

Chapter 33

Modeling and Simulation of Discrete Event Robotic Systems Using Extended Petri Nets

Gen'ichi Yasuda

Nagasaki Institute of Applied Science, Japan

ABSTRACT

This chapter deals with modeling, simulation, and implementation problems encountered in robotic manufacturing control systems. Extended Petri nets are adopted as a prototyping tool for expressing real-time control of robotic systems and a systematic method based on hierarchical Petri nets is described for their direct implementation. A coordination mechanism is introduced to coordinate the event activities of the distributed machine controllers through friability tests of shared global transitions. The proposed prototyping method allows a direct coding of the inter-task cooperation by robots and intelligent machines from the conceptual Petri net specification, so that it increases the traceability and the understanding of the control flow of a parallel application specified by a net model. This approach can be integrated with off-the-shelf real-time executives. Control software using multithreaded programming is demonstrated to show the effectiveness of the proposed method.

1. INTRODUCTION

Complex robotic systems such as flexible manufacturing systems require sophisticated distributed real-time control systems. A major problem concerns the definition of the user tasks and the cooperation between the subsystems,

especially since the intelligence is distributed at a low level (machine level). Controlling such systems generally requires a hierarchy of control units corresponding to several abstraction levels. At the bottom of the hierarchy, i.e. machine control level, are the programmable logic controllers (PLC). The next level performs coordination of the PLCs. The third level implements scheduling, that is, the real-time assignments of workpieces

DOI: 10.4018/978-1-4666-1945-6.ch033

and tools to machines. At the machine level, PLCs perform local logical control operations of flexible, modular, high-speed machines through the use of multiple independent drives (Holding, et al. 1992). Implementation languages can be based on ladder diagrams or more recently state machines (Silva, et al. 1982). However, when the local control is of greater complexity, the above kinds of languages may not be well adapted. The development of industrial techniques makes a sequential control system for robotic manufacturing systems larger and more complicated one, in which some subsystems operate concurrently and cooperatively (Neumann, 2007).

In the area of real-time control of discrete event robotic systems the main problems that the system designer has to deal with are concurrency, synchronization, and resource sharing problems. Presently, the implementation of such control systems makes a large use of microcomputers. Real-time executives are available with complete sets of synchronization and communication primitives. However, coding the specifications is a hazardous work and debugging the implementation is particularly difficult when the concurrency is important. It is important to have a formal tool powerful enough and allowing developing validation procedures before implementation. Conventional specification languages do not allow an analytical validation. Consequently, the only way to validate is via simulation and step by step debugging. On the other hand, a specification method based on a mathematical tool may be more restricting, but analytical procedures can strongly reduce the simulation step. Rapid prototyping is an economical and crucial way to experiment, debug, and improve specifications of parallel applications. Increasing complexity of synchronizing mechanisms involved in concurrent system design makes necessary a prototyping step starting from a formal, already verified model. Petri nets allow to validate and to evaluate a model before its implementation. The formalism allowing a validation of the main properties of the

Petri net control structure (liveness, boundedness, etc.) guarantees that the control system will not fall immediately in a deadlocked situation. In the field of flexible manufacturing cells, the last aspect is essential because the sequences of control are complex and change very often.

When using Petri nets, events are associated with transitions. Activities are associated to the firing of transitions and to the marking of places which represents the state of the system. The network model can describe the execution order of sequential and parallel tasks directly without ambiguity (Silva, 1990). Pure synchronization between tasks, choices between alternatives and rendezvous can be naturally represented. Moreover at the machine level Petri nets have been successfully used to represent the sequencing of elementary operations (Yasuda, et al. 1992). In addition to its graphic representation differentiating events and states, Petri nets allow the possibility of progressive modeling by using stepwise refinements or modular composition. Libraries of well-tested subnets allow components reusability leading to significant reductions in the modeling effort. The possibility of progressive modeling is absolutely necessary for large and complex systems, because the refinement mechanism allows the building of hierarchically structured net models. Furthermore, a real-time implementation of the Petri net specification by software called a token player can avoid implementation errors. This is because the specification is directly executed by the token player and the implementation of these control sequences preserves the properties of the model. In this approach, the Petri net model is stored in a database and the token player updates the state of the database according to the operation rules of the model. For control purposes, this solution is very well suited to the need of flexibility, because, when the control sequences change, only the database needs to be changed.

Some techniques derived from Petri nets have been successfully introduced as an effective tool for describing control specifications and real-

15 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/modeling-simulation-discrete-event-robotic/69304

Related Content

Integration of CAD/CAPP/CAM/CNC

Xun Xu (2009). *Integrating Advanced Computer-Aided Design, Manufacturing, and Numerical Control: Principles and Implementations* (pp. 231-245).

www.irma-international.org/chapter/integration-cad-capp-cam-cnc/8484

Design and Development of Hybrid Stir Casting Process

Abhishek Kamboj, Sudhir Kumar and Hari Singh (2012). *International Journal of Applied Industrial Engineering* (pp. 1-6).

www.irma-international.org/article/design-and-development-of-hybrid-stir-casting-process/93011

Two-Decision-Maker Conflict Resolution with Fuzzy Preferences

Mubarak S. Al-Mutairi (2014). *International Journal of Applied Industrial Engineering* (pp. 40-59).

www.irma-international.org/article/two-decision-maker-conflict-resolution-with-fuzzy-preferences/138308

Industry 4.0 in Cultural Industry: A Review on Digital Visualization for VR and AR Applications

Antonios Kargas and Dimitrios Varoutas (2021). *Research Anthology on Cross-Industry Challenges of Industry 4.0* (pp. 1379-1396).

www.irma-international.org/chapter/industry-40-in-cultural-industry/276880

Operational Production Structures Used in the Multi-Serving System

I. C. Dima and Jozef Novac-Marcincin (2013). *Industrial Production Management in Flexible Manufacturing Systems* (pp. 398-422).

www.irma-international.org/chapter/operational-production-structures-used-multi/73734