To link to this article: DOI: 10.3182/20120523-3-RO-2023.00120
http://dx.doi.org/10.3182/20120523-3-RO-2023.00120

A distributed multi-agent framework for shared resources scheduling

Jiucheng Xu, Bernard Archimède
Agnes Letouzey

University of Toulouse, INP-ENIT, 47 Avenue d'Azereix,
BP1629, F-65016 Tarbes Cedex, France,
(e-mail: {jiucheng.xu, bernard.archimede, agnes.letouzey}@enit.fr)

Abstract: Nowadays, manufacturers have to share some of their resources with partners due to the competitive economic environment. The management of the availability periods of shared resources causes a problem because it is achieved by the scheduling systems which assume a local environment where all resources are on the same site. Therefore, distributed scheduling with shared resources is an important research topic in recent years. In this communication, we introduce the architecture and behavior of DSCEP framework (distributed, supervisor, customer, environment, and producer) under shared resources situation with disturbances. We are using a simple example of manufacturing system to illustrate the ability of DSCEP framework to solve the shared resources scheduling problem in complex systems.

Keywords: distributed scheduling, multi-agent systems, shared resources, disturbances.

1. INTRODUCTION

The definition of shared resources is firstly mentioned in computer field. Shared resource is either a device or piece of information on a computer accessible from another computer, transparently as if it were a resource in the local one (Galvin, 1994). Extending to manufacturing area, shared resources can be any kind of useful resources during the manufacturing process. These resources belong to enterprises (organizations) with independent accounting and different geographical positions, but can be required by each other. Recently the shared resources problem is studied as a hot spot issue because the resources in a single organization seem to be limited to fit for the rapidly changing market environment. Thereby, manufacturers have to share their resources with partners in order to increase the competitiveness and reduce the production cost.

Manufacturing scheduling determines the most appropriate moment to execute each operation, taking into account the temporal relationship between the acting processes and the capacities of resources (Shen et al., 2006). For shared resources scheduling, each organization constructs a local schedule independently to satisfy its own purposes. These local schedules will lead to conflicts for the scheduling of shared resources. The complexity of the shared resources problem can be compared to the prisoner's dilemma (Le et al., 2007). We can build a virtual enterprise (Molina et al., 1998) to encourage organizations to share resources with partners. In this communication, we will focus on the shared resources problems in complex systems, like manufacturing factories, hospitals, and transport systems etc. which adopt distributed scheduling approach.

This paper is organized as following: section 2 reviews the different scheduling technologies and discusses their limitation. Section 3 gives a brief introduction of the multi-agent model SCEP (supervisor, customer, environment, and producer). Following, we provide a DSCEP framework in order to better identify shared resources solution with disturbance in section 4. Section 5 describes the scheduling process using the DSCEP framework particularly focus on a manufacturing system case study. A brief conclusion and perspectives are stated in section 6.

2. SUMMARY OF SCHEDULING TECHNIQUES

2.1 Traditional techniques

Because of its highly combinatorial aspect (NP-complete) (Zweben et al., 1994), dynamic nature, and practical usefulness for industrial applications, the scheduling problem has been widely studied in the literature by various meta heuristics methods.

Fuzzy logic is an analysis method purposefully developed to incorporate uncertainty into a decision model. Fuzzy logic allows to consider reasoning that is approximate rather than precise. These characteristics made fuzzy logic and tools associated with its use to become quite popular in tackling manufacturing related challenges (Azadegan et al., 2011). Fuzzy logic has been used to multi-objective scheduling in a manufacturing cell (Restrepo et al., 2008), and apply to scheduling rules in flexible manufacturing systems by evaluating multiple performance measures (Chan et al., 2003).

Genetic Algorithms (GAs) is an example of mathematical technology transfer: by simulating evolution one can resolve complicated optimization problems from a variety of sources (Sivanandam et al., 2007). Today, GAs is used to facilitate the integration and optimization of the process planning and

10.3182/20120523-3-RO-2023.00120

scheduling in manufacturing area (Shao et al., 2009). And it also used to solve the resource constrained multi-order scheduling problem (Goncalvesa et al., 2008).

Tabu search (Gendreau et al., 2010) is a "higher level" heuristic procedure for solving optimization problems, designed to guide other methods (or their component process) to escape the trap of local optimality. An efficient tabu search algorithm has been developed to ensure quick decision support for the ship routing and planning. It yields optimal or near-optimal solutions to real-life instances within reasonable time. For large and tightly constrained cases, the tabu search heuristic provides much better solutions than the multi-start local search heuristic (Korsvik et al., 2010).

2.2 Synthesis

In most real-world environments, scheduling is an ongoing reactive process where the presence of a variety of unexpected disruptions is usually inevitable, and continually forces reconsideration and revision of pre-established schedules (Ouelhadj et al., 2009). The traditional scheduling methods encounter great difficulties when they are applied to real-world situations, since they use simplified theoretical models and are essentially concentrated on the sense that all computations are carried out in a central computing unit.

Comparing to traditional methods, modern techniques are more effective. The intelligent agent technologies suggest an innovative and lightweight approach on scheduling problem which could support multiple computing units. The distributed approach is more flexible, efficient, and adaptable to real-world dynamic manufacturing environments (Shen, 2002). The advent and development of network (like Internet) and distributed computing technology provide the possibility of production manufacturing with distributed scheduling approach (Kornienko et al., 2004).

Multi-agent systems have been successfully applied to the scheduling problem for some time. In next section, we will describe an existing multi-agent model named SCEP in order to evaluate its capabilities to handle the shared resources scheduling problem in complex systems.

3. SCEP MULTI-AGENT MODEL

3.1 Description of model

The SCEP multi-agent model (Fig. 1) is briefly a model developed for all types of planning activities, which introduces an indirect cooperation between two communities of agents (customer agents called C and producer agents called P), leading to a high level of co-operation. Each customer agent manages one order from the customers; each producer agent manages one resource (machine, raw material or human) of the organization. The cooperation between customer agents and producer agents is performed synchronically through the background environment agent E. All the activities are controlled by the supervisor agent S (Archimede et al., 2001). The detail working procedures and dynamic of the model will be introduced in next section.

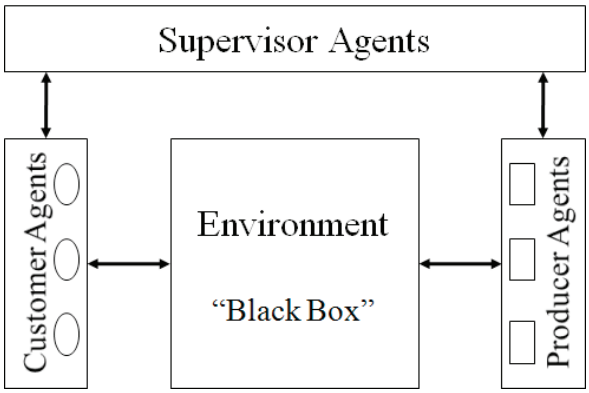

Fig. 1. SCEP model

3.2 Dynamic of model

Each object in the environment is associated with one operation to be achieved in one customer order. The set of objects are related to the routing followed by the intervention domain of concerned agents. In perfect correlation with the model definition, each operation only concerns one customer agent. But some objects can belong to the intervention domains of several producer agents, because multi machines may achieve the same activity. The position format of object O is [(S, F), N], where (S, F) represents a continuous temporal interval between a starting date S and a final date F, and N represents the name of resource executing object O. Each object has four positions, wished position (WP), effective position (EP), potential position (PP), and confirmed position (CP). The WP is the position requested by the customer. The EP results from the scheduling of all the tasks associated with the propositions collected from the environment. The PP results from the scheduling of one task associated with a proposition collected from the environment. The CP is the final position after all the scheduling process.

The supervisor agent provides functions of creating the agent society, generating the inside objects and initializing the environment. Then, the supervisor agent triggers the cycle of cooperation process by activating the customer agents and telling the producer agents to wait. The customer agents firstly ask for EP and PP of the associated objects from the environment. The environment sends the results back, of course the result is null in the first cycle. The customer agents schedule the operations which have not been validated, and influence the associated objects by alterative WP. If the WP of one object is the same as the EP and PP, customer agents will make the confirmation. At last, the customer agents send CP and WP of the associated objects to the environment. Each customer agent performs its actions simultaneously but remains independently from others. It will inform the supervisor agent once its actions are finished.

Once the end of the action from the last customer agent has been recorded by the environment, the supervisor agent activates the producer agents and sends the wait signal to the customer agents. The producer agents firstly ask for the CP and WP of the objects belonging to its intervention domain

from the environment. The environment sends the results back; the producer agents record the CP and schedule the tasks which are not definitely positioned. They influence these objects by alterative EP and PP to the environment. Each producer agent performs its actions independently and informs the supervisor agent as soon as its activities finished. When the end of the action from the last producer agent is recorded, the supervisor agent finishes the first cycle of the cooperation and starts the next cycle immediately. In each cycle (except the first one), at least one object should be confirmed to avoid the deadlock problem (Fig. 2).

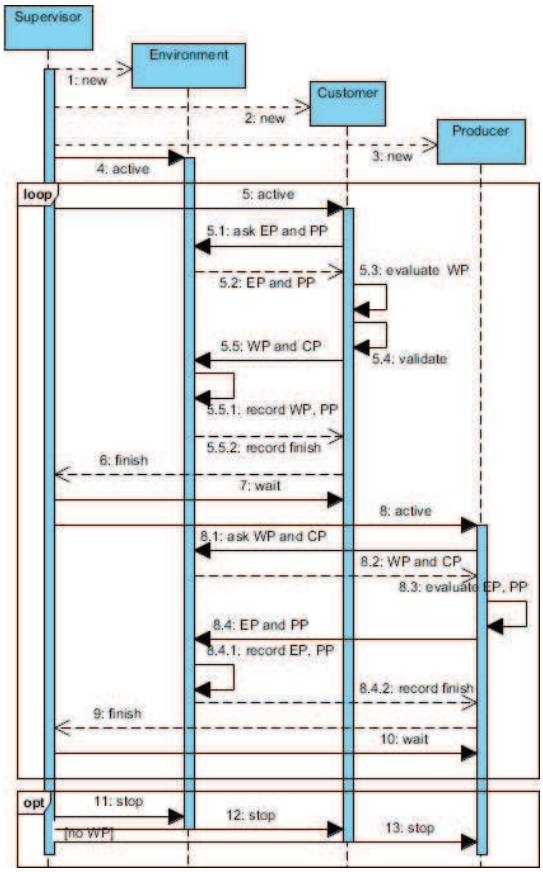

Fig. 2. Sequence diagram of SCEP model

The alternation cycle between the activation of customer agents and producer agents will repeated until the CP of all the environmental objects is fixed. When entire objects are confirmed, there are no WP from customer agents anymore. The alternative (opt) area will be executed and the supervisor agent will terminate the environment, customer and producer agents. The whole scheduling process is finished.

The SCEP model has been used for the production scheduling and maintenance scheduling. In SCEP model, the customer agents share resources managed by various producer agents. However, it only works with the resources/orders managed by producer/customer agents in the same site. In order to share resources located in remote sites, an improved SCEP model has been developed (Xu et al., 2011). This model showed its adaptation to the distributed management of multi-site orders. Although the SCEP model offers to solve the distributed scheduling problem, it only enables resources sharing between orders from the same site. As extension, we propose a DSCEP framework to achieve multi-site and shared resources scheduling between different (both economic and geographical) organizations.

4. DSCEP FRAMEWORK FOR SHARED RESOURCES SCHEDULING MANAGEMENT

4.1 Improvement of SCEP model

In order to fit the requirements of shared resources scheduling, we extend the SCEP model with virtual customer agent (VC) and virtual producer agent (VP). Each virtual customer agent manages entire orders from another SCEP model and basic customer agents manage entire orders from the local one. Each virtual producer agent manages resources from another SCEP model and basic producer agents manage entire resources of the local one (Fig. 3).

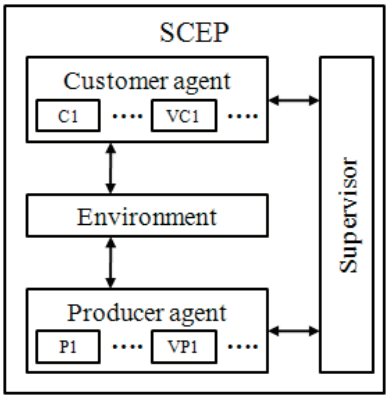

Fig. 3. Improved SCEP Model

4.2 Description of DSCEP framework

We propose the DSCEP framework to synchronize and control the use of improved SCEP models in order to elaborate or adapt a schedule involving shared resources. The whole framework is composed by three kinds of elements: improved SCEP models, shared resources register, and master supervisor. The communications between these elements are made through the communication bus in the framework (Fig. 4).

We classify the improved SCEP models into three categories based on the following rules. Root SCEP (RS) are improved SCEP models which do not manage shared resources but require shared resources from others. On the opposite side, leaf SCEP (LS) are improved SCEP models which provide shared resources but do not require from others. The third category is internal SCEP (IS); these improved SCEP models not only manage shared resources itself but also require shared resources from others. The RS only has several virtual producer agents, the LS only has several virtual customer agents, and the IS have both of them. The virtual customer agents and virtual producer agents should be one-one correspondence in the whole framework.

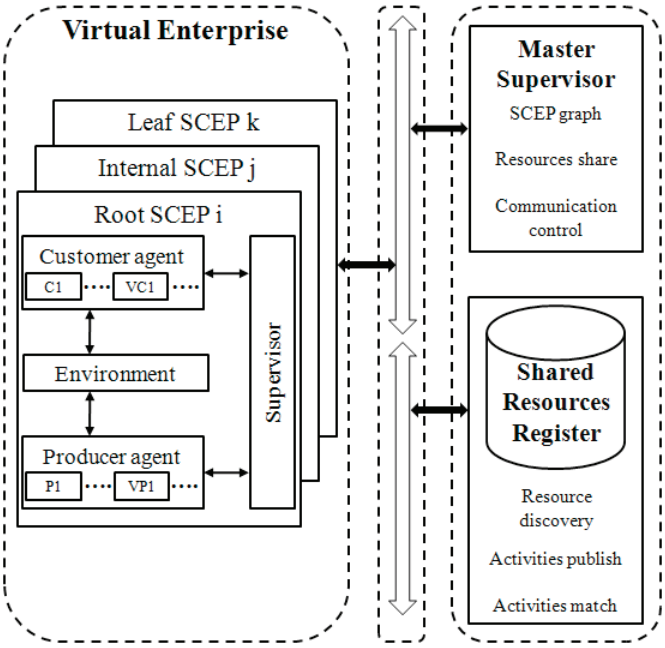

Fig. 4. DSCEP framework

The shared resources register is a database which records all the public activities provided by shared resources. It uses an ontology mechanism to match the activities requirements from improved SCEP models with the published activities recorded in the register.

The master supervisor is a controller which records the existing of entire SCEP models and the connection information of them. It divides SCEP models into three categories based on the ordered graph technology (Dechter et al., 2003). It also manages all the communication activities between SCEP models and shared resources register.

4.3 Dynamic of DSCEP framework

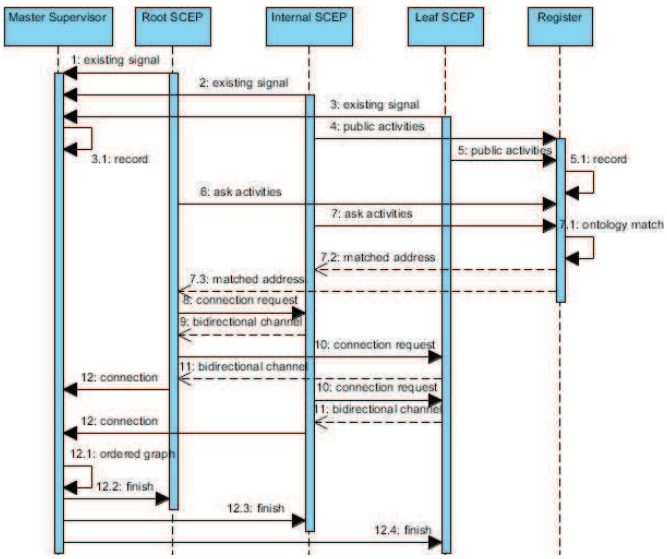

Fig. 5. Sequence diagram of DSCEP scheduling step 1

Each enterprise in the virtual enterprise creates an improved SCEP model based on the rules we introduce in the previous section. Then, all SCEP models send an existing signal to the master supervisor. LS and IS models publish the public activities provided by shared resources to the shared resources register. RS and IS models call register to get the address of the corresponding LS/IS models. In order to identify these addresses, the register achieves matching between required and recorded activities by an ontology mechanism, and sends the address back. Then the RS/IS models send the connection requests to the corresponding LS/IS models which have shared resources. A peer to peer bidirectional communication channel will be established between one virtual producer agent and one virtual customer agent for each couple (A and B) where A is an RS/IS requiring public activities and B is an LS/IS providing these activities. After the channel is build, RS/IS models send connection information to the master supervisor (Fig. 5).

The master supervisor builds and maintains an ordered graph for entire improved SCEP models, in order to control and synchronize the global scheduling process. In this graph each node is associated with an improved SCEP model, each directed segment is associated with an unidirectional invoking of shared resource. All nodes on rank 0 should be RS models and all nodes on the last rank n should be LS models. The nodes on rank m ($0<m<n$) are IS models. We also give the definition of sub-tree, the sub-tree of node x in rank i is a set of nodes in rank j ($j>i$) which contains all the shared resources required by x. For example, $\{IS_1, IS_2, IS_3, LS_n\}$ is the sub-tree of node RS_2 (Fig. 6).

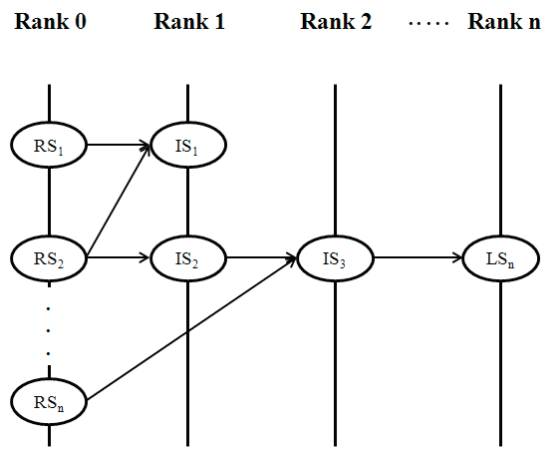

Fig. 6. Ordered graph for DSCEP framework

The orders defined in node x can exploit all the shared resources located in the nodes which belong to the sub-tree of node x. No matter in which rank, the scheduling process of an improved SCEP model x will be achieved in finite number of cycles, as we described in the section 3. In each cycle, a complete scheduling will be achieved for all the improved SCEP models in the sub-tree of x. These schedules may be partially cancelled at new cycle. The scheduling process will be finished when all orders in parent node x are scheduled. The global scheduling is achieved periodically. In that case, the scheduling process will be launched for all nodes in rank 0 at the same time. When node y detects a perturbation (receives new orders), a partial scheduling will be launched only for y and nodes belonging to the sub-tree of y (Fig. 7).

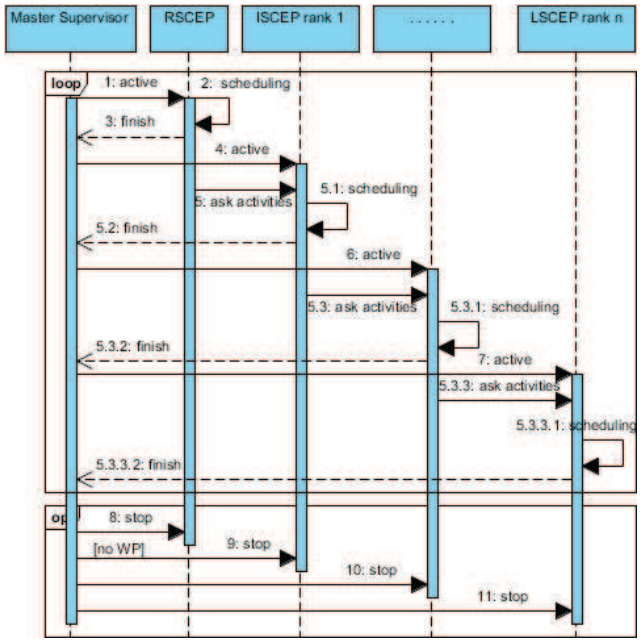

Fig. 7. Sequence diagram of DSCEP scheduling step 2

5. CASE STUDY OF MANUFACTURING SYSTEM

5.1 Case study description and modelization

In this case study, there are three manufacturing departments (A, B and C) in a virtual enterprise which have five resources (A1, A2, B1, B2 and C1). These resources can achieve several activities like cutting, assembling, painting, and gold plating (GP), etc. Since the GP machine located in department C is very expensive, all the departments use it as shared resource.

In order to keep this case simple and understandable, we assume that there are no transport time for products between different machines (departments). For the resources, no set-up time and closure time are considered. Once an operation has been started on a resource, it will be finished on the same one. The resource only has three possible states: available, in processing, or in failure after a breakdown.

Table 1. Resources in all departments

Resource	Rule	Activity	Capability	Cost
A1	FIFO	Cutting	1	1
A2	FIFO	Assembling	1	1
B1	FIFO	Cutting	1	1
B2	FIFO	Painting	1	1
		Assembling	1.5	1.5
C1	FIFO	GP	1	1

The detail of resources in these three departments can be found in Table 1. Each resource can achieve several activities with different capabilities and costs. For example, the activity of assembling for one product can be finished by machine A2 in 1 day with a cost of 1; by machine B2 in 1.5 days with a cost of 1.5. We also suppose that the dispatching rule used for resource management is FIFO (first in first out).

In each department there are several orders from customers, named manufacturing orders (MO). The detail characteristics of all MO are given in Table 2.

Table 2. Manufacturing orders in all departments

Order	Objective	Quantity	Order date	Due date	Routing
MOA1	delay	1	1	8	2
MOA2	delay	1	2	10	1
MOB1	delay	1	2	8	2
MOB2	delay	1	3	11	3
MOC1	delay	1	2	4	4
MOC2	delay	1	4	6	4

We use Gantt diagram to give an intuitive description of all the MO in all departments (Fig. 8).

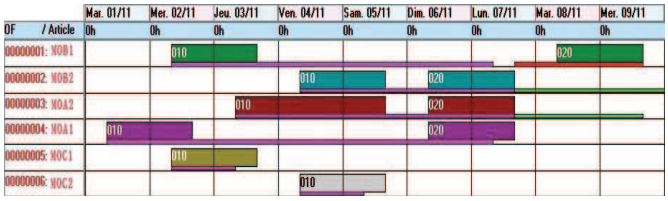

Fig. 8. Gant diagram for manufacturing orders

Manufacturing orders follow the linear routings defined in Table 3. The operating times are defined by the most capable resource.

Table 3. Routing

Routing	Operation	Activity	Operation time
1	1	Cutting	3
	2	Assembling	2
2	1	Cutting	2
	2	GP	2
3	1	Cutting	2
	2	Painting	2
4	1	GP	2

This case study requires negotiation between two RS models associated with department A, B and LS model associated with department C for the shared resource scheduling. The virtual producer agents for GP machine $VPGP_1$ and $VPGP_2$ which is expanded in RS models are connected to two virtual customer agents $VCGP_A$ and $VCGP_B$ which are expanded in LS model C.

5.2 Case study functioning

$VPGP_1$ and $VPGP_2$ send the WP of object MOA1.2 "([3, 5], 0)" and MOB1.2 "([4, 6], 0)" to $VCGP_A$ and $VCGP_B$. $VCGP_A$ and $VCGP_B$ send these positions to the producer agent PGP. The local customer agents in LS model C send the WP of object MOC1.1 "([2, 4], 0)" and MOC2.1"([4, 6], 0)", to PGP. PGP finds a conflict here. Based on the FIFO rule it schedules the orders and sends the EP of these four objects back: MOA1.2 ([4, 6], C) to RS model A, MOB1.2 ([8, 10], C) to RS model B, MOC1.1 ([2, 4], C), and MOC2.1

([6, 8], C) to local customer agents (Fig. 9).

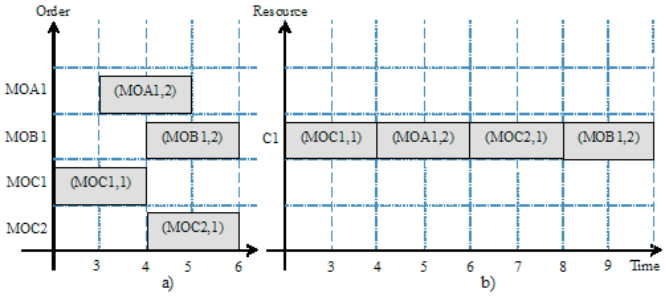

Fig. 9. Scheduling for shared resource

After all the scheduling process is finished, we can see the Gantt result (Fig. 10).

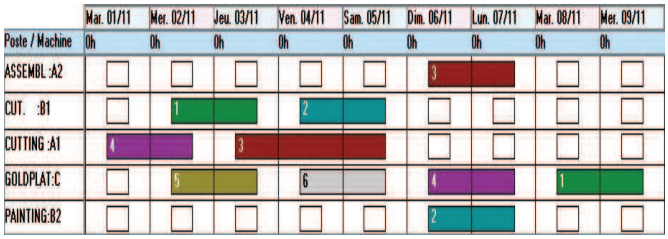

Fig. 10. Result for all orders

6. CONCLUSIONS

In this communication, we introduce the DSCEP framework solve the interoperability problem between different partners in virtual enterprise with ontology mechanism. It also use ordered graph to manage the rescheduling process for the new received orders. In order to solve conflicts during the shared resources scheduling process, DSCEP framework uses the negotiation between virtual producer agents and virtual customer agents. At last, we adopt a simple example to illustrate that the DSCEP framework could help multiple users to schedule their local resources and also support sharing resources scheduling. The efficiency of the SCEP model has been proved by abundant instances (Archimede rt al., 2001), we extend it to the DSCEP framework.

Indeed, there are many hypotheses in our framework and illustration example. For the DSCEP framework, the priority of the manufacturing orders is not defined; the scheduling rule for the resources is limited to FIFO. For the illustrated example, the restrictions during manufacturing process such as transport time, set-up time and closure time are not taking into account. The disturbances such as machine break down and emergency orders are set to low. In the future we will continue to evaluate the scheduling behavior of DSCEP framework with more realistic manufacturing scenarios. We will also develop an automatic software application based on DSCEP framework.

REFERENCES

Archimede, B. and Coudert, T. (2001). Reactive scheduling using a multi-agent model: the SCEP framework. *Engineering Application of Artificial Intelligence*, 14, 667-683.

Azadegan, A., Porobic, L., Ghazinoory, S., Samouei, P., and Kheirkhah, A.S. (2011). Fuzzy logic in manufacturing: A review of literature and a specialized application. *International Journal of Production Economics*, 132 (2), 258-270.

Chan, F., Chan, H., and Kazerooni, A. (2003). Real time fuzzy scheduling rules in FMS. *Journal of Intelligent Manufacturing*, 14 (3–4), 341–350.

Dechter, R. (2003). *Constraint Processing.* Morgan Kaufmann, San Francisco.

Galvin, S. (1994). Operating system concepts. *Distributed file systems*, 17. Addison-Wesley Publishing Company.

Gendreau, M. and Potvin J.Y. (2010). Handbook of Metaheuristics. *International Series in Operations Research & Management Science*, 146, 41-59.

Goncalvesa, J.F., Mendesb, J.J.M., and Resende, M.G.C. (2008). A genetic algorithm for the resource constrained multi-order scheduling problem. *European Journal of Operational Research*, 189 (3), 1171-1190.

Kornienko, S., Kornienko, O., and Priese, J. (2004). Application of multi-agent planning to the assignment problem. *Computers in Industry*, 54 (3), 273-290.

Korsvik, J.E., Fagerholt, K. and Laporte G. (2010). A tabu search heuristic for ship routing and scheduling. *Journal of the Operational Research Society*, 61, 594-630.

Le, S. and Boyd, R. (2007) Evolutionary Dynamics of the Continuous Iterated Prisoner's Dilemma. *Journal of Theoretical Biology*, 245, 258–267.

Molina, A. and Sanchez, J.M. (1998). The virtual enterprise. *Handbook of life cycle engineering: concepts, models, and technologies*, 3, 59–89.

Ouelhadj, D. and Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12 (4), 417-431.

Restrepo, I. and Balakrishnan, S. (2008). Fuzzy-based methodology for multi-objective scheduling in a robot-centered flexible manufacturing cell. *Journal of Intelligent Manufacturing*, 19 (4), 421–432.

Shao, X. Li, X. Gao, L. and Zhang, C. (2009). Integration of process planning and scheduling: A modified genetic algorithm-based approach. Computer & operations Research, 36 (6), 2082-2096.

Shen, W. (2002). Distributed manufacturing scheduling using intelligent Agents. *Intelligent Systems*, 17(1), 88-94.

Shen, W., Wang, L., and Hao, Q. (2006). Agent-based distributed manufacturing process planning and scheduling: A state-of-the-art survey. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 36 (4), 563-577.

Sivanandam, S.N. and Deepa, S.N. (2007). *Introduction to Genetic Algorithms*, 2, 15-39. Springer Verlag.

Xu, J., Archimede, B., and Letouzey, A. (2011). Application of using SCEP model for distributed scheduling with shared resources in hospital system. *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on*.

Zweben, M. and Fox, M.S. (1994). ISIS: A Retrospective. *Intelligent Scheduling*, 3-28. CA: Morgan Kaufmann, San Francisco.