An Agent-Based Fractal Model of Agile Manufacturing Enterprises: Modeling and Decision-Making Issues

Venkat N. Rajan

Industrial and Manufacturing Engineering Department
Wichita State University
Wichita, KS 67260-0035
rajan@ie.twsu.edu

Abstract

In this paper we present an agent-based fractal model of agile manufacturing enterprises. The agent-based model can be applied to all levels of the agile enterprise. A game-theoretic decision-making framework is also presented that considers various combinations of local and global objectives of the agents. Research issues in modeling of, and distributed decision-making in, agile manufacturing enterprises are identified.

Introduction

The manufacturing enterprise has been revolutionized by the introduction of computer technology. Computer processing hardware and software is increasingly used by various design, manufacturing, and related functions. Networking hardware and software is making it possible to integrate these functions into Computer Integrated Manufacturing (CIM) enterprises. With the rapid development of inter-company networks, computer-based communication between manufacturing enterprises is becoming commonplace.

Agility is the term being used to describe the new form of the manufacturing enterprise that is using this technology to respond and survive. While agility can be defined at various levels within the enterprise (such as marketing, production, design, organization, management, and people [1]), the basic concept reflects the ability to quickly respond to change and take advantage of emerging opportunities. The manufacturing enterprise must be designed to be flexible, the people must be versatile, and the systems must be easily reconfigurable.

Clearly, agility requires the manufacturing enterprises to rethink their organization structures, procedures, and systems. Virtual organizations will have to be created that consist of elements of actual manufacturing enterprises. These organizations are formed to exploit the competencies of the elements to quickly respond to opportunities. Decision-making within agile enterprises is characterized by cooperation and conflict. Elements within the enterprise are cooperating and competing to utilize available resources and accomplish identified tasks. When an enterprise plans to participate in a virtual organization, alternate options may be available. The decision to select a particular option is essentially an issue of resolving cooperation and conflict issues. Even after the virtual organization has formed, cooperation and conflict will have to be effectively managed to maintain agility. With geographical distribution of elements within an enterprise as well as the likelihood of such distribution within the virtual organization, decision-making under cooperation and conflict will have to be accomplished in a distributed environment.

In this paper, we propose a new agent-based fractal model of the agile manufacturing enterprise. This model is based on a fractal view of the organization [2]. We will then address the research issues raised by the need for decision-making under cooperation and conflict in a distributed environment. A game-theoretic decision-making framework is also proposed to handle various combinations of local and global objectives of agents.

The rest of this paper is organized as follows: section II addresses the limitations of the current manufacturing enterprise organizational structure and discusses the agent-based fractal model. In section III, current Distributed Artificial Intelligence (DAI) approaches are evaluated from the perspective of the agile enterprise's decision-making needs. Research issues are also identified. Section IV presents the game-theoretic framework for decision-making under cooperation and conflict, and section V presents the summary and conclusions.

Agile Manufacturing Enterprise Model

Traditional Model

The traditional manufacturing enterprise organizational model involves three levels as shown in figure 1: strategic, operation, and execution [3]. At the strategic level, long-term decisions are made and generic resource constraints are set. At the operation level, the long-term decisions are decomposed into medium-term decisions

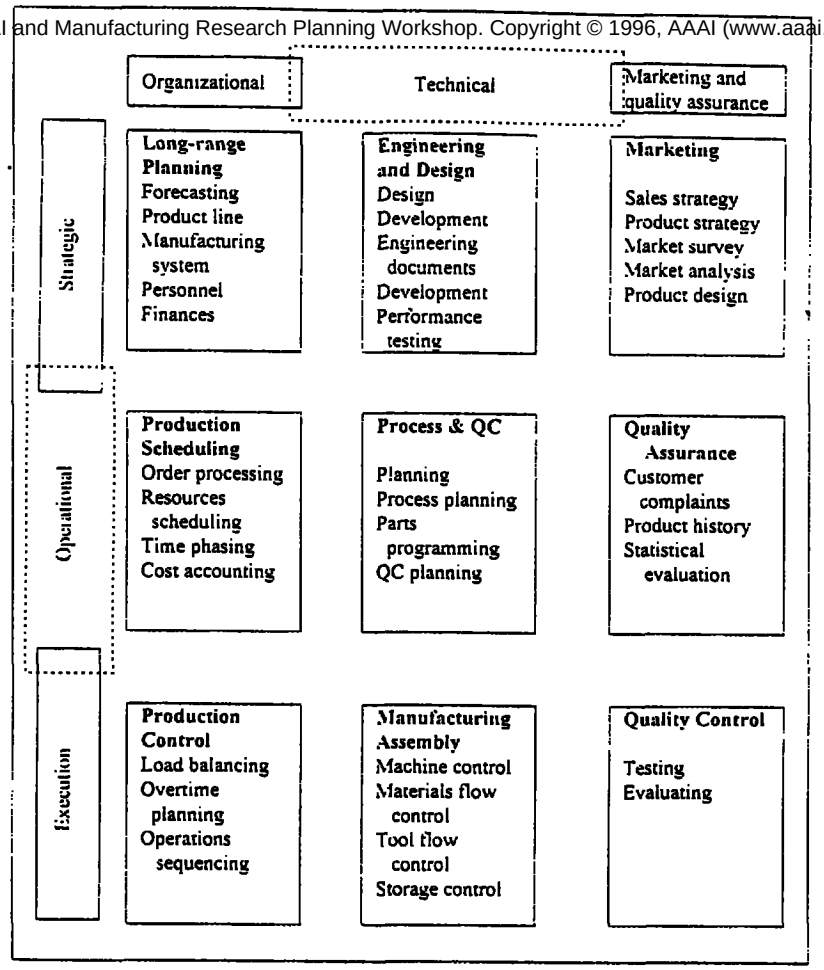

Figure 1: Traditional Manufacturing Enterprise Organizational Model [3].

and the constraints specifications are refined. At the execution level, the short-term decisions are made and executed in conformance with the operation level decisions and constraints.

For a specific example, consider the organizational column of figure 1, which represents the manufacturing planning and control (MPC) activities. The strategic level activity is aggregate or long-range planning in which the long-term demand forecasts and enterprise strategic objectives are used to determine production strategies and set resource constraints. At the operation level, the aggregate plan is decomposed into the master production schedule (MPS), which combines the forecasts with actual customer demands to determine medium-term production schedules for end products, and Material Requirements Planning (MRP), which provides detailed production schedules for subassemblies and components. At the execution level, the production orders are released to the shop floor and executed to produce the components, subassemblies, and end products to meet customer needs. In such a structure, decision-making is rigidly defined. Little or no attention is paid to the interests of participating entities such as departments, machines, or people. Change is difficult and expensive to accomplish, and problems arise due to poor management of resources. Many of the elements of agility, such as a system designed to be flexible, and reconfigurability of the system are lost due to these rigid procedures. Optimization procedures are also difficult to implement and their effect is generally local.

Agent-based Fractal Architecture

In order to maintain agility of the manufacturing enterprise, we propose an agent-based fractal model as shown in figure 2. Each level of the decision-making structure is

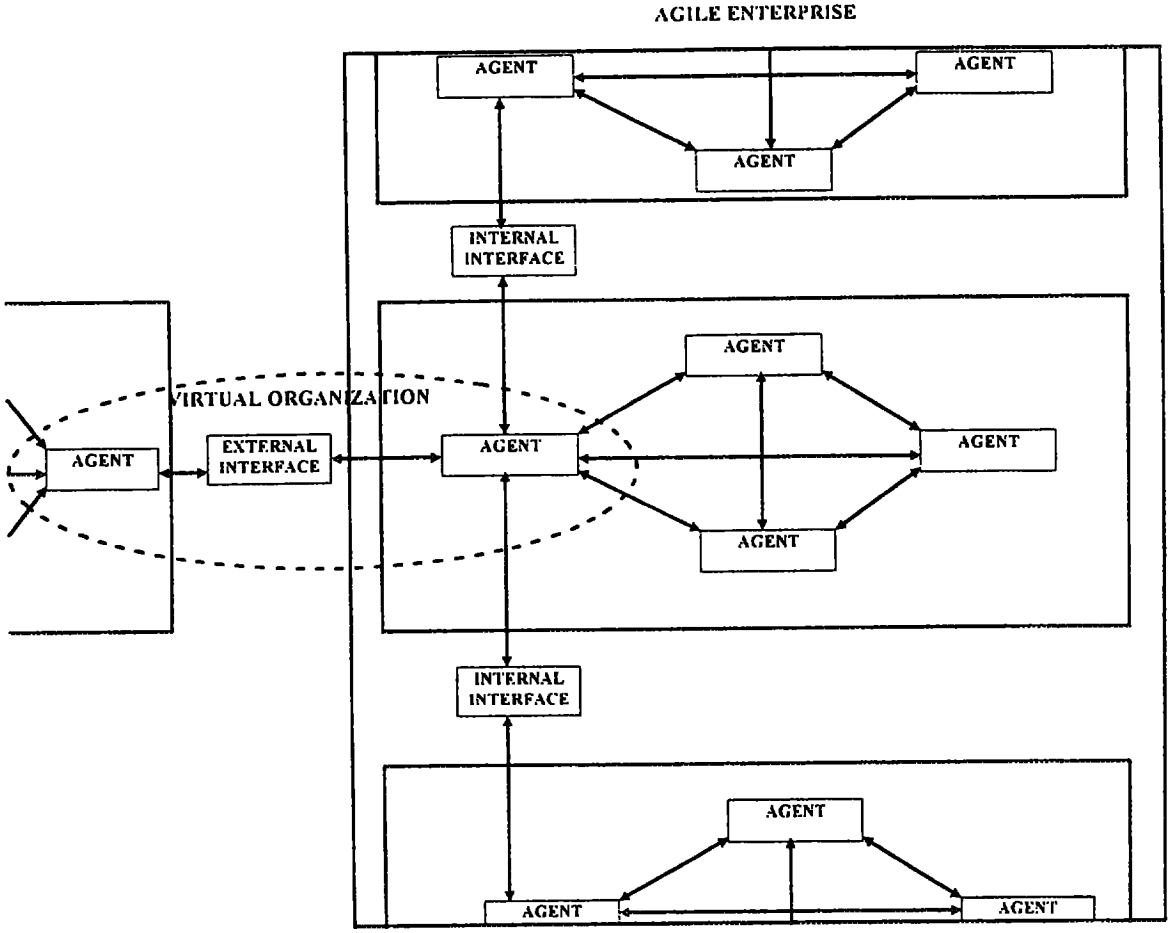

Figure 2: Agent-Based Fractal Model of the Agile Manufacturing Enterprise.

composed of interacting agents. Cooperation and conflict between agents is resolved to obtain decisions. Decisions are propagated from one level to the next by means of clearly defined interfaces. Decisions from higher levels are communicated to agent organizations at lower levels in a top-down model of decision-making. The status of each level is communicated back to the next higher level across the interface to provide feedback for decision-making.

A fractal view [2] of the decision-making structure is obtained by considering the fact that each level of the structure is an organization of agents, as shown in figure 2. In a function-based organization, at the highest levels, the various departments such as marketing, engineering, manufacturing, accounting, etc., would each be considered an agent. The strategic decision-making is accomplished by these agents and those at levels above them. The subfunctions within each department would form an organization of agents. For example, planning, manufacturing engineering, fabrication, and assembly would be the agent organization under the manufacturing department. The operation-level interaction between these agents would accomplish production of end-products. The people and associated systems would be considered agents at the next level. This concept is similar to that proposed by Frost and Cutkosky [4]. For example, in the design environment, the designer and the associated design system would be treated as an agent. In the manufacturing domain, machines and operators would represent agents. As the decomposition occurs, at the lowest levels of the structure, constraints and parameters can be represented as agents. For example, design for manufacture and assembly (DFMA) rules could be modeled as agents, and their interactions evaluated to modify the design. Manufacturing process parameters could be represented as agents and the characteristics of the fabricated component could be

viewed as the result of the interactions between the parameter agents. Thus, the fractal view is expressed by the fact that as one explores deeper and deeper within the structure the same agent pattern and decision-making structure is repeated. The decision-making structure is characterized by cooperation and conflict between agents at any given level. A plug-and-play architecture is also accomplished due to the characteristic of multiple agents making decisions under cooperation and conflict. Therefore, a generic model of a multi-agent system can be developed, and the implementation at each level of the organization architecture can be accomplished by creating instances of system variables. Such a generic model is shown in figure 3, with the feedback interface to the next higher level, and the decision-communication interface to the next lower level. An external interface also exists as shown in figure 3, to allow the agent organization at any given level to interact with organizations in other enterprises thus facilitating virtual organizations.

Cooperation and conflict interactions at any level of the enterprise organization arise due to local and global objectives of agents. Each agent has certain tasks it has to accomplish, and requires certain resources to accomplish these tasks. At the very minimum, contention for resources represents conflict interactions. As mentioned earlier, conflicts between design rules and process parameters can also occur. Cooperation interactions occur due to the global objective of agent organizations to successfully accomplish a set of tasks, to meet a constraint specified by a higher-level organization, or to operate using the feedback obtained from a lower-level organization. The nature and relative importance of local and global objectives may change from one level to the next. As mentioned earlier, the inputs, outputs, and operating knowledge change from one level to the next.

Why is the agent-based fractal model better suited to the agile manufacturing enterprise? As discussed earlier, agility is the ability to respond to change. The agent-based fractal architecture provides the ability to explicitly represent the entities and processes within the enterprise. The competencies of the enterprise elements are encapsulated within each agent organization. Reconfiguration of the enterprise is facilitated by the modular representation and separation of internal and external processes. By redefining the global and local objectives, their relative

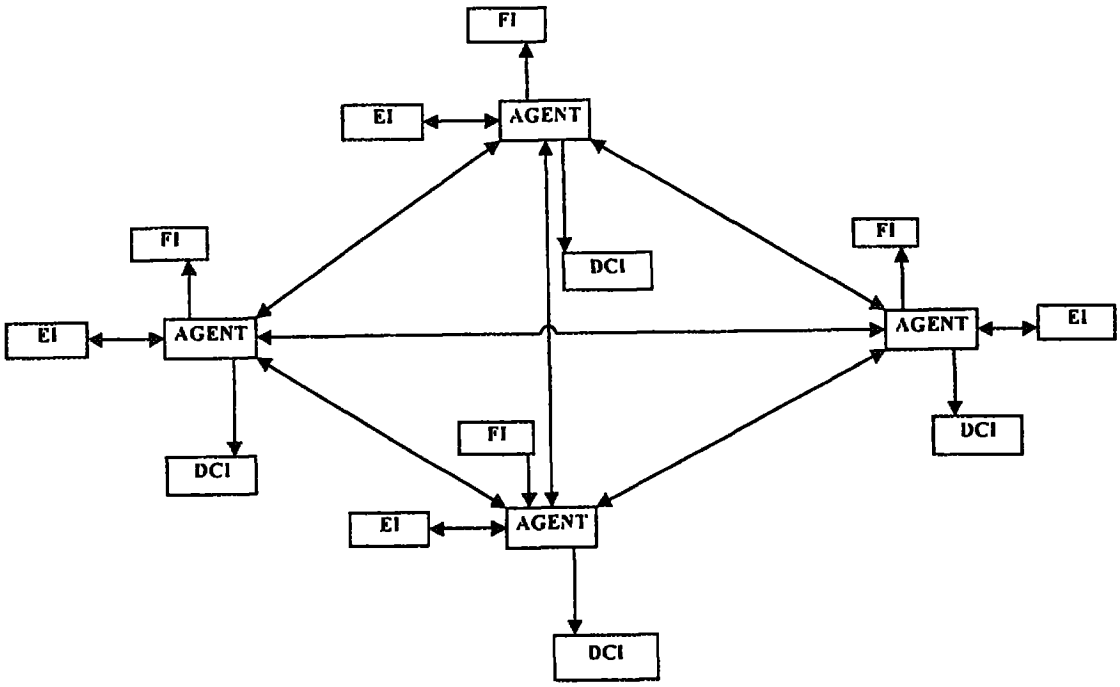

Figure 3: Generic Agent-Based Model that Facilitates a Plug-and-Play Architecture.

importance, and the domain knowledge representing the element competency, the internal processes can be quickly and easily reconfigured. By appropriately modifying the external interface along with the internal feedback and decision-communication interfaces, virtual organizations can be created and modified. Also, by maintaining internal interfaces with an enterprise element that is participating in a virtual organization, appropriate control can be maintained over the element from within the enterprise. Information hiding to protect the proprietary nature of some knowledge can also been accomplished because of the separation between the external interface and the internal knowledge of an agent organization. As entities and processes are created or eliminated within the manufacturing enterprise, the existing model can be updated by creating or deleting agent organizations. As mentioned earlier, the agent-based fractal model has a plug-and-play feature that allow for development of generic agent organizations which are then instanced at various levels within the enterprise. While optimization procedures can be implemented within the decision-making process of the individual agent organizations, the results may still be local. If the interfaces are properly designed, it may be possible to evaluate the global impact of local decisions quickly.

A decision-making approach that supports the agent-based fractal model specified in the previous section must possess the following characteristics:

1. It should take into account both global and local objectives, i.e., cooperation and conflict interactions, simultaneously. It must be possible to model and analyze multi-agent systems that operate under purely global, purely local, or any combination of local and global objectives.
2. It must be possible to predict the behavior of the multi-agent system given the local and global objectives and their relative importance. Thus, the system designer can either manipulate these objectives to obtain a particular behavior, or model existing objectives and predict the behavior.
3. It must be possible to abstract from a particular agent organization implementation to perform decision-making. While this is not directly an issue related to the decision-making approach, it is important that the decision-making approach use an abstract representation of the data to arrive at decisions. This allows us to develop a generic decision-making model that can be implemented in a plug-and-play fashion.

Various research issues related to the agent-based fractal model are currently being addressed [5]. These include a game-theoretic decision-making approach, agent models, and interface specifications. An object-oriented approach is being adopted in which the abstraction, encapsulation, and inheritance capabilities are being exploited to model agent and interface representations and hierarchies. The message passing capability is being used to facilitate decision-making.

Multi-Agent Decision-Making Under Cooperation and Conflict

Various researchers have studied the issue of distributed decision-making within the Distributed Artificial Intelligence (DAI) domain. One of the earliest efforts involved the development of the Contract-Net protocol [6] in which a bidding approach is used to allow agents to cooperatively solve problems. This approach has been applied in the manufacturing domain, mainly to create opportunistic shop-floor scheduling systems [7,8]. Another major research effort involves the development of and experimentation with the Distributed Vehicle Monitoring Testbed (DVMT) [9,10]. In the DVMT implementation, agents monitoring regions are required to cooperate to create global maps of vehicle trajectories. We are not aware of any direct manufacturing applications of this research.

Game theory has been used [11] to study cooperation and conflict within agent organizations. The primary emphasis has been on defining various types of rationality to allow decision-making. The communication needs to allow agents to converge to a single plan are also discussed. In our previous research [12,13,14], we have explored the application of game theory to multi-machine cooperation. It is shown that cooperative game theory concepts are useful to accomplish cooperation in multi-machine systems, however, detailed studies have not been conducted. Other approaches have included resolving conflicts between individual agent plans to create a global plan [15,16], and studies of negotiation protocols to allow agents to converge to a single plan [17].

There have been a few applications of DAI multi-agent concepts to manufacturing related problems. As mentioned above, the contract-net protocol has been applied to the factory control problem [7]. A distributed scheduling approach has also been developed using a multi-agent approach in which partial plans generated by agents are combined to obtain an overall production schedule [18]. Other applications include development of multi-agent architectures for collaborative design [19], for interaction between designers and manufacturing process information sources [4], and production control [20]. Some recent research efforts have attempted to develop an architecture for Distributed CIM [21], concurrent engineering [22], and enterprise information support [23]. The latter approach is particularly interesting because of the integration of DAI and object-orientation concepts. Some attempts have also been made to apply DAI concepts to

robot task planning activities [24,25].

Most of the DAI manufacturing applications are limited in their scope [4,7,18,19,20,22] or lack a formal basis [21,23]. The lack of a formal basis prevents abstract evaluation of parameters and mathematical characterization of system behavior. Most of the manufacturing applications also implicitly assume the existence of a purely global objective [18,20].

So, which of the existing DAI approaches should we apply to the agent-based fractal enterprise model? Most of the current DAI approaches implicitly or explicitly assume the existence of only a global objective [6,9]. The game-theoretic approaches [11,12,14] have the potential to handle any combination of local and global objectives, however little research exists on this issue. Game theory generally considers only the local objectives of the agents. However, in our prior work [14], we have shown that in cooperative game theory, by controlling coalition formation and the characteristic function of the game, global objectives can be imposed.

In order to predict the behavior of a multi-agent organization, given the local and global objectives, a formal mathematical basis is required. Such a basis is also essential for decision-making using an abstract representation of data. Most of the DAI approaches lack a formal basis. They have evolved by logical analysis of cooperative behavior or by empirical study. The game-theoretic approaches possess the extensive mathematical basis as well as the abstract problem and data representation. The information regarding the interactions between agents is generally modeled in the characteristic function form [26] as the number of agents increases. Such an abstract representation allows us to study the nature of cooperation and predict the behavior of multi-agent systems without considering the details of the implementation.

While we have outlined some promising approaches to decision-making in the agile manufacturing enterprise, these approaches still need to be validated by detailed analysis. Many other important DAI research issues also exist that need to be addressed before successful implementations can be developed. Some of these issues are listed below:

1. Are there formal models of multi-agent interactions? What is their relevance to manufacturing enterprises and systems? We believe that game theory provides such a formal approach and that it is relevant to decision-making in agile manufacturing enterprises. Are there other models and are they relevant?
2. Is there a unified methodology for analyzing cooperation and conflict interactions that arise under different combinations of global and local objectives (purely global, purely local, and any combination)? Again, we believe that game theory provides us with this capability. However, detailed development of the methodology is lacking.
3. Would different DAI models be suitable for different levels of the agile manufacturing enterprise?
4. What are local and global objectives at various levels of the enterprise and what is their relative importance?
5. What would be the differences between the decision-making models for different types of manufacturing industries?
6. What are the computation and communication loads for various decision-making theories and architectures (distributed vs. centralized)?
7. Develop benchmarking problems on which various enterprise architectures as well as decision-making approaches can be tested.
8. What is the frequency of decision-making at various levels of the enterprise? Is the decision-making environment monotonic or non-monotonic?
9. What is a good methodology for modeling agents and interfaces? We believe that an object-oriented methodology provides the ability to accomplish the information hiding, abstraction, hierarchical relations, and communication needs of the agent-based fractal model.
10. What are standard protocols for communicating information and decisions? If virtual organizations are to form, data and decisions will have to be transferred between agent systems from different vendors. Currently, standards for information representation are available only for the CAD systems (IGES, STEP) and the newer standards (STEP) capture the information about the product life-cycle. How can the decisions and information at higher levels be standardized?
11. What is the effect of incomplete and incorrect information on the decision-making approach?

Game-Theoretic Decision-Making Approach

N-person Game Theory Concepts

N-person game theory addresses situations of partial conflict of interest with the following attributes [26]:

1. Three or more players participate in the game ($n \geq 3$),
2. Players can openly communicate with each other, and
3. Only one player wins a game (zero-sum) or players can form subsets, called *coalitions*, by freely negotiating binding agreements on how to divide the benefits accrued by acting jointly.

A game can be represented in three ways depending on the level of abstraction used [27]:

1. *Extensive form*: In this form, the moves of the play-

ers are represented in all possible orders in which they can be executed. This results in a game tree. The payoff to each player corresponding to each outcome of the game is known. The analysis is aimed at identifying the set of moves that will be most beneficial for each player given the knowledge of all the moves that can be executed by the other players.

2. *Normal or strategic form*: Instead of dealing explicitly with the moves, the strategies available to each player and the corresponding outcomes are represented. A solution method then has to find the best strategy for each player such that the corresponding payoff is maximized.
3. *Characteristic function form*: In this form, the payoffs for each subset of players is represented in function format, without specifying the actual strategies used to obtain those payoffs. This level of abstraction allows for an analysis of the trade-offs involved in selecting one solution over another.

The n-person non-cooperative game is analyzed in the normal or strategic form. In such a game, each player is looking to maximize his/her payoff while acting independently. It has been shown that such a game has an equilibrium solution defined as the *Nash equilibrium*. The Nash equilibrium solution exists when players consider both deterministic strategies (pure) and their probabilistic combinations (mixed) [28].

The n-person cooperative game is generally modeled in the characteristic function form [27]. The characteristic function represents the benefits obtained by an individual or group of players. Thus, it is also referred to as the *value function* [26]. Formally, the characteristic function v is defined over the set of players n as a real-valued function that assigns a real number $v(s)$ to each set of players s, $s \subseteq n$. In an *essential game*, the value accrued by a set of players s, $s \subseteq n$, is at least as large as the sum of values obtained by any two subsets of s, i.e.,

$$v(s) \geq v(a) + v(b), \text{ for all } a, b, \text{ where } a, b \subseteq s.$$

The objective of cooperative game theory is to determine the payoffs to each player participating in the game. The payoffs are represented by a payoff vector, $X = (x_1, x_2, ..., x_n)$. It is assumed that players behave "rationally". Rationality can be of two forms [27]:

1. *Individual rationality*: Each player is expected to behave rationally, i.e., each player is expected to prefer any payoff vector X that give him/her a higher payoff than if he/she had acted alone, i.e., $x_i \geq v(i)$ for all i, $i \in n$.
2. *Group rationality*: In certain situations, this condition is imposed where it is expected that no subset of players will accept a payoff vector X that gives them less than what they would obtain as the result of forming a coalition, i.e., $\sum_{i \in s} x_i \geq v(s)$, for all s, $s \subseteq n$.

A game is said to be *constant-sum* if

$$\sum_{i \in n} x_i = v(n)$$

There are a variety of solutions defined for an n-person cooperative game depending on the assumptions of individual and/or group rationality. These are the *core*, the *stable sets*, the *Shapley value*, the *bargaining set*, and the *kernel* [27]. We will specifically discuss the *Shapley value* and the *kernel* solution concepts and their implications.

The *Shapley value* [26] is a payoff vector that represents the expected payoffs to each player participating in the game under the assumptions of individual and group rationality. The *Shapley value* is a unique solution that has equity built into it. It is implicitly assumed that the *grand coalition* is formed and this solution describes the way in which the value of the game to the *grand coalition* is divided among the players. If smaller coalitions are desired, then the group rationality assumption needs to be eliminated altogether. This is the basis for the *kernel* solution concept discussed below.

A coalition structure $S = \{s_1, s_2, ..., s_m\}$ is a partition of the set of players n, such that

$$s_j \neq \varnothing$$
$$s_i \cap s_j = \varnothing \text{ for all } i, j = 1, 2, ..., m, i \neq j, \text{ and}$$
$$\cup_j s_j = n$$

An *individually rational payoff configuration* (irpc) is defined as the combination of a payoff vector and coalition structure, i.e., (X,S), satisfying the conditions of individual rationality and

$$\sum_{i \in sj} x_i = v(s_j)$$

The *kernel* is an equilibrium solution that is defined as the set of individually rational payoff configurations such that every pair of players in any coalition in the coalition structure are in *equilibrium* with respect to the specified payoff vector. Equilibrium is defined as the situation where neither player can obtain a larger payoff by forming a new coalition in which the other player is excluded.

In the above solution methods, either the *grand coalition* forms, or a coalition structure forms in which each coalition obtains at least as much as the value of the game, i.e., a sense of equilibrium is maintained.

Agile Enterprise Decision-making Approach

The general decision-making problem at any level of the

agile manufacturing enterprise that we wish to address is formulated as follows:

"A set of tasks need to be accomplished by a set of agents. A task is defined by the inputs and domain knowledge of the agent organization. Decisions need to be made in a consistent manner taking into account the objectives of the agents and the current state of knowledge."

We propose to view the multi-agent decision-making problem from the perspective of various levels of local and global objectives. Thus, three different situations may arise [14]:

1. *Purely global objective*: In this case the agents cooperate to perform the tasks such that only the global objective function is optimized. We will refer to this as the *coordination* problem.
2. *Purely local objective*: When each agent is attempting to perform its assigned tasks such that its local objective is optimized, and no global objective has been specified, then this situation exists. In this case, purely conflict interactions exist between the agents. We will refer to this as the *conflict* problem.
3. *Combination of local and global objectives*: When both local and global objectives exist, the interactions between the agents have to be resolved such that the combination of the local and global objectives are optimized. We will refer to this as the *coalition* problem.

Game theory provides a unified framework to handle all the three situations. The *coordination* problem is basically one of formation of the *grand coalition*. The *Shapley value* concept can be used for decision-making under this mode. As mentioned earlier, the *Shapley value* represents an equity solution and is therefore attractive from the concept of a benevolent situation where agents are not in conflict with each other [11]. However, the game-theoretic approach does not consider any global objective function. Therefore, this function has to be built into the characteristic function. One approach, is by defining a non-zero value for only those coalitions that accomplish all the identified tasks. Such coalitions are called *feasible coalitions*. Any coalition structure that does not provide an agent or set of agents to perform one or more of the tasks will be given a value of 0, and therefore ignored in the computation [13,14].

The *conflict* situation is similar to the assumptions of the n-person zero-sum game, and therefore, the *Nash equilibrium* solution concept can be used to resolve the conflicts. The *coalition* problem is similar to the game-theoretic concept of determining which coalition structure will form given the assumption of only individual rationality. The game-theoretic solution concept of the *kernel* is applicable to the *coalition* problem. The advantages of the *kernel* solution are that it requires very few computations, the number of alternate solutions are limited, and it represents a set of equilibrium solutions [26].

The unifying characteristic of game-theoretic analysis for the decision-making problem arises in the common definition of the characteristic function required by the *coordination* and *coalition* problems. In order to use the above concepts, the characteristic function needs to be defined. For the *conflict* problem, the *Nash equilibrium* solution requires the normal (or strategic) form of the game.

In general, each agent will be aware of all possible actions that can be taken. When combinations of the actions by different agents are considered, the extensive form of the game will result. The normal form is a summary of the combinations without reference to the actual actions that are possible. The normal form can then be converted to the characteristic function form in which only the payoffs are listed against the possible coalitions that can form. Therefore, the input information to each agent organization will have to be converted using the domain knowledge into the possible strategies for each agent. This is sufficient to perform the *Nash equilibrium* analysis for the *conflict* problem. However, for the *coordination* and *coalition* problems, the normal form will have to be further abstracted into the characteristic function form to compute the *Shapley value* and *kernel* solutions. In order to accomplish the implementation of these solutions, the strategies and payoff configurations will have to be converted back into actions that the agents can execute.

The game-theoretic solutions concepts presented above can easily be computed in a centralized decision-making architecture. However, in the agent-based fractal model specified above for the agile enterprise, the decision-making within each agent organization will be distributed. The solution approaches will have to be extended to arrive at the centralized solution using a distributed decision-making architecture. Issues of static and dynamic knowledge representation, monotonicity, and convergence become problems under such an architecture. Some research issues related to the game-theoretic decision-making approach are as follows:

1. How can the inputs and domain knowledge be converted into agent actions and strategies? How can the payoff vectors and the characteristic function form of the game be derived?
2. Is there a distributed decision-making methodology that can reach the solution generated by the centralized approach? Is it guaranteed to converge to a single solution?
3. How can the static and dynamic knowledge be represented in agent models? Is the dynamic knowl-

edge non-monotonic? How does this affect the decision-making process?

4. What are the communication needs of the different game-theoretic approaches?

Summary and Conclusions

Manufacturing enterprises are moving towards an agile environment in which distributed decision-making under cooperation and conflict will be essential. We have presented a new agent-based model of the enterprise that can be applied to all levels of the organization. The new model allows us to develop a fractal view of the enterprise because the same agent pattern can be seen from the highest to the lowest levels.

While many Distributed Artificial Intelligence (DAI) approaches have been developed, most of them lack a formal basis and lack consideration of various combinations of local and global objectives that is essential in an agile enterprise. It has been shown that the game-theoretic approach provides a unified theory for situations involving purely global (*coordination*), purely local (*conflict*), or a combination of local and global (*coalition*) objectives. This is important because different situations may occur at different levels of the enterprise and a unified theory allows us to develop a decision-making framework that is independent of the particular implementation.

Many open research issues have been identified both in modeling of agile manufacturing enterprises and in DAI decision-making approaches. By providing solutions to these problems and developing benchmark cases, the benefits of applying DAI concepts to agile manufacturing enterprises can be realized in the future.

References

1. Goldman, S. L., R. N. Nagel, and K. Preiss. 1995. *Agile Competitors and Virtual Organizations*. Van Nostrand Reinhold.
2. Warnecke, H.-J., and Hueser, M. 1993. *The Fractal Company - A Revolution in Corporate Culture*. Berlin: Springer.
3. Rembold, U., Nnaji, B. O., and Storr, A. 1993. *Computer Integrated Manufacturing and Engineering*. Reading, MA: Addison-Wesley.
4. Frost, H. R., and Cutkosky, M. R. 1996. Design for Manufacturability Via Agent Interaction. To Appear In Proceedings of the 1996 ASME Design for Manufacturing Conference. Irvine, CA.
5. Rajan, V. N., and Shashikiran, P. 1996. A Formal Architecture for Integrated Multi-Level Decision-Making in Agile Manufacturing. Submitted to the Photonics East '96 Symposium on Plug and Play Software for Agile Manufacturing. Boston, MA.
6. Davis, R., and Smith, R. G. 1983. Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence* 20:63-109.
7. Van Dyke Parunak, H. 1987. Manufacturing Experience with the Contract Net. In *Distributed Artificial Intelligence*, M. N. Huhns, ed. Los Altos, CA: Morgan Kaufman.
8. Shaw, M. J., and Whinston, A. B. 1987. A Distributed Knowledge-Based Approach to Flexible Automation: The Contract-Net Framework, BEBR Faculty Working Paper No. 1384, College of Comm. and Bus. Admin., Univ. of Ill.
9. Durfee, E. H., Lesser, V. R., and Corkill, D. D. 1987. Coherent Cooperation Among Communicating Problem Solvers. *IEEE Transactions on Computers*. 36:1275-1291.
10. Durfee, E. H., and Lesser, V. R. 1987. Using Partial Global Plans to Coordinate Distributed Problem Solvers. Proceedings of the 1987 International Joint Conference on Artificial Intelligence. 875-883.
11. Rosenschein, J. S. 1985. Rational Interaction: Cooperation Among Intelligent Agents. Ph.D. diss., Stanford Univ.
12. Rajan, V. N., and Nof, S. Y. 1990. A Game-theoretic Approach for Co-operation Control in Multi-machine Workstations. *International Journal of Computer Integrated Manufacturing*. 3:47-59.
13. Rajan, V. N., and Nof, S. Y. 1992. Logic and Communication Issues in Cooperation Planning for Multi-machine Workstations. *International Journal of Systems Automation: Research and Applications*. 2:193-212.
14. Rajan, V. N. 1993. Cooperation Requirement Planning for Multi-Robot Assembly Cells. Ph.D. diss. Sch. of Industrial Engineering, Purdue University.
15. Georgeff, M. A. 1986. The Representation of Events in MultiAgent Domains. Proceedings of the Fifth National Conference on Artificial Intelligence. 70-75.
16. Georgeff, M. A. 1983. Communication and Interaction in MultiAgent Planning. Proceedings of the Second National Conference on Artificial Intelligence. 125-129. Washington, D.C.
17. Conry, S. E., Meyer, R. A., and Pope, R. P. 1989. Mechanisms for Assessing Nonlocal Impact of Local Decisions in Distributed Planning. In *Distributed Artificial Intelligence, Volume II*, L. Gasser and M. N. Huhns. eds. San Mateo, CA: Morgan Kaufman.
18. Sycara, K. P., Roth, S. F., Sadeh, N., and Fox, M. 1991. Resource Allocation in Distributed Factory Scheduling. *IEEE Expert*. 6:29-40.
19. Bond, A. H. 1989. The Cooperation of Experts in Engineering Design. In *Distributed Artificial Intelli-*

gence, Volume II, L. Gasser and M. N. Huhns. eds. San Mateo, CA: Morgan Kaufman.

20. Shih, W., and Srihari, K. 1995. Distributed Artificial Intelligence in Manufacturing Systems Control. *Computers in Industrial Engineering*. 29:199-203.
21. Spinosa, L. M., Chouraqui, E., and Espinasse, B. 1995. Distributed CIM and DAI: For a Knowledge and Multiagent Approach. Proceedings of the 1995 IEEE International Conference on Systems, Man, and Cybernetics. 3397-3402. Vancouver, BC, Canada.
22. Maturana, F. P., and Norrie, D. H. 1995. Generic Mediator for Multi-Agent Coordination in a Distributed Manufacturing System. Proceedings of the 1995 IEEE International Conference on Systems, Man, and Cybernetics. 952-957. Vancouver, BC, Canada.
23. Mentzas, G. 1994. Object-based Intelligence in Office and Production Processes a View of Integration. *Information and Decision Technologies*. 19:195-210.
24. Kamel, M., and Syed, A. 1994. Multiagent Task Planning Method for Agents with Disparate Capabilities. *International Journal of Advanced Manufacturing Technology*. 9:408-420.
25. Arkin, R. C. 1992. Cooperation Without Communication: Multiagent Schema-based Robot Navigation. *Journal of Robotic Systems*. 9:351-364.
26. Kahan, J. P., and Rapoport, A. 1984. *Theories of Coalition Formation*. Hillsdale, NJ: Lawrence Erlbaum.
27. Rapoport, A. 1970. *N-Person Game Theory: Concepts and Applications*. Ann Arbor, MI: The University of Michigan Press.
28. Shubik, M. 1984. *Game Theory in the Social Sciences: Concepts and Solutions*. Cambridge, MA: The MIT Press.