Rule-Based Agents for Workflow Applications in Manufacturing Information Systems

Ernesto Montaldo, Roberto Sacile, Antonio Boccalatte
LIDO Lab - Department of Communication, Computer and System Sciences (DIST)
University of Genova
Via Opera Pia 13, 16145, Genova – Italy
{montaldo, robby, nino}@dist.unige.it

Abstract

Manufacturing information systems need to integrate and to enhance different information functionalities such as workflow management. Software agents can solve the problem as an additional software layer of an existing manufacturing information system. In this respect, several current research projects focus on workflow management, with the aim of an integration and coordination among plant and business activities.
The approach presented in this paper can be classified as an agent-based architecture to enhance an existing workflow management system, in order to manage new functionalities such as customer relationship management in electronic commerce. This approach is particularly suitable for small medium enterprises with simple manufacturing information system and simple or inexistent computer based workflow management.
A formalization of our approach, and some important aspects related to the definition of a simple workflow are shown.

1. Introduction

In recent years, in order to deal with the need of rapid, continuous changes, computer science is challenged to develop novel interrelated information and communication technology, and to align them with the social needs of co-operating user groups. Workflow systems are among the most advertised technologies addressing this trend [1].

A definition about workflow, which has been given by Workflow Management Coalition, is: *the computerized facilitation or automation of a business process in whole or part* [2].

In order to satisfy manufacturing, and in particular workflow management requirements, researchers have tried to apply new technologies as the one represented by software agents [3-6].

Software agents, defined in a general way by Wooldridge as *a computer system that is situated in some environment, and that is capable of autonomous actions in this environment in order to meet its design objectives* [7], are a practical and successful example of academic research transferred to industrial applications. They can effectively combine all the advantages of distributed systems with decision support, management, and Artificial Intelligence (AI) techniques giving a significantly opportunity to develop distributed systems.

Especially, in manufacturing, software agents can be applied to enhance existing manufacturing information systems with the aim of a two-fold integration. The first one regarding vertical integration in order to integrate plant and business processes, and the second one regarding horizontal integration in order to implement automatic information procedures according to a proper distributed workflow within and outside the enterprise boundaries. Our approach, following the basic agent architectures applied to workflow [4,5], can be classified as an agent-based architecture enhanced to an existing workflow management system, in order to manage an additional commercial layer that is based on electronic commerce technology.

In this work, we propose an ongoing approach to formalize rule-based agent behavior oriented to workflow applications in information systems domain, and the general agent architecture that it is possible to develop using this formalization jointly with rule-based systems theory. The description is related to a rule-based agent system oriented to workflow management in information system environment we are developing.

In section 2, formalism applied to our approach will be provided. Section 3 will discuss the system architecture based on the formalization.

2. The Knowledge Model

One of the concepts related with agent theory is the concept of environment. The agent has to interact, in an autonomous way, with a specific environment to perform actions that are able to change the environment state. Typically, it is important that agents are able to show intelligent behavior to perform the actions. In according with the general concept of learning and intelligence, a system that is capable of learning deserves to be called intelligent, and conversely, a system being considered intelligent is usually expected to be able to learn [8].

In a particular environment, as the one characterized by the information processes in information systems, it may be sufficient that agents are able to learn, although they are still required to reproduce human oriented behavior in frequent, repetitive, multi-alternative tasks.

Humans often solve complex problems using symbolic approaches, and researchers of artificial intelligence have developed techniques embodied in languages or tools, which allow programs to be built resembling human logic, to model the information at higher levels of abstraction.

Rule-based programming is one of the most commonly used techniques for developing them. Rules are used to represent heuristics that specify a set of actions to be performed for a given situation [9]. Each rule is in the form *if* <condition/s> *then* <action/s>, where the conditions are represented with *facts*.

Workflow management and its tasks are naturally rule-based, and simple graphical representations, such as flowcharts, are still commonly used in enterprises.

2.1. Formalization of the Agent Behavior

Several formalizations of agent behavior and multi-agent systems behavior can be found in literature [7, 10-14], an even greater set of references can be found for the formalization of rule-based systems (see [9], for an introduction to the problem and for specific references).

In our approach, an agent acts in an environment that is an information system, such as the management information system of an enterprise. Each agent is autonomous and can perform actions according to its knowledge model. Each action can be an SQL statement applied to database, or a message to other agents, or e-mail to Internet users.

The model is related with the information system environment and with the agent knowledge. The environment can be represented with an information state vector

$$I(t) = \{i_1, i_2, .. i_n, .., i_N\}, i_n \in I_n(t)$$

whose elements representing specific, variable, and time dependent information, can change according to the variations of the associated information stored in the database of the information system. In itself, the environment is active. It has its own functionalities that can change its state, independently of the actions of its embedded agents.

In a general way, a generic agent Ag_i can be expressed as: Ag_i = (Internal_State, Input, Output, Knowledge).

The *Internal_State* is represented with *facts* that completely define the agent. The facts are related to the state of the information system environment, and they represent the mapping between the information state vector $I(t)$, and the *facts space* F (the agent state).

If a fact is defined, only one information space element is associated to it, but one information space element can be associated at more than a single fact.

The *input* represents the sensorial activity the agent Ag_i performs in the environment. It represents the stimulus the agent receives from the information system, and the stimulus (messages) the agent receives from other agent.

The *Knowledge* is represented by a vector of rules:

$$\bar{r} = \{r_1, r_2, .. r_m, .., r_M\} = f(I(t)).$$

Each component represents a particular rule that defines the behavior for a particular state.

The *output* represents the actions the generic agent Ag_i performs in the environment.

A set of actions A

$$A = \{a_1, a_2, .., a_K\}$$

represents a full set of actions, and can be viewed as the union of three sub-sets of actions that are: querying the state of $I(t)$ (actions belonging to $Q_{Ag_i,I}$ set); sending messages to other agents (actions belonging to $\bigcup_j M_{Ag_i,Ag_j}$); e-mailing Internet users (actions belonging to $E_{Ag_i,U}$)

The set of actions can be written as follows:

$$A = \bigcup_j M_{Ag_i,Ag_j} \cup Q_{Ag_i,I(t)} \cup E_{Ag_i,U} .$$

At an instant t (according either to the agent reaction frequency, or to the stimulation of a message $m \in M_{in}$ by another agent), the agent Ag_i can act following a set of sequence of actions $\tilde{A}_{Ag_i}$ in relation to the content of the possible received messages, the specific state of a queried subset of $I(t)$, its own state, and its reasoning capability, formalized in a rule-based memory. An inference engine is able to calculate the behavior, which is formalized as follows:

$$B_{Ag_i}(t) : (M_{in} \times I_1 \times I_2 \times .. \times I_n) \otimes (F, \bar{r}) \rightarrow \tilde{A}_{Ag_i},$$

according to an internal algorithm $(\otimes)$, where F represents the facts space, and $\bar{r}$ represents the rule vector.

A generic sequence A_j is defined as

$$A_j \in \tilde{A}_{Ag_i}$$

$$A_j = (a_1, a_2, .., a_b, .., a_{BAg_i}), a_b \in A$$

The aim of this formalization, for the particular architecture we are developing, is to represent in an abstract way, an agent as a function which has the space of rules as domain and the space of actions as range, in order to study in a rigorous way the agent's behavior.

3. The Agent Knowledge Based Architecture for Workflow Applications

According to the previous formalization, we have developed a software architecture allowing the modeling of an agent. A GUI (Graphical User Interface) allows transferring the user knowledge to the agent, in a rule-based model. The model is then translated in the CLIPS (C Language Integrated Production Systems) language [9]. In a specific workspace, the user can develop a flowchart schema based on blocks representing one or more *if-then* rules. Different facts, that represent the agent state, compose the left side of a rule. The agent state is initialized by querying the database, by a message of another agent, and/or by an action of the same agent. If all the facts of the left side of a rule are true, the rule can fire. When a rule fires, a set of actions is performed.

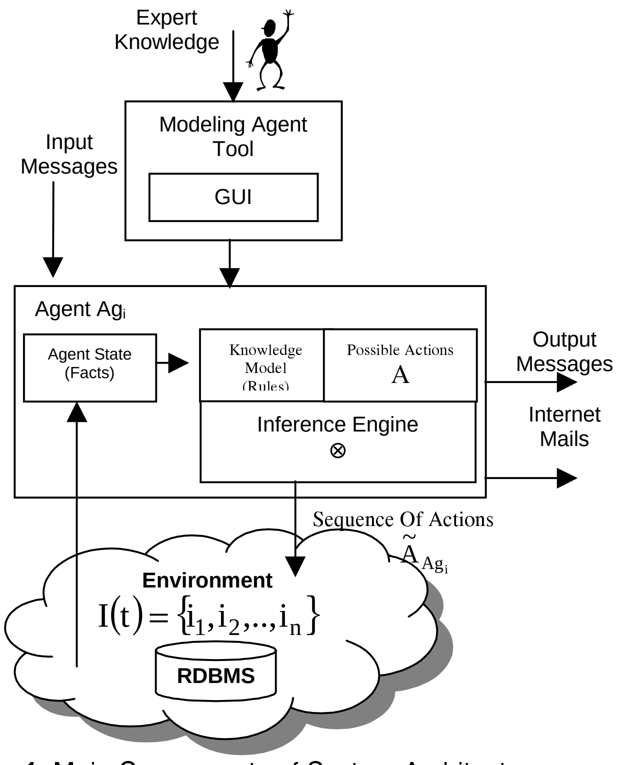

Fig. 1. Main Components of System Architecture

Fig.1 shows the system architecture. Facts, rules, and inference engine are the primary components for the agent knowledge model. The inference engine controls the execution of rules, in particular it decides which rule should be executed and when, thus defining the proper sequence of actions, $\tilde{A}_{Ag_i}$. The output messages and the Internet mails, which are separated from the other actions in order to be emphasized, have to be considered as part of the sequence of actions.

4. Rule-Based Agents applied to Workflow in a simple Manufacturing Information System

One of the urgent needs of manufacturing information systems is to integrate and to enhance different information and related functionalities such as workflow management. Software agents can solve the problem as a new software layer, to be added to an existing manufacturing information system. Our approach, following the basic agent architectures applied to workflow [4,5], can be classified as an agent-based architecture enhanced to an existing workflow management system.

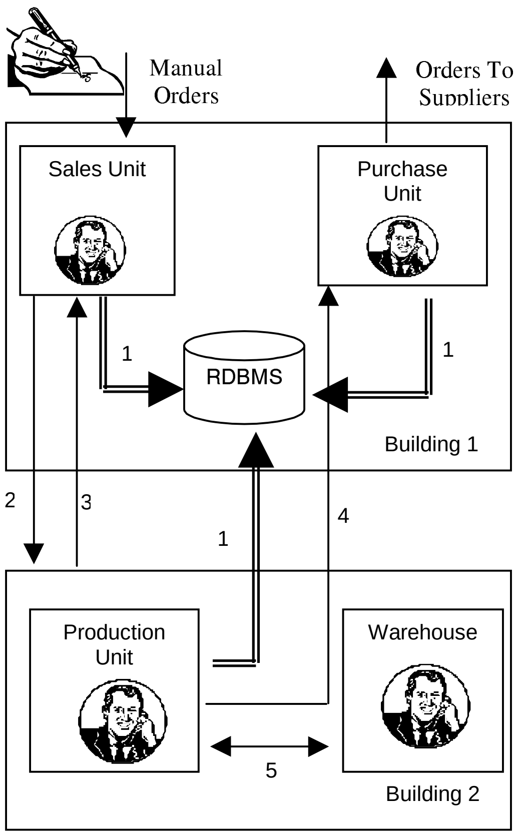

Fig. 2. Representation of a very simple workflow in a classical enterprise

Our case can be summarized as follows (fig. 2). A small enterprise is supposed to produce bicycles. The enterprise is distributed in two main buildings

The current workflow of this company is based on a traditional simplified configuration made of three units: purchase, production and sales. The purchase and sales units are in the same building (administration). A simple manufacturing information system, based on a traditional RDBMS, is based on the following tables: customers, suppliers, orders-from-customers, orders-to-suppliers, raw-material-warehouse, bill-of-material, products-warehouse, and production scheduling. Three PCs are present in the administration building: one server containing the RDBMS, and one client for each unit. The production unit, which is in another building geographically remote from the administration building, uses a PC as a RDBMS client, to update warehouse tables and to access production scheduling.

The LIDO-BIKE workflow is also very simple (fig. 2). The main tasks of the sales unit are related to the management of customer orders from resellers, and to the scheduling of the production. The main tasks of the purchase unit is to verify that the raw material warehouse can satisfy production on a short medium term, and to send orders to raw material purchasers. The main tasks of the production unit are related to the management of the production according to the production scheduling, and to verify and update the warehouse information. In particular, we have that:

1	Querying the database
2	Production request
3	Order ready
4	Component request
5	Production request/ acknowledge request.

LIDO-BIKE would like to expand its marketing on the Web to single customers, receiving orders by resellers also via Web, modifying the production but possibly leaving its simple managing structure unchanged as much as possible.

A Web site is so designed, giving access to the orders-from-customers table for order insertion to thousands of probable new customers. Three agencies are added to a manufacturing information system of a Small Medium Enterprise (SME) in parallel to the workflow of the managing units: purchase agency, production agency and sales agency. Each of these agencies can manage a specific sub-workflow, which is proposed in a simplified view for brevity.
Specifically:

– The sales agency perform these actions: to control if a new order is arrived, to control whether it is possible to ship the order (in this case to send an e-mail to customer and to update the RDBMS) or not (in this case to send a message to the production agency), to get messages from other agencies.
– The production agency performs these actions: to control whether it is possible to produce the good relative to the order and to send the order to the shop floor; to receive information from the shop floor about the state of the production; to send a message to the sales agency when the product is ready; to update the RDBMS. If LIDO-BIKE is not able to produce the bike specified in the order, to send a message to the purchasing agency.
– The purchase agency is supposed to apply a simplified Just-in-Time management (Groenvelt, 1993) for the inventory reduction. This management is performed in two ways. The first way receiving a message from production agency when it is not possible to satisfy the order. The second way checking whether the inventory is below a fixed threshold. In both cases, a related e-mail to the supplier is sent.

The workflow description of these functionalities could be summarized in figure 3:

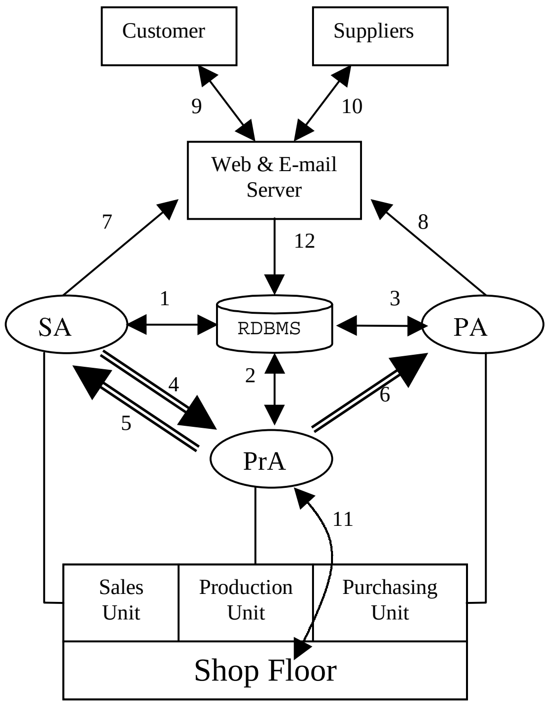

Fig. 3. A Simple integration of a manufacturing information system adding a new layer of agents

Where SA means Sales Agency, PrA means Production Agency, and PA means Purchasing Agency, and the numbers mean, respectively:

1) Querying the RDBMS to check if a new order is arrived; querying the RDBMS to check if the order it is already in the warehouse; updating RDBMS;
2) Checking if it is possible to produce the order; updating RDBMS;
3) Checking if the inventory is under a fixed threshold;
4) XML message sent by sales agency to production agency when the order is not already in the warehouse:

```
<Order_Production>
        <Order_Code>
        </Order_Code>
</Order_Production>
```

5) XML message sent by production agency to sales agency about delivery time:

```
<Delivery_Mail>
        <Customer>
                <Name></Name>
                <Surname></Surname>
                <E-mail_Address></E-mail_Address>
        </Customer>
        <Product_Code></Product_Code>
        <Quantity></Quantity>
        <Delivery_Time></Delivery_Time>
</Delivery_Mail>
```

6) XML message sent by production agency to purchasing agency when it is not possible to produce the order:

```
<Order_To_Supplier>
        <Order_Component></Order_Component>
        <Quantity></Quantity>
</Order_To_Supplier>
```

7) E-mail customer;
8) E-mail supplier;
9) Information from customers; e-mail customer;
10) E-mail supplier; e-mail from suppliers
11) Information to shop floor; information from shop floor;
12) Information from web and e-mail server, such new orders from customer, a new registration customer, or a new delivery from suppliers.

These activities are autonomously implemented by a simple architecture, which can be distributed geographically on the LIDO-BIKE PC's. However, LIDO-BIKE clerks can always say the last word in any critical decision.

To model the LIDO-BIKE agent knowledge, some variables such as the agent name, the agent address, the active directory address (the address of the primary domain controller in the domain), the input queue and the output queue, the name of the clips file are defined.

5. Conclusions

The application of agents to workflow management system is the natural solution to the problem of the definition of additional features of a management information system. Repetitive, heavy, and multialternative tasks deriving from new business processes such as customer relationship management in a e-commerce process, can be effectively implemented, linked and added on existing workflow management system of an enterprise.

The system has already been tested with success with enterprise managers, and one of the most appreciated characteristics was the knowledge modeling process, which allows the involvement of enterprise managers in the workflow definition.

It is also the authors opinion that the formalization approach can be very interesting and powerful in order to optimize the performance of a single agent and the performance of the multi-agent system.

Acknowledgments

This work has been carried out in collaboration with experts from Siemens Orsi Automazione, Genova (Italy) (http://www.orsiweb.com) and CAP S.p.A., Genova (Italy) (http://www.cap.it), working on the development of innovative software systems for workflow management.

References

[1] T. Schal, "Workflow Management Systems for Process Organization", *Lecture Notes in Computer Science*, Springer-Verlag Berlin Heidelberg New York, Vol. 1096, 1996.

[2] Workflow Management Coalition, "The Workflow Reference Model", Available at http://www.wfmc.org, 1995.

[3] M. Aparicio IV, "Internet-Scale Network Intelligence" *IEEE Internet Computing*, IEEE Computer Society, Vol. 3 No. 5, September/October 1999; pp. 38:40.

[4] J.W. Shepherdson, S.G. Thompson, B.R. Odgers, "Decentralised Workflows and Software Agents" *BT Technical Journal*, Vol. 17 No. 4, October 1999.

[5] P.D. O'Brien, W.E. Wiegand, "Agent Based process management: applying intelligent agents to workflow", *The knowledge Engineering Review*, Vol 13:2, September, pp. 1-14.

[6] D. Judge, B. Odgers, J. Shepherdson, Z. Cui, "Agent enhanced workflow", *BT Technical Journal*, Vol. 16, July 1998.

[7] M. Wooldridge, "Intelligent Agents", *Multiagent Systems*, G. Weiss editor The MIT Press, April 1999.

[8] S. Sandip, G. Weiss, "Learning in Multiagent Systems" *Multiagent Systems*, G. Weiss editor The MIT Press, April 1999.

[9] G. Riley, "Clips: A Tool for Building Expert Systems" Available at http://www.ghg.net/clips/CLIPS.html 1999.

[10] P. Maes, "How to do the right thing" *Connection Science Journal*, Vol. 1 No. 3, 1989.

[11] R. Vetschera, "A multi-criteria agency model with incomplete preference information" *European Journal of Operational Research*, Vol. 126 No. Issue 1, October 2000, pp. 152-165.

[12] D.N.P. Murthy, E. Asgharizadeh, "Optimal decision making in a maintenance service operation" *European Journal of Operational Research*, Vol. 116 Issue 2, July 1999, pp. 259-273.

[13] M. Prokopenko, M. Butler, "Tactical Reasoning in Synthetic Multi-Agent Systems: a Case Study", *Linkoping Electronic Articles in Computer and Information Science*, ISSN 1401-9841, Vol. 4 (1999): N. 039 http://www.ep.liu.se/ea/cis/1999/039.

[14] D. Stefanoiu, M. Ulieru, D. Norrie, "Fuzzy Modeling of Multi-Agent System Behavior. Vagueness Minimization", *World Multiconference on Systemics, Cybernetics and Informatics*, SCI'2000, Orlando Florida USA, July 23-26 2000, Vol III, pp. 118-123