Chapter 15
Distributed Coordination Architecture for Cooperative Task Planning and Execution of Intelligent Multi-Robot Systems

Gen’ichi Yasuda
Nagasaki Institute of Applied Science, Japan

ABSTRACT
This chapter provides a practical and intuitive way of cooperative task planning and execution for complex robotic systems using multiple robots in automated manufacturing applications. In large-scale complex robotic systems, because individual robots can autonomously execute their tasks, robotic activities are viewed as discrete event-driven asynchronous, concurrent processes. Further, since robotic activities are hierarchically defined, place/transition Petri nets can be properly used as specification tools on different levels of control abstraction. Net models representing inter-robot cooperation with synchronized interaction are presented to achieve distributed autonomous coordinated activities. An implementation of control software on hierarchical and distributed architecture is presented in an example multi-robot cell, where the higher level controller executes an activity-based global net model of task plan representing cooperative behaviors performed by the robots, and the parallel activities of the associated robots are synchronized without the coordinator through the transmission of requests and the reception of status.

INTRODUCTION
Nowadays, advanced manufacturing systems are generally composed of the following control levels: planning, scheduling, coordination, and low-level control. At the planning level, based on definition of tasks and due dates, resource requirements are specified and an actual time period and resources are assigned to each task and subtask. The monitoring of real time events and modification and adaptation of the cell configuration after detection of an equipment failure are supported by a knowledge-based system. The coordination level is naturally decomposed into two sublevels: the flow of parts through
the buffers and the coordination of a manufacturing operation. The former can be viewed as producer/consumer processes. The flow control and robot synchronization are performed by a multitasking operating system. The coordination level receives a rather static schedule from the upper level and is provided with dynamic information about particular states of a system from the lower level. It sends commands to low-level controllers and may ask for a rescheduling. Thus, the coordinated system behaves concurrently and includes cooperation and competition relations among its subsystems. This concurrency with links to both upper and lower levels should be modeled formally in the control system design. Currently, control systems for the coordination of specific low-level controllers according to the schedule of operations have been mainly developed in an ad hoc manner to fit just one particular system, which requires many man-years of hard programming again and again to build the control systems because specifications are not sufficiently complete and the system environment changes frequently.

The development of industrial techniques makes sequential operation control for robotic systems more large-scale and complicated one, in which some robots or subsystems operate concurrently and synchronously based on sensing information by external sensors and/or inter-robot communications as cooperative multi-robot systems (Yasuda et al., 1991). Multiple robots have the possibility to perform complex and large-scale tasks more efficiently by cooperating in some way than a single robot does. Although there are many tasks that could be constructed for a single complex robot, in many cases there are advantages to using multiple robots, such as robustness and less complexity. However, to perform complex robotic tasks such as cooperative fixing and mating of separate parts by two or more robots, new distributed autonomous control architecture through the integration and cooperation for intelligent multi-robot tasks should be developed (Hörmann, 1991). In such intelligent robotic tasks, synchronization between associated robots is crucial, and cooperation based on ad hoc or temporary master-slave or leader-follower relationships or ideal cooperation through mutual coordination with synchronous communication based on simple reflexive behavioral control on a decentralized control architecture is required (Duffee, 1996). By specifying a conditional sequence of actions and goals or joint plan involving the subject and others, interlocking coordination where each robot adjusts its action selection based on the evolution of the ongoing interaction is achieved. In the distributed planning, the cooperative interaction consists of a series of actions including communication acts such that a robot causes a change in the environment that triggers the precondition of the next action of a sequence.

Besides the algorithm in use, the performance of cooperative tasks depends on the actual implementation. Non-linear systems control theory does not provide tools to analyze and synthesize such complex tasks with respect to output control performance. On the other hand, research on real-time operating systems does not deal with the impact of the organization of the controller on the controlled process, except scheduling, fault tolerance, or liveness. In this context, in order to realize cooperative multi-robot systems, it is necessary to provide effective tools for describing process specifications and control algorithms and developing validation procedures before implementation in a clear and consistent manner. Among the existing modeling formalisms, Petri nets provide a powerful framework to model and analyze robotic manufacturing systems, characterized by their ability to naturally represent the synchronization of processes, the concurrence of activities, the presence of conflicts, causality, resource sharing, mutual exclusion, etc. (Hasegawa, 1996), which are inherent characteristics of multi-robot systems as well as manufacturing systems. Since one can directly convert Petri net definitions into control code for supervisory control of robotic systems after correct Petri net models are constructed, the systematic modeling of robotic systems becomes very important (Caccia, 2005). In this chapter, based on the hierarchical and distributed structure of the manufacturing system, a structural design methodology through a top-down