Tatiana V. Solodukha, Oleg A. Sosnovskiy, Boris A. Zhelezko

Belarus State Economic University, Minsk

MULTI-AGENT SYSTEMS FOR E-COMMERCE

Abstract: The article focuses on multi-agent systems and domains that can benefit from multi-agent technology. In the last few years, the agent based modelling community has developed several practical agent based modelling toolkits that enable individuals to develop agent-based applications. The comparison of agent-based modelling toolkits is given. Multi-agent systems are designed to handle changing and dynamic business processes. Any organization with complex and distributed business processes can benefit from multi-agent technology. The model of the e-commerce platform targeted at online trading and auction systems, e-markets and private trading exchanges is presented.

Keywords: agent, multi-agent system, e-commerce, agent-based modelling.

1. Introduction

Agents and multi-agent systems (MAS) are an emergent technology that is expected to have a significant impact on realizing the vision of a global and informational rich services network to support dynamic discovery and interaction of digital enterprises. Moreover, the agent paradigm is expected to give an impetus to all e-business areas including e-commerce, logistics, marketing, manufacturing, recommender systems, tourism, negotiation, etc. Organizations that have successfully implemented agent technologies include DaimlerChrysler, IBM and the Ministry of Defense of the USA. Significant work has already been done for more than a decade since agents have been claimed to be the next breakthrough in software development, resulting in powerful multi-agent platforms and innovative e-business applications. However, it is currently appreciated that for various reasons agent technologies have not achieved yet that level of maturity as required by advanced e-business applications. Many difficult research challenges remain, and much work is needed to adapt relevant existing agent technologies to the requirements of the new generation e-business systems.

The objective of this article is to find out main domains that can benefit from implementation of MAS and to work out the general model of the e-commerce platform targeted at online trading and auction systems, e-markets and private trading exchanges.

2. Multi-agent systems

The modern business environment is dynamic and constantly changing, but customers expect their specific requirements to be met at the agreed service level. The Internet and mobile computing means that the majority of business processes are "always-on" and increasingly interconnected.

Agent-based solutions are designed to handle complex and changing environments, where there are potentially multiple options and outcomes at each stage of the process. Multi-agent systems can offer measurable business benefit for many organizations.

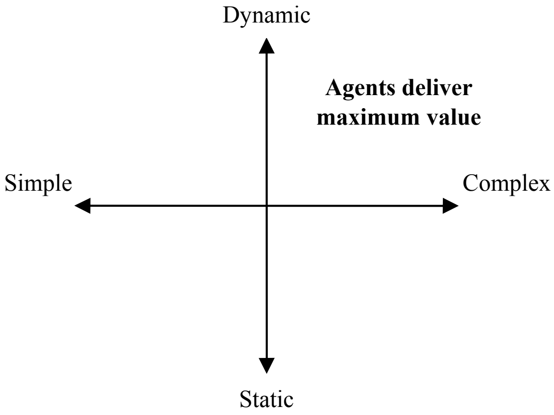

Figure 1. The modern business environment

Source: [4].

The study of multi-agent systems focuses on systems in which many intelligent agents interact with each other. Agents are software programs running on a system or network and implemented to achieve a goal or purpose. They are designed to be autonomous in the pursuit of their goal, but capable of interacting with other agents, either collaboratively or in competition. This means that groups of agents can work together, each carrying out their own plans and functions. This provides a powerful technique for designing software applications for complex or distributed business processes.

Typically multi-agent systems research refers to software agents. However, the agents in a multi-agent system could equally well be robots, humans or human teams. A multi-agent system may contain combined human-agent teams [4].

MAS researchers draw on ideas from many disciplines outside of Artificial Intelligence (AI), including biology, sociology, economics, organization and management science, complex systems, and philosophy.

The dynamic nature of agent distribution motivates research by groups working on the standardization of dynamic collaborative multi-agent systems. Some of these groups are the Foundation for Intelligent Physical Agents (FIPA), the Object Management Group (OMG), the Knowledge-able Agent-oriented System (KAoS), and the General Magic group.

Several researchers have attempted to provide a meaningful classification of the attributes that agents might have. A list of common agent attributes is shown below [1].

Adaptivity: the ability to learn and improve with experience.

Autonomy: the ability to act without any interference from the outside, including human intervention.

Activity: the ability to show its own initiative.

Collaborative behaviour: the ability to work with other agents to achieve a common goal.

Mobility: the ability to migrate in a self-directed way from one host platform to another.

Reactivity: the ability to selectively sense and act.

Temporal continuity: persistence of identity and state over long periods of time.

An agent can be therefore defined as a software entity with goals, actions, and domain knowledge, situated in an environment. The way it acts is called its "behaviour". The agent evaluates the information it receives and formulates the best plan to meet its goals and perform its role within a business process. Collectively, agents in a multi-agent system can operate and interact with a number of other heterogeneous agents to execute an overall business process.

While multi-agent systems are often created from scratch by researchers and developers, some frameworks have arisen that implement common standards (such as the FIPA agent system platforms and communication languages). These frameworks save developers time and also aid in the standardization of MAS development.

In the last few years, the agent-based modelling community has developed several practical agent based modelling toolkits that enable individuals to develop agent-based applications. More and more such toolkits are coming into existence, and each toolkit has a variety of characteristics. Below is Table 1 intended to capture many of the features that are important to agent-based modelling (ABM) toolkit users.

Table 1. Comparison of agent-based modelling toolkits

Platform	Primary domain	License	Programming language required	Operating system required to run	User support
1	2	3	4	5	6
ABLE (Agent Building and Learning Environment)	Building intelligent agents using machine learning and reasoning	Open source (free for academic and non-commercial use)	Able Rule Language (ARL)	OS/2; Windows 95/98/NT; and UNIX (any Java 2 JVM)	FAQ; tutorial; examples; forum; e-mailing developers; API documents
ADK (Tryllian Agent Development Kit) (*FIPA Compliant*)	Large scale distributed applications; Mobile (distributed) agents	Dual licensed: either accept the LGPL or contact Tryllian	Java	Windows; Unix; Big Iron IBM mainframes4; anywhere that the Java Standard Edition version 1.4 runs; Windows 2000/XP; Solaris; GNU/Linux; OS/400 and OS/370	FAQ; defect reporting; documentation; mailing list; quick start guide; examples; API
ECHO	Ecological modeling	Free, open source	C	Unix workstations; Developed on Sun Sparc architecture using Sunos 4.1.3	A few selected publications
AgentBuilder	General purpose multi-agent systems	Proprietary; discounted academic licenses available	Knowledge Query and Manipulation Language (KQML); Java; C; C++	Windows NT/2000/XP; Linux; Sun Solaris; any platform with a Java Virtual Machine	Consulting; training; example; FAQ; users manuals; defect reporting; mailing list
AnyLogic	Agent based general purpose; distributed simulations	Proprietary	Java; UML-RT (UML for real time)	AnyLogic 6 models are standalone Java applications and run on any Java-enabled platform or in any Java-enabled browser; Windows Vista/ XP; Mac OS, Linux	Demos; training; consulting; knowledge base; forum; documentation
JADE (*FIPA Compliant*)	Distributed applications composed of autonomous entities	LGPL version 2	Java	Any Java Platform	FAQ; mailing list; defect list; tutorials; API; documentation
Moduleco	Multi-agent platform	GPL	Java	Windows; Linux; Macintosh	API; minimal documentation
Jade’s sim++	Parallel/applied simulation; network planning; CAD; real time communication simulation	GPL version 2	C++	Available for Meiko and BBN multi-computer systems and can be used on a network with Sun3, Sun4, and HP 9000 workstations	–
SimPlusPlus	Testing Base24 applications	GPL	Fully programmable with any language that can support activeX components	Sim++ can be used with C code or C++ code, but you MUST have a C++ compiler. DOS; Windows (as a DOS application) or OS2 (as a DOS app).	–

1	2	3	4	5	6
StarLogo TNG	Social and natural sciences; teaching basic computer programming skills	The original StarLogo is apparently going to be released under an open source licence soon	StarLogo TNG language – a graphical programming language and a 3d world	Macintosh and Windows	Tutorials; FAQ; documentation; mailing lists; API
NetLogo	Social and natural sciences; help beginning users get started authoring models	Free, not open source; there are some restrictions on redistribution and/or modification	NetLogo	Any Java Virtual Machine, version 1.4.1 or later, is installed; Version 1.5.0_12 or later is preferred	Documentation; FAQ; selected references; tutorials; defect list; mailing lists
Swarm	General purpose agent based	GPL	Java; Objective-C	Windows; Linux; Mac OS X	Wiki; tutorials; examples; documentation; FAQ; selected publications; mailing lists
ZEUS (*FIPA Compliant*)	Rules engine and scripting environment; Distributed multi-agent simulations	Open source (read license)	Visual editors and code generators	Windows 95/98/NT/2000/ XP; Linux; BSD; UNIX-like OSes; Solaris	Documentation; author contact

Source: [3].

3. E-commerce platform

As a result of given research the model of the e-commerce platform targeted at online trading and auction systems, e-markets and private trading exchanges was developed.

Software agents lie at the heart of the platform, automating the business processes that match buyer and sellers. The platform may support e-commerce activity across a wide range of markets and industries. The platform consists of a series of modules supporting different business processes, such as auctions, tenders and negotiations. The modular approach allows the flexibility to implement the most appropriate solution and add additional modules as required. The platform can be delivered as a buy-side, sell-side, broker or marketplace solution.

E-commerce platform modules are core module (market owner's module); software agents (e-commerce platform agents); auction module; sourcing module; negotiation module (Figure 2).

Market owner's module. This core module allows the platform to be configured and branded to meet the basic business requirements and processes. This includes setting business rules and terminology, user groups and access permissions, language, currencies and preferred communications media – e-mail, SMS, WAP, Fax or telex.

Standard catalogue functionality is also provided for setting up and managing product hierarchies.

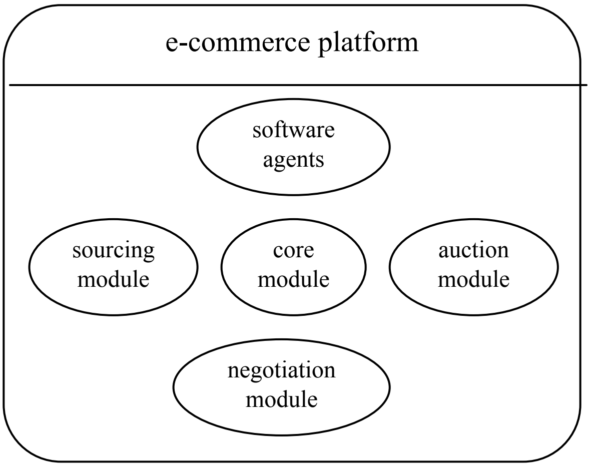

Figure 2. The model of the e-commerce platform

Source: own work.

E-commerce platform agents. These agents work for buyers, sellers, brokers and market owners in order to achieve business goals. They can save time, money and effort by automating simple tasks such as searching, monitoring and sorting. Agents can also assist in more complex tasks including analysis, decision-making and negotiation.

Auction module. The auction module provides an effective way to sell a wide range of products, particularly time sensitive or perishable goods. The module offers a comprehensive range of auction formats, including: English, sealed bid, Japanese, Dutch, Vickery. Additional auction types can be easily added to meet specific requirements. Once bidding and win parameters are defined, software agents can be instructed to set up, or take part in an auction-making, accepting or rejecting bids, monitoring progress and alerting their owners of developments.

Sourcing module. The sourcing module makes it easy to buy simple or tailor-made products. The module allows buyers to define their exact requirements, submit requests for proposals (RFP or tenders) and manage incoming responses. If a short list is produced, sellers can be routed to an auction or a negotiation. Sellers can instruct agents to search and monitor for RFPs, requesting more information, if necessary, or responding appropriately with an offer.

Negotiation module. The negotiation module offers a sophisticated means of automating part, or all, of the buying and selling process. It can handle complex negotiations, incorporating multiple factors such as price, delivery and payment terms. The module cans manage discussions with multiple parties simultaneously to optimize the speed and effectiveness of any negotiation. Once a negotiation strategy is defined, software agents can submit requests, counter offers and complete deals on their owners' behalf, reporting back on progress.

In order to build an appropriate e-commerce platform the agent business analysis should be done (Figure 3).

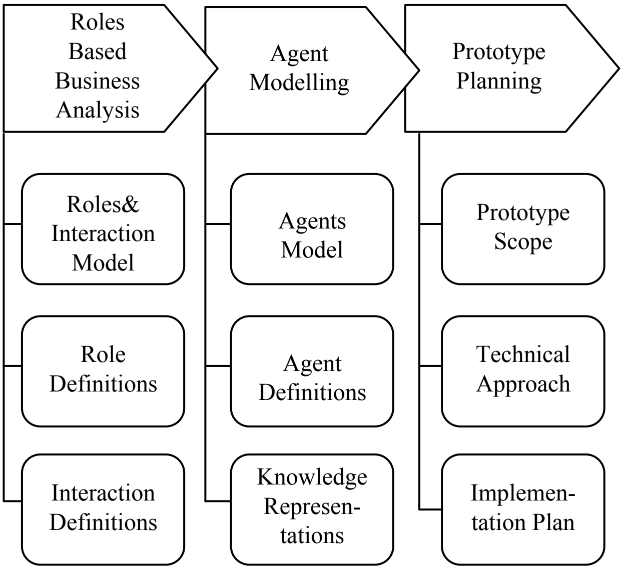

Figure 3. Agent business analysis

Source: [3].

Agent business analysis process enables the roles and interactions that are involved in providing the client's service to be understood. This understanding enables to identify and define the specific value added functions that agents can play in delivering the service. From this information a prototype solution that assists the client in determining how and where best to deploy agent based solutions is created.

Stage 1: Roles-based Business Analysis. This stage maps out the roles in the client's business process of the agreed domain. The starting point is a consideration of the human roles and the information resources that each role uses. The interactions between these roles are identified and added to form a roles and interaction model.

Stage 2: Agent Modelling. This stage identifies the agents required to perform the roles in the Roles and Interaction Model. A role can be implemented in a single agent, or an agent may embrace and perform multiple roles. It is important to establish the business benefit that comes from deciding to use an agent to perform a specific role. Some roles may be better implemented using standard non-agent solutions.

The resulting Agent Model illustrates the boundaries of the agent system, the population of agents required to realize the business goals and the communication links between the agents.

For agents to understand and respond to their environment they must share a common language describing the business terms and concepts. Another key element of the Agent Definition process is therefore a consideration of the Knowledge Representations they will use.

Stage 3: Prototype Planning. On this stage the most appropriate functionality for the prototype is defined. It is essential that the functionality included in a prototype is aligned with the client's business objectives rather than driven by technical capabilities. A key input to defining the Prototype Scope is therefore the evaluation criteria that will be applied to the prototype.

Today's business environment is highly connected, global and extremely competitive. There is increasing demand for greater responsiveness to market challenges and opportunities. In turn this demands greater flexibility in business processes. We are certain that multi-agent software is the correct approach to creating responsive and adaptable business solutions to meet these demands.

Multi-agent systems are designed to handle changing and dynamic business processes. They continue to operate even when there may be incomplete information, but a decision must still be made by the most effective method. Properly integrated into a business process they can offer flexible and intelligently adaptable systems support, which will consistently outperform monolithic and cumbersome traditional systems.

Any organization with complex and distributed business processes can benefit from multi-agent technology.

4. Reasons to use multi-agent systems

It would be incorrect to claim that MAS should be used when designing all complex systems. Like any useful approach, there are some situations for which it is particularly appropriate, and others for which it is not. The goal of this section is to underscore the need for and usefulness of MAS while giving characteristics of typical domains that can benefit from it. For more information, see [2; 3].

The most important reason to use MAS when designing a system is that some domains require it. In particular, if there are different people or organizations with different (possibly conflicting) goals and proprietary information, then a multi-agent system is needed to handle their interactions. Even if each organization wants to model its internal affairs with a single system, the organizations will not give authority to any single person to build a system that represents them all: the different organizations will need their own systems that reflect their capabilities and priorities. The only feasible solution is to allow the various companies to create their own agents that accurately represent their goals and interests. They must then be combined into a multi-agent system with the aid of some of the techniques described in this article.

Even in domains that could conceivably use systems that are not distributed, there are several possible reasons to use MAS. Having multiple agents could speed up a system's operation by providing a method for parallel computation. For instance, a domain that is easily broken into components – several independent tasks that can be handled by separate agents – could benefit from MAS. Furthermore, the parallelism of MAS can help deal with time limitations.

While parallelism is achieved by assigning different tasks or abilities to different agents, robustness is a benefit of multi-agent systems that have redundant agents. If control and responsibilities are sufficiently shared among different agents, the system can tolerate failures by one or more of the agents. Domains that are in particular need of this feature of MAS: if a single entity – processor or agent – controls everything, then the entire system could crash if there is a single failure.

Another benefit of multi-agent systems is their scalability. Since they are inherently modular, it should be easier to add new agents to a multi-agent system than it is to add new capabilities to a monolithic system. Systems whose capabilities and parameters are likely to need to change over time or across agents can also benefit from this advantage of MAS.

Table 2. Reasons to use multi-agent systems

Some domains require it	Scalability
Parallelism	Simpler programming
Robustness	To study intelligence

Source: [3].

From a programmer's perspective the modularity of multi-agent systems can lead to simpler programming. Of course there are some domains with centralized control where no parallel actions are possible and there is no action uncertainty. Single-agent systems should be used in such cases.

Finally, multi-agent systems can be useful for their elucidation of intelligence.

Reasons presented above to use MAS are summarized in Table 2.

5. Conclusion

Multi-agent systems are designed to handle changing and dynamic business processes. Today numerous agent-based modelling toolkits have been developed. They enable individuals to develop agent-based applications. More and more such toolkits are coming into existence, and each toolkit has a variety of characteristics. Each organization can choose the ABM toolkit that will match its goals, business processes and opportunities.

Any organization with complex and distributed business processes can benefit from multi-agent technology.

As a result of given research the model of the e-commerce platform was developed. It is targeted at online trading and auction systems, e-markets and private trading exchanges. The platform may support e-commerce activity across a wide range of markets and industries.

References

[1] Bradshaw J.M., *An introduction to software agents*, [in:] J.M. Bradshaw (ed.), *Software Agents*, AAAI Press, Menlo Park 1997, pp. 3-46.

[2] Morozevich A.N., Zhelezko B.A., Basova E.S., *Personal Multi-Agent Systems of Decision Making Support*, sedok.narod.ru/s_files/ belorussia_2002.htm.

[3] Stone P., Veloso M., *Multi-agent Systems: A Survey from a Machine Learning Perspective*, Carnegie Mellon University, Pittsburgh 2007.

[4] Sycara K.P., *Multiagent systems*, “AI Magazine” 2002, Vol. 2, pp. 79-92.

SYSTEMY WIELOAGENTOWE W HANDLU ELEKTRONICZNYM

Streszczenie: Autorzy skupili się na systemach wieloagentowych oraz na korzyściach, jakie płyną z ich zastosowania. W ciągu ostatnich kilku lat społeczność modelowania wieloagentowego stworzyła wiele praktycznych narzędzi modelowania, które pozwalają na samodzielne tworzenie aplikacji wieloagentowych. Zostały pokazane różnice pomiędzy poszczególnymi pakietami narzędziowymi. Systemy wieloagentowe zostały zaprojektowane tak, aby radzić sobie ze zmieniającym się dynamicznie procesem biznesowym. Każda organizacja ze skomplikowanymi i rozproszonymi procesami biznesowymi może czerpać korzyści z technologii wieloagentowej. Został zaprezentowany model platformy handlu elektronicznego nastawionego na handel, aukcje, sklepy internetowe i prywatną wymianę towarową.