To link to this article: DOI:10.1007/s10845-013-0748-8
http://dx.doi.org/10.1007/s10845-013-0748-8

Towards a distributed multi-agent framework for shared resources scheduling

Bernard Archimede · Agnes Letouzey · Muhammad Ali Memon · Jiucheng Xu

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 this communication, we introduce the architecture and behaviour 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 · Distributed architectures

Introduction

Shared resources are 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. 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. Shared resources can be found in manufacturing domain, when SMEs (Small and Medium Enterprises) share the cost of a very expensive machine like a laser cutting machine, in hospital system, when different services share the use of the image service, particularly the use the MRI (Magnetic Resonance Imaging) machine, or in transportation problems, when transport companies share the purchase and the use of a loading crane. 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). In practice, scheduling a resource shared by different partners can be achieved by different ways:

1. The shared resource is scheduled by each partner in its own schedule, like the other resources. This will lead to possible conflicts between partners for the use of the shared resource.
2. The schedules of all the partners are built altogether by a single scheduling tool. This is not a satisfying solution as the different partners share a resource, not all their resources and data.
3. The schedule of the shared resource is built independently, with one or more time period booked for each partner. The partners use the shared resource only during their allocated times. The obtained schedules are very sensitive to perturbations. The use rate of the shared resource can be very low, because of cancellations.

JiuCheng Xu passed away in a car accident on June 8th 2012, at the age of 29. This article is based on his PhD work and is dedicated to him.

B. Archimede ( ) · A. Letouzey · M.A. Memon · J. Xu
University of Toulouse, INP-ENIT, 47 Avenue d'Azereix, BP 1629, 65016 Tarbes Cedex, France
e-mail: bernard.archimede@enit.fr

The building of virtual enterprises can encourage organisations to share their resources with partners (Molina and Sanchez 1998). Distributed scheduling appears to be able to fit the requirements of shared resources scheduling. In this communication, we will focus on the shared resources scheduling problems in complex systems which adopt distributed scheduling approach. This paper is organized as following: "Summary of scheduling techniques" section reviews the different scheduling technologies and discusses their limitation. "SCEP multi-agent model" section 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 "DSCEP framework for shared resources scheduling" section. "Case study in manufacturing system" section describes the scheduling process using the DSCEP framework particularly focus on a manufacturing system case study. A brief conclusion and perspectives are stated in "Conclusion" section.

Summary of scheduling techniques

In computer science, scheduling is the method by which threads, processes or data flows are given access to system resources. This is usually done to effectively balance the load of a system or achieve a target quality of service (Blazewicz et al. 2001). In manufacturing area, production scheduling is defined as "establishing the timing for performing a task" and observes that, in manufacturing firms, there are multiple types of scheduling, including the detailed scheduling of a shop order that shows the start and complete point of each operation (Wight 1984). Scheduling is also defined as the process of assigning manufacturing resources over time to the set of manufacturing processes in the process plan (Shen et al. 2006). It determines the most appropriate time to execute each operation, taking into account the temporal relationships between manufacturing processes and the capacity limitations of the manufacturing resources. The assignments also affect the optimality of a schedule with respect to criteria such as cost, tardiness, or throughput. In brief, manufacturing scheduling is an optimization process where limited resources are allocated over time among both parallel and sequential activities (Zweben and Fox 1994).

Traditional approaches for job shop scheduling

Because of its highly combinatorial aspect (NP-complete) (Zweben and Fox 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 and Balakrishnan 2008), and apply to scheduling rules in flexible manufacturing systems by evaluating multiple performance measures (Chan et al. 2003).

Genetic Algorithms (GAs) are an example of mathematical technology transfer: by simulating evolution one can resolve complicated optimization problems from a variety of sources (Sivanandam and Deepa 2007). Today, GAs are used to facilitate the integration and optimization of the process planning and scheduling in manufacturing area (Shao et al. 2009). And they are also used to solve the resource constrained multi-order scheduling problem (Goncalvesa et al. 2008). Tabu search Gendreau and Potvin (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). 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 and Petrovic 2009). The traditional scheduling methods encounter great difficulties when they are applied to real-world situations, since they use simplified theoretical models. When one of these approaches is used in a scheduling software, all computations are carried out in a central computing unit. If good results can be obtained with these methods, their only ways to take into account a shared resource in a schedule is either making the schedules of each partners separately, the shared resources scheduled like the other resources, leading to conflicts between partners for the use of the shared resources, or building all the schedules of the partners in one scheduling tool, leading to a lack of confidentiality for the partners. Thus, these approaches cannot respond to the problem of scheduling shared resources.

Scheduling techniques with multi-agent systems

Multi-agent systems (MAS) are the sub-field of Distributed Artificial Intelligence (DAI) which has experienced rapid growth since the available flexibility and intelligence could solve distributed problems (Balaji and Srinicasan 2010). The multi-agent approaches can cope with conflict situations with negotiation technologies, in which the compromises can moderate the satisfaction and frustrations of the agents. Multi-agent technologies have been combined with meta-heuristics in order to achieve optimisation and compromises (Passos et al. 2010). For the dynamic scheduling and shop floor job assignment problem, a real-world manufacturing system in a multi-agent system has been represented, and furthermore improved the global performance by introducing Ant Colony Intelligence (ACI) into agent coordination and negotiation (Xiang and Lee 2008). A distributed multi-agent scheduling system (MASS) based on co-operative approach is proposed to solve static and dynamic job shop scheduling problems (JSSP) (Kouider and Bouzouia 2012). This system is composed of two kinds of agents, Supervisor agents and Resource agents. The Supervisor agent decomposes JSSP into interrelated sub-problems and the Resource agents co-operate, through a distributed approach of local idle time minimization.

Two Multi-Agent approaches based on the Tabu Search (TS) meta-heuristic have been proposed in Ennigrou and Ghedira (2008). Depending on the location of the optimization core in the system, they have distinguished between the global optimization approach where the TS has a global view on the system and the local optimization approach (FJS MATSLO) where the optimization is distributed among a collection of agents, each of them having its own local view. A multi-agents approach to solve job shop scheduling problem using meta-heuristics is presented by Passos et al. (2010). Meta-heuristics approaches when solving scheduling problems have proven to be very effective and useful in practical situations. TS and Genetic Algorithms (GA) have been used to solve optimization problems with success. This approach combining these algorithms brings new perspective to solve this kind of problem. Another multi-agent architecture of an integrated and dynamic system is also developed for process planning and scheduling of multiple jobs. A negotiation protocol is discussed to generate the process plans and the schedules of the manufacturing resources and the individual jobs, dynamically and incrementally, based on the alternative manufacturing processes (Nejad et al. 2011). From the methods mentioned in previous section, agent-based approaches have several potential advantages for distributed manufacturing scheduling (Shen et al. 2006):

- They use parallel computation through a large number of processors, which may provide scheduling systems with high efficiency and robustness.
- They can facilitate the integration of manufacturing process planning and scheduling.
- They make it possible for individual resources to trade off local performance to improve global performance, leading to cooperative scheduling.
- Resource agents may be connected directly to physical devices they represent so as to realize real-time dynamic rescheduling.
- Schedules are achieved by using mechanisms similar to those being used in manufacturing supply chains.

Compared 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 approaches, combined with meta-heuristics or not, revealed to achieve distributed scheduling (Passos et al. 2010). Distributed scheduling answers to some of the requirements of shared resources scheduling:

- Having a schedule for each partner,
- Communication between the partners schedules,
- Preserving partners confidentiality by allowing them to collaborate without risk of disclosure of sensitive information of their organization, methods and data.

None of the methods mentioned above seem to be able to take shared resources into account during the scheduling process. 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.

SCEP multi-agent model

Description of model

The SCEP multi-agent model (Fig. 1) has been used in different context as planning of activities such as production, maintenance and transport. It introduces an indirect cooperation between two communities of agents (customer agents called C and producer agents called P). Each customer agent manages one order from the customers; each producer agent manages one resource (machine 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 and Coudert 2001). The detail working procedures and dynamic of the model will be introduced in next section.

Dynamic of model

Each customer agent manages a project composed of a set of several operations. Each operation requires an activity performed by one ore more producer agent. 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 or zero if the name of the resource is not known. 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

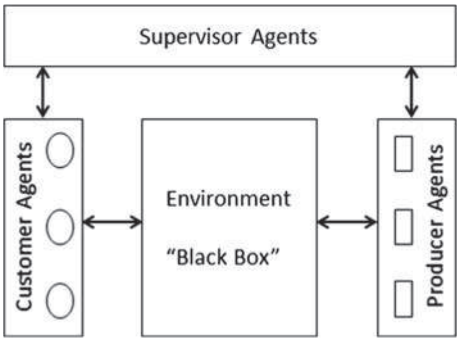

Fig. 1 SCEP model

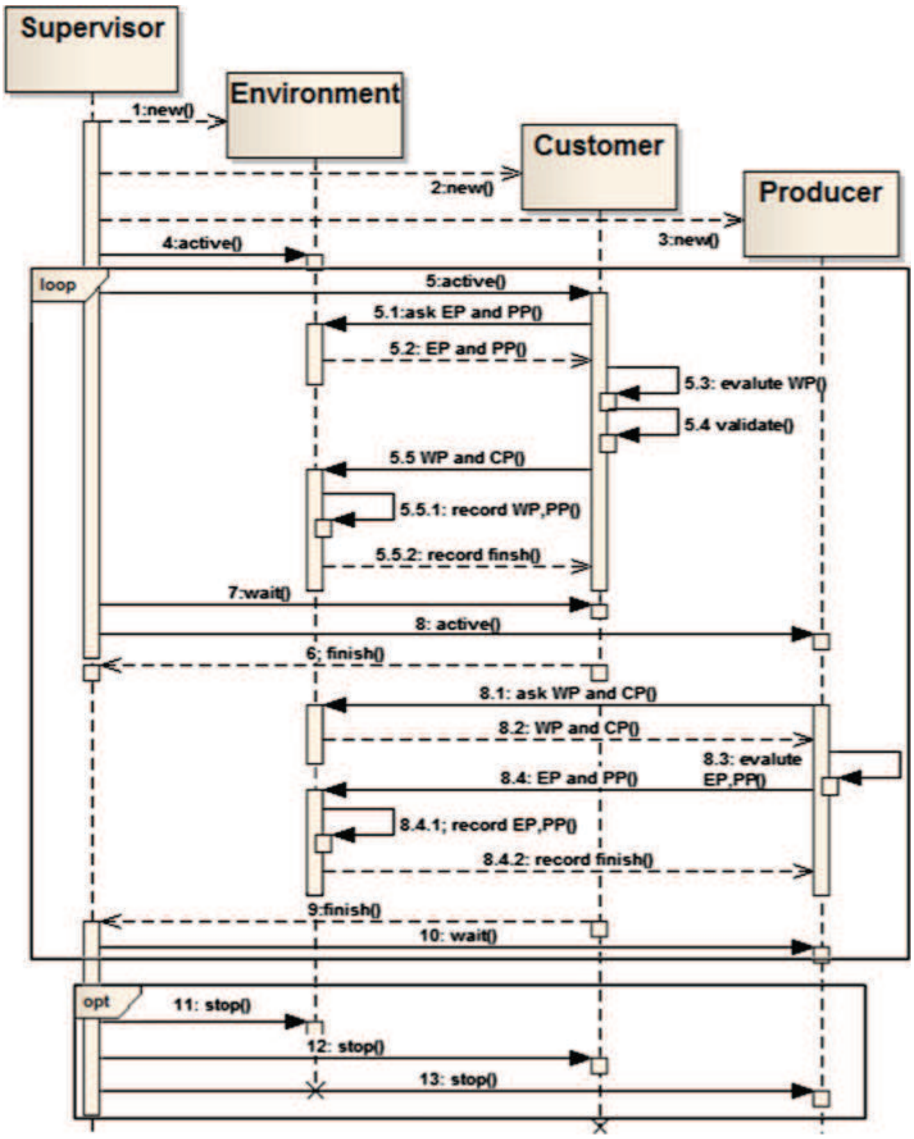

Fig. 2 Sequence chart

scheduling of one task associated with a proposition collected from the environment. The CP is the final position after all the scheduling process which is based on an auction mechanism close to the contract-net protocol (Zhang and Cao 2012).

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 best PP of the same object, customer agents will make the confirmation for this object. On the contrary, the WP will become the best EP. 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 independent from others. It informs 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). The convergence of the method and the deadlock problems have been studied in Archimede and Coudert (2001).

The alternation cycle between the activation of customer agents and producer agents will be 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. In a reactive mode, when the system has to take a disruption into account, naturally only the influence area of this disruption is rescheduled, in order to maintain a certain stability from a schedule to the new one. In SCEP model, the customer agents naturally share resources managed by various producer agents. However, these resources must be located in the same site and the orders must be associated to

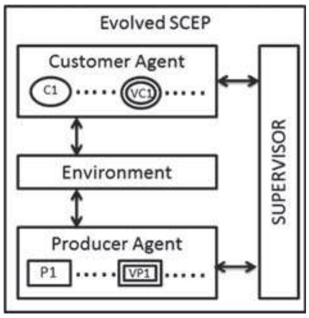

Fig. 3 E-SCEP model

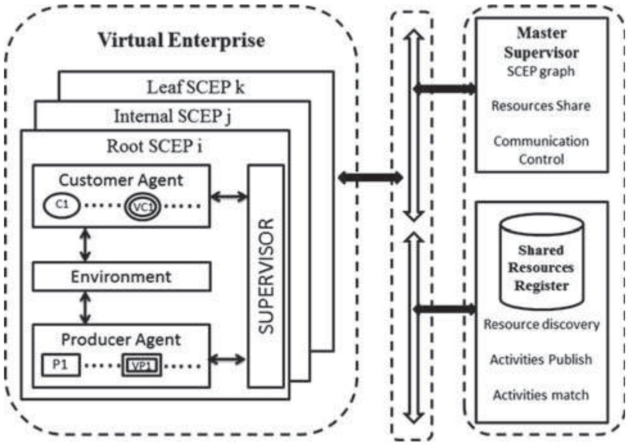

Fig. 4 DSCEP framework

projects defined also in that site. In order to share resources located in remote sites, an improved SCEP model has been developed (Xu et al. 2011). This has been achieved by introducing new concepts of ambassador SCEP and ambassador agent. The particularity of these ambassadors is to get across the boundary of the SCEP models and establish a communication bridge between a SCEP server and a SCEP client.

In this context, an improved network allows establishment of multi-site plans by cooperation between one SCEP client and multiple SCEP servers and supplies a support for distributed scheduling. The number of ambassador agents in SCEP client is equal to the number of the SCEP servers. The ambassador agent in SCEP client gets information from SCEP client environment about the demands (WP), communicates with associated SCEP server ambassador. As soon as the SCEP servers finished their actions, the associated ambassador agents inform the SCEP client that actions are achieved. In practice, the implementation of the communication between ambassadors SCEP and agent may be realized with the bus CORBA, DCOM and .Net. 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

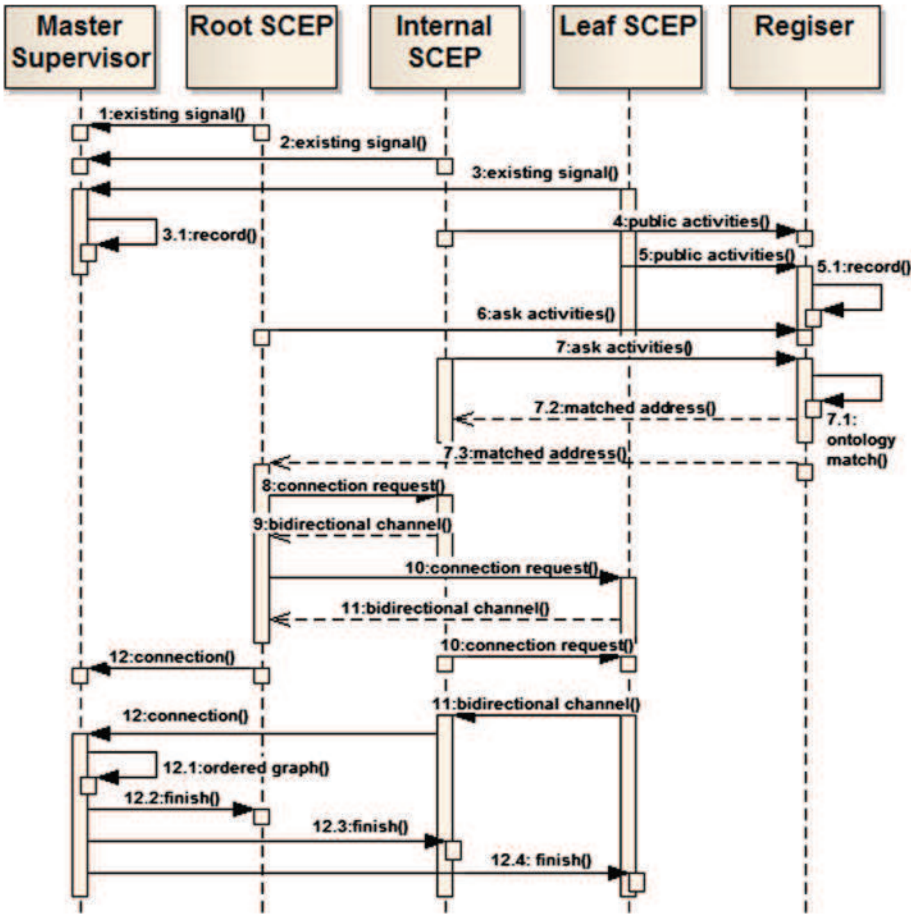

Fig. 5 DSCEP sequence chart

propose a DSCEP framework to achieve multi-site and shared resources scheduling between different (both economic and geographical) organizations.

DSCEP framework for shared resources scheduling

Evolution 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).

Description of DSCEP framework

We propose the DSCEP framework to synchronize and control the use of evolved SCEP models in order to elaborate or adapt a distributed schedule involving shared resources

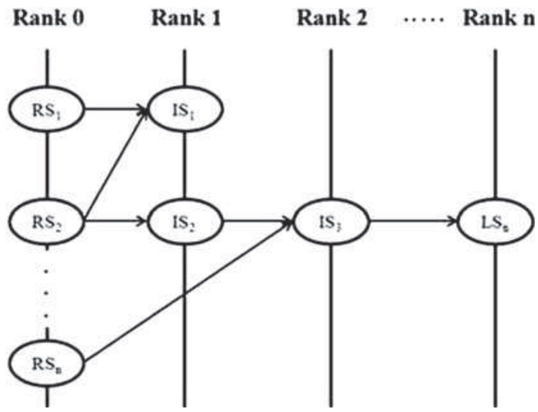

Fig. 6 Ordered graph for DSCEP framework

of an enterprises network. The whole framework is composed by three kinds of elements: evolved 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 evolved SCEP models into three categories based on the following rules. Root SCEP (RS) are evolved SCEP models which do not manage shared resources but require shared resources from others. On the opposite side,

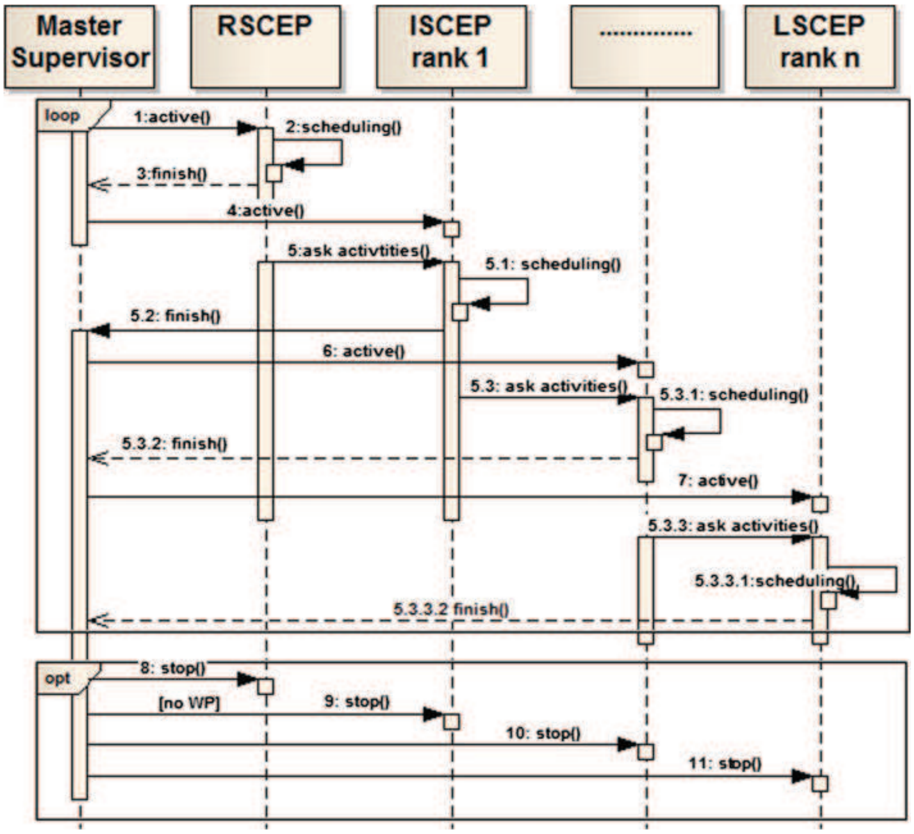

Fig. 7 Sequence diagram of DSCEP scheduling step

Table 1 Resources in all departments

Resource	Rule	Activity	Capability	Cost	Dep
M1	FIFO	Cutting	1	1	A
M2	FIFO	Assembling	1	1	A
M3	FIFO	Cutting	1	1	B
M4	FIFO	Painting	1	1	B
		Assembling	1.5	5	B
GP	FIFO	GoldPlating	1	1	C

Table 2 Manufacturing orders in all departments

Order	Objective	Quantity	Order date	Due date	Routing	Dep
MO1	Delay	1	1	8	2	A
MO2	Delay	1	2	10	1	A
MO3	Delay	1	2	8	2	B
MO4	Delay	1	3	11	3	B
MO5	Delay	1	2	4	4	C
MO6	Delay	1	4	6	4	C

Leaf SCEP (LS) are evolved SCEP models which provide shared resources but do not require from others. The third category is Internal SCEP (IS); these evolved 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. The shared resources register is a database which records all the public activities provided by shared resources coming from Leaf and Internal evolved SCEP models. To solve the interoperability problems between the semantic of activities used by the different SCEP models, an ontology mechanism is used to match the

Table 3 Routings

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

activities requirements from Root and Internal evolved SCEP models with the published activities recorded in the register. The master supervisor is a controller which records the existing SCEP models and the connection links between them. It creates and manages the ordered graph (Dechter 2003) of the three categories SCEP models. It also manages all the communication activities between SCEP models and shared resources register.

Dynamic of DSCEP framework

Each enterprise in the virtual enterprise creates an evolved 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 SCEP 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).

To prevent deadlock situations which could occur during the process scheduling, the master supervisor builds and maintains an ordered graph with no cycle for entire evolved SCEP models, in order to control and synchronize the global scheduling process. In this graph each node is associated with an evolved SCEP model, each directed segment is associated with an unidirectional invoking of shared resource. All nodes on the rank 0 should be RS models. LS and IS models are located on the other ranks. 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 or by successors of x. For example, IS1, IS2, IS3, LSn is the sub-tree of node RS2 (Fig. 6). 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. The ordered graph is used by the Master Supervisor periodically (depends on the production type may be one day or one week and so on) to elaborate a global scheduling for all the entreprise network, and partially when a perturbation occurs (receiving new urgent manufacturing orders, etc.).

For a global scheduling, the scheduling process will be launched at the same time for all nodes in rank 0 associated to Root SCEP models and concerns all the nodes (Fig 7). For a partial scheduling, the process is launched from the node where a perturbation has been detected and concerns all the nodes in its sub-tree. For example if a perturbation is detected on RS2 node, the concerned nodes in the partial scheduling process are RS2, IS1,IS2, IS3, LSn (Fig. 6). In reactive mode the internal functioning is detailed in Archimede and Coudert (2001). The master supervisor records the requests coming from new projects and related to the use of shared resources. If a scheduling is being developed, these projects are by default, taken into account for the next scheduling. If a new urgent project has requested to the use of shared resources, the master supervisor achieves an update of the ordered graph and a partial scheduling for the concerned sub-tree if no cycle is detected. The scheduling process of concerned evolved SCEP models x will be achieved in finite number of cycles, whatever the rank and the node, as we described in the "SCEP multi-agent model" section.

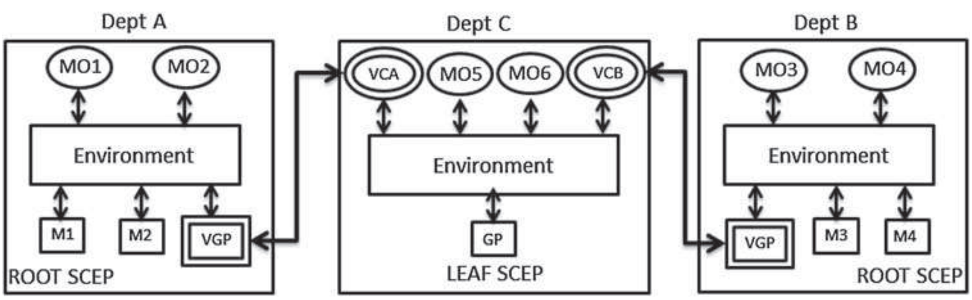

Fig. 8 DSCEP framework for example

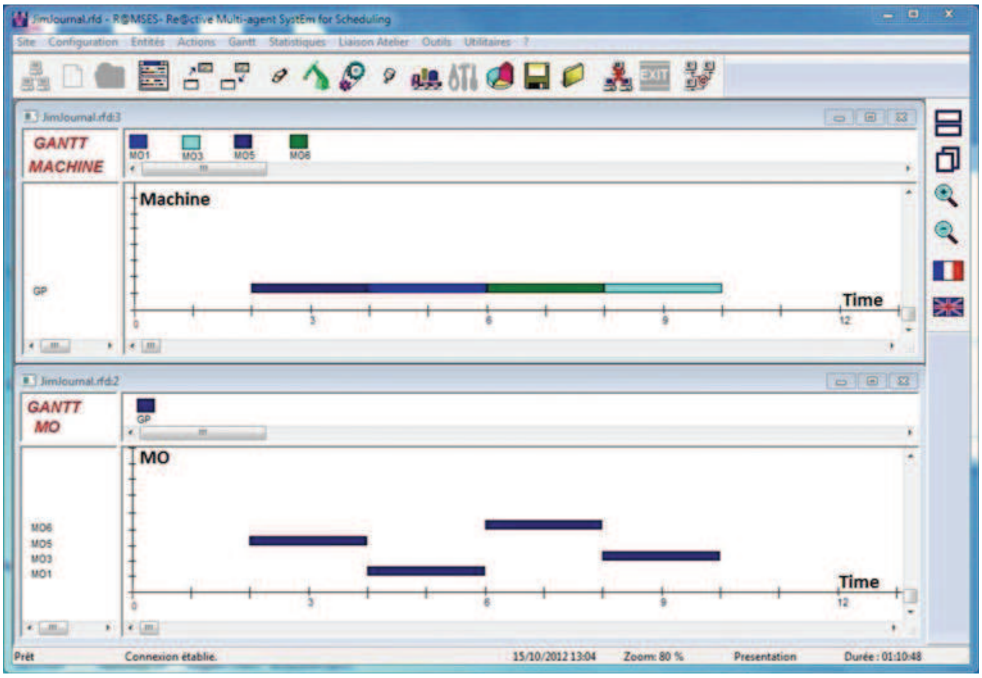

Fig. 9 Scheduling of shared resource in department C

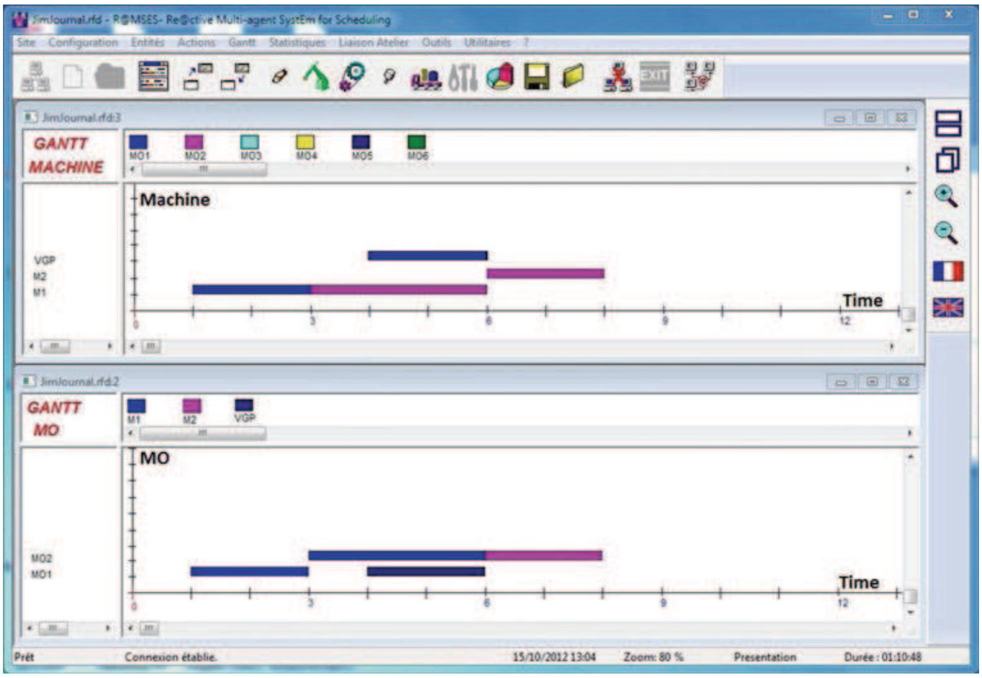

Fig. 10 Gantt diagrams in department A

For each cycle of the scheduling process of an evolved SCEP model *e* associated to node *x*, a complete scheduling will be achieved for all the evolved SCEP models associated to direct successors of *x* in its sub-tree. These elaborated schedules may be completely or partially cancelled at new cycle of *e*. The scheduling process will be finished when all orders in *e* are scheduled.

Case study in manufacturing system

Case study description and modeling

In this case study, there are three manufacturing departments (DepA, DepB and DepC) in a virtual enterprise which have five resources (M1, M2, M3, M4 and GP). These resources

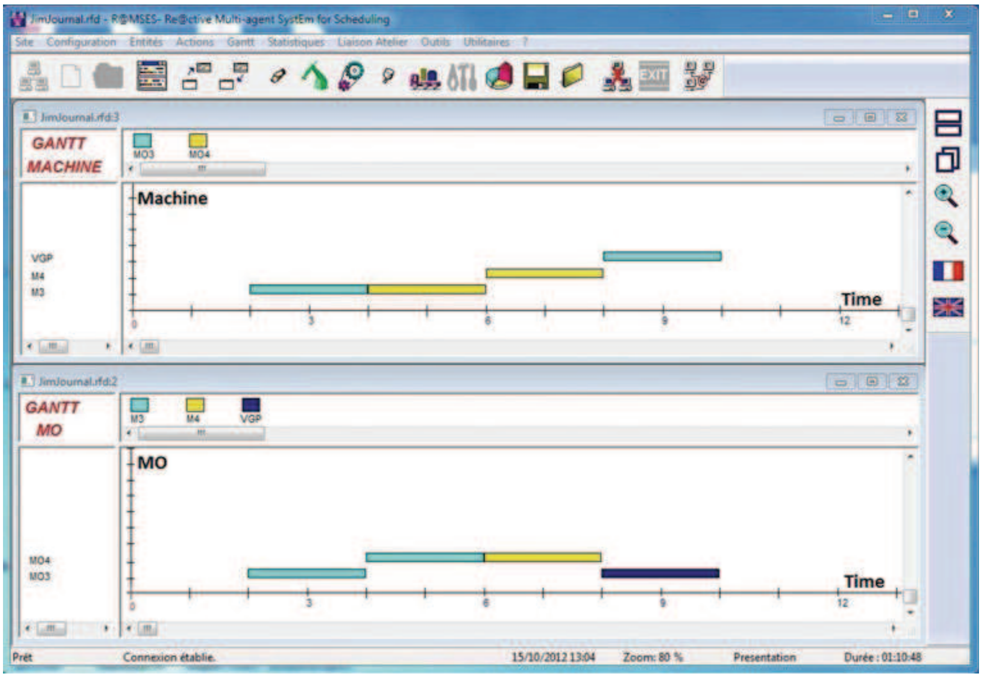

Fig. 11 Gantt diagrams in department B

can achieve several activities like cutting, assembling, painting, and gold plating, etc. M1 and M2 are located in department DepA. M3 and M4 are located in department DepB. Since the GP machine located in department DepC is very expensive, all the departments use it as a shared resource.

In order to keep this case simple and understandable, we assume that there are no transport time for products between different departments. For the resources, no setup 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 has only three possible states: available, in processing, or in failure after a breakdown. 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 M2 in 1 day with a cost of 1; by machine M4 in 1.5 days with a cost of 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. The objective sought for each MO (mentioned as delay in Table 2) is the respect of the due date. We use Gantt diagram to give an intuitive description of all the MO in all departments (Fig. 10). Manufacturing orders follow the linear routings defined in Table 3. The operating times are defined by the most capable resource. This case study requires negotiation between two RS models associated with department DepA, DepB and LS model associated with department DepC for the shared resource scheduling. The virtual producer agents for GP machine VGP of A and VGP of B which is expanded in RS models are connected to two virtual customer agents VCA and VCB which are expanded in LS model DepC (see Fig. 8).

Case study functioning

VGP of A and VGP of B send the WP of object MO1.2 ”([3, 5], 0)” and MO3.2 ”([4, 6], 0)” to VCA and VCB. VCA and VCB send these positions to the producer agent GP. The local customer agents in LS model DepC send the WP of object MO5.1 ”([2, 4], 0)” and MO6.1”([4, 6], 0)”, to GP. GP finds a conflict here. Based on the FIFO rule it schedules the orders and sends the EP of these four objects back: MO1.2 ([4, 6], DepC) to RS model DepA, MO3.2 ([8, 10], DepC) to RS model DepB, MO5.1 ([2, 4], DepC), and MO6.1 ([6, 8], DepC) to local customer agents (Fig. 9). The GP machine is occupied between 2 to 4. Although the second operation of MO1 is ready and could started at date 3, it is not possible because the GP machine is not available (Fig. 10). After all the scheduling process is finished, we can see the Gantt result (Fig. 11).

Conclusion

In this paper, we introduce the DSCEP framework to solve the interoperability problem between different partners in virtual enterprise with ontology mechanism. We also use an 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 and Coudert 2001), we extend it to the DSCEP framework. Indeed, there are many hypotheses supposed for our framework and illustrated 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 taken into account. The disturbances such as machine break down and emergency orders are set to low. In future we will continue to evaluate the scheduling behavior of DSCEP framework with more realistic manufacturing scenarios.

References

Archimede, B., & 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., & 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.

Balaji, P. G., & Srinicasan, D. (2010). An introduction to multi-agent systems. *Studies in Computational Intelligence*, *310*, 1–27.

Blazewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., & Weglarz, J. (2001). *Scheduling computer and manufacturing processes*. Berlin: Springer.

Chan, F., Chan, H., & Kazerooni, A. (2003). Real time fuzzy scheduling rules in FMS. *Journal of Intelligent Manufacturing*, *14*(34), 341–350.

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

Ennigrou, M., & Ghedira, K. (2008). New local diversification techniques for flexible job shop scheduling problem with a multi-agent approach. *Autonomous Agents and Multi-Agent Systems*, *17*(2), 270–287.

Galvin, S. (1994). *Operating system concepts distributed file systems*. Reading: Addison-Wesley.

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

Goncalvesa, J. F., Mendesb, J. J. M., & 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., & Priese, J. (2004). Application of multi-agent planning to the assignment problem. *Computers in Industry*, *54*(3), 273–290.

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

Kouider, A., & Bouzouia, B. (2012). Multi-agent job shop scheduling system based on co-operative approach of idle time minimization. *International Journal of Production Research*, *50*(2), 409–424.

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

Nejad, H. T. N., Sugimura, N., & Iwamura, K. (2011). Agent-based dynamic integrated process planning scheduling in flexible manufacturing systems. *International Journal of Production Research*, *49*(5), 1373–1389.

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

Passos, C.A.S., Iha, V.M., & Dominiquini, R.B. (2010). *A multi-agents approach to solve job shop scheduling problems using metaheuristics, 5th conference on management and control of production and logistic*.

Restrepo, I., & 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., & Zhang, C. (2009). Integration of process planning and scheduling: A modified genetic algorithm-based approach. *Computer and 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., & 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., & Deepa, S. N. (2007). *Introduction to genetic algorithms*. Berlin: Springer.

Wight, O. W. (1984). *Production and inventory management in the computer age*. New York: Van Nostrand Reinhold.

Xiang, W., & Lee, H. P. (2008). Ant colony intelligence in multi-agent dynamic manufacturing scheduling. *Engineering Application of Artificial Intelligence*, *21*(1), 73–85.

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

Zhang, J., & Cao, Y. Q. (2012). Research on cooperation of multiple agent based on contract-net protocol. In *International conference on industrial control and electronics engineering (ICICEE)*.

Zweben, M., & Fox, M. S. (1994). *ISIS: A retrospective, intelligent scheduling* (pp. 3–28). San Francisco, CA: Morgan Kaufmann.