International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:1, 2011

An Agent Oriented Architecture to Supply Integration in ERP Systems

Hassan Haghighi*, Sajad Ghorbani†, Maryam Mohebati‡, Mohammad Mahdi Javanmard§
Faculty of Electrical and Computer Engineering
Shahid Beheshti University
Tehran, Iran
h_haghighi@sbu.ac.ir*,sajad.ghorbani@qiau.ac.ir†,m.mohabati@qiau.ac.ir‡,ma.javanmard@Mail.sbu.ac.ir§

Abstract—One of the most important aspects expected from ERP systems is to integrate various operations existing in administrative, financial, commercial, human resources, and production departments of the consumer organization. Also, it is often needed to integrate the new ERP system with the organization legacy systems when implementing the ERP package in the organization. Without relying on an appropriate software architecture to realize the required integration, ERP implementation processes become error prone and time consuming; in some cases, the ERP implementation may even encounters serious risks. In this paper, we propose a new architecture that is based on the agent oriented vision and supplies the integration expected from ERP systems using several independent but cooperator agents. Besides integration which is the main issue of this paper, the presented architecture will address some aspects of intelligence and learning capabilities existing in ERP systems.

Keywords—enterprise resource planning, software architecture, agent oriented architecture, integration, intelligence, learning.

I. Introduction

ACCORDING to the severe rivalry for obtaining today's target markets, importance of fast respond to the continuously variant requirements doubles. On the other hand, nowadays trying to make added values on each work process in organizations is emphasized and attended. These affairs need new enterprise architecture which as a proper motivation, supplies necessary flexibility for future survival in the current millennium. Enterprise Resource Planning (abbreviated as ERP) systems support such strong architecture [9], [11], [12].

Although many explanations of ERP are given in the literature [13], we use the definition mentioned in [7] as the following paragraph:

An ERP is a collection of independent but integrated modules, ready to be operational (designed and engineered before and based on the best practices), but customizable and changeable, which integrates key commercial and management processes (from those processes based on data) in all aspects of the organization, such as administrative, financial, commercial, human resources and production, in order to create added values for the organization. This goal is achieved via effective planning and control of all enterprise resources.

ERP systems have direct and indirect advantages. Some of direct advantages of using ERP systems are: simplifying and reduction in financial, inventory and human resource operations, integration or elimination of current systems, increase in profits of the organization just a short time after implementation, reduction in cycle time (from receiving order to delivering the product or service), reduction in human resources for doing daily, regular operations. Some of indirect advantages of these systems can be integration of enterprise in both informative and applicative dimension, increasing consistency in data and information available in the organization, easier and more effective communication between different parts of the organization, using best practices of other similar corporations and organizations, and finally flexibility in making changes in the structure and processes of the organization [11], [13].

Although ERP systems have many advantages, these systems also have disadvantages which hazard their implementation [11], [2], [6]. Some of their disadvantages are high-risk in using them because of their sensitive implementation, reduction or elimination of some existing benefits because of a requirement to use standards inside ERP packages, and also high dependency between processes such that if one of processes cannot complete its task correctly and on time, then other related processes may encounter serious problems.

According to the definition given for ERP systems, one of the most important aspects of these systems is making integration between different operations in administrative, financial, commercial, human resources, and production departments in the consumer organization [2]. Because of this reason, one of the main issues for implementing ERP systems is making integration between functionalities existing in different parts of the organization and also making integration between the new ERP system with the legacy systems which are valuable for the organization. Lack of proper software architecture outfits for making integration, makes implementation delayed and longer than usual and also sometimes leads to serious risks.

In this paper, we try to offer a software architecture which is based on some independent but cooperator agents. This kind of agent oriented architecture helps supply integration during implementation of ERP Systems. This paper is based on work done in [10]. In [10], a model of multi-agent ERP system is given which ease implementation of ERP systems. Despite the offered model in [10] leans on software agents, it has not utilized important capabilities of these agents, such as learning, intelligence, and reproduction. Moreover, during introducing these agents, type of agents used and also their features are not indicated.

As mentioned above, in this paper an agent oriented software architecture is offered which effectively leans on different type of agents and also their features. In section 2, we

review different types of software agents along with their features, capabilities, and usage. Then, in section 3, we explain our offered architecture and in section 4, we give a case study which shows how we can apply our architecture when implementing ERP in a sample manufacture. Finally, the last section is devoted to the conclusion and directions for future work.

II. Software Agents

In computer science, software agents are part of software which operate as an interface in order to help users or other software. In fact, users assign the decision authority about what action must be done at each time to agents. These agents are created in order to make an easy and confident way for accomplishing tasks automatically in place of the user interference [4].

According to the Oxford dictionary, agent is defined as *somebody who is allowable to do something instead of other person*. Between computer and artificial intelligence society, concepts related to agents were defined beforehand with titles like software agents or intelligent agents (at the beginning of 80 decade). Despite the fact that different assumptions about agents existed before, there was a compromise about agents which said that an agent is *an isolated computer system set in some environments which can accomplish some tasks flexibly and also automatically in order to reach some planned goals*. In 2000, the word *software agent* indicated computer programs having two capability *autonomous execution* and *domain-based reasoning* [8].

A. Features of Software Agents

In general, agents have different features. Four important features of them are [8], [4], [3], [5]:

- Autonomy: software agents operate without humans or other agent's direct interference. Autonomy gives agents state of control over their operations and interior states.
- Sociability: agents can interact with other agents and also with human. Also, using different interaction languages, they can communicate with other agents.
- Reactivity: an agent can understand its environment and react to changes occurred in its environment. The environment can be real world, graphical user interface, other agents, or even internet.
- Pro-activity: agents can start some goal-based operations without any response to their environment.

B. Capabilities of Software Agents

In order to consider software agents as intelligent agents, they must have seven attributes [4], [1], [3]:

- Interior knowledge extraction and usage
- Fault tolerance against incorrect or unexpected input data
- Usage of special symbolism and also abstraction
- Goal-based behavior
- Learning from environment
- Realtime response
- Interaction with the natural language

Of course, sometimes an agent doesn't need all of these features. For example, an application software which only consist of *agent to agent* interactions do not need interaction with the natural language. Also, real time responding is not necessary for most of applications which require response in a specific time period. Finally, although learning is one of the most favorite features for agents, but we can make capable agents without this feature.

C. Types of Software Agents and Usage of Them

Based on the motion capability, ability to thought, roles, learning capability, and ability to autonomous operation, agents are divided in to 7 categories: cooperator, mobile, informative, internet-based, reactive, composite and intelligent. In continuation of this subsection, some kinds of agents are mentioned and also it is explained how they help users [4], [1], [3], [5], [10].

- Buyer agents: these software help internet users find their required products and services. For example, when a person tries to buy from eBay, at the bottom of the page, there is a list of products which are interested by users who searched that specific product. This idea is based on this assumption that user's tendencies are the same relatively and they search similar products. This technology which is feasible by usage of agents is named cooperative filtering.
- User agents: these agents are used in order to accomplish user's tasks automatically. For example, some of them categorize and order electronic mails according to their requests. Also, some of them fill the internet forms according to the saved user's information.
- Supervisor agents: these agents are used in order to monitor operations of one of equipments like computer systems. For example, agents which record goods quantity in manufactures, monitor contestant's price, or observe changes in stock market are some examples of this kind of agent.
- Data mining agents: this kind of agents is one of the most useful ones in Information Technology. They are used in order to find patterns and procedures from different information resources. Using this kind of agent, users can order the existing data based on his/her desired approach in order to access any information. For instance, there may be an agent which always checks changes in market's situation and reports changes to the users or incorporations so that they can make decisions more appropriately.

In summary, usage of agents are appropriate for situations in which applications consist of distributed computations, environment realization and monitoring, and autonomous behavior. Since agents have reasoning capability, using their interior knowledge, received messages, and their defined goals, they can accomplish sequence of complex computations easily. Every process control situation which must monitor real world and perform some actions in response to real time changes in the current state is a very good context for using agents. Sometimes these systems are as simple as thermometer and

sometimes as complex as control systems used for atomic reactors.

III. The Proposed Architecture

In order to supply integration of ERP systems, the proposed architecture in this paper consists of six kinds of software agent:

1) Coordinator agent
2) Data collection agent
3) Task agent
4) Scheduler agent
5) Priority Learning agent
6) Interface agent

The mentioned agents are put separately in each independent operational area (called *enterprise department* throughout this paper), and relation between enterprise units are accomplished by coordinator agents. The general schema of what we offer as a new architecture is shown in Fig. 1.

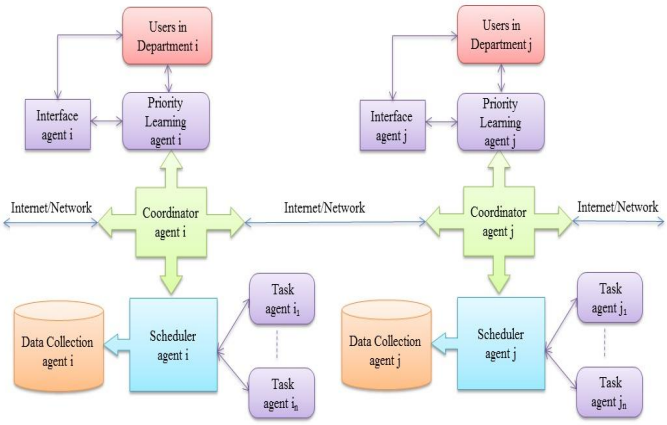

Fig. 1. The proposed agent oriented architecture

A coordinator agent is a delegate for each enterprise department to relate that department to other enterprise departments. This relation can be occurred via the internet infrastructure or based on a local network. Interface agents are used as connecting outfits between coordinator agents and users. Some task agents and also data collection agents are responsible for performing special processes in each enterprise department. The scheduler agent schedule requests originated from the coordinator agent and finally the priority learning agent observes user's requests, records them and has the capability to learn priorities of these requests. In continuing this section we explain the functionality and responsibility of each agent in detail and also explain more how they relate to each other.

A. Coordinator Agent

This agent has all the four attributes mentioned before about agent's, i.e., sociability, pro-activity, reactivity and the most important one, autonomy. This software agent plays an elegant role for enterprise departments when these departments try to communicate with each other. Also, each coordinator agent, as its name indicates, is responsible for controlling other agents inside that enterprise department in which the coordinator agent is. Each enterprise department can have one or more than one coordinator agent based on its complexity. Some of the most important responsibilities for this agent are:

- Receiving instructions and offering reports to the user via interface agents
- Sending request for any collection of data needed for each process to the related data collection agent and receiving this data from the data collection agent; notice that the management of data request is performed by task agent.
- Supplying required data for task agents, assigning tasks to them, and receiving feedbacks from them
- Connecting to other coordinator agents and supplying their require data

Based on its interior knowledge, a coordinator agent has the capability of monitoring, cooperation and conversation with other agents and reaction to different requests, such as assigning tasks to task agents and also data collection agent. It is worth mentioning that the capability of learning predicted in intelligent agents helps a coordinator agent to recognize multiplicity of tasks issued to a specific task agent such that the coordinator agent can ask that specific task agent to repeat itself. Now, for repetition of a task agent, we can use the reproduction attribute considered and predicted in intelligent agents.

B. Data Collection Agent

This agent is reactive and sociable but semiautonomous. Since this agent must operate in response to received requests, the pro-activity of this agent is not as much noticeable as the coordinator agent. Goals of data collection agents are performing queries on databases inside departments and gathering information requested from the coordinator agent residing on the same enterprise department. For doing this task, this data collection agent uses and processes its own interior knowledge.

The intelligence of data collection agents is used in recognition of invalid data and also missing data. This attribute supplies feasibility of restoration of complete and acceptable data to the coordinator agent. Moreover, for queries which are asked repeatedly, this agent can use a cache memory in order to return pre-retrieved results without applying the query to the database again. By using this technology, performance improves noticeably. The structure and capabilities of data collection agents differ because their capabilities somehow relate to the capability of the DBMS (Database Management System) or the data warehouse used inside the correspondent enterprise department. Some responsibilities related to data collection agents are:

- Retrieving required information for the coordinator agent existing in its own enterprise department
- Applying query to the DBMS or data warehouse existing in its own enterprise department

C. Task Agent

A task agent is also social, reactive and autonomous. Processes of a task agent have their own limited jurisdiction

and act independent of coordinator agents. For example, the agent that is assigned to supervise the variation of prices goes directly to providers (without referring to instruction of coordination agents), records the variation of prices, and then provides a report of recorded items.

There is a direct relationship between the complexity and the number of task agents with the number of responsibilities of an enterprise department. Furthermore, strategies that a task agent implements differs from one department to the other, depending on what is necessary to be done in each department. The responsibilities of the task agent are as follows:

- Receiving data from the coordinator agent
- Analyzing the data by special programs or algorithms
- Reporting results to the coordinator agent

D. Schedular Agent

A scheduler agent is a social, reactive and semi-autonomous agent. As its name indicates, this agent is supposed to schedule tasks and data demands received from the corresponding coordinator agent. The interesting point is that, this agent determines the length of the time that is required to process the orders, and accordingly selects the proper scheduling method; hence this agent sends the orders according to their time and scheduling method to the data collection agent. In addition, there is a set of rules and regulations that determine the priority of the orders and requests. These rules and regulations are either established independently by the scheduler agent or formulated directly by the administrator in each enterprise department. Also, the priority of assigning a task depends on the time required for processing tasks, the complexity of tasks and rules established by system administrators. The responsibilities of scheduler agents are as follows:

- Scheduling requests of the coordinator agent, and assigning them to the data collection agent
- Scheduling tasks and sending them to task agents

E. Priority Learning Agent

This agent is autonomous, social and pro-active. It can learn and keep priorities in user's point of view, and at the same time keep an eye on how users function. Taking heeds form the corresponding coordinator agent, a priority learning agent observes and records the user's orders and tendencies. As it will be illustrated in the example of the next section, the recorded data will be later used in the future decisions and also discussions between agents. The responsibilities of priority learning agents are as follows:

- Supervising and recording requests of users to reject or accept offers originated from the coordinator agent
- Learning priorities in user's point of view

F. Interface Agent

An interface agent is social and reactive. This agent acts as a tool to connect the corresponding coordinator agent to the user. An interface agent transforms the reports and the results of the coordinator agent into a comprehensible format, and delivers them to the user. It also processes user's orders, and delivers them to the coordinator agent. Furthermore, an interface agent can automatically report the end of the process to the user, though the user may not ask for the report. The responsibilities of user interface agents are as follows:

- Connecting the user and the corresponding coordinator agent to each other
- Receiving orders from the user
- Translating user's orders into comprehensible orders for the coordinator agent
- Analyzing the results for the user, warning him/her, and giving reports and notifications to him/her

IV. A Case Study: Order Processing

In order to illustrate the function of the proposed agent oriented architecture, this section provides an example about processing orders in a company. The investigated process takes a 4-step approach to answer specific requests existing in the company. This example is extracted from [10], but is changed in some ways to meet the requirements of the proposed architecture of this paper.

A. Environment

To make it much more comprehensible, let's suppose a company with 5 departments: marketing department, production department, accounting department, inventory management department, and distribution department. Furthermore, each of the mentioned departments has its own information system, database, and data architecture. Also, each department has its own coordination agent as well. Because of the complexity of tasks in the production department, we have two coordinator agents in this department: one for improving the quality of products, and the other for scheduling the production. Also, each coordinator agent has its own interface, data collection, scheduler and task agents. The processing environment is shown in Fig. 2.

B. A Common Process

Suppose that a marketing user wants to know "Is company able to provide the product m for as many as n, at price P, for the client X of city C, on Wednesday?" This query will be replied in four steps.

The First Stage: The interface agent of the marketing department asks the coordinator agent of the same department this question: "Is company at the present time able to accept the order of the client W of city M for number X of product Y, at price Z on Wednesday?"

As it can be seen, the parameters of the user's order differs from that of the interface agent. It is because of the fact that the communication language between the user and the interface agent differs from the colloquial langauge of the interface and coordinator agents.

The Second Stage: Based on the knowledge of its own jurisdiction, the coordinator agent of the marketing department processes four tasks simultaneously:

1) Communicates with the coordinator agent of the inventory management department to check the amount of the existing product M.

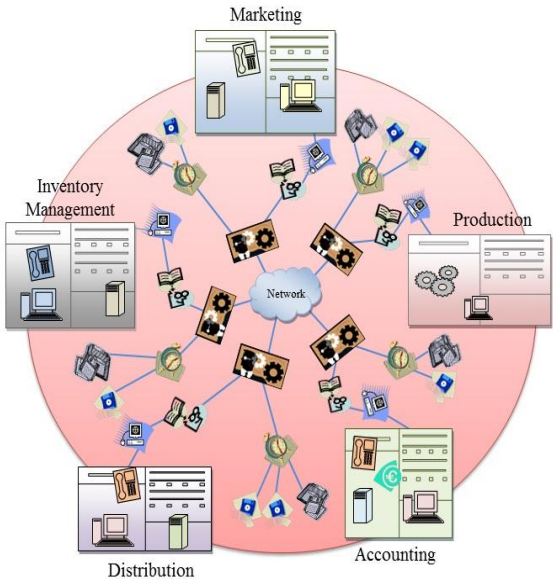

Fig. 2. The processing environment

2) Communicates with the coordinator agent of the distribution department to get information on distribution of the product.
3) Asks the data collection agent of the marketing department to provide it with information about the price of product M.
4) Keeps an eye on the status of the information requested from other agents.

The Third Stage: As soon as the orders of the coordinator agent of the marketing department are received in the second stage, the below three tasks are done in different departments. They are scheduler agents that schedule these tasks:

1) In the inventory management department, the coordinator agent asks the data collection about the amount of the existing product M.
   - The data collection agent asks the corresponding database, and reports the result to the coordinating agent.
   - The coordinating agent of the inventory management department sends the result to the coordinating agent of the marketing department.
2) The coordinator agent of the distribution department sends a request to the task agent of the same department to check whether it is possible to schedule a sale of N number of product M delivered on city C until Wednesday.
   - The task agent will send the results back to the coordinating agent of the distribution department.
   - Then the coordinator agent of the distribution department will send the same results to the coordinator agent of the marketing department.
3) In the marketing department the data collection agent will ask its own database about the price of the product M; it will then calculate the price of N number of product M.

The Fourth Stage: Based on its own interior information and just after receiving all required information from the coordinator agents of the distribution and inventory management departments and also its related data collection agent, the coordinator agent of the marketing department evaluates information and gives its own offer to the user. Based on general results, two categories of procedures are accomplished by the marketing coordinator agent.

(a) First Case: all the conditions for acceptance of the order are satisfied.

1) The interface agent notifies the user that all the conditions for acceptance of the order are satisfied.
2) If the user still requests his/her order, the interface agent of the marketing department makes a connection with its related coordinator agent for accomplishing a sequence of actions written below:
   - The marketing coordinator agent asks the distribution coordinator agent to accomplish the delivery.
   - Then, the distribution coordinator agent issues a request to the corresponding task agent to schedule the delivery process.
3) If the user rejects his/her order, the marketing interface agent accomplishes a sequence of actions written below:
   - The marketing priority learning agent records user's decision in order to be able to predict user's actions in future. .
   - The marketing interface agent notifies its related coordinator agent in order to discard the order.

(a) Second Case: At least on condition for acceptance of the order is not satisfied.

One of the main capabilities of the proposed architecture in this paper is that agents can discuss with each other with or without interference of the user. For example, if at least on condition for acceptance of the order is not satisfied, agents will accomplish a sequence of discussion written below:

1) If the current cost ($P^{'}$) of product m is more than the requested price (P), then the marketing coordinator agent will discuss with the accounting coordinator agent if the requested price for this number of product is acceptable or not?
2) The accounting coordinator agent issues a request to its related task agent in order to estimate P^* (i.e., the least feasible price) for a request of n number of product m.
3) Based on the answer from the related task agent, the accounting coordinator agent accomplishes a sequence of tasks written below:
   - If $P^* \leq P$, then procedures mentioned in the first case will be accomplished.
   - If $P^* > P$, then the least acceptable price (P^*) will be returned to the the marketing coordinator agent.
   - The marketing coordinator agent will notify the user of P^* via its related interface agent.
   - If the user accepts the new price, then a sequence of actions for accepting the order explained in the first

case will accomplished. Also, the accounting priority learning agent will store user's decision. This action is done for future user's tendency prediction.
- If the user doesn't accept the new price, then process will be finished.

V. Conclusions and Future Work

In this paper, with an agent oriented approach, we have proposed a software architecture which assists to supply intended integration during the implementation of ERP based on employing independent but cooperator agents. The basis of the offered architecture is the work done in [10] but unlike the mentioned research, the current model have leaned on expectations of different types of agents and also their effective features. For example, for any agent offered, its type and favorite attributes were mentioned. Most important capabilities of software agents, such as intelligence, learning, cooperation, autonomy and even reproduction were used.

For other researches which are close to what has been offered in this paper and thus can be done in continuing this work in future, we propose to design new software architectures in order to supply other features of ERP systems, such as intelligent notification, having development environment for providing flexibility, gathering and using best practices, software distribution management and finally dynamic document generation.

References

[1] Bose, R., Sugumaran, V., *Application of Intelligent Agent Technology for Managerial Data Analysis and Mining*, Database for Advances in Information Systems, Vol. 30, No. 1, pp. 77-94, 1999.
[2] Calisir, F., *The Relation of Interface Usability Characteristics, Perceived Usefulness, and Perceived Ease of Use to End-User Satisfaction with Enterprise Resource Planning (ERP) Systems*, Computers in Human Behavior, Vol. 20, No. 4, pp. 505-515, 2004.
[3] Chauhan, D., Baker, A., *JAFMAS: A Multi-Agent Application Development System*, Proc. of the Second International Conference on Autonomous Agents, pp. 100-107, 1998.
[4] Coen, M., SodaBot: A Software Agent Environment and Construction System, MIT AI Lab Technical Report 1493, 1994.
[5] Davies, W., Edwards, P., *Agent-based Knowledge Discovery*, Proc. of AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, pp. 34-37, 1995.
[6] Grabski, S. V., Leech, S. A., *Complementary Controls and ERP Implementation Success*, International Journal of Accounting Information Systems, Vol. 8, 2007.
[7] Haghighi, H., Shahhosseini, H. S., Mobasheri, M., Enterprise Resource Planning Software: Development, Evaluation, Selection, and Implementation (In Persian), Nashre Nikpey, 2010.
[8] Henderson-Sellers, B., Giorgini, P., Agent-Oriented Methodologies, Idea Group Publishing, 2005.
[9] Jacobs, F. R., Clay, D., Why ERP? A Primer on SAP Implementation, McGraw Hill, 2000.
[10] Lea, B., Gupta, M. C., Yu, W., *A Prototype Multi-Agent ERP System: An Integrated Architecture and a Conceptual Framework*, Technovation, ol. 25, pp. 433-441, 2005.
[11] O'Leary, D., Enterprise Resource Planning Systems: Systems, Life Cycle, Electronic Commerce, and Risk, Cambridge University Press, 2000.
[12] Sudzina, F., Johansson, B., *Finding ERP Requirements that Support Strategic Management in Organizations*, Proc. of Academic International Conf., Increasing Competitiveness or Regional, National and International Markets Development - New Challenges, 2007.
[13] Sumner, M., Enterprise Resource Planning, Upper Saddle River, NJ, Prentice Hall, 2004.