Combining Operations Research and Agent-Oriented Techniques for Agile Manufacturing System Design

Piotr J. Gmytrasiewicz*, **H. H. Huang**† **and Frank L. Lewis**†
*Department of Computer Science and Engineering
University of Texas, Arlington, TX 76019-0015
†Automation and Robotics Research Institute
University of Texas at Arlington
7300 Jack Newell Blvd. S, Ft. Worth, TX 76118-7115
piotr@cse.uta.edu, hhuang@arri.uta.edu, flewis@arri.uta.edu

Introduction

Our research is centered on the technology of autonomous manufacturing units operating in high-mix low-volume agile manufacturing facilities. The main characteristic of such domains is their continuous and unanticipated change. We believe it is useful to view separate autonomous manufacturing units in such facilities as agents that have to cooperate during the production process, and be able to quickly adjust in response to incoming orders, new unexpected conditions, and technological changes. Ideally, even when some units break down and are carted away for repairs, the others should continue to function, reconfiguring as needed to make up for the lost capacity and missing functionality.

Our approach combines an agent-oriented paradigm that has recently emerged in AI research (see (Shoham 1993) for an example of such approach), with OR techniques that allow quantification of expected benefits of production-related decisions together with the uncertainties involved (Ballot 1986). We use decision theory as a single paradigm for directing an agent's behavior, and apply it to decisions on coordinated physical actions and on communication. To make rational decisions an agent has to represent the relevant information it has about its decision-making situation, including the available alternatives for action, possible actions of the other agents, and the expected utilities of their joint actions. To best coordinate its actions with the actions of the other agents, the agent has to predict what they are likely to do. This could be done by modeling the decision-making situation the other agents are in to determine what action is rational from their perspective. The models of the other agents can further include models the other agents have of others, and so on. Our Recursive Modeling Method (RMM) (Gmytrasiewicz, Durfee, & Wehe 1991a; 1991b) represents the hierarchy of the nested models explicitly, and includes the solution method allowing the agent to process all of the relevant knowledge it has to arrive at the optimal choice of action.

The basic building block of the RMM's representation of an agent's information is the payoff matrix. In a manufacturing context the payoff matrices have been extensively used by OR experts as a convenient and expressive tool to investigate decision-making under uncertainty. In this paper we show how the matrices depicting the decision-making situations of manufacturing agents can be arrived at. As an example we consider a reentrant flow system characteristic for semiconductor manufacturing plants.

Decision-Making in a Reentrant Flow System

Reentrant flow setup is characteristic for semiconductor warefs fabrication facilities (Kumar 1994). A simple example of such a system is depicted in Figure 1. It consists of two manufacturing robots, $R1$ and $R2$, producing three kinds of parts, P1, P2, and P3. The parts visit the machines as shown by the path lines for each part type, possibly revisiting the same machines repeatedly as required by the process.

We view each machine as an autonomous manufacturing agent that has to make a decision as to what to do next. Robot $R1$ has to decide from which buffer to take a part that is to be processed next. Thus there are five alternatives, corresponding to its choosing a part form buffer b_1^1, b_2^1, and so on. We will label these alternatives as a_1^1 through a_5^1, respectively, and additionally include an option of $R1$'s doing nothing at all, labeled a_6^1. The robot $R2$ will similarly have to choose, and its alternatives will be labeled a_1^2 through a_4^2, and a_5^2 for doing nothing. To represent a decision-making situation of each of the manufacturing agents we will assemble the agents' alternatives in payoff matrices. For the robot $R1$ the matrix is 6×5 and has the form depicted in Figure 2.

The entries of the payoff matrix in Figure 2 are the payoffs to $R1$ resulting from the joint actions of the agents. For example, an entry $u_{2,3}$ in this matrix shows

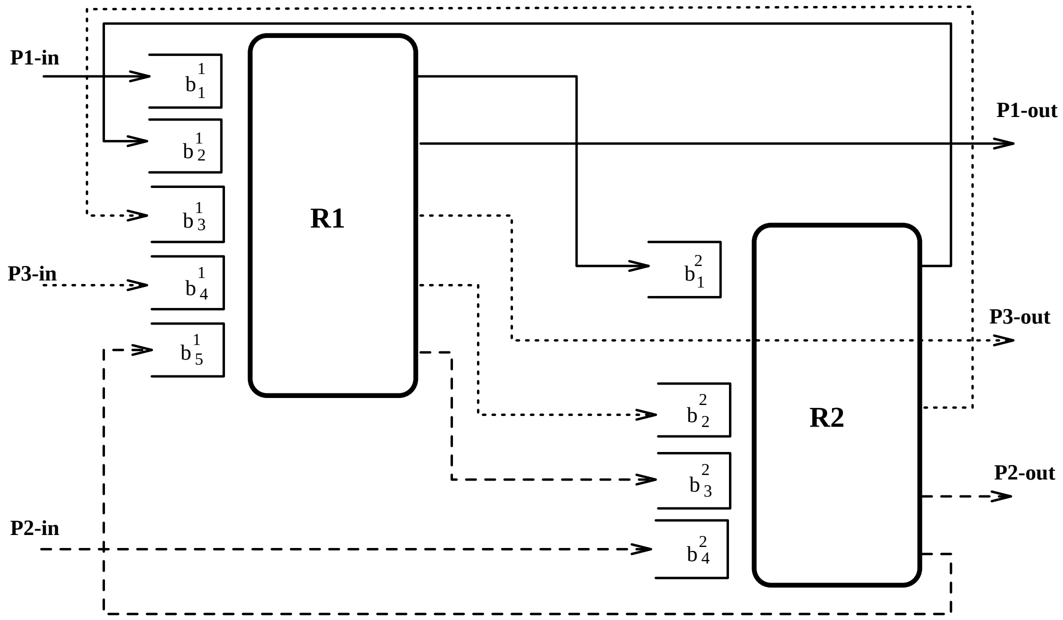

Figure 1: A Simple Reentrant Flow System with Two Machines

the payoff $R1$ expects to obtain as a result of its executing its action a_2^1 and $R2$'s executing a_3^2. The agents, as rational, are driven to maximize their expected payoffs by choosing their manufacturing actions appropriately.

The role of the designer of the system, therefore, is to tie the agents' payoffs to the overall goal of the production facility. There are multiple factors that could be included in the payoffs. For example, a robot should not attempt to reach for a part in an empty buffer, and it should process the parts so as to avoid any of its buffers from overflowing. Also, it may be desirable for the agent's payoff to depend on the value the particular operation contributes to the overall value of the finished product, as well as on the fact that it pays to produce parts that are currently on order, since there is no need for their prolonged storage. Further, we may wish the agents to consider how far along the parts are in the production process, say, in an attempt to give priority to components nearing completion. We now illustrate the ways in which these factors could be included in the individual robots' preferences.

Let us denote as L_k the current length, i.e., the number of parts, in the buffer b_k^1 of robot $R1$. If the capacity of the buffer is B_k, then the space still remaining in the buffer, S_k, can be computed as $S_k = B_k - L_k$. Also, let us say that the part from $R1$'s buffer b_k^1 would have be delivered to a buffer b_m^2 of $R2$ (or any other manufacturing agent including $R1$, if this part needs further processing at all) and the space remaining in this buffer would then be $S_m = B_m - L_m$.

The value added to the product at each production stage can be included by defining the work-in-progress value added to a part currently in the buffer b_k^1 as:

$$\Delta V_k^1 = \frac{V_k^{fin} - V_k^0}{N}, \tag{1}$$

where $V_k^{fin} - V_k^0$ is the difference between the final value of the part and the value of the part before the production process, and N is the number of the operations in the process.[1] The value added by the agent $R2$, ΔV_m^2 can be defined analogously.

Now, the payoff resulting from robot $R1$'s choosing to process the part in its buffer b_k^1 (that is, its choosing the action a_k^1), and from $R2$'s choosing its option a_n^2 may be postulated to have the following form:

$$u_{k,n} = (\Delta V_k^1 + \Delta V_n^2) L_k S_m^* L_n. \tag{2}$$

where S_m^* is equal to S_m if the part is in the need of further processing, and it is made equal to the average of the spaces remaining on all of the buffers of the machine $R2$ otherwise. The above formulation causes the agents never to process a part to be passed to a nighbour's buffer that is already full, thus implementing the Kanban lookahead philosophy (Pastravanu, Gurel, & Lewis 1994), and gives preference to parts in agent's buffers that are filling up. All other factors being equal, the agents will give preference to processing more valuable parts first.

The calculation of payoffs corresponding to either of the agents choosing to do nothing proceeds by setting

[1]Here, we have assumed for simplicity that each operation in the process contributes the same to the final value.

$$\begin{array}{ccc} & & \textbf{R2} \\ & & \begin{array}{ccccc} a^2_1 & a^2_2 & a^2_3 & a^2_4 & a^2_5 \end{array} \\ \textbf{R1} & \begin{array}{c} a^1_1 \\ a^1_2 \\ a^1_3 \\ a^1_4 \\ a^1_5 \\ a^1_6 \end{array} & \left[\begin{array}{ccccc} & & & & \\ & & & & \\ & & u_{k,n} & & \\ & & & & \\ & & & & \\ & & & & \end{array} \right] \end{array}$$

Figure 2: Form of the Payoff Matrix Representing $R1$'s Decision-Making Situation

the corresponding value added factors to zero, and by setting the corresponding buffer contributions to one. Thus, the payoff corresponding to both robots choosing to do nothing, $u_{6,5}$ is zero. The payoffs corresponding to $R2$ doing nothing and $R1$ processing the part from its buffer b^1_k are:

$$u_{k,5} = \Delta V^1_k L_k S^*_m. \tag{3}$$

Finally, the payoffs corresponding to $R1$ doing nothing and $R2$ processing the part from its buffer b^2_n are:

$$u_{6,n} = \Delta V^2_n L_n. \tag{4}$$

Is is not difficult to include other factors in the agents' payoffs. For example, storage costs, C^{Stor}_k, may be included by redefining ΔV^1_k as:

$$\Delta V^1_k = \frac{V^{fin}_k - V^0_k - C^{Stor}_k}{N_k}. \tag{5}$$

This has the effect of favoring products currently on order, since the storage costs for such parts is substantially reduced. An effect of clearing the work-in-progress in the facility and favoring the parts that near the end of their production cycle may be achieved by changing the definition of the payoff $u_{k,m}$ as:

$$u_{k,n} = (V^1_k + V^2_n) L_k S^*_m L_n. \tag{6}$$

where V^1_k and V^2_n are made equal to the value of the part on the current stage of processing:

$$V^1_k = \Delta V^1_k * |k|, V^2_n = \Delta V^2_n * |n|, \tag{7}$$

and where $|k|$ and $|n|$ are the numbers of the current stage of processing for the part in $R1$'s b^1_k buffer, and the part in $R2$'s b^2_n buffer, respectively.[2] With this formulation the payoff reflects the current value of the partially processed product, and the parts further along in the process are favored, which implements the known concept of "Last Buffer First Serve" (LBFS) (Kumar 1994).

As we mentioned, different combinations of the above factors constitute various ways to tie the preferences of each individual production agent to the overall objective of the production facility in terms of effectiveness and throughput, minimal cycle time and its variance, and ultimately profit. During our proposed research we will experiment with other methods, incorporating factors describing the production ability (tools available), production capacity (throughput) and others.

In the next section we show how the payoff matrices derived above can be used by the autonomous agents to coordinate their actions and to intelligently communicate in a dynamic environment.

Rational Coordination and Communication

The paradigm we have adopted in designing the Recursive Modeling Method (RMM) is that of decision-theoretic rationality, defined as a decision-making behavior that maximizes the expected utility of the agent's action, given the agents' preferences, its abilities for action, and all of the relevant information the agent has about the environment (Gmytrasiewicz, Durfee, & Wehe 1991a; 1991b; Gmytrasiewicz & Durfee 1995).

The main goal of our framework, therefore, is to represent and reason with all of the relevant information that an agent may have about the environment, itself, and other agents, in order to estimate expected utilities of its alternative courses of action, and thus to make the rational decision in a multiagent situation. Our

[2]For example, $R2$'s buffer b^2_2 holds parts P3 on the second stage of their three-stage production process.

method lets agents explicitly reason about how the actions of other agents affect the utilities of their own actions. Thus, to choose an action that maximizes its individual utility, an agent should model others to predict their actions. The fact that the other agents could take a similar approach gives rise to the recursive nesting of models. RMM represents this recursion explicitly to allow an agent to arrive at a rational decision in the multiagent environment, and it contains all of the relevant information an agent might have about the decision-making situations it, and other agents, find themselves in. The possible uncertainties are expressed as probability distributions, reflecting uncertainty as to the other agents' intentions, abilities, long-term goals, and sensing capabilities. Furthermore, on a deeper level of the recursion, the agents may have information on how other agents are likely to view them, how they themselves think they might be viewed, and so on.

A decision-making situation in RMM is represented as a payoff matrix, and the whole state of an agent's knowledge has a form of a nested hierarchy of payoff matrices, called a recursive model structure. A simple example, from the point of view of the agent $R1$, is depicted in Figure 3. Here, the matrices were simplified to make the solution process easy to follow.

The top matrix in Figure 3 is the one representing the decision-making situation of the agent $R1$. $R1$ has three alternative courses of action and interacts with agent $R2$, also with three alternatives. The payoffs in the matrix represent the utilities $R1$ expects as a result of the corresponding joint actions of both agents, and for a manufacturing scenario could be derived as discussed in the previous section. In order to make a rational decision $R1$ should compute the expected utilities of its alternatives and choose one with the highest value. To do so, $R1$ needs to predict the actions of $R2$. In this example $R1$ can do it by looking at the decision-making situation of the agent $R2$ to find out what rationality dictates $R2$. In this case, however, $R1$ faces some uncertainty as to what situation $R2$ is facing. This may be due to $R1$ not knowing the abilities of $R2$, or not knowing how $R2$ sees the environment. This uncertainty is represented as two alternative models of $R2$'s decision-making situation on the second level, together with probabilities assigned to each model. In one model one of the actions, a_2^2, is not a viable alternative for $R2$ (as well as, from $R2$'s point of view, for $R1$) and the matrix is 2×2.

In the simple example presented $R1$ was able to form two alternatives describing $R2$'s decision-making situation, but we assumed that $R1$ has no information beyond that. That is, $R1$ has no idea how $R2$ could possibly model $R1$ in making its choice. This lack of information is represented by two uniform probability distributions depicted on the third level of the recursive model structure in Figure 3. Uniform distributions, according to the principle of entropy maximization (Neapolitan 1990), precisely represent the lack of information in this case, since their informational content is zero.[3] We refer to the uniform distributions used in this way as no-information models.[4]

Our approach to finding a solution of the above structure, i.e., to arrive at the rational choice of R_1's action, is to process the information bottom-up, from the leaves of the recursive model structure toward the top matrix of the agent. Thus, on the second level of the model structure, in the model of R_2 corresponding to it not considering the alternative a_2^2, the expected utilities of R_2's actions are 0 and 1 ($= 2\times 0.5 + 0 \times 0.5$), for a_1^2 and a_3^2, respectively. This indicates that R_2 would choose a_3^2 in this case. In the other model of $R2$'s situation, on the other hand, the calculation shows that R_2's preferred action would be a_2^2. The conclusion from both of the models on the second level of modeling is, therefore, that R_2 would pursue a_3^2 or a_2^2, with the estimated probabilities of 0.9 and 0.1, respectively. This results in a probability distribution over R_2's actions a_1^2, a_2^2, and a_3^2 being $[0, 0.1, 0.9]$.

The above result can be propagated up and used to solve the matrix on the top of Figure 3. The expected utilities of R_1's actions can be computed straightforwardly as:

$$
\begin{aligned}
u^{R_1}_{a^1_1} &= 0 \times 1 + .1 \times 5 + .9 \times 1 = 1.4 \\
u^{R_1}_{a^1_2} &= 0 \times 4 + .1 \times 2 + .9 \times 2 = 2 \\
u^{R_1}_{a^1_3} &= 0 \times 2 + .1 \times 4 + .9 \times 0 = 0.4
\end{aligned}
$$

Thus, the result of processing the information available to $R1$ in this case is that the rational choice for $R1$ is the action a_2^1, and $R1$ should expect the payoff of 2 as the result of the interaction.

The bottom-up solution process we have illustrated assumes that the recursive model structure considered is finite and actually does bottom out. This assumption seems justified in practical cases since it is difficult to imagine that agents will have information about each other nested down to infinite levels. Practically

[3] Our representation here is a simplification of the problem of "only knowing". (Halpern 1993; Lakemeyer 1993) contain useful discussion and related references.

[4] Our usage of this kind of a model here is in contrast to traditional work in game theory, in which the models considered contain infinite recursion.

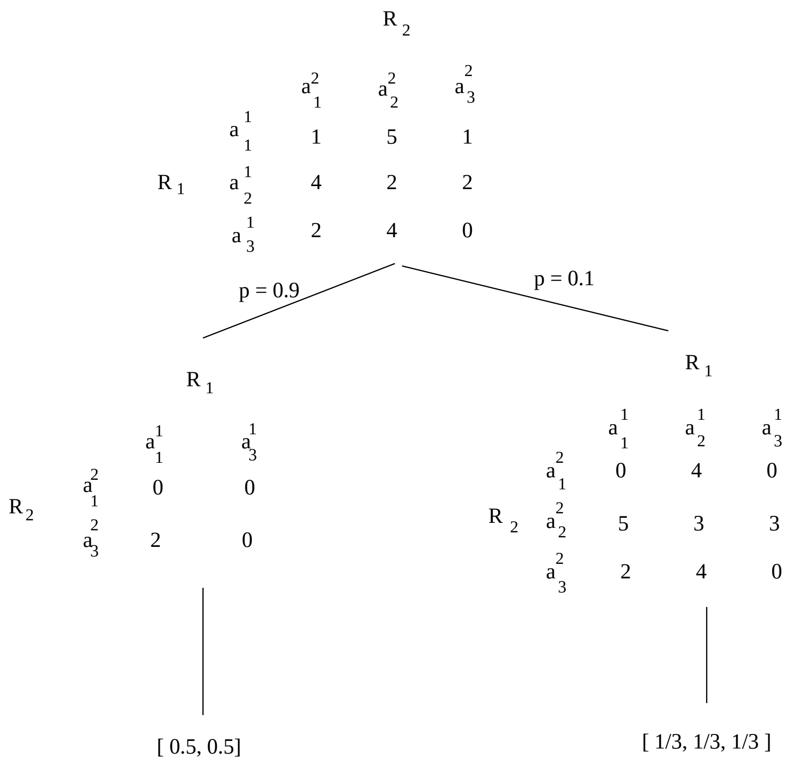

Figure 3: A Simple Recursive Model Structure Depicting R_1's Knowledge.

realizable agents seem bound to run out of knowledge at some level of nesting, at which point the no-information model is used to terminate the recursion.[5].

Using our preliminary implementation of RMM we were able to test the quality of the choices that the agents make. Our tests, mainly in the domains of cooperative reconnaissance and cooperative waste cleanup missions, show that RMM agents can successfully coordinate their actions if they have information about each other down to the third and fourth level of nesting. We have found that in such cases all of the decisions made by our agents are intuitively "correct" to human observers. This is what should be expected, we think, since in every day interactions people rarely model others to a level deeper than third or fourth.

Rational Communication The Recursive Modeling Method can also be used to achieve rational communicative behavior among the interacting agents. While communicating, the agents have to judiciously select the information they transmit to the other agents to make sure that, first, the others have the information needed for effective coordination, and, second, that they are not overwhelmed with unimportant information. The basic idea is that an agent should base its decision on what information to transmit on an estimate of how the message's impact on the sender's and receiver's state of knowledge will improve the expected outcome of their interaction. This expected improvement can be identified as the utility of the message itself. We treat decisions about communication just like decisions about any other actions, and thus employ decision-theoretic techniques to select the action with the highest expected utility.[6]

In the example scenario described we concluded that, given what R_1 knows about the environment and R_2, R_1's best alternative is to choose its action a_2^1, and expect the payoff of 2. Can R_1 improve its expected payoff and the quality of the interaction using communication? The answer is, in this case, yes.

Consider a message M_1 in which $R1$ announces that it will choose its action a_2^1 (by describing it using some common language, if such is available). If this message were sent and reached R_2, then R_1 would be sure that R_2 will know what action to anticipate from $R1$. Thus, the recursive model structure will change as depicted in Figure 4.

[5]We would like to stress that we see the use of no-information models as justified here *only* when the limitation of the agents' knowledge are reached. The no-information models should not be viewed as an *ad hoc* way to terminate the recursive structure on an arbitrary level of nesting. Thus, we assume that the agents have the necessary introspective capacities to recognise the limits of their knowledge.

[6]The notion of the utility of a message we consider here is orthogonal to the notion of the value of information considered in decision theory (Pearl 1988). The latter expresses the value of information to its recipient. We, on the other hand, consider the value of a message to its sender, since, of course, it is the sender that makes the decision of if, and what, to communicate.

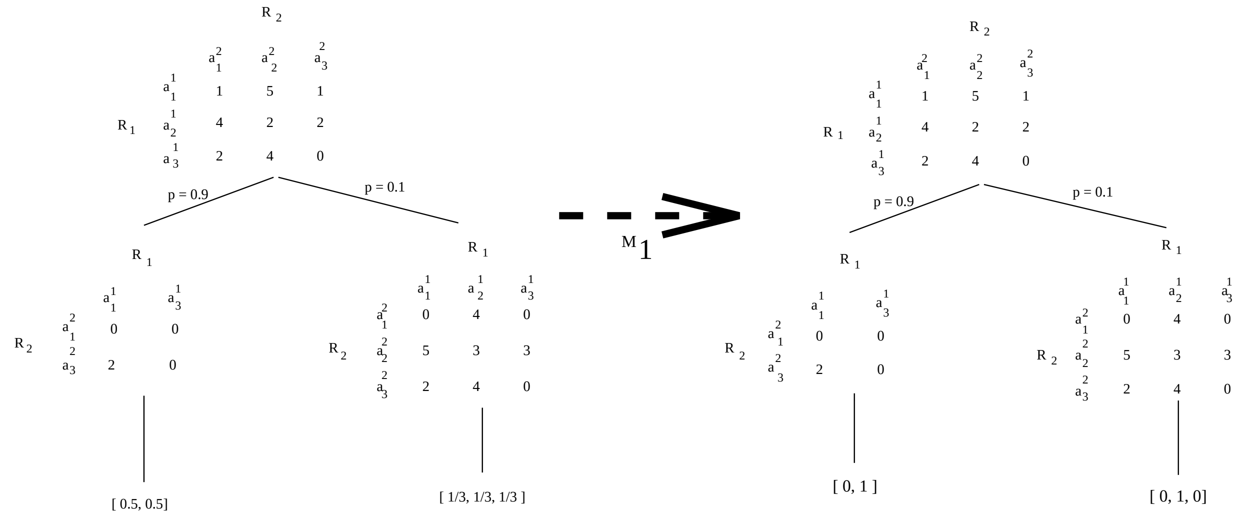

Figure 4: **Expected Change of R_1's Knowledge due to Message M_1.**

The new structure can be easily solved using the same solution method, but now $R1$ would conclude that $R2$ is equally likely to choose a_1^2 or a_3^2, and the utility of $R1$'s action a_2^1 is equal to 3. Thus, by sending the message M_1 to R_2, R_1 was able to increase the expected utility it gets from the interaction from 2 to 3. The utility of sending the message M_1 is, therefore, 1.

The message M_1 may be one of many worth considering in any situation. Other messages can be evaluated analogously, by estimating the change they are expected to bring up in the agents' states of knowledge and the resulting improvement in the quality of the interaction. By computing the expected utilities of messages, the agents can choose to send the message of the highest utility, given the circumstances and the state of agents' knowledge about each other, and thus exhibit rationality in their communicative behavior.

Our preliminary experiments with the above approach to communication are encouraging. In a number of scenarios we have considered our method selects as appropriate the message that humans are likely to favor in the same situations. Sometimes it is the message declaring the intended actions of the agents, as the message M_1 above, but sometimes it is a message in which agents tell the other agents about some features of the multi-agent environment the others have no access to (that the machine X is broken, for example). At still other times, the best communicative act is simply to ask a question.

Conclusion

We have presented an approach that combines the strengths of operations research methods used in decision-making for manufacturing with an agent-oriented paradigm emerging in artificial intelligence research. According to our approach, the autonomous manufacturing agents do not follow the protocol given to them before hand. Instead, they are equipped with their own ability to make rational decisions, based on the current local situation they find themselves in. This allows the whole system to exhibit a level of flexibility in reacting to changing circumstances that could not be achieved with protocols. We expect that no matter what the new circumstances bring, our agents will reconfigure and optimize the production process the best they possibly can, by choosing the optimal coordinated behavior and the most suitable communication patterns.

We feel that our techniques will advance the state of the art for designing autonomous manufacturing agents capable of coordinating actions and intelligently communicating with other agents in the dynamic environments of flexible manufacturing facilities.

Our work brings together diverse threads in the areas of operations research and decision-making in manufacturing (Ballot 1986), as well as in game theory (Aumann & Brandenburger 1994; Brandenburger 1992; Brandenburger & Dekel 1993; Harsanyi 1967; Mertens & Zamir 1985) and artificial intelligence (Horvitz, Cooper, & Heckerman 1989; Russell & Wefald 1991; Cohen & Levesque 1990; Fagin, Halpern, & Vardi 1991; Shoham 1993). The use of artificial intelligence and autonomous agents for distributed manufacturing systems has been advocated (see (Burke & Prosser 1994; Kusiak 1988; 1992; Parunak 1988; Yuta & Premvuti 1990) and references therein), and various methodologies for communication system design for manufacturing were investigated, for example in (Stuckey & Pennington 1988; Ishida *et al.* 1992).

In our future work we will explore alternatives to designing the payoff matrices for the production agents, beyond these presented in this paper. We would like to explicitly include the information about the changes in the agents' production capabilities, as opposed to the agents learning about them indirectly from the amount of parts that await processing in the neighbor's buffer. Furthermore, we will investigate the ways to integrate the inter-agent decision-making presented here with modern controller designs for the reentrant flow scheduling investigated in (Huang, Barzin, & Lewis 1994; Lewis, Huang, & Pastravanu 1994; Lewis, Pastravanu, & Huang 1993).

References

Aumann, R. J., and Brandenburger, A. 1994. Epistemic conditions for Nash equilibrium. Working paper, Harvard Business School, Harverd.

Ballot, M. 1986. *Decision-Making Models in Production and Operations Management.* Robert E. Krieger Publishing Company.

Brandenburger, A., and Dekel, E. 1993. Hierarchies of beliefs and common knowledge. *Journal of Economic Theory* 59:189–198.

Brandenburger, A. 1992. Knowledge and equilibrium in games. *Journal of Economic Perspectives* 6:83–101.

Burke, P., and Prosser, P. 1994. The distributed asynchronous scheduler. In Zweben, M., and Fox, M. S., eds., *Intelligent Scheduling.* Morgan Kaufmann.

Cohen, P. R., and Levesque, H. J. 1990. Rational interaction as the basis for communication. In Cohen, P. R.; Morgan, J.; and Pollack, M. E., eds., *Intentions in Communication.* MIT Press.

Fagin, R. R.; Halpern, J. Y.; and Vardi, M. Y. 1991. A model-theoretic analysis of knowledge. *Journal of the ACM* (2):382–428.

Gmytrasiewicz, P. J., and Durfee, E. H. 1995. Formalization of recursive modeling. In *To appear at the First International Conference on Multiagent Systems, ICMAS'95.*

Gmytrasiewicz, P. J.; Durfee, E. H.; and Wehe, D. K. 1991a. A decision-theoretic approach to coordinating multiagent interactions. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, 62–68.

Gmytrasiewicz, P. J.; Durfee, E. H.; and Wehe, D. K. 1991b. The utility of communication in coordinating intelligent agents. In *Proceedings of the National Conference on Artificial Intelligence*, 166–172.

Halpern, J. Y. 1993. Reasoning about only knowing with many agents. In *Proceedings of the National Conference on Artificial Intelligence*, 655–661.

Harsanyi, J. C. 1967. Games with incomplete information played by 'bayesian' players. *Management Science* 14(3):159–182.

Horvitz, E. J.; Cooper, G. F.; and Heckerman, D. E. 1989. Reflection and action under scarce resources: Theoretical principles and theoretical study. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1121–1127.

Huang, H.-H.; Barzin, N.; and Lewis, F. L. 1994. A matrix framework for controller design for reentrant flow scheduling. In *Proceedings of the 33rd Conference on Decision and Control*, 1558–1562.

Ishida, Y.; Tomita, S.; Asama, H.; Ozaki, K.; Matsumoto, A.; and Endo, I. 1992. Development of an integrated communication system for multiple robotic agents. In *Proceedings of the International Symposium of Distributed Autonomous Robotic Systems*, 193–198.

Kumar, P. R. 1994. Scheduling semiconductor manufacturing plants. *IEEE Control Systems* 9(1):33–40.

Kusiak, A. 1988. *Artificial Intelligence in Industry.* Springer-Verlag.

Kusiak, A. 1992. *Intelligent Design and Manufacturing.* John Wiley & Sons.

Lakemeyer, G. 1993. All they know about. In *Proceedings of the National Conference on Artificial Intelligence.*

Lewis, F. L.; Huang, H.-H.; and Pastravanu, O. C. 1994. A matrix formulation for design and analysis of descrete event manufacturing systems with shared resources. In *Proceedings of the Systems, Man and Cybernetics.*

Lewis, F. L.; Pastravanu, O. C.; and Huang, H.-H. 1993. Controller design and conflict resolution in for descrete event manufacturing systems. In *Proceedings of the 32nd Conference on Control and Decision*, 3288–3293.

Mertens, and Zamir. 1985. Formulation of bayesian analysis for games with incomplete information. *International Journal of Game Theory* 14:1–29.

Neapolitan, R. E. 1990. *Probabilistic Reasoning in Expert Systems.* John Wiley and Sons.

Parunak, H. V. D. 1988. Distributed artificial intelligence. In Kusiak, A., ed., *Artificial Intelligence in Industry.* Springer-Verlag. 223–251.

Pastravanu, O. C.; Gurel, A.; and Lewis, F. L. 1994. Petri net based deadlock analysis in flowshops with kanban type controllers. In *Proceedings of the 10th ISPE/IFAC International Conference on CARs and FOF's*, 75–80.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufman.

Russell, S., and Wefald, E. 1991. Principles of metareasoning. *Artificial Intelligence* 49:361–395.

Shoham, Y. 1993. Agent-oriented programming. *Artificial Intelligence* 60(1):51–92.

Stuckey, R. J., and Pennington, R. A. 1988. Map and top. In Kusiak, A., ed., *Artificial Intelligence in Industry.* Springer-Verlag. 191–221.

Yuta, S., and Premvuti, S. 1990. Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple robots. In *Proceedings of the International Symposium of Distributed Autonomous Robotic Systems*, 173–181.