Agents’ roles in B2C e-commerce *

Luigi Palopoli a,**, Domenico Rosaci b and Domenico Ursino b
a DEIS,
Universit`a della Calabria
Via P. Bucci, 87036 Rende (CS), Italy
E-mail: palopoli@deis.unical.it
b DIMET,
Universit`a Mediterranea di Reggio Calabria
Via Graziella, Localit`a Feo di Vito,
89060 Reggio Calabria
E-mail: {domenico.rosaci,ursino}@unirc.it

The rapid evolution of the Internet from a general information space to an electronic market space provides the users with the possibility to navigate among thousands of Web sites for comparing products and merchants, for making their purchases or for obtaining some desired services. However, from the user’s viewpoint, such a navigation often requires an high cost in terms of time to spend on the Web to perform a satisfying comparison of the various alternatives. From the supplier’s viewpoint, it is needed to propose products in a suitable way to customers, taking into account the typology of each customer, her/his preferences, habits, etc.

In this context, generally denoted as Business-to-Customer (B2C) e-commerce, the use of software agents as mediators in e-commerce activities seems to be particularly promising.

In this paper, we describe and analyze the various roles agents have assumed in B2C e-commerce applications. Furthermore, we propose a Consumer Buying Behaviour model, called E²-CBB, that considers new emergent issues, as the capability to solve semantic heterogeneity, and the adaptive presentations of Web stores. By using such a model, we classify and compare a number of agent-based approaches for managing B2C e-commerce, proposed in the literature in the last ten years.

Keywords: B2C E-commerce, intelligent agents, multi agent systems

1. Introduction

Nowadays, the Internet is evolving from a general information space to a market space presenting a large variety of commercial services, ranging from electronic Web-stores to auctions, online booking and other services [1,157,29,94]. The rapid growth of networking systems is stimulating an ever-increasing number of businesses to participate in e-commerce worldwide [125]. Users have the possibility to navigate among thousands of Web sites for exploring desired sources and making their purchases. However, for performing the various tasks typically involved in e-commerce transactions, a customer has often the necessity to spend a large amount of time on the Web. Indeed, a user has to (i) identify those products appearing suitable to satisfy her/his needs; (ii) compare the different offers for products of interest; (iii) perform a negotiation, if the product’s price is not fixed [59,58,75]. In addition, a customer is often not able to perform a satisfying comparison of the various available alternatives, since the following complex problems must be dealt with:

- The increasing number of electronic marketplaces makes it impossible for customers to keep track of the ever-changing offer, demand and price situation.
- Some activities, as the participation to auctions, often need a continuous on-line presence of the customer for monitoring offers’ evolution.
- Due to the heterogeneity of Web sources, a product, or a product category, may have different identification in different Web sources, making comparisons really difficult.
Customers are not properly assisted for taking their decisions: they would often need an expert opinion for making the right decision. On the other hand, electronic suppliers have similar problems: they have the necessity to propose their products to customers in the most suitable way, taking into account their preferences, habits, etc. as well as competitors' offers.

The previously described issues are typical of the so called Business-to-Customer (B2C) e-commerce; most of them lead to the necessity for humans to find new ways of being simultaneously present at as many e-commerce sites as possible, in order to carry out profitable transactions. In this context, the use of software agents as mediators in e-commerce activities has been indicated as a promising direction to follow [58,105].

A software agent is a computing entity capable of (i) perceiving dynamic changes in the environment, (ii) autonomously performing some tasks delegated by users, (iii) possibly communicating and cooperating with other agents operating in the environment. Agents are considered useful in electronic business because they can solve problems in an adaptive, goal-driven way, acting as proactive catalysts [83]. They can retrieve information, monitor other trading agents, watch for potential opportunities, establish connections and negotiate with many potential partners at once. Moreover, they can automatically: (i) provide users having special interests with target information [104], (ii) filter information [90], (iii) integrate information from heterogeneous sources [81,82], (iv) derive stereotypical behaviour and, finally, (v) regulate transactions for more efficient business interactions [67].

The important role of Software Agents in the B2C e-commerce context is witnessed by the large number of agent models and architectures proposed in the literature [130,58,137,138] as well as by the large variety of real e-commerce applications adopting agent technologies within their sites. In this respect, it is worth mentioning the first Trading Agent Competition (TAC) held in Boston in July 2000; such a competition provided benchmark problems for the complex domain of e-marketplaces [53].

The purpose of this paper is to both provide a description and to illustrate a taxonomy of most of the agent-based approaches to B2C e-commerce proposed in the literature. A number of other surveys on this subject have been produced in the past. In particular, the work [59] represents the first attempt to review agent-based systems for B2C e-commerce. Other important works describing the state of the art in such a context are [154,98,179,143,11,19,74,96,101,112,65,70,61]. With respect to these surveys, the present paper provides the following contribution:

- Due to the rapid evolution of the agent technology, some important issues have been recently emerged that are not covered by the aforementioned works, in particular the aspect of the agent-mediated visit to an e-commerce site. In this paper, we propose a general model, called E²-CBB, for representing the customer behaviour in an e-commerce context; E²-CBB introduces a new phase, called site visiting, besides the other ones generally taken into account in the previous surveys. This allows recent agent-based approaches, dealing with this specific problem, to be classified.

- The site visiting phase mainly concerns the extraction and the organization of information associated with both the site content and customer behaviour in accessing it. Problems such as the construction, the management and the exploitation of a customer profile, as well as the adaptation of the site presentation to different visiting customers, are studied in this context. From our point of view, the site visiting phase is central in the modern agent systems, since the other phases of the e-commerce process, such as the identification of the most suitable products and merchants, can suitably exploit the information extracted during this phase.

- We systematically analyze a number of recent systems and build a taxonomy of them based on the proposed categorization.

- Finally, we have taken into account that the subject underlying this survey involves concepts belonging to quite different fields, such as artificial intelligence, software system engineering, economics and so on. As such, we extensively describe various concepts that are not strictly pertinent to the computer science field (as, for instance, the various types of auction protocols) and refer several authoritative work about e-commerce, considered from the economics viewpoint. In addition, we give
some basic concepts about agents, by providing also an agent taxonomy, that can be useful to better understand the various agent typologies we introduce in this survey.

The plan of the paper is as follows. Some basic concepts about e-commerce are given in Section 2. Section 3 is devoted to introduce software agents. Agent-based approaches to B2C e-commerce applications are discussed in Section 4. Section 5 deals with the new emergent role of the Semantic Web Services. Finally, in Section 6, some conclusive remarks on the future perspectives of the research in agent-based B2C e-commerce are drawn.

2. E-Commerce: from B2C to B2B

Generally, the term e-commerce denotes any business activity carried out electronically rather than face-to-face [157,174]. Since entities involved in e-Commerce activities are basically firms (generally denoted as businesses), customers and governments, six kinds of e-commerce have been defined in the literature [48]: Business-to-Business (B2B), Business-to-Customer (B2C), Business-to-Government (B2G), Customer-to-Customer (C2C), Government-to-Customer (G2C) and Government-to-Government (G2G). However, the two major forms of e-commerce are B2C e-commerce, in which businesses sell directly to customers, and B2B e-commerce, in which businesses carry out their tasks with other businesses [151].

For instance, consider a customer who visits a megastore, buys some products and pays for them: in this case she/he is carrying out a B2C commerce activity, since the involved entities are a Business (the megastore) and a Customer. Analogously, when the customer carries out the same activities on an e-shop, we say that she/he carries out a B2C e-commerce activity. On the other hand, the megastore presumably buys all the goods it sells from other businesses: such a task is exactly a B2B commerce activity, since both the entities involved in the underlying transactions are businesses. Analogously, if an e-shop carries out its buying activity by exploiting electronic engines provided by other selling companies, we say that it performs a B2B e-commerce activity.

Since it is easier to reach customers through the Internet than to set up communication channels among different organizational networks, B2C e-commerce has grown faster than B2B e-commerce. Such a trend has changed in the last five years because firms have understood that huge costs can be saved by carrying out their inter-business transactions directly via Web. As a consequence, they are paying growing attentions to B2B e-commerce applications and, indeed, B2B e-commerce is nowadays expected to grow faster than B2C e-commerce.

The U.S. Department of Commerce estimates that, in the year 2004, between $4 billions and $14 billions on-line sales will be performed in the B2C e-commerce area in the US; since the number of Internet users grew enormously in the past years (e.g., from 100 to 300 millions between 1997 and 2000), it is possible to anticipate a significant growth of B2C e-commerce in the next years.

From an agent-system viewpoint, there are many differences between B2B e-commerce and B2C e-commerce, which deeply influence the characteristics of the corresponding agents. The most relevant of them are summarized by the following considerations:

- B2C e-commerce involves only a subset of company operations, basically those business aspects dealing with marketing and selling to consumers; instead, B2B e-commerce deals with a superset of B2C e-commerce, encompassing several aspects of doing business, from advertising and marketing to sales, ordering, manufacturing, distribution, customer service, after-sales support, etc. [147].

- The major factors driving the implementation of B2C e-commerce include increasing revenues, developing new sale channels and new customer groups, obtaining competitive pressures and extending global reach [1]. Vice versa, the major factors driving the implementation of B2B e-commerce include: improving business processes, improving supplier and customer relations, increasing profitability, saving costs, increasing distribution, pressing from customers and competitors.

- Finally, while interoperability is not a major issue for B2C e-commerce, it plays a key role in B2B e-commerce. With regard to this, it is worth pointing out that, in B2C e-commerce, interaction with customers is generally carried out via a Web browser, whereas, currently, in B2B e-commerce, no form of interaction has been adopted as a “de facto” standard.
This survey is mainly focused on agents conceived for performing B2C e-commerce activities.

3. Agent technology: a way for reducing user work

Agent technology has largely developed in the last few years, evolving rapidly along a number of typologies such as intelligent agents, mobile agents, collaborative agents, etc. Such a large development has mostly been due to the necessity of assisting and simplifying the interaction between users and computers.

When users directly perform everyday activities over the net, such as managing e-mail, acquiring news, surfing the Web, selecting products from e-commerce sites and so on, they have both to explicitly start all such tasks and to monitor all involved events. In order to simplify such a complex interaction between users and computers, autonomous software agents [102,71,18] can be adopted. These are software modules which collaborate with users by acting as personal assistants. The idea of exploiting agents for carrying out some of the user activities, more specifically those too onerous to be performed through a direct interaction with a computer, emerged in some pioneeristic works like those of Alan Kay [77,78], Brenda Laurel [92,93] and Nicholas Negroponte [114,115] that advocate agents as personal assistants acting on user’s behalf in the cyberspace. In the last few years, several agent theories, languages and architectures have been developed and a lot of definitions have been provided for explicating the use of the word “agent” [152,142,141,103,60,35,50,173,175,124]. We try to synthesize the requirements expressed for agents, as follow:

Definition 3.1 A software agent is a computing entity capable of perceiving dynamic changes in the environment and, consequently, of autonomously performing user delegated tasks which may change that environment, possibly by communicating and cooperating with other agents operating in the environment.

Moreover, on the basis of the various definitions cited above, some properties can be introduced for characterizing agents’ behaviour [50,117]. In more detail, an agent is:

- **Reactive**, if it is capable of answering to environment changes.
- **Autonomous**, if it can operate on its own.
- **Deliberative**, if it owns an internal symbolic reasoning model.
- **Pro-active**, if it is able to take the initiative.
- **Continuous**, if it is a continuous running process.
- **Adaptive**, if it adapts its behaviour on the basis of its own experience.
- **Mobile**, if it is capable of transferring itself from one machine to another.
- **Flexible**, if its actions are not fully pre-defined.
- **Sensible**, if it has a believable personality and a representation of emotional states.
- **Social**, if it is capable of communicating with other agents.
- **Collaborative**, if it shares its own knowledge with other agents in order to pursue its own or shared tasks.

According to [117], software agents can be classified as follows:

- **Interface Agents**. Maes defines interface agents as “computer programs that employ artificial intelligence techniques to provide active assistance to a user with computer-based tasks” [102]. In particular, an Interface Agent can be seen as a learning interface, operating in applications exploited by the user, which gradually builds a knowledge base of the actions the user carries out in certain situations. It is desirable that such an agent exhibit a learning potential to improve its decision-making methods.

- **Mobile Agents**. Mobile Agents are software engines capable of moving around networks such as the Internet, interacting with other hosts, gathering information on the behalf of their owners and returning any interesting information they found [117]. Mobile agents stop their execution when they move around a network; they continue their execution only when they arrive at an host capable of re-starting them. The necessity of suspending their execution often represents a serious limitation here.

- **Collaborative Agents**. A collaborative agent is able to communicate with, and react to its environment when other agents operate therein. Collaboration exists when the actions of an agent achieve not only its own goals,
but also those of the other agents present in the environment. Agent collaboration is obtained by singling out, in an agent community, the agents potentially suitable for cooperating, and by sharing knowledge among them [24]. This can be obtained, as an example, by computing similarities between agent ontologies, thus forming clusters of agents having similar interests and behaviour (see, for instance, the approach proposed in [139]). In order to make it easier, an agent Knowledge Interchange Format (KIF) can be exploited (the Knowledge Query and Manipulation Language - KQML [49] is an interesting example of a KIF).

- **Reactive Agents.** Some works, such as [22], tried to show that an intelligent behaviour in agents can be realised without symbolic representations of the agent environment, which resulted in the definition of reactive agents. These are particular agents which do not own internal symbolic models of their environment. Instead, they react to a stimulus or input caused by some state or event in their environment. Reactive agents typically have a high level of autonomy as they simply react reliably to events in their environment.

- **Hybrid Agents.** As shown in [117], all previously introduced agents are characterized by some strengths and weaknesses and no agent class is perfectly suited for each given task. Therefore, it appears particularly interesting to exploit the characteristics of several agent classes at the same time, (e.g., autonomous and cooperative agents) for developing agent-based applications. This is the classical scenario for exploiting hybrid agents. These are simply systems consisting of different agent classes, together with an agent control unit allowing each agent class to effectively interoperate with the other ones.

- **Heterogeneous Agents.** Whereas hybrid agent systems consist of agents belonging to different classes, heterogeneous agent systems consist of of agents (belonging to either the same class or different classes) that have their own knowledge expressed in different formats (e.g, XML ontologies, E/R knowledge bases, etc.) and inter-operate by means of controller agents [136,14,140]. These last make the interoperability easier by exploiting standard agent communication languages (ACL) and methods.

- **Information Agents.** These agents usually exploit Web search engines, such as Google, Yahoo and Altavista, in order to support a navigator in the management of the huge amount of information found over the Internet.

- **Holonic Agents.** An holonic agent [51] consists of a structured hierarchy of sub-agents (called holon) each possibly including further sub-agents. One sub-agent of the holon, called *head agent*, moderates the activities of all the other sub-agents.

In this paper we are interested in surveying about software agents specifically operating in e-commerce environments. In the next sections, we will concentrate our attention on them.

## 4. Agents supporting B2C e-commerce activities

There are several reasons which justify the exploitation of agents in e-commerce applications; the most relevant of them are:

- **Intelligence.** Agents can exploit learning techniques for taking advantage of their own experience in order to improve their performances.

- **Personalization.** An agent assisting an Internet user can store an internal representation of her/his preferences, i.e. a *profile*, and can tune its behaviour according to it.

- **Social behaviour.** Agents are able to interact with each other. Such a communication capability may allow them to negotiate, e.g. over prices, services and transactions.

- **Autonomy.** In the previous section we have pointed out that the most relevant feature of an agent is the ability to work independently on the human intervention. In e-commerce applications, this ability can be exploited for delegating to agents several tasks that would require a relevant time cost for the user, e.g. for automatically searching the most convenient products, for suitably negotiating with sellers, and so on.

The points we have listed above allow to conclude that agents can be advantageously exploited as mediators in B2C e-commerce. In order to explore and categorize the various agent-based approaches to e-commerce proposed in literature, it
is useful to have a common reference model representing the underlying e-commerce activities. For B2C e-commerce, it is possible to directly refer to the buying behaviour of the customers and, thus, to use one of the most classical models proposed in the traditional market theory. We briefly describe it in the next subsection.

4.1. Modeling Consumer Buying Behaviour

4.1.1. Existing CBB Models

Agents operating in e-commerce activities can be seen as mediators between the actual subjects involved in this context, i.e. customers and businesses.

It is useful to consider the behaviour of an agent in the context of a uniform framework representing the various phases of its activity. Traditional marketing research has developed many descriptive theories and models that attempt to capture the consumer behaviour in finding, buying and using goods and services. The research about the Consumer Buying Behaviour (CBB) deals primarily with B2C e-commerce although several CBB concepts are still valid for B2B and C2C e-commerce processes. Below, we describe three proposals for modeling the consumer behaviour within an e-commerce site:

1. **The Traditional CBB models (T-CBB).** There are several models that were developed in the CBB context, such as the Nicosia Model [116], the Howard-Sheth Model [64], the Engel-Blackwell Model [45], the Bettmann Information Processing Model [15] and the Andreasen Model [5]. They differ in various aspects; however, as noticed in [59], all of them propose six relevant steps in the consumer buying behaviour. These steps, graphically represented in Figure 1, are listed below:

   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 that is 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. This stage is called *Product Recognition* in the Engel-Blackwell model [45].

2. **Product Brokering.** Once a consumer has identified a need to satisfy, she/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. **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.

Note that some classical marketing models, such as the Nicosia model [116], merge both product brokering and merchant brokering phases into a unique *search evaluation* phase, whereas other authors, e.g., [45], enucleate in both the brokering phases two orthogonal stages called *Information Search* and *Evaluation of Alternatives*.

4. **Negotiation.** During this step, the various terms of the transaction as, for instance, the price, are determined.

In real-world situations, a negotiation step often increases transaction costs in too an onerous way for both consumers and merchants. Furthermore, there often might be impediments, notably time and space constraints, to exploit negotiation mechanisms. These impediments mostly disappear in the Internet world.

The benefit of negotiating the price of a product instead of fixing it is that it relieves the merchant from needing to determine the value of the good a priori [105]. Rather, the price is dynamically determined by the marketplace. The duration and complexity of the negotiation strictly depend on the market.

Some traditional CBB models do not explicitly identify a negotiation phase, but some of them [116,45] consider a *Choice* or a *Decision* phase that is comparable with the negotiation process described above.

5. **Purchase and Delivery.** This step can either represent the termination of the ne-
gotiation phase or can occur sometimes afterwards. It is possible, in some cases, that payment or delivery options can influence product and merchant brokering.

6. Service and Evaluation. This step involves the final activities of the buying process, such as product service, customer service, evaluation of the overall process and so on.

In Figure 1 we highlight some e-commerce processes, such as publicity, search engines, portals, auctions, etc, that might support the six steps we cited above.

- The Extended CBB model (E-CBB). [61] proposes to extend the traditional CBB model described above in order to cover the buyer coalition formation behavior. Indeed, it observes that 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. Such a coalition groups customers cooperating with each other in order to achieve a common goal. If each customer has an associated software agent, the buyer coalition formation can be viewed as a set of software agents that communicate with each other for performing a common task, e.g., trying to approach the merchant with a large order in order to obtain a leverage. E-CBB does not include the two last stages of T-CBB (purchase and delivery, product service and evaluation) since, according to this model, these stages do not present aspects strictly requiring the use of the agent technology.

- The CBP model. The main limit of the previous models is that the various steps might overlap and reorder differently. In order to overcome such a limit, Bohm, Felt and Uellner [16] propose a different model focusing on consumer buying behaviour within e-commerce sites. This model, called e-commerce Customer Buying Process (CBP) and depicted in Figure 2, allows to decompose, in a more detailed way than the models we have previously presented, a buyer process action into phases (like first contact, information, etc.) and processes (like welcome to shop, Registration, etc.).

4.1.2. A new CBB model

After determining the most suitable merchant for buying the desired products, before beginning negotiation, the customer often visits the merchant site, in order to know more details about the product offer or to find other offers related to products of interest. Agents can suitably mediate such a stage, as follows:

- Customer’s viewpoint. An agent can retrieve, on behalf of its owner, the most interesting information about the selected products. Furthermore, it can automatically explore the merchant site for determining other products of interest, and can give advice to the customer about their presence in the site.

- Seller’s viewpoint. Some seller agents can be present on the merchant site for providing the customer with the most suitable presentation, w.r.t her/his profile. For instance, a seller agent may recognize that the customer is not interested in multimedia presentations and, consequently, may decide to propose a light presentation, without animations, applets, etc. Such a capability is related to the well-known issue of the adaptivity in the Web sites.

Both from the customer’s and the seller’s viewpoint, it is possible to note that the operations executed by a unassisted customer, whenever she/he buys a product or a service from a B2C e-commerce site, can be roughly grouped into two main activities, namely: (i) the exploitation of a Web search engine (such as Google) for finding potentially interesting sites; (ii) the use of a Web browser for navigating through the pages of a site that the customer has judged to be interesting. These two activities are important for the customer, that has to carry out them for searching interesting goods, and also for the seller, that in influencing is interested in attracting her/him to its own site and products. While a large variety of tools already exists over the Internet for helping a customer in the search of potentially interesting sites, only a relatively limited number of supports are available for assisting her/him in the navigation within an interesting site.

One of the most common type of such supports are recommender systems [145,146] which have been conceived for helping the customer of an e-commerce site in purchase activities. In order to carry out its task, a recommender system provides the customer with the list of those products and services available on the site and appearing to be the most successful, according to the opinion and past purchases of other customers. Moreover, it may analyze the past behaviour of the customer on the site for predicting those goods which, presumably, will be of interest for her/him in the future. Last, but not the least, it could classify a customer according to some parameters and might provide offers taking into account this classification. In order to perform these activities, a recommender system must process available data. Such a task can be carried out by exploiting various computation tools such as cross-sell lists and information filters. Note that, as such, recommender systems are concerned with the first four stages of the CBB model.

Another important tool that is provided in a large number of e-commerce sites is a (often simple) handler of customer profiles, exploited for
proposing personalized offers. With such tools, profile construction usually needs a relevant intervention of the customer who is required to provide information about her/his interests and desires; the profile is usually just intended as a summary of this information. This profile construction method requires the customer to spend a certain amount of time for constructing and/or updating her/his profile. In addition, these systems make often quite a straightforward management of user’s interests as stored in the profile.

When a customer accesses an e-commerce site, generally she/he must personally search the issues of interest through the site. As an example, consider the bookstore section of Amazon.com: whenever a customer looks for finding a book of interest, she/he must carry out an autonomous personal search of the book throughout the pages of the site. For improving the effectiveness of the e-commerce services, it would be necessary to increase the quality of the interaction between the site and the customer, on the one hand, and to exploit richer customer profiles, taking into account desires, interests and behaviour, on the other hand. In order to obtain these results three main issues have to be taken into account.

First, a customer can access many e-commerce sites; a faithful and complete customer profile can be constructed only taking into account the contents of all these sites. In order to construct such a customer profile, it may be necessary to recognize semantically similar concepts represented by different constructs in the various sites and to represent them by a unique construct in the customer profile.

Second, it appears necessary to represent in the customer profile, not only her/his interests, but also her/his behaviour in accessing interesting items. This issue has been considered in some works in the Information Systems field ([14,28,165,23]). As far as the e-commerce research is concerned, a solution for handling the large heterogeneity of product formats has been found in the exploitation of intensional descriptions of product characteristics, generally referred to as customer ontologies [118].

Third, for a given customer and e-commerce site, it would be useful to compare the profile of the customer with the offers of the e-commerce site for extracting those issues that probably are of interest for the customer. Existing techniques for satisfying such a requirement are mainly based on the exploitation of either log files [26,180] or cookies [126]. Techniques based on log files are capable of registering some information about the actions carried out by the customer on accessing the site; however, they are not capable of matching customer preferences and site contents. Vice versa, techniques based on cookies are able to carry out a certain, even if primitive, matching; however they need to know and exploit some personal information that a customer might consider private.

it appears mandatory to solve the above issues concerning the management of the site visits. In fact, by using information on customer navigation, it is possible to support the search of those products appearing to be the most promising according to the customer past behaviour, as well as to support the merchants to propose interesting offers to customers. As a consequence, customer past behaviour is a potential input for the product brokering and the merchant brokering phases. Moreover, the construction of customer profiles, that can be performed in the site visiting stage, may be used for comparing different customers and for discovering similar tastes and preferences. Such an information could be exploited for supporting the buying coalition formation and the negotiation phases.

None of the aforementioned CBB models include such a site visiting stage; however, several recently proposed agent-based systems provide some capabilities useful to help customers and sellers to manage this stage.

In the following of this paper, we will use a further extension of the E-CBB model, covering also the site visiting stage, as the reference model of the Consumer Buying Behaviour for our e-commerce agent classification. We choose the E-CBB model since it is more schematic (though less detailed), and therefore more suitable for a classification, than the CBP model. Hereafter, we will denote our extended E-CBB model simply as E²-CBB model. It presents the following six stages (see Figure 3): (a) need identification; (b) product brokering; (c) buyer coalition formation; (d) merchant brokering; (e) site visiting; (f) negotiation. The meanings of each of these stages is that specified in the above discussion. In this figure, we have represented by solid lined the chronological path of the customer in performing the various stages, while we have used dotted lines for representing relationships between stages. The figure highlights that, on the one
hand, the visit of the site strongly influences other stages as need identification, product brokering, merchant brokering and buyer coalition, since this visit contributes in defining those customer preferences that will be exploited in the other stages and, on the other hand, the site visit itself is influenced by the other stages because the site presentation may strictly depend on the choices made in the past by the customer in those stages.

4.2. A taxonomy of agent-based approaches supporting B2C e-commerce applications

In the past, various agent-based approaches have been proposed for supporting e-commerce activities. In the following, we analyze them into detail, according to their relevance w.r.t. the various steps of the \(E^2\)-CBB model. As a synoptical view of such an analysis, we propose the taxonomy illustrated in Figure 4.

At the end of this Section, we have included a sub-section concerning two additional issues that, in our opinion, are worth of being analyzed, namely (i) support to novice users in configuring a complex product and (ii) the recommendation strategies that do not suggest just similar products but also take care of diversity

4.2.1. Agents supporting Need Identification

Below we overview some existing agent systems that help to anticipate consumers’ needs in an automatic way. Most of them are very simple; their only purpose is to inform the customer when a good of interest becomes available.

- **Amazon.** To keep customers interested in Amazon [3](see Fig. 5), this firm offers two forms of e-mail-based services, namely the “Eyes” program and “Editors Service”, to its registered customers. The “Eyes” program is a personal notification service in which customers can register their interests for a particular author or topic. Once registered, they are notified each time a new book written by one of their favorite authors or discussing topics of their interest becomes available.

  The “Editors Service” program provides editorial comments about featured books via e-mail. Many full-time editors at Amazon read book reviews, examine customer orders and survey current events to select the featured books. These and other editors employed by the firm provide registered users with e-mail updates on the latest books they have been reading.

  The “virtual” services described above benefit of the automatic user identification, performed by the software agent.

- **Fastparts.** The FastParts service at the site [47] provides online trading facilities for registered members. Among the others, the AutoWatch agent allows members to list goods they need that are not currently available on the site. The agent will notify the customer by e-mail when those goods are posted for sale.

- **Firefly.** The Firefly agent [148] helps customers to find products of interest by comparing product rates of a shopper with those of the other customers and identifies groups of homogeneous buyers on the basis of their rates. Such a shopper grouping is exploited by the Firefly agent for recommending to each customer those products which received a high rate from the other customers of the same group but which she/he did not consider yet.

- **Classified 2000.** Excite’s Classified 2000 [34] is a very simple tool supporting need identification. It behaves as follows: the customer can specify what she/he is looking for by price and type. A tool called Cool Notify then sends an e-mail message to the customer when some goods having the specified features have been located on the Excite site.

- **National Instruments.** At the site [149], it is possible to create a personal profile by filling in some fields of a form and registering as a member. The system provides each registered member with e-mails about National Instruments calendar event notification, closing stock prices and news releases.

- **SERSE.** In the system SERSE [164], a notification agent receives XML messages from a Notification Server and parses them. It can then resend the content as an ACL message via aPortal Agent. The messages can be the notification of a new OWL ontology or of new content acquisitions. If the notification refers to a new ontology, the Notification Agent extracts the concepts definitions from the OWL file and populates the system with the routers. It also informs the routers of which agents are their immediate neighbours.
4.2.2. Agents supporting product brokering

After a customer has identified a need, she/he has to examine the various products available in the marketplace that might satisfy this need. Several agent systems have been developed for supporting such a task. Generally, they adopt one of the following techniques:

1. **Feature-based filtering**: Products are selected on the basis of feature keywords. If a customer is interested in a product with some particular features (e.g., the category, the price), its agent specifies them to the seller site, and obtains all the products with those features.

2. **Collaborative filtering**: Products are selected on the basis of the similarity holding amongst different customers between different customers. Generally, systems using this technique associate with each customer a profile, storing her/his preferences. If a customer $c$ needs some recommendations, the agent determines all the other buyers having a profile similar to that of $c$, and recommends to $c$ the products they considered the most valuable.

3. **Constraint-based filtering**: Systems using this technique allow the customer agents to specify some constraints on the desired products (e.g., the price range, the delivery time, etc.) and return all the products satisfying these constraints.

Below we describe some agents supporting product brokering. In Figure 4 we classify them on the basis of the adopted technique (feature-based, collaborative filtering or constraint-based).

- **Amazon**. Amazon’s recommender tool, called *Customers Who Bought*, provides an information page for each book, showing some details and purchase information. When a customer visits the site and selects a book from the catalog, two different recommendation lists proposed, she/he gets. The first one includes books frequently purchased by other customers who bought the selected book. The second one includes authors whose books are frequently purchased by customers who bought works by the author of the selected book.

- **CDNOW**. Also CDNOW [31] exploits a recommendation tool, called *Album Advisor*, that performs some different tasks. The first two are similar to those carried out by the *Customers Who Bought* tool in Amazon: a customer can select the information page of a given album (resp., artist) and the system recommends ten other albums related to the selected one. The third task is called *Gift Advisor*: it allows the customers to type the names of up to three artists and, after this, generates a list of artists who are considered similar to the specified ones.

A further feature, called *My CDNOW*, enables customers to set up their personalized music store, based on albums and artists they like.
When a customer buys something from CDNOW, the purchases are automatically put into an *own it* list; the user, on the basis of her/his satisfaction, can also classify purchases of the *own it* list into two categories called *own it and like it* and *own it and dislike it*. This information is exploited by the system for further recommendations. A customer can also request recommendations on the basis of her/his past behaviour; when this sup-
Fig. 5. The Amazon's need identification tool

- **Drugstore**. Drugstore [43] provides its customers with a recommendation tool called **Advisor**, that allows them to indicate their preferences when they desire to buy a product of a certain category. For instance, a customer willing to buy from the category *cold and flu remedies*, can indicate the symptoms to relieve (e.g., runny nose and sneezing), the desired form for reliefs (e.g., caplets) and the age category (e.g., adult). After that the customer has provided the above information, **Advisor** returns a list of recommended products.

- **eBay**. eBay [44] is an auction support system. Two recommendation tools are present at this site. The first, named **Feedback Profile**, allows both buyers and sellers to give a feedback for updating the profiles of those they have interacted with. The feedback consists of a satisfaction rate that can assume one of the values *satisfied*, *neutral*, *dissatisfied*. Purchasers are able to view the profile of each seller, consisting of a table with the rates she/he received in the past 7 days, in the past month, and in the past six months, as well as an overall summary. The second tool, called **Personal Shopper**, allows a customer to indicate items of interest. Customers can enter a *short term* (30/60/90 days) and a set of keywords specifying their desires and price limits. The site periodically carries out a search over all auctions, on the basis of the provided specifications, and informs the customer via e-mail about the results of the search.

- **MovieFinder**. At the site [110], three recommendation systems are available. The **User Grade** system allows customers to register themselves at the site and to give a rate between *A* and *F* to the movies they have seen. These rates are averaged over all customers and reported in the site. Furthermore, a rate for each movie is provided by the site within the **Our Grade** system. The third system, called **Top 10**, allows a customer to select a category from a list and provides her/him with the top ten movies in that category, as judged by the site editors.

- **Reel**. At the site [133], the **Movie Matches** tool provides the customer with recommendations about movies. Movie Matches behaves in a way similar to Amazon’s **Who Bought**. In particular, recommendations consist of two set of movies: the *close matches* set contains some links to the information pages of movies appearing to be very similar to that a customer decided to buy. The *creative matches* set con-
tains some links to the information pages of movies considered similar, w.r.t. some particular respects; the recommendation system also provides motivations for its choices.

- **LIBRA.** LIBRA (Learning Intelligent Book Recommending Agent) \[109\] is a new project aiming at using machine learning techniques to recommend books to a reader by learning about her/his interests. The system learns by examining abstracts and other short descriptions of books that the reader has marked as desirable or undesirable.

- **MovieLens.** MovieLens \[111\] requires the user to supply information about the typologies of movies she/he likes. Then, it elaborates the supplied information for providing recommendations about other movies possibly interesting for the user.

- **Alexa.** Alexa \[2\] is a browser plug-in that provides users with recommendations about sites based on the Web page they are currently visiting. More specifically, it supplies the list of sites that are someway relevant and related to that the customer is currently visiting.

- **PersonaLogic.** PersonaLogic \[129\] offers to e-commerce companies complete, customized, personalized decision guides for their consumers, completely served into their own Web sites. PersonaLogic is used within Netscape, AOL, Excite, Lycos, American Express, ZDNet etc. The PersonaLogic agent, installed in an e-commerce site, allows each customer of the site to specify some constraints on product features. Then, it exploits a constraint satisfaction problem (CSP) solving-technique for filtering out those products that do not satisfy all the specified constraints and returns a list of the products that best match them, ordered by a coefficient representing their matching degree. For instance, in searching for a car, the agent asks a customer to differentiate the importance of certain factors such as price, size, features, safety, technical aspects and manufacturer(see Figure 6).

- **Firefly.** The Firefly agent helps customers to find products of interest by generating recommendations via a “word of mouth” recommendation system called Automatic Collaborative Filtering (ACF). Instead of filtering products on the basis of constraint specifications, like PersonaLogic, Firefly compares product rates of a customer with those of the other customers and identifies clusters of homogeneous buyers on the basis of their rates. Such a clustering is exploited by the Firefly agent for recommending to each customer those products which received an high evaluation from the other customers of the same group but which that customer did not consider yet. This way, Firefly carries out both a need identification and a product brokering activity.

- **Tete-a-Tete.** Like PersonaLogic, the Tete-a-Tete agent \[59\] exploits a CSP technique for assisting customers in their product brokering activity. In particular, the customer provides the system with hard and soft constraints; these are exploited by the agent for identifying the most promising products.

- **WEBS.** WEBS (Web-based Electronic Brokering System) \[91\] consists of some networked brokering agents, each handling a particular category of products or services, such as security trading, books, computer software, etc. For each brokering agent, several sub-agents are provided for handling the various steps of the CBB model. In particular, a profiling sub-agent is responsible for need identification and product brokering. Each profiling sub-agent associated with a particular buyer registers her/his preferences w.r.t. the products of a particular domain. The profile is exploited by the agent for supplying information about products presumably of interest for the customer. Once information about a product has been supplied to a buyer, a form-based interface is presented, that allows to require further information, in the case that the customer is indeed interested in the product. The buyer reaction to the presented product information is stored in a dialog file. This is analyzed by the learning component of the profiling agent for updating the profile.

- **SETA.** SETA (SErvizi Telematici Adattativi) \[7\] is a toolkit for building adaptive Web Stores. It exploits some user modeling techniques for personalizing the interaction with the customer. In particular, it generates the catalogue pages of an e-commerce site on the basis of the customer preferences. SETA defines a conceptual representation of a Web store by constructing the so-called Prod-
Product Taxonomy, which stores both the definition of all products present in the Web store and their relationships. The Product Taxonomy is exploited for generating a rational sequence of Web pages making the customer navigation easier. The taxonomy is directly mapped in the structure underlying the hypertext forming the Web catalogue. Besides this intensional representation of products, SETA also stores extensional information about the various items in a Product Data Base.

Furthermore, SETA handles knowledge about customers, such as personal data, preferences about product properties (e.g., quality, ease of use, etc), receptivity, domain expertise, and so on.

When a customer accesses the Web store for the first time, the system requires some personal information to be specified. Then, it constructs a rough customer profile by matching customer personal data with those included in a stereotype knowledge base. After this, whenever the customer browses the product catalogue, dynamic user modeling techniques are exploited to modify the associated profile on the basis of of behaviour customer’s.

The Product Brokering activity in SETA is mainly handled by: (i) a Personalization Agent, which generates the HTML code for the Web pages to be shown to the customer on the basis of her/his profile, and (ii) a Product Extractor, which assists the customer in product selection by suggesting her/him a list of products, ordered by their probability to satisfy her/his desires.

Meng et al. In [108], a new business agent architecture is presented, in which the Business Spy Agent (BSA) automatically retrieves supply and demand information from e-commerce web sites. The Supply and Demand Analysis Mechanism (SDAM) uses Natural Language Processing (NLP) technologies to extract products and trading information.

InterMarket. InterMarket [87] is a Multi-Agent System in which agents have the autonomous decision-making and mobility capabilities. They can proactively monitor trading opportunities, search for trading partners and products, and they are able to move to the e-marketplaces through the Internet and can be initiated from different computer platforms and mobile devices such as mobile phones and PDAs.

Easishop. Easishop [79] is an intelligent mobile shopping system, that allows the user to specify a composite set of preferences, profile and shopping list information. The agents in
the system are implemented by using agent factory, that is a cohesive framework that supports a structured approach to the development and deployment of agent-oriented applications. In order to facilitate interagent data coordination, the system uses a product ontology based upon the United Nations standard products and services code (UNSPSC).

4.2.3. Agents supporting buyer coalition formation

In the buyer coalition formation phase, customers that have made a decision about what products to buy, decide to form a coalition with other customers interested in the same products. This phase is not strictly necessary, and a customer having completed the product brokering stage might move directly to the merchant brokering stage, without interacting with other customers. However, in the present agent-based e-commerce systems, the utility for the customers to interact with one another and to form customer coalitions is considerably emerging, and interaction can be efficiently managed with the support of software agents.

In Multi-Agent Systems literature, the problem of constructing groups of agents that collaborate for improving their results has been dealt with, according to three different approaches:

– **Agent teams.** An agent team is a set of agents that collaborate to pursue some shared or common goals. This approach explicitly requires agents to share a sort of joint goal to be reached; thus, each agent of the set is not completely self-interested since the agents’ utility measure is strictly dependent upon the performance of the group as a whole. Relevant research in this setting is that of [69,156,163].

– **Agent coalitions.** An agent coalition is a set of agents that work together to achieve a larger goal. Each agent of the set is completely self-interested and participates in the coalition since, by this participation, it receives a higher utility than if it worked alone. Notable examples of research in this setting are [150,144,84]. In [177], a buyer coalition formation scheme, called GroupBuyAuction, is proposed. In such a scheme, buyers form a group based on a category of items. A buyer can indicate a list of items and can specify a desired reservation price for each item. Sellers can bid volume discount prices. A particular group leader agent divides the group into coalitions, selects a winning seller for each coalition, and calculates surplus division among buyers. In [167], a buyer coalition model consisting of five stages is proposed. The stages are negotiation, leader election, coalition formation, payment collection and execution. An implemented system using this model is tested in a collective book purchasing context, showing how the supplier agent gives a volume discount according to the size of the coalition.

The most important problem for the coalition formation technologies is the computational complexity of coalition structure generation. That is, once a group of agents has been identified, the time cost for partitioning them in order to maximize the global social payoff is NP-hard and even finding a sub-optimal solution requires searching an exponential number of solutions. In [40], a novel algorithm for coalition structure generation is presented, that produces solutions that are within a finite bound from the optimal.

– **Agent congregation.** An agent congregation [21] is a long-lived entity composed by one or more self-interested agents that interact each other for satisfying their needs. The concept of agent congregation is closer to that of agent coalition than that of agent teams, since the agents of a congregation are self-interested. However, differently from an agent coalition, that usually deals with the accomplishment of a single task, an agent congregation is a more long-lived entity that can support agents in the accomplishment of several tasks. Furthermore, differently from the agent coalition, in an agent congregation an agent may not know the identities of the other congregation members when it decides whether to join or not. A congregation serves to allow agents to come together and make bilateral exchanges, without any joint goal to pursue by the cooperation. In [20], the use of agent congregation for supporting market formation is analyzed. In particular, authors illustrate how members of a multi-agent system self-organize into a set of markets such that agents are able to find suitable partners while retaining low computational costs.

In [168], a congregation formation mechanism for the electronic marketplace, that extends
the existing transaction-oriented coalitions to long-term ones, is presented. This proposal focuses on self-interested agents that trade goods or knowledge in a large-scale multi-agent system. In this context, congregations are used as a means of grouping agents for improving their cooperation and coordination in such a way to increase their individual benefits.

4.2.4. Agents supporting merchant brokering

During the merchant brokering stage, a customer compares the various merchants, that might provide a product of interest, for choosing the most convenient one. Nowadays, there exist several price-comparison sites, as [132,131,17,30], that simply compare the price of several sellers with respect to the same product. However, more sophisticated several agent-based systems have been proposed for automating this stage [171,80]. Below we discuss the most significant ones:

- **BargainFinder.** BargainFinder [10] is a shopping agent able to compare the price of a specific product proposed by different e-commerce sites (see Figure 7). Several on-line merchants cooperate with BargainFinder in this task in order to both improve their competitiveness and receive more visibility. A shopper can use the agent directly at the BargainFinder site, specifying a product, and the agent returns the list of the best offers for that product, selected from all the e-commerce sites joined with BargainFinder.

- **Jango.** Jango [42,68] behaves similarly to BargainFinder. However, in order to make a complete comparison of prices, Jango generates the price request for a product from consumer using Web browsers instead of a central site. This way, requests to merchants appear as real customer’s requests. Clearly, this behaviour is quite aggressive w.r.t. the merchants.

- **Kasbah.** Kasbah [32,76] is an on-line multi-agent transactions system developed at the MIT Media Lab, supporting both the merchant-brokering and the negotiation stages. From a merchant-brokering viewpoint, it allows a customer, wishing to buy a good, to create an agent, to provide it with some specifications and to send it into a centralized marketplace. Kasbah agents are able to proactively search potential sellers for the selected good. The goal of each agent is to determine the most suitable sellers, by satisfying some customer-specified constraints such as the highest (or the lowest) acceptable price and the date by which to complete the transaction. Kasbah incorporates also a mechanism, called Better Business Bureau that, upon completion of a transaction, allows both parties (customer and seller) to rate how well the other party managed its half of the deal. This way, a sort of reputation is defined for each user; this can be exploited by the agents in the choice of the parties to consider for the next deals.

- **Karacapilidis and Moraitis.** In the approach of [75], agents can take the initiative to contact their owners in order to start transactions that seem to be promising, or can trigger an owner action (e.g., they can inform their merchant that a specific offer has been of no interest in the market during the last month). This framework assumes a long, or even permanent, presence of the agent in the e-market. During this time the agent maintains a profile of its associated owner. These agents are capable of refining transaction criteria, handling incomplete, inconsistent and conflicting information and, finally, performing a progressive synthesis and a comparative evaluation of the existing offers in the virtual marketplace.

- **Wang and al.** In [172], an agent-mediated shopping architecture is presented whose aim is to improve the performances of the merchant information gathering activity. Generally, since the amount of potentially interesting information to examine is particularly large and available time is limited, the agents of a system that supports merchant brokering cannot perform an exhaustive search for gathering merchant information. In this context, the system presented in [172] allows to obtain a trade-off between the quality of the information gathering process and the time spent for obtaining it. Authors propose an anytime algorithm for handling such a trade-off. Anytime algorithms is a class of algorithms that can be stopped at any moment, but the quality of the yielded results improves gradually as the computation time proceeds. They are particularly useful for solving problems where the search space is large and quality of the results can be mediated. In the proposed trading ar-
chitecture, the agent autonomously gathers merchant information and buyers can inquire the current lowest price at any time. Buyers can also wait for a price decreasing process with the elapsing time.

4.2.5. Agents supporting Site Visiting

A customer, after having determined the best merchants whom to buy from, generally visits the merchant site for retrieving more accurate descriptions of the goods she/he is interested and/or examining other offers that might meet her/his needs. Information on customer navigation can be exploited by a suitable agent for constructing a customer profile non-intrusively. Such a profile plays a key role in supporting a customer in the search of those products appearing to be the most promising ones according to her/his interests, as well as for helping merchants to propose the most appropriate offers to the customers. Thus, from our point of view, the site visiting stage is central in the agent-supported e-commerce process since, during this stage, most of the information necessary for supporting the other stages is collected.

The management of the site visiting stage implies to deal with some important problems, namely: (i) the presence of semantic heterogeneities among the various Web sites and (ii) the necessity to adapt site presentations to customer profiles. In the following sections, we overview some recent proposals for tackling these two problems.

Agent-based approaches handling semantic heterogeneity

In this section we analyze three recent agent-based approaches for B2C e-commerce capable of handling semantic heterogeneities. The first two of them, namely B-SDR and MOMIS systems, can be also used to support the product brokering stage of the E²-CBB model; vice versa, the third system, called MORPHEUS, applies also to the merchant brokering stage.

- Mobile agents using MOMIS. MOMIS (Mediator environment for Multiple Information Sources) [14] 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. Figure 8 depicts the application of MOMIS to the classical product brokering problem. Instead of manually performing a lot of queries on several Web pages for retrieving information about a product (e.g., a car satisfying certain requirements), MOMIS first retrieves information from various interesting data sources (e.g., Ford, Volkswagen, FIAT, and so on) and, then, integrates it in a unique homogeneous view called Virtual Catalogue. The MOMIS component carrying out such a task is called SI-Designer [12]. A Virtual Catalogue is, therefore, a tool presenting, in a unified view, product information derived from heterogeneous catalogues. So doing, customers do not need to interact with multiple heterogeneous catalogues but with a unique one, uniformly representing all of them. After that a Virtual Catalogue has been constructed, MOMIS extracts a list of products fulfilling customer requirements from it. [13] extends this approach by introducing mobile agents in the MOMIS framework for allowing the autonomous management and the coordination of both the integration and the querying activities. Mobile agents can significantly save bandwidth by moving to the resources they need and executing their code therein. In addition, mobile agents are capable of dealing with non-continuous network connections and, consequently, they are intrinsically suited for mobile computing systems.

B-SDR Agent. The B-SDR Agent, proposed in [25,140], is a multi-agent system for representing and handling e-commerce activities. In such a system, an agent is present in each e-commerce site, handling the information stored therein. Furthermore, an agent is associated with each customer, handling her/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 handle behavioural information, (i.e., information about the behaviour of a customer on accessing data sources).

For each customer, a profile is constructed and maintained, storing information about the visits she/he carries out at the various e-commerce sites. Whenever a customer accesses an e-commerce site, the associated agent updates her/his profile. This operation is carried out by deriving similarities between concepts present in different e-commerce sites; each group of similar concepts is represented by a unique concept in the profile. When a customer accesses an e-commerce site, the site agent sends its B-SDR network to the customer agent. This latter activates a function for computing semantic similarities between portions of the B-SDR network associated with the site and portions of the B-SDR network representing the customer profile. Each of these similarities represents an issue of interest for the customer which is present in the e-commerce site. For each interesting product thus selected, the site and the customer agents cooperate for presenting to the customer a B-SDR network illustrating the offers of the site for that issue. This provides the capability to support the customer in searching, in the e-commerce sites, those products best satisfying her/his interests. Since the computation of semantic similarities between the e-commerce site and the customer profile is carried out at the customer side, no information about the customer profile is sent to the e-commerce site, thus preserving customer privacy.

MORPHEUS. MORPHEUS [178] is a comparison shopping agent that automatically collects product descriptions from a group of on-line stores on user's behalf. Since the stores are heterogeneous, a wrapper must be built and maintained for each store. In particular, MORPHEUS comprises: (a) a wrapper generator, that is a learning module that constructs a wrapper for each store; (b) a wrapper interpreter, that is a module that executes the wrapper for collecting product information from the corresponding store; (c) an output generator, that integrates search results from several on-line stores and produces a unified output.

The MORPHEUS approach is based on two important assumptions, namely (i) a proper keyword is provided for each test query; (ii)
each product description contains the price attribute which plays a key role in product search. Authors plan to introduce XML technology in their system in the future. In this case, the wrapper interpreter will be simply an XML interpreter and the DTD will contain the ontological information about specific domains.

Adaptation and Personalization of e-commerce sites' presentations An important issue in the B2C e-commerce concerns the presentation of site contents. Indeed, a static product presentation, i.e., a presentation that is the same for all customers visiting the site, does not always meet customer expectations. In fact, various customers might have different preferences for multimedia elements, and might be interested in different information concerning the product. It is, therefore, important to provide e-commerce applications with tools able to adapt the product presentation to the specific customer currently visiting the site. Several agent-based approaches for adapting site presentations have been proposed in the literature (see, e.g., [160, 86, 4]). In order to provide an idea of their behaviour we describe below two of them.

TELLIM [72] focuses on a user modeling mechanism that monitors the behaviour of the customer during a visit to a site and exploits collected information for constructing adaptive presentations. As an example of how this system works, Figure 9 shows in the center how a car, in a car factory site, is presented at the beginning of a session. Depending on the user’s behaviour, the next presentation is generated. Assume that the user has chosen the text in the first presentation and has not enlarged the image, then the next presentation would be generated like the image in the left. Vice versa, if the customer has ignored the textual links, but has enlarged the image, the presentation of the next car would be like the image in the right. This means that every product information is visible to the customer, but preferred elements are presented in the foreground and presumably less interesting elements are presented only as links in the background.

SETA. Besides supporting product brokering (see Section 4.2.2), SETA [7], identifies also customer preferences and interests and tailors product presentation to them. In this architecture, a Dialog Manager, handling the interaction with the customer, invokes a Personalization Agent and specifies to it the page typology to be produced as well as the product to be described for that particular visiting customer. The Personalization Agent applies a set of customization rules and dynamically generates the page to present to the customer. All the decisions about the layout and the content of a page are made on the basis of the characteristics stored in the customer profile. Thus, different pages may be produced when the same product is presented to different users.

Agent-based approaches handling adaptivity to the device It is worth to observe that, with the evolution of devices for the interaction with information services, adaptivity to customer profiles is not enough, and the access to these services depends also on device characteristics and on the context (e.g., work, home, holidays, etc.) where the customer is currently operating.

More in particular, with the development of information technologies, ubiquitous network environment (UNE) has been emerged, which includes ubiquitous network and ubiquitous devices. In such an environment, ubiquitous networks are composed of mobile network, Internet, PSTN and other wireless access network, such 802.11 WLAN, BlueTooth, Infrared communication, etc., while ubiquitous device includes general PC, PDA, handheld, and other devices that can connect with UN. The increasing amount of Internet accessing non-PC devices, as well as the fast development of mobile web applications produced the necessity of device independence as a desirable feature in web site construction. The purpose of device independence is that the web site creator can develop web resources and services to fit all kinds of devices. In this way, any kind of device can access web resources and get the most suitable presentation. The device independence principle, obtained by exploiting user agents, is analyzed in detail in [170] of W3C. B2C e-commerce is particularly interested in this issue, since the possibility of capturing customers by means of a proper product presentation is felt as essential by the e-sellers. As an example, in [41], an agent-based approach supporting presentation adaptivity is proposed that, besides user preferences, behaviour and experience, considers other features such as customer location, context and emotional state as well as technical features of the exploited device. This information is used to personalize the way the information is accessed and presented (e.g., browsing in a large
information space, searching for some specific data or receiving fast and well focused “hints”). Authors see the environment as a distributed information source in which Personal Presentation Agents select the information that is appropriate to the user within the given context and organize it in a clear, coherent and complete way. Another interesting approach is presented in [62], in which some mechanisms for adapting navigation support to device characteristics and its context of use are designed, thereby considering that user goals and the resulting expected navigation behavior might be subject to change.

The above approaches, and many other as [57, 55, 54], that are designed for general web applications, might be fruitfully applied in E-Commerce contexts. In [57], authors present a web site model and publication platform, called WebUnify. This platform allows to adapt the published resources to different device, such as PC, PDA, and Handheld.
Authors propose an ontology-based approach, by building a conceptualized model of the site and creating web resources as its instances. In this approach, content authoring, style designing, content processing and web site distribution are separated from each other. Authors also consider the screen adaptation as a sub-issue of device independence and implements the construction approach in an experimental system named Orion. Moreover, a device independent web site, in which the feedbacks are adaptive for the presentation capabilities of variant devices, has been built as a test-case.

Other works, as [55,54], describe new approaches providing adaptable client interfaces for web-based information systems. Developers specify web-based interfaces using a high-level mark-up language based on the logical structure of the user interface and, at run-time, this single interface description is used to automatically provide an interface for multiple web devices, as well as highlight, hide or disable interface elements depending on the current user and user task.

In [153] a selection-based approach, where the user selects interesting concepts by a textual interface, is presented, that can be adapted to a number of devices with heterogeneous capabilities, including mobile phones and PDAs. The simplicity of the selection tasks and the use of text broadens the range of devices that can be targeted. Device adaptation can be implemented by simply mapping the length of terms and relation lists to the device characteristics.

In [56], authors examine the possibilities of applying of independent user interface descriptions in E-Commerce contexts, that can be deployed on any device to reduce the development time and cost as well as to increase the maintainability and accessibility of the supported services.

Also, it is worth pointing out that, in the context of device adaptivity in B2C E-Commerce, another important objective is to reduce the user-perceived latency in web content delivery. Many techniques have been proposed in the literature, as that presented in [113], that describes the development of a user adaptive content delivery mechanism that integrates transmission time control and transmission order control. Based on the user’s preference profile, such a technique dynamically prioritizes inline objects in terms of content quality and delivery order, and provides user with adapted inline objects. A prototype system based on this technique has been implemented in Java, and gives an example of adapted content delivery.

4.2.6. Agents supporting Negotiation

Negotiation is the E2-CBB stage where the product price is determined. Negotiation is used in various commerce fields such as stock markets (e.g., NYSE and NASDAQ), fine arts auction houses (e.g., Sotheby’s and Christie’s), flower auctions (e.g., Aalsmeer, Holland) and other diverse businesses such as car dealerships and commission-based electronic stores. There is an evident benefit in dynamically negotiating a price for a product instead of fixing it; in fact, a dynamic negotiation allows to determine the price directly into the marketplace instead of forcing the merchant to fix it a priori. On-line negotiation is easier to implement than real-world one; in fact, some impediments, such as the necessity for all parties of being geographically co-located in an auction house and the increment of the transaction cost, that occur in physical world negotiation, disappear in the Internet world. Systems like OnSale [120] and eBay’s AuctionWeb [8] do not require that participants be geographically co-located.

As an example, in Fastparts, members buy and sell electronic parts, products and accessories from one another, directly and anonymously. In the ComponentConnectTM of fastparts, members post bids-to-buy and offers-to-sell in records that contain all the information about what is offered or bid upon. Fastparts automatically matches buyers to sellers at an online auction. In the Auction venue, prospective buyers place winner-take-all bids on items that the seller offers for sale.

The automation of negotiation can significantly reduce necessary time, making it possible for a large amount of transactions to be performed in small amounts of time, and may also remove some of the human reticence to engage in negotiation [99].

In an agent-based negotiation, agents both prepare bids on behalf of the buyers and evaluate offers on behalf of the sellers they represent. This is done according to some negotiation strategies, and by following some negotiation protocols. The negotiation protocol defines the rules the agents must follow, determining, at each time, what an agent can say and whom it can say to. We follow the approach introduced in [61] and, in the next subsections, classify agent-based approaches supporting negotiation on the basis of the protocol they use.
**Auctions**  Auction is today the most exploited negotiation protocol in e-commerce systems. In the Internet Auction List [66] there are currently listed more than 2,500 auctions, and various guides exist on the Web that list and categorize auction sites (such as, for instance, [27] and [63]). In an agent-based auction, there are two types of agents, namely *auctioneers* and *bidders*. One or more auctioneers initiate the auction, and the bidders make bids, according to the rules determined by the auction protocol. At the end of the auction, a winner is determined among the bidders (for the most fundamental concepts about auction see [85], and for a complete taxonomy of auctions see [176]), however it is possible to classify the existing methods into two main categories, namely the *single-sided auctions* and the *double-sided auctions*. In a single-sided auction, either bids are solicited from buyers by a monopolist seller for items to be sold, or offers are solicited from sellers by a monopsonist buyer for items to be bought. In contrast, double-sided auctions are auctions in which several buyers and sellers submit bids and offers simultaneously in the same marketplace. The interested reader can find in [89] a complete description of the main auction methods belonging to those two categories.

**Bilateral Negotiation** Differently from an auction, that is a form of one-to-many, many-to-many or many-to-one negotiation, bilateral negotiation is one-to-one. Here, a supplier and a customer search a mutually acceptable agreement over the terms and conditions of a trade, involving several contract attributes such as price, quality, delivery date and so on. In the literature there exists a large number of agent-based approaches, tackling bilateral negotiation, that propose different strategies for performing a good negotiation. [61] classifies these approaches into three groups, namely:

- **Decision making by explicitly reasoning about the opponent’s behaviour.** Agents belonging to this group observe and reason about the behaviours of their opponents, in order to decide what behaviour they have to assume as a response. Examples of such an approach are:
  
  * Vidal-Durfee. [169] presents an approach for determining when an agent should behave strategically (i.e. by considering the behaviour of the other agents), and when it should behave as a simple price-taker buying at the lowest possible price. Authors propose a framework for developing agents with incremental modeling/learning capabilities and show how, in some circumstances, agents benefit by building and using models of the other agents in the community.

  * Bazaar. In [181], a sequential decision making model for agent negotiation, called Bazaar, is proposed. Bazaar models agent beliefs about the negotiation environment by means of a probabilistic framework that uses Bayesian learning. More specifically, a bayesian network is exploited to update the knowledge each agent has about the other agents and the environment, and bayesian probabilities are used for leading the negotiation among agents.

- **Decision making by finding the current best solution.** Agents belonging to this group try to find the negotiation solution that maximizes their profits, taking into account existing constraints, the opponent’s behaviour and the current negotiation situations. Some proposals following such an approach are:
  
  * Luo and al. In [100], authors propose a fuzzy constraint based model to support a bilateral, multi-issue negotiation. In this model, agents try both to conclude a good deal for the two involved parties and to maximize their own profit, showing a semi-competitive behaviour merging both competitive and cooperative attitudes. The key idea of such an approach is that, if an agent proposes an offer and the opponent agent does not accept it, the proponent agent has to find an alternative offer that it considers equally acceptable, but that is also more acceptable to the opponent agent. If such a trade-off solution is not available, the proponent agent would make a concession. The model is based on the exploitation of prioritized fuzzy constraints that both represent the trade-offs between the different possible values of the negotiation issues and indicates how to make concessions when this is necessary. The proposed model ensures that the two agents reach a trade-off solution if it exists.

---

1 The term "monopsony," first used in a print by Joan Robinson (1969), means a single buyer in a market.
* FeNAs. Fuzzy e-Negotiation Agents (Fe-NAs) [88] handles a multi-issue negotiation in presence of limited common knowledge and imprecise constraints and preferences. It aims at finding a negotiation solution that maximizes agent’s utility subject to its acceptability to other agents; in order to carry out such a task, it exploits principles of utility theory and fuzzy constraint-based reasoning.

* Matos and al. In [106], a service-oriented negotiation model is presented. A service-oriented negotiation is a real-world business process management system in which the agents need to agree on who should perform a particular service and under what terms and conditions. In the proposed framework, autonomous agents adapt their strategies and tactics to the evolving circumstances by using an evolutionary approach in which strategies and tactics correspond to the genetic material in a genetic algorithm.

* Kasbah. In the Kasbah system [76], a straightforward mechanism of bilateral negotiation between buying and selling agents is provided. After buying and selling agents have been matched, the only valid action in the negotiation protocol is for buying agents to offer a bid to sellers. Selling agents respond with either a binding yes or no. With this protocol, Kasbah provides a buyer with three possible strategies, namely anxious, cool-headed and frugal, corresponding to an exponential, quadratic and linear function, resp., for increasing bids for a product over time. Such a negotiation mechanism is very intuitive to understand for the involved parts, and this has determined a general user acceptance of it, as observed in some Media Lab experiments [32,76].

* Tete-a-Tete. Differently from most other on-line negotiation systems, which behave in a competitive way, Tete-a-Tete [166,58] operates a cooperative negotiation among the agents. Such a negotiation ranges across multiple terms of a transaction such as warranties, delivery times, service contracts, gift services, loan options, return policies and so on. Tete-a-Tete does not use simple raise or decay functions; on the contrary, similarly to [127], it exploits an argumentative style of negotiation with sale agents. The evaluation constraints determined during the product brokering and merchant brokering stages are exploited as dimensions of a multi-attribute utility which allows a customer agent to order merchant offers w.r.t. their capability to satisfy customer preferences. It is worth pointing out that, in this way, in Tete-a-Tete product brokering, merchant brokering and negotiation phases are strictly integrated.

* V-Market. In [135] the authors propose an approach to the design of agent mediated e-commerce systems, through the definition and implementation of an object-oriented framework, mainly dealing with e-commerce applications based on virtual marketplaces. A virtual marketplace is an Internet based system in which software agents interact and negotiate on behalf of their respective users, to buy, sell or find specific goods and services. In this type of system, all users can be potential buyers, sellers or both, depending on their specific interests. The main goal of V-Market is to facilitate the creation of this type of applications, as well as to make them more robust and flexible. V-Market allows developers to customize virtual marketplaces, and define transaction categories on demand, incorporating many possible products and services that can be traded online. Users can create new transaction types and items based on individual needs; they can define customized software agents handling the new offered products and services. Software agents in V-Market pro-actively broker and negotiate with interested buyers and sellers represented by their own agents. They can be customized with any set of desired behaviors, thereby enabling the consumer to have a virtual presence in the marketplace looking for interesting goods, relieving customers from the need of constantly monitoring the market. The philosophy underlying V-Market is similar to that inspiring Kasbah [32,76] in that the system concentrates mainly on the ability to create applications allowing buying and selling agents to interact and negotiate.
In [46] a formal bilateral negotiation model is presented. In this model offers and counter-offers are generated by linear combinations of simple functions, called tactics. It is possible to vary the importance of a given modeled criterion by using different weights in the linear combination. Authors use the term strategy to denote the way in which agents change the tactics over time. The model defines a number of tactics which agents can employ during the negotiation process and indicates how an agent can adopt various strategies by changing these tactics over time.

MAUT. [9] proposes a generic negotiation architecture that uses the Multi-Attribute Utility Theory (MAUT) principle to reach agreements that satisfy multiple independent objectives. In particular, the architecture aims at both giving a constraint optimization formulation to the MAUT principle and using a constraint optimization solver to find the best deals. Retrieved deals are, then, proposed to other agents via a second component supporting conversational interactions. The system provides also a systematic constraint relaxation protocol allowing an agent, when receiving proposals that are disjoint from what it can currently accept, to generate the most acceptable deals. This approach is built on top of a Negotiation Engine, a generic architecture for coordination and negotiation. The system is currently used to automate negotiation in the electronic components domain.

Paurobally and Cunningham. In [128], the authors propose to represent a shopping scenario between a retailer and a customer by using an adaptation of dynamic logic. Agents are able to determine where negotiation processes arise and the negotiation model which they comply with. Authors provide an abstract theory of the states involved in the shopping scenario and derive the paths leading to a particular shopping state. Moreover, they give several negotiation models that can be followed by agents engaged in a negotiation process; these are exploited by a particular dynamic logic for deriving the possible negotiation paths. From an initial state, an agent can follow a set of execution paths to perform the negotiation process and end in a state that satisfies its goal. Stored paths form the plans of the agent. The individual goals and intentions of the agent will determine the chosen paths.

When a provider and a customer negotiate there is already some common knowledge that the others have about each other. This common knowledge of the other’s intentions also plays a role in the choice of the optimal path. As the negotiation progresses, each agent can build up more beliefs about the other agents. With its beliefs and goals, an agent can use its logical analysis of paths to seek the best way to satisfy its goal.

The advantage of exploiting a dynamic logic consists in providing a precise language for describing and reasoning about actions and states; this allows to handle various important tasks e.g. inconsistency checks. The approach is also significant in that it takes into account other interesting problems, such as concurrency.

Lin. In [97], authors present a research model for influencing factors of pricing decision for an organizational procurement. The model, representing the evolution of organizational market, is built based on literature review of inter-organizational negotiation, interviews with domestic industry, and empirical data analysis. The relationship between framing, personal risk attitudes, performance of decision choice and intuitive preference of pricing decision for an organizational procurement are proposed to assist the trades between industries.

Argumentation. Agents belonging to this group try to exchange additional information when they interact in a negotiation process. In such a context, for example, an agent a that decides to reject the offer of another agent b, can provide b with a motivation of its decision, explaining in detail why the proposed offer cannot be accepted (e.g., the delivery time is too long). Examples of such an approach are:

Sierra and al. In [155] authors provide a formal framework in which agents can op-
erate persuasive negotiation for modifying beliefs and preferences of each others, using an expressive communication language. Authors define the static aspects of the negotiation process, as shared ontology, social relations, communication language and protocol. Furthermore, they propose a negotiation model able to recognize different types of arguments that agents can supply to support their proposals and to indicate how these arguments can be generated and interpreted by agents.

* Giacomini. In [52] the author proposes a multi-agent system where agents perform argumentation activity based on both knowledge stated in their knowledge-bases and acquired from other agents. The approach is based on the observation that in a negotiation process, different arguments may contradict with each other; this implies that an important problem is the computation of the defeat status, namely, determining which arguments emerge undefeated from the conflict and which conclusions should be believed. The paper presents a distributed approach to argumentation and proposes two self-stabilizing algorithms for the computation of the defeat status.

4.2.7. Other Interesting Issues: Product Configuring Support and Diversity-Based Recommendation Strategies

Product Configuring Support In the last two decades we have seen, on the one hand, an overwhelming growth of the demand for customer-individual, configurable products and, on the other hand, a constantly improvement of the required supporting software systems. Consequently, a lot of tools based on different AI techniques (e.g., Constraint Satisfaction [73]), are today available on the market, that can be used to avoid the reduction of the order lead time and of the number of faulty configurations. However, more and more complex products, assembled from configurable sub-products supplied by highly-specialized providers, are nowadays offered to the customer as integrated solutions, thus the necessity of new technologies able of adequately distributing and configuring heterogeneous integrated systems have to be provided. The main problem for the customers of these integrated systems is that the increasing complexity of products makes the interaction with configuration systems lengthy, due to the fact that many configurable parameters have to be specified during the configuration process.

From this the necessity arises to create configuration procedures usable not only by technical people, but also by less experienced customers: consequently, these systems should be extended with user-adaptive interfaces, which tailor the interaction to the individual user, guiding her through the configuration process in a personalized way. Different recent product configuration support technologies cover the above issue [134,159]. As an example, a European project called CAWICOMS [6] aims at enhancing the state of the art in configuration systems in order to take such issues into account. Within this project, an environment called CAWICOMS Workbench has been realized, that supports the creation of user-adaptive Web-based configuration systems. CAWICOMS Workbench offers both a configuration shell supporting distributed configuration and personalization and a set of tools for the specification of the domain-dependent knowledge about the products/services to be configured and the types of end-users to be served. The shell is based on advanced distributed configuration and adaptive hypermedia techniques for the management of personalized configuration interactions involving the cooperation between suppliers.

It is worth to point out that, in the CBB model we have proposed, when there is no single product which matches the needs, it is possible that such a product can be assembled in the sense described above. In such a case, the model should be revised for taking into account also a product configuring step.

Diversity-Based Recommendation Strategies Recommender systems can be seen as a potential solution to the information overload problem that are particularly well-suited for B2C e-commerce applications. In fact, they help users to navigate through complex product spaces in pursuit of suitable products by presenting sets of alternative recommendations (during a series of recommendation cycles) and taking advantage of user feedback to guide future cycles. As we have seen, recommender systems are used both in product- and merchant-brokering stages and, in these contexts, they usually adopt a similarity-based recommendation strategy, selecting cases for recommendation on the basis of maximal similarity to the cur-
rent user query. However, recently this emphasis on similarity has been questioned. In particular, it has been shown [158] that a set of recommendations may be similar to the current query but if they are also very similar to each other, i.e. if they lack diversity, then these recommendations provide only limited coverage of the recommendation space. Consequently, many techniques for improving the diversity of a set of recommendations without affecting similarity to the user query have been proposed [158,121,38]. In particular, in [121], an approach is presented that tries to improve the quality of recommender systems by validating these findings in the context of more complex models of collaborative filtering, as well as by demonstrating that such techniques also preserve recommendation diversity, one of the key issues affecting traditional recommender systems.

Another aspect in using diversity is the possibility to combine it with critiquing. Critiquing is a well-known form of user feedback in case-based recommender systems. A critique encodes the users preference in relation to a particular feature (e.g., in a digital camera recommender a user may be allowed to indicate whether they are interested in cameras with a lower resolution than the one currently presented; so 'lower resolution' is an example of a critique over the resolution feature). Recent research [107] demonstrates how the dynamic generation critiques that operate over multiple features can deliver significant performance improvements in recommendation systems.

5. The Role of Semantic Web Services in B2C Applications

The main problem deriving from the overwhelming amount of information contained in the Web today, is that such information is not directly available for automatic computation, as it would be desirable in many application contexts and, in particular, in agent-based e-commerce applications. As noticed in [162], two main issues are recently emerged relative to the possibility of realizing a more powerful exploitation of Web contents: On the one hand, the Semantic Web can provide the tools for the explicit markup of the content of Web pages; on the other hand, the development of Web Services gives to the Web users the possibility to exploit programs acting as independent agents that become the producers and consumers of information and enable automation of business transactions. In such a context, many research has been recently developed for bridging the gap between the present state of the Web and the Semantic Web providing Web services. This researches propose the vision of Web services as autonomous goal-directed agents which select other agents to cooperate with, acting at times in client server mode, or at other times in peer to peer mode. Furthermore, these Web services exploit agent ontologies to automate both tasks and agent interactions. In other words, Semantic Web services use the Semantic Web to both realize an effective information discovery capability and provide a fruitful interoperability. In order to achieve these objectives, it has been necessary to develop suitable formal languages for representing and reasoning with core concepts of Web services. The main languages in this context are the following:

**OML (Ontology Markup Language)** [119] is an ontology specification language based on Conceptual Graphs [37]. It allows the representation of concepts organized in taxonomies, relations and axioms in first order logic.

**CKML (Conceptual Knowledge Markup Language)** [33] is an extension of OML, by the addition of theories, theory morphisms (which model concrete conceptual scales), and infomorphisms (which model realized conceptual scales). While OML defines the types for nouns and verbs, the theories in CKML offer a specification form for the modifiers, adjectives and adverbs. Moreover, the theories in CKML offer a means for the specification of "controlled vocabularies." Finally, CKML introduces the distinction between a theory (abstract conceptual scale) and a theory interpretation (concrete conceptual scale) that corresponds to the distinction between a terminological ontology and an axiomatic ontology.

**DAML+OIL** [39] provides modelling primitives commonly found in frame-based languages (such as an asserted subsumption hierarchy and the description or definition of classes).

**OWL (Ontology Web Language)** [122] has been conceived for defining and instantiating Web ontologies. An OWL ontology may include descriptions of classes, properties and their instances, and the OWL semantics specifies how to derive logical consequences from a given ontology.

**Semantic Web Rule Language (SWRL)** [161] is a
combination of two OWL sub-languages with the Unary/Binary Datalog RuleML sub-languages of the Rule Markup Language. This approach extends the set of OWL axioms to include Horn-like rules, that model causal implications. Thus, in SWRL it is possible to combine Horn-like rules with an OWL knowledge base.

In the last five years, many research on E-Commerce has dealt with Web Services and the Semantic Web (see, for instance, the Workshop on Business Agents and the Semantic Web - BASEWEB ’05 - and the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service). In particular, in [123], authors describe how to use the language DAML-S to control the interaction between Web services. DAML-S defines a DAML+OIL ontology for the description of Web services which provides: (i) the description of the capabilities of the Web services to specify what service is provided; (ii) the specification of how the Web service accomplishes its task to specify in details how the service achieve its goals, and what are the requirements of potential requesters that wish to interact with it; (iii) the specification of how the information exchanged by the Web service and requesters maps into actual messages exchanged by the different parties. The approach uses DAML-S to describe capabilities of Web services so that they are able of finding each other on the basis of the information that they provide, rather than referring to incidental properties such as their name, port, or a free text description of what they do; moreover, in the paper is also showed how DAML-S can be exploited to control the autonomous interaction between Web services without any need of pre-programming neither the sequence of messages to exchange nor the information to be transmitted. The work [95] shows how Semantic and Web Services technologies can be used to support service advertisement and discovery in e-commerce. In particular, the authors describe the design and implementation of a service matchmaking prototype which uses a DAML-S based ontology and a Description Logic reasoner to compare ontology based service descriptions. This approach also shows how ontologies can be used to describe services so that agents (both human and automated) can advertise and discover services according to a semantic specification of functionality. The work [36] deals with the possibility of representing the complex data structures needed to support electronic commerce applications in the semantic web using ontologies. The conventional subtype-oriented refinement of the ontology is supplemented by a method of coordinated refinement based on category theory. Authors show how the combined methods make ontologies a much more powerful tool for organizing the semantic web.

6. Conclusion: Future perspectives and new challenges

Due to the increasing difficulty for human users to manually manage the overwhelming amount of information present in e-commerce sites today, the exploitation of software agents for automating the various phases involved in e-commerce processes seems to be particularly well suited. We have modelled the B2C e-commerce stages by using the E²-CBB model, that extends the E-CBB model by including the site visiting stage. Furthermore, we have examined a large number of agent-based approaches for supporting B2C e-commerce activities that have been proposed in the literature in the last decade, and we have provided a taxonomy of them w.r.t. E²-CBB.

In our opinion, three are the crucial issues for the future development of agent technology in B2C e-commerce arise:

- The site visiting site, that we have introduced in this work, is becoming more and more important with the increasing need of e-commerce service personalization. Such a task can be performed in a non intrusive way by exploiting techniques able to derive the profile of a customer by monitoring her/his activities on the Web. The development of these techniques is ongoing, and several current research already produced solutions to problems involved with this issue.

- New needs about personalization are also arising, such as that of adapting the content of the e-commerce sites to the device (WAP, palmtops, etc.) used by customers. This issue leads to face some technological problems mainly studied in the User Modeling domain in the past.
- Last, but not the least, the overwhelming production of both logic-based and expert systems-based approaches for supporting the negotiation stage in a more and more completely automated way, points out the great interests to realize agent-systems showing “human-like” capabilities, such as prediction and autonomy, for handling such a stage.

We believe that the aforementioned issues will be the key subjects for future research in this area.

References


