Multiagent Framework for Transitional Operations

Nenad Ivezic, Tom Potok, and Line Pouchard
Oak Ridge National Laboratory

Lean principles in manufacturing support cost-efficient, customer-driven management of operations.* For example, while traditional approaches to manufacturing management rely on centrally scheduled production or "push" systems, lean practices suggest customer-driven task scheduling or "pull" systems and globally synchronized scheduling or "takt" systems that minimize costs of inventory and work-in-progress.

Transitioning from traditional to lean manufacturing, however, is a complex, enterprise-wide process, requiring major investment and business process change. Understanding the scope and impacts of these changes is critical, and requires multidisciplinary expertise to investigate the current cost and state of manufacturing processes and to compare these values to the desired lean operation. Compounding the problem is the effect this transition has on the external supply chain, typically not controlled by the core company.

We have developed the Manufacturing Agent-Based Emulation System as an open framework for design and analysis of discrete manufacturing systems . MABES currently supports the transition from traditional to lean manufacturing in two major functions:

- Analysis of alternative agent-based scheduling and control approaches that can be implemented across the extended enterprise.
- Real-time collaboration of design teams during manufacturing line design and analysis stages.

*For more information on lean manufacturing principles, see the Lean Aerospace Initiative, available online at http://lean.mit.edu/.

MABES bases its support for these functions on on two system paradigms: distributed agents and synchronous collaboration.

Distributed Agents

The MABES scheduling and manufacturing control structure is a distributed autonomous agent framework. Each agent is responsible for monitoring and acting on a component of the manufacturing process. A component may be a process, such as a simple drill press; a process center, such as a collection of welding robots; or a stack, such as an inventory of preprocessed components. The agents interact to control the flow of parts through either a traditional push or a lean pull or takt system. Within the adopted approach, the overall desired behavior for a manufacturing line emerges from individual behaviors of, and interactions among, distributed agents.

We have identified several agent types and coordination protocols from analysis of traditional and advanced scheduling approaches. Figure 1 illustrates the types and their interactions. It shows a part being "pulled" through three process centers by four types of agents: customer, stack, process center, and process agents. (The same agent types appear in other approaches, such as push and takt.) These agents are responsible for components of the manufacturing process; they do not overlap in their respective competencies and responsibilities, and together they cover all parts of a process line.

In the case of a manufacturing line implementing a pull approach, as shown in Figure 1, the agent interactions are initiated when the customer agent sends requests, or pull signals, for products to the output stack agent at scheduled times. The receiving agent propagates the signal upstream to the other agents in the process line with the goal to "pull" the resources required for completing the product.

The agents communicate through messages, taking into account their local context. The context includes

- commitments to deliver resources to requesting downstream agents or to complete processes that would generate requested products, and
- knowledge of expected product completions.

Agent behavior patterns, encoded as separate rule-based systems, also take the current context into account, and an agent performs an action either on its respective component within the manufacturing line or in further communication with other agents.

This distributed approach to scheduling and execution control is both robust and easy to implement across an extended enterprise because it relies only on local rules of behavior defined for each agent type and on an agent-to-agent coordination protocol. The modular nature of this approach allows anomalies in the manufacturing line, such as machine failure or supplier tardiness, to be localized and dealt with by applying only the local rules of agent behavior and coordination. In this way, the global integrity of the manufacturing line is more likely to be preserved than in the case of traditional, centralized scheduling. The modularity also allows simple replication of the agent scheduling and control mechanism throughout the internal and external supply chains.

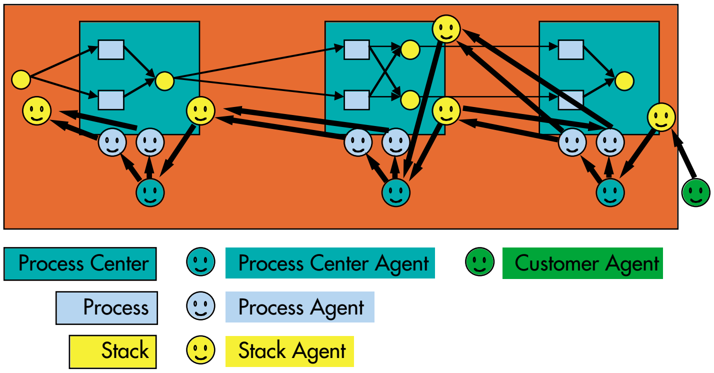

Figure 1. Four agent types and their interactions in "pulling" a part through three process centers and various numbers of stacks.

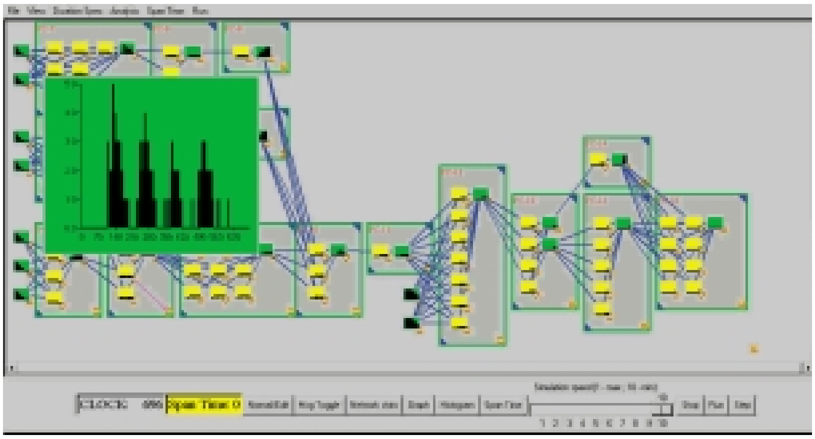

Figure 2. The Manufacturing Agent-Based Emulation System collaborative interface for analyzing performance of alternative scheduling strategies.

Synchronous Collaboration

To support collaborative work of manufacturing teams, we have evolved the initial single-user interface to MABES into a synchronous, multiuser interface. We adapted a synchronous collaborative framework to support real-time interactions among multidisciplinary teams. A collaborative team can synthesize a model of a manufacturing line in real time, defining an agent-based scheduling model for the manufacturing process, specifying a schedule, simulating the model, and animating the result of the simulation. In this way, MABES greatly improves the time-consuming, error-prone information exchange and negotiation of collaborative decision-making.

Figure 2 shows the MABES collaborative interface to the multiagent design and analysis tool. The tool can animate message passing among agents and display manufacturing-process statistics for several metrics, including span-time and work-in-progress. We used the MABES tool to analyze the performance of alternative scheduling strategies such as push, pull, and takt. We also analyzed special events, such as production disruptions and bottlenecks, in the context of these alternative strategies.

Implementation

In the initial MABES prototype, graphical interfaces were implemented in Tcl/Tk language, while the manufacturing and agent models and simulation engine were implemented using CLIPS language. To extend the user interface and to support synchronous collaborative work, we employed the TeamWave Workplace Software Development Kit, available from TeamWave Software. ■

ACKNOWLEDGMENTS

Our work is supported by Oak Ridge National Laboratory, managed by the Lockheed-Martin Energy Research Corp. for the U.S. Department of Energy. This article was written under contract no. DE-AC05-96OR22464.

We would like to thank Bob Boykin, John Krajca, Tommy Lee, and Mark Taylor of Lockheed Martin Tactical Aircraft Systems for their support and expertise in the MABES project.

Nenad Ivezic is a guest researcher at National Institute of Standards and Technology, on leave from Oak Ridge National Laboratory. He received his MS and PhD from Carnegie Mellon University, specializing in computer-aided engineering.

Thomas E. Potok is a research staff member in the Collaborative Technology Research Center at Oak Ridge National Laboratory. He received his BS in computer science and MS and PhD in computer engineering from North Carolina State University.

Line Pouchard is a post-doctoral research scientist at the Collaborative Technologies Research Center. She received her MS in information sciences from Univ. of Tennessee, Knoxville, and a PhD from City University of New York.

Readers may contact the authors at {ivezicn, potokte, pouchardlc@ornl.gov.