the marketplace, the site builder identifies the product(resp., the product category) in a unique manner, by assigning it a unique identifier, and thus avoiding heterogeneous representations of the same good.

The ontology manager and the communication manager have the same role than in the customer agent architecture. Obviously, the five modules that support the CBB stages operate in this case on the seller's viewpoint.

**E=**{**Book,Cd,Negotiation**}
**C=**{**JOHN,BOOKS,NARRATIVE,POETRY,BEHAVIOUR**}
**K=**{$k_1, k_2, k_3, k_4$}
**A=**{$a_{John}$}

Figure 3: The $John$'s ontology

**E=**{**(Book,**{**Book1,Book2,Book3,Book4**}**), (Cd,Cd1),(Negotiation,Negotiation1)**}
**C=**{**(JOHN,JOHN1),(BOOK,BOOKS1),**
**(NARRATIVE,NARRATIVE1),(POETRY,POETRY1),**
**(BEHAVIOUR,BEHAVIOUR1)**}
**K=**{$k_1, k_2, k_3, k_4$}
**A=**{$a_{John}$}

Figure 4: An instance of the $John$'s ontology

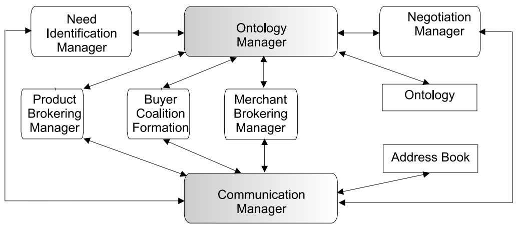


Figure 5: The customer agent's architecture.

## 5 CONCLUSIONS

This paper describes a new model for representing agent ontologies in a B2C e-commerce scenario, suitable to support the various stages of the Consumer Buyer Behaviour (CBB) model. The proposed model is capable of representing both the concepts involving in customers and sellers realities as well as the behaviour of customers and sellers in performing their activities. An architecture of a multi-agent system based on the presented ontology model is also briefly described. Our ongoing research mainly deals with the definition of machine learning techniques for efficiently extracting the knowledge patterns of the agent ontologies by directly observing customer and seller behaviours.

## REFERENCES

Bergamaschi, S., Castano, S., Vincini, M., and Beneventano, D. (2001). Semantic integration and query of heterogeneous information sources. *Data & Knowledge Engineering*, 36(3):215–249.

Buccafurri, F., Lax, G., Rosaci, D., and Ursino, D. (2002a). A user behavior-based agent for improving web usage. In *Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics Conference (ODBASE 2002)*, Lecture Notes in Computer Science, pages 1168–1185, Irvine, CA, USA. Springer-Verlag.

Buccafurri, F., Rosaci, D., Sarné, G., and Ursino, D. (2002b). An agent-based hierarchical clustering approach for e-commerce environments. In *Proceedings of the 3th E-Commerce and web Technologies (EC-Web 2002)*, Lecture Notes in Computer Science, pages 115–118, Aix-en-Provence, France. Springer-Verlag.

He, M., Jennings, N., and Leung, H. (2003). On agent-mediated electronic commerce. *To appear in IEEE Transactions on Knowledge and Data Engineering*.

Maes, P., Guttman, R., and Moukas, A. (1999). Agents that buy and sell. *Communications of the ACM*, 42(3):81–91.

OAS 2002 (2002). Workshop on ontologies in agent systems, first international joint conference on autonomous agents and multi-agent systems (aamas:oas 2002), available at http://autonomousagents.org/oas/2002/proceedings.pdf.

Omelayenko, B. (2001). Syntactic-level ontology integration rules for e-commerce. In *Proceedings of the 14th International FLAIRS Conference (FLAIRS-2001)*, pages 324–328, Key West, Florida, USA. AAAI Press.

Rosaci, D., Sarné, G., and Ursino, D. (2002). A multi-agent model for handling e-commerce activities. In *Proceedings of the International Database Engineering and Applications Symposium (IDEAS 2002)*, pages 202–211, Edmonton, Canada. IEEE Computer Society.

Terracina, G. and Ursino, D. (2000). Deriving synonymies and homonymies of object classes in semi-structured information sources. In *Proceedings of International Conference on Management of Data (COMAD 2000)*, pages 21–32, Pune, India. McGraw Hill.

Yang, J., Seo, H., and Choi, J. (2001). Morpheus: a more scalable comparison-shopping agent. In *Proceedings of the 5th International Conference on Autonomous Agents (AGENTS 2001)*, pages 63–64, Montreal, Quebec, Canada. ACM Press.