International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:11, 2011

An Agent Oriented Architecture to Supply Dynamic Document Generation in ERP Systems

Hassan Haghighi, Seyedeh Zahra Hosseini, Seyedeh Elahe Jalambadani

***Abstract*—**One of the most important aspects expected from an ERP system is to mange user\administrator manual documents dynamically. Since an ERP package is frequently changed during its implementation in customer sites, it is often needed to add new documents and/or apply required changes to existing documents in order to cover new or changed capabilities. The worse is that since these changes occur continuously, the corresponding documents should be updated dynamically; otherwise, implementing the ERP package in the organization encounters serious risks. In this paper, we propose a new architecture which is based on the agent oriented vision and supplies the dynamic document generation expected from ERP systems using several independent but cooperative agents. Beside the dynamic document generation which is the main issue of this paper, the presented architecture will address some aspects of intelligence and learning capabilities existing in ERP.

***Keywords*—**enterprise resource planning, dynamic document generation, software architecture, agent oriented architecture, learning, intelligence

I. Introduction

VALUE added is one of strategies for today challenging in work process that is emphasized and attended in organizations. 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].

An ERP is a collection of independent but integrated modules, ready to be operational (designed and engineered before and based on 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 [7, 13, 14].

H. H. Faculty of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran (phone: 9821-2990-4136; e-mail: h_haghighi@sbu.ac.ir).
S. Z. H. Faculty of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran (e-mail: z.hoseini@mail.sbu.ac.ir).
S. E. J. Faculty of Electrical and Computer Engineering, Shahid Beheshti University, Tehran, Iran (e-mail: e.jalambadani@ mail.sbu.ac.ir).

ERP systems have a lot of benefits, including improving work process, facilitating access to available data for decision making, quick response to changing business operations and market condition, resulting in improved competitive advantages and finally more flexibility in changing organization structures and behavior [11, 13].

Although ERP systems have many advantages, these systems also have disadvantages which hazard their usability [2, 6, 11]. Some of their disadvantages are high risk in using them because of their sensitive implementation and reduction (or even elimination) of some benefits of existing processes in the organization because of a requirement to use standards inside ERP packages.

According to the definition given for ERP systems and also because of high risk during their implementation, supporting dynamic document generation is one of the important aspects of these systems. Because of frequent changes in an ERP package during its implementation, it is required to apply changes to associative user\administrator manual documents as soon possible. More precisely, since these changes occur continuously, the corresponding documents should be updated dynamically; otherwise, implementing the ERP package in the organization encounters serious risks because users\administrators cannot be aware of numerous and considerable changes applied recently. This absolutely results in less usability and even malfunctioning yielding high risks when using an enterprise scale package, such as ERP.

In this paper, an agent oriented software architecture is offered which supplies the dynamic document generation expected from ERP systems. This architecture effectively leans on different types 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. Section 4 is where we give a case study which shows how we can apply our architecture in a sample ERP system. 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 systems. 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 allowed to do something instead of another*

person. In computer and AI societies, concepts related to agents were defined beforehand with titles like software agents or intelligent agents (at the beginning of 80s). 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. Some of important ones are [3, 4, 5, 8]:

- 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 using different interaction languages.
- 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.
- Transparency and accountability: an agent can be clear and when needed, records its activities.
- Acting as proxy: agents can act as the proxy of a person, do according to her favor, accept his responsibility or try for her benefit.
- Cooperation: an agent can act cooperative with other agents; the success or fail relates to this relationship.
- Able to deal with heterogeneity: an agent can be able to deal with heterogeneity of other agents or databases.
- Reproduction: an agent can create a new agent like itself.

B. Capabilities of Software Agents

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

- 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 using the natural language

Of course, sometimes an agent does not need all of these features. For example, application software which only consists of agent to agent interactions does not need interaction by 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 [1, 3, 5, 10].

- Buyer agents: these software agents 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 the 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 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 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 situation and reports changes to the users or incorporations so that they can make decisions more appropriately.

In summary, usages 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 a 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 dynamic document generation in ERP systems, the proposed architecture in this paper consists of four software agents:

1) Data collector & editor agent
2) Keyword agent

3) Classification agent
4) OCR agent

At first, the user creates initial documents and gives them to the OCR agent. The OCR agent determines different parts of the documents and creates text documents. Then the Keyword agent finds document keywords and sends them along with the documents themselves to the Data Collector & Editor agent to apply changes to relative documents.

The general schema of what we offer as our architecture is shown in Fig. 1.

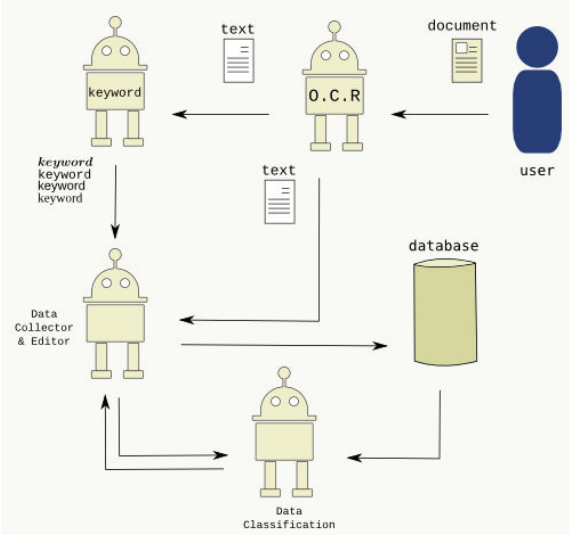

Fig. 1 The proposed agent oriented architecture

A. Keyword agent

This agent is proxy, proactive and transparent and accountable but semiautonomous since this agent operates in response to received requests.

The main goal of the Keyword agent is to find keywords from documents. It is supposed that each document has special fields in which the author writes related keywords. The writer could give a weight to each word, i.e., higher weights show more relative words. The Keyword agent will deal with these defined keywords. The user checks\unchecks some box to inform whether the entered keywords are complete. If not, the Keyword agent should find extra keywords. Three cases may occur.

- When the user herself determines all keywords for a document and approves that they are complete, the system considers them as final keywords.
- If there is not any keyword provided for a document, the Keyword agent finds some keywords using its learning capability.
- When the user does not mention all required keywords, the Keyword agent uses the ontology system to generate some new keywords and let the user select from them.

The Keyword agent consists of several parts (see Fig. 2):

1) *Keyword database*: in this database, we collect keywords used in stored documents.
2) *Adding keyword agent:* this agent adds keywords to the database by applying specific criteria.
3) *Similar keyword finder agent*: when the user selects some keywords for a document, if the selected keywords are not in the database, this agent will show similar keywords to the user by ontology. Then the user selects one of the recommended keywords or introduces a relationship between her keyword and recommended ones. When the user wants to declare relationships, the selected keywords (provided by the user), the recommended keywords (provided by the agent) and the relationship itself are sent to the Adding keyword agent. This agent stores both keywords and their relationship into the database. If the keyword which is selected by the user does not have the similar meaning with the agent's keyword, the user introduces a new keyword and sends it to the Adding keyword agent; if this is a new keyword, it will be stored in the database and after that, the user can use it as keyword.

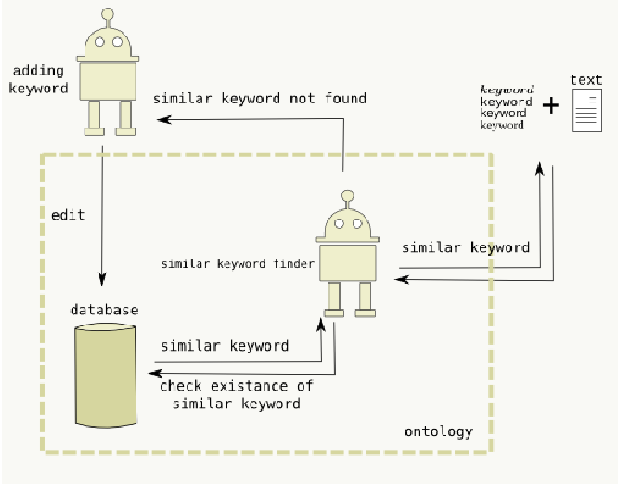

Fig. 2 The Keyword agent

B. Data collector & editor agent

This agent finds those parts of documents that should be updated. This agent is sociable because it communicates with other agents and adaptive because it adapts its goal from system goals using its learning capability.

The details of how this agent works are represented in Fig. 3. It first receives an input document along with its keywords from the Keyword agent and sends a request to the Classification agent to find those documents which are related

to the input document. Then this agent finds those places of the related documents which should be changed according to the input document and its keywords. Finally, it shows the result to the user who approves\rejects changes in order to be aware of her comments. In this way, this agent applies the comments of the user to the documents and also learns new something to consider when it is supposed to apply changes later.

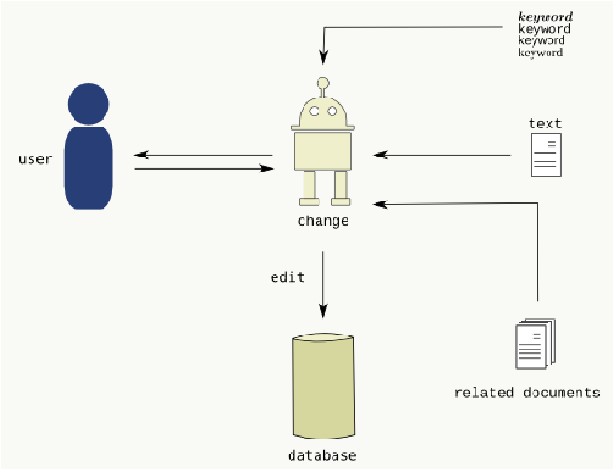

Fig. 3 The Data collector & editing agent

C. Classification agent

This agent is autonomous, cooperative, able to deal with heterogeneity and able to do reproduction. This agent directly connects to the keyword database of the Keyword agent and creates a new Classification agent for each keyword stored in this database using its reproduction capability. More precisely, if a new keyword is added to the keyword database, the Classification agent creates a new classification agent which is responsible for this new keyword. Also, if a change occurs in the keyword database, the main Classification agent updates the information of created Classification agents accordingly; the details of this agent are shown in Fig. 4.

Each Classification agent has its own database containing addresses of those documents related to the keyword which is under the responsibility of that agent. If the main Classification agent receives some keyword, this keyword is sent to all the classification agents. If the keyword corresponds to one of the classification agents, that agent finds and returns addresses of related documents.

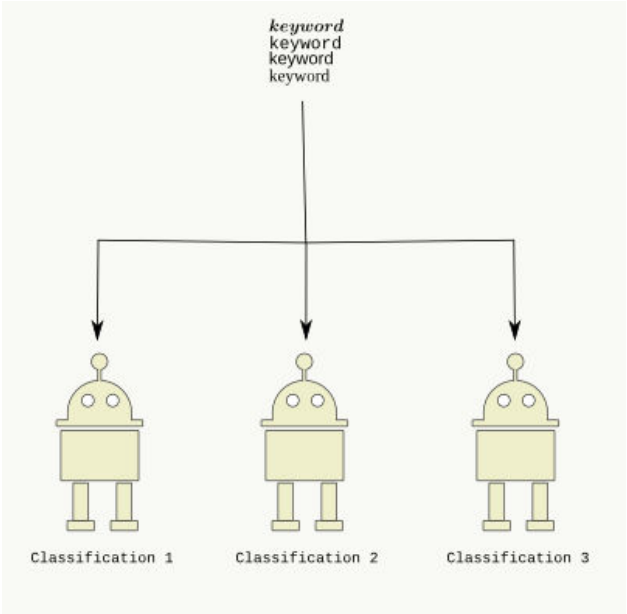

Fig. 4 The Classification agent

D. OCR agent

This agent is autonomous and proactive. It also has learning capability. OCR (optical character recognition) allows converting scanned documents into searchable text files. Advantages of using OCR are:

- Fast speed in accessing information
- Reduction in the space of storing

The OCR agent consists of five parts (see Fig. 5):

1. Preprocessing
2. Segmentation
3. Representation
4. Reorganization
5. Postprocessing

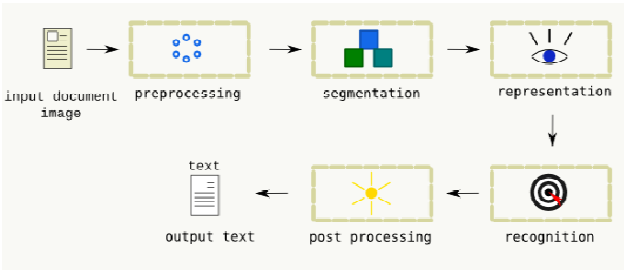

Fig. 5 The OCR agent

The preprocessing step is called for all activities that should be done on the image beforehand, like making the image binary and removing noises. The segmentation stage which is an important part of an OCR system covers both external and internal segmentation. In the representation step, each pattern is corresponded to a feature vector. In the recognition step, however, each pattern is corresponded to one of the space model classes. The last step, i.e., postprocessing, tries to

improve the result of the recognition step using extra information like the set of valid words.

IV. A Case Study: Adding document

In order to illustrate the functionality of the proposed agent oriented architecture, this section provides an example showing how to process adding a new document. The investigated process takes a 4-step approach to answer specific requests existing in the organization. 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

We suppose the implementation team wants to add new documents to the system. New documents may have some heterogeneity with other documents, but the new document must apply changes to the older ones.

B. A Common Process

Suppose that one of the users of the implementation team wants to add a new document. There are a lot of documents in the database which have conflicts with the new one and thus somewhere in them should be changed accordingly. Adding the new document and applying change to other documents are processed in five steps:

The first stage: the user creates the new document.

(a) *First case:* the user inserts all of keywords.
(b) *Second case:* the user adds some keywords but not all of them.
(c) *Third case:* the user does not add any keyword.

The second stage: the OCR agent converts the document to a searchable text.

The third stage: the Keyword agent finds keywords of the new document.

(a) *First case:* the user inserts all of keywords - The Keyword agent uses user's keywords as final document keywords.
(b) *Second case:* the user adds some keywords but not all of them - The Keyword agent uses ontology to add new keywords.
(d) *Third case:* the user does not add any keyword - The Keyword agent uses ontology and its learning capability to add new keywords.

The fourth stage: the Classification agent classifies documents which are in the database and then returns those documents related to the provided keywords.

The fifth stage: the Data collector & editor agent collects related documents (sending request to the Classification agent), finds places which should be changed and applies changes to them.

V. Conclusion

In this paper, based on several independent but cooperative agents, a software architecture has been suggested helping to implement dynamic report generation in EPR systems. In this architecture some features of agents, such as intelligence, learning, cooperation, autonomy and even reproduction were used to achieve this objective

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 agent oriented software architectures in order to supply other features of ERP systems, such as having development environment for providing flexibility, gathering and using best practices and finally software distribution management.

References

[1] Coen, M., SodaBot: A Software Agent Environment and Construction System, MIT AI Lab Technical Report 1493, 1994.
[2] O'Leary, D., Enterprise Resource Planning Systems: Systems, Life Cycle, Electronic Commerce, and Risk, Cambridge University Press, 2000.
[3] 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, Computer in Human Behavior, Vol. 20. No. 4, pp. 505-515, 2004.
[4] Grabski, S. V., Leech, S. A., Complementary Controls and ERP Implementation Success, International Journal of Accounting Information Systems, Vol. 8, 2007.
[5] Jacobs, F. R., Clay, D., Why ERP? A Primer on SAP Implementation, McGraw Hill, 2000.
[6] 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.
[7] Haghighi, H., Shahhosseini, H. S., Mobasheri, M., Enterprise Resource Planning Software: Development, Evaluation, Selection, and Implementation (In Persian), Tahsin Publisher, 2010.
[8] Sumner, M., Enterprise Resource Planning, Upper Saddle River, NJ, Prentice Hall, 2004.
[9] Henderson-Sellers, B., Giorgini, P., Agent-Oriented Methodologies, Idea Group Publishing, 2005.
[10] Chauhan, D., Baker, A., JAFMAS: A Multi-Agent Application Development System, Proc. of the Second International Conference on Autonomous Agents, pp. 100-107, 1998.
[11] 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.
[12] 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.
[13] 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.
[14] Haghighi, H., Ghorbani, S., Mohebati, M., Javanmard, M.M., An Agent Oriented Architecture to Supply Integration in ERP systems. Word Academy of Science Engineering and Technology. Issue 73, 2011.