Intelligent Decision Technologies 5 (2011) 273–293
DOI 10.3233/IDT-2011-0111
IOS Press

An agent-based simulation approach in an IDSS for evaluating performance in flow-shop manufacturing system

Noria Taghezout[a,b,*] and Pascale Zaraté[b]
[a]*Oran University, Oran, Algeria*
[b]*Toulouse University, IRIT, France*

Abstract. The main issue that is addressed in the current paper is a framework based on an efficient coordination for product design through evaluation, planning, and real-time monitoring by calculating the cost at the different stages. The proposed approach integrates agents into an Intelligent Decision Support System (IDSS) and presents a new type of multi-agent-based coordination engine in which the information exchange between agents is controlled via an efficient algorithm. The proposed system mainly includes six components: Resource Agent (RA), Planning Agent (PA), Performance Evaluation Agent (PEA), Database Management Agent (DMA), Rules and Criteria Selection Agent (RCSA) and User Interface Agent (UIA). The multi-agent simulation is used to allow agents to cooperate using an intelligent behavior, and to coordinate their goals and action plans in order to solve a problem. We use UVA methodology to calculate the production costs. This method provides the enterprise with new information on its performances, the profitability of its customers, markets, product, which will generate decisions in all business functions for a permanent progress. One objective of this work is to demonstrate that the UVA method is not only an accounting method of repartition but is also built on the choice of one measuring unit and one specific analysis approach. Experimental results demonstrate that the proposed IDSS could implement effective production control decision-making for solving the flow-shop manufacturing system. The study reports the basic design principles of the system as well as details of the application.

Keywords: Intelligent Decision Support System (IDSS), Multi-Agent System (MAS), "Unité Valeur Ajoutée" (UVA) method, performance, Agent-Based Simulation (ABS)

1. Introduction

Manufacturing systems are dynamic, non-linear and often chaotic environments, subjected to the occurrence of unexpected disturbances that leads to deviations from the initial plans and usually degrades the performance of the system. The treatment of exceptions and disturbances is one major requirement to the next generation of intelligent manufacturing control systems that should be capable to treat emergency as a normal situation [11].

Assessing the performance of a system is a difficult problem that requires one to take into account various components (human, organizational, technical) involved in a differentiated way to its overall performance. With the increasing complexity of production systems and the importance given to their ability to correctly and continuously run, the need to accurately model their functional and dysfunctional behavior and then assess their overall performance, become increasingly urgent.

In other words, flexibility, cost, quality, and technology are considered as the strategic core areas of the enterprise [1]. Flexibility is defined here as the ability of the manufacturing system to cope with changes such as product, process, load, and machine breakdown.

The cost function is primarily related to the duration of production. This latter can be influenced by any perturbation or disturbance which occurs in the production

*Corresponding author. E-mail: taghezoutnour@yahoo.fr.

ISSN 1872-4981/11/$27.50

system. In order to reach this objective simulation is a tool aiding the decision that helps us to find solutions to disturbances and finally reduce the production time.

Basically, product cost systems measure a product's cost by integrating the price of a product's inputs and the technological coefficients of the production processes used in its manufacturing. Traditional cost systems estimate the cost of resources used to create a product by tracing direct material and labor to products and allocating overhead based on one or more volume-based cost drivers. However, traditional cost systems have been criticized for their emphasis on meeting the requirements of financial reporting to the detriment of their usefulness for resource allocation decisions [10].

Our review and analysis of past research indicates that effective decision-making often relies on timely information monitoring, which aims to capture the newest status of the critical information items that are updated in different ways. The review also indicates that properly designed and implemented agent-based control systems result in a performance that is flexible, robust, adaptive and fully tolerant, which are key factors for manufacturing success in the increasingly global market place [11].

In other words, the multi-agent systems are very interesting for the following reasons:

1. Decisional process modeling: The cognitive and interaction capabilities of the agents permit to model and simulate the decisional processes of an industrial system [12];
2. Simulation model distribution: Because the MAS are naturally distributed, the agents can support the distribution of the simulation models [9,19];
3. Modular modeling: The autonomy of the agents allows to our methodology to be more modular and to support a component modeling approach.

This research investigates the use of agent technology in evaluating the global performance of a production system. The main motivation of the paper is to put into evidence how an efficient coordination algorithm can be used to support the decision maker in the task giving information about the related cost of an order or a product by using the equivalence based method (UVA). UVA method also, called the equivalence based method is an innovative approach of the economic and strategically piloting of the firm, it is considered as the fruit of successive adaptations of the GP (Georges Perrin) Method. This method has been the subject of a certain number of works over the last few years.

In order to achieve this goal, two components are considered as important:

1. The framework which provides the functional relationship among constituent agents;
2. And the coordination protocol providing a means of structured communication and coordination.

The remainder of this paper is organized as follows: After having defined the concept of product system, Section 2 reviews a number of approaches for which authors have analyzed and modeled such systems. Section 3 surveys Intelligent Decision Support System (IDSS) and examines the potential integration of agent technology into a framework of IDSS. Then Section 4 briefly discusses work related to coordination, with an emphasis on agents. Further, an overview is given of the major components of the proposed model in Section 5. Section 6 outlines the problems under consideration in this research. Also, the functionality of the system is briefly discussed in terms of decision-making process, the cost evaluation methodology (UVA method) and the coordination protocol; In Section 7, the simulation environment used to develop and test the model is discussed, while in Section 8 conclusions and further researches in this area are drawn.

2. Related works

Considering the Resource Allocation Problem (RAP) in a distributed environment, the solution is not generally solved by a single entity, but by many; and the way it is solved is essentially done through the entities assuming responsibility for generating and maintaining a solution over a decomposed part of the problem, i.e. the local problem. Once the solution to the local problem is found there is a need for the entities to check for consistency and resolve any conflicts primarily through coordination.

An appropriate coordination protocol is important to develop for two reasons: Firstly, it helps to establish and conduct communication; and secondly, it can be used similarly to a control law to manage prescribed situations, thereby guaranteeing the pursuit of appropriate actions by the entities (this, of course, has to be balanced with the necessary autonomy of the entity). However, the design and use of control laws can be limited when a system is complex because a complete prediction of every situation is not possible.

Multi-Agent Systems (MAS), where agents work collectively to solve specific problems, provide an effective platform for coordination and cooperation among multiple functional units in an enterprise.

This paper attempts to apply multi-agent technology to the Decision Support System (DSS), and proposes a multi-agent architecture for an Intelligent Decision Support System (IDSS).

The equivalence based method (UVA) is proposed to be combined with an agent coordination so as to make autonomous agents adaptive to changing circumstances and to give rise to efficient global performance.

In our approach, every agent has the capacity of local control. On its own, it makes the decision for the actions it must accomplish according to its local objective. The system performance is not entirely planned; it emerges from the dynamics of real-time interactions among agents. Thus, the system does not necessarily need to switch between planning phase and implementation phase. However, its behavior is a result of concurrent decisions of local agents.

To achieve their tasks, Planning Agents (PA) adapt their coordination strategy to the dynamic of the system. By using our approach, the coordination strategy may achieve more timely identification of important information updates using a controlled program implying more agents.

We were inspired by two main research works. Firstly, in [7] an intelligent decision support system was developed to tackle the addressed production control problem, in which radio frequency identification (RFID) technology based data capture system was presented to collect the real-time production data from the flexible assembly line (FAL) and a production control decision support system (PCDS) model was presented to assist in production control decisions on the FAL. In the PCDS model, the Modified bi-level genetic algorithm (MBiGA) was used to generate the operation assignment to workstations and task proportions of each shared operation being processed in different workstations. This study investigates the production control problem on a FAL so as to meet the desired cycle time of each order and minimize the total idle time of all workstations on the FAL. The mathematical model of the addressed problem was presented and time constant learning curve model was adopted to describe the variable operative efficiencies on the FAL.

A heuristic operation routing rule was also developed to route the shared operation of each product to an appropriate workstation on a real-time basis. This study considers the change of operative efficiency based on the learning curve theory. Since the change of operative efficiency can also be influenced by some other factors such as negligence, re-learning, and status of machine and operator, future research can focus on the effects of these factors on production control decision-making on the FAL and other production systems.

Secondly, Bing-hai et al. addressed in [3] the problem of dynamic scheduling in a Flexible Manufacturing System (FMS) by developing a decision-making process; this latter is described as follows: in order to respond to both external and internal changing objectives and conditions, certain system objective criterion is determined in the first time. Then the most appropriated dispatching rule/heuristic algorithm is selected to generate the dynamic scheduling with regard to the selected system objective criterion. They proposed an agent-based Decision Support System for dynamic scheduling of a flexible manufacturing system. The proposed DSS includes several components such as User Interface Agent, Criteria Selection Agent, Performance Evaluation Agent, Scheduling Decision Selection Agent and FMS database. In their work, an agent is responsible of the criteria and rules selection. This agent is considered as the heart of the agent-based DSS.

The two approaches mentioned here are interesting, each in its proper way, in evaluating the performance of a production. The first deals with learning techniques and measures the overall system performance on the basis of performance estimated on the machines (resources) and operators. While the second approach delegates two agents PEA and SDSA to evaluate the scheduling performance. This performance evaluation is based on the use of rules and heuristics. It, therefore, focuses on the interactions of two only agents.

This state is associated with a simulation run. The results are saved in the history of the simulation for each acceptable scheduling. The approach tries to avoid the system blockage by assuring a normal and permanent functioning owing to the Planning Agent (PA). The equivalence based method (UVA) (production cost calculating method) is proposed to be combined with agent coordination. This is to make autonomous agents adaptive to changing circumstances and to give rise to efficient global performance. Our fundamental hypothesis is as follows: Integrating an agent-based simulation in the IDSS will provide the system with intelligence capabilities that will enable it to respond quickly and effectively to new information without human intervention, and recommend actions to deal with complex situations.

Accordingly, the need here is for a model that reflects the interactions among products with respect to their price, resource usage, and cost within the constraints of available resources and product demand. In flow-shop manufacturing system particularly, the model is simplified because of the design and production of one product. The performance of the flow-shop system is evaluated using various measures. Simulation experiments are carried out for the different scenarios that arise out of the combination of full product cost calculation method and coordination rules.

3. Intelligent Decision Support System (IDSS)

Recently, many improvements have been witnessed in the DSS field, with the inclusion of artificial intelligence techniques and methods, as for example: knowledge bases, fuzzy logic, multi-agent systems, natural language, genetic algorithms, neural networks and so forth. The inclusion of AI technologies in DSS aims to develop computer based systems that mimic human qualities, such as approximate, reasoning, intuition, and just plain common sense. The new common denomination is: Intelligent Decision Support Systems (IDSS) [17] (see Fig. 1).

Intelligent Decision Support Systems are interactive computer-based systems that use data, expert knowledge and models for supporting decision makers in organizations to solve complex, imprecise and ill-structured problems by incorporating artificial intelligence techniques [17].

They draw on ideas from diverse disciplines such decision analysis, artificial intelligence, knowledge-based systems and systems engineering. Using IDSS intends to improve the ability of operators and decision makers to better perform their duties and work together.

There may be different ways to make a DSS more intelligent; the most frequently suggested method is to integrate an Expert System (ES) into a DSS. Turban and Watson [23] suggested two fundamental ES/DSS integration models: (1) ES are integrated into DSS components, the incorporation of ES aims to enhance the function of particular components in a DSS; for example, integrating an ES into the Data Base Management System (DBMS) of a DSS, which adds reasoning capability to data manipulation. This particular integration enables users to perform higher-level queries. According to Turban, the integration of ES in DSS components could be applied independently. (2) ES is integrated as a separate component in the DSS; an ES is an add-on to the original DSS. We argue that an IDSS is able to capture the domain knowledge and provide intelligent guidance during the process. While the data and model manipulations are done through the DSS, decision makers can focus solely on the process issues.

A number of works (cf. [1,16,20]) have also addressed issues associated with IDSS. In particular, in [18], Sevtsenko et al. present a multi-agent based IDSS. The proposed system supports the work of different ERP systems of enterprises as one network. Thus, the computer network-centric multi-agent approach is intended to facilitate interactions between many agents participating in the product and manufacturing process development.

Whereas, in [16] the author proposes a general multi-agent architecture for intelligent decision support system MAIDSS, and a decision process based on this architecture. The architecture is very useful for complex multi-objective, multi-constraints real-world problem because a set of agents can embody different objectives and able to generate different solutions.

The approach described here aims to solve complex problem by dividing agents into several teams, each team takes charge of one sub-problem. Thus, each agent in a team embodies its own algorithm for creating a solution, and one agent may modify the solutions of the other agents by means of negotiating. Humans are also involved as agents in the process of decision-making.

4. Multi-agents and coordination

4.1. Coordination

Coordination is a key research area in engineering design and distributed artificial intelligence, as well as a number of other disciplines. In his work [6] defines some key characteristics of coordination as methodology identified as coherence, communication, task management, resource management, schedule management, and real-time support. From his point of view, coordination enables inter-related tasks to be undertaken whilst utilizing resources, of varying performance capability, in an optimized fashion according to dynamically generated schedules with a distributed design environment. Thus, coordination in real-time ensures that adjustments only occur if appropriate and, if so, periods of resource adjustment and task re-allocation are utilized effectively.

Any necessary information must be managed such that it is made available to allow the task to be completed. Monitoring facilitates the detection of devia-

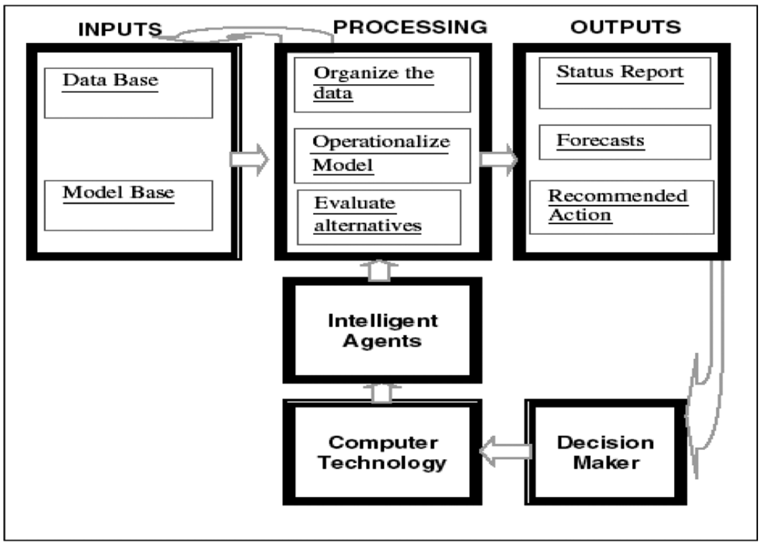


Fig. 1. Components of an IDSS [17].

tions between the actual and expected performance of a resource.

We can say that these characteristics can be used in the development of a methodology, which coupled with agent technology or simply taken in consideration while defining a coordination agent. We have applied this methodology in order to better define the coordination agent's responsibilities. In addition, with regard to the methodology and its realization within the Design Coordination System (DCS) developed by Graham, the coordination protocol defined in our study provides an effective mean of aiding distributed manufacturing flow-shop system.

4.2. Contribution

In this paper, we propose a multi-agent coordination by adding the following features:

1. From our point of view, the decision-making knowledge stored locally in the agents will cause the global behavior of the system to operate in a way that cannot be precisely predicted. For this purpose, five agents are proposed, namely: a Planning Agent (PA), a Performance Evaluation Agent (PEA), a Resource Agent (RA), a Rule and Criteria Selection Agent (RCSA), a Data base Management Agent (DMA) and User Interface Agent (UIA).
2. In our approach, we consider that the performance of the distributed organizational structure is depending on the ability of the agents to coordinate their activities; The system performance is not entirely planned; it emerges from the dynamics of real-time interactions among agents. Thus, the system does not necessarily need to switch between planning phase and implementation phase. However, its behavior is a result of concurrent decisions of local agents. The multi-agent simulation is used to allow agents to cooperate with an intelligent behavior, and to coordinate their goals and action plans to solve a problem. We use UVA method to calculate the production costs. This method provides the enterprise with new information on its performances, the profitability of its customers, markets, product, which will generate decisions in all business functions for a permanent progress.
3. To predict the behavior of the system as a whole, we exploit the fact that in the agent-based system, no central unit exists and represents a critical barrier to the wider application of agent-based ideas [13]. Simulation can help to understand the system behavior. Under condition that all the types of possible behavior will be covered, explored and studied. In our coordination protocol, each agent plays a precise role. As each agent takes part to the user's requests resolution, the processing is distributed. By using our approach, the coordination strategy may achieve more timely identification of important information updates using a controlled program implying more agents.
4. In order to facilitate the interaction among agents, agents are built using one of the common agent

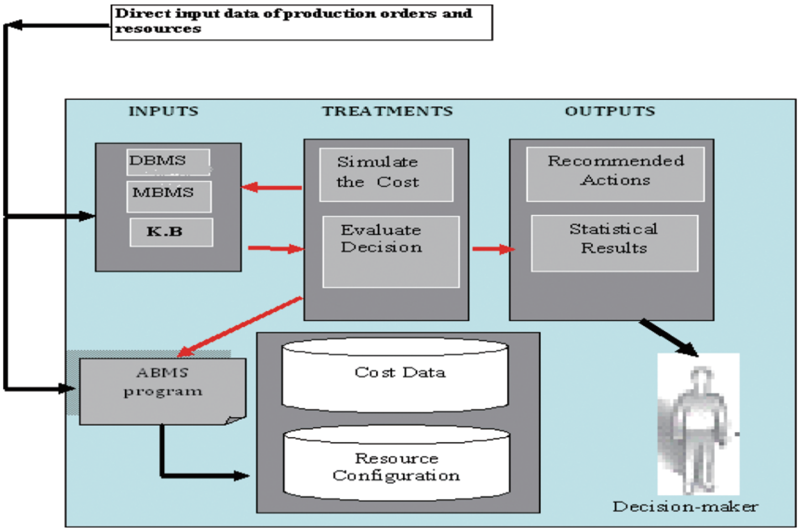


Fig. 2. The proposed model.

development platforms, JADE [8]. This latter also provides the agent with facilities to understand an Agent Communication Language (ACL) that expresses task requests and knowledge representation in a common format. As mentioned in [13], the *agentification process* associated with installing agents into the production management level provides an elegant mechanism for system integration trough using a tool to support the *technology migration* from centralized systems towards the agent-based architectures.

5. Proposed model

Effective decision-making requires the integration of expert knowledge, data, and simulation models to solve practical problems and provide a scientific basis for decision-making. User-friendly decision support tools must necessarily support different stakeholder groups, understand, evaluate and share alternative management strategies. The tools should integrate a suite of components consisting of database management systems, other systems like simulation models, decision models, and user-friendly interfaces that could then be available to different stakeholder groups.

As described in Fig. 2, the most important components of the proposed model are:

The input component comprises:

1. *The Data Base Management System* (*DBMS*) mainly contains a relational database which is managed by the database management system, and which provides speed data retrieval, updating, and appending;
2. *The Model Base Management System* (*MBMS*) includes many statistical, management scientific models, or other quantitative models that offer the system's analytical or forecasting capability to solve future outcomes. Optimization models, such linear programming and dynamic programming, are often adopted to determine the optimal resource allocation to maximize or minimize an objective function;
3. *Knowledge Base* (*KB*) provides the needed scheduling and production information. Some data are related at this level, for example data concerning manufacturing resources, manufacturing orders, process planning, maintenance planning of resources, etc.

The Treatment component represents at the same time the organization and the structure of the decision problem, determining all possible resolutions in order to support the user or the decision-maker choosing the best resolution.

The Output component represents the statistical results as well as recommended actions.

The Forth component contains dynamic data; the most important for our study can be summarized in status data, resource status (normal, breakdown, repairing etc.), data of current executing scheduling, current manufacturing order status, decision variables, and scheduling and dispatching rules.

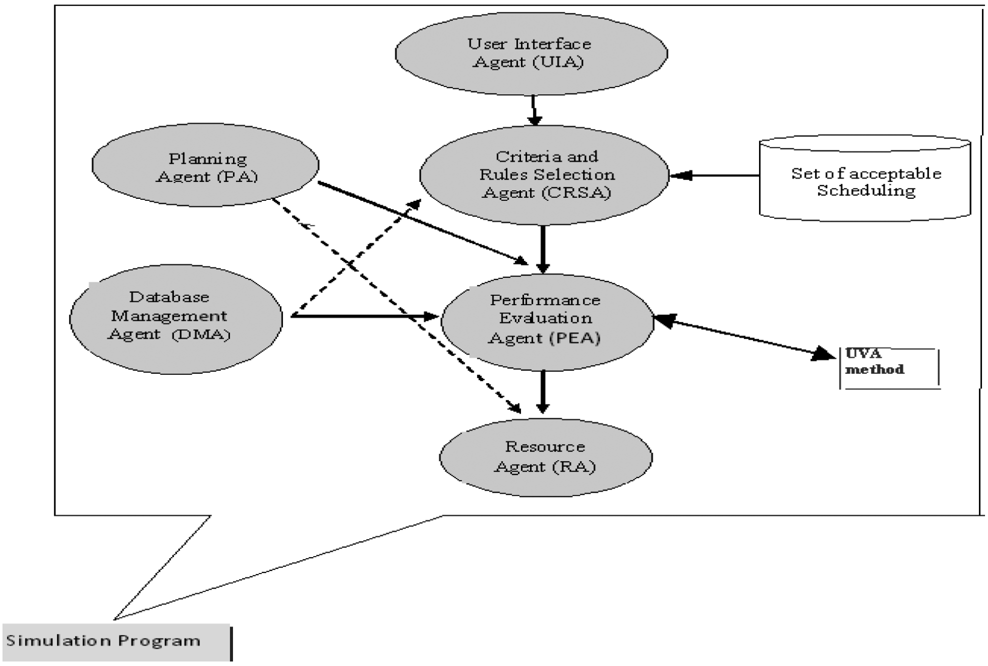


Fig. 3. The manufacturing system built as a MAS.

6. Issues under consideration

In this paper, three issues in our developed model, namely distributed problem solving by using a multi-agent system, cost calculation by using UVA method and an agent collaboration protocol are addressed.

6.1. The Agents architecture

Identifying a problem is an important aspect of designing a MAS, it strongly influences the performance and efficiency of the system to solve a problem.

An agent is defined as an autonomous and communicating entity, as part of our work using a protocol; the agents can then coordinate with other agents for collectively and dynamically covering the performance of a complex task with interrelationships and dependence among its actions [1].

Figure 3 shows an overview of the architecture of the multi-agent system. Each of the six types of agent fulfills a particular role and is capable of performing a number of activities. An outline is now given of the main responsibilities of each agent:

1. *Resource agent* (*RA*) represents enterprise's internal resource such as software, machine and workers. It comprises five functional modules: interface, proposal generator, Data Base, Knowledge base and control. RA will prepare a proposal according to its knowledge (including its capability, schedule, status, and cost) and send it.
2. *Planning agent* (*PA*) performs most of the autonomous problem solving: (1) receives user delegated task specifications from an Interface System, (2) interprets the specifications and extracts problem solving goals, (3) forms plans to satisfy these goals, (4) identifies information seeking sub-goals that are present in its plans, (5) decomposes the plans and coordinates with appropriate sub-systems for plan execution, monitoring, and results composition. Planning Agent has the following knowledge: 1) knowledge for performing the task (e.g. query decomposition, sequencing of task steps), 2) information gathering needs associated with the task model, 3) knowledge about relevant information, modeling, diagnosis, and action components that it must coordinate with in support of its particular task, 4) coordination rules that enable coordination with the other relevant agents.
3. *Performance evaluation agent* (*PEA*) defines the performance indicators which are of two types: indicators for monitoring processes information and indicators for Simulation. The PEA establishes the scheduling performance for each deci-

sion objective criterion under different rules or a same algorithm.

4. *Database management agent* (*DMA*) captures all the data and knowledge from databases and knowledge bases.
5. *Rules and criteria selection agent* (*RCSA*) defines the threshold of performance by establishing rules for evaluating the performance and criteria. The most relevant criteria used in this study, have been described in [21].
6. *User interface agent* (*UIA*) is one of the most important components that determines the success, usefulness and flexibility of the proposed IDSS. Interaction between manager and/or decision maker and the system is accomplished through the UIA.

The Planning Agent (PA) first gets input data through the user interface agent (UI). Next, CRSA searches for rules to select a suitable scheduling program and to execute the model to get analytical results. Additionally, all the parameters values needed by the models are retrieved from the database via the DMA. Resource Agents (RA) monitor their physical resources (machines) entities, and report maintenance or breakdown to the shop. PEA calculates the cost production by using UVA method. PA and RA use the results of the model analysis to identify the fault causes and to perform a suggested action plan.

6.2. Decision-making process

When the scheduling tasks, the system current status and the system priorities are given to the RCSA, the decision-making mechanism is activated for determining the decision objective criteria, and rules for calculating costs and algorithms in the RCSA. Once the selection completed, RCSA will provide possibility to interact with decision maker if manual modifications are necessary.

Moreover, there are a lot of factors affecting the selection of decision criteria and the scheduling rules for the RCSA. The main factors:

- System priorities, such as priorities of manufacturing orders, priorities based on tardiness criteria, priorities based on some combined weighted criteria about resource utilization and preparation.
- System disturbances status.
- Different scheduling tasks.

There are many sources of uncertainty in real-world manufacturing system, which trigger disturbance events in dynamic scheduling. Such disturbances should be explicitly represented in our model. Two types of disturbances are considered in this work: resource-related disturbance, and task-related disturbance.

Resource-related disturbance refers to the disturbance caused by unreliability coming from resources (machines), including machine breakdown and machine recovery.

Task-related disturbance refers to the disturbance caused by the changes in production orders, including new tasks and existing task cancellation.

The MAS described above has been evaluated in several simulation experiments. In the next section, we describe the simulation software, the experiment design, and the results of the experiments.

6.3. UVA method

The “Unité Valeur Ajoutée” (UVA)[1] method has several tools that allow the company to make a strategic choice and defend it. Its main advantage is the fine study of all processes in the organization and knowledge of all consumption of resources they generate. All these efforts are directed towards better management of customer relationships, but also allow appreciating the importance of suppliers to create value in the company [1,4,15].

An innovative approach of the economic and strategically piloting of the firm, UVA method was progressively developed from concept of “the unification of the production measure “stated by George Perrin at the beginning of 1950s.

This approach focuses on the strategic use of cost and may lead to developments based on interesting simulations on the strategically plan [4].

In his work, [15] argued that the UVA method is used to adopt a method for fine decomposition of activities, allowing precise analysis of costs and profitability of sales while saving time. As mentioned in [25] many problems are encountered in other cost methods for example: (i) the lengthy data collection process; (ii) complex updating process necessitating repeat interviews

[1]The equivalence unit or ‘UVA’ method was developed in the 1940s by Georges Perrin, a French engineer. It was not only disseminated in France but also in Brazil under the name UEP (Unitades de Esforco de Producao) or (Unitades de Producao). It was even imported to Great Britain from Brazil.

In order to distribute time over activities; (iii) multiplication of the number of activities and the need of high data processing and storage capacities. There are methods which try to simplify costing, the most important are Activity-Based Costing (ABC) method [10] and UVA method. From our previous research, we can say that these different methods belong to the same trend, which is seeking to clarify the relationship between costs and products/services while also simplifying the accounting process.

The most relevant items listed in Table 1 resume the comparison of ABC or TABC method and UVA method principles. We choose to use UVA methodology to calculate the cost because it is precise in its analysis. The UVA methodology is a management tool as well as a decision support allowing the company to ensure that its processes are customer focused and cost effective manner. This method adopts a simple logic based on equivalences, it analyses the working of the shop. The simplification of costing systems enables a much more detailed analysis of the company activities, without need substantial resources. From our point of view, once the method has been set-up, the indices calculated can be used for many years.

The construction phase begins with the calculation of charges attributable to each item UVA (the rate of posts). The interviews will then be used to establish the process of value-added costs following the position rate. The analysis of products and customers determines the ranking of the latter according to their resource and process consumption. We calculate the result by adding the sale cost of products to that of the customer expressed in UVA. The data will be transcribed from the educational aspect of a curve of return (classification of sales by order of increasing profitability) to classify sales into different categories. The structuring effect given to the organization of the enterprise is achieved through the careful analysis of all activities and the delineation of their contribution to the transverse processes of each elementary business: the sale. So, we can resume the basic principles under this method in the following: determining equivalence coefficients is, as we will see, a relatively long process, but once done, it allows costs to be quickly evaluated in steps whereby, for the reference period: 1) determine the post rate; 2) determine the rate of the reference article; and 3) calculate the post indexes; Furthermore, during each period, to determine the result per invoice: a) evaluate the activity of the period in equivalence units (or UVAs); b) determine the equivalence unit (or UVA) rate; and c) evaluate the invoices.

As mentioned in [25], the UVA method enables graphs (as ABC) to be produced for sales, invoices, etc. These analyses are made monthly and there is a wide range of potential analyses for aiding the decision-making process: invoice, product, customer, market-based analyses.

The realization of simulation is easier and can even be used to make valuable offers or quotes by incorporating the full estimated costs for a sale. This gives business managers the knowledge of price and hence gives them flexibility in negotiating trade.

6.4. *Proposed approach for coordination*

In production systems, the use of multi-agent architectures provides a framework for modeling and simulating resources of the workshop. The multi-agent simulation is used to allow agents to cooperate with an intelligent behavior, and to coordinate their goals and action plans in order to solve a problem [1,21,27].

As discussed in Section 4, coordination is critical in MAS design. Previous studies focused on resolving conflicts among the agents through various techniques such as planning [26], social knowledge encoding, task structuring and negotiation (e.g. [21]). Basically, the coordination strategies fell into two main categories: Centralized coordination and distributed coordination. Within multi-agent system environments, the issue of centralized or distributed control is widely reported [14, 22]. A centralized approach involves a single agent having all of the necessary information to make planning decisions for the entire agent community. It is acknowledged that centralized approaches are theoretically better at pursuing global system performance; however they suffer drawbacks such as reliability and reactiveness [14], and are ineffective in highly dynamic systems.

A decentralized approach involves each agent in the environment having the necessary information to make autonomous decisions, while a common goal is reached through cooperation among the community. However, environments with no central authority are viewed as being complex to manage and, as such, require an appropriate approach to ensure effectiveness and efficiency. Coordination multi-agent systems are viewed as a suitable means of delivering such an approach.

Our earlier work provided the base functionality needed to integrate agents into a DSS for the purpose of automating more tasks for the decision maker, enabling more indirect management, and requiring less direct manipulation of the DSS. In particular, to meet

Table 1
The comparison of ABC method and UVA method [25]

Evaluation criteria	ABC method	UVA method
Information research	Comparisons between the actual and standard costs are possible at all levels: activity, product. . . Comparisons between the consumed work units and the standards ones are also possible.	The method provides different information. It is not possible to have information on actual costs; imputations are made only for the reference period. On the contrary, for every post, it is possible to compare the number of the used UVA to the number of standard UVA. The number of UVA produced by post is another appropriate indicator of an activity.
The allocation of indirect and fixed costs	The ABC method is used to identify the most relevant work units or cost drivers regardless to the activity.	The UVA method allows a more detailed analysis than the ABC method does because the imputation is done only once.
Modeling the costs behavior	With ABC method, the level of analysis is coarser than for the method because there are fewer activities than positions. By contrast, standards can be updated if they correspond no more to reality	From the production lines, it is easy to simulate (new products or reorganization of production).
Understanding the causes of cost	This understanding would be independent of the used method. However, it would be facilitated by detailed knowledge of the cost behavior.	The analysis helps to understand the causes of costs. However, the lack of monitoring of actual consumption represents an obstacle to this understanding. . .
Maintenance and implementation costs	Many ABC models take a long time to become operational as their development time is too lengthy.	Easy to implement and maintain.
Consideration of the under-use of production capacity	ABC model did not enable under-activity to be evaluated: the times necessary for executing an activity are determined under usual operating conditions.	In calculating the UVA cost for each period, the cost of unused capacity is reintroduced.
Precision	Depends on the multiplication of the number of activities.	Depends on the number of analysis units or work posts.

the need of communication and coordination between entities (i.e. agents) in manufacturing environment, a dynamic scheduling algorithm has been developed in which an extended version of Contract Net Protocol (CNP) was adapted to resolve agent's conflicts [21]. The established negotiation model suggested a protocol where agents proposed bids for requests. The bids also included counter proposals and counter requests.

But in that work, the lack of coordination flexibility limits the modelling capability at both the agent level and the global cooperation behaviour. The coordination task mainly ensured the management of multiple negotiation steps and synchronized the various obtained results among resources agents. The coordinator agent was responsible for maintaining data consistency during the process by controlling the flow of messages.

We consider the coordinator more flexible if it can automatically coordinate the agent behaviour according to more complex resource sharing constraints and in various situations including resource-related disturbance, and task-related disturbance.

So, this study proposes a new coordinator (PA) design method that allows for progressively enhancing the flexibility of coordination. Coordination is performed concurrently and continuously in the shop and is classified into various coordination types: breakdown, release, allocate, and sequencing coordination.

The current configuration of the MAS that we developed enhances both production planning, and production control. Furthermore, it attempts to attain coherent global performance by means of local decision-making through various forms of agent coordination. The planning agent has learning features that facilitate decision-making in the system. It provides reasoning capabilities according to Case-Based Reasoning (CBR) paradigm. This aspect will be more developed in our future research paper.

6.4.1. Coordination algorithm

The coordination protocol contains two parts: the agent part (see Fig. 4) and the coordinator part (see Fig. 5). The shop scheduling processes are illustrated by a diagram using AUML in Fig. 4, showing the interactions among agents and coordination steps. The steps in the coordination algorithm are summarized in Fig. 5. The PA dynamically monitors and updates the agent states and RAs priorities, after which it processes the RAs requests in their order of priorities. With the State List in the PA, the system can process multiple requests of the RA simultaneously. We can describe in the following the main steps:

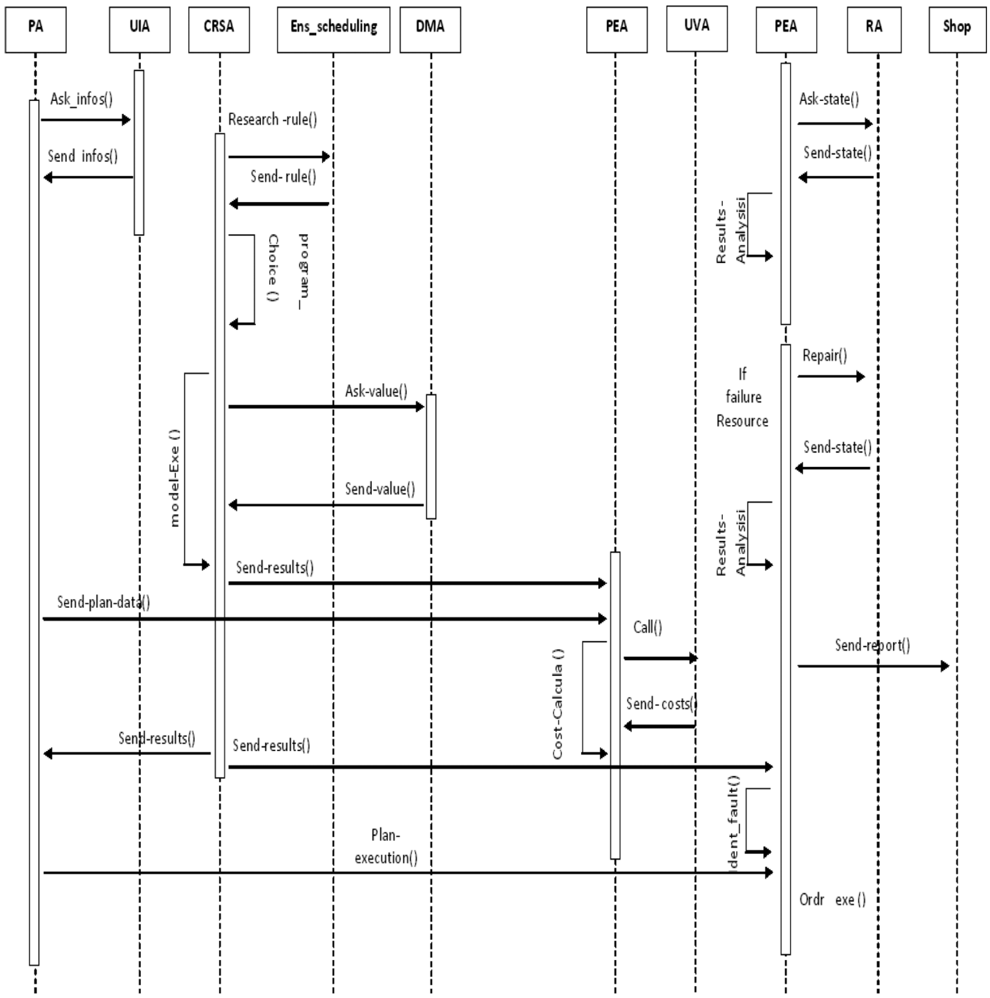


Fig. 4. Agent coordination in the shop: scenario description with AUML.

1. Read the list of resources with break downs,
2. Treat the first request, PA checks the time of the failure of resources.
3. If the times of failures of resources are different, then PA will treat the RA's request whose failure has occurred and applies the diagram shown in Fig. 4.
4. Otherwise if there is equal time of breakdown (presence of conflict), and then PA consults the knowledge base to search corresponding diagnosis:
   4.1. If diagnosis exists in KB of PA then it treats the request of the agent RA and starts again this treatment until the end of the list of the breakdowns of equal time.
   4.2. If diagnosis does not exist in KB then PA passes to the following RA's request.
   4.3. If all requests (with equal time) and (no diagnosis in KB), then PA chooses a random and treat its application by applying the reasoning given in the diagram (Fig. 4).

The first coordination starts from a *release coordination* between resource agents and planning agent. The PA starts to release available operations to the shop by requesting the RA for resources for specific process. By *sequencing coordination*, PA finds the most appropriate operation in the queue to be processed next.

When the machine breaks down or begins maintenance, the *breakdown coordination* starts with a trigger of the breakdown event from the shop, the RA rejects current operations in the queue back to PA, stops all messages flowing in and out from the queue. When the machine resumes work, it reports a machine up-event to shop, resumes the processing of the previous sus-

```
Begin
Var j, i, n : integer ;
Var OK: boolean;
Var Time_max.PA   (Coordinator Agent), H : string;
T [Ri](** Ti:        :ate**), D [Ri] = Array [R1…R2] of String (**Diagnosis state**)
Repeat
     Read (n);
    Read (T [R1]);
    For i := 1 to n do                              (** Resource with the highest priority**)
        Begin
              Read T [Ri];
                          if T [Ri] > Time_max then R1 ← Ri
         End
    Grant the request from the agent responsible of  R1;
    Goto protocol given in  figure4
    N ← n-1;
    For i := 1 to n do
         Begin
              Read T [Ri];
              j:=i+1;
              If T [Ri] = T [Rj]  then
                          Begin
                        If D [Ri]  ∈{H}then OK ← Vrai
                                   CA ← Ri;
                                            Goto protocol given in  figure 4;
                                            n ← n-1;
                                            else
                        if D [Rj]  ∈{H}then OK ← Vrai
                        CA ← Rj;
                        n ← n-1;
                        else
                        if D [Rj] and D [Rj]  ∉{H}then OK ← False
                        CA ← choose a resource randomly;
                        Goto protocol given in  figure 4;
                        n ← n-1;
    End
End
    Until system terminated or (n=1);
End
```

Fig. 5. Coordination protocol for the Planning Agent (PA).

pended operations and activities. The Planning agent receives the production order to execute and allocates resources to different agents all this by monitoring the execution order of the overall tasks of the system (*allocate coordination*).

6.4.2. *The Planning agent* (*PA*)

It is composed of three modules: decision module, communication module and coordination module (see Fig. 6):

Decision module: It contains:

- Historical failures of resources and repair services for each failure.
- Projects: each project has a code and a specific sequence.
- In addition, the agent coordinator has direct access to two tables:
  Resources table and spots table.

Communication module: It consists of two parts:

Sending messages: PA sends messages to:

- DMA.
- Resource agent.
- Maintenance service.

Receiving messages: PA receives messages from:

- Resource agent.
- Maintenance service.
- Confirmation of operations carried out by the DMA.

Coordination module: it is the algorithm which manages actions to be made for every situation.

Clearly, all the modules representing the inner structure of an agent may depend on each other. This is especially true for the decision module and the coordination module which does not only exchange real time information but, in addition, must coordinate their decision rules and performance criteria. If we consider the relationship between the coordination module, the control module, and the knowledge base, we found that they have to make sure the data needed available. The communication between the agents may roughly

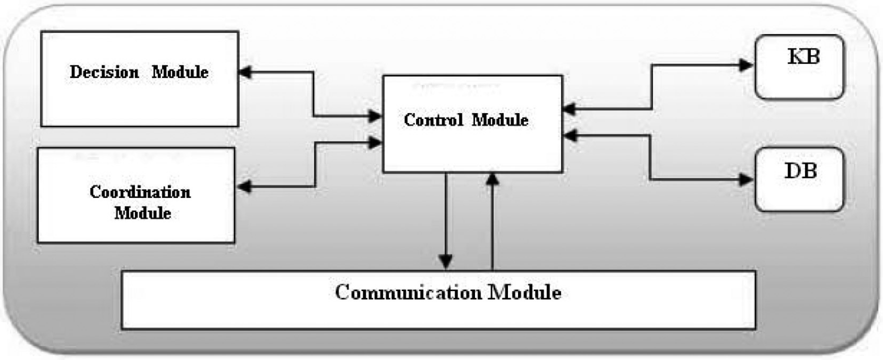


Fig. 6. Planning agent structure.

be described by the coordination module and the communication module. They describe the way that agents may communicate.
The PA is responsible for additional functions such as:

- Record all nonproductive time.
- The resolution of conflicts.
- Define preferences of execution of tasks according to sequence.

The steps outlined in the diagram shown in Fig. 7 summarize the actions taken by PA for the treatment of a resource failure arriving in an unexpected manner.

7. Simulation environment

To support the development of MAS-oriented solutions, simulation tools and platforms require new types of simulation systems and platforms within which MAS principles are embedded.

The MAS as well as the simulation environment was implemented in JADE (Java Agent DEvelopment framework) [2]. This latter also provides the agent with facilities to understand an agent communication language (ACL) that expresses task requests and knowledge representation in a common format. An overview of communication among agents is given in Fig. 8.

Thanks to its distributed nature, JADE is particularly suited to create applications that run on distributed low cost architectures, such as blade, and therefore it perfectly meets the scalability requirement [2,8].

Scalability is certainly the main reason that led to the usage of JADE as the basis for our simulation model, but other reasons were identified and had an impact on the final choice, such as: (i) capability to serve well reactive processes, such as activation and troubleshooting; (ii) capability of executing long and fairly complex tasks such as network element configurations upload; and (iii) capability of automatically exposing agent capabilities as Web Services (as future extensions of our system) [2].

As reported in [5] the design of manufacturing system is an excellent candidate for the application of agent-based technology. In many implementations of multi-agent systems for scheduling and control, the agents model the resources of the plant, the scheduling and control of the tasks is done in a distributed way by means of cooperation and coordination of actions amongst agents. In such systems, the need to find a feasible solution is much greater than to find one that is optimal under the condition that the manufacturing control and execution is a real time application.

In our system, we consider that resource agents (RA) depend on the system description and planning agents (PA) depend on the existing tasks.

So, we tested the simulation program on a simple case study, a company, to produce a special computer installation in terms of hardware and software for a customer. Three kinds of packages have been developed: the plant network, the UVA code and the statistical package. The plant package consists of six agent types: PA, RA, PEA, DMA, CRSA and UIA. Figure 9 presents the software module that we created to execute the UVA program in JAVA. The statistical package collects the data at the end of every simulation run and it generates reports, histograms, graphs and statistical analyses.

Two series of simulation experiments have been made. In the first series of experiments, the system uses the set of agents that are present and available at that time, to give the system a high degree of adaptability to the dynamic nature of the manufacturing aspect.

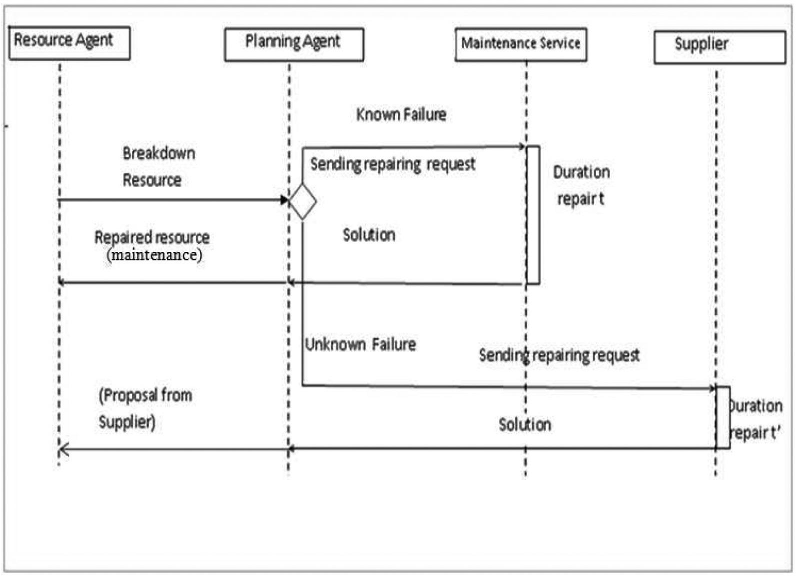


Fig. 7. Diagram describing the arrival of failure event and its management.

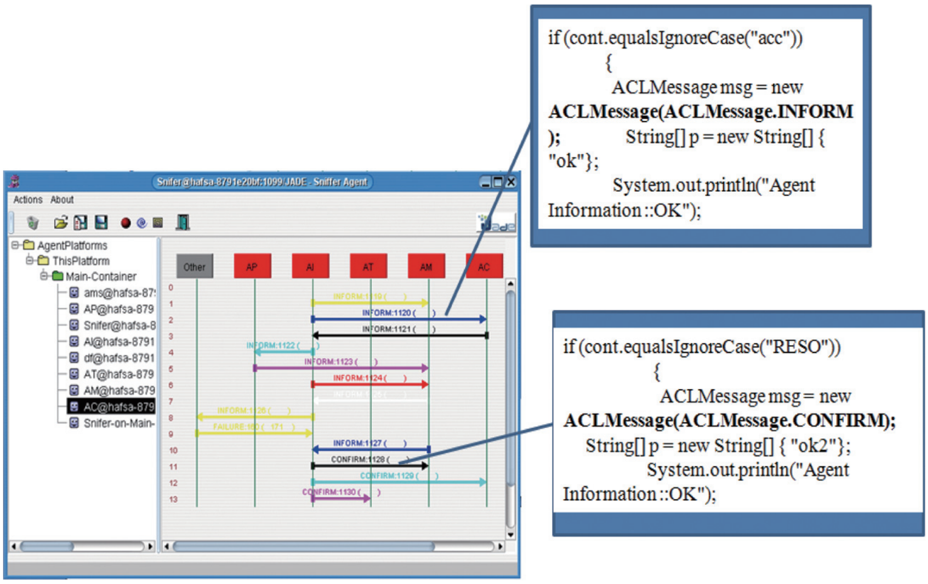


Fig. 8. Communication among agents in JADE.

In the second series of experiments, we deal with the case having some disturbances; thus, we need to reschedule the tasks by involving only the agents directly affected, without disturbance to the rest of the agents that can continue with their work.

By using equivalence coefficients, costs can be evaluated simply and quickly, and it is best to apply the UVA methodology on a company manufacturing one product only (as our case-study).

We next need to choose the base article, a unit that will serve as a measure for evaluating the company's activity. For example as it is described in Figs 10 and 11 the base article is Piece1, piece 2, etc . . . its base rate is calculated. The third step is to calculate the post index or UVA index for each post. Once these indexes are evaluated, the method is considered to be in place.

At the beginning of every simulation run, the planning agent (PA) generates the production order (PO).

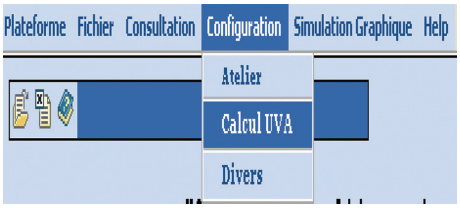
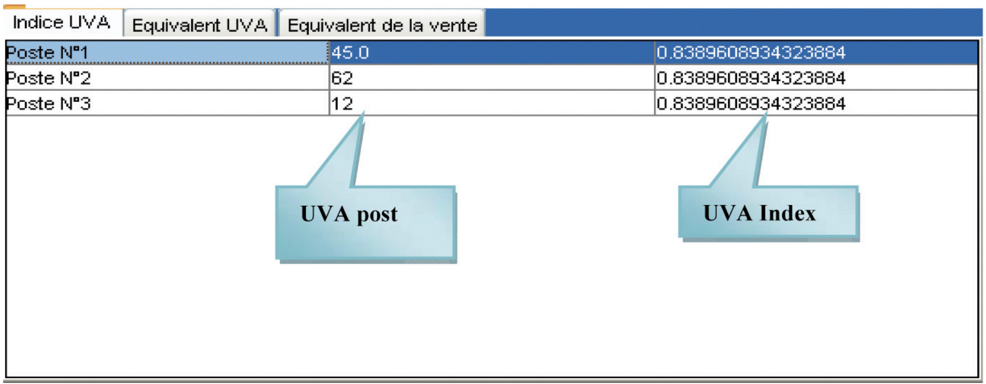
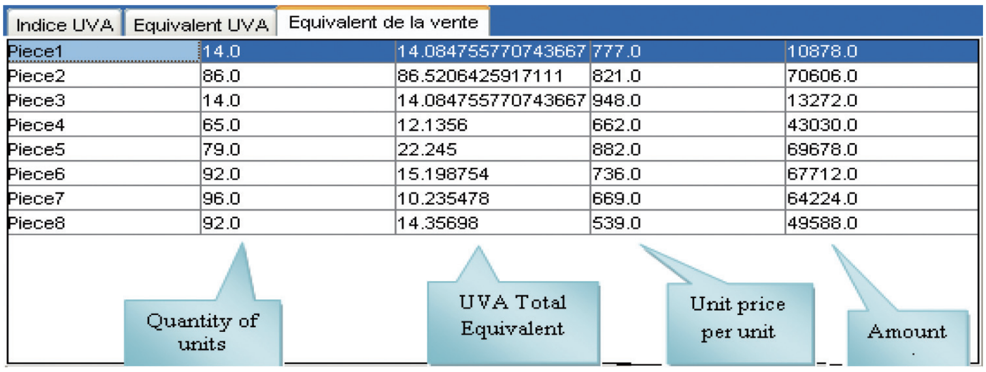


Fig. 9. Cost calculation by UVA method.



Poste N°1	45.0	0.8389608934323884
Poste N°2	62	0.8389608934323884
Poste N°3	12	0.8389608934323884

Fig. 10. UVA index table positions.



Piece1	14.0	14.084755770743667	777.0	10878.0
Piece2	86.0	86.5206425917111	821.0	70606.0
Piece3	14.0	14.084755770743667	948.0	13272.0
Piece4	65.0	12.1356	662.0	43030.0
Piece5	79.0	22.245	882.0	69678.0
Piece6	92.0	15.198754	736.0	67712.0
Piece7	96.0	10.235478	669.0	64224.0
Piece8	92.0	14.35698	539.0	49588.0

Fig. 11. Some relevant parameters are used to calculate total cost in UVA method.

Simulation begins when an order arrives to the RA, at this point, PA schedules the actions to be followed according to the coordination mechanism implied. Each RA makes a local planning and provides the PA with related information.

The prime responsibility of DMA is the pre and post management of input and output data concerned with simulation tool executions. As such, the DMA provides data for use by planning agents (PA), whose principal duty is to execute its associated program for scheduling.

The primary responsibility of the planning agent (PA) is to schedule the outstanding simulation tool executions utilising the workstation within the manufacturing environment. Planning agent looks for database which stores available scheduling plans and returns the process plan for a specific production order. The planning agent has a number of additional responsibilities including maintaining of the task model and managing dependencies between simulation tool executions and decision-making regarding the general scheduling.

Thus, PEA executes UVA program with the right input data at the right time. This involves in addition, carrying out dependency checks to make sure that any preceding simulation executions have been completed before that under consideration can start.

Test case 2 can be regarded as another variation of

Piece	Tps d'execution	Tps debut de panne	Tps fin de panne	Tps debut d'attente	Tps fin d'attente	Durée de transfert	TOTAL
Piece1	21	00:00:00	00:00:00	21:26:26	21:28:18	00:00:05	21.433332
Piece2	5	00:00:00	00:00:00	21:26:26	21:27:55	00:00:05	21.437778
Piece3	11	00:00:00	00:00:00	21:26:26	21:28:09	00:00:05	21.436111
Piece4	21	00:00:00	00:00:00	21:26:26	21:28:26	00:00:05	21.433332
Piece5	8	21:27:35	00:00:00	21:26:26	21:26:26	00:00:05	21.456112
Piece6	12	00:00:00	00:00:00	21:26:26	21:28:26	00:00:05	21.435833
Piece7	10	00:00:00	00:00:00	21:26:26	21:28:21	00:00:05	21.436388
Piece8	27	00:00:00	00:00:00	21:26:26	21:28:22	00:00:05	21.431667
Piece9	29	00:00:00	00:00:00	21:26:26	21:28:26	00:00:05	21.43111

Fig. 12. Time management during production: the most important values.

test case1, where certain resources break down. The breakdown only in resource caused a significant degradation in system performance. The realization time is greatly increasing from 50 to 100 units. The realization duration is an important factor in calculating overall production cost. In Fig. 13, the results demonstrate that when the frequency of resources disturbances increased, the average production time and the percentage of products on time were affected.

The performance measures considered in such a generic shop model include mean flow time, mean tardiness, throughput, buffer size, and machine utilization (see Fig. 12):

– The mean flow time refers to the average time taken from the release of a job into the shop to its departure after its last operation;
– The mean tardiness represents the average difference time from the due date of a job to the real finishing time of a job;
– The throughput is the total number of completed.

To help the human operator to improve his decision-making in managing the production system, we translated the relevant data such as the calculation of no-producing times and statistical breakdowns of resources in the form of a histogram and sector. Access to these statistics is done via the interface coordinator agent. These statistics are intended to improve production while minimizing the time. The following figures (Figs 14 and 15) illustrate the statistics calculated by the planning agent.

7.1. *Simple scenario: Coordination aspects*

In order to test the proposed coordination strategies a company which produces a special computer installation with its own hardware and software for a customer has been simulated through the described simulation environment. The first set of experiments studied the performance of the system according to the developed package and dealing with a normal situation.

The second set of experiments was to study the system replanting behavior when external and unpredictable event that affect normal behavior of the system. In order to carry this out, the simulation was tested introducing resources break downs that would cause changes in the system behavior.

Let us consider a Production Order (PO) which consists of three tasks T1, T2 and T3. These tasks are performed on resources R11, R12 and R13 respectively, and under the control of the Resource Agent RA1 (T3 begins when T1 and T2 have completed their executions). In normal operation (no failure or unexpected event) the state of execution is shown in Fig. 16, where all movements in the shop and detailed execution of tasks on resources are preserved in the overall agenda of the coordinator agent (PA).

7.1.1. *Test case 1: Local decision-making by the resource agent*

Keeping the previous example and assume that the resource R11 fails: the agent consults its resource table (fault history) and find the trace of the fault. It informs the coordinator agent (PA) of the current state of production and takes a local decision (see Fig. 17).

Fig. 13. Simulation results.

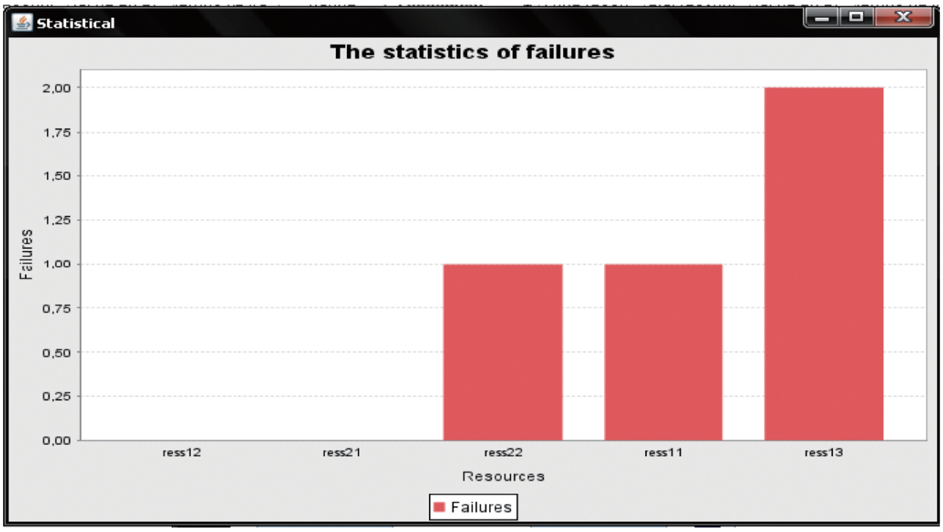


Fig. 14. Statistics on the number of resources breakdowns.

7.1.2. *Test-case 2: Decision–making on the planning agent*

When the coordinator agent (PA) receives the message from the resource agent informing that resource R12 breaks down (P6), it consults its history of failures, and identifies the failure P6 as shown in the Fig. 18. For this scenario, when a machine breaks down, the RA checks its faults history table then it realizes that it is a new break down, RA calls PA. This latter also consults its table that is more general than the RA and finds the

(a) Some relevant time values. (b) Statistics at the nonproductive time of resources

Fig. 15. Statistics at the nonproductive time of resources.

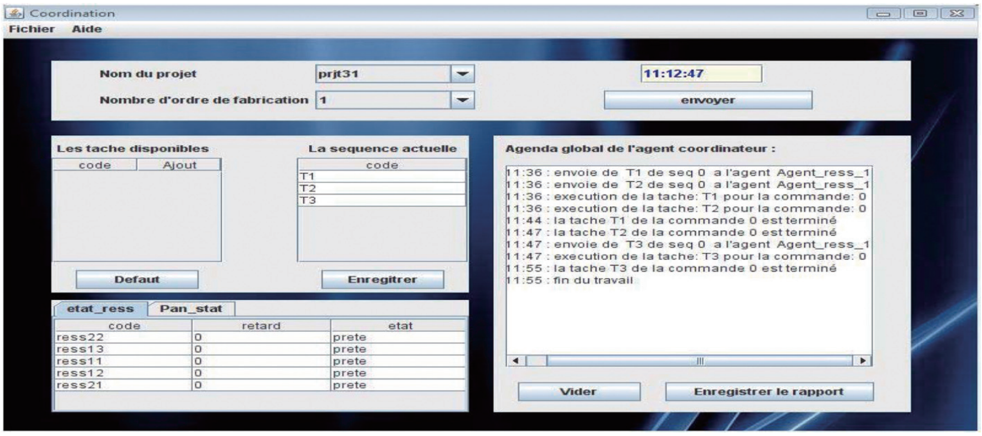
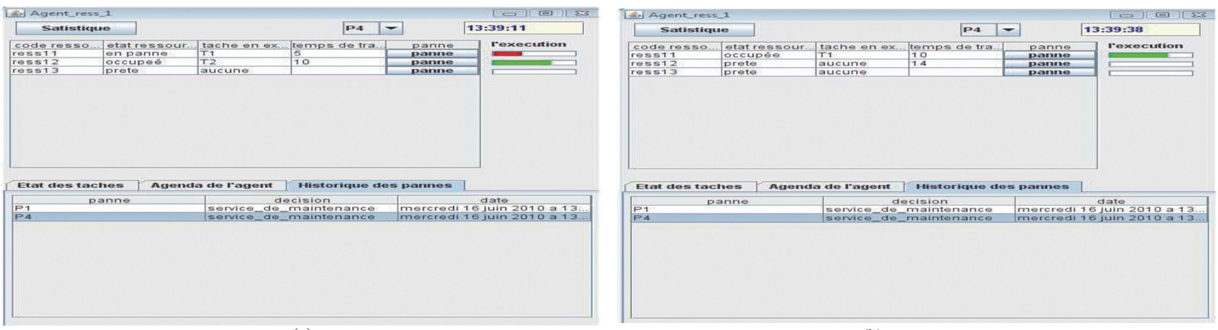


Fig. 16. The details of execution kept on the agenda of the coordinator.



(a) (b)

Fig. 17. (a) Resource R11 breaks down (P4). (b) Problem resolution locally.

corresponding decision in respect of the failure. PA sends the decision to RA so it can update its history table.

Otherwise, PA will contact repairing service to find a solution (see Fig. 19). In this case, the Planning Agent (PA) ignores the fault (P22) because it does not keep track of its diagnosis in his table. Therefore, he contacted the maintenance service or the provider (Internet access). Figure 20 shows the interactions among agents in JADE platform.

8. Conclusions and future work

Agent-orientation is an appropriate design paradigm to enable automatic and dynamic collaborations, especially for manufacturing systems with complex and distributed transactions. The approach presented in this paper proposes a methodological framework for simulation and software architecture. This last is based on the concepts of the multi-agent systems.

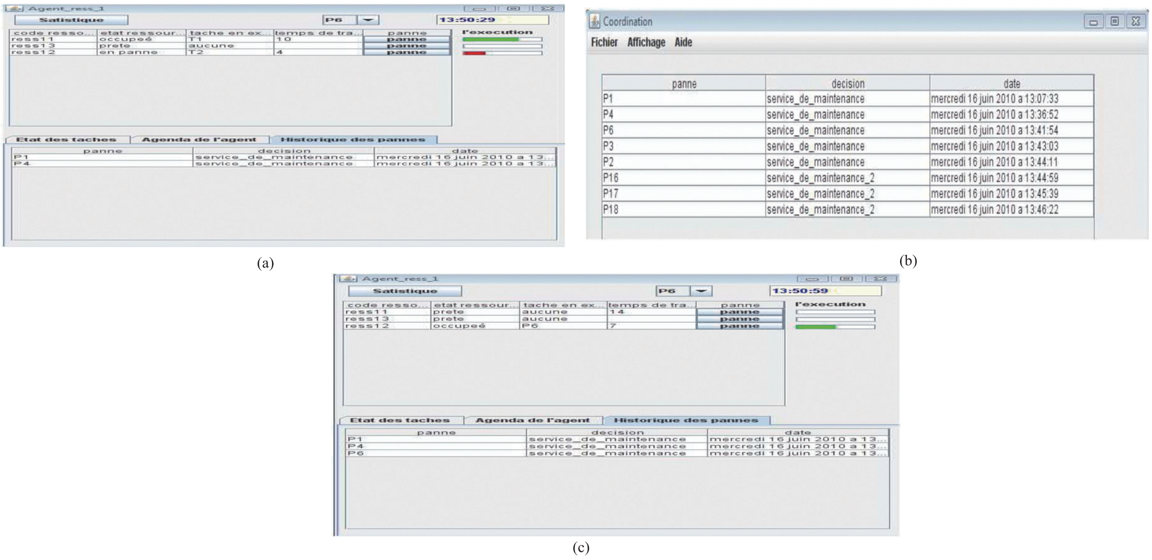


Fig. 18. (a) Resource R12 fails (P6); The failure (P6) doesn't exist in the table. (b) Failure (P6) exists in PA's table. (c) Problem resolution at PA level.

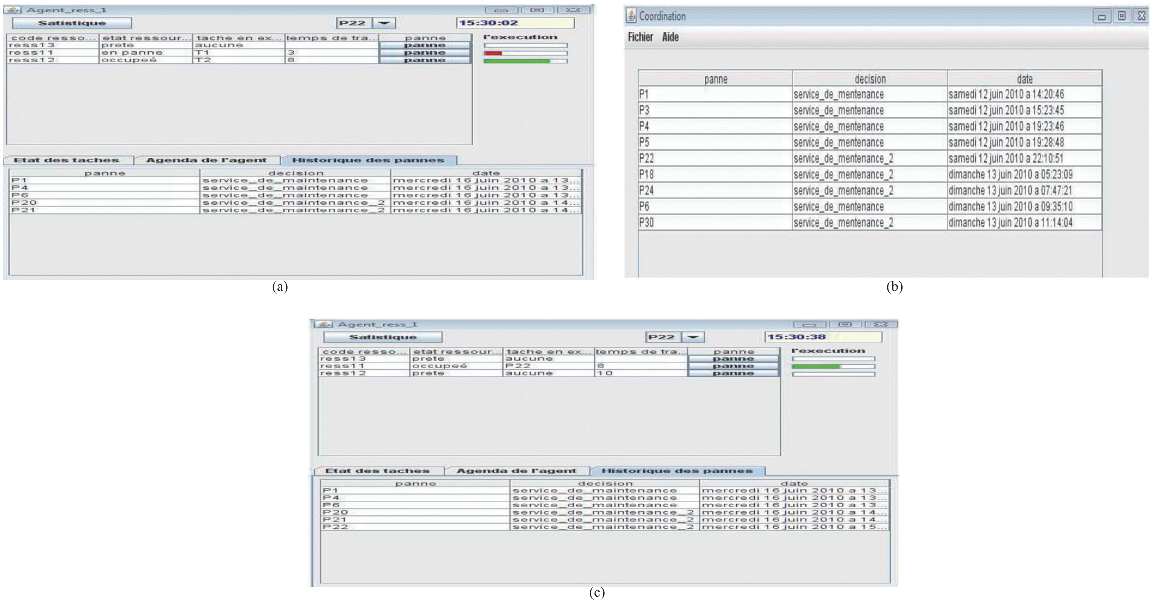
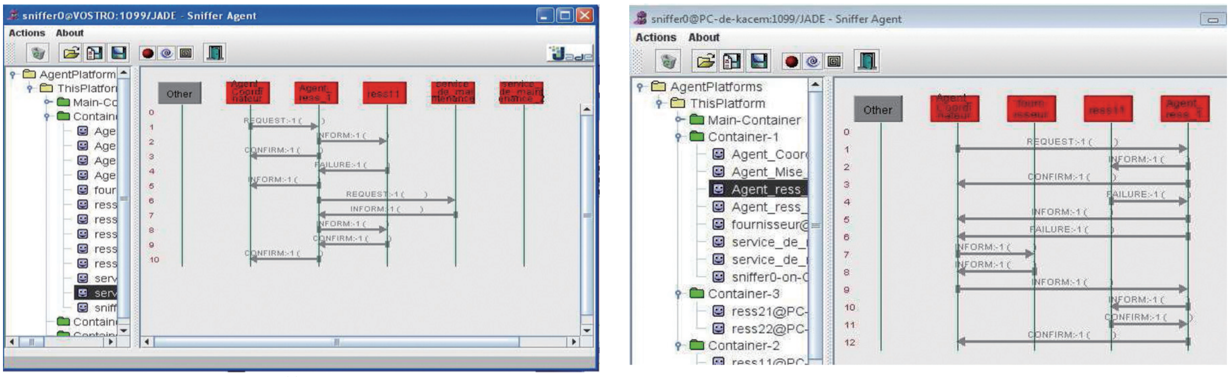


Fig. 19. (a) Resource R11 fails (P22); The failure (P22) doesn't exist in the table. (b) Failure (P22) is unknown for PA. (c) Problem resolution at repairing service.

We have presented a brief state of the works on the three domains that influence our works: the modeling of enterprise, the Intelligent Decision Support System (IDSS) and the multi-agent systems.

Our methodology allows a more powerful use of sub models. It also permits a better understanding of the simulation model in order to distinguish the different kinds of flows in the manufacturing system. In this paper, the proposed approach provides some features which address additional aspects of coordination that we have presented in previous work [21].

Integrated in DSS, agents have the ability to adapt to the changing environment and to handle the system complexity. Thus, our contribution includes developing the results we presented depending on UVA method, and identifying some of the several factors that play a



(a) (b)

Fig. 20. (a) JADE platform agents-test-case1. (b) JADE Platform agents-test-case2.

role in determining the coordination of our modeling agents.

The model of decision support presented in this paper will contribute positively to the improvement of performance in a production system. The results obtained by the multi-agent simulation using the UVA cost method will provide an efficient and scalable tool for decision making in real time. Moreover, the intermediate results will be exploited later to improve the best model. The multi-agent simulation allows us to represent all agents able to act and react to changes in production to improve overall performance at each level of decision-making.

Experiments are performed to prove that the underlying information technologies used in the prototype system, such as coordinating multi-agent and DSS using, are useful by statistical test. There are some limitations of the present experimental setup. It focuses on cases where the resource conflicts are direct and immediately perceived, the resources are homogeneous, and the agents all use the same decision-making protocol. In addition, we can say that JADE is particularly suitable for our application because it involves several interactions among internal agents and permits the execution of possibly long and complex tasks that can be triggered at any point in time specially for the simulation program execution.

In future work we need to consider several improvements in order to achieve greater scalability in the agent's coordination to allow an access to the WEB. We have validated our decision-making mechanism on a very simple example of production and it would be very interesting to adapt it to more complex model.

References

[1] H. Abbad and N. Taghezout, Proposition d'un modèle décisionnel pour évaluer la performance d'une production, *The International Conference on applied Informatics ICAI09*, Bordj Bou Arreridj, November, (2009), 15–17.

[2] F. Bellifemine, G. Caire, A. Poggi and G. JADE, A software framework for developing multi-agent applications. Lessons learned, *In Information and Software Technology* **50** (2008), 10–21.

[3] Z. Bing-hai, W. Shi-jin and X. Lifeng, Agent-based decision support system for dynamic scheduling of a flexible manufacturing system, *Int J Computer Applications in Technology* **32**(N°. 1) (2008), 47–62.

[4] H. Bouquin, *Comptabilité de Gestion*, Economica, (2000), 97.

[5] Y. Chevaleyre, N. Maudet, P. Sousa et al., *Issues in Multiagent Resource Allocation, In Informatica* **30** (2006), 3–31.

[6] G. Coates, Agent co-ordination aided distributed computational engineering design, *In Expert Systems with Applications* **31** (2006), 776–786.

[7] Z.X. Guo, W.K. Wong, S.Y.S. Leung and J.T. Fan, Intelligent production control decision support system for flexible assembly lines, *In Expert Systems with Applications* (2008), doi:10.1016/j.eswa.2008.03.023.

[8] JADE Web Site, Available from: <http://jade.tilab.com/>.

[9] N.R. Jennings, Coordination techniques for distributed artificial intelligence, in: *Foundations of Distributed Artificial Intelligence*, G.M.P. O'Hare and N.R. Jennings, eds, Wiley, London, 1996, 187–210.

[10] R. Kee: The sufficiency of product and variable costs for production related decisions when economies of scope are present. In *International Journal of Production Economics* (2008), doi:10.1016/j.ijpe.2008.03.003,1-32.

[11] P. Leitao, Agent-based distributed manufacturing control: A state-of-the-art survey, *Engineering Applications of Artificial Intelligence* **22** (2008), 979–991.

[12] M. Mahesh, S.K. Ong and A.Y.C. Ne, Communication and Coordination in Multi-Agent Manufacturing System, *International Journal of Manufacturing Research* **1**(1) (2006), 59–82.

[13] V. Marik and J. Lazansky, Industrial applications of agent technologies, *Control Engineering Practice* **15** (2007), 1364–1380.

[14] G. Lo. Nigroa and S. Noto, Coordination policies to support decision making in distributed production planning, *Robotics and Computer Integrated Manufacturing* **19** (2003), 521–531.

[15] Y. L. Pesor and O. de La Villarmois, Origine et développement d'une méthode de calcul des coûts: la méthode des Unités de Valeur Ajoutée (UVA), *In Communication In 4e Congrès International de Génie Industriel*, (June 2001).

[16] Q. Liu, Y. Liu, J. Zheng, Multi-Agent Based IDSS Architecture and Negotiation Mechanism, *In Journal of Beijing Institute of Technology* (*Chinese Language*) **21**(1) (Feb 2002), 6–10.

[17] R. Ribeiro, Intelligent Decision Support Tool for Prioritizing Equipment Repairs in Critical/Disaster Situations. In: *Proc Of Euro Working On Decision Support Systems Workshop*, London, England, 2006.

[18] E. Sevtsenko, R. Kutt and T. Kaaraulova, Using of multi-agents in an intelligent decision support system for collaborative SME-s, *In 1er Nordic on Product Lifecycle Management-NordPLM'06*, Goteborg, 25–26 January 2006, 123–134.

[19] W.M. Shen, Distributed Manufacturing Scheduling Using Intelligent Agents, *In IEEE Intelligent Systems* **17**(1) (2008), 88–94.

[20] N. Taghezout and P. Zaraté, A Hybrid Approach for Designing an Adaptive User Interface: *IDSS and BDI Agents*, T.-h. Kim et al., eds, ASEA 2008, CCIS 30,. Springer- Verlag Berlin Heidelberg, ISSN 1865-0929 (Print) 1865-0937 (Online), 2009, 164–182.

[21] N. Taghezout and P. Zaraté, Supporting a multicriterion decision making and multi-agent negotiation in manufacturing systems, *In Intelligent Decision Technologies Journal, Publisher IOS Press*, ISSN 1872-4981, **3**(3) (2009), 139–155.

[22] A. Tharumarajah, Survey of resource allocation methods for distributed manufacturing systems, *Production Planning & Control* **12**(1) (2001), 58–68, DOI: 10.1080/09537280150203 988, URL: http://dx.doi.org/10.1080/09537280150203988.

[23] E. Turban and J. Aronson, Decision support systems and intelligent systems, *Prentice-Hall International*, Upper Saddle River, New Jersey, 2001.

[24] J. Tweedale, N. Ichalkaranje, C. Sioutis, B. Jarvis, A. Consoli and G. Phillips-Wren, Innovations in multi-agent systems, *J Network and Computer Applications* **30** (2007), 1089–1115.

[25] Olivier de La Villarmois and Yves Levant, Simplified Customer Profitability Evaluation: The equivalence based method, *Cost Management* (2009), 28–34.

[26] F. Von Martial, Cooperation and Coordination of Intelligent Agents, in *Coordinating Plans of Autonomous Agents*, Springer-Verlag, 1992.

[27] W. Xiang and H.P. Lee, Ant colony intelligence in multi-agent dynamic manufacturing scheduling. *In Engineering Applications of Artificial Intelligence* **21** (2008), 73–85.