Flexible Workflow Driven Job Shop Manufacturing Execution and Automation Based on Multi Agent System

He Yanli[1], Yang Haicheng[2], He Weiping[1], Zhang Wei[1], Hu Xinping [3]

[1] The Key Laboratory of Contemporary Design and Integrated Manufacturing Technology, Ministry of Education, Northwestern Polytechnical University, Xi'an, China

{heyl,weiping,zhangw}@nwpu.edu.cn

2 China Aerospace Science and Technology Corporation, Beijing, China

{yanghaicheng}@vip.sina.com

3 Capital Aerospace Machinery Co, Beijing, China

Abstract

Manufacturing automation with adoptability to the changing environment is a competitive advantage for shop floors. This paper presents a flexible workflow enacted job shop manufacturing execution and automation system. The manufacturing process sequence of each part is modeled as an abstract workflow and each manufacturing job resulting in a batch of the part items is recognized as corresponding workflow instance. The production schedule is generated to establish executable workflow instances to be enacted by the workflow engine and executed by the resources in the shop floor. Agent technology is applied to construct the system to take the advantage of distributed system architecture, reactivity, adaptability and cooperation. The proposed system can help to achieve production efficiency without huge investment on industrial automation system. The system architecture and implementation methods are discussed with demonstration and the performance of manufacturing shop floor is enhanced after the application.

1. Introduction

Job shop manufacturing involves a set of jobs and a set of machines. Each machine can handle at most one job at a time, and each job consists of a chain of operations, each of which must be processed during an uninterrupted time period of given length on a given machine. For many plants, the manufacturing process is still scheduled and managed by manual implementation. Statistics shows that only 10% of the machine time is used for material processing, and 90% of time is dead time which is spent for waiting and which does not add value to the business[1]. Manufacturing process automation can enable the shop floor to operate more effectively by reducing the dead time dramatically.

For this purpose, Manufacturing Execution System (MES) should embed workflow to support process automation besides detail scheduling, work instrument delivery, online tracking, etc. However, the automation should also be flexible enough to respond agilely to the changes in the shop floor environment. Recently, agent has seen increasing application in flexible MES and workflow systems [2-4]. This paper applies agent technology to design and implement a flexible workflow enacted MES to support job shop manufacturing process automation.

2. The system model and framework

Flexible workflow is introduced for process automation in job shop because, when treating the manufacturing process sequence as a workflow, the resources responsible for executing tasks in workflow instances are not specified in process model in advance, but are determined through production scheduling before (and often modified) or during the job execution. Also, the executers of the same task in different workflow instances are likely to be different. For this purpose, abstract workflow [5] is used for modeling the manufacturing process of each part. Here, an abstract workflow is the workflow described by abstract resource or abstract service rather than the fixed resource instances. The abstract resource (service) is the description of resources (service) abstracted to express a class of resources (service) providing the same function. And each work sequence

Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06)
0-7695-2748-5/06 $20.00 © 2006 IEEE

resulting in a batch of identical discrete part items (a job) is recognized as a workflow instance.

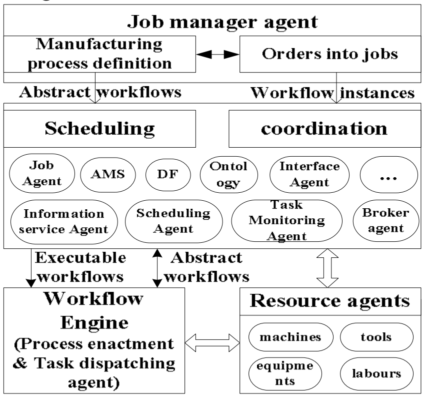

Figure 1. The system framework

The workflow driven MES is constructed on the basis of multi agent system in which agents are used to encapsulate the resources in the shop floor and to implement the functional components of MES for distributed scheduling, automated task dispatching and notification, execution and coordination .In other words, the manufacturing execution can be considered as multiple concurrent workflow processes that is planned, performed, communicated, and coordinated in a multi-agent environment.

The system framework is illustrated in figure 1. The shop floor received the manufacturing order consisting of a set of part orders from EPR/MRP system, each of which describes the part item to be manufactured, the lot size, due date, among others. Each part is associated with a manufacturing process sequence. The manufacturing order is then submitted to the job manager agent (JMA) in which the manufacturing process of each part order is modeled as an abstract workflow by describing the tasks and the precedence relations between tasks using process definition tools. Also in the JMA, each part order is split into one or many independent work batches or jobs (this is often the case if the lot size of part order is too large or the material volumes available is not fully enough) and each job is recognized as the workflow instance (of corresponding abstract workflow model).

The abstract workflow instances should be translated into executable ones by replacing the abstract resource and service with resource and service instances through resource selection and bonding. The physical resource like machines, tools and equipments in shop floor are encapsulated as resource agents (RAs) which possess necessary knowledge and capability to perform the tasks assigned. The translation from abstract workflow instances into executable workflow instances is made through production scheduling and coordination layer and the job schedule indicates when and where (which RAs) the manufacturing tasks in the workflows should be executed.

For agents to be legally identified in the system, each agent should register itself to Agent manager server (AMS). Directory facilitator (DF) is a yellow page server which allows RAs to advertise and offer their services to other agents. Ontology server is implemented to provide the common understanding among agents to ensure correct conversation. Each job is represented by a job agent (JA) to maintain its local schedule. Interface agent (IA) is used to provide GUI interface for human being to operate the system. Information service agent (ISA) is used to store and manage the resources and other information of RAs. Scheduling agent (SA), broker agent (BA) and task monitoring agent (TMA) are used to support production scheduling, resource selection and state monitoring respectively.

The workflow engine mainly consists of process enactment agent (PEA) and task dispatching agent (TDA) which enacts and automates the executable workflow instances and dynamically invocates corresponding resource agents for task execution. If exceptions like machine failure or unavailability of material occur which make it impossible for the executable workflow to proceed, a re-scheduling process is initiated to reactive to the changes in which the production reservation is modified before execution continues. In some circumstances, the abstract workflow instances can also be enacted by workflow engine for real-time scheduling in which the decision of resource allocation is made during the job execution according to the real-time situation of the RAs (we do not discuss this circumstance in this paper).

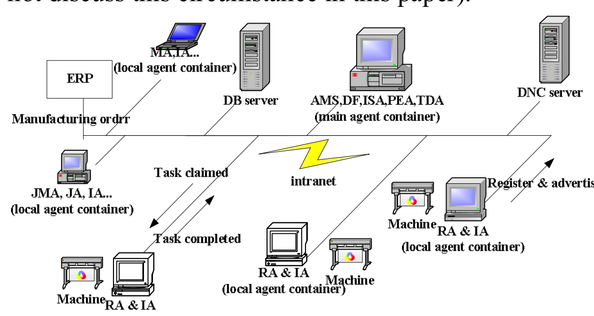

Figure 2 .The system deployments

Figure 2 illustrates the deployment of the system and shows how the cycle of production is being run. The system has a server which hosts the main agent container in which AMS, DF, ISA, PEA, TDA runs and client ends which host local agent containers in which RA, JMA, JA ,SA etc and their corresponding IA runs. Each machine in the shop floor is equipped with a touch-screen computer connected to the intranet. The RA runs on the computer and acts on behalf of the corresponding machine. When initialized, the RA will register itself using the machine's identity as the agent name to the AMS, advertise the capability (the functionality the machine can perform) to the DF and records other information like the name, type, etc in the ISA. The detail production schedule is made weekly and verified each day by the manager. The tasks in the schedule are distributed among RAs so that each RA will receive the tasks assigned when joining the system.

When a job stars, the JMA starts the first task in the job, and inform the responsible RA that the task is ready for execution. Then the operator of the RA will claim the task and perform the task. When the task is finished, the operator will change the state of the task through the interface, and RA then will inform the PEA to trigger the next task. This process continues till the last task in the job is completed.

3. The implementation methods

3.1. Ontology Construction

Besides the common agent communication language (FIPA ACL) and interaction protocols, agent needs the same understanding about the domain for conversation. The domain classes, attributes, relations and axioms are defined in the ontology, serving as standards for agent communication by providing shared vocabulary and semantics of the language expressing the content of messages. Main domain concepts include *Job*, *Process, Task*, *Resource* etc, as shown in figure 3.

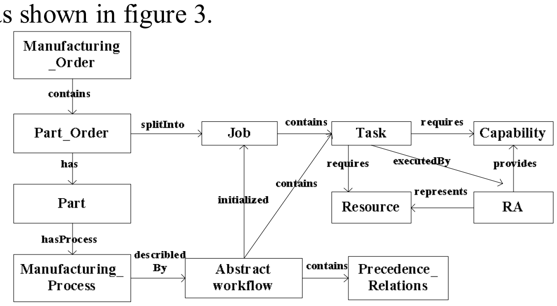

Figure 3. Some concepts in the ontology

The ontology is built by protégé and translated into java beans. The information is encoded as the object of ontology class and further encoded into an ACL message as content object to be sent to other agents. The conversation among agents are designed and programmed following the FIPA interaction protocol.

3.2. Abstract workflow for process definition

Manufacturing process of each part is defined as abstract workflow using activity based techniques which focus on modeling the work involved in a process and their dependences. The abstract workflow process is described by a set of tasks representing manufacturing activities and ECA rules representing the control flow. The control logic like *sequence*, *join*, *fork*, *choice*, *merge* are all expressed in ECA rules (see figure 4). In the abstract workflow process, the resource or capability requirement of the task (e.g. a NC machine of type SAJO 1200) rather than task implement or executor is specified. The task requirements are used to find corresponding RAs that can implement the task.

R1 Event : On END (Task 1)
Condition：With (Quality_OK)
Action : DO START (Task 3)

Figure 4. The sequence control in ECA rules

The abstract workflow process is modeled by JMA through the GUI and saved in database for process scheduling and enactment. Figure 5 shows the abstract workflow process description of a part. Changes in the process will only cause changes in tasks and ECA rules, as provides a more powerful way to support ad-hoc, adaptive, flexible and dynamic business process modifiable in run time.

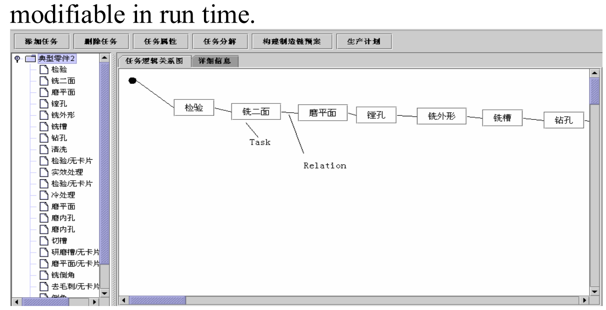

Figure 5 .The abstract workflow process

3.3. Job scheduling

A job represents the workflow to manufacture a batch of a particular part items. If the part order is spitted into many jobs, these jobs will share the same manufacturing process and so are regarded as different workflow cases of the same process model with an unique ID to distinguish one case from another(see figure 6). In the workflow cases, each task will be executed N times before the transition takes place where N represents the batch size of the job.

The production scheduling manager will make the production schedule for all tasks in all the jobs through the SA in which the start time and finish time are identified and a RA is selected as the executor for each task (shown in figure 7). The scheduling can be made with the support of rule based approach or search strategy like genetic algorithms for optimization. For manual scheduling, when a task is selected, the BA will search the DF and ISA, matching the resource and capability implement requirement against the available resources and capability specification advertised by the RAs, and lists the poetical RAs that can satisfy the task requirement, from which an appropriate one will be selected . The match-matching algorism in BA is based on the ontology to ensure the semantic equivalence.

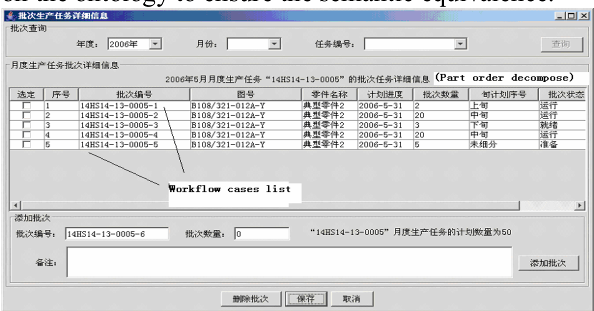

Figure 6. Split part order into jobs

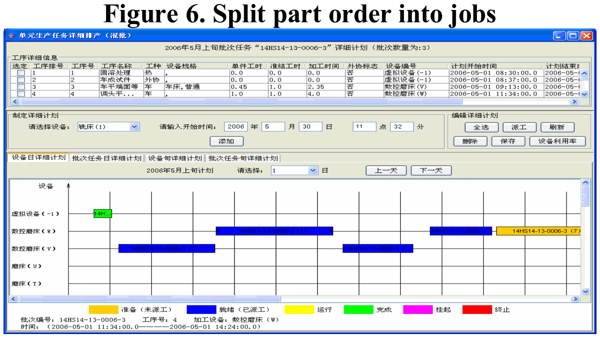

Figure 7. The scheduling agent interface

With the executors for each task of each job are indicated in the production schedule, the executable workflow instances are established and saved in the database and further sent to the workflow engine to be enacted and executed by the RAs over time.

3.4. Agents for workflow process execution

The manufacturing processes of jobs are automated by workflow engine by PEA according to the process logic of the workflow. The PEA is implemented with event driven rule mechanism on the basis of the ECA rules. The interactions among agents are shown in figure 8.

(1)All tasks in the jobs are set to be in “INACTIVE” state by default. When joining the system, the RA will receive the tasks in different jobs assigned to him according the schedule and lists the task items by days in the work-list GUI in different color, e.g. in red if the task is not ready to be executed, green if the task is ready (the preceding task is completed), and gray if the task is completed (see figure 9).

(2)When the manager start a job, the first task in the job was initiated its sate is set to be “READY”.

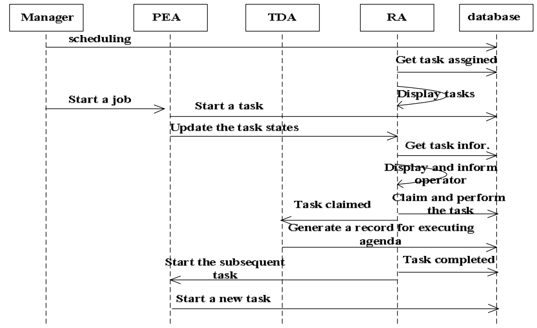

Figure 8. The interaction for process execution

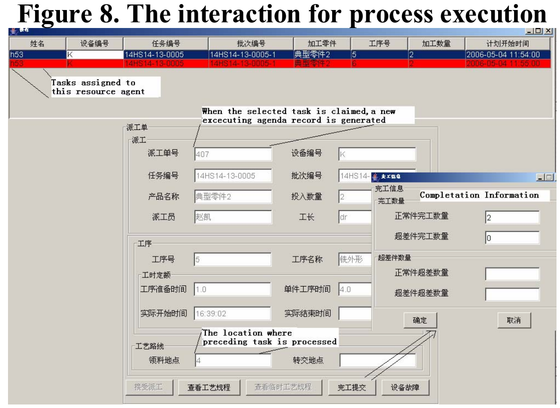

Figure 9 . The operation interface of RA

(3)For each task initiated as “READY”, the PEA will send an inform message to the corresponding RA responsible for the task to update the task state in the work-list. On receiving the message, the RA will redraw the task-list interface in which the corresponding task item is displayed in green color, informing the operator that the task may be executed.

(4)When the operator claims the task, the state of the task is changed into “RUNNING” and a new record is generated in executing agenda by the TDA. The operator then gets the WIP from the machine where the previous task is processed following the instruction in the GUI, and works on it. When the task is completed, the operator will change the task state to “COMPLETED” through the RA interface by submitting the result information (Figure 9). RA then informs the PEA to start the subsequent task.

(5)On receiving the massage indicating the completion of a task, the PEA will search the ECA rules to match the event and test the transition conditions required by the ECA rules, then trigger the subsequent task by setting its state to “READY”.

(6)Continue step 3 to step 5 till the whole job is completed. And the actual executing agenda of the job is generated gradually as a consequence.

3.5. Process inspection and coordination

The manufacturing processes of the jobs are displayed in graphical interface of the TMA for managers to inspect. When required, the TMA will show the progress of each job, and further the state, the actual start time, completion time of each task according to the information gathered during the execution, as shown in figure 10.

Figure 10. TMA interface showing the job progress

Things rarely go as expected in job shop manufacturing environment. The coordination is implemented in the system to responds to various kinds of exception. If machine fails, the responsible RA will initiate a Contract Net process to subcontract the tasks to other RAs. However, if the problem can not be settled through this way, the RA will send a inform message to the manager for appropriate actions to be taken, e.g. to re-schedule all the tasks which is still in "INACTIVE" state. The modified schedule will replace the previous one to make sure that the right RA will receive the right work list at right time.

4. System implementation and application

A flexible workflow driven MES system is constructed on the basis of JADE agent framework [6]. The system agents are programmed in java class extending agent class. The workflow participants ,either worker or shop floor manager, will run the executable workflow client, connect and get registered as an agent to the jade main container server to form a multi agent distributed environment. The production scheduling manager can model the abstract manufacturing workflow, make schedule and manage all the tasks through the JMA interface. The RA can advertise resources and services to the system. For each task in the job, the BA can provide the potential resource agents for manager to support the production scheduling. Tasks in the jobs are enacted by PEA, and the state of each task during execution is displayed in the TMA. The system is tested and applied in a shop floor of Capital Aerospace Machinery Co. and the manufacturing efficiency and capability was increased so that millions of dollars previously spent for subcontracting was saved.

5. Conclusion

This work provides an agent based workflow enacted MES system for job shop manufacturing execution and automation. The proposed solution supports manufacturing process automation without requirements of automatic equipments in the shop floor. However, much effort has to be done to improve the system performance (e.g. the scheduling and re-scheduling methods) and to analysis the tangible and intangible profit in the application. The solution and system is expected to see more applications in discrete manufacturing industry in the near future.

6. Acknowledgement

The work is supported by Aeronautic Science Program (Grant No.04H53063), the Commission of Science Technology and Industry for National Defense R&D Program (Grant No.B2720060292) and NWPU international cooperation foundation program. The authors would thank Dong R, Zhao K, Hao GK, Chen JL, Zhang GF, Liu HY and Wu Zhen for their contribution in the work.

7. References

[1] Business processes in Internet–time. Ultimus technical report,http:// http://www.ultimus.com/

[2]Cheng F.T, Chang C.F and Wu S.L. "Development of holonic manufacturing execution systems",Journal of intelligent manufacturing, 15,2004,pp.253-267.

[3]Arndt Luder, Jorn Peschke, Thilo Sauter, etal. "Distributed intelligence for plant automation based on multi-agent systems: the PABADIS approach",Production planning & Control, 2005, 15(2),pp. 201-212.

[4]Paul A.Buhler, Jose M.Vidal. "Towards adaptive workflow enactment using multi agent systems", Information technology and management, 6(2005),pp.61-87.

[5] Wang WJ, Liu Xp, Luo YW, et al. "Mapping Business Workflows onto Network Services Environments", Lecture Notes in Computer Science, 3251(2004) ,pp.97 – 104.

[6]http://jade.cselt.it/.