

## Agents and P2P Combined For E-commerce

Nader Troudi and Peter Kropf

Department of Computer Science and Operations Research

University of Montreal

Montreal, Quebec, Canada, H3C 3J7

troudi@newtrade.com, kropf@iro.umontreal.ca

**Keywords:** Agent, Peer-to-Peer, multi-agent systems, E-commerce application.

### ABSTRACT

Electronic commerce technology is more and more present and people are getting connected together. Companies and users want to automate their task and discover new business opportunities. Many standards and initiatives have been already proposed. Most of them focus on business-to-business protocol definition. Rosettanet, cxml, xcbl and bizTalk are only a few examples. Users and companies then have a set of protocols they need to understand and master for implementing solutions. With the increasing number of protocols, companies need a solution permitting them to easily integrate different protocols. Unfortunately, the commercial solutions provided today do not provide sufficient support for integration.

Software agents have already been proposed as a key technology for connecting people and adding new features in electronic trading. However, most of the agent-based solutions already proposed have not really changed the user's experience, neither have they brought a major shift from traditional activities in classic markets. This is mainly due to their inflexibility. In fact, these solutions predefine rules and policies in ways that users feel uncomfortable and are limited in their actions. Agent-based systems seem to appear as a key technology that gives users more autonomy and that is yet flexible enough to integrate new business scenarios. In this paper we propose an agent-based solution that could be used to wrap in existing systems and applications. We show how it could be easy to support new protocols without being forced to rewrite solutions from scratch.

The work proposed combines both, advantages from software agent technology, and from P2P networking technology. The solution has four layers where each one encapsulates a specific set of functionalities: communication, business, cooperation and coordination. Users can adapt or modify the internal structure of one layer without necessarily affecting the others.

### INTRODUCTION

The continuous development of Internet technology and the increasing number of people getting connected to it, explains the many efforts of attempts to create virtual markets. In such a market, agents can negotiate, buy and sell goods. Those agents could either be controlled by humans or be autonomous. Many projects have proposed intelligent agent-based systems supporting many aspects of e-commerce. For example, many agent-based online shopping services have been created [1,2,3,4,5]. Despite their added value for the user shopping experience, they lack facilities for automated negotiation and agent cooperation.

Other more sophisticated applications have been proposed. Kasbah [6] is a Web-based system where users can create autonomous agents to buy and sell goods. Doing this, users do not need to stay connected or watch their agents. However, Kasbah can be qualified as a closed system. It is a proprietary system, which uses a very limited message protocol. Thus, it is impossible to integrate new heterogeneous agents.

MAGMA is an agent-based electronic commerce architecture, which focuses on the components of traditional markets such as communication, transfer of goods, money handling and transaction mechanisms [7]. The MAGMA architecture models traditional market activity through a platform-independent API. It uses a central communication relay server, which could be the failure point of any industrial adaptation.

The traditional market and the centralized systems already proposed could be classified as classic solutions. These classic solution share the following list of limitations [8]:

- *Inflexibility:* every e-commerce solution has its own trading environment. Unfortunately, it's often impossible to combine concepts coming from more than one environment in the same transaction.
- *Only price counts:* most current systems, offer auctions or negotiation to select either buyer or seller. However, they use only the price as the key point as a selection criterion. Unfortunately, this is not the main criterion for striking a deal [9].

- *Centralization*: trader may want to have more control. Currently, the central coordinator has the last word. He decides what to do, how to match and when.
- *Static predefined behavior*: the proposed systems suppose that the environment is stable and predictable. Thus, they predefine the agent's behavior, its relationship and its trading partners.
- *Weak automation*: many operations are still not supported.

Overcoming these limitations is necessary to implement a realistic solution dedicated to an open environments. Open environments are characterized by having components that are (1) autonomous (acting independently), (2) heterogeneous (designed independently), (3) of dynamic membership (joining, changing, and leaving arbitrarily) and (4) of large scale (numerous). These properties are compatible with both the agent and the peer-to-peer paradigm. In fact, peer-to-peer (P2P) networks, consisting of nodes that are peers of each other, provide a suitable paradigm for implementing dynamic trading. Nevertheless, P2P tools are not able to exchange complex data and to deal with heterogeneity, coordination and data management problems. Agents are persistent computations that can perceive, reason, act and communicate [10]. Agent technology is of interest here because it adds more intelligence over the P2P technological layer. The combination of agent and P2P technology brings more autonomy and flexibility to the proposed e-commerce system. This paper proposes an agent-based integration system. The purpose of this agent system is to enable existing applications and infrastructures to conduct intelligent interaction with users or other business systems. The idea is to have a flexible integration framework, which facilitates dynamic partner selection and supports agents' heterogeneity.

The following sections explain the P2P paradigm and present a few P2P-based applications. Then a brief overview of related work is given. After that, we present our agent architecture for open e-commerce integration. The last section discusses the future work and direction of our research.

## THE PEER-TO-PEER PARADIGM

P2P applications have started to emerge, creating more and more hype, and taking the attention of the media. However, the P2P concept is not a new one. We can say that when two computers were first connected they formed the first P2P network. Mail servers, network news servers (NNTP), and domain name servers (DNS) operate in P2P networks. Today, many factors make P2P practical for a large number of applications. These factors include the explosion of connected devices, the rapid increase of affordable bandwidth, increase of computing power, larger

storage capacities, and the proliferation of information at the edges of the network.

P2P is a style of computing that makes the network interactions more symmetrical. Even though there may be centralized services (like the Napster catalog) the end user peer is a significant focus of the application. The distribution of the system reduces the dependency to network and communication failures. P2P computing takes advantage of existing desktop computing power and network connectivity in allowing economical clients to leverage their collective power to benefit the entire community. P2P networks replace the traditional centralized structure of client-server interactions with peer interactions, where each machine acts as both a client and a server in the network. Unlike the client-server architecture where a client generates a request and a sever responds to the request, peers understand both queries and responses. The peers have identical capabilities and responsibilities, and all communicate symmetrically. Napster is the good example of a monolithic centralization that causes all the P2P functionality to fail if the main server fails or is disconnected. Gnutella is the opposite because no single peer, in case of failure, will significantly affect the quality of the network. Peer-to-peer computing enables applications that are collaborative and communication-focused: it leverages available computing performance, storage, and bandwidth found on systems connected to each other in a world-wide network.

Today's best known peer-to-peer applications are Napster [11], Kazaa [12], Gnutella [13], Freenet [14] or SETI@Home [15], but various research projects have been initiated in the past few years, such as Pastry [16] and Chord [17]. Although the different P2P applications share the same notion of peer-to-peer networking, the intended usage and approach varies from application to application.

Napster, Kazaa and Gnutella are primarily file-sharing applications: exchanging of files between peers. Nester's approach to information search is traditionally client-server, while Kazaa and Gnutella adhere more to the peer-to-peer philosophy and forward information search requests to neighboring peers in the network by flooding. However, they both recently introduced super-nodes for more scalable information retrieval. Freenet is more like a distributed information storage system. It pools unused disk space across potentially hundreds of thousands of desktop computers to create a collaborative virtual file system.

SETI@Home is the world's largest distributed computing system with a capacity of an estimated 26 Teraflops/sec. If we consider that the main server is the equivalent of a very large peer, SETI could be considered a P2P system. The client peers contact the main server peer to report results and to request new data for processing. They don't need to stay connected as the average processing time could be as much as 14 hours.

Pastry is a scalable, distributed object location and routing infrastructure for wide-area peer-to-peer applications. It can be used to support a variety of peer-to-peer applications, including global data storage, data sharing, group communication and naming. Chord, on the other hand, focuses on a scalable peer-to-peer lookup service to efficiently locate the node that stores a particular data item. Chord provides support for just one operation: given a key, it maps the key onto a node.

The JXTA [18] project works on core network computing technology to provide a set of flexible standards that can support peer-to-peer computing. It is not an application and does not define the application itself. The focus is on creating basic mechanisms and leaving policy choices to application developers.

### Related Work

A number of projects combing P2P networking with agent technology have been proposed. Bussler [19] studied the difference between business-to-business integration (B2BI) and application-to-application integration (A2AI). He argued that the P2P paradigm is suitable for B2BI since it enables an organization to exchange messages without involving third parties. He proposed an integration server architecture for both B2B and A2A. He described many factors that should be taken into consideration while designing such a server. In this work however, we propose an agent based integration solution. With agents, the solution is more autonomous and flexible than the one presented by Bussler.

Youll [8] introduced the so-called Atomic market. The idea here is a P2P agent-based system for e-commerce applications. He proposed a modified version of the contract net protocol to support the dynamic negotiation between peers. This work has the following limitations: (1) no support for already existing system, it's not clear how to integrate them; (2) the atomic market uses a central registry (like a catalog) and binds communication language to Java objects; (3) it's not clear how the solution could support dynamic group creation and how peers (agents) should select others peers.

Coordination and cooperation are very important issues in designing e-commerce open systems. Lars Rasmusson [20] gives a survey on decentralized coordination techniques. He based his work on multi-agent research, coordination science and market oriented control by the use of virtual markets for resource allocation. Deugo [21] on the other hand, identifies a set of software patterns for agent coordination. Cooperation behavior of agents trading in open environments has been studied by Rasmusson [22]. He argues that agents are mostly selfish but cooperative and benevolent agents are still possible.

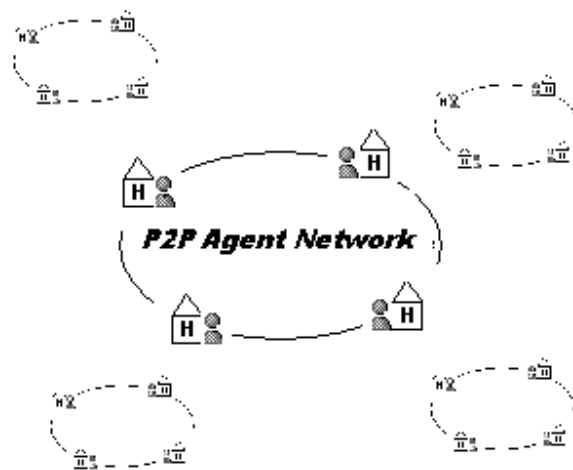
A lot of work is being done in the field of dynamic organization creation and maintenance [23]. This is a very important issue for e-commerce open systems where static

relationship and networks are not efficient. There is a need to be reactive and to adapt the organization to the reality of the exchanges. Mathieu and his group have proposed three principals to adapt the organization dynamically: "having an address book", "sharing knowledge" and "recruiting new able collaborators" [24].

Standardization of e-commerce activities and technologies is evolving continuously. Several organizations and companies are now working on the building blocks of e-commerce, especially in the B2B setting. Many of those developments are based on XML; like the Internet Open Trading Protocol, Common Business Language, Commerce XML, Open Buying on the Internet, XML/EDI, and Trading Partners Agreement Markup Language. In interoperability a number of different initiatives have started. EbXML is the latest initiative. The goal of ebXML is to provide a common platform for conducting electronic business and to integrate old frameworks such as eCo and BizTalk.

### AGENT ARCHITECTURE

This section describes in details how agents and the P2P paradigm will be used in e-commerce applications. As described in Figure 1, every company will be represented by an agent system in the P2P network. Thus, providers and users will have the possibility to initiate direct communication, thus bypassing intermediaries. P2P communication brings more flexibility and freedom to all the participants. Intermediaries will still be important in the network but as service providers. Participants will decide if they want to use this service or not. Using a service may require paying some fees.



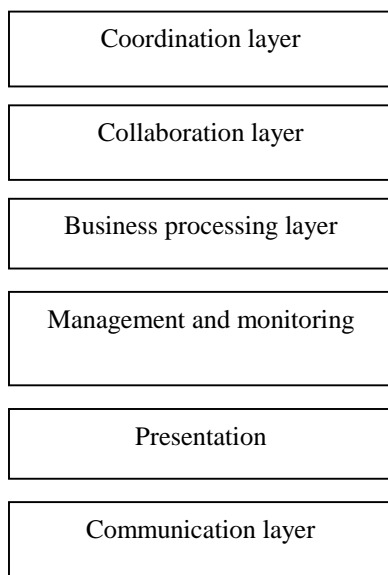
**Figure 1.** Agents communicate in a P2P network and enabling e-commerce activities.

Companies have already invested a lot of resources and money in their existing systems. They need a solution that still allows to leverage from their previous investments in information systems. We propose here a solution that could be used as a wrapper or a bridge between their internal systems and the P2P network. Every company could adopt this integration agent to enable intelligent and direct interaction with other agents. All the available solutions are centralized so companies and users are forced to pass through a central hub to communicate with each other. There are many reasons why users don't necessary like this interaction model. Apart from the fact that this central hub could be a point of failure, it could also be a business threat. In fact, users do not necessarily want to use some middleman that can know what you want from the "right person" and then make use of this information for his personal advantage. Moreover, this can be costly since the middleman may ask for a payment only for having helped you to get in touch with the right person.

Using this agent-based integration solution, companies could have more autonomy and more control over their business decisions and their partner selection. Companies and users will create their own relationship and maintain their partner network. Users or companies could use services of helpers, but without the need of central entity.

### The Internal Agent Architecture

We propose a layered architecture (Figure 2) for our P2P integration server. The main advantages of such architecture are the clear separation between each layer. Every layer has its own concerns and focuses on specific functionalities. Layers could be reused in different contexts. Moreover, a layer could be implemented in many different ways. Doing so, this architecture encourages easy integration, interoperability and standardization.



**Figure 2.** Every agent has different layers.

We think that four layers are necessary for our P2P agent architecture: the communication layer, the business processing layer, the collaboration layer and the cooperation layer. We also believe, for practical reasons of this solution, agent should have two others layers that could be seen as orthogonal functionalities: the presentation and the management layers. The role of each layer is discussed in the following:

- *Communication layer:* communication is the core element in most multi-agents systems. This layer abstracts from all the technical layers related to transporting messages from one agent to another one.
- *Business processing layer:* this layer concerns all the business level functionalities. It includes the business event handling, business workflow processing or the business logic.
- *Cooperation layer:* coordination can be considered as the bilateral dissolution of conflicts between agents stemming from their defined behavior. For software agents in electronic commerce, three scenarios of inter-agent co-operational behavior can be distinguished: benevolent, cooperative and selfish [22]. Benevolent or cooperative behavior can be noticed when agents share the same principal and have the same goal. Benevolent agents are those who help other agents to maximize utility even if their own utility decreases through this action. Cooperative agents are those, which collaborate even if the utility of the other agent knowingly increases more than their own. The first two behaviors can be assumed in closed or proprietary multi-agent systems with a single designer. This is not true for open marketplaces, where agents will act competitively on similar utility scales. Designers or creators will probably define their agent's strategy and goals according to their own ones. Agents are not assumed to always have the same cooperative behavior. They should have context awareness and decide by themselves whether to cooperate, not to cooperate or to act as benevolent.
- *Coordination layer:* coordination could be defined as the act of managing interdependencies between agent activities, given either a resource to be shared or a timing interdependency. Three reasons why a group of agents need to be coordinated can be identified [10]: (1) to manage interdependencies between their activities; (2) to meet global constraints; and (3) because one agent might lack all the competencies and resources to accomplish its tasks. For instance, software agents in an electronic marketplace need to coordinate their activities such as exchanging money vs. goods or managing services and access rights. This coordination process continues until every participant is satisfied with regard to his own utility maximization goal. An institution has to coordinate how much agent A will have to pay to agent B to satisfy the utility goals of both. This institution might be either of the agents, a

trusted third party, or the “invisible hand”. Organizational and economic theory offers the use of either more centralized or decentralized coordination concepts.

- *The presentation layer:* Agents should adapt the way they present their services according to the capabilities of their partners. Thus, when interacting with a cell phone peer they should use a more compact data representation. The agent could send more data when the partners have more bandwidth or a larger screen for example.
- *The agent management layer:* agents need to be managed and monitored. This could be done from anywhere and using many technologies. Remote management adds more flexibility and more robustness to the agent. Agents could be configured to send notification when they judge that they should. Notification concerns errors, failures, users events or application events. Users could configure all these notifications.

## CONCLUSION

Many agent-based electronic commerce solutions have been proposed. However, users complain about the lack of freedom these solutions leave to them and the centralized approaches they adopt. In this paper we proposed a distributed architecture dedicated for inter-agent interactions. Every agent is considered as an autonomous peer. These agents will easily wrap existing systems extending them to support new e-commerce functionalities. The agent architecture proposed in this paper has many layers. Each layer abstracts a set of functionalities and has many components. Our future work will concentrate on the dynamic behavior of each agent and mechanisms for dynamic group creation. We will also need to cover the issues related to security and trust relationship.

## REFERENCES

[1] J. Morris, and P. Maes. Sardine: An Agent-facilitated Airline Ticket Bidding System Software Demo. *In Fourth International Conference on Autonomous Agents (Agents 2000)*. 2000. Barcelona, Catalonia, Spain.

[2] Personalogic/AOL, <http://www.personalogic.com>.

[3] Krulwich, B. The BargainFinder agent: Comparison price shopping on the internet. *In Williams, J., ed., Bots and Other Internet Beasties*, pages 257-263. SAMS.NET, 1996.

[4] Excite Inc. Redwood City, CA. <http://www.jango.com>

[5] eBay’s AuctionWeb. <http://www.ebay.com/aw>

[6] A. Chavez, D. Dreilinger, R. Guttman, and P. Maes. “A Real-Life Experiment in Creating an Agent Marketplace.” *Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM’97)*, pages 159-178. London, UK, April 1997.

[7] M. Tsvetovatyy, M. Gini, B. Mobasher, and Z. Wieckowski. MAGMA: An agent-based virtual market for electronic commerce. *Journal of Applied Artificial Intelligence*, vol. 11, no. 6, pages 501-523. 1997.

[8] J. E. Youll. Peer to Peer Transactions in Agent-mediated Electronic Commerce. Master’s Thesis, School of Architecture and Planning, MIT, Cambridge 2001.

[9] E. Brynjolfsson, and M.D. Smith. Frictionless Commerce? A Comparison of Internet and Conventional Retailers. Aug. 1999, MIT Sloan School of Business. <http://ecommerce.mit.edu/papers/friction>.

[10] N. R. Jennings, "On Agent-Based Software Engineering", *Artificial Intelligence* 117 (2000), pages 277-296, Elsevier Press, April, 2000.

[11] Napster Inc. <http://www.napster.com>

[12] Sharman Networks Limited. <http://www.kazaa.com>

[13] Gnutella, <http://www.gnutelliums.com>

[14] Freenet, <http://freenetproject.org/lang/en/>

[15] <http://setiathome.ssl.berkeley.edu/>

[16] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *In Middleware 2001*, LNCS 2218, pages 329–350, Berlin, Germany, 2001. Springer-Verlag.

[17] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. *In Proc. of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM’01)*, pages 149–160, San Diego, CA, Aug. 2001.

[18] Sun Microsystems, Inc. Project JXTA: An open, innovative collaboration. White Paper, Apr. 2001. <http://www.jxta.org/project/www/docs/OpenInnovative.pdf>,

[19] C. Bussler. P2P in B2BI. Hawaii International Conference on System Sciences (HICSS) Big Island, Hawaii, vol. 9, pages 302-312. USA. 7-10 January 2002.

[20] L. Rasmusson. Decentralized coordination for open distributed systems. SICS Report, October 1997.

[21] D. Deugo, M. Weiss, and E. Kendall. Reusable Patterns for Agent Coordination. In: Omicini, A., *Coordination of Internet Agents*, pages 347-368. Springer, 2001

[22] L. Rasmusson and S. Janson, "Agents, self-interest and electronic markets", *Knowledge Engineering Review*, vol. 14, no 2, pages 143-150. 1999.

[23] P. Marcenac. Modeling MultiAgent Systems as Self-Organized Critical Systems, 31th Hawaii International Conference on System Sciences, HICSS-31, IEEE Computer Society Press, Daniel R. Dolk eds, vol. 5, pages 86-95.1998.

[24] P. Mathieu, J. Routier and Y. Secq: Dynamic organization of multi-agent systems. *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002*, Bologna, Italy, pages 451-452.

## Acknowledgements

This work has been supported by Newtrade Technologies, 63 de Brésoules, bureau 100. Montréal (Québec) Canada H2Y 1V7