

Modelling a multi agent system for dairy farms for integrated decision making

A. Thangaraj^a, A. Patricia^b and S. Samarasinghe^c

^a *PhD Student, Department of Informatics and Enabling Technologies, Lincoln University, Christchurch, New Zealand*

^b *Senior Lecturer, Department of Informatics and Enabling Technologies, Lincoln University, Christchurch, New Zealand*

^c *Professor, Department of Informatics and Enabling Technologies, Lincoln University, Christchurch, New Zealand*

Email: Anitha.Thangaraj@lincolnuni.ac.nz

Abstract: Currently, dairy farmers are using various software tools to enable them to make decisions on different aspects of farming such as pasture management, herd management, nutrient management and finance management. Some of the widely used software tools for dairy farming are as follows: Rapid Plate Meter, Overseer, Farm Enviro Walk, and Cash Manager. Each of these tools is used only for particular aspects of dairy farm management. For example, Overseer is used to check the nutrient levels present in the soil. Since these tools operate independently, farmers often find it difficult and time consuming when making integrated decisions relating to farm management. Often, they need to extract data from each of these tools, combine and integrate them and use another general purpose tool to perform analysis before making any decision. The complexity of the decision making is increased with the amount of information and, to make matters worse, information loss may occur during the integration.

The main objective of the proposed research is to model a multi agent system platform to automatically integrate data from the various software tools and perform a variety of analysis to support decision making on the farm. The proposed multi-agent system environment provides an interface in which farmers can enter queries related to farming which requires agents to communicate with the various tools in order to respond to these queries. This is possible as the agents possess special abilities such as autonomy, dynamism and intelligence. Each management activity has its own functions and outputs.

- Pasture management - deals with pasture related works.
- Nutrient management - performs tasks involving nutrient application.
- Herd management - perform tasks related to cattle health, herd cleaning and other herd related activities.
- Financial management - contains tasks like calculating revenue, stock maintenance and other related financial activities.

An individual agent is designed for each management aspect of dairy farming and co-ordination between them is managed by a coordinator agent to form a multi-agent system. Depending on the problem to be solved, different agents are invoked. Agents can communicate with other agents whenever necessary when making decisions in a multi agent system. The model consists of the following agents: interface agent, control agent, pasture management agent, herd management agent, nutrient management agent and finance management agent. The interface agent is the bridge between user and the multi agent system. The interface agent senses the environment and passes the user information to the control agent. The control agent is the controller who decides what action plans need to be executed based on the information received from interface agent. The control agent will trigger the appropriate actions based on the changes in the environment. All other agents deal with tasks related to the appropriate dairy farm management. For instance, the pasture management agent handles queries related to dairy pasture management. The main advantage of this integrated model is that farmers will be able to make farming decisions more efficiently and effectively without the hassle of having to work through the data from various data sources.

Keywords: *Software agents, Multi Agent System (MAS), integrated decision making, dairy farms*

1. INTRODUCTION

The number of on-farm computers in dairy farms is on the rise as software tools enable farmers to manage their daily farming data (Pomar *et al.*, 2003). There is a huge volume of data collected on dairy farms such as pasture growth information, herd health information, milking data, nutrient levels in soil and farm payment information. The volume of data is increasing as it is collected on a regular basis. There are various software tools that are used to analyze these data and produce useful information for farmers such as Overseer, Rapid Plate Meter and Cash Manager (Allen and Wolfert, 2011). To be able to manage the farm effectively, farmers would need to use different software for different purposes. For example, pasture management software is used to control and monitor the growth of grass by reducing weeds, applying proper nutrients at the correct time and in the correct proportions, managing movement of cattle in paddocks and applying water on time (DairyNZ, 2015). Herd management deals with milking, health management of cows, manure handling, herd cleaning, calf management and breeding (DairyNZ, 2013). Nutrient management tools are used to reduce the application and leaching of fertilizers like nitrogen and phosphorous into soil and waterways (Tamminga, 1992). Each of these systems provides some advantages to the dairy industry. Financial management tools play an important role in dairy farming as without good financial management, it is difficult to achieve goals and maintain profitability in any industry, including dairy (Kadam, 2012).

Each of these software tools and systems has been created for particular aspects of the dairy industry. For example, when a farmer needs to make a decision about application of nutrients, he/she has to check both pasture management and nutrient management tools (Overseer and Rapid Plate Meter). Since, nutrients should be applied in the appropriate growth periods, it is necessary to apply the correct amount based on the amount of nutrients present in soil. Dairy farming is a dynamic and integrated production system as it requires continual and effective decision making (Cabrera, 2012). Most dairy farming systems such as pasture management, financial management, herd management and nutrient management are interrelated (Rotz, *et al.*, 2011). Therefore, a multi-agent framework that integrates various farming systems may be useful for successful dairy farming management and decision making (Giordano *et al.*, 2011), (Cabrera *et al.*, 2006), (Groenendaal *et al.*, 2004), (Meadows *et al.*, 2005).

A software agent is defined as follows: "An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its delegated objectives", (Wooldridge and Jennings, 1995), (Wooldridge, 2009, p. 21). The main characteristics of software agents are:

- Autonomy – agents are capable of doing their assigned task without human intervention.
- Reactivity – agents recognize the changes in the environment and act accordingly.
- Pro-activeness – agents assure that their assigned task is satisfied before reacting.
- Social ability – agents communicate with other agents in the current environment through message passing.

These characteristics enable software agents to make decisions and solve the given task more effectively when compared to traditional decision making systems (Yongyong, 2012). When agents work together as a team to achieve a certain goal it is referred to as a multi-agent system (Grosz and Kraus, 1996).

Multi agent systems (MAS) consist of number of autonomous entities that interact and coordinate their knowledge to successfully solve the given problem (Weiss, 2013). The various agents present in the environment communicate through a high level language called agent communication language (ACL). In a multi agent system, each agent should exhibit the following characteristics (Goel *et al.*, 2005):

- The goals and objectives should be clearly defined.
- Have knowledge about other agents in the environment.
- Communicate with other agents when required.

As discussed earlier, dairy farming is dynamic and complicated in that it contains multiple systems, thus requires efficient decision making systems that must be able to respond to farmers' queries, perform analysis on data automatically, generate essential reports upon requests without farmers having to access the different software tools. In addition to that, it should be able to sense the incoming data to the dairy farm management system, provide an up-to-date analysis and proactively alert farmers if unusual events occur. Multi agent systems are useful as well as efficient when the system is complicated, dynamic and involves various components (Wooldridge, 2009), (Lavendelis *et al.*, 2012), (Adams, 2001). Each component has its own input and output. Hence, interaction between them is necessary to solve the problem of the whole system. In the same way, dairy farming is a dynamic system which involves different systems such as pasture management, financial management, herd management and nutrient management. Each system has its unique input and output, and they have to communicate with others whenever necessary to support decision making on the dairy

farm. Hence, the main objective of the proposed work is to model an integrated decision support system that can automatically combine data from various software tools when required.

Each agent in the proposed multi agent model is a specialist agent that can provide additional functionalities to the existing software tools. These agents continuously sense the environment, identify any changes and perform actions based on that. For instance, pasture agent continuously monitors the feed requirement and if the feed is in deficit, it sends alert to the farmer about the feed shortage so that farmers can take immediate actions to maintain production. The specialist agent can also send the alert with additional information about the cause of the alert and required actions to address the issue. For example, if the milk production is showing a decreasing trend, the herd agent will attempt to identify the reasons for the dip and will send an alert to the farmer along with reasons and possible corrective actions. In addition, all specialist agents can work together to support decision making in the farm. For example, the proposed system will be able to answer the following user's query "Can I buy 100 cows next month?" In this case, the pasture agent will need to work together with the herd agent and the finance agent to retrieve information about the land availability, feedstock and existing funds and decide whether the farmer can purchase 100 cows.

The remainder of the paper is organized as follows: Section 2 discusses some related work on the application of agents and multi agents in agriculture. Section 3 describes the proposed multi-agent architecture. Section 4 describes the implementation of the proposed work. Finally, the paper discusses future works in Section 5.

2. LITERATURE REVIEW

In the last decade, the use of multi agent systems and their applications has become increasingly popular because of their beneficial characteristics. To date, agent based technology has been applied in areas such as e-commerce, health care, military, industrial systems, micro-finance and agriculture. Software agents have also been used in dairy farming to develop integrated decision support systems.

Parrott *et al.* (2003) investigated the feasibility of using a multi agent system in agriculture particularly in dairy farming and developed a simple multi agent system for heifer management. Their system consists of two main agents called Heifer App agent and genetic data agent. The main purpose of Heifer App agent is to get the input from the user and access the local heifer database to service the request. The Heifer app agent contacts the genetic data agent through message passing whenever information is required. The message passing process uses an agent communication language which consists of four parameters (sender IP address, sender port number, performative (ask, reply, sorry, error) and content of the message). The system produces reports on herd details based on a user's request. However, this work focuses only on herd management in dairy farming. If the decision support system that involves all other management systems (pasture, nutrient and finance) is developed, then it will reduce the manual decision making time and will also assist farmers to enhance the management process of the whole dairy farm.

A livestock precision feeding system was designed and developed by Pomar *et al.* (2003) using a multi agent system to automate the feeding process. The system consists of an agent communication interface, agent user interface and a control unit. These agents collaborate with each other to function as an automatic feeding system. Their architecture has four main agents; feeder agent, feed mixer agent, formulator agent and feed to feeder transport agent. The main functions of the feeder agent are to identify the animal and its weight at arrival time into the feeder and control the amount of feed. The feeder agent makes decisions about the composition of feed by measuring the weight of the animal. The feed mixer agent produces the composition of the feed specific to each animal based on the request received from feeder agent and then sends the request to the formulator agent. The formulator agent controls the formulation of feed. The feed is then transferred to a feeder via the feeder transport agent which is responsible for sending the request to the hopper to fill the feed. The performance was evaluated against that of a manual feeder and the results showed that the simulated feeder saved time and money.

Goel *et al.* (2005) developed a multi-agent system which consisted of three agents, namely the facilitator agent, the miller agent and the producer agent to implement electronic contracting of food grains. They integrated various actors such as millers and producers in a food supply chain for better negotiations to reach a mutually acceptable price. The miller agent sends its requirements to the facilitator agent. The facilitator agent informs all the producer agents registered in the system. The producer agents that are interested send their bids to the facilitator agent. The facilitator agent evaluates all bids for the auctions and decides on the winning bid. Finally, the result is sent to the producer and the miller agent. The performance of this system was evaluated using bid clustering and the degree of competition among the agents.

3. PROPOSED ARCHITECTURE

The overall architecture of the proposed multi-agent system is shown in Figure 1. It consists of six agents which are the interface agent, control agent, nutrient management agent, herd management agent, Pasture management agent and financial management agent. It is assumed that the nutrient management agent, herd management agent, pasture management agent and financial management agent have direct access to the existing individual tools and they are considered the specialist agents in their respective areas. In this architecture, only four specialists are used. However, in a real world setting this can be extended to cater for other types of agents.

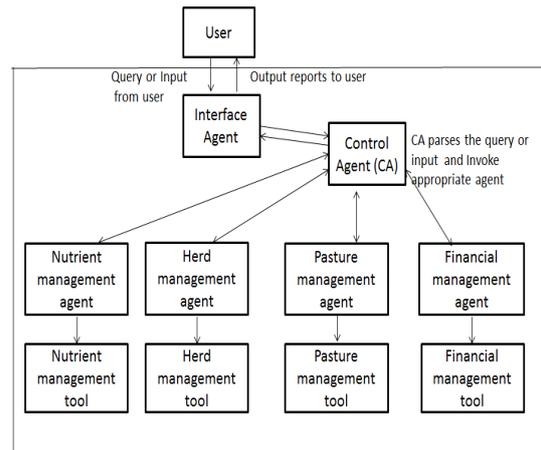


Figure 1. Proposed multi agent architecture

3.1 Interface Agent

The main purpose of the interface agent is to provide connectivity between the user and the multi-agent system. The input for the system is a query with a predefined format. The interface agent parses this query before passing it to the control agent for further processing.

3.2 Control Agent

The control agent is responsible for processing the parsed query sent by the interface agent. The control agent decides what information is required for the user's query and identifies which agent needs to be communicated with next for the intended query. One or more agents may be contacted based on the query. This agent sends ACL messages to the necessary agents and waits for the reply from those agents. Finally, the control agent is responsible for integrating all the information and processing it further (if required) before passing this result back to the interface agent to be presented to the user. The control agent has access to a plan library and a knowledge base in determining what actions need to be performed based on the query submitted by the user.

3.3 Pasture Management Agent

The main role of the pasture management agent is to manage all pasture related activities for dairy farming. Some key jobs include irrigating pasture, calculating pasture moisture and identifying pasture growth rate. The pasture management agent is invoked by the control agent based on the incoming user input. The pasture management agent will then interact with the pasture management tool and retrieve the required data from that tool. The results are passed to the control agent for further processing.

3.4 Nutrient Management Agent

The nutrient management agent is responsible for making sure that the right amount of nutrients is applied in the dairy fields. The main nutrients applied to pasture are nitrogen and phosphorous. Nutrients are also emitted as cattle urine. When the applications of nutrients are high, negative impacts in the environment will be large, hence nutrients should be applied at the correct time to reduce the excess leakage. This nutrient management agent interacts directly with the existing nutrient management tools to facilitate data processing and decision making related to nutrient management.

3.5 Financial Management Agent

The financial management agent deals with all activities related to finance. The finance management agent is responsible for calculating expected income, total gain, managing stock numbers and employee salaries. The financial management agent interacts with the financial management tool to extract the relevant data. This output will then be passed on to the control agent for further processing.

3.6 Herd Management Agent

The herd management agent interacts with the herd management tool to deal with herd related activities such as animal health, milk collection, breeding, calf management and herd maintenance. The herd management behaves in a similar way to the other three agents (pasture, nutrient and financial agents) to provide specialised services related to herd management.

4. IMPLEMENTATION AND DISCUSSION

The proposed architecture is implemented using the Java Agent Development framework (JADE). JADE is a middleware that complies with FIPA's (Foundation for Intelligent Physical Agents) specification that enables a programmer to design multi agent systems efficiently (Bellifemine et al., 1999). This framework is fully constructed using the Java language for the development of interactive software agents in a distributed manner. JADE includes a run time environment, library and graphical tools.

The running instance in JADE is called a Container. Agents sit inside the Container and perform their actions. Each agent should have its own identity such as a name and identification number. These attributes are assigned by the agent management system. JADE requires only the Java Runtime Environment (JRE) to program and execute agents.

Results are fetched in a timely manner from the databases with the necessary actions completed instantly. The agents may be local or remote depending on the design of a particular system. Directory facilitator is used by the agents to identify other agents that are active in the JADE platform. In JADE, communication between agents is done via ACL (Agent Communication Language) messages. The agent identifier is the main component of ACL messages. It also has many features other than identifier depending on the required action and environment.

The development of the current multi-agent system is still at its preliminary stage. The main objective of this preliminary version is to ensure that agents can communicate and process the user queries appropriately. The user queries are kept simple at this stage. The user submits a query via the user interface which is handled by the user interface agent (GUIAgent). The interface agent parses the query and prepares the ACL message to be forwarded to the control agent. The control agent interprets the ACL message upon receipt and decides what action needs to be executed. It also decides which agents need to be contacted and prepares the content of the ACL messages. The ACL messages are then sent to the appropriate agents and the control agent waits for the other agents' responses. Upon receiving the information from the required agents, the control agent processes the data and forwards the output to the user interface agent to be presented to the user via the user interface. Currently, the ACL performatives 'INFORM' and 'REQUEST' are used to facilitate message passing between agents. The content of the message is embedded into ACL and is sent to the appropriate agents using agent identifier.

Figure 2 shows the agent user interface where a user can select from predefined queries. Once the user selects the pasture checkbox, for example, the predefined queries related to pasture management are displayed in the possible queries combo box. The interface agent reads the query selected when the user hits the OK button. Similarly, the user can select other options when required. The General option checkbox is used to facilitate queries that involve multiple specialist agents. The first example query, Query 1 - "What is the pasture growth?" is shown in Figure 3 and Figure 4. For this query, the interface agent reads and parses the query. The interface agent identifies the keyword(s) in the user input, it prepares the ACL message and attaches the keywords with that ACL message. The ACL message contains the agent identifier, actual message and performative type. In this query, the performative type used is "REQUEST".



Figure 2. Agent user interface

The control agent receives the ACL message from the interface agent where it then decides that the pasture management agent is the right one to contact. Next, the control agent prepares the ACL message, sets the agent identifier of the pasture management and sends the query to retrieve the information from the pasture management agent. Figure 3 shows the interaction threads of the different agents for Query 1 and Figure 4 shows the output for Query 1 which is pasture growth information including growth rate (GRate kg DM/ha/day, Temperature °C and Humidity (%)).

The next example query is "What is the nutrient levels in the soil? (Query 2)". With this user's query, the interface agent interprets the input, prepares an ACL message and sends it to the control agent. The control agent then decides that this query is relevant to the nutrient management agent, prepares an ACL message and

passes the message to the nutrient management agent. The nutrient management agent replies to the control agent with the appropriate information. The information received from the nutrient management agent is then sent to the user interface via the interface agent. The agent interaction for Query 2 is depicted in Figure 5.

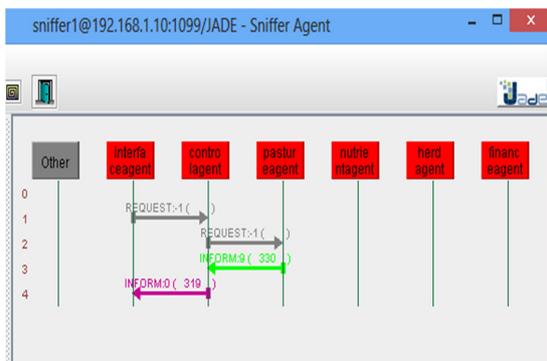


Figure 3. Interaction thread for Query 1

Plot ID	GRate date	GRate	Temp	Humidity
1	2016-01-01	59	20	50
1	2016-02-01	58	20	50
1	2016-03-01	32	18	80
1	2016-04-01	18	20	94
2	2016-01-01	45	20	50
2	2016-02-01	23	20	50
2	2016-03-01	34	20	80
2	2016-04-01	16	18	94
2	2016-05-01	35	19	87

Figure 4. Sample output (Query 1)

To process queries that require feedback from multiple agents, the control agent prepares the required number of ACL messages and sends these messages to the appropriate agents. Then it waits for the replies from the agents and combines the information to display the required output to the user. For instance, let's say the query is "What is the water requirement for the dairy farm this month? (Query 3)". In this query, the control agent decides that it needs to communicate with the pasture management agent and the herd management agent since water is required for both herd and pasture. The control agent receives the query from the interface agent, then prepares two separate ACL messages and sends them to the pasture and nutrient management agents. It combines the information received from both agents and passes back the final information to interface agent which is then displayed. The ACL messages which are generated for Query 3 is depicted in Figure 6.

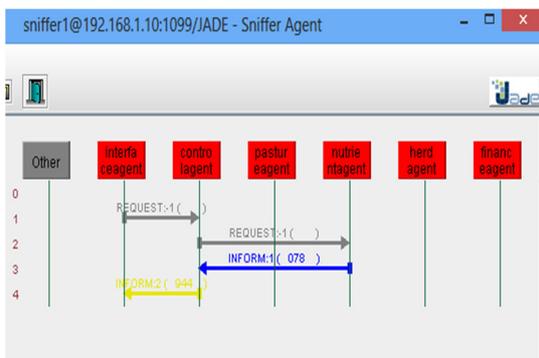


Figure 5. Interaction thread for Query 2

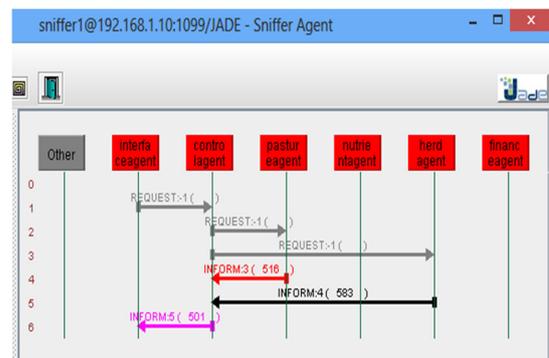


Figure 6. Agent interaction for Query 3

This preliminary model is working correctly as intended and responds appropriately to user requests. The queries used at this stage are quite simple, but are sufficient to show that the agents are working together in delivering the desired output based on the query.

5. FUTURE WORK

In this paper, we described the modeling and implementation of a multi agent system that can be used to process and manage dairy farming data effectively. The proposed work will reduce the time spent by the dairy farmers in making routine decisions. This preliminary work showed that the multi agents can be used to integrate different management systems in dairy farming for making decisions. Future work will involve extending the model further to accept a query in a free format, developing parser for the query, developing ontology, refining the decision making engine and evaluating the performance of the MAS under more complex decision making conditions.

REFERENCES

- Adams, J. A. (2001). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence :A Review*. Artificial Intelligence, 22.
- Allen, J., and Wolfert, S. (2011). Farming for the Future: towards better information-based decision-making and communication. In: *International Farm Management Conference* in Methven.
- Badjonski, M., and Ivanovic, M. (2000). A multi-agent system for the determination of optimal hybrids in crop production. In: *Computers and Electronics in Agriculture* 25 233–243, Elsevier B.V.
- Bellifemine, F., Poggi, A., and Rimassa, G. (1999). JADE–A FIPA-compliant agent framework. In: *Proceedings of PAAM* Vol. 99, No. 97-108, p. 33.
- Cabrera, V. E. (2012). DairyMGT A Suite of Decision Support Systems in Dairy Farm Management. In: *Decision Support Systems*, ISBN 978-953-51-0799-6, 282 pages, Publisher: InTech, Chapters published October 17 2012.
- Cabrera, V. E., Hildebrand, P. E., Jones, J. W., Letson, D., and de Vries, A. (2006). An Integrated North Florida Dairy Farm Model to Reduce Environmental Impacts under Seasonal Climate Variability(113), 82-97.
- DairyNZ.- Herd management (2013). Retrieved from <http://www.dairynz.co.nz/animal/herd-management/>.
- DairyNZ - Nutrient management on your dairy farm (2013). Retrieved from http://www.dairynz.co.nz/media/757901/nutrient_management_on_your_dairy_farm.pdf.
- DairyNZ. Pasture management (2015). Retrieved from <http://www.dairynz.co.nz/feed/pasture-management/grazing-management>.
- Goel, A., Zobel, C. W., and Jones, E. C. (2005). A multi-agent system for supporting the electronic contracting of food grains. In: *Computers and Electronics in Agriculture* 48 123–137, Elsevier B.V.
- Groenendaal, H., Galligan, D. T., and Mulder, H. A. (2004). An Economic Spreadsheet Model to Determine Optimal Breeding and Replacement Decisions for Dairy Cattle. *Journal of Dairy Science*, 87, 2146-2157.
- Kadam, R. N. (2012). Financial management for the organizational success: Challenge before finance managers. *International Journal of Advanced Research in Management and Social Sciences*, 1(1).
- Meadows, C., Rajala, S. P. J., and Frazer, G. S. (2005). A Spreadsheet-Based Model Demonstrating the Nonuniform Economic Effects of Varying Reproductive Performance in Ohio Dairy Herds. *Journal of Dairy Science*, 88, 1244-1254.
- Nute, D., Potter, W.D., Cheng, Z., Dass, M., Glende, A., Maierv, F., Routh, C., Uchiyama, H., Wang, J., Witzig, S., Twery, M., Knopp, P., Thomasma, S., and Rauscher, H. M. (2005). A method for integrating multiple components in a decision support system. In: *Computers and Electronics in Agriculture* 49 44–59, Elsevier B.V.
- Parrott, L., Lacroix, R., and Wade, K. M. (2003). Design considerations for the implementation of multi-agent systems in the dairy industry. In: *Computers and Electronics in Agriculture* 38 79–98, Elsevier B.V.
- Pomar, J., Lopez, V., and Pomar, C. (2011). Agent-based simulation framework for virtual prototyping of advanced livestock precision feeding systems. In: *Computers and Electronics in Agriculture* 78 88–97, Elsevier B.V.
- Rotz, C., Chianese, M. F., Hafner, S. D., and Coiner, C. U. (2011). The Integrated Farm System Model. Retrieved from <http://www.ars.usda.gov/SP2UserFiles/Place/19020000/ifsmreference.pdf>.
- Tamminga, S. (1992). Nutrition Management of Dairy Cows as a Contribution to Pollution Control. *Journal of Dairy Science*, 75(1), 345-357. doi:[http://dx.doi.org/10.3168/jds.S0022-0302\(92\)77770-4](http://dx.doi.org/10.3168/jds.S0022-0302(92)77770-4).
- Weiss, G. (2013). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*: MIT Press.
- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. UK: A John Wiley and sons.
- Wooldridge, M., & Jennings, N. R. (1995). *Intelligent Agents:theory and practices*. The Knowledge Engineering review, 10(2), 115-152.