PERGAMON

Computers & Industrial Engineering 38 (2000) 107–123

computers & industrial engineering

www.elsevier.com/locate/dsw

A generic event control framework for modular flexible manufacturing systems

Jonghun Park [a,*], Jinwoo Park [b], Jongwon Kim [c]

[a]School of Industrial and Systems Engineering, Georgia Institute of Technology, 765 Ferst Drive, Atlanta, GA 30332-0205, USA

[b]Department of Industrial Engineering, Seoul National University, Seoul, South Korea

[c]School of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea

Abstract

A generic event control framework is presented for a class of modular flexible manufacturing systems (FMSs). A modular FMS is characterized as a set of flexible workstations inter-connected by a material handling system consisting of a transporter and a stocker. Many of the existing FMS implementations in industry fall into this class. Furthermore, the modularity also allows the capability to effectively model a complex FMS by decomposing it into several modular FMSs. In order to achieve reconfigurability of control system for various FMS implementations and control policies, the proposed control framework is defined as a set of distributed resource controllers and a central system supervisor coordinating them. The resource controllers are further classified into workstation, transporter, and stocker controllers. As the controllers exchange a series of events according to pre-defined protocols, they are modelled as event handlers in which control actions are made based on the event occurrences. Specifically, for each controller, an event-based control structure specified in terms of generic logical and performance control functions, is presented. Since the proposed framework is defined by the use of the interacting distributed processes with the well-defined protocols and computationally efficient algorithms, it is expected that the framework is easily implementable for most industrial FMSs. Finally, the performance and reconfigurability is demonstrated by the distributed simulation from which we can also verify the correctness of the proposed algorithms. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Flexible manufacturing system; Control algorithms; Event control

* Corresponding author. Tel.: +1-404-894-4202.
E-mail address: jonghun@isye.gatech.edu (J. Park).

PII: S0360-8352(00)00032-2

1. Introduction

Flexible manufacturing systems (FMSs) are promising means to meet the increasing needs for agility of modern manufacturing systems. Also, flexible automation is often an inevitable choice when the parts are too large or small to be handled by a human operator, requiring automation of material handling functions. In an FMS, finite resources such as buffers, tools, pallets, and material handling devices are shared by jobs, and therefore control system plays an important role as it is responsible for allocating the resources to concurrently competing jobs so that logical correctness and high performance can be achieved. However, the control system design for FMSs is a complex task since it requires a clear understanding of the underlying concurrent and asynchronous system dynamics. For this reason, many researchers have proposed various formal models to facilitate the development of control software. In Naylor and Volz (1987) give an excellent discussion about the formalism necessary to develop a generic control software, and present a conceptual modeling framework for integrated manufacturing system characterized as an assembly of software/hardware components. A series of rigorous research results for automating control software development process has been presented by a group of researchers in Joshi, Mettala and Wysk (1992), Smith and Joshi (1994), Joshi, Mettala, Smith and Wysk (1995) and Smith, Hoberecht and Joshi (1996). Specifically, Joshi et al. (1992) propose an approach to automate the development of control software by use of a context-free grammar. Smith and Joshi (1994) present a message-based part state graph (MPSG) to represent the execution module of shop floor controller as a communicating finite state machine. A graph-based model equivalent to pushdown automata is proposed as a formal model of flexible manufacturing cell in Joshi et al. (1995). Further approaches along this line of research can be found in Daltrini and Kumara (1996), where a Moore machine is extended to provide a generic representation of distributed and timed control environments. Other important formal models which have been widely accepted for FMS control are Petri nets and their variants. Ezpeleta and Colom (1997) present an automatic synthesis method based on a colored Petri net. Venkatesh and Zhou (1998) propose a combined approach using object modeling diagrams and Petri nets so that object-oriented modeling helps to support reusability and extensibility of the control software. Detailed descriptions on applying Petri nets to FMS control are given in Zhou and Venkatesh (1999) and Zhou and Dicesare (1993). Finally, supervisory control theory pioneered by Ramadge and Wonham (1989) has also been applied to synthesize the control software. In particular, an implementation methodology which utilizes the supervisory control theory in conjunction with programmable logic controller, is presented in Lauzon, Mills and Benhabib (1997), and a method to rapidly prototype the control software using the supervisory control theory is developed by Qiu and Joshi (1996).

While the approaches based on the above mentioned formal models are usually generic enough to be applied to any kind of FMS configurations, the complexity in modeling and computation has been recognized as the main barrier for their successful implementations in the large-scale, real-world FMSs. Hence, motivated by the need for readily implementable control systems for industrial, large-scale FMSs, we take a different approach in this paper. We confine our modeling domain to a class of FMSs which is characterized as *modular*, and propose an event control framework by specifying a set of easily implementable and