A Design Tool to Develop Agent-Based Workflow Management Systems

Marco Repetto, Massimo Paolucci, and Antonio Boccalatte

Abstract— **This paper describes a methodology to design a workflow management system where a set of intelligent software agents composes an interactive scenario. The Workflow Management Coalition explains how to formalize workflows, using business process definition, and a reference model for a high level abstraction design is asserted: the Workflow Reference Model, an architectural representation of such systems. In recent years, agent and Multi-Agent Systems technologies have been considered a suitable approach to develop intelligent decentralized systems. Effectively, workflow systems need to decentralize decisions and activities to entities, in order to uncouple asynchronous information and to process incoming events in a more efficient way. The presented approach adopts Agent Developer Studio, a tool to develop autonomous agents called MAX (Modular Autonomous eXpert) that allows to model the agents' behaviour and to implement them in an expert system shell. The challenge is making agents suitable for a proposed workflow management system architecture, in which autonomous agents achieve functionalities needed to make a role-based organization model.**

Index Terms—**Autonomous Agents, Multi-Agent Systems, Workflow Systems, Expert Systems, Agent Modeling.**

I. Introduction

WORKFLOW is a critical enabler for achieving enterprise competitiveness with today's hot technologies, as portals, e-business and e-commerce.

A definition for workflow is given from the Workflow Management Coalition (WfMC) [1]: a workflow is the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for actions, according to a set of procedural rules. On the other hand, there are many definitions for business processes, and in one of them Schäl [2] highlights how each process produces an output with value for customers. Value is thus considered a measure of customer satisfaction about the purchased goods and services.

A generic business process is composed by a number of logical steps, each of which is denoted as *activity*. Following the WfMC paradigm [3], an activity can involve manual interaction with a user or workflow participant, or an activity might be executed using machine resources. Workflow technology has a clear application in the domain of automated and highly repeatable business processes. Since work delivery is critical to increase efficiency in a business unit, the coordination of activities is fundamental in workflow management. Consequently, scheduling is one of most analyzed matters, especially in manufacturing workflows because of typical dynamicity of such an environment [4].

Matching the above requirements with a technology capable to treat them, workflow management has become an interesting field of applications for agent technology [5-6-7].

An agent is an autonomous, goal-oriented software process that operates asynchronously, communicating and coordinating with others agents as needed [8]. Ranking definitions about agents are given by Wooldridge, Nwana, and Maes [9-10-11]. Since many actors (humans, machines, resources) interact in a complex system with different goals and strategies, collaboration among agents is a critical issue in workflow requirements analysis. Multi-Agent Systems (MAS) [12] are considered an interesting approach to build up distributed intelligent systems, using the paradigm of distributed artificial intelligence.

In many workflow management applications, like in the Supply Chain Management (SCM), an agent should not autonomously take a locally optimal decision, but it should determine the effects of such a decision on the other agents, coordinating with them in order to choose and achieve an optimal solution over the entire supply chain [8].

A system that defines, creates and manages the execution of workflows through the use of software and, if required, invokes the use of IT tools and applications, interacting with workflow participants, is called a Workflow Management System (WFMS) [3].

All data representing enterprise business information, as well as the business processes, reside in an Information System (IS). Secure payments, delivery of goods to customer, inventory reduction management, optimization and management of the supply chain, management of customer and supplier services, and automation in

Marco Repetto is with the Department of Communication, Computer and System Sciences (DIST), University of Genova, Via Opera Pia 13, 16145 - Italy (e-mail: marept@ tin.it).

Massimo Paolucci is with the Department of Communication, Computer and System Sciences (DIST), University of Genova, Via Opera Pia 13, 16145 - Italy (e-mail: paolucci@ dist.unige.it).

Antonio Boccalatte is with the Department of Communication, Computer and System Sciences (DIST), University of Genova, Via Opera Pia 13, 16145 - Italy (e-mail: nino@ dist.unige.it).

delivering orders are all activities that need to be represented as workflows. These activities are usually performed by the functions included in the Enterprise Resources Planning (ERP) systems , a class of ISs quite diffused in manufacturing industries.

The actual state-of-the-art of ERP caused insufficient ability of ISs to respond to new requirements of internet/intranet integrated systems (such as, e-commerce and e-business systems) [13]. Consequently, on one hand this has stimulated the development of more and more network oriented systems, as the supply chain management and Customer Relationship Management (CRM) systems, but on the other hand many firms are still working with not integrated subsystems, whereas to increase their ROI (Return On Investment) they should face serious costly data migration processes. “Small medium enterprises” (SME) need short start-up phases and select a simple solution, modeling a limited set of functionalities, achieving flexibility with light migration processes and involving reasonable financial investment (also in time).

The architecture proposed in this paper is characterized by a simple and common software infrastructure, typical of each existing SME Information System. Furthermore, the approach is distributed systems oriented, responding at topical needs of workflow management systems.

In the following section, Workflow Management Coalition paradigm and workflow formalization are both presented. Then, previous research results and agents requirements are introduced, discussing main issues on implemented models, and the proposed agent-based approach is detailed. The work core is a methodology for translating a workflow into a MAS architecture, specifying formalized rules for modeling agents behaviors. The agents so defined can be deployed by Agent Developer Studio (ADS) [14], a tool for building autonomous MAX (Modular Autonomous eXpert) agents with an expert systems oriented approach. The framework with supported WFMS architecture deployment, based on ADS, as well as interaction properties and ontology of the entire system are then described. Finally, a complete example of supply-chain business process is presented, showing all steps from process modeling to MAS implementation through the role-based translation activity.

II. Workflow Management Systems

Modern small medium enterprises are supposed to play a key role in the world economy of the next years. Whereas significant effort has been spent to demonstrate the benefits of the new paradigm on large companies, SME are still missing a proper approach to co-operative manufacturing [15]. The trend concerns with driving the manufacturing enterprises from mass production to mass customization, and Internet is highlighted as the technology supporting this transformation allowing the global communication between customers and manufactures [16]. Recent business applications such as mCommerce (mobile commerce), Customer Relationship Management (CRM), Supply Chain Management (SCM), e-Commerce (electronic commerce) and Application Service Provision (ASP) conduct companies over networks, facing new markets with new profits opportunities, but also unprecedented efficiency levels are required to survive in such a competitive environment. For these reasons, business processes definition and workflow management within and beyond an organization have become really essential. Therefore, WfMC has introduced standards to allow the interconnection of several workflow systems.

All workflow systems are process oriented. A process definition, a representation of what should happen, is created, and it typically contains some sub-processes and, in turn, activities. Business processes (BP) are defined analyzing the processes in the various firm’s areas, also in those involving customers. Schäl [2] gives a definition of business processes: “a business process creates an output, which is value for the customer; this value is expressed by the customer’s satisfaction with the product/service delivered by the supplier. The business process perspective represents the scope and internal logic of the business process to achieve customer satisfaction by identifying customers (internal and external), suppliers, conditions of satisfaction, roles and responsibilities (of process owners and managers) and the total workflow cycle-time (..)”. Consequently, roles and responsibilities identification is a mission critical key-point in making an IS suitable to model enterprise relationships in an effective approach.

In a process, each activity can interact with other activities and in such a case information should be distributed by means of integration technologies. A typical example of such technologies is represented by the ISs based on Relational Database Management System (RDBMS) integrated with a WFMS. Business logic and business processes are modeled and executed in the WFMS, requiring components and interfaces definition. WfMC standardized a Workflow Reference Model specifying just interfaces and functional components but without giving any definition of them.

Consequent issues are the representation of the business processes in a form that supports automated manipulation, such as modeling, or enactment by a WFMS. Business process analysis and design is mainly a necessary phase to get a workflow model implementation because of the entire business modeling.

A Reference Model core component is the Process Definition Tool, defined as a software application that allows the business process modeling activity. A further component is the Workflow Enactment Service, defined as a set of one or more Workflow Engine(s) which create, manage and execute workflow instances. Invoked Applications and Workflow Client Applications are front-end applications that need to access workflow management engine functions to provide integration between workflow and other desktop tasks.

In workflows modeling, object-oriented design languages as Unified Modeling Language (UML) diagrams, and especially Activity Diagrams (AD), are used for business processes representation [17-18]. Nevertheless UML does not represent a standard in workflow modeling, but typically proprietary modeling languages are used and, in general, others diagrams are shown suitable for processes representation.

In our approach we use the ADs for workflow modeling at high abstraction level, whereas at low level we have developed a custom modeling and design environment that better fits the typical workflow activities with agents requirements described in this paper.

III. Design Issues for a MultiAgent-Based Approach

A. Agents and Workflow Management

The proposed methodology aims at matching the main requirements that a modern Manufacturing Information System (MIS) should satisfy: enterprise integration, distributed organization, interoperability, cooperation, human integration with software and hardware, agility, scalability and fault tolerance. Nevertheless ISs integration should satisfy the social needs of user groups coordination in formal organizations [2], respecting behavioural requirements and structural connections of human organizations.

Some limits in workflow systems have been evidenced in [19-20]: they highlight an inability to cope with dynamic changes in resource levels and task availability, a limited ability to manage the decomposition and recombination of complex items, and an inability to improve dynamically both the business process and how it is managed.

Autonomous agents represent a promising solution to overcome the above limits in workflow technology, and at the same time to address the requirements of a modern small-medium manufacturing enterprise [19-21]. Especially the systems are moving to be composed by several autonomous agents following the increasing in using distributed systems, and the increasing in integrating more applications [22-23]. Furthermore, others requirements as empowerment of stakeholders (including operators, manufacturing engineers, and managers), require human function as peer elements in the system rather than being run by the system: that is easier to meet if the architecture is pervaded by the principle that individual elements (whether human or computer) are autonomous [24]. Consequently, an autonomous agents architecture better matches above requirements.

From a theoretical point of view, the presented approach represents an agent-enhanced workflow, realized combining a layer of agents with an existing WFMS, rather than an agent-based workflow. In the former the agent layer has the responsibility for the provisioning and compensation phases of business process management, while the existing WFMS handles the process enactment. In the latter, software agents take full responsibility for process provisioning, enactment and compensation. Consequently, purpose of our work is the development of an application software to integrate with existing distributed WFMS based on Relational Database Management Systems.

B. Agents Requirements

Autonomy is a main requirement for agents in the presented methodology because of the customizable behavior the user can design. When a programmer wishes to model a business process he should be enabled to analyze the selected context, mapping organization roles into a given agents architecture and following some well formalized construction rules. This approach is oriented to make a multiagent system integration easy to build up. A methodology for translating business processes into a multiagent architecture should allow at programmers to find the same constraint relationships of the given organization structure. Informative uncertainty can be located in each organizational role of a given organization chart; it cannot be removed since it is intrinsic at the business process, even if the process has been object of a Business Process Reengineering project that should improve it, managing automated procedures.

Because of the nature of existing ISs, the use of Structured Query Language (SQL) language is a well defined requirement for implementing an agent-enhanced workflow. As a matter of fact, with such a design feature, agents does not require any gateway or proxy to interface to relational databases. Autonomous authenticated access at the relevant data of ISs via the network and main office automation functionalities represent a functional requirements set for each agent. In addition, agents must be able to communicate each other by exchanging messages and with humans through appropriate interfaces. All such functionalities are common to the type of agents composing the WFMS, but different roles are assigned to them on the basis of business process decomposition presented in next section.

IV. Methods: from Business Processes to a MAS Architecture

Multiagent architecture deployment consists of three steps, which make up the entire methodology for modeling a workflow. The first step represents the answer to the question “what the agents should do”, the second to the question “how can they do that”, and finally the third step corresponds to the software implementation of previous design steps.

A. Business Processes Modeling

To achieve dramatic improvements in critical contemporary measures of performance, Business Process Reengineering (BPR) represents the fundamental way to rethinking and redesign business processes [25]. BPR activity involves all enterprise resources to redesign existing procedures in the form of business processes. This requires organization specifications which represent enterprise parameters, e.g. context, environment, market, number of employees, levels stratification, interaction models and coordination elements. Not less important is customer satisfaction and the value the customer assigns to products or services: customer value awareness for enterprise is a first goal to achieve for business improving. A lack of value awareness can produce not satisfying results. Such a class of questions can be represented with UML Use Cases diagrams, a view composed by actors and use cases, units of functionality provided by a system which represent real attractiveness for customers. A next step is business process modeling with Activity Diagrams. This phase requires at analysts and engineers to identify all involved resources and activities, drawing an easy to understand graphical schema, even for stakeholders. This is a critical issue since business model correctness and efficiency depends on customer satisfaction factors modeled by enterprise managers. Typically, a process involves several entities making activities. Activity diagrams allow to represent virtual boundaries among entities to divide activities off performed from each of them. Swimlanes create several partitions in an activity diagram, and data flow can follow a path across all the swimlanes. This feature well shows a character of each activity, representing a key point for next agent modeling phase. Nevertheless, many of customized formalizations for activity diagrams [18] can provide further support for workflow modeling, enabling a better activity decomposition and a more careful graphical representation.

B. Translating business processes to a MAS Architecture

The required decomposition of business processes is oriented to identify functional areas in which the same enterprise functionalities might be grouped. Working with this organizational approach leads to assign single functional roles to agents, e.g. logistics, warehouse, purchasing, finance, etc. A functional area is a responsibility center with equal discretionary powers, where activities involve a limited resources set and relationships with others areas are acknowledged. In this way, tasks for each role can be defined. Moreover, information managed by a single agent is attributable to a single area instead of many, thus satisfying a locality criteria required for a distributed multiagent architecture. That supposed scenario might lead to violate constraints in business organizations since agents behaviors would present some conflicts without pursuing well-defined goals. For example, an opposing or cooperating role is related to agents goals; consequently, a correct business logic decomposition is needed to avoid an inconsistent configuration in the architecture. A formal translation process of a BP can be defined by decomposing an enterprise organization chart in functional areas, and assigning them to one or more autonomous agents. Load balancing can be achieved by using several agents for carrying out activities assigned to same area.

According to a single enterprise organizational structure, an area can be matched with departments, business units, business centers, etc. After functional roles have been identified in translating a business process into agents with opportunely defined behaviors, the following important translation rule is applied: in activity diagrams, parallel paths departing from fork items and ending in join items should be decomposed into different agents' behavior. Consequently, parallel activities execution is converted by associating one agent with each path. The resulting model may require the synchronization among agents; this feature can be achieved imposing a coordination mechanism among agents, for example, by defining a suitable messages exchange protocol, or making the agents access some shared data of the database in order to verify the state of the workflow execution..

C. Agent Behavior Definition

To develop MAX agents a knowledge modeling tool developed by the authors, Agent Developer Studio (ADS) [13], can be used. The tool provides a easy-to-make way to model agent behavior with a activity diagram like graphic approach. In ADS focus is on agents behavior, developing a lower abstraction level than in activity diagrams, because of the involved agent functionalities set (i.e. SQL queries, desktop applications, communication messages, etc.). For these reasons, activities in modeling business processes using UML diagrams are decomposed into more detailed actions, mapping one-to-one actions with graphical blocks schema in ADS. MAX agents interact with RDBMS and their knowledge is modeled in order to operate on tables, to access to database and to evaluate stored data and decisions taken according to their knowledge-model. Furthermore, behavior is made up of blocks that are automatically translated into CLIPS rules: these constitute the knowledge-base of the MAX agents. Then, an expert system with a behavior is implemented and can be executed in a single execution process, without forcing a user to write CLIPS code.

Previous described requirements for autonomous agents are satisfied by MAX (Modular Autonomous eXpert) agents, which are rule-based agents implemented as single expert system CLIPS (C Language Integrated Production System) [26] shells. They are autonomous agents enabled to aggregate in functional or hierarchical structures, but being anyhow able to achieve their goals in a independent way with self-computing approach.

Agents' behavior lacks of parallel computation and, as described above, a multiagent definition is used as solution for achieving parallel computation in workflows. Nevertheless, in CLIPS each rule is fired from several data, which match constraints in rule patterns, giving rise to overlapped execution of rules with different inputs.

V. The Framework

Agent-enhanced workflows need to interface existing ISs for processing enterprise information and automating activities. For these purposes, the implemented MAX agents operate interacting with RDBMS and carrying out SQL queries. Databases represent a source for agent to update knowledge and a mean to distribute information among enterprise entities. In a MAX agents architecture, the ontology is mainly defined by programmer while he models agents behavior. Blocks programming is done by means of data constraints description, referencing to database tables in order to logically connect names in the IS with meaning the agents manage. Moreover, each agent binds tables and fields with the same meaning, specifying relationship among them on the basis of same fields names. As a matter of fact, data stored in the agent knowledge-base are described from *deftemplate* constructs in CLIPS language, a kind of record types which may contain structured information. *Deftemplate* fields are labeled and managed on the basis of tables and fields names in the database. Furthermore, each data constrain modeled in the agent behavior contributes to stop or forward data in execution flow, enabling data processing and agent interpretation of information.

MAX agents are able to exchange asynchronous messages by means of Microsoft Message Queue (MSMQ). Two special available blocks set may be used in behavior modeling to define run-time messages exchange among agents. During their execution, the agents then connect to databases and send or receive messages. Messages ontology is defined by the programmer who models the behavior, deciding how the agent will manage received structured information. Recent work efforts have been spent in supporting XML language and communicating with structured messages as FIPA speech acts.

MAX agents are thought to be always alive to execute their tasks and to achieve their goals. This approach expects a fixed roles configuration in enterprise workflow, enabling a functional collection of autonomous agents. Consequently, no interaction control is needed in the presented architecture, making the agents life-cycle a customizable scenario: in particular, each firm in the supply-chain can design and implement their own agents (e.g. order acquisition agent, logistics agent, broker agent, etc.), afterwards it can make them communicate over the network and participate at an existing workflow. In such a scenario, collaboration and information exchange among enterprises are needed, as well as an agreement for every interaction form existing among the firms.

MAX multiagent architecture aims at taking advantage from distributed information systems, implementing a multiagent system where all the agents may run on a single computer as well as a computers network like Internet. Making distributed systems ability a successful requirement for agents leads to the concept of virtual enterprises [27], in which users and resources collaborate enabling knowledge exchange. Virtual enterprise is defined as the integration of suppliers and customers with the firm; such a concept needs an high structured communication infrastructure, supporting a free from geographical constraints collaboration framework. Trivial ISs show a lack of previous requirements, slowing down horizontal integration.

In our approach, the Workflow Reference Model is implemented proposing a framework WfMC compliant. CLIPS inferential engine represents the Workflow Engine in the model, therefore each agent is provided of a workflow engine, making an agents-based distributed workflow enactment service (Fig. V-1). Agent Developer Studio takes place in the architecture as Process Definition Tool, allowing business processes modeling and definition by mean of a GUI (Graphical User Interface), even if ADS is used just for the second of two graphical representation steps, after activity diagrams workflow modeling. In our approach, UML diagrams cannot be automatically translated since activity diagrams need to be decomposed for matching groups of activities with roles and for solving parallel paths in as many agents. Nevertheless, adoption of SQL language for composing database queries, on which MAX agents are strongly based, lead to detail much the agents behavior.

In the WfMC reference model, Invoked Applications are external software applications executed by passing workflow data managed from Workflow Engines. In MAX architecture, an example of such applications is represented by supported office automation tools like mail, spreadsheets, etc.

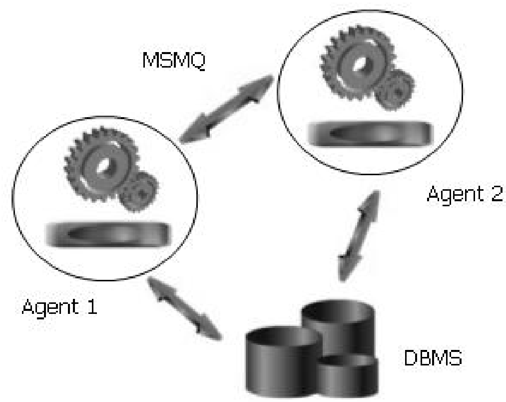

Fig. V-1. General multiagent system architecture

At present no human interaction is allowed during execution phase since human employees usually work with Workflow Client Applications which process data in WFMS, achieving required integration between computers and humans.

In common Microsoft Windows-based environment, agents visibility is guaranteed by Active Directory registration, so that the agents can find and communicate with other agents in the network by applying supported directory services as a yellow-pages implemented service.

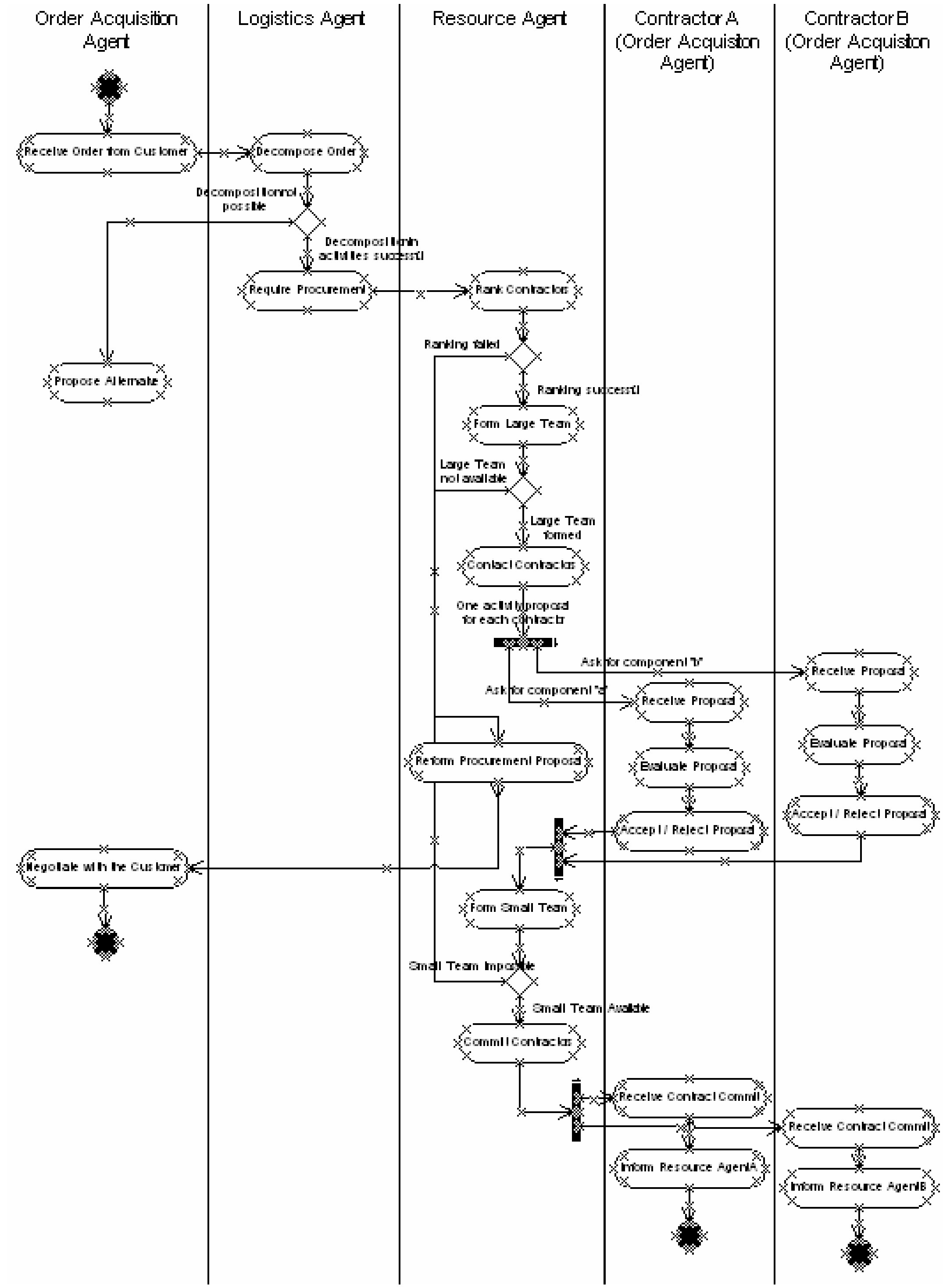

Fig. VI-1. UML Activity Diagram for a Business Process in Supply Chain Management. Swimlanes activities are mapped with respective agents roles.

VI. Supply Chain Management: a Practical Example

A simple example will be used to cover all the aspects of the process, starting from the business process modeling, agents' schema definition and produced files which contain automatically generated CLIPS code, representing agents implementation. Following analyzed business process is extracted from a supply-chain example presented in Fox et al. [8]. They show a virtual enterprise approach to design coordination structures to handle the dynamic formation and operation of teams. A subset of such a conversational plan has been analyzed and in Fig. VI-1 an UML activity diagram of a business process is drawn. In the diagram swimlanes describe different roles and to highlight agents roles each swimlane label shows an agent's name. In this example translation from activity diagram to agents roles appears very easy, because of structured activities decomposition. Nevertheless, efforts to identify single roles in an activity diagram are limited to analyze the organization and its structure, typically result of BPR activity.

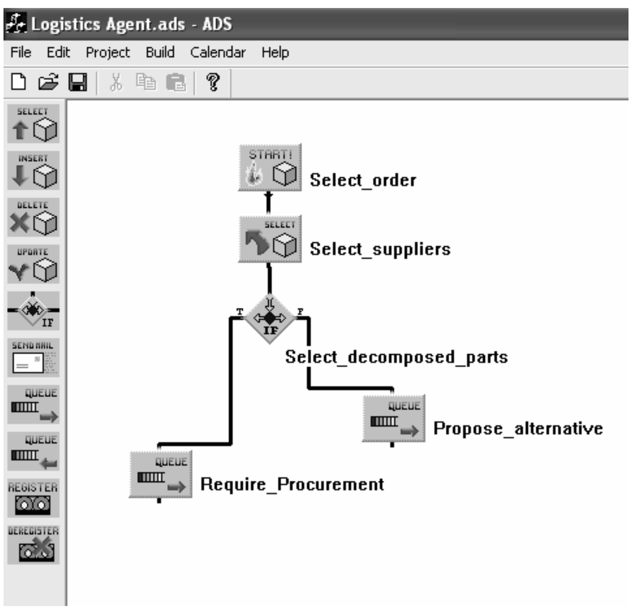

Fig. VI-2. Logistics agent behavior defined by a blocks diagram in Agent Developer Studio.

Next step is represented by agents' behavior definition. For each agent of the activity diagram a rule-based behavior can be modeled drawing it by means of Agent Developer Studio graphical interface; dragging selected blocks from the blocks toolbar is possible to make a blocks schema suitable for representing the related activities. Ordered sequence of activities is followed connecting each block with others by wire. Behavior modeling may need a decomposition of some activities to many elementary blocks, because of different abstraction contexts. For example in Fig. VI-2, Logistics agent blocks diagram presents a set of blocks where first two blocks perform the first activity "Decompose order" of Logistics agent business process in Fig. VI-1.

Similar issues are shown in Fig. VI-3, where the two IF-blocks, connected by wires in a cascade way, work for delivering the exact type of message, on the basis of the defined ontology. In this example there are three types of messages and Contractor agent looks for "Receive Proposal" message.

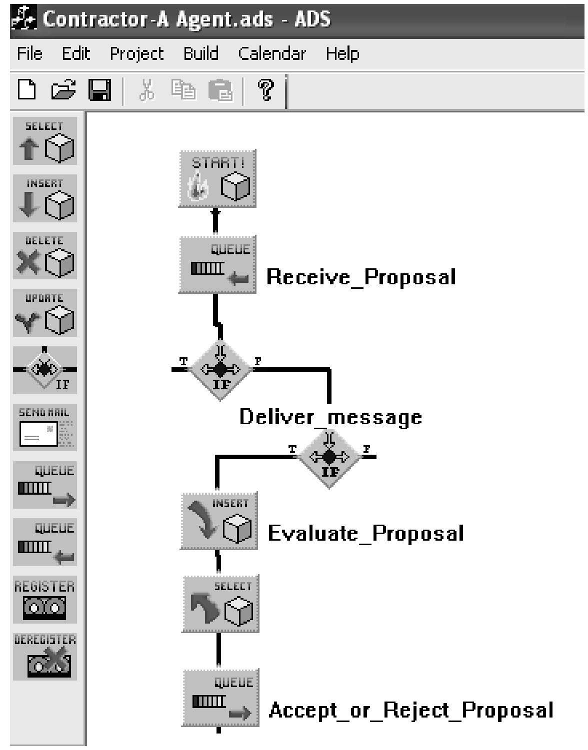

Fig. VI-3. Contractor A agent behavior.

Furthermore, "Evaluate Proposal" activity is modeled with two queries, an *Insert* and a *Select*, in order to update IS data and invoking a client application, not described in this example, that performs a given evaluation of proposal.

After blocks diagrams drawing phase, automatic translation from behavior schema to CLIPS code can be started. ADS allows to automatically check schema correctness and finally it produces the code. Fig. VI-4 shows a fragment of file containing generated rules and CLIPS *deftemplates*, which represent data in agent execution. Such records embody ontology used by agents in produced architecture.

```
(deftemplate INCOMINGORDERS
     (slot status (type SYMBOL) )
     (slot ORDERCODE (type INTEGER) )
     (slot PRODUCTCODE (type INTEGER) )
     (slot QUANTITY (type INTEGER) )
     (slot CUSTOMERCODE (type INTEGER) )
)

(deftemplate STORES
     (slot status (type SYMBOL) )
     (slot PRODUCTCODE (type INTEGER) )
     (slot QUANTITY (type INTEGER) )
)

; Agent Developer Studio  - Project File : NetShop - rule module

(defrule RULE-1
=>
(execQuery  "SELECT * FROM INCOMINGORDERS" "Provider=Microsoft.Jet.OLEDB.4.0;
Data Source=M:\NetShop.mdb;" "INCOMINGORDERS"  "(status 'HOOK-1') ")
)
```

Fig. VI-4. Agent's knowledge is described by rules and deftemplates in CLIPS generated code.

VII. Conclusions

Several agents architectures and developed framework present a hierarchical approach [28], especially in workflow management and in manufacturing information systems. In this paper a different approach to model a WFMS as a multiagent system is presented. Anyway, main focus is on the methodology for translating a workflow, that is the automation of a business process, into a suitable multiagent system. Main features of such a desirable architecture are a compatibility to common software architectures e.g. Windows platform and a high level system distribution as Internet networks.

Future improvements are oriented to better support business process modeling, enabling UML formalization to become a MAX agents behavior modeling tool.

An important feature of presented methodology is highlighted in workflow management applications: system scalability represents a critical issue to take advantage in market to exceed competitors. Autonomous agents and MAX agents may be a key factor to enable such a property in MAS architecture development and deployment.

References

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

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

[3] Layna Fischer, *Workflow handbook 2001*. Future Strategies Inc., Book Division, Lighthouse Point, Florida, 2001.

[4] Weiming Shen, Douglas H. Norrie, “An Agent-Based Approach for Dynamic Manufacturing Scheduling,” *Proceedings of the 3rd International Conference on the Practical Applications of Agents and Multi-Agent Systems, PAAM-98*, 1998.

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

[6] Ernesto Montaldo, Roberto Sacile, and Antonio Boccalatte, “Rule-Based Agents for Workflow Applications in Manufacturing Information Systems,” *WOA 2001, Dagli oggetti agli agenti – Tendenze evolutive dei sistemi software*, Modena, 4-5 Settembre, 2001.

[7] Ernesto Montaldo, Roberto Sacile, and Antonio Boccalatte, “A Modeling Tool to Design Multi-Agent Systems for Decision Support in Manufacturing Workflows,” *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics, SCI2002*, Orlando, Florida, USA, July 14-18, 2002.

[8] Mark S. Fox, Mihai Barbuceanu, and Rune Teigen, “Agent-Oriented Supply-Chain Management,” *The International Journal of Flexible Manufacturing Systems*, Vol.12, Kluwer Academic Publishers, Boston, 2000.

[9] M. Wooldridge, Multiagent Systems: a modern approach to distributed artificial intelligence, Cambridge: The MIT Press, 1999.

[10] H. S. Nwana, “Software Agents: an overview,” *Knowledge Engineering Review*, Vol.11, No.3, 1996.

[11] P. Maes, “Intelligent Software,” *Scientific American*, Vol.273, No.3, 1995.

[12] G. Weiss, *Multiagent Systems: a modern approach to distributed artificial intelligence*, Cambridge: The MIT Press, 1999.

[13] C. Koch, “The ABCs of ERP,” *Enterprise Resource Planning Research Center*, (2002), available at http://www.cio.com/research/erp/index.html

[14] Marco Repetto, Christian Vecchiola, and Antonio Boccalatte, “A Knowledge Modeling Tool for Rule-Based Agents,” *WOA 2001, Dagli oggetti agli agenti – Tendenze evolutive dei sistemi software*, Modena, 4-5 Settembre, 2001.

[15] P. D. Ktenidis, A. T. Paraskevopoulos, “Information and communication technology (ICT) infrastructure for small and medium enterprises (SMEs) networks: achievements and work-in-progress from three European projects,” *Proceedings of the Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises*, Los Alamitos, CA, USA, 1999, 44-51.

[16] A. D. Baker, H. van Dyke Parunak, K. Erol, “Agents and the Internet: Infrastructure for Mass Customization,” *IEEE Internet Computing*. 3(5), 1999, 62-69.

[17] R. Eshuis, R. Wieringa, “A Formal Semantics for UML Activity Diagrams - Formalising Workflow Models,” *CTIT Technical Report 01-04*, University of Twente, The Netherlands, 2001.

[18] Guido Wirtz, Mathias Weske, and Holger Giese, “Extending UML with Workflow Modeling Capabilities,” *Proceedings of the Cooperative Information Systems, 7th International Conference, CoopIS 2000*, Eilat, Israel, September 6-8, 2000.

[19] D. W. Judge, B.R. Odgers, J.W. Shepherdson, Z. Cui, “Agent Enhanced Workflow,” *BT Technical Journal*., 16(3), 1998, 79-85.

[20] P.D. O’Brien, W.E. Wiegand, “Agent Based Process Management: Applying Intelligent Agents to Workflow,” *The Knowledge Engineering Review*. 13(2), 1998.

[21] N. R. Jennings, P. Faratin, T.J. Norman, P. O'Brien, M.E. Wiegand, C. Voudouris, J.L. Alty, T. Miah, E.H. Mamdani, “ADEPT: Managing Business Processes using Intelligent Agents,” *Proceedings of BCS Expert Systems 96 Conference*, Cambridge, UK, 1996, 5-23.

[22] W. Shen, D. H. Norrie, “Agent Based System for Intelligent Manufacturing: A State of the Art Survey,” *Knowledge and Information Systems, an International Journal*. 1(2), 1999, 129:156.

[23] V. R. Lesser, “Cooperative Multiagent Systems: A Personal View of the State of the Art,” *IEEE Transactions on Knowledge and Data Engineering*. 10(1), 1999, 133-142.

[24] H. van Dyke Parunak, A. D. Baker, S. J. Clark, “The AARIA Agent Architecture: From Manufacturing Requirements to Agent-Based System Design,” *Proceedings of Workshop on Agent-Based Manufacturing, ICAA’98, Minneapolis, MN*, 1998.

[25] M. Hammer, J. Champy, *Reengineering the Corporation: a manifesto for business revolution*, Harper, New York. 1993.

[26] G. Riley, “CLIPS: A Tool for Building Expert Systems”, Available at http://www.ghg.net/clips/, 1999.

[27] C. M. Pancerella, N. M. Berry, “Adding Intelligent Agents to Existing EI Frameworks,” *IEEE Internet Computing*, 3(5), 1999, 60-61.

[28] H. V. D. Parunak, A. D. Baker, and S. J. Clark, “The AARIA Agent Architecture: An Example of Requirements-Driven Agent-Based System Design,” *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, CA, 1997.