Journal of Manufacturing Systems 32 (2013) 752–763

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical Paper

A bio-inspired mobile agent-based integrated system for flexible autonomic job shop scheduling

Yu-Cheng Chou [a,*], Huajun Cao [b], Harry H. Cheng [c]

[a] Department of Mechanical Engineering, Research and Development Center for Microsystem Reliability, Institute of Biomedical Technology, Chung Yuan Christian University, Zhongli, Taoyuan 32023, Taiwan
[b] Institute of Manufacturing Engineering, Chongqing University, Chongqing 400044, China
[c] Department of Mechanical and Aerospace Engineering, University of California at Davis, Davis, CA 95616, USA

ARTICLE INFO

Article history:
Received 11 December 2011
Received in revised form 13 January 2013
Accepted 22 January 2013
Available online 15 March 2013

Keywords:
Autonomic system
Mobile agent-based system
Distributed
Manufacturing
Flexible job shop scheduling
Dynamic job shop scheduling
Autonomic computing

ABSTRACT

This paper presents a bio-inspired mobile agent-based integrated system for flexible autonomic job shop scheduling. The system matches the autonomic system architecture, inspired by the autonomic nervous system and proposed by the IBM, and has the IBM-defined fundamental self-managing properties, so that it can manage itself with little human intervention. The system conforms to the IEEE FIPA (Foundation for Intelligent Physical Agents) standard. Therefore, the interoperability between agents of the system and agents from many active heterogeneous FIPA compliant agent platforms can be ensured. The system supports the execution of C/C++ mobile agent codes. Thus, it is applicable to a variety of applications, especially for distributed mechatronic and embedded systems. In addition, since the system is composed of agents, including stationary and mobile agents, the system has a high scalability and flexibility to integrate and adopt various scheduling models and algorithms for different scheduling requirements. An overall architecture of the system and critical implementation details about the agency and agents in the system are presented in this article. An energy saving job shop scheduling example is used to validate one autonomic property of the system.

1. Introduction

As the size of manufacturing systems rapidly increases, it has been the trend to adopt decentralized and distributed techniques to control and manage those systems in order to reduce complexity and cost, increase flexibility, and enhance fault tolerance. Meanwhile, most manufacturing systems, such as job shops, operate in a dynamic environment where inevitable and unpredictable disturbances necessitate the revision of established schedules during manufacturing processes.

An agent is an encapsulated computer system that can autonomously perform reactive, proactive, and social actions in its execution environment [1]. In the area of design and manufacturing, a manufacturing resource, such as a machine or an operator, may cooperate and negotiate with other agents for task assignment; an existing engineering software may be integrated with a distributed design and manufacturing system. As opposed to traditional systems, agent-based systems do not have a centralized system control structure and pre-defined agenda for the system execution [2]. Thus, agent technology can significantly enhance the design and analysis of systems whose problem domain is distributed, and whose subsystems exist in a dynamic environment and need to interact with each other more flexibly [3].

A comparative study showed that multi-agent approach is a promising technique in dynamic manufacturing scheduling due to its autonomy, flexibility, modularity, robustness, and heterogeneity [4]. In addition, agent-based approaches have the following advantages over traditional approaches for distributed manufacturing scheduling [5,6].

(1) They facilitate the building of robust and efficient scheduling systems, because they employ parallel computation through a large number of processors.
(2) They facilitate the integration of manufacturing process planning and scheduling.
(3) They lead to cooperative scheduling, because they allow individual resources to trade off local performance to improve global performance.
(4) They facilitate the building of reliable and fault-tolerant scheduling systems, because agents can directly connect to physical devices in order to realize real-time dynamic rescheduling.

Abbreviations: SCY, scheduling center agency; SCT, scheduling center agent; MCY, manufacturing cell agency; MCT, manufacturing cell agent; BT, bidder agent; OT, order agent.

* Corresponding author. Tel.: +886 3 2654316.
E-mail addresses: ycchou@cycu.edu.tw (Y.-C. Chou), hjcao@cqu.edu.cn (H. Cao), hhcheng@ucdavis.edu (H.H. Cheng).

0278-6125/$ – see front matter © 2013 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jmsy.2013.01.005

(5) Schedules are obtained by negotiation rather than search. Thus, the manufacturing capabilities of manufacturers can be shared with each other, and optimization is allowed across all levels, namely, the supply chain level, shop floor level and enterprise level.
(6) Different techniques, such as simulated annealing [7] and genetic algorithms [8,9], can be incorporated at different levels for decision-making purposes.

Listed below introduces some multi-agent based approaches employed in dynamic manufacturing scheduling. A just-in-time dynamic scheduling approach was proposed in [10], where jobs and machines were modeled as agents that negotiate using market-based contract net protocol (CNP). A dynamic scheduling approach based on mediator and contract net protocol was presented in [11]. The approach models parts and machines as agents, and machine mediator and resource mediator as manager and coordinator of agents, respectively. A response threshold model for dynamic scheduling with flexible routing and sequence dependent setups was introduced in [12], where only machines were modeled as agents. An Ant Colony Intelligence model was presented in [13], where both machine selection and job sequencing problems were considered. A hierarchical control architecture with five CNP rules for flexible manufacturing system (FMS) scheduling was proposed in [14], where both job selection and machine selection rules were considered. A multi-agent system (MAS), called MASDSche-GATS, for distributed manufacturing scheduling with genetic algorithm and Tabu search was introduced in [15,16], where agents were able to find the local solution by genetic algorithm or Tabu search, and cooperated with one another to achieve global schedule. A MAS approach that modeled both tasks and resources as agents was presented in [17], where ant colony optimization was adopted to schedule both original and dynamic tasks. A multi-agent scheduling system was developed in [18] to solve job shop scheduling problems that consider dynamic events as well as routing and process flexibility. In [19], an agent-based parallel genetic algorithm for job shop scheduling problems was proposed, which improved the performance and quality of solutions for a genetic algorithm. A multi-agent Tabu search model, called FJS MATSLO+, was introduced in [20], where local optimization and new diversification techniques were employed to solve flexible job shop problems. In [21], a parallel modular simulated annealing algorithm was developed to tackle classical job shop problems through multi-agent systems. In [22], an agent-based service-oriented integration architecture was proposed to leverage manufacturing scheduling services on a network of virtual enterprises.

Many multi-agent systems applied to manufacturing are based on stationary agents. Mobile agents have all the features of stationary agents with an additional mobility attribute that creates the following advantages over stationary agents:

(1) *Local interaction*: A mobile agent can go to a remote machine to interact locally with other agents residing in that machine, thereby reducing the network load.
(2) *Parallel execution*: Multiple mobile agents can go to different remote machines to perform same or different tasks in parallel, thereby enhancing the efficiency.
(3) *Disconnected operation*: A mobile agent can operate without an active connection between itself and the machine where it is created.

An investigation showed that mobile agent technology can improve system integration and agility in the distributed manufacturing domain [23]. Several mobile agent-based manufacturing systems or mechanisms were developed in the past. A multi-agent protocol, ECNPro (the Extended Contract-Net-like multilateral Protocol) [24], was developed for handling buyer–seller negotiations in supply chain management. A software architecture, A^3M [25], was proposed to handle, within a single manufacturing cell, automatic assignment of control tasks to controllers, monitoring of cell functionalities, and dynamic cell reconfiguration. An integrated hierarchical framework based on mobile agent technology was proposed as an approach to resolve problems of scalability and management efficiency in large-scale networked manufacturing domains [26]. A mobile agent-based manufacturing decision system was prototyped to demonstrate the ability of mobile agents to support interoperable STEP-NC compliant manufacturing [27]. A flexible maintenance system integrating mobile agent technology and wireless sensor network was developed and applied to a real-world numerical control machining center [28]. A manufacturing control system was designed as a self-organizing multi-agent system where three main kinds of mobile agents were employed for coordination purposes [29]. A negotiation mechanism, MAN-Pro (Mobile Agent-based Negotiation Process) [30], was proposed to handle the information exchange in a shop floor control system. A two-level scheduling model was implemented using mobile agents to handle high-level process scheduling and low-level node scheduling for manufacturing chains [31].

This paper presents an autonomic mobile agent-based system for distributed job shop scheduling. The main features of the presented system, that are different from those of aforementioned systems or mechanisms, include:

1. The system matches the autonomic system reference architecture, which was inspired by the human body's autonomic nervous system and proposed by the IBM. Thus, the system has the four IBM-defined fundamental self-managing properties, namely, self-configuration, self-optimization, self-healing, and self-protection [32,33]. This system can manage itself with little human intervention.
2. The system conforms to the IEEE FIPA (Foundation for Intelligent Physical Agents) standard [34]. This compliance ensures the interoperability between its agents and other agents from an increasing number of FIPA compliant, heterogeneous agent platforms.
3. The system supports the execution of C/C++ mobile agent codes. This feature enables the system to be applicable to a wide range of applications due to C/C++'s comprehensive functionality, broad use, and international standard.

The rest of the article is organized as follows. Section 2 illustrates the system architecture. Section 3 presents the system implementation. Section 4 validates the presented system's self-configuration property using an energy saving job shop scheduling example. Section 5 draws the conclusion of this article.

2. System architecture

This section illustrates the architecture of the prototype autonomic mobile agent-based system. As shown in Fig. 1, the system contains two kinds of agencies, scheduling center agency (SCY) and manufacturing cell agency (MCY). An agency in this article is an agent platform where agents operate. In this prototype system, there is one scheduling center that deals with job shop scheduling in multiple manufacturing cells. Thus, there are one SCY and multiple MCYs, MCY_1 to MCY_N, in Fig. 1. The scheduling center agent (SCT) and manufacturing cell agent (MCT) are stationary agents. They are not dynamically generated and do not travel to other agencies.

When the SCT receives a new order from the order database, it creates as many bidder agents (BTs) as MCTs. A BT is one kind of