ELSEVIER

Computers in Industry 46 (2001) 189–207

COMPUTERS IN INDUSTRY

www.elsevier.com/locate/compind

A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources

J. Sun, D. Xue*

Department of Mechanical and Manufacturing Engineering, University of Calgary, Calgary, Alberta, Canada T2N 1N4

Received 24 July 2000; accepted 13 March 2001

Abstract

This research introduces a dynamic reactive production scheduling mechanism for modifying the originally created schedules when these schedules cannot be completed due to changes of production orders and manufacturing resources. Production order changes include removal of an order that is canceled by a customer and insertion of an order that has to be completed within a short period of time. Manufacturing resource changes include breakdowns of machines and sudden sickness of workers. Match-up and agent-based collaboration approaches are employed to modify only part of the originally created schedules for improving the reactive scheduling efficiency, while maintaining the scheduling quality. The dynamic reactive production scheduling system was implemented using Smalltalk, an object oriented programming language.

Keywords: Reactive scheduling; Predictive scheduling; Match-up approach; Multi-agents; Artificial intelligence

1. Introduction

Production scheduling aims at allocating available manufacturing resources for the required manufacturing tasks and identifying the sequence and timing parameter values to accomplish these tasks. Typical manufacturing resources include facilities, persons, materials, and so on. Competitiveness of products can be improved by identifying the optimal production schedule that needs the minimum production efforts.

The research on production scheduling was started by developing algorithms for generating the optimal sequence to complete the required tasks considering either only one processor (machine) or multiple processors (machines) [1]. In each of these scheduling methods, an objective function, such as the minimum total make-span to complete all the selected tasks, or the minimum mean flow of these selected tasks, is selected for identifying the optimal schedule.

Most of the earlier developed scheduling methods have difficulty for solving actual industrial problems, due to the complexity of real-life manufacturing constraints. First, the industrial scheduling problems are dynamic in nature, i.e. new orders are received continuously during the production process. Second, the created schedule may be changed to reflect the changes of production orders and manufacturing conditions during production process. Production order changes include removal of an order that is canceled by a customer and insertion of an order that has to be completed within a short period of time. Manufacturing condition changes include disturbance events of resources such as breakdowns of machines and sickness of workers.

* Corresponding author. Tel.: +1-403-220-4168;
fax: +1-403-282-8406.
E-mail address: xue@enme.ucalgary.ca (D. Xue).

0166-3615/01/$ – see front matter
PII: S0166-3615(01)00119-1

With the advances in computer technologies, many new methods and systems considering industrial constraints were developed for two different types of production scheduling: predictive scheduling and reactive scheduling [2,3]. Predictive scheduling creates the optimal schedule based on given requirements and constraints prior to the production process. Most of the scheduling algorithms and systems were developed for predictive scheduling. Reactive scheduling, on the other hand, is a process to modify the created schedule during the manufacturing process to adapt changes in production environment. Reactive scheduling is also called rescheduling.

The intelligent system approaches have been proved effective for conducting both predictive scheduling and reactive scheduling [2,4,5]. For predictive scheduling, generally intelligent approaches aim at identifying the optimal schedule through iterative search process. For reactive scheduling, most approaches attempt to revise only part of the originally created schedule for responding to the production environment changes without rescheduling all the required tasks [6–9]. Because a large number of tasks are usually considered in predictive scheduling and reactive scheduling, the optimal schedules created using these developed methods are not the true global optimal schedules. Quality of scheduling result can be improved by employing stochastic computing methods, such as genetic algorithm and simulated annealing, to prevent the result from falling into the local optimal points [2].

Despite the progress, many problems have to be solved for predictive scheduling and reactive scheduling. These problems are summarized as follows.

1. In the presently developed scheduling systems, manufacturing requirements are usually modeled directly based upon customer requirements. The manufacturing requirements, together with manufacturing resource descriptions, are used as constraints for production scheduling. Product design descriptions and constraints, however, are not considered in these systems. Because many new designs are created using existing modules as their components, modeling of the manufacturing requirements of these component modules and identification of the manufacturing tasks of these designs by considering constraints among these components are required.
2. The production scheduling mechanisms in these systems were primarily developed based on centralized computing architecture, in which all the knowledge bases and databases were modeled at the same location. This control architecture has difficulty in handling complex manufacturing systems that require knowledge and data to be distributed at different locations. Therefore, development of distributed production scheduling systems is required.

In our previous research, an intelligent predictive scheduling system has been developed to solve these two problems [10,11]. In this system, product descriptions and design constraints are represented using a feature-based modeling approach. Manufacturing requirements for producing the products, including tasks and sequential constraints for accomplishing these tasks, are represented as part of the product feature descriptions. Manufacturing resources, including facilities and persons, are modeled as distributed agents that are coordinated by two mediators. The optimal production schedule and its timing parameter values are identified using constraint-based search and agent-based collaboration approaches. This project was initiated from the requirements of a building product manufacturing company — Gienow Building Products Ltd., where production tasks are created from customer orders.

The research presented in this paper is a further development of this intelligent production scheduling system by introducing a reactive scheduling mechanism for responding to changes of production orders and manufacturing resources. Changes of production orders include cancellation of previously scheduled orders and insertion of urgent orders. Changes of manufacturing resource conditions include breakdowns of machines and sickness of persons. Match-up and agent-based collaboration approaches are employed for rescheduling the tasks and identifying the optimal timing parameter values of these tasks, for improving the scheduling efficiency while maintaining the scheduling quality.

The remaining of this paper is organized as follows. Section 2 introduces the previously developed predictive scheduling mechanism. Section 3 proposes the architecture of the intelligent production scheduling system that supports both the predictive scheduling function and the reactive scheduling function. Section 4 presents the reactive scheduling algorithms and examples for responding to changes of orders and resources. Section 5 gives a number of case study examples to show the effectiveness of the introduced approach. Section 6 summarizes this research.

2. Review of a previously developed predictive scheduling mechanism

The previously developed predictive scheduling mechanism is composed of three sub-systems: product modeling sub-system, resource management sub-system, and scheduling sub-system.

2.1. Product modeling sub-system

In the product modeling sub-system, a product is modeled by primitives called features [12,13]. Features are described at two different levels, class level and instance level, corresponding to standard product libraries and special product data, respectively. Instance features are generated using class features as their templates. A feature is composed of element features, attributes, qualitative relations among features, and quantitative relations among attributes. For instance, Fig. 1 shows a product modeled by three instance features. The top-level instance feature, `c`, is generated from a class feature `WindowCenter` that is composed of two element features: `?Left` and `?Right`. When the class feature, `Window-Center`, is used to generate its instance feature, `c`, the two element features are also generated as instance features, `cl` and `cr`, respectively.

The manufacturing requirements for producing each feature are defined by a graph of tasks, representing the sequential constraints to accomplish these tasks. For instance, the right component, `cr`, shown in Fig. 1 can be produced by six tasks, including cutting, `C`, framing, `F`, assemblies, `A1`, `A2`, and `A3`, and glazing, `G`. A task in an instance feature is carried out in production only when all the tasks in this feature's element features have been completed. Each task is defined by its type, requirements of resources including facilities and persons, and time period to carry out this process.

2.2. Resource management sub-system

In the resource management sub-system, the facility resources and person resources are modeled as

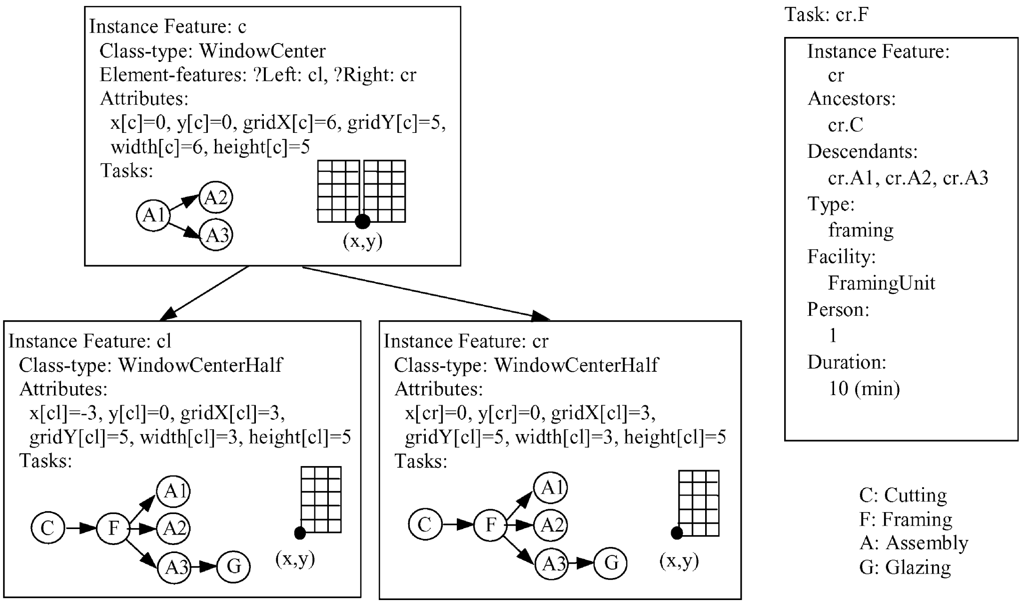

Fig. 1. Feature-based product and manufacturing requirement representation.

agents, which are coordinated by a facility mediator and a personnel mediator during the predictive scheduling process. The idea to model resources using agents comes from the distributed modeling approach for improving flexibility of manufacturing systems [14–19]. A facility resource agent is defined by its type, manufacturing functions, and time constraints including available periods and unavailable periods. A person resource agent is defined by the facilities that the person is responsible for, and time constraints including available periods, regular schedule, and unavailable periods.

2.3. Scheduling sub-system

The scheduling sub-system aims at identifying the optimal schedule for the orders received from customers. When an order is received, an order agent is then created to represent the customer requirements. The order agents negotiate with the resource agents using the corresponding design constraints and manufacturing requirements, which are preserved in the instance features, to identify the optimal production schedule. Constraint-based search and agent-based collaboration approaches are employed for identifying the optimal schedule, as shown in Fig. 2.

2.3.1. Constraint-based search

The optimal sequence of tasks for a customer order is identified using best-first search [20], as shown in Fig. 2. Each node in the search tree represents a partial schedule developed so far. A start node describes an empty schedule, while a goal node describes the schedule in which all the tasks of the customer order have been allocated with required resources and timing parameter values. In predictive scheduling, each time the best node is selected for generating its sub-nodes. When a sub-node is generated, an unscheduled task is then selected for resource allocation and timing parameter value instantiation through collaboration among relevant agents. Evaluation to this node is conducted using a heuristic function. This process is conducted continuously until the selected best-node is the goal node. The scheduling results are described

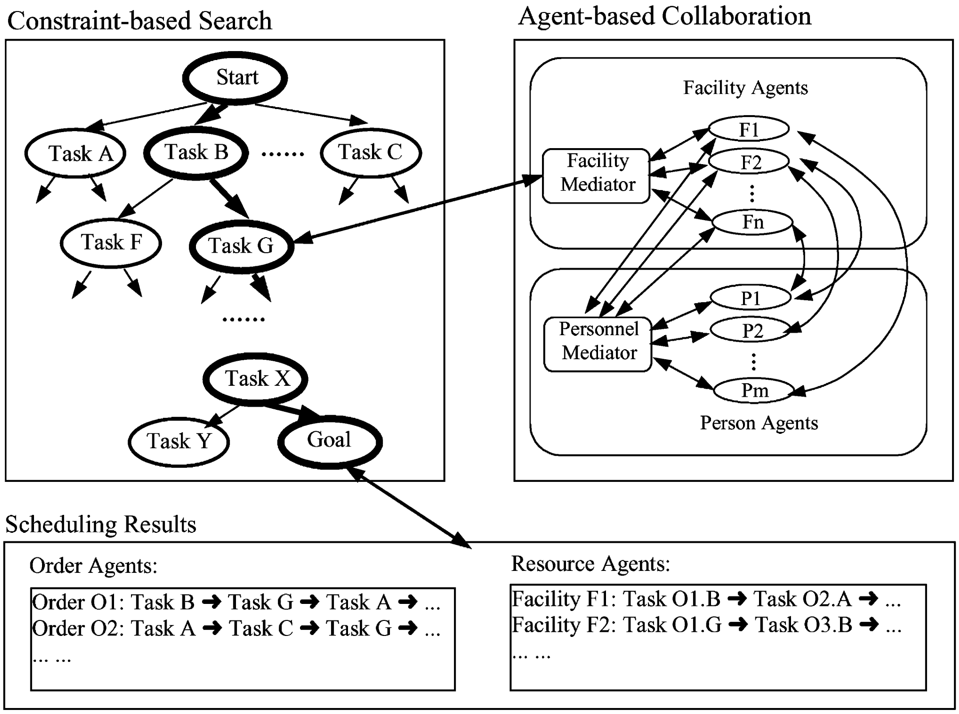

Fig. 2. Predictive scheduling using constraint-based search and agent-based collaboration.

by sequences of tasks that are preserved in order agents and resource agents, as shown in Fig. 2. In predictive scheduling, the created schedule should satisfy the following temporal constraints: (1) a task in an instance feature can be carried out in production only when all the tasks in this feature's element features have been completed. (2) A task can be carried out only when all its ancestor tasks have been completed.

Two heuristic functions have been developed in this research: (1) F_{max} — the latest task finish time considering all the scheduled tasks of an order and (2) S_{min} — the earliest task start time considering all the scheduled tasks of an order. Two search strategies for predictive scheduling have also been developed based upon the two heuristic functions: (1) earliest-delivery-time-based scheduling strategy — to provide the product to the customer as early as possible by selecting the node with the minimum value of the F_{max} as the best node, and (2) due-time-based scheduling strategy — to start the product manufacturing as late as possible to reduce the space for storing the produced product by selecting the node with the maximum value of the S_{min} as the best node.

2.3.2. Agent-based collaboration

Allocation of resources and instantiation of timing parameter values for the required tasks are conducted based upon agent-based collaboration using the contract net protocol [21]. Two timing parameters of tasks, *start time* and *finish time*, are considered in scheduling. The agent-based collaboration in predictive scheduling is conducted at two different levels: order-facility collaboration level and facility-person collaboration level, as shown in Fig. 2. When the facility mediator receives a to-be-scheduled task from the order agent, this mediator sends messages to all the relevant facility agents it knows. Each facility agent then starts negotiation with the relevant person agents through the personnel mediator and sends a bid (with the proposed start time, finish time, and person) to the facility mediator. The facility mediator selects the facility that provides the best bid, such as the one with the earliest product manufacturing completion time for the earliest-delivery-time-based scheduling, or the one with the latest product manufacturing releasing time for the due-time-based scheduling.

3. Architecture of an intelligent production scheduling system

The dynamic reactive scheduling mechanism introduced in this research was developed as a

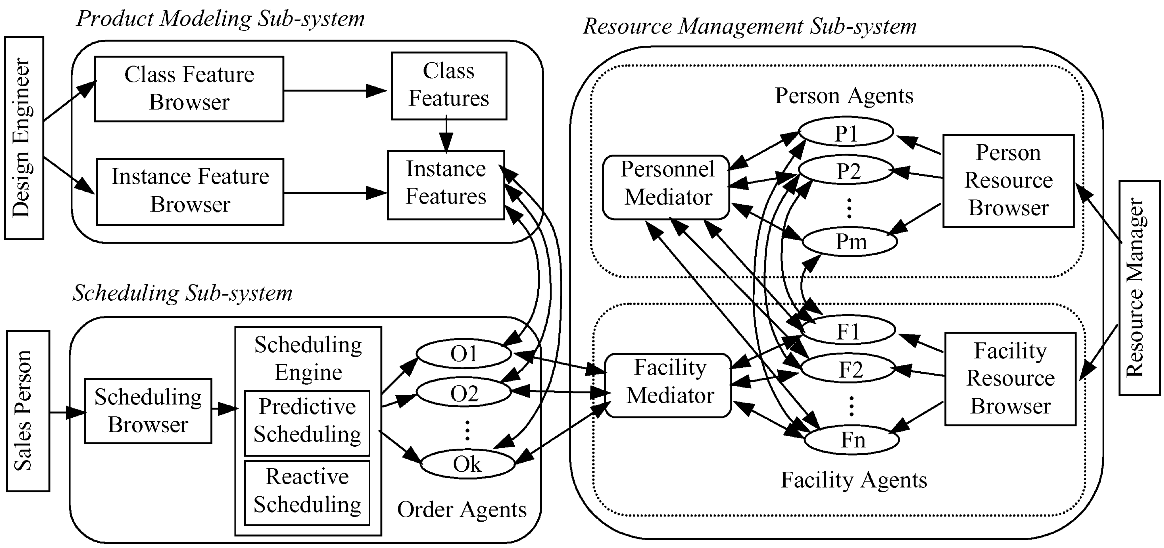

Fig. 3. Architecture of the intelligent production scheduling system.

function block added to the scheduling module of the previously developed intelligent production scheduling system. Architecture of the whole intelligent production scheduling system with both predictive scheduling function and reactive scheduling function is shown in Fig. 3. The system is composed of three sub-systems: product modeling sub-system, resource management sub-system, and scheduling sub-system.

Predictive scheduling is conducted to allocate resources and their timing parameter values for producing the products ordered by customers prior to the production process. Reactive scheduling, on the other hand, is conducted to modify the created schedule for responding to the changes of customer orders (such as cancellation of orders or insertion of urgent orders) and manufacturing conditions (such as machine breakdowns and persons' sudden sickness) during the production process.

The intelligent predictive/reactive production scheduling system was implemented using Smalltalk, an object oriented programming language [22]. Smalltalk was used due to its good user interface environment and large class library. Details of the developed dynamic reactive scheduling mechanism will be described in the next section.

4. A dynamic reactive scheduling mechanism

In this research, two reactive scheduling algorithms have been developed for responding to the changes of customer orders and manufacturing resource conditions, respectively, by partially modifying the originally created schedule. Changes of customer orders include cancellation of scheduled orders and insertion of urgent orders. Changes of manufacturing resource conditions include breakdowns of machines and sudden sickness of persons.

In both algorithms, the previously scheduled tasks are modified. The objective of this research is to develop a reactive scheduling method to minimize the schedule changes for improving the efficiency of reactive scheduling, while maintaining the quality of reactive scheduling. Since the revised schedule can maximally match up with the original schedule, this reactive scheduling approach is also called a *match-up reactive scheduling* approach.

4.1. Reactive scheduling for customer order changes

Changes of customer orders are of two cases: (1) cancellation of scheduled orders and (2) insertion of urgent orders. When a previously scheduled order is canceled by its customer, all the resource time slots assigned to the tasks of this order should be released. To improve the quality of the overall schedule considering all the other ordered products, the tasks scheduled using the due-time-based scheduling strategy could be moved forward towards their due-time measures, while the tasks scheduled using the earliest-delivery-time-based scheduling strategy could be moved backward towards their ordering time measures. When a feasible schedule cannot be identified for an order using the predictive scheduling strategy due to its urgent due-time requirement, some of the previously scheduled tasks can be temporarily released for inserting the tasks in the urgent order to the schedule. The released tasks should be rescheduled to satisfy the product and manufacturing constraints.

Since one of the objectives in this research is to integrate production scheduling with product design, when design parameters are changed, the manufacturing requirements are then updated automatically. When these manufacturing requirements cannot be satisfied by the currently created schedule, change of the production schedule can be conducted simply by canceling the original order and inserting the modified order.

To minimize the schedule change, while maintaining the quality of the overall schedule, the following rules have been employed in reactive scheduling for customer order changes.

1. The customer orders previously scheduled using the due-time-based scheduling strategy are rescheduled prior to the revision of the customer orders previously scheduled using the earliest-delivery-time-based scheduling strategy.
2. In the revised schedule, the sequence of tasks for each to-be-rescheduled order remains the same as the sequence in the original schedule to satisfy the task precedence constraints and improve the rescheduling efficiency.
3. In the revised schedule, each rescheduled task is still allocated with the facility resource and

person resource that were originally allocated to this task.

Reactive scheduling algorithm responding to customer order changes is formulated as the following three steps.

1. *Step 1*: Initialize for rescheduling.

   Generate a copy of the original schedules that are preserved in the resource agents. Consider all the orders that have not been manufactured so far as the to-be-rescheduled orders and remove their original schedules from the resource agents. In case of canceling an order, the canceled order should not considered in further rescheduling process. In case of inserting an order, this order is scheduled first using the due-time-based predictive scheduling approach.
2. *Step 2*: Reschedule the to-be-rescheduled orders that were previously scheduled using the due-time-based scheduling strategy.
   2.1. For these to-be-rescheduled orders, identify the to-be-rescheduled tasks from the copy of the original schedules. In case of canceling an order, the tasks that precede the tasks of the canceled order in the original schedules should be considered as the to-be-rescheduled tasks. In case of inserting an order, the tasks whose original schedules are in conflict with the schedule of the inserted order should be considered as the to-be-rescheduled tasks. Sort the list of the to-be-rescheduled tasks according to the finish time values of these tasks. The to-be-rescheduled task with the largest finish time value is placed at the beginning of the list.
   2.2. Select the first element from the to-be-rescheduled task list as the current to-be-rescheduled task. Recover schedules of the tasks that are in the to-be-rescheduled orders and will start after the finish time of the current to-be-rescheduled task. Reassign timing parameter values to the current to-be-rescheduled task using agent-based collaboration mechanism. The current to-be-rescheduled task should be removed from the list of the to-be-rescheduled tasks.
   2.3. Check if the reassigned timing parameter values are the same as those in the copy of the original schedules. If they are not the same, the following tasks belonging to the to-be-rescheduled orders should be added to the list of the to-be-rescheduled tasks: (1) the tasks preceding the current to-be-rescheduled task in the copy of the original schedules preserved in the relevant facility agent and person agent that were allocated for the current to-be-rescheduled task, (2) the task preceding the current to-be-rescheduled task in the task sequence of the corresponding customer order, and (3) the tasks whose original schedules are in conflict with the revised schedule for the current to-be-rescheduled task.
   2.4. Check if the list of the to-be-rescheduled tasks is empty. If the list is not empty, go to Step 2 (2.1).
   2.5. Recover all the tasks of the to-be-rescheduled orders that are preserved in the copy of the original schedules and have not been rescheduled so far in the reactive scheduling process.
3. *Step 3*: Reschedule the to-be-rescheduled orders that were previously scheduled using the earliest-delivery-time-based scheduling strategy.
   3.1. For these orders, identify the to-be-rescheduled tasks from the copy of the original schedules. The tasks whose original schedules are in conflict with the revised schedules are considered as the to-be-rescheduled tasks. In case of canceling an order, the tasks that follow the tasks of the canceled order in the original schedules should also be considered as the to-be-rescheduled tasks. Sort the list of the to-be-rescheduled tasks according to the start time values of these tasks. The to-be-rescheduled task with the smallest start time value is placed at the beginning of the list.
   3.2. Select the first element from the to-be-rescheduled task list as the current to-be-rescheduled task. Recover schedules of the tasks that are in the to-be-rescheduled orders and will be completed before the start time of the current to-be-rescheduled task. Reassign timing parameter values to the current to-be-rescheduled task using the agent-based collaboration mechanism. The current to-be-rescheduled

task should be removed from the list of the to-be-rescheduled tasks.

3.3. Check if the reassigned timing parameter values are the same as those in the copy of the original schedules. If they are not the same, the following tasks belonging to the to-be-rescheduled orders should be added to the list of the to-be-rescheduled tasks: (1) the tasks following the current to-be-rescheduled task in the copy of the original schedules preserved in the relevant facility agent and person agent that were allocated for the current to-be-rescheduled task, (2) the task following the current to-be-rescheduled task in the task sequence of the corresponding customer order, and (3) the tasks whose original schedules are in conflict with the revised schedule for the current to-be-rescheduled task.

3.4. Check if the list of the to-be-rescheduled tasks is empty. If the list is not empty, go to Step 3 (3.2).

3.5. Recover all the tasks of the to-be-rescheduled orders that are preserved in the copy of the original schedules and have not been rescheduled so far in the reactive scheduling process.

An example is given in Fig. 4 to illustrate the reactive scheduling algorithm for responding to customer order change. In this example, an urgent order `E` needs to be inserted in the original schedule, which is shown in Fig. 4a. The orders `A` and `B` were scheduled using the earliest-delivery-time-based scheduling strategy, and the orders `C` and `D` were scheduled using the due-time-based scheduling strategy. The sequences of tasks for the orders `A`, `B`, `C`, and `D` in the original schedule are:

$$\text{Order A} : \text{A1} \to \text{A2} \to \text{A3}$$
$$\text{Order B} : \text{B1} \to \text{B2} \to \text{B3} \to \text{B4}$$
$$\text{Order C} : \text{C1} \to \text{C2} \to \text{C3}$$
$$\text{Order D} : \text{D1} \to \text{D2} \to \text{D3}$$

In Step 1, as shown in Fig. 4b, orders `A`, `B`, `C`, and `D` are identified as the to-be-rescheduled orders and removed from the original schedule temporally. Then, order `E` is scheduled using the due-time-based scheduling strategy.

In Step 2, as shown in Fig. 4c, the orders `C` and `D` that were scheduled previously using the due-time-based scheduling strategy are rescheduled. The tasks `D1` and `D2` are identified as the to-be-rescheduled tasks at the beginning, due to their conflict with the tasks `E1` and `E2`. Based on the finish time values, `D2` is selected as the first current to-be-rescheduled task. Task `D3` that is to be started after the finish time of `D2` is recovered in the revised schedule. Then `D2` is assigned with new timing parameter values. Since the timing parameter values of `D2` are changed in the revised schedule, two tasks are identified as the new to-be-rescheduled tasks: `C2` that precedes `D2` in the original schedule considering facility `F2` and `D1` that precedes `D2` in the task sequence of the original schedule for order `D`. Only `C2` is added to the to-be-rescheduled tasks, since `D1` has been already identified as a to-be-rescheduled task at the beginning of this step. Next, `D1` is selected as the current to-be-rescheduled task and assigned with new timing parameter values. Although the new timing parameter values of `D1` are different from the original ones, no to-be-rescheduled tasks are further identified. Then task `C2` is selected as the current to-be-rescheduled task. Task `C3` that is to be started after the finish time of `C2` is recovered in the revised schedule. The task `C2` is rescheduled with the same timing parameter values. After task `C2` is rescheduled, the to-be-rescheduled task list becomes an empty one. Task `C1` is recovered with its original schedule.

In Step 3, as shown in Fig. 4d, the orders `A` and `B` that were scheduled previously using the earliest-delivery-time-based scheduling strategy are rescheduled. The original schedule of `B4` is in conflict with the revised schedule of `D2`, thus, `B4` is identified as the to-be-rescheduled task at the beginning. Next, `B4` is selected as the current to-be-rescheduled task and the tasks `A1`, `A2`, `A3`, `B1`, `B2`, and `B3`, whose finish time values precede the start time of `B4` in the original schedule, are recovered in the revised schedule. Then, `B4` is assigned with new timing parameter values. Since no additional tasks can be identified as the to-be-rescheduled tasks, the reactive scheduling process is terminated.

In this example, among the 13 tasks in the four orders, only 3 tasks are revised during the rescheduling process for responding to the change of customer order.

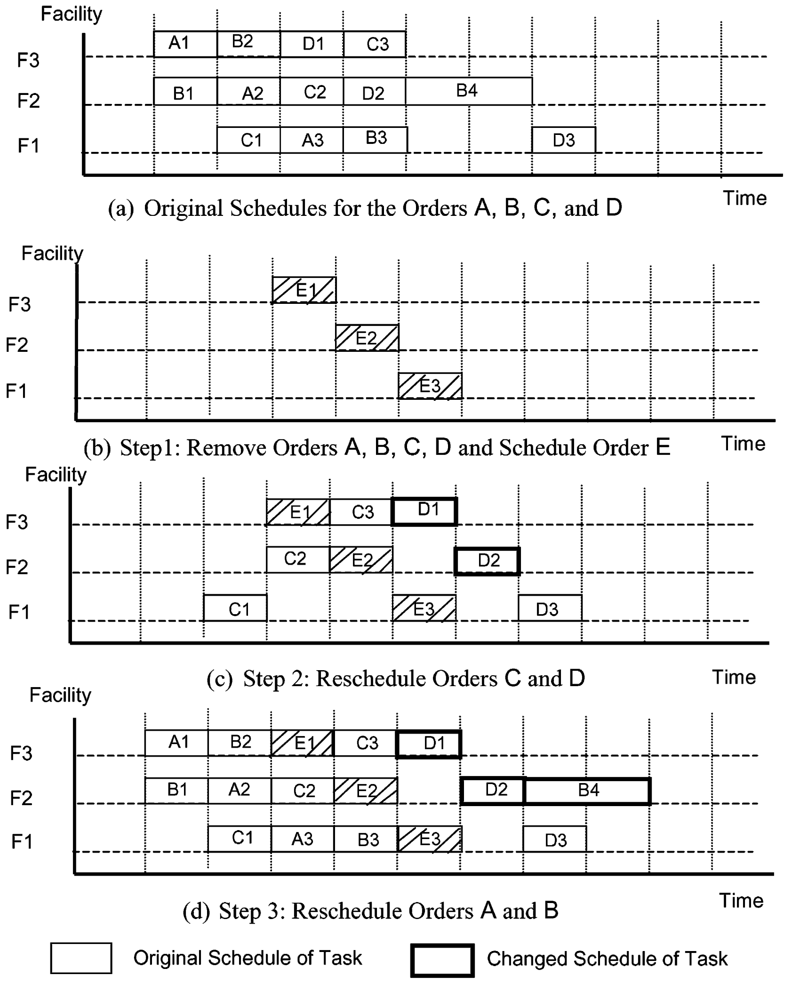

Fig. 4. Reactive scheduling for customer order change.

In the rescheduling process, reassignment of timing parameter values to the to-be-rescheduled tasks is conducted based upon agent-based collaboration among resource agents through coordination of the two mediators in the resource management sub-system. Two task timing parameters, *start time* and *finish time*, are considered in this research.

An example is shown in Fig. 5 to illustrate the process of reassigning timing parameter values to a to-be-rescheduled task of an order, which was originally scheduled using the earliest-delivery-time-based scheduling strategy. During the rescheduling process, the current to-be-rescheduled task `A` needs to be reassigned with new timing parameter values. The resources, facility agent `F1` and person agent `P1`, were originally allocated for this task. The earliest possible start time is T_{es} that was determined based on task precedence constraints and feature relations using the previously developed predictive scheduling method. First, the facility mediator reassigns this task to the facility agent `F1`. Upon receiving this message, the facility agent `F1` identifies the related person agent `P1` through the personnel mediator. Then the facility agent `F1` negotiates with the person agent `P1` to determine the proper time slot for the task `A`. The time slot should provide the minimum value of the task start time in the earliest-delivery-time-based scheduling, while satisfying

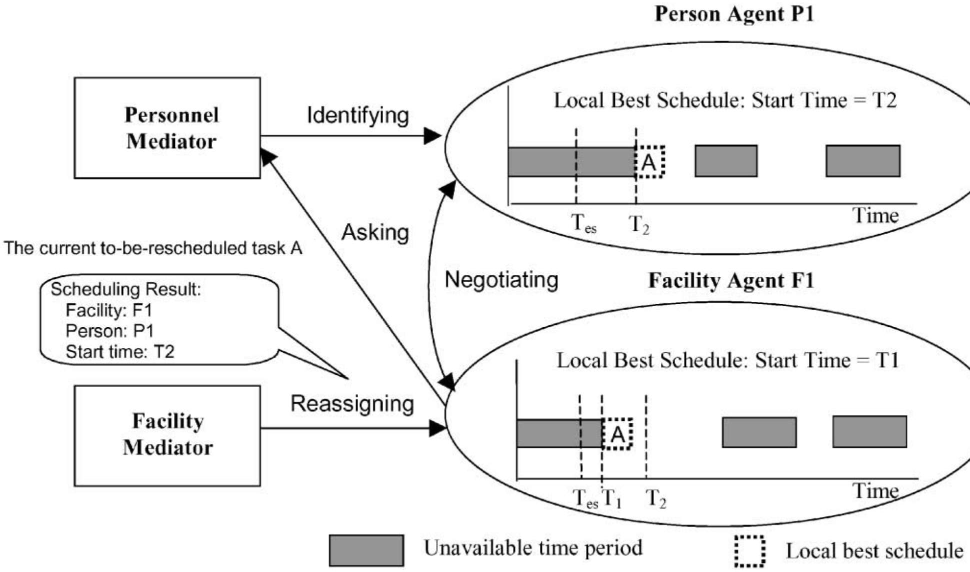

Fig. 5. An example of rescheduling timing parameter values through agent-based collaboration.

all the manufacturing requirement and resource constraints.

4.2. Reactive scheduling for manufacturing resource changes

Manufacturing resources changes are of two cases: (1) facility breakdowns and (2) persons' sudden sickness. When such a disturbance occurs, the schedules of the affected tasks have to be revised. Match-up rescheduling approach is also employed to minimize the changes to the originally created schedules, while satisfying the product and manufacturing constraints. The following rules have been used for reactive scheduling to respond to the changes of manufacturing resources.

1. The reactive scheduling mechanism first tries to move the tasks that are affected directly by the resource changes to other resources without changing the timing parameter values of these tasks.
2. If the alternative resources can not be identified for the affected tasks, match-up-based rescheduling is then conducted. The customer orders previously scheduled using the due-time-based scheduling strategy are rescheduled prior to the revision to the customer orders previously scheduled using the earliest-delivery-time-based scheduling strategy.
3. If some orders, which were previously scheduled using the due-time-based scheduling strategy, cannot be rescheduled to satisfy the due-time constraints due to the changes of resource conditions, the directly affected orders will be rescheduled with modified due-time values, after all other orders with due-time requirements have been rescheduled.
4. In the revised schedule, the sequence of tasks for each to-be-rescheduled order remains the same as the sequence in the original schedule to satisfy the task precedence constraints while improving the rescheduling efficiency.
5. In the revised schedule, if alternative resources cannot be identified for the affected tasks, each rescheduled task is still allocated with the resources that were originally assigned to this task for improving the rescheduling efficiency.

Reactive scheduling algorithm responding to manufacturing resource changes is formulated as the following five steps.

1. *Step 1*: Identify the alternative resources.

   For each of the tasks whose original schedules cannot be completed due to the changes of resources, identify the alternative resources that can accomplish this task without changing the timing parameter values of this task. If all the affected tasks can be allocated with alternative resources, the rescheduling process should be terminated. Otherwise, go to Step 2.
2. *Step 2*: Initialize for rescheduling.

   Generate a copy of the original schedules that are preserved in the resource agents. Consider all the orders that have not been manufactured so far as the to-be-rescheduled orders and remove their original schedules from the resource agents. The affected time periods of the resources should be marked as unavailable time periods and not be considered in the rescheduling process.
3. *Step 3*: Reschedule the to-be-rescheduled orders that were previously scheduled using the due-time-based scheduling strategy.
   - 3.1. For these to-be-rescheduled orders, identify the to-be-rescheduled tasks from the copy of the original schedules. The tasks, whose original schedules are affected directly by facility breakdowns and persons' sickness, should be considered as the to-be-rescheduled tasks. Sort the list of the to-be-rescheduled tasks according to the finish time values of these tasks. The to-be-rescheduled task with the largest finish time value is placed at the beginning of the list.
   - 3.2. Select the first element from the to-be-rescheduled task list as the current to-be-rescheduled task. Recover schedules of the tasks that are in the to-be-rescheduled orders and will start after the finish time of the current to-be-rescheduled task. Reassign timing parameter values to the current to-be-rescheduled task using the agent-based collaboration mechanism. The current to-be-rescheduled task should be removed from the list of the to-be-rescheduled tasks.
   - 3.3. Check if the reassigned timing parameter values are the same as those in the copy of the original schedules. If they are not the same, the following tasks belonging to the to-be-rescheduled orders should be added to the list of the to-be-rescheduled tasks: (1) the tasks preceding the current to-be-rescheduled task in the copy of the original schedules preserved in the relevant facility agent and person agent that were allocated for the current to-be-rescheduled task, (2) the task preceding the current to-be-rescheduled task in the task sequence of the corresponding customer order, and (3) the tasks whose original schedules are in conflict with the revised schedule for the current to-be-rescheduled task.
   - 3.4. Check if the list of the to-be-rescheduled tasks is empty. If the list is not empty, go to Step 3 (3.2).
   - 3.5. Recover all the tasks of the to-be-rescheduled orders that are preserved in the copy of the original schedules and have not been rescheduled so far in the reactive scheduling process.
4. *Step 4*: If some orders cannot be rescheduled in Step 3 because of the due-time constraints, all the schedules created in Step 3 should be removed. In this case, the directly affected orders that were previously scheduled using the due-time-based scheduling strategy should not be considered as the to-be-rescheduled orders temporarily. Go to Step 3 and reschedule other to-be-rescheduled orders that were previously scheduled using the due-time-based scheduling strategy. These directly affected orders should be rescheduled with modified due-time requirements.
5. *Step 5*: Reschedule the to-be-rescheduled orders that were previously scheduled using the earliest-delivery-time-based scheduling strategy.
   - 5.1. For these orders, identify the to-be-rescheduled tasks from the copy of the original schedules. The tasks whose original schedules are affected directly by facility breakdowns and persons' sickness and the tasks whose original schedules are in conflict with the revised schedules should be considered as the to-be-rescheduled tasks. Sort the list of the to-be-rescheduled tasks according to the start time values of these tasks. The to-be-rescheduled task with the smallest start

time value is placed at the beginning of the list.

5.2. Select the first element from the to-be-rescheduled task list as the current to-be-rescheduled task. Recover schedules of the tasks that are in the to-be-rescheduled orders and will be completed before the start time of the current to-be-rescheduled task. Reassign timing parameter values to the current to-be-rescheduled task using the agent-based collaboration mechanism. The current to-be-rescheduled task should be removed from the list of the to-be-rescheduled tasks.

5.3. Check if the reassigned timing parameter values are the same as those in the copy of the original schedules. If they are not the same, the following tasks belonging to the to-be-rescheduled orders should be added to the list of the to-be-rescheduled tasks: (1) the tasks following the current to-be-rescheduled task in the copy of the original schedules preserved in the relevant facility agent and person agent that were allocated for the current to-be-rescheduled task, (2) the task following the current to-be-rescheduled task in the task sequence of the corresponding customer order, and (3) the tasks whose original schedules are in conflict with the revised schedule for the current to-be-rescheduled task.

5.4. Check if the list of the to-be-rescheduled tasks is empty. If the list is not empty, go to Step 5 (5.2).

5.5. Recover all the tasks of the to-be-rescheduled orders that are preserved in the copy of the original schedules and have not been rescheduled so far in the reactive scheduling process.

An example is given in Fig. 6 to illustrate the reactive scheduling process for responding to manufacturing resource change. In the original schedules shown in Fig. 6a, the orders `A` and `B` were scheduled using the earliest-delivery-time-based scheduling strategy, and the orders `C` and `D` were scheduled using the due-time-based scheduling strategy. It is predicted that the facility `F2` will need emergency maintenance service in the time period (t_{b1}–t_{b2}). The sequences of tasks for the orders `A`, `B`, `C`, and `D` in the original schedule are:

$$\text{Order A}: \text{A1} \rightarrow \text{A2} \rightarrow \text{A3}$$
$$\text{Order B}: \text{B1} \rightarrow \text{B2} \rightarrow \text{B3} \rightarrow \text{B4}$$
$$\text{Order C}: \text{C1} \rightarrow \text{C2} \rightarrow \text{C3}$$
$$\text{Order D}: \text{D1} \rightarrow \text{D2} \rightarrow \text{D3}$$

In Step 1, no alternative facilities can be identified for the affected tasks `D2` and `B4`.

In Step 2, as shown in Fig. 6b, orders `A`, `B`, `C`, and `D` are identified as the to-be-rescheduled orders and their schedules are removed temporally. The time slot (t_{b1}–t_{b2}) is recorded as the unavailable period for facility `F2`.

In Step 3, as shown in Fig. 6c, the orders `C` and `D`, which were previously scheduled using the due-time-based scheduling strategy, are rescheduled. Since task `D2` is affected directly by the change of resource condition, this task is identified as the to-be-rescheduled task at the beginning. Then, task `D3` that starts after the finish time of `D2` is recovered with its original timing parameter values in the revised schedule. As the current to-be-rescheduled task, `D2` is assigned with new timing parameter values. Since the reassigned timing parameter values of `D2` are different from the original ones, task `D1` that precedes `D2` in the task sequence of the original schedule for order `D` and task `C2` that precedes the `D2` in the original schedules considering facility resource `F2` are added to the list of the to-be-rescheduled tasks. Next, `C2` is selected as the current to-be-rescheduled task. Task `C3` that starts after the finish time of `C2` is recovered with its original timing parameter values. Then `C2` is assigned with new timing parameter values. Task `C1` is then identified as a new to-be-rescheduled task, since `C1` precedes `C2` in the task sequence for order `C`. The reactive scheduling process is continued until `C1` is rescheduled. The revised schedules satisfy all the constraints for orders `C` and `D`.

In Step 4, since it is found that all the orders, which were previously scheduled using the due-time-based scheduling strategy, can be rescheduled in Step 3, rescheduling process goes to Step 5.

In Step 5, as shown in Fig. 6d, the orders `A` and `B`, which were previously scheduled using the earliest-delivery-time-based scheduling strategy, are rescheduled. Since the original schedule for task `B4` is affected by the resource condition change and tasks

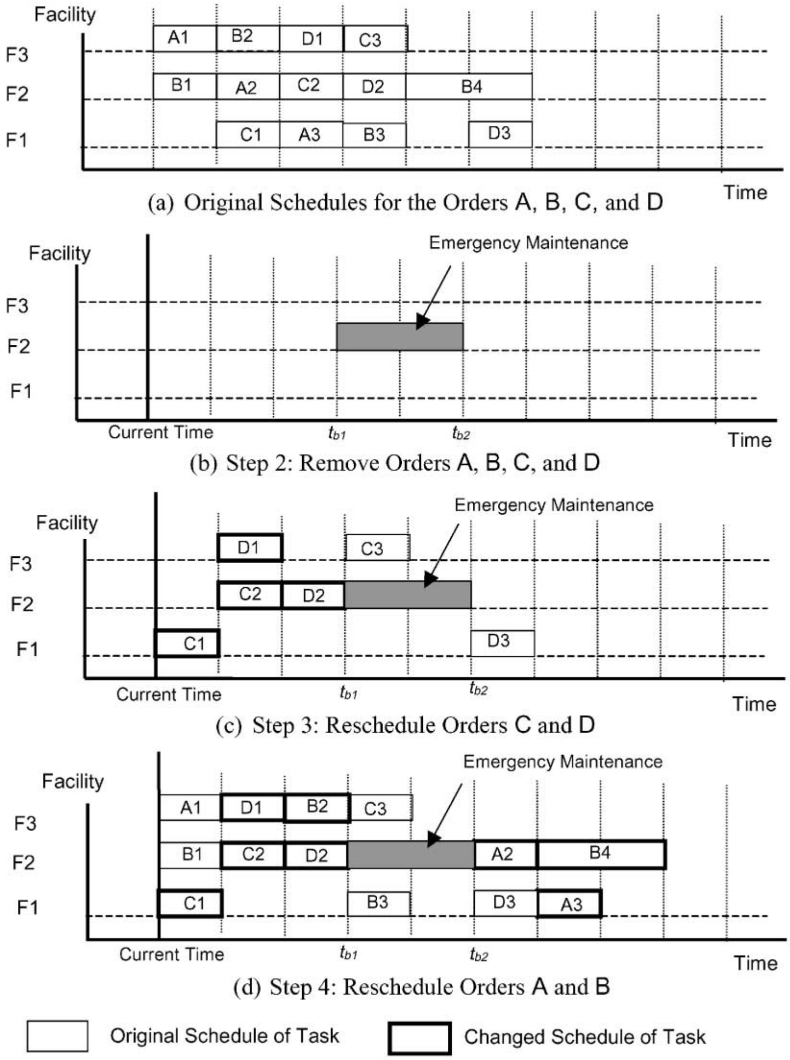

Fig. 6. Reactive scheduling for manufacturing resource change.

A2 and B2 are in conflict with the revised schedules, these tasks are identified as the to-be-rescheduled tasks at the beginning. First, task A2 is selected for rescheduling. Since tasks A1 and B1 complete before the start time of A2, these two tasks are recovered with their original timing parameter values in the revised schedule. The task A2 is then assigned with new parameter values. The reactive scheduling process is continued until task B4 is rescheduled.

In this example, among the 13 tasks in the four orders, only 8 tasks are revised during the rescheduling process for responding to the change of manufacturing resource condition.

5. Case study examples

In this section, a number of case study examples are given to illustrate the effectiveness of the developed dynamic reactive production scheduling mechanism. In these examples, only one type of product, represented by instance feature c and its element features

Table 1
Manufacturing requirements for three instance features

Instance features	Tasks	Ancestors	Descendants	Types	Facilities	Duration (min)
c	c.A1		c.A2, c.A3	Assembly E	Assembly unit	15
	c.A2	c.A1		Assembly A	Assembly unit	10
	c.A3	c.A3		Assembly A	Assembly unit	10
cr	cr.C		cr.F	Cutting	Cutting machine	10
	cr.F	cr.C	cr.A1, cr.A2, cr.A3	Framing	Framing unit	10
	cr.A1	cr.F		Assembly A	Assembly unit	10
	cr.A2	cr.F		Assembly A	Assembly unit	10
	cr.A3	cr.F	cr.G	Assembly A	Assembly unit	10
	cr.G	cr.A3		Glazing	Assembly unit	15
cl	cl.C		cl.F	Cutting	Cutting machine	10
	cl.F	cl.C	cl.A1, cl.A2, cl.A3	Framing	Framing machine	10
	cl.A1	cl.F		Assembly A	Assembly unit	10
	cl.A2	cl.F		Assembly A	Assembly unit	10
	cl.A3	cl.F	cl.G	Assembly A	Assembly unit	10
	cl.G	cl.A3		Glazing	Assembly unit	15

cl and cr as shown in Fig. 1, is used for modeling production orders. The manufacturing requirements for these instance features are given in Table 1.

A total of 11 facility agents and 11 person agents are used in these examples. The 11 facilities include 2 cutting machines, 1 framing machine, 7 assembly units, and 1 packing unit, while 11 persons are assigned to work for the relevant facilities. Table 2 gives the resource definitions of the 11 facility resources.

5.1. An example for canceling an old order

In this case study example, five orders, Order 1, Order 2, Order 3, Order 4, and Order 5, were scheduled previously. Each order is defined by a requirement to produce a window product c, as shown in Fig. 1, which is an instance feature of class feature WindowCenter. Order 1, Order 2, Order 3, and Order 4 were scheduled using the due-time-based scheduling strategy, while Order 5 was

Table 2
Definitions of facility resource agents

Facility agents	Types	Functions	Time constraints
FC01	Cutting machine	Cutting	Available periods: [1 October 1998, 0:00 to 31 December 1998, 23:59]; unavailable periods: [16 October 1998, 8:00 to 16 October 1998, 8:30]
FC02	Cutting machine	Cutting	Available periods: [1 October 1998, 0:00 to 31 December 1998, 23:59]
FF01	Framing machine	Framing	The same as those in FC02
FA01	Assembly unit	Assembly A	The same as those in FC02
FA02	Assembly unit	Assembly A	The same as those in FC02
FA03	Assembly unit	Assembly A	The same as those in FC02
FA04	Assembly unit	Assembly A	The same as those in FC02
FA05	Assembly unit	Assembly E	The same as those in FC02
FA06	Assembly unit	Glazing	The same as those in FC02
FA07	Assembly unit	Glazing	The same as those in FC02
FP01	Packing unit	Packing	The same as those in FC02

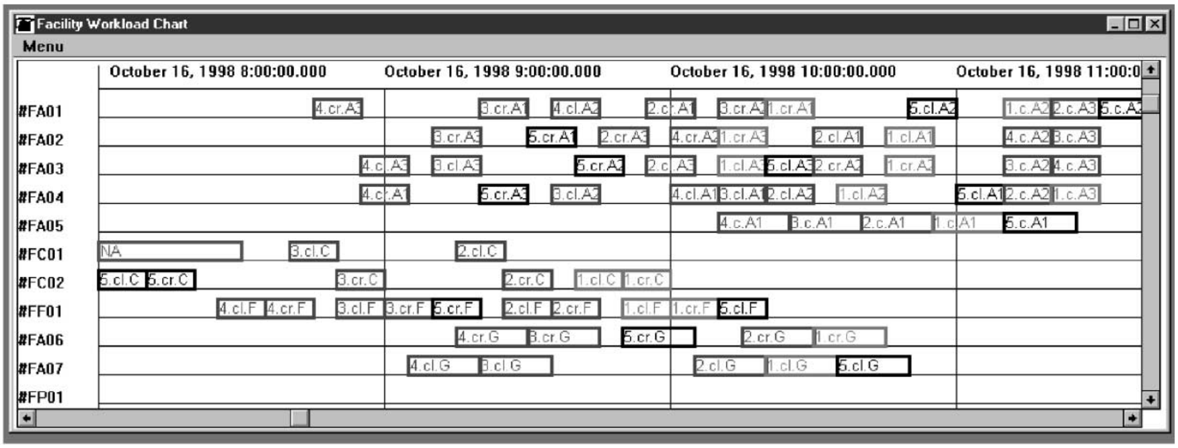

Fig. 7. The original schedules for `Orders 1`, `2`, `3`, `4`, and `5`.

scheduled using the earliest-delivery-time-based scheduling strategy. The `Order 1` needs to be canceled based upon the customer's request.

Fig. 7 shows the original schedules for `Order 1`, `Order 2`, `Order 3`, `Order 4`, and `Order 5`. Fig. 8 shows the revised schedules after `Order 1` has been canceled. Evaluation of the reactive scheduling mechanism for this case study example is summarized in Table 3. The release time values for `Order 2`, `Order 3`, and `Order 4`, which were previously scheduled using the due-time-based scheduling strategy, are shifted towards to the due-time measures, while the completion time value for `Order 5`, which was previously scheduled using the earliest-delivery-time-based scheduling strategy, is shifted backwards to the release time. The original schedule is partially revised for responding to the cancellation of `Order 1`. The ratio of the revised task number and the total task number is 48/60.

5.2. An example for inserting an urgent order

In this case study example, four production orders, `Order 1`, `Order 2`, `Order 3`, and `Order 4`, were scheduled previously. Each order is defined by a requirement to produce a window product `c`, as shown in Fig. 1, which is an instance feature of class feature `WindowCenter`. `Order 1`, `Order 2`, and

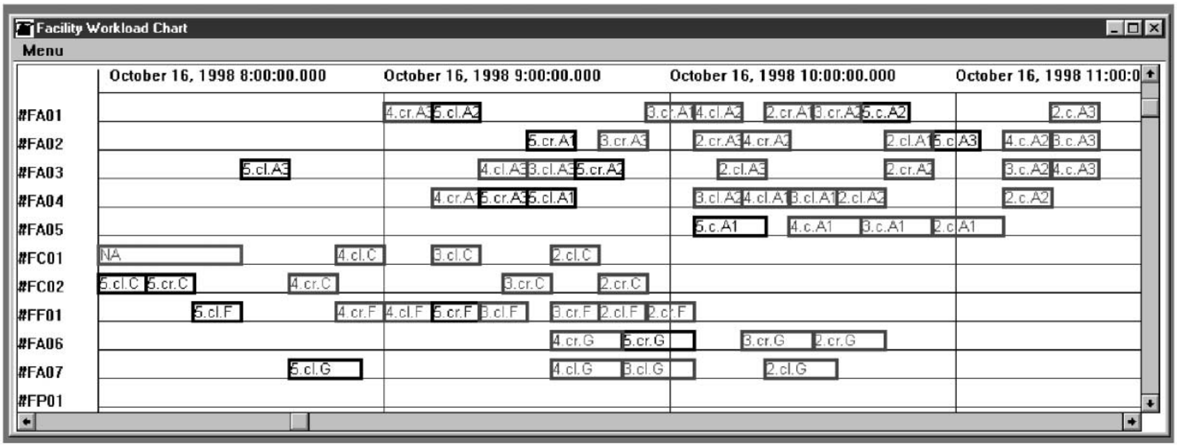

Fig. 8. The revised schedules after cancellation of `Order 1`.

Table 3
Evaluation of a case study example for canceling an old order[a]

Order	1	2	3	4	5
Scheduling strategy	D	D	D	D	E
Due-time	11:30	11:30	11:30	11:30	
Original schedule					
Release time	9:40	9:15	8:40	8:25	8:00
Completion time	11:30	11:30	11:30	11:30	11:40
Revised schedule					
Release time		9:40	9:10	8:40	8:00
Completion time		11:30	11:30	11:30	11:05
Total number of tasks		15	15	15	15
Number of revised tasks		13	13	13	9

[a] D: due-time-based scheduling; E: earliest-delivery-time-based scheduling.

`Order 3` were scheduled using the due-time-based scheduling strategy, while `Order 4` was scheduled using the earliest-delivery-time-based scheduling strategy. A new order, `Order 5`, needs to be inserted with an urgent due-time requirement.

Evaluation of the reactive scheduling mechanism for this case study example is summarized in Table 4. Although the release time values for `Order 1`, `Order 2`, and `Order 3`, which were previously scheduled using the due-time-based scheduling strategy, are decreased, and the completion time value for `Order 4`, which was previously scheduled using the earliest-delivery-time-based scheduling strategy, is increased, the due-time constraints for `Order 1`, `Order 2`, and `Order 3` are still satisfied. The original schedule is partially revised for responding to the insertion of `Order 5`. The ratio of the revised task number and the total task number is 47/60.

5.3. An example for responding to a facility breakdown event

In this case study example, four production orders, `Order 1`, `Order 2`, `Order 3`, and `Order 4`, were scheduled previously. Each order is defined by a requirement to produce a window product `c`, as shown in Fig. 1, which is an instance feature of class feature `WindowCenter`. `Order 1` and `Order 2` were scheduled using the due-time-based scheduling strategy, while `Order 3` and `Order 4` were scheduled using the earliest-delivery-time-based scheduling strategy. It is predicted that the facility `FF01` will not be available from `9:00 a.m.` to `9:20 a.m.`

Evaluation of the reactive scheduling mechanism for this case study example is summarized in Table 5. Although the release time value for `Order 2`, which was previously scheduled using the due-time-based scheduling strategy, is decreased and the completion time value for `Order 4`, which was previously scheduled using the earliest-delivery-time-based scheduling strategy, is increased, the due-time constraint for `Order 2` is still satisfied. The original schedule is partially revised for responding to the breakdown event of the facility `FF01`. The ratio of the revised task number and the total task number is 17/60.

Table 4
Evaluation of a case study example for inserting an urgent order[a]

Order	1	2	3	4	5
Scheduling strategy	D	D	D	E	D
Due-time	11:00	11:00	11:30		11:10
Original schedule					
Release time	9:10	8:45	8:50	8:00	
Completion time	11:00	11:00	11:30	10:00	
Revised schedule					
Release time	9:00	8:30	8:30	8:00	9:20
Completion time	11:00	10:50	11:30	11:40	11:10
Total number of tasks	15	15	15	15	
Number of revised tasks	14	15	9	9	

[a] D: due-time-based scheduling; E: earliest-delivery-time-based scheduling.

Table 5
Evaluation of a case study example for responding to a facility breakdown event[a]

Order	1	2	3	4
Scheduling strategy	D	D	E	E
Due-time	11:00	11:00		
Original schedule				
Release time	9:10	8:45	8:00	8:30
Completion time	11:00	11:00	10:00	11:15
Revised schedule				
Release time	9:10	8:20	8:00	8:30
Completion time	11:00	11:00	10:00	11:40
Total number of tasks	15	15	15	15
Number of revised tasks	0	3	0	14

[a] D: due-time-based scheduling; E: earliest-delivery-time-based scheduling.

5.4. An example for responding to a person's sudden sickness event

In this case study example, four orders, `Order 1`, `Order 2`, `Order 3`, and `Order 4`, were scheduled previously. Each order is defined by a requirement to produce a window product `c`, as shown in Fig. 1, which is an instance feature of class feature `WindowCenter`. `Order 1` and `Order 2` were scheduled using the due-time-based scheduling strategy, while `Order 3` and `Order 4` were scheduled using the earliest-delivery-time-based scheduling strategy. It is predicted that the person `PM03`, who is responsible for the facility `FF01`, will not be available from `8:30 a.m.` to `9:00 a.m.`

Evaluation of the reactive scheduling mechanism for this case study example is summarized in Table 6. In this example, since the `Order 2`, which is affected directly by the person's absence event, cannot be rescheduled to satisfy the original due-time constraint, its due-time is changed from `11:00 a.m.` to `11:30 a.m.` The completion time values for `Order 3` and `Order 4`, which were previously scheduled using the earliest-delivery-time-based scheduling strategy, are increased in the revised schedule. The original schedule is partially revised for responding to the person's absence event. The ratio of the revised task number and the total task number is 36/60.

Table 6
Evaluation of a case study example for responding to a person's sudden sickness event[a]

Order	1	2	3	4
Scheduling strategy	D	D	E	E
Due-time	11:00	11:00		
Original schedule				
Release time	9:10	8:45	8:00	8:30
Completion time	11:00	11:00	10:00	11:15
Revised schedule				
Release time	9:10	8:40	8:00	8:15
Completion time	11:00	11:30	11:20	12:00
Total number of tasks	15	15	15	15
Number of revised tasks	0	15	8	13

[a] D: due-time-based scheduling; E: earliest-delivery-time-based scheduling.

6. Conclusions

This paper presented development of a dynamic reactive scheduling mechanism for an intelligent production scheduling system. This dynamic reactive scheduling mechanism works together with a previously developed predictive scheduling mechanism for identifying the optimal production schedule based upon manufacturing requirements and resource constraints. The predictive scheduling mechanism is used to create the optimal schedule prior to the production process, while the reactive scheduling mechanism aims at modifying the originally created schedule during the production process when the original schedule cannot be completed due to the changes of production orders and manufacturing resources. The match-up and agent-based collaboration approaches are employed in this research for developing the dynamic reactive production scheduling mechanism.

The characteristics of this developed dynamic reactive scheduling approach are summarized as follows.

1. By employing the match-up rescheduling approach to modify only part of the previously created schedule rather than rescheduling all the tasks, efficiency of the reactive scheduling is improved.
2. This reactive scheduling mechanism integrates production scheduling function and product design function into the same environment. When product parameters are changed, manufacturing requirements are updated automatically. When the current schedule cannot satisfy these manufacturing requirements, this schedule is then modified simply by canceling the original order and inserting the modified order.
3. By modeling different types of resource knowledge in different agents, identification of resources and their timing parameter values is conducted at different locations. The multi-agent system approach is effective for handling the complex manufacturing systems that are distributed at different locations.
4. By integrating the reactive scheduling mechanism with a previously developed predictive scheduling mechanism, complete scheduling functions to

predict the optimal schedule and modify the optimal schedule to adapt changes of customer requirements and manufacturing conditions are provided in the developed intelligent production scheduling system.

Acknowledgements

The authors would like to thank the local manufacturing company — Gienow Building Products Ltd., for the efforts to initiate this project.

References

[1] K.R. Baker, Introduction to Sequencing and Scheduling, Wiley, New York, 1974.
[2] M. Zweben, M.S. Fox, Intelligent Scheduling, Morgan Kaufmann, Los Altos, CA, 1994.
[3] A. Kott, V. Saks, A. Mercer, A new technique enables dynamic replanning and rescheduling of aeromedical evacuation, in: Proceedings of 15th National Conference on Artificial Intelligence, Madison, WI, 1998, pp. 1063–1070.
[4] M.S. Fox, Constraint-directed search: a case study of job-shop scheduling, PhD Thesis, CMU-RI-TR-85-7, Intelligent Systems Laboratory, The Robotics Institute, Carnegie Melon University, 1983.
[5] M.S. Fox, N. Sadeh, C. Baycan, Constrained heuristic search, in: Proceedings of International Joint Conference on Artificial Intelligence, Menlo Park, CA, 1989, pp. 309–316.
[6] A. Collinot, C. Le Pape, C. Pinoteau, SONIA, a knowledge-based scheduling system, Artificial Intelligence in Engineering 3 (2) (1988) 86–98.
[7] J.C. Bean, J.R. Birge, J. Mittenehal, C.E. Noon, Match-up scheduling with multiple resource, release dates, and disruption, Operation Research 34 (8) (1991) 2299–2315.
[8] J.S. Smith, OPIS, a methodology and architecture for reactive scheduling, in: M. Zweben, M.S. Fox (Eds.), Intelligent Scheduling, Morgan Kaufmann, Los Altos, CA, 1994, pp. 29–66.
[9] E. Szelke, R.M. Kerr, Knowledge-based reactive scheduling-state of art, International Journal of Production Planing and Control 5 (1994) 124–145.
[10] J. Sun, D. Xue, D.H. Norrie, An intelligent production scheduling mechanism considering design and manufacturing constraints, in: Proceedings of the Third International Conference on Industrial Automation, Montreal, 1999, pp. 24.11–24.14.
[11] D. Xue, J. Sun, D.H. Norrie, Development of an intelligent system for building product design and manufacturing — part II: production process planning and scheduling, in: Proceedings of the Second International Workshop on Intelligent Manufacturing Systems, Leuven, 1999, pp. 57–66.
[12] D. Xue, Z. Dong, Feature modeling incorporating tolerance and production process for concurrent design, Concurrent Engineering: Research and Applications 1 (1993) 107–116.
[13] D. Xue, S. Yadav, D.H. Norrie, Knowledge base and database representation for intelligent concurrent design, Computer-Aided Design 31 (1999) 131–145.
[14] M.J. Shaw, A distributed scheduling method for computer integrated manufacturing: the use of local area networks in cellular systems, International Journal of Production Research 25 (9) (1987) 1285–1303.
[15] K.P. Sycara, S. Roth, N. Sadeh, M. Fox, Distributed constrained heuristic search, IEEE Transactions on Systems, Man and Cybernetics 21 (6) (1991) 1446–1460.
[16] H.V.D. Parunak, Application of distributed artificial intelligence in industry, in: G.M.P. O'Hare, N.R. Jennings (Eds.), Foundations of Distributed Artificial Intelligence, Wiley, New York, 1996, pp. 139–164.
[17] J. Butler, H.D. Ohtsubo, ADDYMS: architecture for distributed dynamic manufacturing scheduling, in: A. Famili, D.S. Nau, S.H. Kim (Eds.), Artificial Intelligence Applications in Manufacturing, AAAI Press, The MIT Press, Cambridge, MA, 1995, pp. 199–214.
[18] K. Saad, K. Kawamura, G. Biswas, Performance evaluation of contract net-based heterarchical scheduling for flexible manufacturing system, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95), Workshop on Intelligent Manufacturing, Montreal, Canada, 1995.
[19] F. Maturana, D.H. Norrie, Multi-agent mediator architecture for distributed manufacturing, Journal of Intelligent Manufacturing 7 (1996) 257–270.
[20] P.H. Winston, Artificial Intelligence, 3rd Edition, Addison-Wesley, Reading, MA, 1992.
[21] R.G. Smith, The contract net protocol: high-level communication and control in a distributed problem solver, IEEE Transactions on Computers 29 (12) (1980) 1104–1113.
[22] A. Goldberg, D. Robson, Smalltalk-80: The Language and Its Implementation, Addison-Wesley, Reading, MA, 1983.

J. Sun is now a research scientist at Alberta Research Council, Canada. He is currently responsible for developing industrial applications of intelligent systems. His research interests include intelligent systems, software development, product design and manufacturing, and measurement and control technology. J. Sun received his MSc degree in Manufacturing Engineering from University of Calgary, Canada, MSc degree in Measurement and Control Technology and Instrumentation from Hefei University of Technology, China, and BSc degree in Mechanical Engineering from Beijing University of Posts and Telecommunications, China.

D. Xue is an Associate Professor at Department of Mechanical and Manufacturing Engineering, University of Calgary. He received his PhD and MSc degrees in Precision Machinery Engineering from University of Tokyo, and his BSc degree in Precision Instrument Engineering from Tianjin University. His research interests include product life-cycle modeling and integrated concurrent design, intelligent planning, scheduling, and control, design methodology and intelligent CAD, engineering optimization, tolerance modeling, engineering applications of artificial intelligence, and engineering applications of object oriented programming. He is a member of ASME, SME, AAAI, and IPSJ.