SHOP-FLOOR SCHEDULING AND CONTROL:
A SYSTEMS APPROACH

Richard A. Wysk, Richard J. Mayer and Hyuenbo Cho
Texas A&M University

Sanjay Joshi, Jeffrey Smith and Walter Hoberecht
The Pennsylvania State University

C. Dennis Pegden
Systems Modeling Corporation

and

Albert L. Jones
National Institute for Standards and Technology

Abstract

Shop-floor control of complex manufacturing systems continues to be a major obstacle in CIM implementation. The complexity of the problem is seen in the detailed scheduling requirements of product through the manufacturing resources, the allocation of tooling resources within the system, the communication requirements of the system, and the data base maintenance and access requirements of the system. In order to effectively control these activities, it is necessary to define an architecture and functional perspective of how a shop-floor control system operates.

In this paper, we define the architectural requirements and an overall control schema that we feel fits the complexity of the problem. The architectural elements include definitions of the following architectures: functional, control, data, computer, communication and factory. The definitions and interactions of these architectures along with a view for automated shop-floor control is discussed. An architecture for shop-floor control along with the application environment to implement such a system is also presented.

BACKGROUND
Integration is one of the most important problems facing U.S. manufacturers today. While it is still

G. Fandel et al. (eds.), Operations Research in Production Planning and Control
© Springer-Verlag Berlin-Heidelberg 1993
believed that Computer Integrated Manufacturing (CIM) can have a major positive impact on U.S. productivity, it is far from commonplace in our factories. A major problem is the difficulty involved in integrating products (software and hardware) together with information (internal and external) into a smooth running system. The reason is that these products are typically supplied by a variety of vendors who do not provide the necessary "integration hooks". That is, the products from one vendor are not meant to be (and in some cases are impossible to be) integrated with products from a different vendor.

The hooks referred to here fall into two categories: functional and control. Consider the example of computer aided process planning (CAPP) and scheduling. Clearly, integration of these is essential in every factory. There are, in fact, numerous products available today which claim to perform these two functions. However, no two CAPP systems perform exactly the same functions, and the situation is worse for scheduling systems. Furthermore, the input/output and control requirements can vary dramatically from one product to another. This includes software (content and format), hardware (communication protocols and capabilities), and database content and representation. Consequently, it is almost impossible to integrate an arbitrary CAPP system with an arbitrary scheduling system. The same situation exists for almost all manufacturing functions. To make matters worse, each user invariably requires some special feature or option. This increases development and maintenance costs and exacerbates the integration problem.

The underlying integration problem is both architectural and technological in nature. Without a detailed architecture, vendors do not know what functionality and interfaces to provide in their products and what technologies to use to build those products. Consequently, they provide a range of both. In terms of the earlier example, this means that vendors provide several different products all claiming to do process planning. In reality, these products can have vastly different capabilities using a wide variety of technologies at varying costs.

Without an architecture, users do not know what to expect from a given product. They are left to find the one that meets their own definitions, objectives, and costs. As for integration, they typically have four choices: go where they went before; pay handsomely for specific integration tools; do it themselves; or ignore it. Invariably, they do not have the resources for option number three. Consequently, closed-systems and islands-of-automation continue to exist with neither vendors' incentive nor users' where-with-all to change our current dilemma.

METHODOLOGY
The control of CIM systems continues to be a difficult applications problem for both industrial and computer engineers. The complexity of a manufacturing system continues to perplex researchers