Computer Integrated Manufacturing Systems Vol. 8, No. 2, pp. 93–103, 1995
Copyright © 1995 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0951-5240/95 $10.00 + 0.00

0951–5240(95)00003–8

# Development and implementation of a dispersed decision process: an FMS scheduling example

**Shung-Kuang Kung* and James R. Marsden†**

**Department of Management Information Systems, College of Business, Chung-Yuan Christian University, Chung-Li, Taiwan*
*†Department of Operations and Information Management, 368 Fairfield Road, U-41 IM, School of Business Administration, University of Connecticut, Storrs, CT 06269–2041, USA*

**An automated process was constructed which provides a platform for conducting analyses in general distributed multi-participant decision making environments, including the special case of FMS scheduling. The detailed example illustrates the use of the tool for analysing distributed decision making in a multi-machine, multi-task FMS setting. The example relates a controlled laboratory experiment using human 'operators' interacting in an electronic auction. The generality of our automated process supports this type of experimentation, as well as complex simulations using computerized expert systems as 'participants', simulations that enable us to perform detailed comparisons between the performance of the distributed process and existing centralized FMS scheduling heuristics.**

Keywords: flexible manufacturing systems, distributed decision making, laboratory experiment

## Introduction

Production decision processes and mechanisms may conjure up physical relations and tangible products. But what is there that is truly unique about these settings? Are they truly singular, or are they set apart because of restrictive definitions that we choose to characterize them? Consider flexible manufacturing systems (FMS). One commonly cited definition is that of Kusiak[1]: a set of machine tools linked by a material handling system, all controlled by a computer system. Suppose, instead, that we adopt a much broader approach, defining an FMS as *a production system where one or more machines are capable of performing more than one task* (where suitably broad definitions of 'machine' and 'task' are intended). We note that this definition is sufficiently expansive to include a principal (major contractor) dealing with numerous potential subcontractors.

In what follows, we demonstrate how, using such a broad definitional approach, we are able to construct an automated process which provides a platform for conducting analyses in general distributed multi-participant decision making environments, including the special case of FMS scheduling. Our detailed example, provided below, illustrates the use of the tool for analysing distributed decision making in a multi-machine, multi-task FMS setting. The example relates a controlled laboratory experiment using human 'operators' interacting in an electronic auction. The generality of our automated process supports this type of experimentation as well as complex simulations using computerized expert systems as 'participants'.

Though much of our motivation for developing the tool rested with our desire to be able to investigate thoroughly alternatives to centralized scheduling in existing and likely future FMS environments, we quickly realized that the approach held possibilities for investigating the performance of more general distributed decision making environments. Whether dealing with extremely complex (NP-complete) FMS scheduling problems or with less demanding assignment problems, a key determination must be the performance comparison of centralized *versus* decentralized decision making processes. As detailed below, our automated process provides a convenient tool for analysing the performance of distributed processes.

The next section sets forth the specifics of the general environment we model and the automated process we have developed to perform analyses within this general environment. Two FMS example experiments we conducted are then detailed and the experimental outcomes summarized. Finally, our concluding remarks and suggestions are provided.

## Research tool

The research tool we implement is a flexible operationalization of the approach set out in Shaw and Whinston[2]. These authors 'followed the distributed artificial intelligence framework' of Davis and Smith[3] and Smith[4] using the contract net approach to meet three specific requirements they detail[2]:

1. A model of the problem-solving process that, through the communications network, dynamically distributes tasks among cells.
2. Design of an interface language that enables effective communication among cell hosts.
3. Programming and execution of this problem-solving process at each cell in a decentralized manner.

In our operationalization, the general or system problem is decomposed by a manager (human or automated) into sub-problems which are auctioned across a network to contractors (human or machine) for solution or for further decomposition. If a second round of decomposition is used, then individual contractors act as managers for such subsequent rounds. Using the broad definition of an FMS set out in our introductory remarks, the following are the roles and activities we incorporated:

- *Central auctioneer/manager* – tracks job arrivals, decomposes jobs into tasks, announces tasks to be auctioned, collects bids, assigns tasks, provides incentives and/or penalties for behaviour of agents in system, monitors and manages processes including re-bidding of tasks due to machine breakdown or queue constraint violations;
- *Contractor/agent* – monitors individual node capabilities and activities, determines own bids for auctioned tasks, assigns tasks won to individual machines at own node, monitors and completes awarded tasks.

We use the terms *task* and *decision* interchangeably. The process we operationalized can be viewed as a decentralized decision or task completion process. The auctioneer or central manager is charged with completing a set of jobs, each possibly composed of several individual tasks. New jobs may arrive in prespecified arrays, but are generally expected to arrive randomly. The central manager decomposes each job into tasks (for which precedence constraints may exist), conducts auctions of the tasks, and assigns them to agents with the lowest bids for each task (decentralized decision maker). The central manager provides incentives (payments for tasks) typically set equal to the winning bid amount, and imposes penalties on agent behaviour that violates process constraints.

Each individual agent operates a node which possesses specific task capabilities. Each agent can bid on auctioned tasks, and is responsible for completion of awarded tasks. Where multiple task capabilities exist, agents must assign awarded tasks to specific machines at their nodes. In situations where agents are grouped or teamed, intermediate managers for each team act as conduits with the central manager, sending messages (e.g. bids) and receiving instructions and auction results.

We developed our operationalization to be functional on a low-cost, commonly available system. We used an IBM token ring local area network (LAN) comprised of IBM PS-2 Model 50s and 80s (Intel 286 and 386 machines, respectively) connected using multi-station access units (MAU). The network has a 4 Mbit transmission rate on twisted-pair wiring using a token access method within a ring topology. The LAN adapter provides 16 Kbyte of on-board shared RAM. Even with such a relatively modest platform, we were able to develop an easy-to-operate, functional and flexible scheduling process with distributed decision making.

Programming was in C and made liberal use of NetBIOS commands. Agents and the auctioneer communicate with each other through the token ring LAN. The shell was installed in the Department of Decision Science and Information System's MIS Research Lab at the University of Kentucky. In the lab, workstations (IBM model 50s) and file servers (IBM model 80s) were connected together using MAUs. One file server acted as the auctioneer, while the workstations were used to provide the auction interface and local DSS for the agents. Auctions were conducted through message passing between the auctioneer and agents, using the Datagram Transmission mechanism provided by IBM NetBIOS to transmit the required messages.

Prototyping took place over a four month period, and concentrated on the development of an informative and easy to use DSS for each node. Doctoral students in the Department of Decision Science and Information Systems were used as test subjects in repeated trials for modifying the DSS and interface. After each trial, subjects were queried concerning information needs and relevance. Our goal was to locate the highest ranked information in convenient, on-screen areas. Lower ranked information (still commonly indicated as relevant by subjects) was relegated to 'on request' status, accessible as pop-up screens. After repeated testing over the four month period, screen design and pop-up information availability appeared appropriate to subjects. Additional test runs were conducted and subject views did not change.

*Figure 1* provides a typical node-DSS screen for agents. The area marked on the first line of the screen provides updated information on current status, and includes the following: balance (i.e. initial funds plus profits made minus losses incurred; current balance in the example screen is \$606.56), queue size permissible at any machine (set at 3 for this example), penalty fee for violating queue constraint or for machine breakdown leading to forced re-bidding of task (set at \$10 per occurrence), time remaining to submit bid (91 seconds here). There are four functions that can be activated using function keys:

- Key F5 – displays task (production) capabilities (what tasks each machine can do) at the node – *Figure 2* provides the screen format when this feature is activated;
- Key F6 – enables agent to assign a task to alternative machines – for a specified task, displays possible

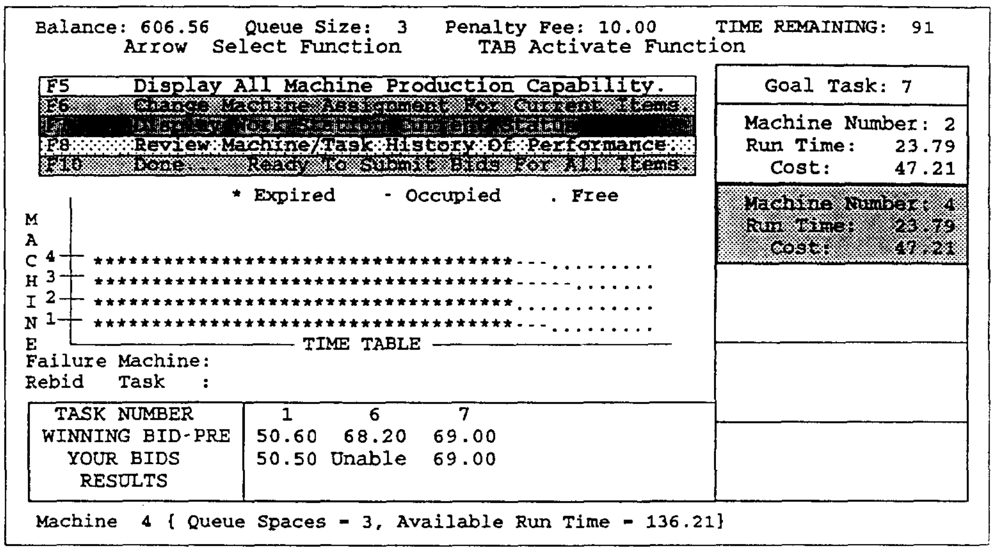
alternative machines and associated costs and run times for completing the task at each alternative (information is displayed on right side of screen as in *Figures 1* and *3* – activation of F6 permits toggling in this area of the screen and the highlighting of a machine selected for a task – *Figure 3* indicates assignment to machine 2 while *Figure 1* indicates assignment to machine 4);

- Key F7 – displays the current status of all machines at the node (workstation) – see *Figure 4*;
- Key F8 – initially displays pop-up (*Figure 5*) offering choice of pop-up providing the workstation's history for each task (*Figure 6*), the history of each machine at the workstation (*Figure 7*), and the history of winning bids for each task capable of being performed at the workstation (*Figure 8*).

The function key F10 simply signifies a 'finished with analysis, submit bids' function that triggers the sending of bids to the auctioneer.

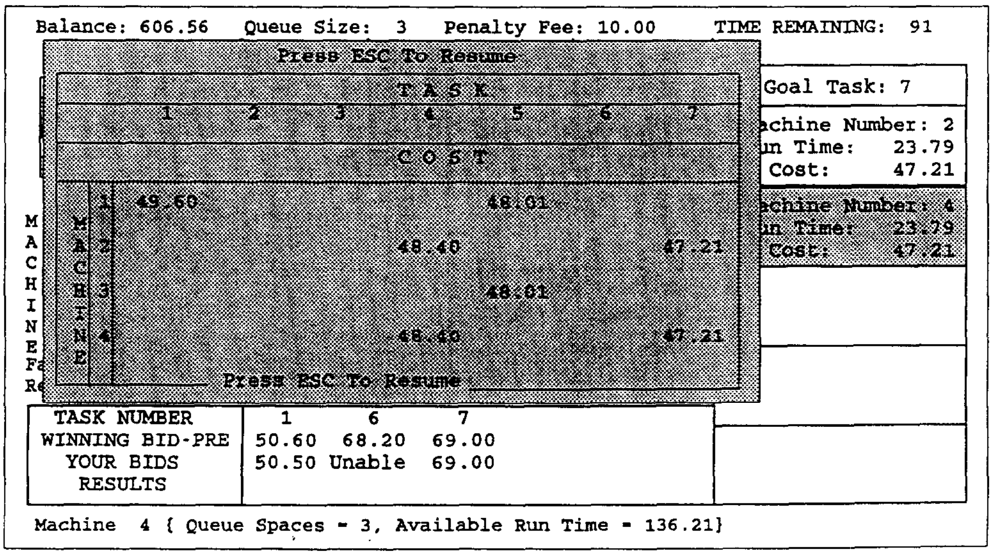
Time allocation is provided in the middle of the agent interface screen (*Figure 1*). This indicates, for each machine at the node, how much of the total available production time (commonly an eight hour shift) is already used or committed because of queued tasks.

The bidding block is presented toward the bottom of the agent's interface screen. When an auction is announced, messages sent out across the network fill the top two lines of this block – **TASK NUMBER**(s) and **WINNING BIDS PRE**(VIOUSLY). Each agent then enters the amount he/she wishes to bid on each task. In the example presented in *Figure 1*, the agent has chosen to bid 50.50 on task 1, a task for which the previous winning bid was 50.60. As the agent toggles through the tasks being auctioned which the machines at his node are capable of performing (tasks 1 and 7 in *Figure 1*), the right side of the screen automatically changes to provide the information on machine capabilities, run time, and costs for each task. For example, in *Figure 1*



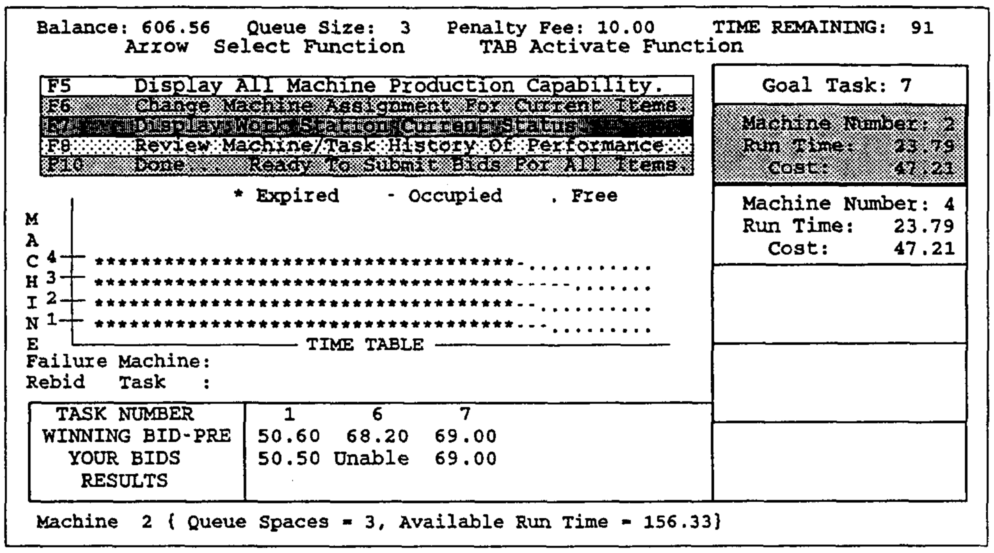
```
Balance: 606.56   Queue Size:  3   Penalty Fee: 10.00     TIME REMAINING:  91
       Arrow  Select Function        TAB Activate Function

F5     Display All Machine Production Capability.        Goal Task: 7
F6     Change Machine Assignment For Current Items.
F7     Display Node's [illegible] Current Status          Machine Number: 2
F8     Review Machine/Task History Of Performance.        Run Time:    23.79
F10    Done...   Ready To Submit Bids For All Items.      Cost:        47.21

                 * Expired     - Occupied     . Free      Machine Number: 4
M                                                          Run Time:    23.79
A                                                          Cost:        47.21
C 4 ************************************---.........
H 3 ************************************----.......
I 2 ************************************.........
N 1 ************************************---.........
E  ───────────────── TIME TABLE ──────────────────
Failure Machine:
Rebid   Task    :

 TASK NUMBER      |  1      6      7
 WINNING BID-PRE  | 50.60  68.20  69.00
  YOUR BIDS       | 50.50 Unable  69.00
   RESULTS        |

Machine  4 { Queue Spaces - 3, Available Run Time - 136.21}
```

**Figure 1** Typical node DSS pop-up screen



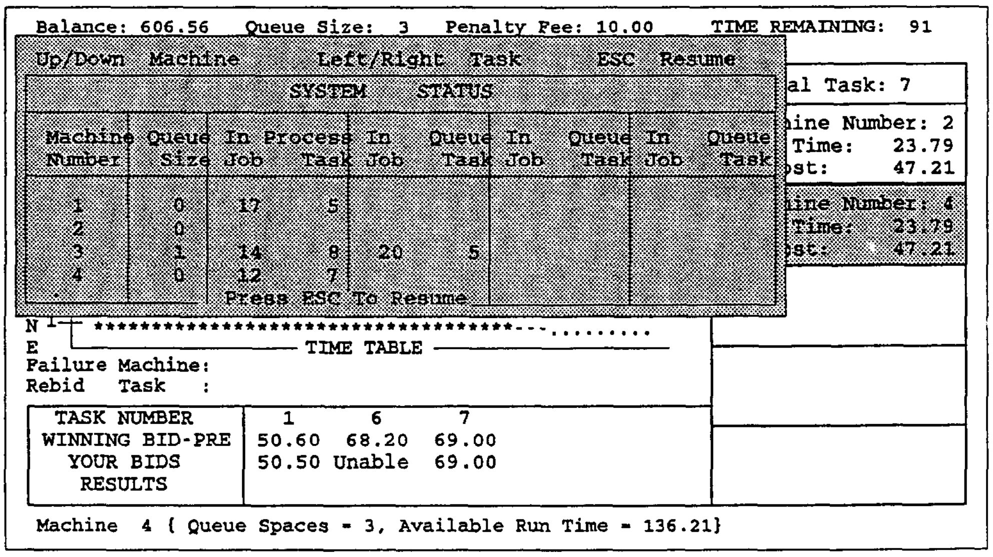
```
Balance: 606.56   Queue Size:  3   Penalty Fee: 10.00     TIME REMAINING:  91
               Press ESC To Resume
                     T A S K                              Goal Task: 7
        1     2     3     4     5     6     7
                     C O S T                              chine Number: 2
                                                          n Time:    23.79
  M  1  49.60                   48.01                     Cost:      47.21
  A  2                    48.40               47.21
  C                                                       chine Number: 4
  H  3                          48.01                     n Time:    23.79
  I                                                       Cost:      47.21
  N  4                    48.40               47.21
  E
            Press ESC To Resume

 TASK NUMBER      |  1      6      7
 WINNING BID-PRE  | 50.60  68.20  69.00
  YOUR BIDS       | 50.50 Unable  69.00
   RESULTS        |

Machine  4 { Queue Spaces - 3, Available Run Time - 136.21}
```

**Figure 2** Productivity pop-up screen



```
Balance: 606.56   Queue Size:  3   Penalty Fee: 10.00    TIME REMAINING:  91
        Arrow  Select Function        TAB Activate Function

F5      Display All Machine Production Capability.      Goal Task: 7
F6      Change Machine Assignment For Current Items.
F7      Display Work Station Current Status.             Machine Number: 2
F8      Review Machine/Task History Of Performance.      Run Time:    23.79
F10     Done...  Ready To Submit Bids For All Items.       Cost:      47.21

                 * Expired     - Occupied     . Free     Machine Number: 4
M                                                        Run Time:    23.79
A                                                          Cost:      47.21
C 4-- ***********************************-...........
H 3-- ***********************************----.......
I 2-- ***********************************--.........
N 1-- ***********************************---........
E     ----------------- TIME TABLE -----------------
Failure Machine:
Rebid   Task    :

  TASK NUMBER      1      6      7
WINNING BID-PRE  50.60  68.20  69.00
   YOUR BIDS     50.50  Unable 69.00
    RESULTS

Machine   2 { Queue Spaces = 3, Available Run Time = 156.33}
```

**Figure 3** Machine assignment screen



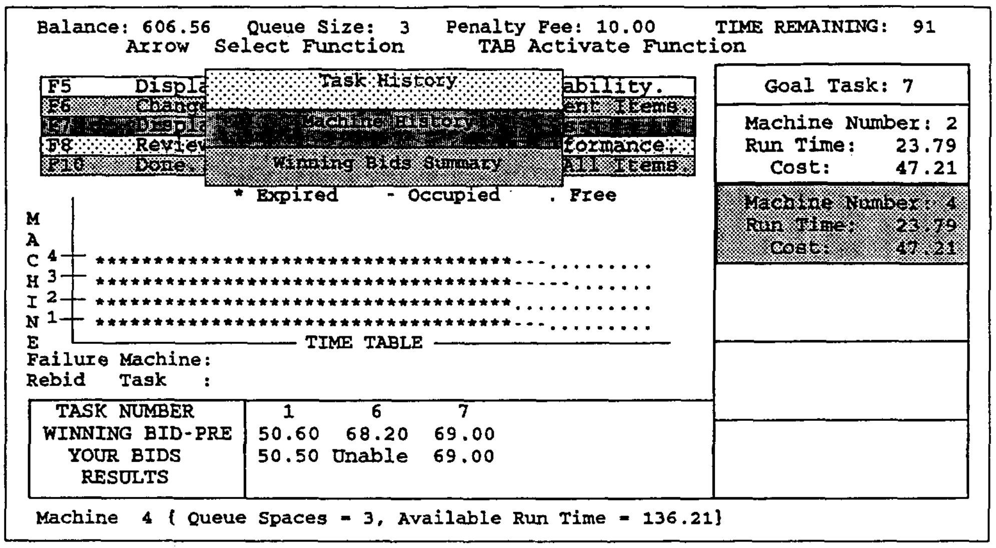
```
Balance: 606.56   Queue Size:  3   Penalty Fee: 10.00    TIME REMAINING:  91
Up/Down  Machine      Left/Right  Task        ESC  Resume
                      SYSTEM   STATUS                          al Task: 7
```

| Machine Number | Queue Size | In Process Job | In Process Task | In Job | Queue Task | In Job | Queue Task | In Job | Queue Task |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 17 | 5 | | | | | | |
| 2 | 0 | | | | | | | | |
| 3 | 1 | 14 | 8 | 20 | 5 | | | | |
| 4 | 0 | 12 | 7 | | | | | | |

```
                 Press ESC To Resume
N 1-- ***********************************---........
E     ----------------- TIME TABLE -----------------
Failure Machine:
Rebid   Task    :

  TASK NUMBER      1      6      7
WINNING BID-PRE  50.60  68.20  69.00
   YOUR BIDS     50.50  Unable 69.00
    RESULTS

Machine   4 { Queue Spaces = 3, Available Run Time = 136.21}
```

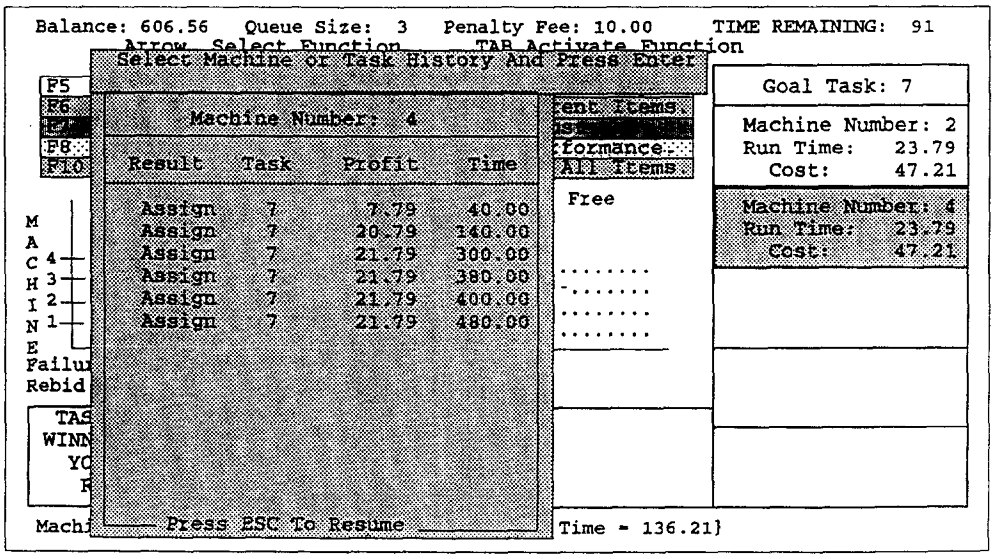
(Right panel partially visible: "ine Number: 2 / Time: 23.79 / st: 47.21 / ine Number: 4 / Time: 23.79 / st: 47.21")

**Figure 4** System status pop-up screen



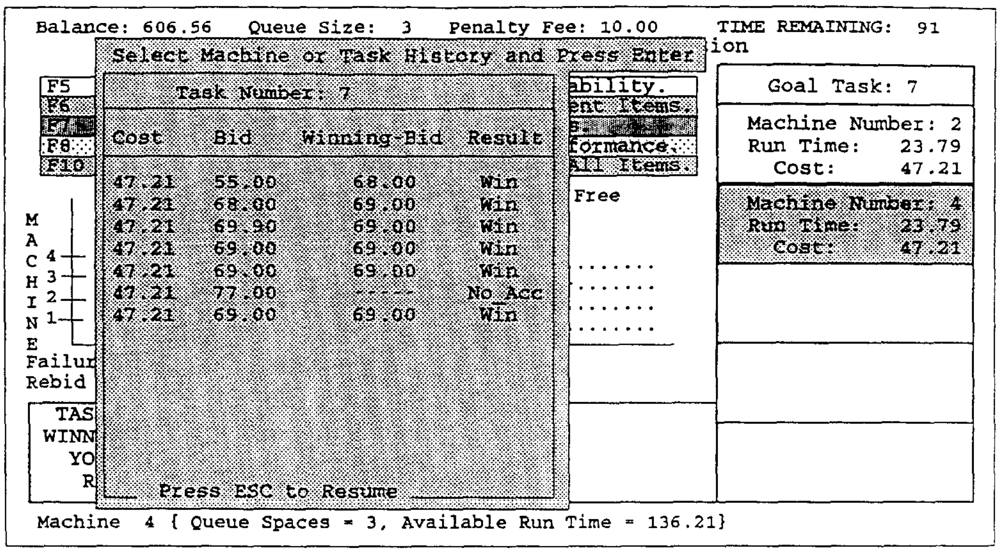
```
Balance: 606.56   Queue Size:  3   Penalty Fee: 10.00    TIME REMAINING:  91
        Arrow  Select Function        TAB Activate Function

F5      Displa        Task History          ability.    Goal Task: 7
F6      Change                              ent Items.
F7      Displ        Machine History                     Machine Number: 2
F8      Review                              formance.    Run Time:    23.79
F10     Done.      Winning Bids Summary     All Items.     Cost:      47.21
                 * Expired     - Occupied     . Free
M                                                        Machine Number: 4
A                                                        Run Time:    23.79
C 4-- ***********************************---.........     Cost:      47.21
H 3-- ***********************************----.......
I 2-- ***********************************...........
N 1-- ***********************************---........
E     ----------------- TIME TABLE -----------------
Failure Machine:
Rebid   Task    :

  TASK NUMBER      1      6      7
WINNING BID-PRE  50.60  68.20  69.00
   YOUR BIDS     50.50  Unable 69.00
    RESULTS

Machine   4 { Queue Spaces = 3, Available Run Time = 136.21}
```

**Figure 5** History menu pop-up screen



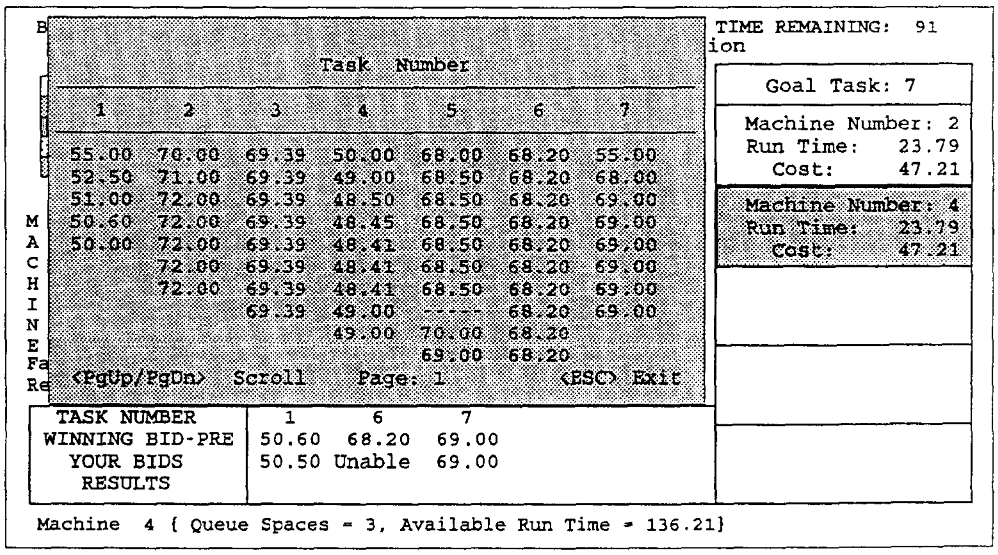
```
Balance: 606.56   Queue Size:  3   Penalty Fee: 10.00    TIME REMAINING:  91
      Arrow  Select Function        TAB Activate Function
     Select Machine or Task History And Press Enter
F5                                                         Goal Task: 7
F6          Machine Number:  4                 ent Items.
F7                                             s.          Machine Number: 2
F8       Result    Task     Profit    Time     formance.   Run Time:    23.79
F10                                            All Items.   Cost:       47.21
         Assign     7        7.79     40.00    Free
M        Assign     7       20.79    140.00                Machine Number: 4
A        Assign     7       21.79    300.00                Run Time:    23.79
C 4      Assign     7       21.79    380.00    .......      Cost:       47.21
H 3      Assign     7       21.79    400.00    -.......
I 2      Assign     7       21.79    480.00    .......
N 1                                            .......
E
Failu
Rebid
 TAS
WINN
  YO
   R
Mach         Press ESC To Resume               Time = 136.21}
```

**Figure 6** Machine history pop-up screen



```
Balance: 606.56   Queue Size:  3   Penalty Fee: 10.00    TIME REMAINING:  91
     Select Machine or Task History and Press Enter     ion
F5          Task Number: 7                     bility.     Goal Task: 7
F6                                             ent Items.
F7       Cost      Bid     Winning-Bid  Result  s.         Machine Number: 2
F8                                             formance.   Run Time:    23.79
F10      47.21    55.00      68.00      Win    All Items.   Cost:       47.21
M        47.21    68.00      69.00      Win    Free
A        47.21    69.90      69.00      Win                Machine Number: 4
C 4      47.21    69.00      69.00      Win    .......     Run Time:    23.79
H 3      47.21    69.00      69.00      Win    .......      Cost:       47.21
I 2      47.21    77.00      -----      No_Acc .......
N 1      47.21    69.00      69.00      Win    .......
E
Failu
Rebid
 TAS
WINN
  YO
   R         Press ESC to Resume
Machine  4 { Queue Spaces = 3, Available Run Time = 136.21}
```

**Figure 7** Task history pop-up screen



```
B                                                       TIME REMAINING:  91
                                                        ion
                       Task  Number                      Goal Task: 7

     1      2      3      4      5      6      7         Machine Number: 2
                                                          Run Time:    23.79
   55.00  70.00  69.39  50.00  68.00  68.20  55.00        Cost:       47.21
   52.50  71.00  69.39  49.00  68.50  68.20  68.00
   51.00  72.00  69.39  48.50  68.50  68.20  69.00       Machine Number: 4
M  50.60  72.00  69.39  48.45  68.50  68.20  69.00       Run Time:    23.79
A  50.00  72.00  69.39  48.41  68.50  68.20  69.00        Cost:       47.21
C         72.00  69.39  48.41  68.50  68.20  69.00
H         72.00  69.39  48.41  68.50  68.20  69.00
I                69.39  49.00  -----  68.20  69.00
N                       49.00  70.00  68.20
E                              69.00  68.20
Fa
Re <PgUp/PgDn>  Scroll    Page: 1          <ESC> Exit
   TASK NUMBER        1      6      7
 WINNING BID-PRE    50.60  68.20  69.00
   YOUR BIDS        50.50  Unable 69.00
    RESULTS
Machine  4 { Queue Spaces = 3, Available Run Time = 136.21}
```

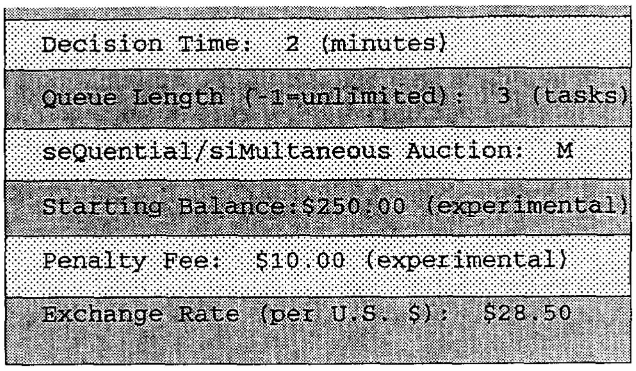
**Figure 8** History summary pop-up screen

the agent has just entered a bid of 69.00 for task 7, and the information relating to task 7 is given on the right-hand side of the screen. Two machines (2 and 4) at the workstation can perform task 7. In this case, both machines 2 and 4 happen to have an identical cost (47.21) and run time (23.79), though this is typically not the case in our experiments. At the bottom of the screen, additional information is provided on the highlighted machine (4), that is, there are currently three spaces available in the queue and available run time is 136.21.

A limited amount of time (chosen by experimenter or auctioneer) is provided for bid formation and submission. The remaining time is presented at the top of the screen (see *Figure 1*), and repeatedly updated. If an agent's bids are not submitted prior to the indicated time (i.e. prior to remaining time reaching '0'), partial bids are automatically captured and submitted. If the agent has entered numerical values for a task bid, these are automatically submitted. If no numerical entries or only a partial entry has been made, a 'no-bid' message is submitted. This feature can be easily modified to 'no-bid' if an agent is not finished completely with bid formulation. Our choice of process reflects preferences expressed by subjects during numerous trial runs.

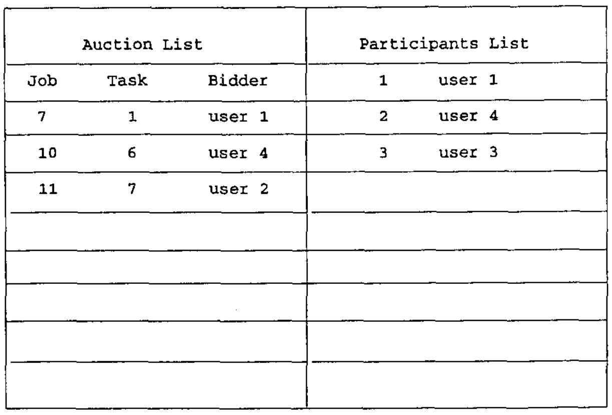
*Figures 9* and *10* show the auctioneer interface screens used in initializing and running the necessary auctions. *Figure 9* is the initialization screen where the auctioneer sets the auction process parameters. *Figure 10* is the record keeping screen for an ongoing auction. When all bids are collected (all submitted, or the bidding time has run to zero), the auctioneer assigns tasks to the winning bidders and maintains assignment and completion records. In our formulation, the auctioneer is fully automated.

In this system, the central manager's role is straightforward, and the operative screens reflect this. The auctioneer records arriving tasks (either tasks decomposed from newly arrived jobs or tasks ready to auction because of the completion of precedent-constrained tasks), broadcasts the tasks over the network, collects bids, and assigns tasks to low bidders. If a task fails to receive an acceptable bid (under a specified ceiling bid or 'outsourcing' level), the system may be set to re-trigger the bidding process for that task.



```
Decision Time:  2 (minutes)
Queue Length (-1=unlimited):  3 (tasks)
seQuential/siMultaneous Auction:  M
Starting Balance:$250.00 (experimental)
Penalty Fee:  $10.00 (experimental)
Exchange Rate (per U.S. $):  $28.50
```

**Figure 9** Server initialization screen



| Auction List | | | Participants List | |
|---|---|---|---|---|
| Job | Task | Bidder | 1 | user 1 |
| 7 | 1 | user 1 | 2 | user 4 |
| 10 | 6 | user 4 | 3 | user 3 |
| 11 | 7 | user 2 | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

**Figure 10** Server record-keeping screen

*Operating environment and communication procedures*

The auctioneer plays an active role while agents initially play passive roles. The auctioneer controls the flow of the process, and agents in the market respond to the auctioneer's announcements. The control sequences are programmed and stored in the auctioneer's station. The auctioneer conducts the system according to this pre-programmed sequence by announcing the predefined signals to all agents in the system. On the other hand, each agent station in the system listens to the auctioneer's signals and takes action in response to the signals received. For each signal, the corresponding response actions for each agent are programmed in separate routines such that the integration of additional response and/or modifications of current actions can be maintained easily.

There are nine different signals used by the auctioneer:

1. Agent enrollment.
2. Current time.
3. Ready to collect tasks for rebidding.
4. Ready to collect completed tasks.
5. List of current auction tasks.
6. Start auction.
7. List of winning bidders for current auction.
8. Repeat last message.
9. End auction.

Each auction period is handled independently using the first eight of the signals listed. The ninth signal is used only at the end of the final auction period to release all agent stations and return to a DOS environment.

When 'enrollment' is signalled, agents in the system register to the auctioneer for preparing the list of bidders. The 'current time' messages sent from the auctioneer to the agents indicates to each agent that an auction is about to begin, and that all completed tasks and tasks for re-bidding will be collected later. When the 'ready to collect completed tasks' signal is sent, each agent reports completed tasks to the auctioneer. An equivalent process holds for tasks that must be re-bid because of a queue constraint violation or a machine breakdown occurrence. When the current list of all tasks (new and re-bid) to be auctioned is compiled, the auctioneer announces the list to all agents. At this point, the announced auction list is not displayed on any agent's screen, but rather is kept in local station memory. When the auction list has been correctly delivered to all agents, the auctioneer sends the 'start auction' signal to all agents. The stored auction list is then displayed on each agency's screen, and the decision making (bidding) process begins. During this decision making stage, the auctioneer continuously monitors the entire system to collect bid decisions submitted by agents. When all agents have submitted their bids or the auction time has expired, the winner (if any) for each task is selected, and agents are sent messages about which of their bids were losing and which were winning bids. At any step, when a signal is mishandled by either the auctioneer or agent(s), the eighth signal listed above is initiated by the receiving site(s) to request the sending site to repeat the message.

Some transmitted messages require that receiving sites take simultaneous actions while other messages require an accuracy check at receiving sites. The broadcast datagram transmission was used when the auctioneer required that all receiving sites act in unison. Acknowledgements from receiving sites were used to guarantee proper delivery of key messages, such as descriptions of completed tasks, tasks for re-bidding and tasks being auctioned. When a guarantee is required by the system, receiving sites feed received messages back to sending sites for direct comparisons. When the acknowledged message is compared and is correct, a flag indicating correct message delivery is then set if the auctioneer is the sending site. If agents are the sending sites, a 'return to listen mode' is triggered. If an acknowledgement appears to be incorrect, the original message is resent and the acknowledgement process repeats.

As the number of nodes increases, this accuracy check process can involve significant communications among sending and receiving sites, and can occupy a large portion of the limited memory available on the IBM LAN adapters. In the token ring configuration, messages are transmitted from sending sites to receiving sites through all nodes in-between. If we used the broadcast datagram transmission process, messages would be accepted and handled by each and every node in this route to lessen the network load. When messages require guaranteed delivery, we use the plain datagram transmission mode where messages are handled only by specified recipients and ignored by non-specified recipients. This process trades off the ease of the broadcast process for a reduction in network congestion. To facilitate this process, each station (including the auctioneer and all agents) registers a common group name for all broadcast datagram communications and a unique local name for plain datagram communication.

As outlined earlier, the agent at each local node is provided with a DSS to deal with the local scheduling problem. Following the description provided by Bonczek *et al.*[5], a DSS consists of four elements:

1. Language system.
2. Presentation system.
3. Knowledge system.
4. Problem processing system.

Decision makers issue problem processing requests through the language system, which in turn feeds problems to the problem processing system for suggestions. The problem processing system, with the aid of the information available in the knowledge system, presents suggestions or alternatives via the presentation system. Thus, our system provides a set of distributed DSSs, where the auctioneer (central scheduler, if you will) makes requests to schedule a set of jobs in a specified format through the auctioneer station. These requests are distributed over the network and processed by agents at individual nodes. The auctioneer decomposes the jobs into tasks and distributes these tasks to all agents in the system. Knowledge relating to scheduling the requests (jobs) is maintained and stored locally at each agent node. The auctioneer maintains knowledge about available agents in the system and about payments to agents. Scheduling problems are processed in the form of auctions, with agents able to access local DSSs for bidding and scheduling decisions, and the auctioneer maintaining information on agent performance and jobs completed, in process, or ready to be offered for auction. *Figures 1* to *8* detail the local DSS language and interfaces, while *Figure 9* illustrates the auctioneer's language system and *Figure 10* the corresponding interface.

### Knowledge system

The system we developed creates a three-dimensional link list. The machines provided at each station occupy one dimension, the production capabilities of each machine are maintained in another dimension, and the tasks assigned to each machine are in the third dimension. Since the production-related data maintained in our system requires quick and easy access, and as the memory size required for this data is fixed, we maintain the information internally. Historical information is maintained externally.

### Problem processing system

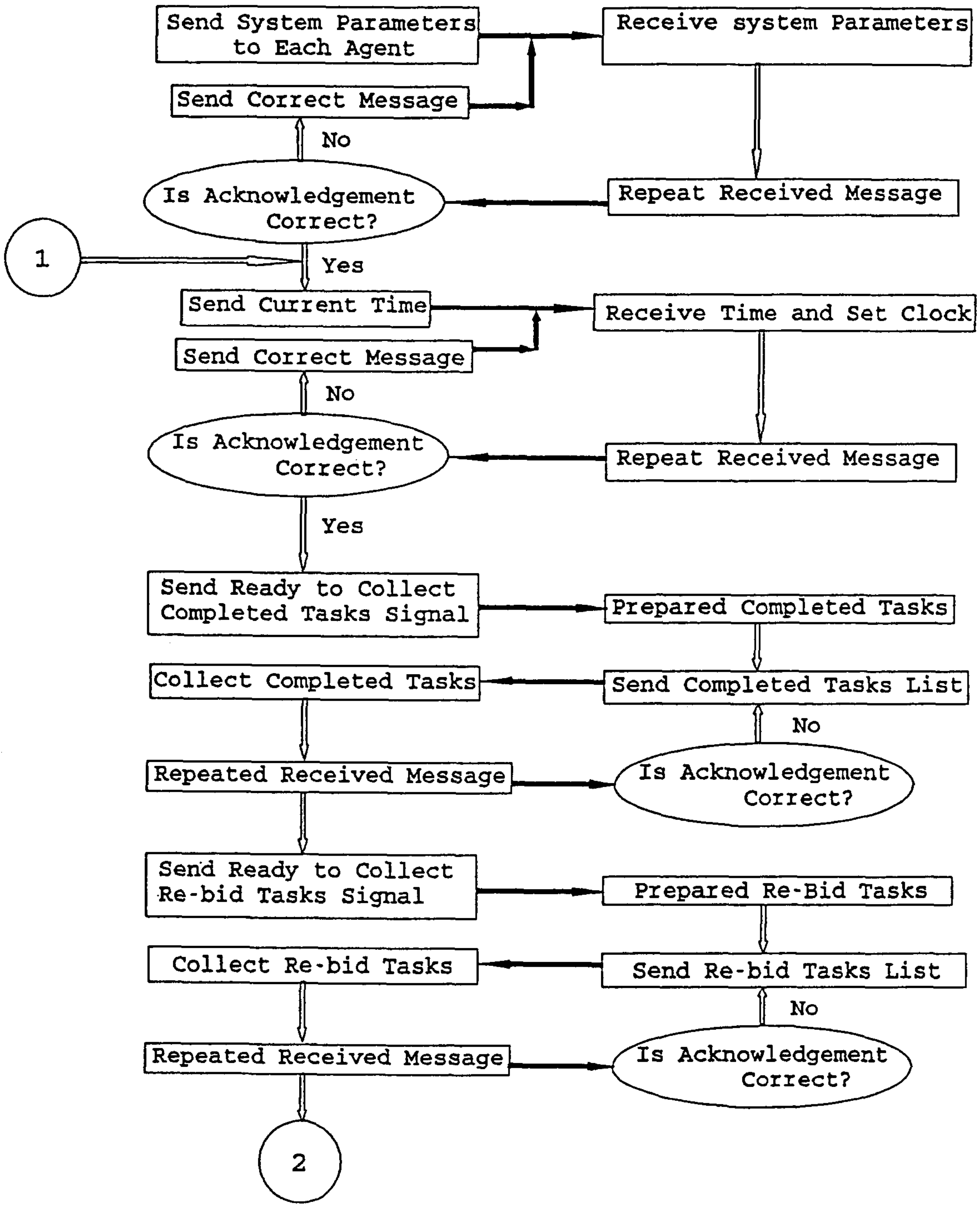
Operation follows a pre-specified sequence of actions, outlined in the flow control charts presented in *Figures 11* and *12*. Following the necessary registration confirmation and dissemination of parameter values, the auctioneer initiates the auction using broadcast datagram transmission, and the local DSSs are triggered to present the auction information to each local agent. During the collection of bids stage, the auctioneer opens a communication channel for each agent and continuously checks each channel to collect agents' bids in a submitting list ordered in sequences of arrival. When all bids are collected (submitted or collected because the decision time has expired), the auctioneer selects the lowest bidder for each item as the winner. A first-in-wins rule is used as a tie breaker. Ceiling prices or 'outsourcing costs' are used to avoid the possibility of extremely large profits for any given task. If no bids on a task are submitted, the task is set for re-auctioning in the next auction period.

When the steps are completed, each local station is informed of all the winning bid amounts, and each of their bids is identified as a winning or losing bid. Winners are offered a specified time in which to alter their tentative machine assignment decisions in the light of auction outcomes. When this is completed, the system begins the second period in the auction cycle. When the last auction in the specified period has been completed, the auctioneer broadcasts a simple 'end-of-auction' announcement to shut down shell operations.

Having discussed the flexibility of the environment constructed, we move to illustrate its operation and provide results obtained in our initial experimentation using the shell.

## Experimental results

The experiment reported here involved auction cycles of 25 auction periods with four participants (local agent nodes). Experiments used the induced value approach of Vernon Smith[6–8], where subjects receive monetary



Send System Parameters to Each Agent
Receive system Parameters
Send Correct Message
No
Is Acknowledgement Correct?
Repeat Received Message
1
Yes
Send Current Time
Receive Time and Set Clock
Send Correct Message
No
Is Acknowledgement Correct?
Repeat Received Message
Yes
Send Ready to Collect Completed Tasks Signal
Prepared Completed Tasks
Collect Completed Tasks
Send Completed Tasks List
No
Repeated Received Message
Is Acknowledgement Correct?
Send Ready to Collect Re-bid Tasks Signal
Prepared Re-Bid Tasks
Collect Re-bid Tasks
Send Re-bid Tasks List
No
Repeated Received Message
Is Acknowledgement Correct?
2

**Figure 11** Process control flow – chart 1

rewards based on their performance. As Smith explains in detail, it is important that potential monetary rewards are sufficient to influence subjects to seek to perform at their best. In the experiment discussed here, subjects received a $5 'show-up' fee and a participation fee based on performance that had an expected value that averaged $12.50 per subject for approximately a 50 minute session.

Our system was set up for these experiments to provide two FMS environments. One environment provides each node operator with four machines, with each machine capable of performing two different tasks (referred to as the **4M2T** environment). The second experimental environment provides each node operator with two machines, with each machine capable of performing four different tasks (referred to as the **2M4T** environment). Some of the tasks in each environment were competitive (i.e. they could be completed at more than one node), and some were idiosyncratic (i.e. they could only be done at one single node). For example, as shown in *Figure 13*, in the **2M4T** environment, task 1 could be done at every node, while task 7 could only be performed at node I. In the **4M2T** environment, task 4 could be performed at every node, while task 9 could only be performed at node IV (with two machine options, A and C, at that node).

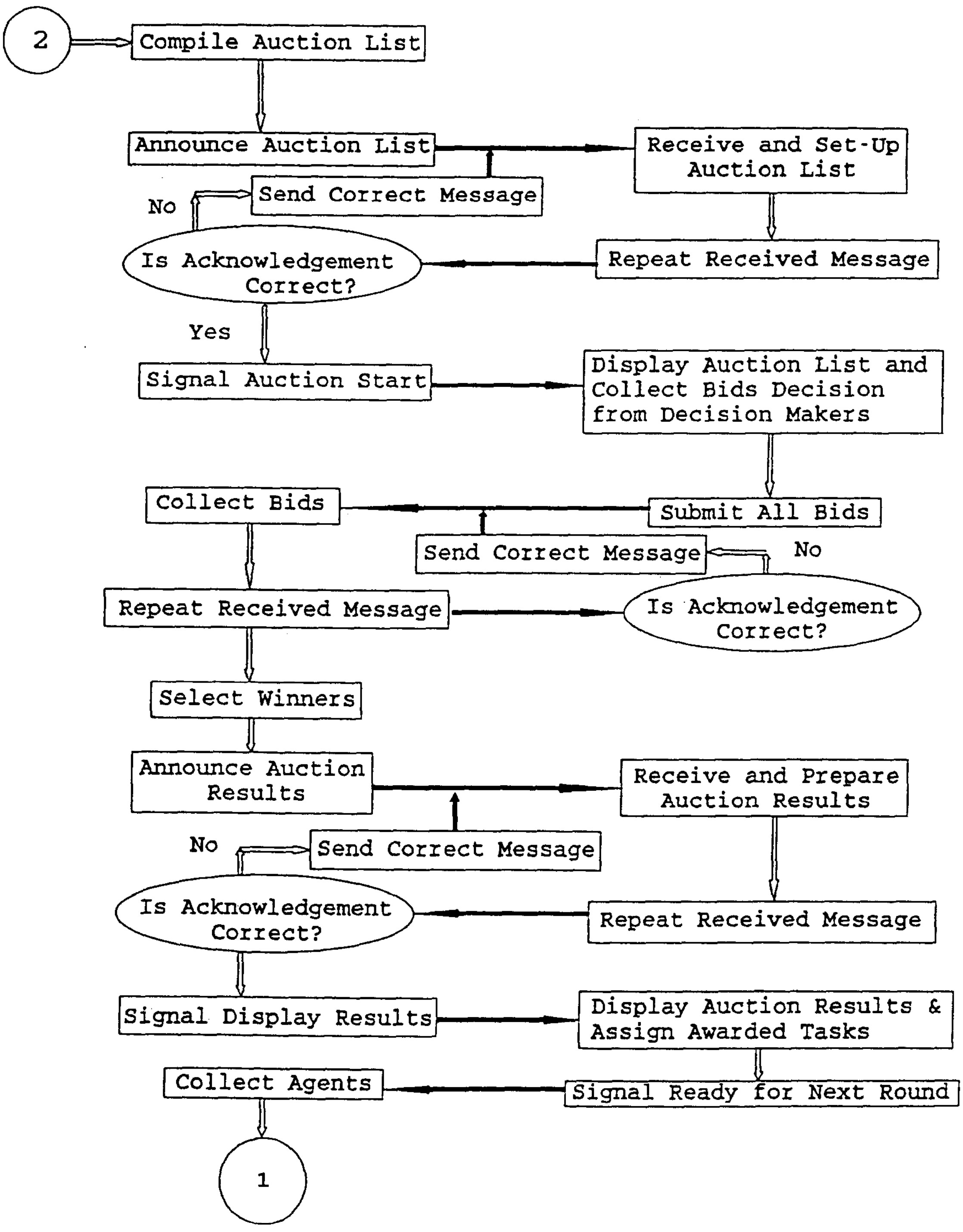


**Figure 12** Process control flow – chart 2

System Setting for 2M4T Cycle

| | Machine | Capability | | Machine | Capability |
|---|---|---|---|---|---|
| User 1 | 1 | 1, 2, 4, 8 | User 3 | 1 | 1, 4, 5, 8 |
| | 2 | 1, 4, 7, 8 | | 2 | 1, 4, 8, 10 |
| | Machine | Capability | | Machine | Capability |
| User 2 | 1 | 1, 3, 4, 8 | User 4 | 1 | 1, 4, 6, 8 |
| | 2 | 1, 4, 8, 9 | | 2 | 1, 4, 8, 11 |

System Setting for 4M2T Cycle

| | Machine | Capability | | Machine | Capability |
|---|---|---|---|---|---|
| User 1 | 1 | 1, 6 | User 3 | 1 | 1, 2 |
| | 2 | 3, 4 | | 2 | 4, 11 |
| | 3 | 6, 8 | | 3 | 2, 8 |
| | 4 | 1, 3 | | 4 | 8, 11 |
| | Machine | Capability | | Machine | Capability |
| User 2 | 1 | 1, 5 | User 4 | 1 | 1, 9 |
| | 2 | 4, 7 | | 2 | 4, 10 |
| | 3 | 5, 8 | | 3 | 8, 9 |
| | 4 | 4, 7 | | 4 | 1, 10 |

**Figure 13** Experimental environment

**Table 1 Even interval net profit percentage gains**

| | Interval 1 | Interval 2 | Interval 3 |
|---|---|---|---|
| Idiosyncratic tasks (2M4T) | $\bar{X} = 81.1494$ | $\bar{X} = 91.3277$ | $\bar{X} = 91.1607$ |
| | $S^2 = 31.5487$ | $S^2 = 0.7480$ | $S^2 = 0.7607$ |
| | $N = 18$ | $N = 30$ | $N = 28$ |
| Idiosyncratic tasks (4M2T) | $\bar{X} = 70.0455$ | $\bar{X} = 85.4204$ | $\bar{X} = 86.8243$ |
| | $S^2 = 7.1436$ | $S^2 = 0.71196$ | $S^2 = 0.2354$ |
| | $N = 20$ | $N = 26$ | $N = 30$ |
| Competitive tasks (2M4T) | $\bar{X} = 1.5171$ | $\bar{X} = 0.4860$ | $\bar{X} = 1.9053$ |
| | $S^2 = 0.004194$ | $S^2 = 0.001132$ | $S^2 = 0.1625$ |
| | $N = 7$ | $N = 5$ | $N = 15$ |
| Competitive tasks (4M2T) | $\bar{X} = 8.8286$ | $\bar{X} = 0.7850$ | $\bar{X} = 1.8394$ |
| | $S^2 = 0.5683$ | $S^2 = 0.019016$ | $S^2 = 0.02427$ |
| | $N = 7$ | $N = 4$ | $N = 17$ |

$\bar{X}$ = average net profit percentage gains
$S^2$ = variance of net profit percentage gains
$N$ = number of competitive tasks announced

Subjects for this example experiment were four doctoral students in the DSIS doctoral program at the University of Kentucky. None were involved in this project. Two experimental cycles were conducted, one using the **2M4T** environment and one using the **2M4T** environment. We use average net percentage of possible profits as our measure of performance. For each auctioned task, possible profit percentage ranges from 0 (for a winning bid identical to the cost of performing the task) to 100 (for a winning bid identical to the outsourcing price or upper bound). For winning bids between these levels, the percentage of possible profit equals the following:

$$\frac{\textit{Winning bid} - \textit{Production Cost for Winning Bidder}}{\textit{Outsourcing Price} - \textit{Production Cost for Winning Bidder}}$$

*Table 1* provides the average net profit percentages obtained in the auction in each of our two experiments. Tasks are grouped into idiosyncratic and competitive tasks. Results are reported for three intervals corresponding to the first eight, ninth through sixteenth, and seventeenth through 24th observations. All participants appeared to quickly identify both their idiosyncratic and competitive tasks, though they begin the auction with only local node knowledge not knowing whether any of the others can do any or all of the tasks that their local node can perform. For idiosyncratic tasks, average net profit percentages in even the first period set were 81% for the **2M4T** case and 70% for the **4M2T** set. For competitive tasks, net profit percentages appeared to be quickly driven downward, averaging 1.5% in the first eight periods for the **2M4T** FMS environment and 8.9% for the **4M2T** environment. For idiosyncratic tasks, average net profit percentages for subsequent periods were 91.3% and 91.2% for **2M4T** and 85.4% and 86.8% for **4M2T**, respectively. For competitive tasks, average net profit remained at under 2% for all three experimental intervals for both sets of experiments.

These initial results suggest that the mechanism was performing as expected in the FMS environments studied. Based upon these results, we are now beginning an in-depth analysis of the performance of the system as we alter the complexity of the local node scheduling problems, permissible queue size, penalty amounts, performance incentive mechanisms, subject characteristics, and overall size and complexity of the FMS environment.

Post-experiment interviews were conducted with subjects as a means of identifying process problems that could be addressed prior to subsequent experimentation. For example, in the experiments reported here, subjects suggested that, for competitive tasks, competition was initially the driving market force. In subsequent periods, however, some participants indicated that resource constraints due to queue size limitations began to play an important role in the bidding decision processes of some of the participants, suggesting the need to analyse alterations in this parameter value in subsequent experimentation.

## Summary and conclusions

We outlined the development and potential use of a flexible operationalization based on the approach of Shaw and Whinston[2]. Operational details and possible uses for this system were illustrated through two example experiments with human subjects as decision makers in FMS environments. The results reported were consistent with expectations for both competitive and idiosyncratic tasks.

The development of the system is significant because of its many possible uses, including comparison, for various FMS environments, of dispersed decision making performance (e.g. system cost, production cost and throughput time) with the performance of existing centralized scheduling heuristic in those environments. We can study how alterations in the complexity of local node environments, and/or changes in incentive, impact the performance of decentralized scheduling. We can investigate whether expert systems can be developed and, if so, how well the system operates with automated bidding systems rather than human bidders.

## Acknowledgements

The authors are indebted to Dr Ram Pakath, Dr David Pingry and Kimlynn Marsden for many helpful comments and suggestions. This research has been supported in part by NSF Grant No. IRI-8921603: Dr A B Whinston and Dr C W Holsapple (Co-Principal Investigators) and in parts by the MIS Research Lab Endowment Fund, Department of Decision Science and Information Systems, University of Kentucky.

## References

1 **Kusiak, A** 'Application of operational research models and techniques in flexible manufacturing systems', *Euro. J. Operat. Res.*, Vol 24 No 3 (March 1986) pp 336–345

2 **Shaw, M J and Whinston, A B** 'A distributed knowledge-based approach to flexible automation: the Contract Net framework', *Int. J. Flexible Manuf. Syst.*, Vol 1 No 1 (1988) pp 85–104

3 **Davis, R and Smith, R G** 'Negotiations as a metaphor for distributed problem solving', *Artif. Intell.*, Vol 20 No 1 (1983) pp 63–109

4 **Smith, R G** 'The Contract Net protocol: high-level communication and control in a distributed problem solver', *IEEE Trans. Comput.*, Vol 29 No 12 (1980) pp 1104–1113

5 **Bonczek, R H, Holsapple, C W and Whinston, A B** *Foundations of Decision Support Systems*, Academic Press, New York (1981)

6 **Smith, V L** 'Experimental economics: induced value theory', *Am. Economic Rev.*, Vol 66 No 2 (1976) pp 274–279

7 **Smith, V L** 'Bidding and auctioning institutions: experimental results', in *Bidding and Auctioning for Procurement and Allocation*, Y Amihud (ed), New York University Press (1976) pp 43–64

8 **Gardner, C L, Marsden, J R and Pingry, D E** 'The design and use of laboratory experiments for DSS evaluation', *Decision Support Syst.*, Vol 9 (1993) pp 369–379