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

J. Sun, D. Xue

Computers in industry 46 (2001) 189-207

2013. 02. 25

신 정 훈

# Contents

# Introduction

# Difficulty of scheduling

Most of the earlier(previous) developed scheduling methods have **difficulty for solving actual industrial problems**

- The industrial scheduling problems are **dynamic in nature**
  - **New orders** are received **continuously**
- The created **schedule** may be **changed**
  - To **reflect** the changes of **production orders and mfg. conditions**
    - Production **order change**
      - **Removal** of an order
      - **Insertion** of an order
    - Mfg. **condition change**
      - **Breakdown** of machines
      - **Sickness** of workers

# New methods and systems

Two different types of production scheduling

- **Predictive** scheduling
  - Creates the **optimal schedule** based on **given requirements and constraints** prior to production process
- **Reactive** scheduling(called **rescheduling**)
  - A process to **modify the created schedule** during the mfg. process **to adapt changes** in production environment

The **intelligent system approach**(proved effictive)

- For predictive scheduling
  - Aims at identifying the optimal schedule **through iterative search** process
- For reactive scheduling
  - To **revise only part** of the originally created **schedule**(responding environmental change)

# Problems

Many problems have to be solved for Predictive scheduling and Reactive scheduling

- mfg. requirements are **modeled** directly based upon **customer requirements**
  But **Product design descriptions and constraints are not** considered

- production scheduling mechanisms were primarily developed based on **centralized computing architecture**
  - difficulty in handling **complex manufacturing systems** that require knowledge and data to be **distributed at different locations**

# Previous research

- **Product descriptions and design constraints** are represented using a **feature-based modeling** approach(next chapter)
- **Manufacturing resources** 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

# Review of a previously developed predictive scheduling mechanism

- Previous developed system consists 3 sub-systems
  - product modeling sub-system, resource management sub-system, scheduling sub-system

# Product modeling sub-system

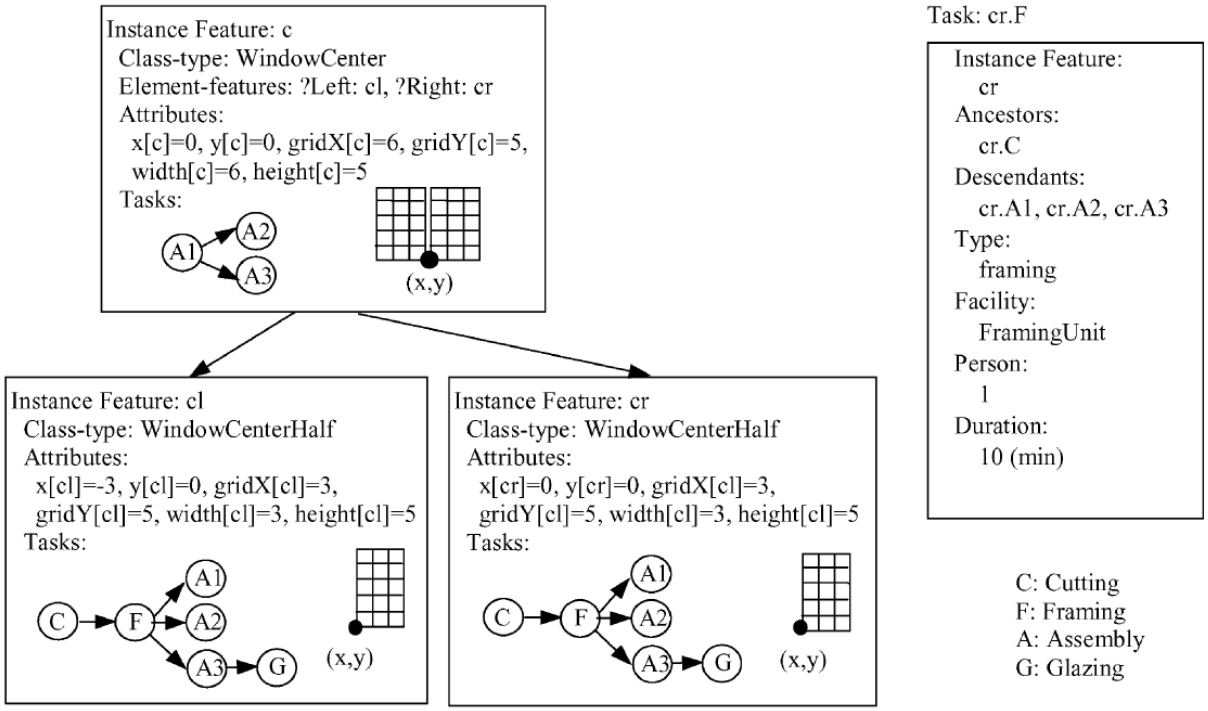



Fig. 1. Feature-based product and manufacturing requirement representation.

- product is modeled by primitives called **features**
  - **class** level : standard product libraries
  - **instance** level : special product data
- feature is **composed** of
  - **element features**, **attributes**, qualitative relations among features, and quantitative relations among attributes
- 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

# Resource management & Scheduling sub-system

- **resources are modeled as agents**(comes from the distributed modeling approach)
  - **facility** resource agent defined by : its **type**, manufacturing **functions**, and **time** constraints(available periods and unavailable periods)
  - **person** resource agent defined by : the **facilities** that the person is responsible for, and **time** constraints(available periods, regular schedule, and unavailable periods)

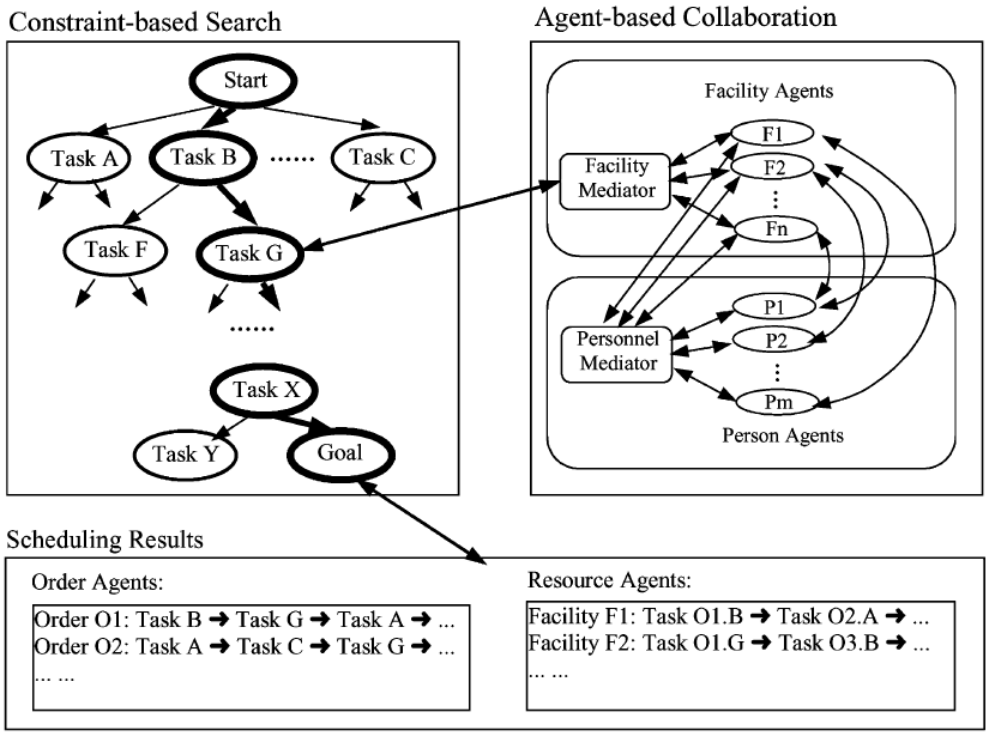



Fig. 2. Predictive scheduling using constraint-based search and agent-based collaboration.

- scheduling sub-system aims at identifying the **optimal schedule for the orders**
  - When an order is **received**, an order **agent** is then **created**
  - order agents **negotiate** with the **resource agents** using the corresponding design constraints and manufacturing requirements, which are preserved in the instance features
  - **Constraint-based search** and **agent-based collaboration approache**s are employed

# Scheduling sub-system(Constraint-based search)

- **The optimal sequence** of tasks → using **best-first search**(최상우선탐색: tree에서 깊이 우선 탐색을 최적화)
  - **Each node** in the search tree represents a **partial schedule** developed so far
  - **start node** describes an **empty** schedule
  - **goal node** describes the schedule in which **all the tasks** of the customer order have been **allocated** with required resources and timing parameter values.

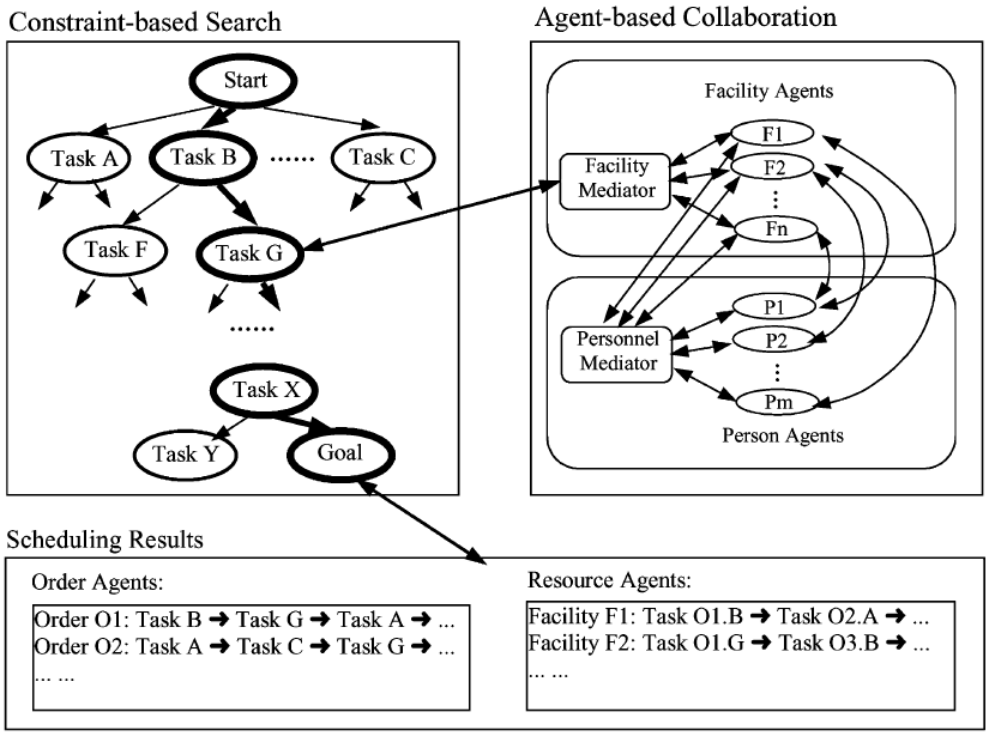



Fig. 2. Predictive scheduling using constraint-based search and agent-based collaboration.

- In **predictive** scheduling
  - each time the **best node is selected** for **generating its sub-nodes**
  - an **unscheduled task** is then **selected through collaboration** among relevant agents.
  - **Evaluation** to this node is conducted **using a heuristic** function
  - **scheduling results** are described by sequences of tasks that are preserved **in order agents and resource agents**

# Scheduling sub-system(Constraint-based search)

- Temporal **constraints** satisfied by the created schedule(in **predictive** scheduling)
  - 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**
  - A **task** can be carried out only when **all its ancestor tasks have been completed**

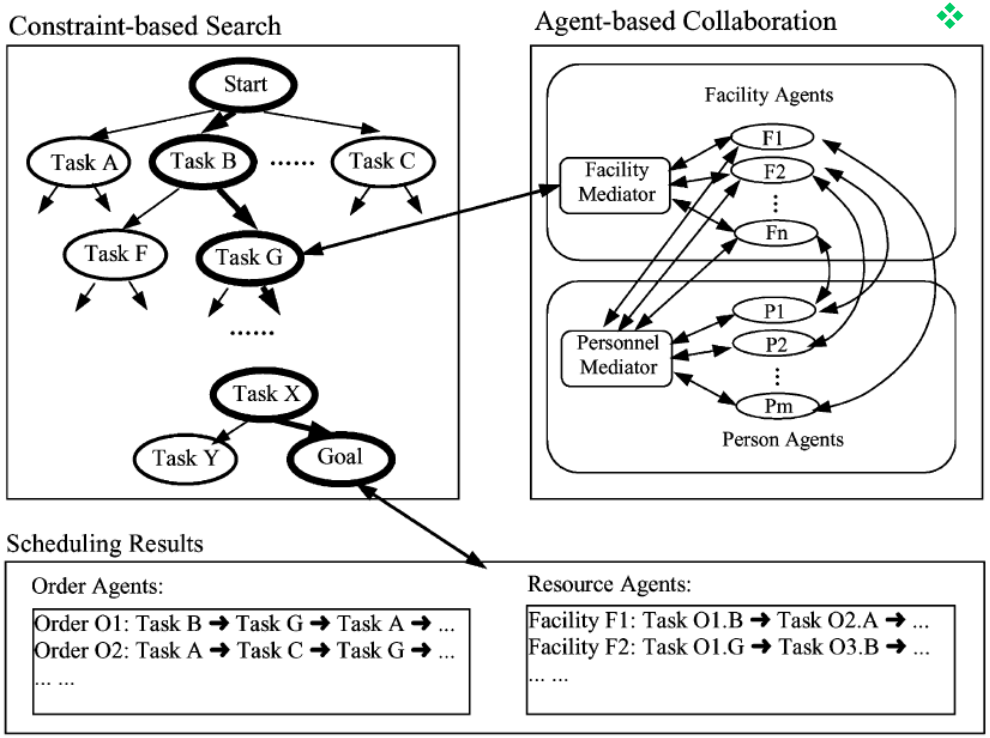



Fig. 2. Predictive scheduling using constraint-based search and agent-based collaboration.

- Two heuristic functions have been developed in this research
  - **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(**Backward**)
    - $F_{max}$ : the latest task finish time considering all the scheduled tasks of an order
  - **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(**Forward**)
    - $S_{min}$ : the earliest task start time considering all the scheduled tasks of an order

# Scheduling sub-system(Agent-based collaboration)

- agent-based collaboration using the **contract net protocol**(CNP : a task-sharing protocol in multi-agent systems)
  - CNP 5stage : Recognition → Announcement → Bidding → Awarding → Expediting
  - Two timing parameters of tasks(**start time and finish time**)are considered in scheduling

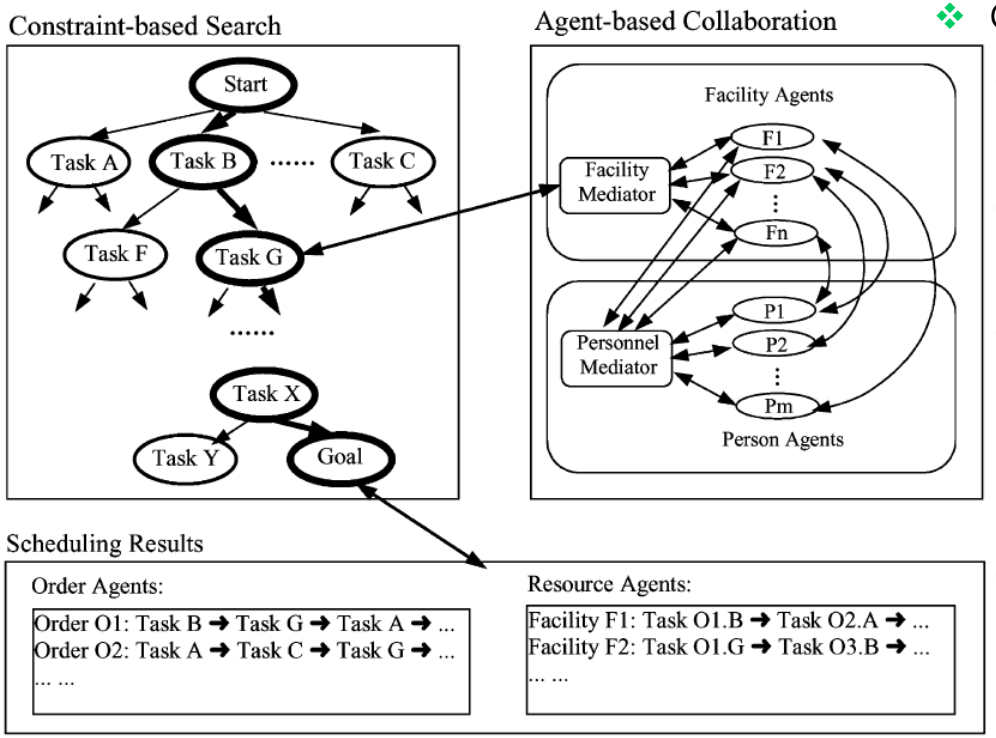



Fig. 2. Predictive scheduling using constraint-based search and agent-based collaboration.

- Collaboration(5stages of CNP)
  - facility mediator receives a to-be-scheduled task from the order agent
  - facility 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
  - facility mediator selects the facility that provides the best bid

    * order agent have a role of Expediting stage

# Architecture of an intelligent production scheduling system

# Architecture of an intelligent production scheduling system

- Predictive and Reactive scheduling in the whole intelligent production scheduling 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 is conducted to **modify the created schedule** for responding to the changes of customer orders and manufacturing conditions **during the production process**

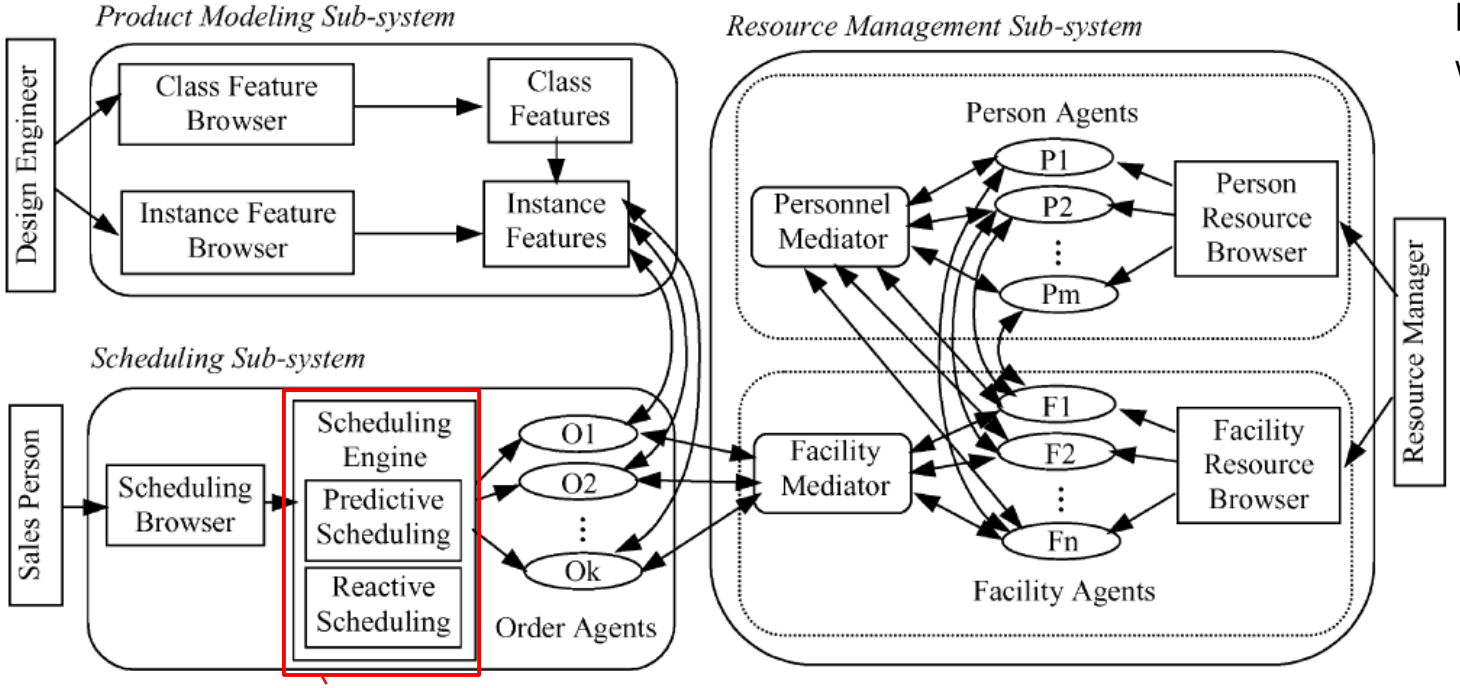



Fig. 3. Architecture of the intelligent production scheduling system.

dynamic reactive scheduling mechanism introduced in this research

- The intelligent predictive/reactive production scheduling system was implemented using **Smalltalk**
  - 제록스 파크(PARC)에서 앨런 케이(Alan Kay)와 동료들이 개발한 순수 객체지향 언어

# A dynamic reactive scheduling mechanism

- Reactive scheduling for customer order changes
- Reactive scheduling for manufacturing resource changes

# The objective of this research

- The objective of this research
  - 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

# Customer order changes

- Changes of customer orders are of two cases
  - cancellation of scheduled orders
  - insertion of urgent orders
- To improve the quality of the overall schedule
  - the tasks scheduled using the **due-time-based** scheduling strategy **could be moved forward** towards their **due-time measures**
  - 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 due to its urgent due-time requirement, some of the **previously scheduled tasks can be temporarily released**
- Rules
  - Orders scheduled **due-time-based** scheduling strategy are **rescheduled prior to** orders scheduled **earliest-delivery-time-based** scheduling strategy
  - In the **revised schedule**, the **sequence of tasks** for each to-be-rescheduled order remains the **same as** the sequence in the **original schedule**
  - In the **revised schedule**, each rescheduled task is **still allocated** with the **facility resource and person resource** that were **originally allocated**

# Customer order changes

Three step algorithm

- Step 1: **Initialize** for rescheduling
  - 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 **canceled order** should not considered in further rescheduling
  - **inserted order** is **scheduled first** using the **due-time-based** predictive scheduling
- Step 2: **Reschedule** orders that were previously scheduled using the **due-time-based** scheduling strategy

  2.1
  - 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(the **largest finish time** value is placed **at the beginning**)

# Customer order changes

- 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**(CNP)
  - **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) 설비기준 선행(precede) task
    (2) task sequence 기준 선행 task
    (3) 신규 스케줄과 상충되는 task
- 2.4
  - the **list** of the to-be-rescheduled tasks가 남아있으면 Step 2.1로 감
- 2.5
  - Recover all the tasks of the to-be-rescheduled orders(have not been rescheduled so far in the reactive scheduling)

# Customer order changes

- Step 3: **Reschedule** orders that were previously scheduled using the **earliest-delivery-time-based** scheduling strategy

  3.1

  - 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(the **smallest start time** value is placed **at the beginning**)

- 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 **agent-based collaboration mechanism**(CNP)
  - **current** to-be-rescheduled task **should be removed** from the list of the to-be-rescheduled tasks

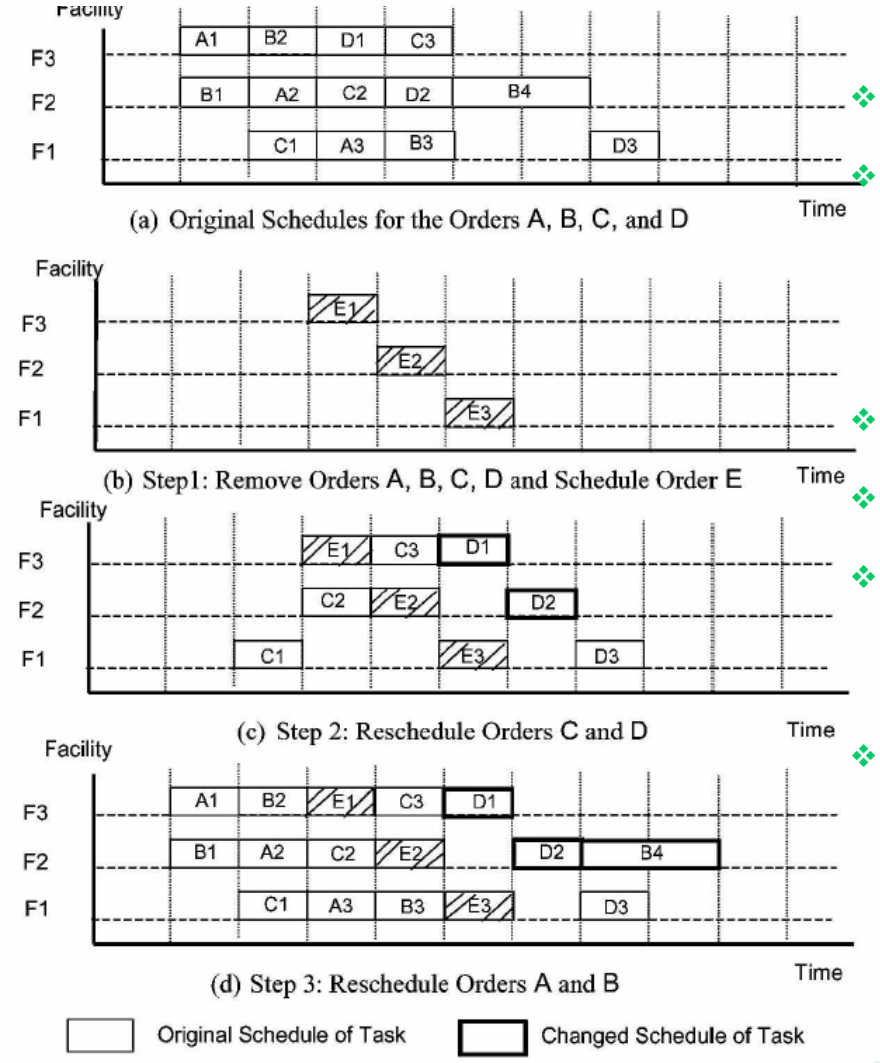
# Customer order changes

- 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) 설비기준 후행(follow) task
    (2) task sequence 기준 후행 task
    (3) 신규 스케줄과 상충되는 task
- 3.4
  - the **list** of the to-be-rescheduled tasks가 남아있으면 Step 3.2로 감
- 3.5
  - Recover all the tasks of the to-be-rescheduled orders(have not been rescheduled so far in the reactive scheduling)

# Customer order changes(Example)

Order A : A1 → A2 → A3
Order B : B1 → B2 → B3 → B4
Order C : C1 → C2 → C3
Order D : D1 → D2 → D3

A, B : earliest-delivery-time-based
C, D : due-time-based
E : inserted urgent order




(a) Original Schedules for the Orders A, B, C, and D

(b) Step1: Remove Orders A, B, C, D and Schedule Order E

(c) Step 2: Reschedule Orders C and D

(d) Step 3: Reschedule Orders A and B

Fig. 4. Reactive scheduling for customer order change.

- 2.1) orders C, D are rescheduled : previously using the due-time-based scheduling strategy → task D1, D2 are conflict with the tasks E1 and E2 → by finish time, D2 is first task
- 2.2) D3 recover → D2 is assigned with new timing parameter values
- 2.3) D2의 설비선행task인 C2와 seq선행인 D1을 (to-be-rescheduled) list에 포함(D1은 이미 포함됨)
  - D1 is assigned with new timing parameter values
  - C2, C3 recovered
- 2.4) list empty
- 2.5) recover all tasks
- 3.1) orders A, B are rescheduled : previously using the earliest-delivery-time-based scheduling strategy → task B4 is conflict with the revised schedule of D2 → B4 is first task
- 3.2) A1, A2, A3, B1, B2, and B3, whose finish time values precede the start time of B4 in the original schedule, are recovered → B4 is assigned with new timing parameter values

4Order 13Task 중에 3Task만 바뀜

# Customer order changes

## Reassigning timing parameter

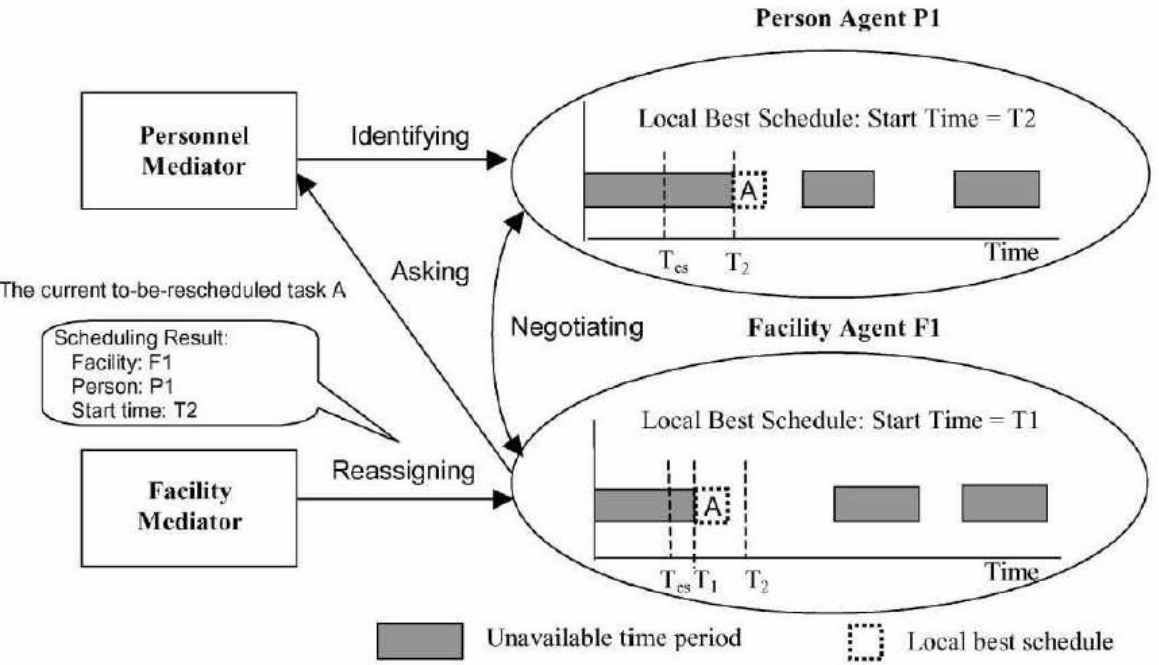



Fig. 5. An example of rescheduling timing parameter values through agent-based collaboration.

- Task A was originally scheduled using the earliest-delivery-time-based scheduling strategy
- **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**
- 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($\mathbf{S_{min}}$) in the earliest-delivery-time-based scheduling, while satisfying **all the manufacturing requirement and resource constraints**

# Mfg. resource change

- **Match-up** rescheduling approach
  - is also employed to **minimize the changes** to the originally created schedules, while **satisfying** the product and manufacturing **constraints**
- Rules
  - first tries to **move the tasks** that are **affected directly** by the **resource condition changes to other resources without changing the timing parameter** values
  - If the **alternative resources can not be identified** for the affected tasks, **match-up-based rescheduling** is then conducted(**due-time-based** 로 scheduling된 order를 **earliest-delivery-time-based**로 scheduling된 order보다 먼저 rescheduling)
  - **due-time-based** 로 scheduling된 일부 order가 **due-time제약을 만족하지 못하면**, **due-time제약이 있는 모든 Task의 rescheduling후**에 리소스 변경의 **직접 영향을 받은 Order의 due-time을 수정하여 reschedule**
  - Revised schedule에서 스케줄 대상 목록의 Task순서는 original schedule과 동일(to satisfy the task precedence constraints while improving the rescheduling efficiency)
  - 영향을 받은 Task의 대안리소스를 찾지 못하면 원래 리소스에 할당함

# Mfg. resource change

Five step algorithm

- Step 1: Identify the **alternative resources**
  - Resource change의 영향을 받은 모든 Task에 대하여 **timing parameter의 변경 없이 대안리소스로 옮길 수 있는지 확인**하고, 모두 가능하면 종료한다
- Step 2: **Initialize** for rescheduling
  - 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
- Step 3: **Reschedule** orders that were previously scheduled using the **due-time-based** scheduling strategy 3.1
  - The tasks, whose original schedules are **affected directly by resource condition changes**, should be considered as the to-be-rescheduled tasks
  - **Sort** the list of the to-be-rescheduled tasks according to the **finish time** values(the **largest finish time** value is placed **at the beginning**)

# Mfg. resource change

- 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 **agent-based collaboration mechanism**(CNP)
  - **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) 설비기준 선행(precede) task
    (2) task sequence 기준 선행 task
    (3) 신규 스케줄과 상충되는 task
- 3.4
  - the **list** of the to-be-rescheduled tasks가 남아있으면 Step 3.2로 감
- 3.5
  - Recover all the tasks of the to-be-rescheduled orders(have not been rescheduled so far in the reactive scheduling)

# Mfg. resource change

- 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**
  - **the directly affected orders** that were previously scheduled using the **due-time-based** scheduling strategy **should not be considered temporarily** → directly affected orders를 제외한 것들을 Step3로 보내서 rescheduling한 뒤 directly affected orders는 due-time을 조정하여 reschedule

- Step 5: **Reschedule** orders that were previously scheduled using the **earliest-delivery-time-based** scheduling strategy

  5.1
  - The **tasks** whose original schedules are **affected directly by resource condition changes** and whose **original schedules are in conflict with the revised schedules** are considered as the to-be-rescheduled tasks
  - **Sort** the list of the to-be-rescheduled tasks according to the **start time** values(the **smallest start time** value is placed **at the beginning**)

# Mfg. resource change

- 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 **agent-based collaboration mechanism**(CNP)
  - **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) 설비기준 후행(follow) task
    (2) task sequence 기준 후행 task
    (3) 신규 스케줄과 상충되는 task
- 5.4
  - the **list** of the to-be-rescheduled tasks가 남아있으면 Step 5.2로 감
- 5.5
  - Recover all the tasks of the to-be-rescheduled orders(have not been rescheduled so far in the reactive scheduling)

# Mfg. resource change(Example)

Order A : A1 → A2 → A3
Order B : B1 → B2 → B3 → B4
Order C : C1 → C2 → C3
Order D : D1 → D2 → D3

A, B : earliest-delivery-time-based
C, D : due-time-based
F2 : Emergency maintenance

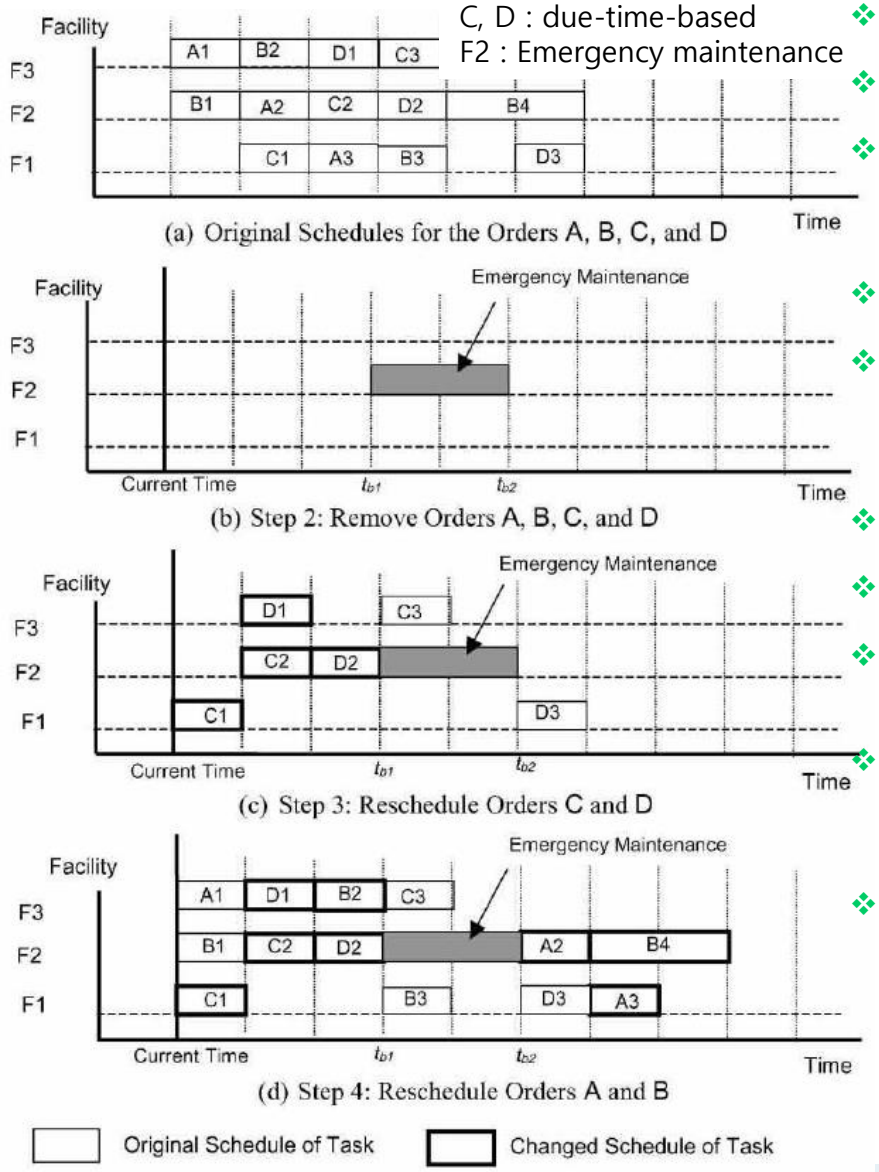



Fig. 6. Reactive scheduling for manufacturing resource change.

- 1) no alternative facilities for affected tasks D2 and B4
- 2) The time slot is recorded as the unavailable period for facility F2
- 3.1) orders C, D are rescheduled : previously using the due-time-based scheduling strategy → task D2 is affected directly by resource condition change → D2 is first task
- 3.2) D3 recover → D2 is assigned with new timing parameter values
- 3.3) D2의 설비선행task인 C2와 seq선행인 D1을 (to-be-rescheduled) list에 포함
  - D2 is assigned with new timing parameter values
  - C3 recovered, C2 is assigned with new timing parameter values
- 3.4) list empty
- 3.5) recover all tasks of orders C and D
- 4) since all the orders can be rescheduled in Step 3, rescheduling process goes to Step 5
- 5.1) orders A, B are rescheduled : previously using the earliest-delivery-time-based scheduling strategy → task B4 is affected by the resource condition change and tasks A2 and B2 are in conflict with the revised schedules
- 5.2) recover A1, B1 → A2 is assigned with new timing parameter values → B2, A3, B4 is assigned with new timing parameter values → B3, D3 recover

4Order 13Task 중에 8Task만 바뀜

# Case study examples

# Case study examples

Table 1
Manufacturing requirements for three instance features

| Instance features | Tasks | Ancestors | Descendants | Types | Facilities | Du |
|---|---|---|---|---|---|---|
| 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 |

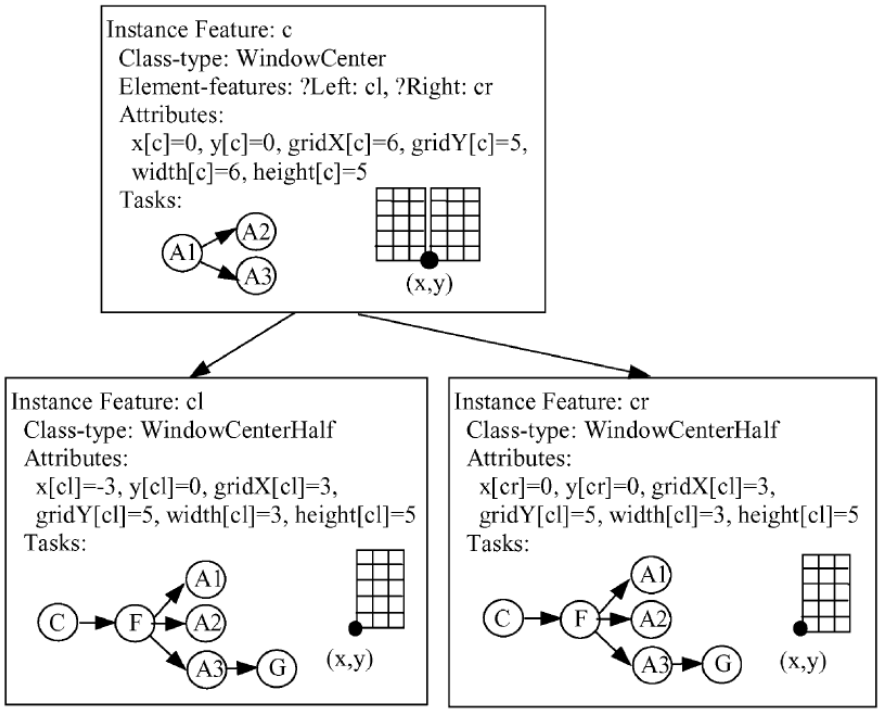
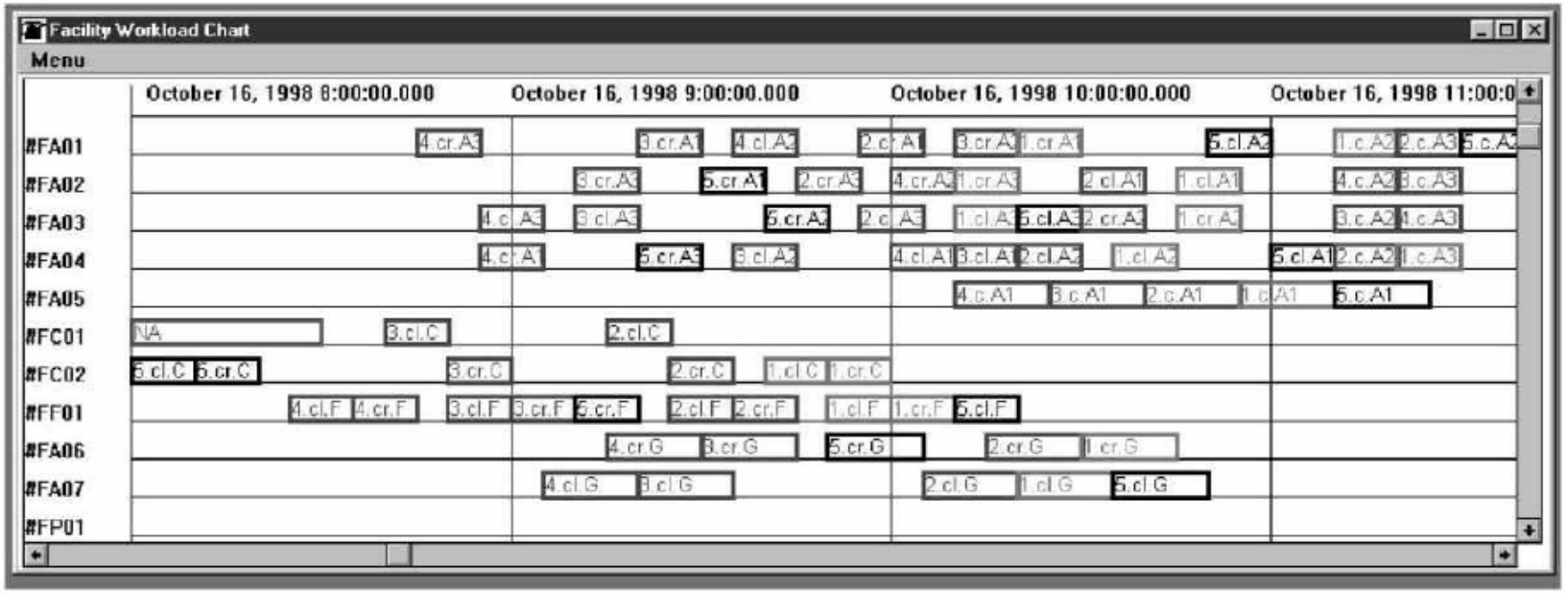
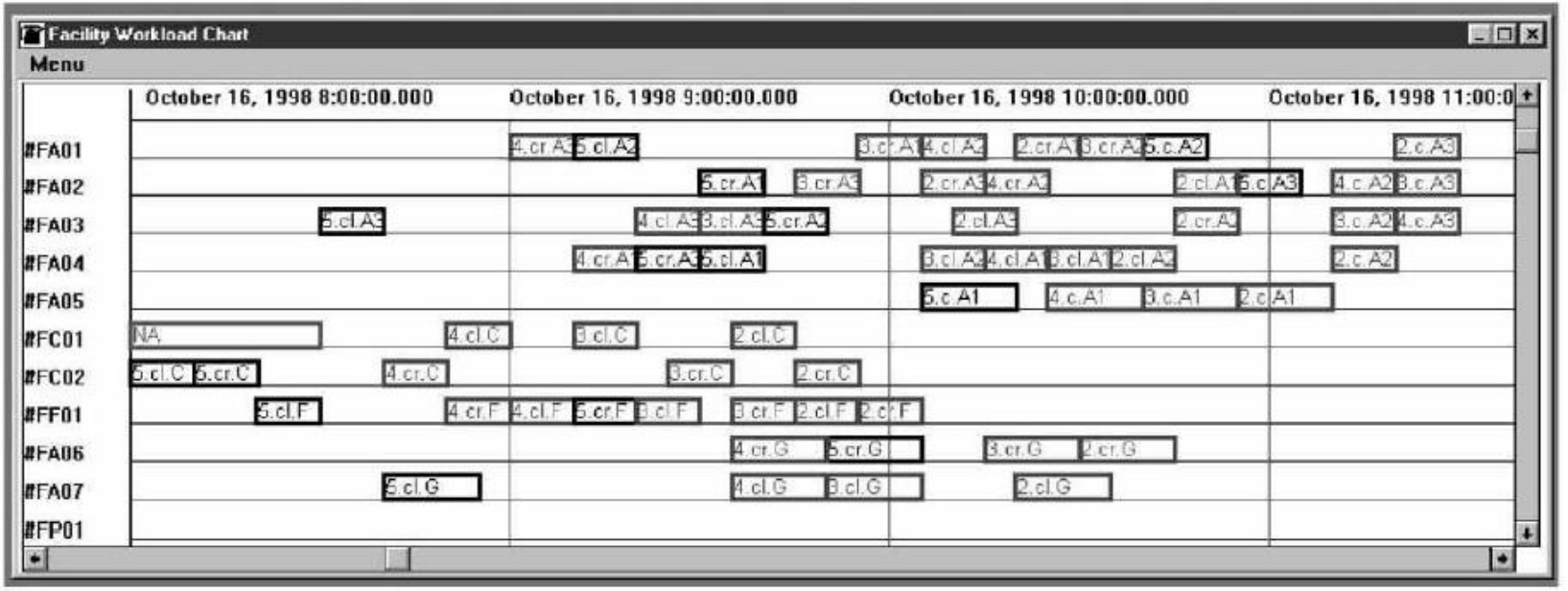



Table 2
Definitions of facility resource agents

| 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 |

# Case study examples-1




- 가정
  - Order1~4 : due-time-based scheduling, order5 : earliest-delivery-time-based scheduling
  - Order1 canceled
- 결과
  - Order2~4는 due-time-based scheduling 하였으므로 Due-time measure를 사용하여 toward(forward)로 shift
  - Order5는 earliest-delivery-time-based scheduling 하였으므로 release-time을 사용하여 backward로 shift

# Case study examples-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 | Cancel | 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.

- 가정
  - Order1~4 : due-time-based scheduling,
    Order5 : earliest-delivery-time-based scheduling
  - Order1 canceled
- 결과
  - Order1은 cancel
  - Order2~4는 due-time-based scheduling 하였으므로 Due-time measure를 사용하여 toward(forward)로 shift → Release time 늦어짐
  - Order5는 earliest-delivery-time-based scheduling 하였으므로 release-time을 사용하여 backward로 shift → Completion time 당겨짐
  - 48/60(revised/total)

# Case study examples-2

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.

- 가정
  - Order1~3 : due-time-based scheduling, Order4 : earliest-delivery-time-based scheduling
  - Order5 inserted(urgent due time)
- 결과
  - Order1~3은 due-time-based scheduling 하였으므로 Due-time measure를 사용하여 toward(forward)로 shift 해야 하나 constraint의 허용범위 내에서 반대로 shift → Release time 당겨짐
  - Order4는 earliest-delivery-time-based scheduling 하였으므로 release-time을 사용하여 backward로 shift 해야 하나 constraint의 허용범위 내에서 반대로 shift → Completion time 늦어짐
  - 중간에 비는 시간에 Order5가 insert
  - 47/60(revised/total)

# Case study examples–3

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.

- 가정
  - Order1~2 : due-time-based scheduling,
    Order3~4 : earliest-delivery-time-based scheduling
  - Facility FF01이 9:00~9:20 동안 not available
- 결과
  - Order2의 Release time 당겨짐
  - Order4의 Completion time 늦어짐
  - 17/60(revised/total)

# Case study examples–4

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.

- 가정
  - Order1~2 : due-time-based scheduling,
    Order3~4 : earliest-delivery-time-based scheduling
  - Person PM03, who is responsible for Facility FF01
    8:30~9:00 동안 not available
- 결과
  - Order2, 4의 Release time 당겨짐
  - Order2, 3, 4의 Completion time 늦어짐
  - 36/60(revised/total)

# Conclusions

# Conclusions

- presented **development of a dynamic reactive scheduling mechanism** for an intelligent production scheduling system.
  - **works together with** a previously developed **predictive 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**
  - Used The **match-up**(modify only part of the previously created schedule) **and agent-based collaboration approaches**
  - **integrates** production **scheduling function** and product **design function** into the same environment


감사합니다

# 논문의 의의

- 스케줄링 체계에 대한 시스템 아키텍쳐 서술에 대한 논문 서치(for ISGMA)
  - 지난번 논문보다 시스템 아키텍쳐 서술에 대한 구체성 및 방법론이 명확함
  - 알고리즘 설명에 대한 방법을 배움

- Agent 기반의 스케줄링 시스템에 대한 보다 구체적인 이해가 가능하였음
  - 알고리즘 & 아키텍쳐 & 코드화 가능
  - 프로젝트 스케줄링과 생산스케줄링의 중간(?) → i-mfg의 스케줄러와 자율적응의 스케줄러의 중간정도로 인식