Multi Agents System for Enterprise Resource Planning selection process using distributed computing architecture

Augusto Alberto Pacheco-Comer, Juan Carlos González-Castolo
Universidad de Guadalajara, Centro Universitario de Ciencias Económico Administrativas, Doctorado en Tecnologías de Información
apacheco@cucea.udg.mx, jcgcastolo@cucea.udg.mx

Abstract
This paper shows a review research literature regarding multi-agents systems related with enterprise resource planning systems and high performance computing. A multi agent system model proposal for enterprise resource planning selection process using distributed computing architecture is presented. The enterprise resource planning selection process is divided in six steps. The proposal model is an aid for evaluation process and has five different types of agents. A distributed computing architecture is proposed as a mean to improve the performance in the use of proposal model.

Keywords: Enterprise Resource Planning, Multi agent system

1. Introduction
The process of selecting an Enterprise Resource Planning (ERP) system is a complex problem which involves multiple actors and variables, since it is a decision-making process which is characterized as unstructured type [1, 2]. Multi Agent Systems (MAS) are designed to assist in the process of solving complex problems [3], so its use can be recommended for modeling and simulation of the selection process of ERP systems.

In this paper we present literature related to MAS and ERP systems. How this two information system types could be related. Following with a propose MAS architecture for ERP selection process using distributed computing architecture to leverage high computer power (HPC) to process a simulation prototype MAS proposed. Finishing with a couple suggestions for future research topics regarding HPC and MAS aligned to ERP implementation.

The paper is organized beginning in section two with a research literature review regarding MAS, Distributed Computing System and their relation with ERP systems. Section three presents a MAS proposed architecture for ERP selection process. Section four presents discussion and conclusion regarding the proposed model; how the model could be include it on Distributed System research area, ideas for future research work and next step for the model.

2. Research literature
Agent-based computing is one of the tools used by computers intelligences to find solutions to complex problems. [3]

MAS seek to address the trends in computer science such as ubiquity, interconnectedness and intelligence [4]. An agent is a computer system capable of undertaking activities independently to benefit its owner or user [4]. A multi agent system consists of a set of agents that interact within an environment, which act on behalf of the objectives and motivations of their owners. Agents have the ability to cooperate, coordinate and negotiate looking at all times comply with the purposes for which they were created [4].

Intelligent agents have the ability to make decisions, those decisions are defined by their inherent properties, which are: reactivity, pro-activity and sociability. These properties are intended to enable the agent to meet the objectives for which they were designed, following rules of behavior that enable them to communicate with their environment [4].

In the development of multi-agent systems, an important point is how the agents represent users, both in physical appearance, as in the activities. In a study
by King [5], was identified in a preliminary way for people to better identify an agent when it has human physical characteristics because these characteristics are perceived as a sign of greater intelligence. And if the face of representation allows eyes blink, the perception of intelligence is higher [5].

Performance requirements embedded in agents must be clearly defined, as mentioned by Feber [6]. He indicates that the requirements of organizations can be represented by a set of agents. He identified four types of behavioral requirements, which are: Requirement of behavior for the individual role, for the interaction within groups, for successful communication within groups and for interaction between groups [6].

Moreover Zhang [3] mentioned that the resolution of complex problems include how the subsystems of the problem can be managed, at times, autonomy, intelligence, relevance and control of their own state and behavior. In his work, He used, as an example of the use of multi-agent system, hierarchical distribution of the coordination of industrial process control, where the control system is divided into four agents: Agent for scheduling system administration, control agent for work in process for each of the workstations, the agent for field monitoring and the environment agent. The management scheduling system agent has communication with ERP system end user (Figure 1) database to receive work orders [3].

Another example of multi-agent system related to ERP is presented by Kehagias [7]. He mentioned the connection with the ERP database designing a new functional layer. This layer corresponds to a MAS architecture which control sales orders and give a recommendation on how could be fulfilled. This system consists of five agents: the client orders agent, the recommender agent, the customer profile identifier agent, the inventory profile identifier agent and the supplier profile identifier agent [7]. The goal of the system is to obtain a recommendation of how the customer orders should be fulfilled, but also, each of the agents should recommend additional actions to perform, such as: the generation of purchase orders according to statistics inventories and time range, addiction to customer orders based on their purchase history and recommend suppliers according to products stock, statistical spend, lead times and historic operations with that suppliers. One aspect that seems important to stress is the proposed layer structure. If we take the network layers of the OSI model and we compare that model with this layer model, we could indicate which agents are integrated in which layers. This layered structure, as the author mentioned, allows the possibility to superimpose MAS layer over different types of systems, according to the characterization of systems proposed by Laudon [8].

Reviewing some of the case studies using agents, we can identify that the development and operational framework is not clearly defined, in this respect, the
work of Flores-Mendez [9] allows us to learn more about this research area because he introduces us to the different terminologies used in this topic and provides us with information regarding the different groups that are working in defining what should a MAS architecture must have, the language with which the agents must communicate and how the ontologies used by agents could be defined. Kendall [10] mentioned that MASs should be designed based on principles of development of valid systems, rather than an architecture tailored to each project. This would allow the design and development of MASs based on a proper software architecture and design. He suggested a layer design of agents because a layered architecture allows: a) upper layers with more sophisticated behaviors, this layer should dependent on the capabilities that lower layers can provide. b) layer only depend on their own surrounding and neighbors elements and layer. And c) layer promotes bilateral communication between neighboring layers. The layer structure suggested: Sensory, Beliefs, Reasoning, Actions, Collaboration, Translation and Mobility. Sensory is the lowest layer and Mobility the highest layer [10]. The created agents may or may not contain all seven layers, but all should be designed with the ability to operate the seven layers. He also mentioned that the systems may contain four patterns agents, such patterns are: reactive, deliberative, the opportunistic and interface. Each agent acting according to the needs of the system, but with the behaviors needed to communicate with peer layers of other agents.

The behavior with which the agents are infused is important because this behavior can be benevolent or selfish [11]. The benevolent behavior is concerned to achieve agent common goal, while the selfish behavior only care to achievement of its own goals. This type of behavior is highly dependent on system requirements for which they were created, as there are functions in which selfishness is what will achieve the expected success of the project, where a benevolent behavior would not be appropriate [11]. For example: programing a machine agent with complete defined limits due to exceeding these limits could result in breaking down the machine requires a selfish agent to care for its own integrity. This does not prevent that any agent could have both behaviors, selfish and benevolent, since a machine agent should communicate their status to other agents, this behavior is a collaborative one.

Chaturvedi [12] identifies that agents can be used to simulate administrative and economic problems. As an example he mentioned stock market where different agents can be defined: buyers, sellers and controlling environment. Where buying and selling agents must achieve a balance so they can achieve adequate margins, meanwhile controlling agent should be aware regarding that the stock market rules are fulfilled. Jingrong [13] refers to the use of a MAS in order to identify the standards used by the flows of processes regarding an already implemented ERP system, since the use of agents allows that each functional unit are defined as an agent than could interacts with other agents in order to identify the exchange of knowledge and how they should handle coordination of functional units.

Ferber [14] in his work regarding analysis and design of organizations through the use of MAS, claims that its model can be used to describe any type of organization using only three elements: agents, roles and groups. Since agents can belong to any group and include several roles, which occur naturally in any organization. It is relevant to the study of ERP systems and the selection process because the selection process takes place within an organization, which could be modeled as a MAS.

Kosoresow [15] claims for a use of agent in computer supported cooperative work as auction, negotiation and haggling agents, using cash as metric to reach agreements with other agents. The agents will contain behavioral rules to follow; the negotiation agent will be the most complex, since it needs to know the opponent intention model.

Shi et al [16] claim that agent could be used it to develop three classes of systems: open, complex and ubiquitous computing. That is because agents are capable of create dynamically changing structures, make modular systems and interaction become an equal partnership between agents. They propose a four layer model for agent-based grid computing. They indicate that MultiAgent Environment (MAGE) is a good tool to support development of agent-based grid computing system.

Peleg [17] claims that main characteristic of distributed systems is that there may be autonomous processors active in the system at any moment. And all these processor should be in communication between each other to reach a reasonable level of cooperation.
They can be distributed in different geographic locations. The type and purpose of cooperation depends on the individual of each processor participant and user, as on MAS.

Peleg [17] also claims that issues and concerns regarding distributed computing could be categorized as communication, incomplete knowledge, coping with failures, timing and synchrony; and algorithmic and programming difficulties.

Marlowe [18] claims that distributed systems has the following defining properties: multiple computers (nodes), resource sharing, concurrency and message passing. Also the following characteristics properties: heterogeneity, handle of multiple protocols, openness, fault tolerance, persistence, security, insolation and decentralized control. All this properties allow a distributed system to perform a never-ending stream of diverse operations that should fulfill safety and liveness requirements.

Tanenbaum [19] define a distributed system as autonomous computers that work together to give the appearance of a single coherent system. A distributed system is transparent, scalable, open and failure tolerant. Its architecture is layer-based. The main goals of a distributed system are easily connect users to resources, hide the fact that resources are distributed across a network, allow the connection of different kind of computers at any single moment without stopping if a component crash.

Rotem-Gal-Oz [20] claims that software industry has been writing distributed system for several decades taking eight assumptions that is now known as the 8 fallacies of distributed computing: the network is reliable, latency is zero, bandwidth is infinite the network is secure, topology does not change, there is one administrator, transport cost is zero and the network is homogeneous. So, to create a distributed system designer need to address that network is unreliable, take latency in consideration to make as few as possible connections between distributed components, try to simulate production environment considering bandwidth availability, security is not only beyond perimeter, topology at production environment is in the wild and out of control, there are many administrators and all of them need to know how to diagnose a problem, transport of data cost a lot of money and resources, can increment latency and reduce bandwidth; and finally, network are not homogeneous, there are a lot of proprietary technologies and protocols, be aware of that.

3. MAS proposed architecture for ERP selection process

ERP selection process consists of several stages, among them are identification of needs and requirements, identification of suppliers, analysis of vendor ERP solutions, conform evaluation criteria, evaluation process and making decision [21-46].

The intent of the prototype proposed model is exclusive for evaluation process stage, which is where the selection process and decision making is modeled [1, 2]. This unstructured problem requires establishing evaluation criteria that has to be followed [2, 28, 47].

Based on literature [26, 28, 47, 48], we could define that the categories to be evaluated include: functional, operational, technical and economic aspects. These categories are the easiest to represent in a model and simulation tool, since these categories could be translated to quantitative rules easily.

As important as evaluation categories, the actors who performance the evaluation process are important. Such actors correspond to leadership roles in the implementation project [49]. Some of those roles are: chief executive officer, chief financial officer, chief market officer, chief operational officer, chief information officer and functional department managers.

Finally any ERP selection process needs to identify the ERP Solutions to evaluate. The solutions have individual characteristics, functionalities and features that need to be qualify.

In this case, a multi-agent environment could be a good place to model the evaluation process in ERP selection stage. At Figure 2 it can be observed the proposal MAS architecture.

In this architecture there are five different kinds of agents. Those agents could be composed by two databases, one for knowledge and behaviors and one to archive characteristics.
The five agents modeled are: Evaluate Characteristics Agent (ACE), Top Management Agent (AAD), Functional Manager Agent (AGF), ERP System Agent (ASERP) and Monitoring and Control Company Agent (AMCE).

ACE agents will be responsible to contain criteria information that must be met for each ERP. Also, ACE agents databases will contain the characteristics to evaluate and the rules that permit an interaction between those characteristics, with different levels of uncertainty. It also attends requests filed by AAD and AGF agents. ACE agent is a set of agents whose amount could varies according to the amount of aspects need it to evaluate.

AAD agents will contain features and personal profiles of top management behaviors, presented in accordance with data obtained from the ACE and ASERP agents. Also contain decision rules based on profiles of AAD, ACE and ASERP agents. AAD agent is a set of agents whose amount varies according to the number of people at the top management.

AGF agents will contain features and personal profiles of functional management and behaviors, presented in accordance with data obtained from ACE and ASERP agents. Also contain decision rules based on profiles of AGF, ACE and ASERP agents. AGF agent is a set of agents whose amount varies according to the number of people at the functional management, are very similar to AAD agents, but their level of influence in decision making is less.

ASERP agents will contain the information and characteristics of ERP systems, which are linked to the criteria that must be met for each of the characteristics evaluated. ASERP agents attend the request filed by AAD and AGF agents, so that they can apply their own set of rules of decision. ASERP agent is a set of agents whose amount varies according to the number of ERP systems to evaluate.

AMCE agent will be responsible to initiate the evaluation process. AMCE agent will be responsible to register and release ACE, ASERP, AAD and AGF agents. With ACE and ASERP agents, AMCE agent is going to keep communication related to their creation. ACE and ASERP agents will be responsible by themselves to administer their databases. With AAD and AGF agents, AMCE agent will maintain communication in order to inform them regarding the creation of new ACE and ASERP agents, so they can renew their decision-making processes according to new agents involved. AMCE agent will monitor the decision making process performance by AAD and AGF agents. Receive the correspond results from them and perform the final scoring process. AMCE agent will contain the necessary knowledge to issue a suggested ERP system solution based on the information contained at ASERP agents, based on evaluations submitted by the AAD and AGF agents, combined with the levels of influence on decisions of the agents mentioned.

The proposed MAS could be implemented as a distributed computing system as the architecture of MAS meets the definition of a distributed system.

Each of the agents can be placed on different computers which will connect to the central agent that will allow agents to communicate between them in order to achieve the purpose for which they were created.

The MAGE framework [16] contains all the features of a distributed system, just as the JADE framework. These frameworks allow the creation of a distributed system that enables communication between its components and the transparency necessary to identify the system as a single entity, with the possibility of scaling the system to different environments and equipment. Similarly, there is fault tolerance, because failures of one agent do not affect operation of the rest.
4. Conclusions

MAS are composed by one or more agents that behave as individuals. There is a set of control agents, which are responsible to sensor and monitor the environment where agents are, updating individual status and characteristics of agents. Controlling also fault tolerance, openness, transparency and security need it in a distributed computing system. All his activities should be done by the environment agent. It interacts with all the other agents and it is responsible to fulfill the goals for wish the MAS was development and also keep the coherence of a distributed computing system.

There are related literatures that allow the development of solid distributed computing system based in an agent, by example, Buyya [50] present a Network Enabled Parallel Simulator (NEPSi) which can be seen as a multi agent system since each component could be programmed as agents: The arbiter, the daemons and the Very High Speed Hardware Integrated Circuits Hardware Description Language simulator. Each one needs communication and coordination between each other to fulfill their particular goals.

JADE framework is a good platform on which MAS systems could be development. It contains most of the elements found in the infrastructure required for such types of systems, additionally, It meets some of existing standards in terms of agents and MAS. And it also fulfills requirements to develop distributed computing systems using message passing protocols that can be operated on multiprocessor or multicomputer systems.

MAS architecture form a solid distributed computational system that can be used to solve complex problems, since it permit multiple computers systems, resource sharing, concurrency, message passing heterogeneity, the use of multiple protocols, openness, fault tolerance, persistence, security, insolation and a partial decentralized control.

Multi agents systems can be used in the design of distributed decision support systems, where designers have the information needed to characterize the variables, interactions, behaviors, rules, standards, trends and scenarios of particular environments to be simulated.

Next step in this research include develop of a system prototype using JADE, including in its database all the selection variables found in literature, rules, algorithms, evaluation criteria and intelligence behavior. At the same time, perform an empirical study focus on Guadalajara, México organizations with the purpose of identify additional variables involve in ERP selection process and related stakeholders behavior.

Even after the evaluation and implementation of any kind of system, distributed multi agent systems can be used to process a success assessment of those systems. This success assessment system will require different model architecture that the presented in this paper, since it should handle other variables, relations and criteria rules; but certainly distributed multi agents system can be used in the resolution and simulation of the problems involved in the selection, implementation and post implementation of information systems.

Our proposal to mix MAS, DS and ERP in a decision support system prototype that help organization in the selection process of complex information systems, it is because this kind of decisions have multiple variables involved. They also have a lot of interaction between variables, rules and stakeholders. All this elements require high power computers because the calculus and application of evaluation criteria algorithms could be very computing demanding, since decision support and intelligence systems usually demand high computational power.

Future work in this topic could be oriented on the simulation of different decision scenarios and how those scenarios could affect the decision process and also the intelligence needs to evaluate the scenarios. As a secondary field of research, the graphical representation of the decision process using agents as stakeholders, characterize intelligence and program those agents with it, verifying in what kind of environment the agents are more efficient.

5. References


[30] U. Cebeci, "Fuzzy AHP-based decision support system for selecting ERP systems in


