MAPPS : A Multi-agent Environment for Market-Driven Production Planning

M.Kamal and M.Zaki

Computer Engineering Department, Al-Azhar University, Egypt
Moumen.kamal@gmail.com, mzaki.azhar@gmail.com

Abstract

This paper presents MAPPS as an agent based software environment for market-driven production planning in manufacture enterprise. Such advanced technology is promising and can be utilized effectively to improve the decision process. MAPPS, as a software facility that aims at exploiting the multi-agent production planning has common features with other similar software environments. These features include the ability to measure every agent stat and to ensure planning consistency. However MAPPS has several unique characteristics that are: providing adaptive solutions, inspiring STRIPS as an intelligent planning scheme, accommodating the enterprise production model, utilizing an arbitrary chosen strategy to find *o*ut the appropriate resource, and performing active actions in response to its observations.

MAPPS consists of four agent-based modules: global manager, monitor, database mobile reader and workshop handler agent. The enterprise environment, as such, is simulated and tested. The experimental results at various operation conditions have confirmed:

i- The effectiveness of the approach of multi-agent production planning for industrial and/or financial enterprise.

ii- The value added by distributed artificial intelligence to software decision environment.

iii-The practical significance of the unique features of MAPPS.

Keywords: *Market driven, Production planning, STRIPS, Multi-agent platform, Production Model.*

1. Introduction

Market driven planning is a real problem for financial and industrial enterprise. Such problem to be properly solved, needs solution to the following sub problems: market survey, locating suitable resource and providing feasible plan for goal realization [1].Industrial and software engineers have found out that distributed artificial intelligence can make use of multi-agent systems to present effective solutions to the above sub problems and consequently to the entire problem of market-driven planning.

In this paper MAPPS is presented as a multi-agent software environment to support production planning systems. Thus, similar to multi-agent plan coordinators it realizes:

i- Resolving conflicts between system agents.
ii- Avoiding consistence flows by affording synchronization between communications actions.
iii- Measures for evaluating every agent present state.

In addition the unique characteristics of the proposed multi-agent system (MAPPS) can be pointed out in the following:

1- Use of explicit search strategy to find out the required resource>
2- Providing adaptive solutions that can accommodate multiple plan strategies.
3- Executing active actions by the system software agents.
4- Inspiring STRIPS [2] as an intelligent planning scheme.
5- Hosting and performing the enterprise production model.

Basically, MAPPS consists of 4 modules and its architecture is illustrated in the following:

1- Manager Agent that consists of two layers:
   i- Interface layer that acts as a user interface for MAPPS.
   ii- Processing layer that acts as a kernel for coordinating all MAPPS, either static or mobile.
2- Monitor Agent: it is responsible for:
   i- Keeping track of every agent's and consequently, the global system stat.
   ii- Securing the system by analyzing the exchanged messages to ensure that there is no attacks (either external or internal).
3- Database Mobile Agent: to travel across market databases in order to collect and return up to date values for the quantities already exists in the underlying market.
4- Work Shop Agent: that agent can perform:
   i- Computation of a particular amount of certain product so that a best utilization of the available resources could be achieved.
   ii- Realization of the enterprise production model.
   iii- Active actions that may include shutdown of certain resource and/or disabling a particular communication part.

MAPPS has been designed and belt on JADE platform [3].

Accordingly it is tested under varicose operational conditions with utilizing either first-fit or best-fit as a resource location search strategies. Actually, such tests have not only verified and confirmed the concept of using multi-agent systems for market-driven production planning but also emphasized superiority of MAPPS as a STRIPS based system.

The rest of the paper is organized as follows section 2 includes the related works either for multi-agent coordination or multi-agent plan coordination, while section 3 presents MAPPS architecture. Section 4 describes the MAPPS implementation and investigates its performance.

Eventually section 5 presents the conclusion and the recommendation for future work

2. Related Work

Plan coordination works best when agents' plans are loosely-coupled, meaning that their individual plans for the most parts contain weak interactions. Otherwise, the agents might benefit more from centralized planning. This characteristic of the plan coordination problem has lead to solution techniques that emphasize viewing the coordination process as effectively merging the individually-formed plans into a coordinated whole. Plan merging techniques have been developed both in the context of single agents (where plans for different sub goals are generated separately and then merged into an overall plan for an agent), and for multiple agents. The emphasis in most of the prior work has been on analyzing the ways that possible events could play out over time given the separate plans, and identifying problematic states that could arise, requiring adjustments to the separate plans to avoid these states.

2.1 Multi-agent Coordination

Multiple agents operating in a shared environment might, either by necessity or by choice, want to identify and then either avoid or exploit potential interactions between their activities. Such interactions can arise in a variety of circumstances, including when there exist scarce resources that the agents must share, or when one agent operates on the same part of the world that another agent does. There has been a great deal of research on possible approaches to solving these problems, including work with Contract Nets [4], market-based multi-agent resource allocation methods [5], as well as the development of social laws that facilitate cooperation between agents [6].However, much of this work assumes that information about the points of interaction (e.g., resources) are predefined and thus known to the coordination method at the outset of the problem.

2.2 Multi-agent Plan Coordination

Multi-agent plan coordination has been an issue of interest in the multi-agent systems community for many years, dating at least back to the seminal work of Georgeff [7]. Georgeff's multi-agent planning work concentrated on the problem of resolving conflicts and potential clobbering actions between the plans of different agents with different goals to ensure safe plan execution. Thus, it emphasized finding consistency flaws, and resolving them by inserting synchronizing communication actions. Georgeff modeled individual plan actions as consisting of a series of states, represented as an unordered set of conditions, where these conditions were considered to hold during the action. This specification of "during conditions" allows his system to reason about the implications of actions occurring in parallel. Specifically, actions with conflicting "during conditions" could not be executed simultaneously. In the majority of that work, the emphasis of coordination has been devoted to ensuring consistency. Thus the actions of various agents should be ordered in such a manner that the underlying agents can avoid interference with each other, since negative interactions could entirely prevent goals from being achieved.

2.3 Multi-Agent Planning for Constrained Satisfaction Problem

The work proposed by [8] presents A Distributed Multi-Agent Planning Algorithm, DMAPA, that uses distributed constrain satisfaction to coordinate between agents, and local planning to ensure the consistency between these coordination points [9]. To solve the Constrain Satisfaction Problem CSP efficiently, the existing methods should be modified to take the advantage of the structure of the underlying planning problem. This algorithm shows the scalability beyond the state of the art centralized solver. This work also provides a real-world setting for testing and evaluating distributed constrain satisfaction algorithm in structured domain and illustrate how existing techniques can be altered to address such structure.

2.4 Mapping Planning to a Constraint Satisfaction Problem

Do and Kambhampati [10] describe a method of translating GraphPlan's planning graph [11] into a CSP that can then be solved using standard CSP solvers. More recent work by Lopez and Bacchus [12] extends this work, by passing the Graphplan structure altogether to better exploit the structure of the planning problem, resulting in even better computational performance as well as the generalization of their method to richer planning models. The work of in translating the MPCP to a Constraint Optimization Problem, COP, extends this past work, to offer new ways for encoding the problem as a COP in order to fit the constraints of the distributed optimization framework that has been used.

2.5 Distributed Coordination of Mobile Agent Teams

This work considers the problem of coordinating a team of agents engaged in executing a set of inter-dependent, geographically dispersed tasks in an oversubscribed and uncertain environment [13]. Where there are sequence-dependent setup activities (e.g., travel), we argue that there is inherent leverage to having agents maintain advance schedules. In the distributed problem solving, each agent begins with a task itinerary, and, as execution unfolds and dynamics ensue (e.g., tasks fail, new tasks are discovered, etc.), agents must coordinate to extend and revise their plans accordingly. The team objective is to maximize the utility accrued from executed actions over a given time horizon.

The approach to solving this problem is based on distributed management of agent schedules. They describe an agent architecture that uses the synergy between intra-agent scheduling and inter-agent coordination to promote task allocation decisions that minimize travel time and maximize time available for utilization activities. Experimental results are presented to compare the underlying agent's performance to that of an agent using an intelligent dispatching strategy. Across a range of problems involving a mixture of situated and non-situated tasks the scheduling approach that has been proposed outperforms the system dispatching strategy.

2.6 ExPlanTech: Applying Multi-agent Systems in Production Planning

The mission of the ExPlanTech technology transfer project is to introduce, customize and exploit the multi-agent production planning technology [1] (ProPlanT multi-agent system research prototype) in two specific industrial enterprises. An agent driven service negotiations and decision process based on usage-centered knowledge about task requirements substitutes the traditional production planning activity. Methodology has been introduced for integration of the project-driven production planning based on agent-based engineering within the existing enterprise resource planning system. This production planning technology will facilitate optimization of resource utilization and supplier chain while meeting the customer demands.

2.6.1 ProPlanT Multi-Agent Platform

ProPlanT agents have been implemented and they run on WinNT 4.0 operation system. Each agent is an independently running application. The agents exchange messages via TCP/IP sockets. The agents use, KQML as an ACL (agent communication language) and KIF as the ACL content language.

2.6.2 ProPlanT Production Planning Architecture

In the ProPlanT the classical planning and scheduling mechanisms have been substituted by the processes of negotiation, job delegation and task decomposition within a community of autonomous agents, each of which represents production or information unit(s) of the modeled factory. There is no central agent or any central control mechanism. ProPlanT system relies upon two fundamental super-classes of agents: intra-enterprise (IAE) agents and inter-enterprise (IEE) agents. In the category of the IAE agents ExPlanTech distinguishes among production planning agents, production management agents , production agents , customer agents, and meta agents.

2.6.3 Role of ExPlanTech

The optimal production plan should balance the available LIAZ resources while maximum number of orders is produced. Considering the limited LIAZ workshops/shop floors capacity the decision has to be made whether the specific task/subtask will be provided internally or subcontracted externally. Such decision might be crucial in order not to threaten successful completion of other orders and misbalance the whole production flow. The efficient supply chain/service provision management can be handled using the agent-based approach as described in the ProPlanT technology within the inter-enterprise and extra-enterprise level. The ExPlanTech technology integration is aimed at the improved strategic decision making, e.g. to efficiently use available resources (human, material supply, workshop capacity) while the maximum number of clients' orders have been satisfied and carried out.

2.7 Discussion

The above works have a common strength that is providing an integration of plan based production strategies with agent based engineering for better utilization of enterprise resources. However all of them do not afford any of the following aspects:

i. Proposing an explicit algorithm to determine how to discover the required resource.

ii. Obtaining maximum benefit from MAS capacities.

iii. Utilizing an intelligent yet formal planning technique to perform an adaptive production policy for the enterprise.

iv. Providing active actions including shutdown the resource when no production is taking place (for example due to malfunctioning).

In section 3, MAPPS is proposed as a multi agent system for market driven production planning that exploit the strength and avoid the weakness mentioned above.

3. MAPPS Architecture

The architecture of the MAPPS is based on multi-agents technology in which each agent as a separate role to be accomplished in a certain phase figure(1).

The main agents of the system are:

Manager Agent(MA), Database Mobile Agent(DBMA), Workshops Agent(WSA), and Monitor Agent (MNA)

These agents cooperate with each other to achieve the desired goals. Each agent will be described and his role will be demonstrated.

3.1 Manager Agent (MA)

Manager Agent is the coordinator of all agents in the system Figure(1). It consists of two layers namely Interface Layer and Processing Layer.

Interface Layer

This Layer is responsible for receiving the customer inputs which are: the product type (item), the item quantity, the threshold time (in days).

Interface Layer demonstrates the system outputs in each state and enables the user to choose the plan strategy he/she prefers to implement and finally it displays the produced plan.

Processing Layer

It represents the main function for MA to direct and coordinate other agents in the system to complete their functions each in his turn. The function of the MA starts upon receiving the user requirement from the interface layer. After that MA coordinates the DBMA to start his way to fetch different market databases. Manager Agent receives the needed product situation in the market from DBMA figure(1).

At this time MA has all the needed knowledge (user request and the market situation) to decide according to these values, whether to continue generating a production plan or the market already has enough amounts to fulfill the customer request.

MA sends messages for all WSAs to query their production capacities then MA receives from the involved WSAs their capacities, finally MA makes the plan (will be described later) then sends for each WSAs the result of the generated plan (the time to work and the power percentage they should work with) fig.(1). If the workshop capacity is zero the MA sends to that WSA an order to shut down the workshop.

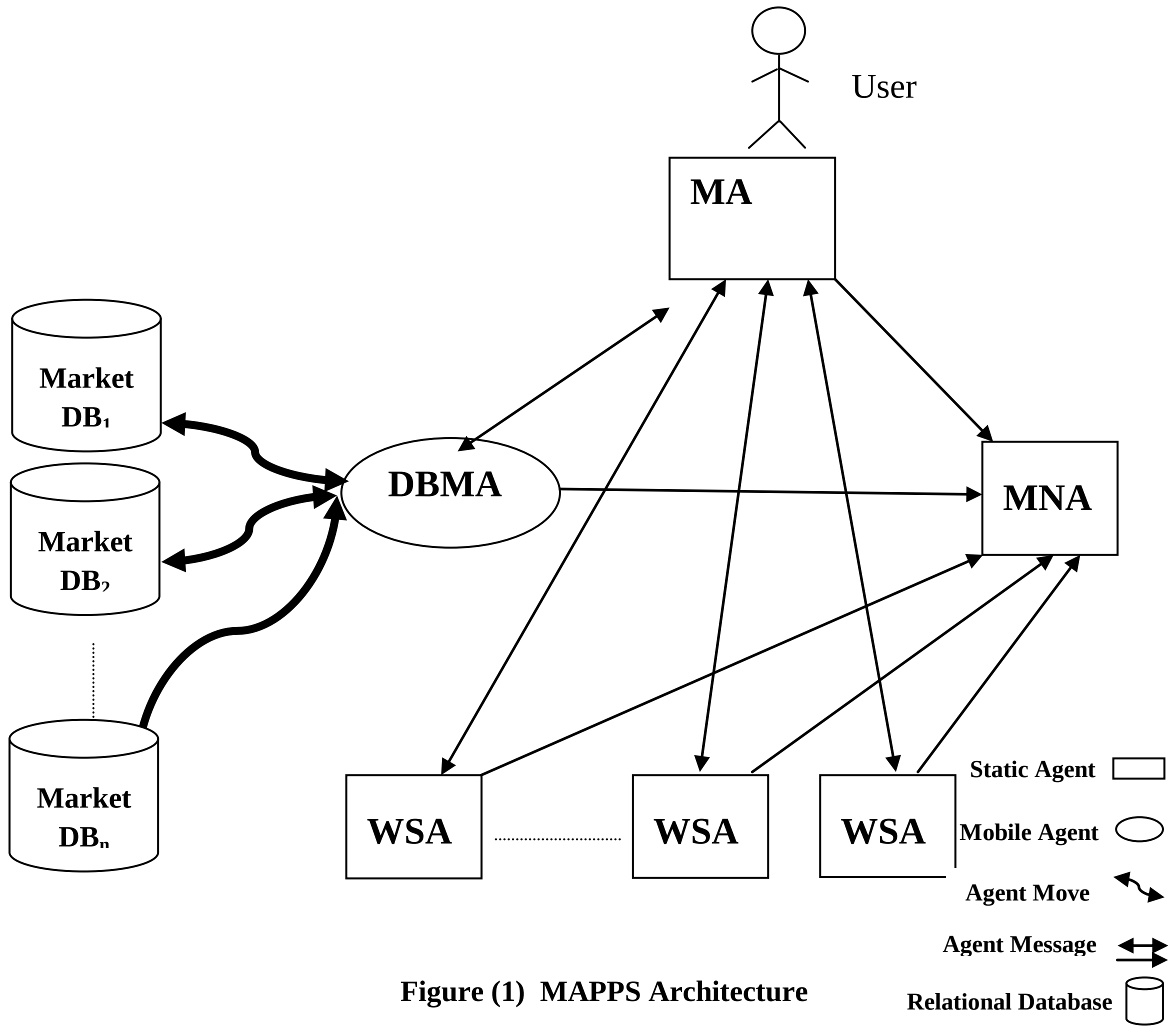

Figure (1) MAPPS Architecture

3.2 Monitor Agent (MNA)

This agent is responsible for watching the system status along the time. MNA receives a copy of all messages circulated between all agents in the system Figure (1) keeping track of each agent's state and thus the whole system state.

MNA helps in securing the system by analyzing the transferred messages and the resulting behavior to ensure that no malicious agents or any external platform is trying to attack the system, MNA also helps in debugging the system by tracing the messages sequence and attach it with the related actions to explain the reason of any error may happened or unrealistic results may be produced. The Algorithm of MNA is not provided since it is usually available as readymade agent.

3.3 Database Mobile Agent (DBMA)

The role of DBMA starts from receiving MA request to start its journey among the market databases to query the market values, Figure (1). DBMA travels to the first database node by its code and state to query the database that stores the values which indicate the stored amount and the requested amount of a certain product. DBMA finishes catching the product status from that database node then it travels with its code and preserving its status, it appends the product status in the second database node to the previous status and so on until it returns back to MA location with all states in all database nodes. It sends a message to the MA contains the result of the product status.

3.4 Workshop Agent (WSA)

This Agent represents the workshop and exists in the system as many as the existence of the manufacturing workshops in real life, Fig. (1). Each workshop has the capability of producing a particular amount of a certain product per day. At the beginning WSA receives a message from MA requesting their production capacities and as a result, WSAs send their capacities (the amount of product a work shop can produce per day) to MA, finally and as last stage WSAs receive accept for proposal message contains the no of days a workshop is going to work and the working power percentage it is going to operate with.

3.5 How does MAPPS work?

MAPPS starts when the user enters:

1 - The product type that customer wants to have.
2- The required product amount.
3- That available days to deliver the need.

The user would like to obtain a plan that provides best utilization from the available resources (workshops), Figure(1). The 1 to n with I in between i.e. and workstations range from $1 \leq i \leq n$.

The capacity of each workstation per dayis c_i (c_i : tons / day).

Thus one of the available plans is chosen to be processed. The two implemented planned strategies are Best Fit and First Fit. These two strategies have been already implemented in the system but any plan strategy could be utilized. The user also has a choice to take the market status into consideration or to ignore it. So the user makes up his mind between four choices, First Fit plan strategy with market status in consideration, Best Fit plan strategy with market status in consideration, First Fit plan strategy with ignoring the market, Best Fit plan strategy with ignoring the market.

The user chooses one of the choices. In case of market status in consideration, MA requests the MDBA to get the market status for a certain product, MDBA receives MA message with the needed product. MDBA travel to the first DB node to query market DB wrt that product to find its consumption versus product availability ratio, then it leaves the current node and travel to the next node holding the result it found, MDBA repeats the previous work and append the new result to the previous one. This happened for each market DB node and the result is appended. After MDBA finishes this work for all available nodes it returns back to MA with the results. MDBA sends message to MA containing the total production supplied to the market and the total market consumption Figure (2).

If the customer needs can be satisfied by the available items in the market then MA will decide to stop the process and inform the user there is no need for workshop production, the market available items can compensate the customer. If the available product will cover only part of that needs, or even market itself need supply MA starts a process to produce a plan.

MA sends messages to all WSAs asking them to send him there production capabilities in the form of the quantity can be produce per day in tons Figure (2). MA receives WSAs capabilities and starts the steps to produce the plan.

3.6 Planning

The planning process passes through three phases Pre Condition, Plan execution, and Post Condition. These phases are to ensure achieving the customer goal, with the best utilization of the available resources.

i. Pre Condition

It is the first phase in plan production in which MA scans the WSAs capabilities. If only one of WSA can afford to produce the whole customer needs, then the pre condition phase reaches the desired plan and the task will be assigned to this workstation, in this case no need for Plan Execution phase just jump to Post Condition phase to refine the results.

$d \times c_i \geq q \qquad 1 \leq i \leq n$

i : the chosen workstation that can afford produce the needs.

d : number of days to the dead line.

q : required quantity to be produced.

ii. Plan Execution

In case of the Pre Condition phase could not accomplish the customer needs the Plan Execution phase will start and the chosen strategy among the two implemented strategies will be discussed next section. By the end of this phase either there is no solution for this customer need in the needed time limit, or a plan showing the workshops will be used and the duration will be taken in production.

The set of all available workshops is

$$\{C\} = \{c_1, c_2, c_3, \ldots\ldots c_i, \ldots\ldots\ldots c_n\} \qquad 1 \leq i \leq n$$

The set of the chosen workshops that produced after executing one of the available plan strategies is

$$\{\boldsymbol{C}\} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3, \ldots\ldots \boldsymbol{c}_j, \ldots\ldots\ldots \boldsymbol{c}_m\} \qquad 1 \leq j \leq m$$

Where $m \leq n$ $\qquad \{\boldsymbol{C}\} \subseteq \{C\}$

The plan execution result is

$$d(\boldsymbol{c}_1 + \boldsymbol{c}_2 + \ldots\ldots + \boldsymbol{c}_j + \ldots\ldots\ldots + \boldsymbol{c}_m) \geq q \qquad 1 \leq j \leq m$$

iii. Post Condition

In case of a plan produced Post Condition Phase will take place. The advantage of this phase is to enable co-operation between the chosen WSAs, by other words all WSAs will co-operate with each other to produce the amount of product needed by the customer in the time limit.

MA compares between the maximum capacities for WSAs that involves in the produced plan and the customer needs. MA generates a fraction in order to decrease the work load on the workshops by this fraction each WSA will work by a certain percentage of its maximum capability so all WSAs co-operate together to accomplish the user demands.

w = loading (workload) fraction.

$$d\, w_1 \times c_1 + dw_2 \times c_2 + \ldots\ldots + dw_j \times c_j + \ldots\ldots + d\, w_n \times c_m = q \qquad 1 \le j \le m$$

For convenience take

$$w_1 = w_2 = w_j = w$$

$$d\, w \times c_1 + dw \times c_2 + \ldots\ldots + dw \times c_j + \ldots\ldots + d\, w \times c_m = q$$

$$d\, w(c_1 + c_2 + \ldots\ldots + c_j + \ldots\ldots + c_m) = q$$

$$w = q / d\,(c_1 + c_2 + \ldots\ldots + c_j + \ldots\ldots + c_m)$$

MAPPS resource utilization strategies

Several strategies can be exploited to afford the needed production scheduling. Regardless of the underlying strategy, a bit map can be used to represent the status of the system workshops [14]. Corresponding to whether the workshop is busy or free it is expressed by 1 or 0 (or vice versa). Another way of keeping track of the system workshop is making use of linked list of allocated and free workshops. The simplest scheduling algorithm is first fit. The MA can simply search along the list of workshops until it finds a free workshop that is capable enough, and then such a workshop is chosen to perform the task. A minor variation in first fit is next fit [14]. It works the same way as first fit, except that it keeps track of where it is when it finds a workstation. The next time it is called, it starts searching from it left off, instead of always it starts at the beginning, Anther well known algorithm is best fit. It searches the entire list of workstations and chooses the free workshop of the enough smallest capacity.

The proposed system Implements two plan strategies the user can choose between them to be used in Plan Execution.

3.6.1.1 First Fit Strategy

A "first fit" algorithm is a simple algorithm which doesn't care about how "good" a solution is, it just returns the first one that works. In this strategy the system tries using the first two workshops capabilities to fulfill the user needs, if they do not fulfill the requirements the system tries to use the first and the third workshops, and so on until the system expose to the user the first two workshops that can do the work if they are already available then jumps to the Post Plan phase.

If no two workshops capabilities can accomplish the task, the system start to try the available combinations of three workshops together, and by the same concept the system expose to the user the first three workshops that can do the work if they exist, and etc until the system finds any successful group of workshops to present it, then the system jumps to Post Plan phase, if the system did not find any successful group of workshops so the task could not be done through the available workshops capabilities and during the given time limit.

3.6.1.2 Best Fit Strategy

A "best fit" algorithm is an algorithm which cares about how "good and suitable" a solution is, it does not return the first one that works as the previous algorithm, but it tries to enhance the plan results according to a certain embedded logic. In this strategy the system tries to finish the task by two workshops, so it scans all the available two workshops combinations and compare between them to choose the least two workshops capabilities that are more than the customer demands to present hem to the user.

If no two workshops capabilities can accomplish the task, the system tries all available combinations of three workshops together, and by the same concept the system expose to the user the best fit three workshops (the least workshops capabilities that are more than the customer demands) if they exist, and so on until the system finds the best fit group of workshops to present it, then the system jumps to Post Plan phase, if the system did not find any successful group of workshops so the task could not be done through the available workshops capabilities and during this time limit.

One of the proposed system advantages is that the plan strategies are modular and scalable so that any other plan strategy can be implemented and invoked to be used by the system.

3.6.2 Active Actions

Active Action is a process or a work done which in his turn changes the state of the target object. This action is fired based on a certain event that meets a certain condition so a trigger is fired to execute the active action.

Regarding to the proposed system WSAs send its production capabilities to MA, in case of this capability is less than the minimum production value that is allowed by MA to continue in process MA will decide to close the workshop that is represented by this WSA, as a result of that a shutdown command will be issued from the MA and transmitted to the WSA, consequently the workshop state is going to be changed from running state to off state. Actually the software agents of MAPPS replace the entire set of human actions. However, in addition to shutting down the not-needed workshops. They execute the following actions:

1 – Pprotecting the system nodes (PC's) by closing that ports which may lead to vulnerability.

2 – Changing dynamically the communication path, in case of multi-criteria mesh networks.

4. Implementation

The Multi Agent System Architecture for Market Driven Production Planning (MAPPS) model has been implemented and tested on a certain environment to watch the performance and record the results.

We will introduce the different levels of implementation environment starting from the hardware until the MAPPS application passing through the used operating system and network also the development environment.

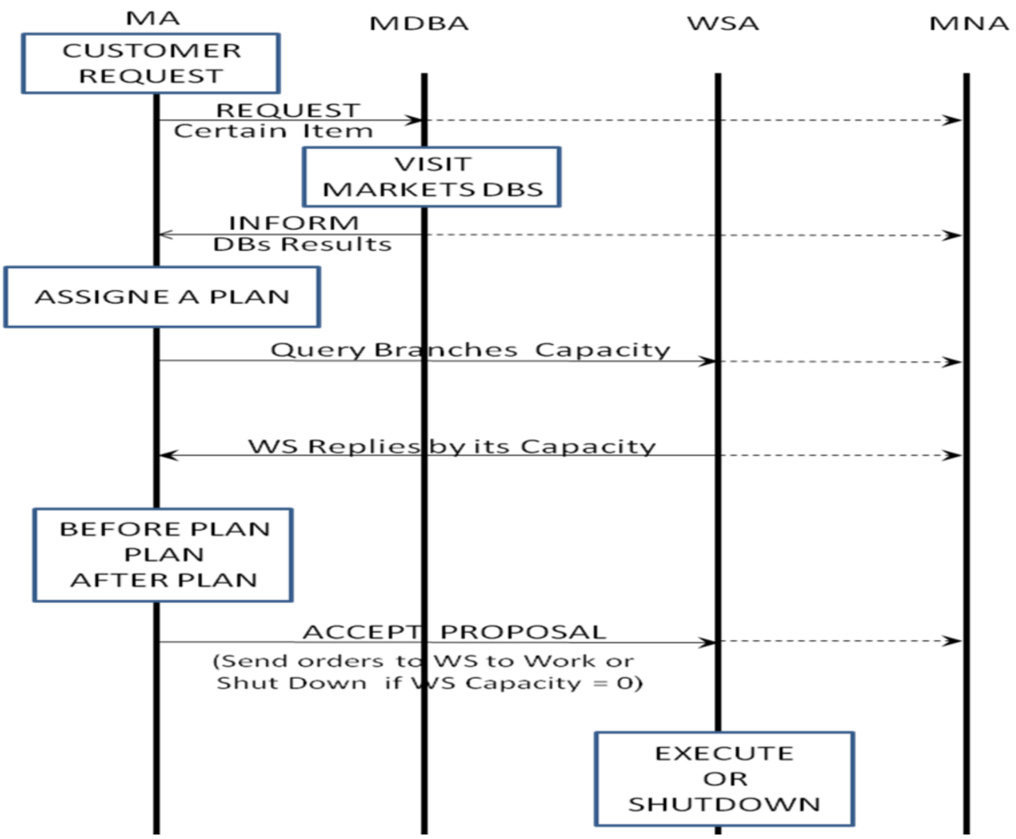

Figure (2) MAPPS sequence diagram

4.1 The setup

The hardware used to establish the implementation environment consists of 4 computer devices with the normal market available configuration Processor : Intel(R) Core(TM)2 Duo with 2.26 GHz. Installed Memory (RAM): 3 GB. MAPPS is portable because it has been built using Java as a development language which is platform independent and can be ported to deferent operating systems.

4.2 Database system

MAPPS would like to be aware of market status as an important piece of information to be taken in consideration and combine them with the user needs cause both are the exact needs that the plan should fulfill. The model represents the market status as a data stored on a database system and as markets are distributed in several places these database systems are distributed on several nodes. The system uses Microsoft access both 2007 and 2010 containing the market status in each database node, The market status in each node is represented by "Market" Table that holds the items and both the produced and consumed quantity.

4.3 Multi agent environment

MAPPS is based on JADE as a software platform that provides the basic middleware-layer functionalities which are independent of specific application and which simplify realization of distributed applications that exploit the software agent abstraction (Wooldridge and Jennings, 1995). A significant advantage of JADE is that it implements this abstraction over well-known object-oriented language, Java, providing a simple API to deal with.

4.4 Request Execution

A customer would like to 700 kg of iron for example in duration of 2 days this is going to be applied to the system and get the result plan to utilize the available resources to provide the customer needs with one of the available strategies (first fit or best fit). First of all the DBMA will query the iron stock value by visiting all the markets DBs, let the first node indicates there is available 10kgs and the other 2 nodes indicate that the available quantity is 30kgs per each so the DBMA returns to the MA with the knowledge that the available iron quantity is 70kgs, so a plan is needed to guide the workshops to produce 630Kgs, Ma requests from WSAs to send their capacities and WSAs reply by 100Kgs/Day, 200Kgs/Day, 300Kgs/Day. MA generates a plan with the chosen first fit strategy and chooses WSA1 and WSA3 for 2 days but this will produce 800Kgs so much more than both market needs and user request so the post plan phase is entered to enhance the results and a workload fraction is produced (0.79) so that WSA1 will produce 158kgs and WSA2 will produce 474Kgs at the end of the time period (2days) 632kgs as a total that quantity satisfies both user and market needs and the plan is put according to the chosen strategy. Ma also issues shutdown command to the malfunction workstation that sent their capacities as 0Kgs/Day

4.5 System Performance

Measures has been taken using the previous described platform with 2WSAs ,3WSAs, and 4WSAs also 2Market DBs, 3Market DBs, and 4Market DBs, in two modes once these WSA and Market DBs, are on one machine and the other time each of them are on separate machines.

Each action is measured with respect to time taken to complete this action. It has been founded that the request MA sends for either DBMA to query market DBs or to WSAs to query their capacities is fraction of millisecond but the time consumed form DBMA to move through market DBs and time consumed to MA receive the capacities of each WSA is the major factor that affect the performance.

Fig (3) is a graph between time (in milliseconds) needed for the MDBA to move through the market DBs and return back to MA with the result versus the number of market DBs either it is located on a single machine or in different machines

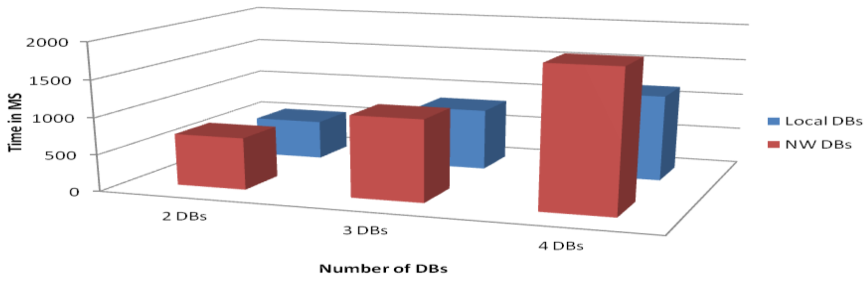

Figure (3) Time needed for DBMA to send reply.

The graph represents that there is direct proportion between time and number of market DBs and this is logically clear because each added node is going to cost MDBA an extra visit process also having the nodes on the same machine consumes less time than having them on separate machines because of the network latency.

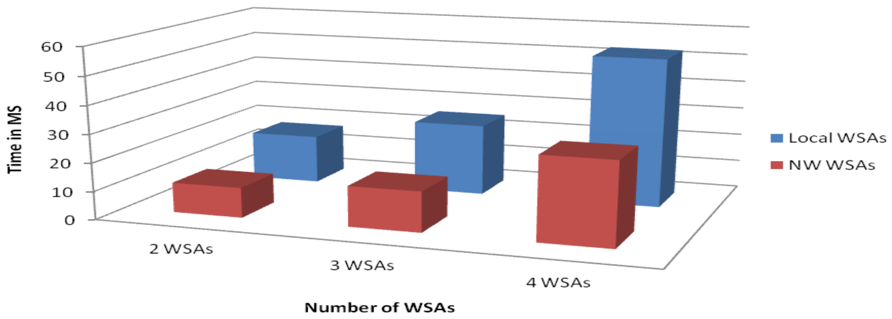

Figure (4) Time needed for MA to receive proposal

Figure (4) is a graph between time (in milliseconds) needed for the MA to receive capacity proposal from WSAs versus the number of WSAs either it is located on a single machine or in different machines.

The graph in figure (4) represents that there is direct proportion between time and number of WSAs and this is logically clear because MA is going to wait more until receive proposal from each added node also having the nodes on the same machine consume more time than having them on separate machines due to the fact that the difference in processing power (between the two cases) is much greater the communications overhead.

5. Conclusion

In this paper MAPPS has been proposed as a software environment that exploits the multi-agent production planning technology in the system the decision making is carried out on the basis of: market information, enterprise supplier capabilities and intelligent planning.

The architecture of MAPPS consists of 4 modules that include a manager agent, monitor agent, database mobile agent, and workshop agent.

Its capabilities when simulated and tested have indicated that it could realize:

i- STRIPS-like planning.
ii- Avoiding consistency flows.
iii- Accommodation of the enterprise production model
iv- First-fit and best-fit search for resource allocation.
v- Active (human-like) actions.

Currently the only artificial intelligent feature of MAPPS is its ability to accomplish production planning. In future is planned to enhance the system agents with machine learning techniques in order to improve the decision process

References:

[1] M Pechoucek1, A Riha1, J Vokrinek1, V Marik1, V Prazma2 1 Gerstner Laboratory, Department of Cybernetics, Czech Technical University, Technicka 2, Prague 6, 166 27,Czech Republic, ExPlanTech: Applying Multi-agent Systems in Production Planning, available at http://agents.felk.cvut.cz/cgi-bin/docarc/public.pl/document/49/3-Pechoucek.pdf

[2] R. Fikes and N. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving, Artificial Intelligence, 2pp189-208, 1971.

[3] F Bellifemine, G Caire, D Greenwood, Developing multi-agent systems with JADE, John Willy and Sons,2007.

[4] R G. Smith, The contract net protocol: High-level communication and control in a distributed problem solver, IEEE Transaction on Computers, Vol. C-29 No. 12, pp 1104–1113, 1980.

[5] M P. Wellman, W E. Walsh, P R. Wurman, and Jeff K. Mackie-Mason, Auction protocols for decentralized scheduling. Games and Economic Behavior, 35:pp 271–303, 2001.

[6] C V. Goldman and J S. Rosenschein. Emergent, coordination through the use of cooperative state-changing rules. In Proc of the 12th Int. Workshop on Distributed AI, Seattle, WA, USA, pp 171–185, 1993.

[7] M Georgeff, Communication and interaction in multi-agent planning systems, In Proceedings of the 8th International Joint Conference on Artificial Intelligence, Karlsruhe West Germany pp 125–129, 1983.

[8] J Cox and E Durfee, Efficient and distributable methods for solving the multi-agent plan coordination problem, available at http://ai.eecs.umich.edu/people/durfee/papers/MAGS09CoxDurfee.pdf

[9] R Nissim, R I. Brafman and C Domshlak, A General, Fully distributed multi-agent planning algorithm, available at http://aamas.csc.liv.ac.uk/Proceedings/aamas2010/pdf/01%20Full%20Papers/27_04_FP_0192.pdf

[10] M B Do and S Kambhampati, Planning as constraint satisfaction: Solving the planning graph by compiling it into CSP. Artificial Intelligence, 132:pp151–182, 2001.

[11] A Blum and M Furst, Fast planning through planning graph analysis, In Proceedings of the 14th International Joint Conference on Artificial Intelligence, San Francisco, CA, USA, pp 1636–1642, 1995.

[12] A Lopez and F Bacchus. Generalizing graph plan by formulating planning as a CSP. In Proceedings of the 8th International Joint Conference on Artificial Intelligence, pages 954–960, 2003.

[13] L Barbulescu, Z B. Rubinstein, S F. Smith, and T L. Zimmerman, The advantage of planning ahead, available at http://www.lirmm.fr/~mahdi/aamas10/27_06_FP_0775.pdf

[14] : A.S Tanenbaum, Modern operating systems, Prentice Hall Inc., London, 1992.