DYNAMIC NEGOTIATION FOR REAL-TIME MANUFACTURING EXECUTION

L.Q. Zhuang, J.B. Zhang, B.T.J. Ng, Y. Tang, Y.Z. Zhao
Singapore Institute of Manufacturing Technology, 71 Nanyang Drive, Singapore

Keywords: Multi-Agent System, Performance and Cost for Manufacturing Execution, Agent Consortium, Dynamic Negotiation

Abstract: This paper presents a dynamic negotiation framework for real-time execution in self-organised manufacturing environments. The negotiation strategies in this framework bridge the gap between distributed negotiation of self-interested agents and cooperative negotiation among agent groups. In particular, the proposed framework is based on the model of Performance and Cost for Manufacturing Execution (PCME). By forming the dynamic organisation called agent consortium, individual agents negotiate over the PCME in order to optimise the resource allocation under time constraints and uncertainty of job execution, and resolve the conflicts to fulfil the goal of the overall system. The ultimate goal of the framework is to reduce the negotiation time, make effective use of resources, adapt to the changes in execution and increase the throughput of the entire system. Experimental work based on PCME has been carried out to demonstrate the high performance of this approach despite unanticipated and dynamic changes in the manufacturing execution environments.

1 INTRODUCTION

In modern complex manufacturing execution environments, autonomous or self-organised systems are being deployed to deal with increasingly diverse operations and to control real-time dynamic situations under uncertainty. These real-time tasks are further complicated by coalition coordination. The need for more flexibility, robustness and scalability and the trend to handle increasing complexity is driving research into the area of distributed intelligent computation.

Distributed artificial intelligence (DAI) is the emerging information technology to meet the new challenges. Over the past few years, agent-based computing has been hailed as the next significant breakthrough in areas of DAI (Sargent, 1992). Intelligent agents and multi-agent systems (MAS) are able to conduct independent jobs in open and unpredictable environments. The properties of an intelligent agent, such as social ability, mobility, autonomy, reactivity and pro-activeness have made MAS a very relevant technology for various manufacturing domains such as enterprise integration, manufacturing planning, scheduling and control, and holonic manufacturing systems (Shen, 1999). The MAS architecture is able to adapt itself to changes and disturbances in the manufacturing execution such as dynamic execution changes, process changes and equipment failures etc. The MAS architecture is also able to model the manufacturing execution processes in distributed ways to reduce the complexity of manufacturing systems and to increase the interoperability between the heterogeneous systems at the same time. As such, more and more researchers are introducing the agent technologies and MAS architecture to real-time manufacturing areas, including manufacturing execution, automated material handling and autonomous robotic control systems (Deen, 2003).

Odell proposed “The Agile Manufacturing Information System”, an agent-based model, which defined cell agent architecture as self-contained unit that had its own structure and behaviour (Odell, 2002). This model provided a conceptual architecture and a general approach for agent-based manufacturing systems. Jennings et al. proposed the

Q. Zhuang L., B. Zhang J., T. J. Ng B., Tang Y. and Z. Zhao Y. (2004).
DYNAMIC NEGOTIATION FOR REAL-TIME MANUFACTURING EXECUTION.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 321-326
DOI: 10.5220/0002651003210326

agent-based control systems for electricity transportation management and manufacturing line control (Jennings 2003). The proposed negotiation protocols between the agents in this distributed manufacturing system had proven the effectiveness of the framework. Fatima et al. introduced an organisational policy known as TRACE (Task and Resource Allocation in Computational Economy) (Fatima 2001). The task allocation in the model is NP-complete; hence the centralised solutions to the problem are not feasible. TRACE is able to adapt itself to any changes in the computational load by reorganising the MAS. Dias et al. used market-based coordination mechanism for optimisation of task and resource allocation for multi-robot control systems (Dias 2002). The optimisation for the activities of robots can be achieved in an adaptive way. Ng et al. proposed a framework called Self-Organising Multi-Equipment Control (SOMEC) for the holonic manufacturing systems (Ng, 2003). Intelligent units (IU) in a set of equipment worked together to achieve a global goal via the cooperative negotiation approach. The above research work has demonstrated that MAS is an effective architecture to handle the adaptive manufacturing execution processes in dynamic and uncertain environments. However most of above research work only used the fixed negotiation strategies for the different manufacturing execution aspects and did not provide goal recovery mechanism and time boundary of negotiation for the autonomous agents.

For real-time manufacturing execution systems, the response time is one of the important issues in order to fulfil the manufacturing execution targets. The normal timeframe for each task is restricted within a few seconds for most of the execution activities. Hence MAS architecture to support the above manufacturing system only allows very limited time for agents to interact with each other. On the other hand, the resource utilisations within the execution domain need to be optimised as well. So the agent negotiation and coordination in such a framework must be carried out in an adaptive way. Traditional global optimisation techniques used in the planning and scheduling systems are not suitable for the resource and task allocation in the real-time execution environment.

Agent negotiation and coordination for real-time execution requires a unique mechanism. In this paper we propose a multi-phase dynamic negotiation framework that is focused on the fast and flexible decision making aspect of the system so that it is responsive enough for the time critical tasks and also adaptive for dynamic environments. In the proposed framework, a specific virtual organisation concept called agent consortium is introduced for the MAS architecture. The PCME is applied as a measurement for agents to negotiate among the consortia and the PCME value can be calculated based on the structure of each consortium.

The remainder of the paper presents details of the framework that can be used to coordinate the execution activities of physical equipment units in the manufacturing environment. Section 2 defines the mathematical model for the dynamic negotiation. Section 3 defines the multi-phase negotiation strategies for real-time manufacturing execution. In Section 4 we demonstrate the experimental work of applying the proposed framework for an autonomous Automated Storage and Retrieval Systems (ASRS). In Section 5 the benefits of the proposed framework will be summarised.

2 DYNAMIC NEGOTIATION MODEL

Through decomposition, abstraction and organisation, the traditional manufacturing execution system that is organised in the hierarchical structure can be transformed into the MAS architecture. Dynamic integrative negotiation strategies are used in the proposed framework to reach the contract agreement in a co-operative way at the interests of global goals, whereas individual agents are responsible for allocating resources to ensure their own interests when they carry out the tasks. The proposed dynamic negotiation strategies aim to strike a balance between the distributed negotiation and the cooperative negotiation so that a more effective and adaptive mechanism can emerge in the new framework.

2.1 Agent Consortium

The agent consortium is defined as a group of relevant agents with some capabilities to fulfil the specific job in the system. The proposed approach defines an initiator agent for each job. The initiator agent issues the job contract with a specific workflow. According to the workflow requirement, each agent that is qualified for the job joins a consortium with other agents involved in the job processes. So the consortium can be formed by relevant agents in form of a partially ordered set (poset) according to the workflow dependencies. The agent consortium can be represented by a directed graph (digraph, see an example in Figure 1). More than one consortium could be formed for each job. In Figure 1, a node from P1 to P9 represents the agent for the different cell controller or equipment.

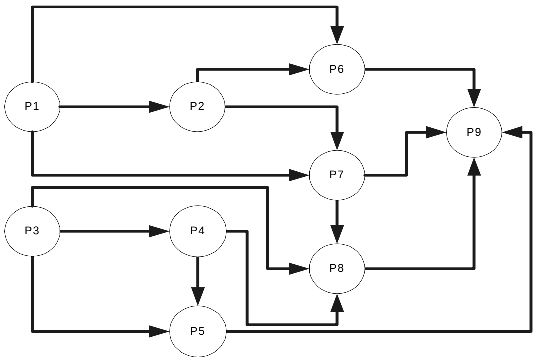

Figure 1: Digraph for an agent consortium.

The digraph representing the agent consortium can be further transformed to an upward drawing, which is called the Hasse diagram in order to show the workflow dependencies. Figure 2 shows the Hasse diagram that is converted from digraph in Figure 1.

Based on the Hasse diagram constructed for the agent consortium digraph, the topological sorting of the agent members can be generated for the above case: {P1, P3, P2, P4, P6, P7, P5, P8, P9}.

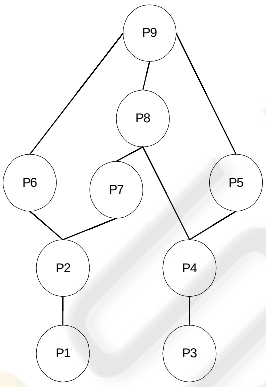

Figure 2: A Hasse diagram for the agent consortium.

In the above case, the poset can be further divided into five logical groups: G_1: {P1, P3}, G_2: {P2, P4}, G_3: {P5, P6, P7}, G_4: {P8} and G_5 {P9}. Hence, the sequential relationship can be set up among the logical groups. The concurrent and redundant relationship can be set up among the agents in each group.

The following procedure shows the algorithm to form the logical groups for a consortium.

```
; Logical Grouping for Hasse Diagram
proc LG(S:finite poset)
     k:=1;
     while S<>Φ
     begin
         Gk:= set of minimal element of S
         S: = S - Gk
         k:= k +1
     end
endproc
```

Definition for minimal element: Let <A, ≤> be a poset, where ≤ represents an arbitrary partial order. Then an element b ∈ A is a minimal element of A if there is no element a ∈ A that satisfies a ≤ b.

2.2 Performance and Cost for Manufacturing Execution

In the proposed framework, four performance indicators are defined for the PCME model: time for process, resource cost, system reliability and process throughput. Three fundamental structures are defined as the foundation for the PCME model: the sequential structure, the concurrent structure and the redundant structure.

Figure 3 shows the sequential structure in the PCME model. PCME calculation formulas are shown in (1) – (4).

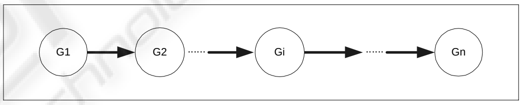

Figure 3: PCME for sequential structure.

$$\text{Time for Process} = \sum_{i=1}^{n} T(G_i) \tag{1}$$

$$\text{Resource Cost} = \sum_{i=1}^{n} C(G_i) \tag{2}$$

$$\text{System Reliability} = \prod_{i=1}^{n} R(G_i) \tag{3}$$

$$\text{Process Throughput} = \frac{u}{\sum_{i=1}^{n} T(G_i)} \tag{4}$$

In (4), u is the maximum number of units that can be processed within the timeframe T.

Where $1 < u < n$ and $T = \sum_{i=1}^{n} T(G_i)$

Figure 4 shows the concurrent structure in the PCME model. PCME calculation formulas are shown in (5) – (8).

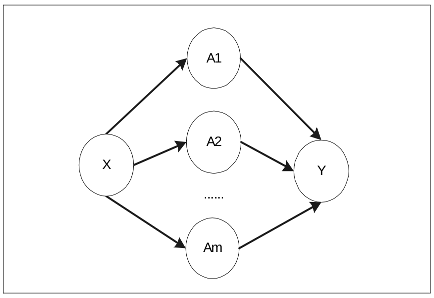

Figure 4: PCME for concurrent structure.

Time for Process=Max{T(A_i)}(i∈ {1,2,…m})(5)

Resource Cost = $\sum_{i=1}^{m} C(A_i)$ (6)

System Reliability = $\prod_{i=1}^{m} R(A_i)$ (7)

Process Throughput = $\frac{m}{Max\ \{T\ (A_i)\}}$ (8)

Figure 5 shows the redundant structure in the PCME model. PCME calculation formulas are shown in (9) – (13).

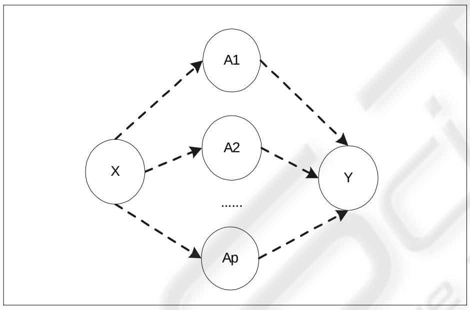

Figure 5: PCME for redundant structure.

Time for Process=Min{T(A_i)}(i∈ {1,2,…p}) (9)

Resource Cost = Max{C(A_i)} i∈ {1,2,…p}) (10)

System Reliability = 1 – $\prod_{i=1}^{p} F(A_i)$ (11)

Where F(A_i) = 1 – R(A_i) (12)

Process Throughput = $\frac{1}{Min\ \{T\ (A_i)\}}$ (13)

2.3 Performance Level for Equipment Execution

For individual equipment, the performance level for equipment execution can be measured by the extended and enhanced quality of service (QoS) concept for the manufacturing execution, which was introduced by Wong et al. (Wong, 2003). The performance levels will be taken as negotiation objects in the agent negotiation for the low-level equipment execution. The formula to calculate the performance level is shown in (14).

$$PL(t) = \frac{\sum_{i=1}^{N} (\ p_i(t) \times \omega_i)}{N} \quad (14)$$

Where

PL(t) = the performance level at time t

$p_i(t)$ = points for the award or the penalty for i^{th} factor at time t

ω_i = coefficients for i^{th} factor

N = number of the award and the penalty factors

3 NEGOTIATION PROCESSES

The proposed dynamic negotiation can be viewed as distributed search in the space of PCME as well as performance levels of manufacturing execution (Jennings, 2001). The negotiation objects are the performance indicators defined in the PCME model as well as capabilities of the equipment. A decision-making model of agents is designed in two levels: at the consortium level, it will follow the cooperative negotiation approach, and at logical group level, it will follow the self-interested negotiation approach.

3.1 Cooperative Negotiation Approach

Every job generated in the autonomous manufacturing system is associated with a contract. Contract negotiation (Mathieu, 2002) is carried out by a group of agents that perform specific tasks at different stages of the whole contract in a dynamic environment with common resources. Depending on its position in the consortium, each agent may assume the dual roles of being an initiator and a consortium member. The agent members in the consortium collectively issue a bid using the PCME-based reasoning and the contract will be granted to one of the consortia with the best PCME results.

A 6-tuple $M = (J, I_0, G, f, C, A)$ is defined for the modelling of the consortium, where J is a set of job contracts, G is goal of contract, $I_0 \in A$ is initial agent, A is set of all agents, $C \subseteq A$ is consortium and $f: J \times I_0 \times G \rightarrow C$. This model is the basis to dynamically generate feasible consortia.

The maximum four-round negotiation strategies are defined for the framework at the consortium level. The negotiation search space in the next round will be smaller if the negotiation is not resolved in the current round.

Strategies for the 1st round negotiation: To negotiate over the time for process in PCME model and form a set of consortia in terms of Hasse diagrams. Go to the 2nd round negotiation if more than one consortium is formed, or reach contract agreement and go to the group level negotiation.

Strategies for the 2nd round negotiation: To negotiate over the cost in PCME model among the set of consortia in the 1st round negotiation and form a subset of consortia in terms of Hasse diagrams. Go to the 3rd round negotiation if more than one consortium is formed, or reach contract agreement and go to the group level negotiation.

Strategies for the 3rd round negotiation: To negotiate over the reliability in PCME model among the set of consortia in the 2nd round negotiation and form a subset of consortia in terms of Hasse diagrams. Go to the 4th round negotiation if more than one consortium is formed, or reach contract agreement and go to the group level negotiation.

Strategies for the 4th round negotiation: To negotiate over process the throughput in PCME model among the set of consortia in the 3rd round negotiation and form a subset of consortia in terms of Hasse diagrams. Reach contract agreement and go to the group level negotiation.

3.2 Self-Interested Negotiation

Once contract agreement is reached and the job is granted, agents in the consortium will negotiate internally by their own interests for tasks of the job. The negotiation objects are based on the performance level each agent can provide. Individual agent negotiates over performance level to reach agreement for task execution. As the result, one agent may join multiple consortia and hold more than one job at certain point of time.

The finite state automaton (FSA) can represent the operation model of the agent. Let $A = (Y, \Sigma, \eta, y_0, Y_m)$ be a 5-tuple, where Y is the set of equipment states, Σ is the set of actions, η is the set of process transitions ($\eta: Y \times \Sigma^* \rightarrow Y$), q_0 is the initial state of the equipment and $Y_m \subseteq Y$ is the set of final equipment status.

3.3 Enforcement and Re-negotiation

In practice, the actions of the agents always have a temporal extent. The FSA can be extended to a 7-tuple timed automaton (Alur & Dill, 1994): $TA = (Y, \Sigma, \eta, y_0, Y_m, X, D)$ where X is a finite set of clocks and D is set of other data structures.

Timed action should have some checkpoints and intermediate effects could be incorporated with the agent state at these checkpoints. The temporal logic is used to describe the time constraint of the actions (Subrahmanian, 2000; Collins, 2002; Dix, 2001).

The goal of job is verified upon state changes of the timed automaton. If the goal is not met after certain stage, the agent will try for the goal recovery by re-negotiation within the same cluster, which is represented by a redundant structure. Another agent in the same cluster will take the enforcement action.

4 IMPLEMENTATION AND RESULTS

Current research project used an ASRS environment as the test platform. It integrates with assembly lines, packing areas, kitting areas and incoming/outgoing interfaces. Figure 6 shows the layout of one level of the entire ASRS architecture.

PCME specifications and performance levels are defined for each category of the automated equipment such as Rail Guided Vehicle (RGV) and Stacker Crane (SC). The following performance level specifications are defined and used as parameters for the negotiation strategies: job execution priority, number of jobs on hold, position on storage, distance and congestion.

The Java agent development framework (JADE) was used as development toolkit for the test.

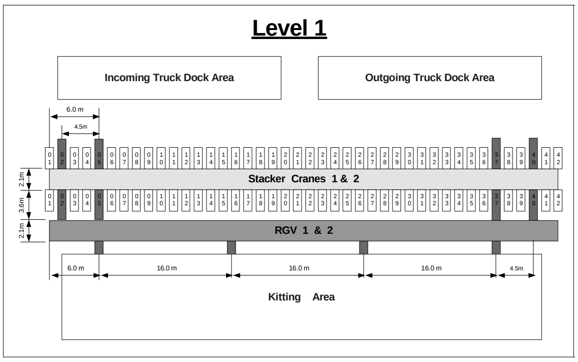

Figure 6: Level 1 of ASRS test environment

We used 30, 60 and 90 pallets for different test loadings. The test results are shown in table 1.

Table 1: Result for number of negotiation rounds.

Pallets	2-round	3-round	4-round	>4
30	86.7%	13.3%	0%	0%
60	66.7%	26.7%	6.6%	0%
90	53.3%	28.9%	11.1%	6.7%

The average time spent on negotiation is shown in table 2. The average time spent on handling is shown in table 3.

Table 2: Result for average negotiation time.

Initiator	Negotiation Time (Old)	Negotiation Time (New)
RGV	6.36 (seconds)	3.12 (seconds)
Stacker Crane	8.22 (seconds)	3.68 (seconds)

Table 3: Result for average handling time.

Initiator	Handling Time (Old)	Handling Time (New)
RGV	56.76 (seconds)	43.32 (seconds)
Stacker Crane	68.92 (seconds)	48.90 (seconds)

5 CONCLUSION

This paper presents a PCME-based dynamic negotiation approach that is particularly applicable to the distributed manufacturing system, which is dynamic and time-critical in nature. The research work uses a real-time multi-equipment material handling system as a test platform. This system is time-critical in operation and therefore, requires an adaptive, fast and efficient decision-making mechanism.

The approach discussed in this paper adopts the strategy of balancing of the cooperative negotiation and the self-interested negotiation. It also effectively sets the boundary of negotiation and reduces the rounds of negotiation through the use of PCME as the assessment criteria.

The dynamic negotiation approach has been applied in the execution control of an ASRS system. The results of the experiments show that this approach is sufficiently efficient and has achieved higher percentage of goal attaining in terms of average task execution time.

A test model with more sophisticated environment is being built for the future research work.

REFERENCES

Sargent, P., 1992. Back to school for a brand new ABC. In *The Gauardian.*

Shen, W.M. and Norrie, D.H., D.H., 1999. Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. In *International Journal of Knowledge and Information System.* Vol. 1(2), pp. 129-156.

Deen, S.M., 2003. *Agent-based Manufacturing,* Springer Verlag. Heidelberg, 1st edition.

Odell, J.J., 2002. Agent-Based Manufacturing: A Case Study. In *Journal of Object Technology.* Vol. 1(5), pp. 51-61.

Jennings, N.R., 2003. Agent-Based Control Systems. In *IEEE Control System Magazine.* Vol. 23(3), pp. 61-74

Fatima, S.S. and Wooldridge, M., 2001. Adaptive Task and Resource Allocation in Multi-Agent Systems. In *the fifth International Conference on Autonomous Agents.*

Dias, M.B., 2002. Opportunistic Optimization for Market-Based Multirobot Control. In *IROS2002, International Conference on Intelligent Robots and Systems*.

Ng, B.T.J., Zhang, J.B., Lin, W.J., Wong, M.M., Luo, M. and Ma, H., 2003. Fast Self-Organizing Holonic Based Multi-Equipment Control. In *ICCA'03, the Fourth International Conference on Control and Automation.*

Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., and Wooldridge, M., 2001. Automated negotiation: prospects, methods and challenges. In *International Journal of Group Decision and Negotiation.* Vol. 10(2), pp. 199-215.

Wong, M.M., Zhang, J.B., Tang, Y., Zhuang, L.Q., 2003. A QoS-aware Dynamic Transfer Order Optimisation Methodology for Automated Material-handling Systems. In *ICCA'03, the Fourth International Conference on Control and Automation.*

Mathieu, P., and Verrons, M.H., 2002. A genetic model for contract negotiation. In *AISB2002, Artificial Intelligence and the Simulation Behaviour.*

Alur, R., and Dill, D., 1994. Automata for Modelling Real-time Systems. In *Theoretical Computer Science.* Vol. 126(2), pp. 183-236.

Subrahmanian, V.S., Bonatti, P., Dix, J., Eiter, T., Kraus, S., and Ozman, F., 2000. *Heterogeneous Agent Systems.* MIT Press, Boston 1st edition.

Collins, J., Ketter, W., and Gini, M., 2002. A Multi-agent Negotiation Testbed for Contracting Tasks with Temporal and Precedence Constraints. In *International Journal of Electronic Commerce.* Vol. 7(1), pp. 35-57.

Dix, J., Kraus, S., and Subrahmanian, V.S., 2001. Temporal Agent Programs. In *Artificial Intelligence Journal.* Vol. 127(1), pp. 87-135.