A Practical Scheduling Method based on Workflow Management Technology

HUIPING LIN [A], YUSHUN FAN [B], ELEANOR T. LOIACONO [A]

[a] *Department of Management,Worcester Polytechnic Institute, Worcester, MA 01609 USA*

[b] *Department of Automation, Tsinghua University, Beijing, 100084 P.R.China*

In literature, most scheduling models are developed using mathematics method such as programming model or graphic model such as Petri net. Although they are efficient in computing, they are difficult to be built and understood by the industry users. In order to make the scheduling model more capable and practical for industry use, a new scheduling model based on workflow management technique (SMWMT) is proposed here. It is a process oriented compound model that featured as various process constraints definition, activity behavior description, flexible resource definition and mapping, and graphic presentation that offer intuitional understandings to users. The corresponding scheduling system based on workflow simulation is provided. A cost oriented iterative algorithm is proposed to enhance the scheduling performance. The example of the model and the simulation results show that the proposed model and scheduling method are effective.

Key word: Scheduling Model, Workflow Management, Workflow Simulation, Iterative Algorithm

1 Introduction

Job shop scheduling, which is usually characterized as sequencing *n* jobs on *m* machines to optimize production performance, has received a considerable attention from researchers. Most endeavors in literature were devoted to developing scheduling algorithm such as tubo search[1], genetic algorithm[2], neural networks[3] and so on to improve the scheduling efficiency. However, modeling issue is still one of the obstacles that hamper various methods from being widely used in real industry. Currently, there are two major scheduling models. One is developed by using mathematical method such as line programming model. Although it is convenient for computing, they do not offer an intuitional understanding to users. Also it has limitation in describing complex process constraints. The other one is graphic model such as Petri net[4,5,6], which has the potential to describe complicated process constraints like concurrence and synchronize. The major problem of it is that the model scale will grows dramatically with the system complexity. It still lacks of a comprehensive

Corresponding to: Dr. Huiping Lin, Department of MIME, Northeastern University, Boston, MA, 02115 USA, E-mail: linhp@coe.neu.edu

scheduling model that possess both of following two features.

- Capable of describing complicated relations among jobs, machines, and routings.
- Could being easily established and understood by industry users to make it more practical in industrial application.

In order to meet above two requirements as “capable” and “practical”, a new scheduling model based on workflow modeling technique, the corresponding scheduling system and the enhanced algorithm are proposed. The paper is organized as follows. First, a brief introduction to workflow management technology is given to answer the question why it is chosen as foundation of our method. Then, the structure and semantics of the new scheduling model are discussed in detail. The related scheduling method is provided followed. Finally, in order to improve the scheduling efficiency, a cost oriented iterative scheduling algorithm is discussed. Examples and experimental results show that the proposed scheduling model and method is effective.

2 Introductions to Workflow Management Technology

Workflow management is one of the hot research areas that attracted great attention from researchers, developer and users since 1990’s. Over years, various definition of workflow are given. For example, Workflow Management Coalition defines workflow as “the computerized facilitation or automation of a business process, in whole or part”[7]. Giga Groups call “the operation aspects of a business process, the sequence of tasks and who perform them, the information flow to support the tasks, and the tracking and reporting mechanism that measure and control them” workflow[8]. And Fan defines workflow as “computerized process model which can be operated by workflow management system in order to realize business process integration and automation” [9].

Although above informal definitions vary more or less, we still could find two identical similarities. First, a workflow model is ready to describe three aspects of

Corresponding to: Dr. Huiping Lin, Department of MIME, Northeastern University,
Boston, MA, 02115 USA, E-mail: linhp@coe.neu.edu

the business processes as:

- what is a process (definition of the activities that buildup the process);
- how is the process (definition of the logic relationships between activities);
- who will perform the process (definition of the resources);

Second, workflow model is an executable model that can be read, operated, and controlled by certain workflow management system. Thus, we can tell that workflow model is a comprehensive process model that has the potential to be extended to describe manufacturing process and schedule the activities. The typical workflow model is usually established by activity-based method with a graphic presentation, which offers a direct view to users.

3 Scheduling Model based on Workflow Modeling Technique (SMWMT)

3.1 Model structure

Due to the complexity of real manufacturing system, it is very difficult to contain all the information necessary for scheduling in a single model. Hence, three views are introduced to compose the scheduling model SMWMT as the Process View, the Resource View and the Job View. Different views are designed to describe different aspects of the process. The views and the relationship among them present various constraints and conditions of the scheduling problem (Figure 1).

(1) Process View

The process view is built using activity-based method. It is made up of multi processes, each of which defines activities and their logic relationships (process constraints) needed for one type of job. Thus, each type of job might have its pre-defined manufacturing routing in SMWMT. A directed acylic AOV diagram, where nodes indicate activities and arcs indicate dependencies, is used to present each process (job). In order to describe five typical process logic such as Serial, And-Joint, And-Split, Or-Joint and Or-Split logic, and indicate the start/end point of a process instance, six logic nodes other than activity node are introduced into the model. They are:

- Start node: Notation of the beginning of one process instance;
- End node: Notation of the termination of one process instance;
- And-joint node: Being triggered only if all of its in-going nodes are executed;

- And-split node: Triggering all of the subsequent activities;
- Or-joint node: Being triggered if *any* of the in-going nodes are executed;
- Or-split node: Triggering only one of subsequent activities.

For each activity node, we have three kinds of description: *property definition, resource mapping* and *behavior description.*

- Property definition: Defining *static properties* and *dynamic properties* of every activity. The former refers to the properties that would not be changed during operation, such as activity ID and description of activity's function. The latter refers to those whose value be determined by run-time system status. Typical dynamic properties include activities' begin time, complete time and the priority.
- Resource mapping: Allocating resources to activities. It is a connection between the process view and the resource view, which describe resource constraints of scheduling problem.
- Behavior description: Defining the activity execution rules by using ECA rules, which is formed as "*if Event and Condition then Action*", where "event" refers to the events generated by the scheduling system, "condition" refers to activity enable conditions, and "action" means the activity's status is transferred from one to another.

(2) Resource View

An important feature of SMWMT is that it has independent resource view with the process view so as to manage the complexity more effectively. Two kinds of resource entities – the *Individual Resource* and the *Resource Pool* are introduced in the resource view. The individual resource refers to real equipments that participate in production. The resource pool is in fact a classification of individual resources according to their functions or geographical positions so that the individual resources in the same resource pool can be substituted for each other. It makes the model flexible in dealing with the "parallel machine" problem.

During the model definition period, static mapping from individual resources to resource pools and resource pools to activities are established separately. Specially, if there is no parallel machine in the problem, we can establish the mapping between the individual resources and the activities directly. Then during scheduling, the individual resource will be dynamically allocated to the activities based on the static definition and running time individual resource status.

(3) Job view

Job view describes the properties of jobs needed to be produced, which includes the jobs' arrival time, type, amount, due date, priority, cost, and so on.

120

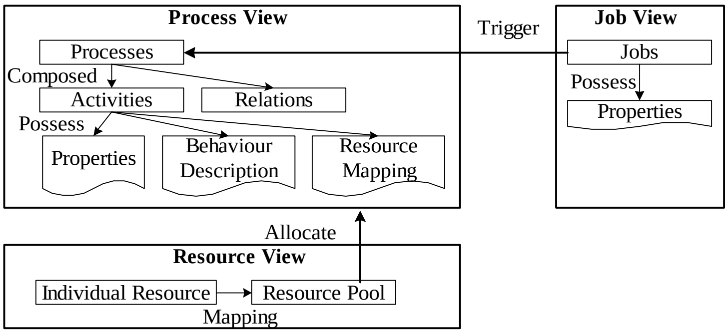

Fig 1 Structure of SMWMT

3.2 Formal Definition

A briefly formal definition of SMWMT is provided here.

Definition 1 *SMWMT*=(*PROCESS, RESOURCE, JOBS*), in which:

- $PROCESS = \{process_i \mid process_i = (N_i, A_i), i = 1,2,...n\}$, where

$N_i = \{n_{ij} \mid n_{ij} \in N_{task} \cup N_{start} \cup N_{end} \cup N_{and-split} \cup N_{and-join} \cup N_{or-split} \cup N_{or-split}, i, j = 1,2,...n\}$ is the set of all activities. $A_i \subseteq N_i \times N_i$ is the set of connections.

- $RESOURCE = \{r_i, rp_j, f \mid f = \{r_i\} \times \{rp_j\}, i, j = 1,2,...n\}$, where r_i and rp_j present the individual resource and resource pool respectively.
- $JOBS = \{job_{ij} \mid i, j = 1,2,...n\}$, where job_{ij} means the *j*th job associated with p*rocess*$_i$. Specially, if there is only one job connected with a process, the subscript *j* can be omitted.

Definition 2 $\forall n_{ij} \in N_{i\,task} \in process_i, Description(n_{ij}) = (PN_{ij}, f_{ij}, ECA_{nij})$, where

- PN_{ij}={*ID, FUNCTION*, $t_{ij\text{-}start}$, $t_{ij\text{-}end}$, p_{ij}, PrN_{ij}} indicating the activity's ID, function description, initiating time, end time, processing time, and priority respectively.

- $f_{ij} \in N \times RESOURCE$ is the mapping between activities and resources.
- ECA_{nij}=(*if Event and Condition then* δ), $\delta \subseteq S_{nij} \times S_{nij}$, S_{nij} is status set of n_{ij}.

Definition 3 $\forall job_{ij} \in JOBS$, *Description*(job_{ij}) = ($t_{ij\text{-}arrival}$, d_{ij}, q_{ij}, $t_{ij\text{-}complete}$, PrJ_{ij}, LC_i, SC_i), indicating the job's arrival time, due date, required job quantity, priority, tardiness penalty cost, job's earliness storage cost respectively.

3.3 Model Example and conclusions

A SMWMT modeling tool with friendly users' interface is developed to build the scheduling model. Suppose there are 5 jobs and six machines r_1, r_2, r_3, r_4, r_5 and r_6 available in the production, in which r_5 and r_6 can be replaced by each other. Then we can have 5 resource pools as rp_1, rp_2, rp_3, rp_4, and rp_5, where rp_5 contains r_5 and r_6. The jobs' routings are listed in Table1. Each activity's process time is drawn from a uniform distribution between 1.0 and 3.0 hours. The process view, the activity's definition interfaces, and the resource view can be showed in Figure 2.

Table 1 Description of the example

	Job1	Job2	Job3	Job4	Job5
1	(n11,rp1)	(n21,rp2)	(n31,rp2)	(n41,rp3)	(n51,rp2)
2	(n12,rp2)	(n22,rp3)	(n32,rp1)	(n42,rp4)	(n52,rp5)
3	(n13,rp3)	(n23,rp1)	(n33,rp4)	(n43,rp1)	(n53,rp3)
4	(n14,rp4)	(n24,rp5)	(n34,rp3)	(n44,rp5)	(n54,rp1)
5	(n15,rp5)	(n25,rp4)	(n35,rp5)	(n45,rp2)	(n55,rp4)

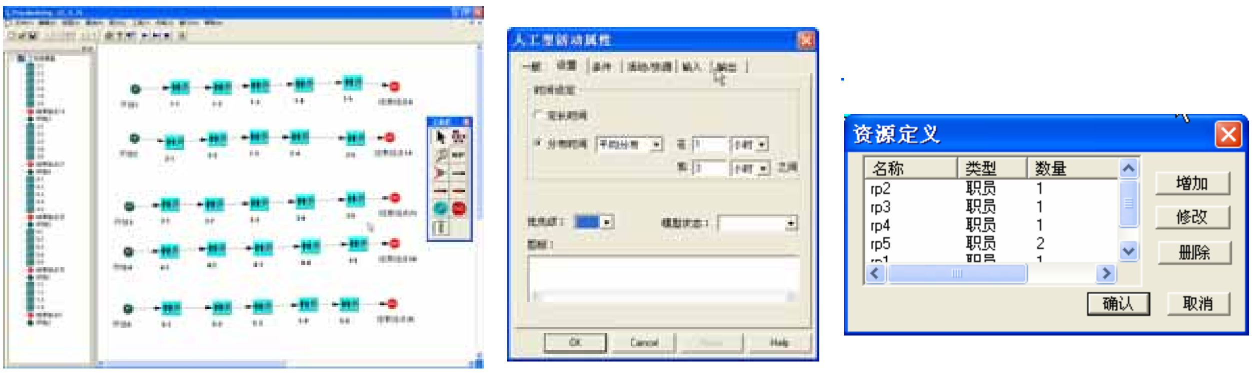

Fig2 Interfaces of process definition, activity definition, and resource definition

In summary, SMWMT is characterized as following features:

- Introduction of logic nodes allows SMWMT to deal with complicated process constraints.
- Activity behavior definitions make SMWMT an executable model.
- Independent resource view and process view and the mapping between them fit people's natural way of understanding manufacturing process.
- Graphic presentation of the process view offers an intuitional understanding to

users.

4 Scheduling Method based on SMWMT

One of the important reasons that we use workflow management technology for our scheduling method is that it provides a possible way to integrate scheduling with process management through sharing the same SMWMT model between the scheduler and the workflow engine. It not only reduces the work to establish two models, most importantly, it also helps to maintain the consistency between scheduling and process operation.

The SMWMT scheduler is in fact a workflow simulation tool integrated with certain scheduling policy. Simulation results are the task sheets that include information about where (which equipment) and when to execute what activity. They will be transferred to workflow engine, which executes and monitors the process later according to the same model.

4.1 Workflow Simulation based Scheduler

The simulation can be divided into two sections as initiation and running period. During initiation period, the following three steps are be done:

(1) Generating all resource pool instances and individual resource instances according to the resource view. Two kinds of resource waiting lists, virtual waiting list and real waiting list, are assigned to every individual resource instance.

(2) Generating job instances according to the job view definition.

(3) Initiating the related process instances and activity instances by job instances.

During running, the scheduler is a discrete event driven simulator that will generate and respond to event. The two (virtual/real) resource waiting lists are used to satisfy all process constraints through following operations: At the time when an activity is instanced, the activity instance is allocated to resource virtual waiting list. It can not be moved to real waiting list until its previous activities are completed (the process constraints being satisfied). Then, at any event that a resource is released, the activity instance with the highest priority only from its real waiting list will be selected as next activity to execute.

When an activity instance is going to be assigned with a resource's virtual waiting list, the current working load of each individual resource in the resource pool should be considered so that the individual resource with minim working load can be selected. Suppose *VW* is the set of all activity instances already existing in the resource's virtual waiting list, *RW* is the set of those in the resource's real waiting list, then the resource loading level can be calculated as:

$$Resource\,Work\ Load = \sum_{n_{ij} \in VW} p_{ij} + \sum_{n_{ij} \in RW} p_{ij}$$

4.2 Cost oriented iterative scheduling method

Dispatching rule will be used in our method to select the activity from resources' real waiting list. Dispatching rule [10] is convenient scheduling method that is widely used in industry. However, the major criticism of it is that there is no single universal rule, and the effectiveness of any rule depends on the scheduling criterion and the prevailing conditions in the job shop. In order to solve this problem, some the dispatching rules combination method [11] or selection algorithm [12, 13, and 14] were proposed in literature. These rules combination or selection algorithms are very complicated themselves and cost a lot of time.

There, we proposed a simple iterative scheduling algorithm to enhance the scheduling performance. The main feature of it is that it builds multi simulation runs and makes use of the previous simulation results, which have the information about how well the current rule works under the system status, to adjust the jobs' priority and activities' priority automatically. Because various cost information often offers a more realistic reflection on real business concerns, cost-oriented measurements such as earliness storage cost and tardiness penalty cost will be taken into account along with time related measurement.

Suppose we are in a make-to-order production environment. After a simulation run, the set of the tardy jobs is *L*, the set of the early jobs is *E*. The algorithm could be briefly described as follows.

Algorithm 1

Step1: Initial parameter

Suppose the jobs come to the system periodically. At a decision making point time

t, only those jobs with $t_{ij\text{-}arrival} < t$ will be considered. The initial priority of job_i is set as 1.0, that is PrJ_i =1.0. The initial priority of task n_{ij} is assigned as the combination of PrJ_i and EDD (Earliest Due Date) dispatching rule.

$$PrN_{ij} = PrJ_i / d_i$$

Step2: Run the simulation.

Step3: Reset the jobs' and the activities' priority by multiplying adjust function f :

$$f_i = \begin{cases} 1 + LC_i(t_{i-complete} - d_i) \Big/ \sum_{k \in L} LC_k(t_{k-complete} - d_k) \\ 1 - SC_i(d_i - t_{i-complete}) \Big/ \sum_{k \in E} SC_k(d_k - t_{k-complete}) \end{cases}$$

The job's new priority should be: $PrJ_i^{'} = PrJ_i \ f_i$

The activity's new priority should be: $PrN_{ij}^{'} = PrJ_i^{'} / d_i$

Step4: Return to Step 2, until preset maximum number of simulation runs is reached.

Step5: Select the best result from all the simulation runs.

4.3 Simulation implementation

For comparison purpose, four other dispatching rules, including EDD, SPT, SL/OP, and SLACK (Table 2) are also studied in the simulation experiments.

Table 2 Description of the rules

Dispatching Rule	Rank	Description
EDD (Earliest Due Date)	Max	$1/d_i$
SPT(Shortest Process Time)	Max	$1/p_{ij}$
SL/OP(Smallest Slack per Remaining Operation)	Max	$m_i / (d_i - t - \sum_{k=j}^{m_i} p_{ik})$
SLACK(Smallest Slack)	Max	$1/(d_i - t - \sum_{k=j}^{m_i} p_{ik})$

Note: m_i indicates the remaining operations

Our simulation experiments consist of two different machine cells with 5, 10 machines, and three types of jobs loading: 5, 10 and15. Each job's routing through the machines is previously determined by the SMWMT model. Processing times are uniformly distributed from 1 to 3 hours. Assume jobs' tardy penalty costs, per hour of each job, are drawn from a uniform distribution between 0.0 and 100.0 cost unit, and early storage costs is between 0.0 to 20.0 cost unit. Because tight due date often provide a more severe scheduling environment, we assign the due date over a uniform distribution between 1.0 to 1.5 times the total job processing times. The preset maximum simulation run is 10.

Two performance measures are defined to test the effectiveness of our methods.

(1) Total Cost (*TC*).

$$TC = \sum_{i}^{i \in L} LC_i (t_{i-complete} - d_i) + \sum_{i}^{i \in E} SC_i (d_i - t_{i-complete})$$

(2) Total Absolute Difference (*TAD*) between the jobs' complete time and their due date.

$$TAD = \sum_i | t_{i-complete} - d_i |$$

Total Cost Index and TAD Index are considered in simulation results (Table 3).

Table 3 Simulation Results

(a) Total Cost Index $\alpha = TC^i / TC^{iterative}, i = EDD, SPT, SLACK, SL/OP$

α / Rules	5 Machines			10 Machines		
	5 Jobs	10 Jobs	15 Jobs	5 Jobs	10 Jobs	15 Jobs
Iterative	-	-	-	-	-	-
EDD	1.60	1.14	1.07	1.17	1.07	1.03
SPT	1.11	1.21	1.28	1.00	1.37	1.57
SLACK	1.09	1.48	1.63	1.03	1.41	1.59
SL/OP	1.09	1.51	1.72	1.03	1.49	1.61

(b) TAD Index $\beta = TAD^i / TAD^{iterative}, i = EDD, SPT, SLACK, SL/OP$

β / Rules	5 Machines			10 Machines		
	5 Jobs	10 Jobs	15 Jobs	5 Jobs	10 Jobs	15 Jobs
Iterative	-	-		-	-	-
EDD	1.07	1.03	0.93	1.17	1.04	1.03
SPT	0.93	1.01	1.03	1.00	1.32	1.57
SLACK	1.01	1.04	1.07	1.03	1.32	1.59
SL/OP	1.01	1.03	1.06	1.03	1.35	1.61

From the results we can tell that while single priority rule's performance varies under different scheduling environment, our method works better than the other four rules in most of the cases. The iterative scheduling method based on SMWMT is effective.

5 Conclusions

In this paper, a new scheduling model SMWMT based on workflow modeling technology is proposed in order to meet industry requirement of "capable" and "practical". It is a compound model made up of the process view, resource view and job view. Logic nodes and ECA rules are adopted to build the process view so that it can deal with various process constraints and activities' dynamic behaviors. The individual resource, resource pool, and resource mapping are introduced in the resource view to make flexible resource definition. The main feature of it is that it capable of describing complicated manufacturing system as well as offing an graphic process presentation to users. Also it is an executable model that can be used for process management further, which helps to maintain the consistency between the scheduling and process operation.

The related scheduling system based on workflow simulation is proposed. The establishment of the resource virtual waiting list and real waiting list guarantees all the process constraints are satisfied. The dispatching rule will be used to determine the activities sequence. In order to enhance dispatching rule performance, a cost oriented iterative scheduling method is put forward. It constructs multi-simulation runs and makes use of previous simulation results to adjust the jobs' and the activities' priority automatically.

The SMWMT example model is given. The simulation results show the proposed iterative algorithm has better performance than some other dispatching rule like EDD, SPT, SL/OP, and SLACK under various conditions. Thus, the proposed modeling method and scheduling algorithm is effective.

References

1. F. Glover. Tabu Search – Part I. ORSA Journal on Computing 1989(1), 190~206
2. J. H. Holland. Adaptation in natural and artificial systems. 1st ed., 1975, 2nd ed., Cambridge, MA: MIT press, 1992
3. J. Hopfield, D. Tank. “Neural” computation of decisions in optimization problems. Biological Cybernetics, 1985, 52: 141~152
4. J. M. Proth, I. Minis. Planning and scheduling based on Petri nets. Petri Nets in Flexible & Agile Automation, M C Zhou, ed., Norwell, MA: Kluwer Academic, 1995, 109~148
5. D. Y. Lee, F. DiCesare. Scheduling FMS using Petri nets and heuristic search. IEEE Trans. On Robotics and Automation, 1994,Vol. 10, No. 2, 123~132
6. C. M. Zhou , V. Kurapati. Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach. World Scientific, 1999
7. Workflow Management Coalition. The workflow reference model. WFMC TC00-1003, 1994.
8. C. Mohan. Recent Trends in Workflow Management Products, Standards, and Research. Available at URL: http://www.almaden.ibm.com/u/mohan/wfnato97.ps
9. Y. S. Fan, H. B. Luo , H. P. Lin, and etc. Fundamentals of Workflow Management Technology. Tsinghua University Publisher, Springer Publisher, 2001
10. S. Panwalkar, I. Wafik, A survey of scheduling rules. Operations Research, 1977, 25(1), 45-61.
11. O. Holthaus. Design of efficient job-shop scheduling rules. Computer & Industrial Engineering, 1997, 33(1-2), 249-252.
12. S. C. Park, N. Raman, M Shaw. Adaptive scheduling in dynamic flexible manufacturing systems: A dynamic rule selection approach. IEEE Transactions on Robotics and Automation, 1997, 13(4), 486-502.
13. B. Grabot., L. Geneste. Dispatching rules in scheduling: a fuzzy approach. International Journal of Production Research, 1994, 32(4), 903-915.
14. V. Subramaniam, T. Ramesh. Job shop scheduling with dynamic fuzzy selection of dispatching rules. The International Journal of Advanced Manufacturing Technology, 2000, 16, 759-764.