Leadership Based Agent Architecture

Evolutionary Agent Architecture for Distributed Engineering Design

Joshua D. Summers and Jami J. Shah
Design Automation Laboratory, Arizona State University, Tempe, AZ 85287, USA

Abstract: This paper discusses an intelligent and evolving agent based CAD system to support collaborative design. Such systems consist of autonomous and mobile agents, each performing specialized functions. The question is how to resolve conflicts between agents. No structured general methods have yet been established to this end. This paper proposes selecting a leadership style dynamically in the agent architecture based upon the state of the engineering design environment. This tool is derived from research in the psychology, sociology, and political science fields of leadership theory and applied to the evolving design artifact in mechanical engineering design.

Key words: Agent Architecture, Agent Leadership, and Engineering Design Automation

1. INTRODUCTION

From the work of sociologists and psychologists, it is recognized that many human activities and decision-making processes, including engineering design, are collaborative (Epstein and Axtell, 1996; Resnick, 1998). Engineers, during collaboration in the design process, each have a speciality or expertise. They exert influence over the process when issues arise that need the expertise of the engineers. The engineers must collaborate, negotiate, and resolve conflicts between each other in order to develop the best possible design. Thus, the efforts to expand the computer science field of AI into collaborative communication, decision-making, and conflict resolution led to the development of agents.

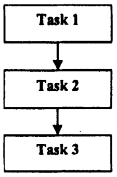
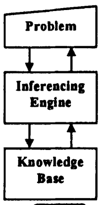
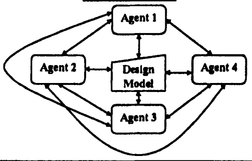
Agents may be generally defined as autonomous programs, or actors, that possess one or more of the following characteristics: reactive, collaborative, and mobile. More concisely, agents are considered to be autonomous, mobile, social entities that communicate and interact with other agents (Deshmukh, 1999). Agents have been used to aid engineers and humans in negotiation (Krishna and Ramesh, 1997), management systems (Azmoodeh and Davison, 1997), constraints solving (Eaton, et al., 1998), and many other situations (Gero and Sudweeks, 1991; Darr and Birmingham, 1992; Edmonds, et al., 1994). In order to provide the infrastructure necessary for agent architecture implementation, several issues must be addressed: agent communication (Petrie, 1996; Sundstead, 1999; Danesh and Yan, 1999), conflict resolution between agents (Krishna and Ramesh, 1997; Cohen, 1999; Goel, et al., 1996), decision-making (Deshmukh, 1999; Barber, et al., 1999), and agent organization (Corkel and Lander, 1999; Azmoodeh and Davison, 1997).

The current work with agent architecture lacks a discernible accommodation in the structure for the evolution of the design throughout the entire design process. Agent architecture has been primarily focussed on single application programming. The importance of identifying a method of selection of leadership styles is clear when considering the multiple states through which the design environment may progress (Pahl and Beitz, 1995; Ullman, 1992). This paper will briefly discuss the four fundamental issues of agents: communication, decision-making, conflict resolution, and organization. Next, a discussion will be provided to suggest the need for an evolving leadership construct. Based upon this, an agent system architecture is detailed. Finally, a simple example is provided to illustrate this system.

2. BACKGROUND

Engineering design has been facilitated through computer-aided design for several years. Three basic CAD architectures have evolved for engineering design: procedural, knowledge based, and agent based. These general architectures are compared in Table 1. The system architectures provide different benefits and encounter different limitations. Two basic agent architectures are found in the literature: the "black board" and distributed. The "black board" utilizes a brokered communication scheme, where a single agent handles the messages between the various agents, distributing problems and collecting responses. In this manner, conflicts are resolved in a centralized approach. The second type of agent architecture is a distributed approach with direct communication between agents. Conflicts may arise between agents competing for resources or offering solutions to sub-problems.

Table 1. CAD Architecture Comparison

CAD Architecture	Description	Limitation	Benefit	Illustration
Procedural	Classical programming technique	Single application, not flexible, knowledge and reasoning intertwined	Easy to develop	
Knowledge Based	Separates the knowledge base from the reasoning engines	All knowledge must be defined before commencing	Easy to modify KB, adapt reasoning engine	
Agent Based	Distributed decision making based upon expertise	Suitable for complex problems, flexible	Agents activate when needed with own view	

2.1 Communication

Communication is the prominent area of research with agents. Much work is being done to develop faster means of information transfer between agents. Further, agents must communicate in a common language. Unfortunately, the value that one agent seeks to increase might be different in nature from the value sought by a second agent. These might be interrelated variables. Communication is the skeleton of organizations, both natural and synthetic; without which, organizations would lose efficiency and eventually dissolve (Sundsted, 1999).

Agent communication is dependent upon the connectivity between agents. Situations arise when agents of different function families may need to communicate. Thus, a common language is needed or a new translation agent is required (Lyons, et al., 1999). Completeness of information and the reliability of information are other issues for consideration in agent-based systems. Decisions must be made in the design process with incomplete and assumed information (Orelup, et al., 1987). Thus, the information provided to the agent to make decisions based on variable decisions is not completely fixed and thus the decisions must be viewed skeptically. Approaches to addressing the reliability issue include probabilistic design (Pahl and Beitz, 1995), fuzzy logic (Elkan, 1993), truth maintenance (Jin and Zhou, 1999), and sensitivity analysis (Dixon and Poli, 1995). As the design progresses and the information gains stability, the decisions become more reliable.

2.2 Decision Making

Decision-making is important in developing agent architectures. Decision-making is focussed on how individual agents make choices with the available information. A straight deterministic form of decision-making would be ideal if all agents were dealing with a single common utility value, such as cost. However, a single agent may be concerned with how its decision will affect multiple different objective functions. Thus, the agents need to decide locally between different options with different outcomes for each. Different forms of decision-making have been introduced into agent architectures. Some general strategies used in decision-making include: utility theory, game theory, cost-benefit analysis, rule based, constraint satisfaction, and uncertainty modelling. Some types of uncertainty modelling include probabilistic design, fuzzy logic, and truth maintenance. Each of these approaches may be used in developing strategies for agents based decision-making. The two more popular approaches are utility and game theory.

In order for utility theory to be truly useful, agents must be capable of developing utility functions without complete system knowledge, leading to uncertainties and potential errors (Goel, et al., 1996; Hazelrigg, 1996). The utilities used in the decision making process internal of many agents are the Von Neumann-Morgenstern utilities. This allows combination of multiple utilities because of the independence condition. Agents with many goals may satisfy the independent utilities separately. Decision theory provides choices between alternatives based upon the value or utility of the result of the choice. The utility of the choice is based upon the usefulness of the design. The metrics used to determine the utility of a choice differ depending upon domain, application, and designer preference. Utility is generally modelled as a function against a parameter of the design.

Some agent systems employ a game theory approach to model a specific human social system and are thus descriptive models (Krishna and Ramesh, 1997). These systems allow co-operation between agents to achieve a higher benefit than would achieve individually working in a competitive system. This theory has its roots in the analysis of conflict as well as modelling and simulation (Thomas, 1987). Game theory considers many of the states of the design environment: number of players, type of information, and type of game (Rapoport, 1974). The type of information available in most design environments may be considered incomplete. A complete set of information would require perfect understanding of the design space. The type of game, a zero sum game or a win-win situation, is important when developing agent architecture: a zero sum game implies a winner and loser and a win-win situation implies all receive some benefit from a negotiated solution. Game theory assumes rational choices and decision-making.

A final type of decision-making that is currently being used in many distributed systems is user oriented. The system may provide the user with alternatives, but ultimately user makes the decision (Yang, et al., 1999). In this manner, there is no automation of the decision-making process.

2.3 Conflict Resolution

Conflict resolution arises from proposed design changes by agents that are in direct conflict with each other's objectives. Conflict exists in all societies (Bercovitch, 1984). The resolution of the conflict may be achieved through three basic modes: coercion, bargaining and negotiation, and third party intervention. In agent architecture, all three of these modes may be employed. However, the most prominent is the use of bargaining and negotiation. Some sources of conflict resolution include limited resources, different values, incompatible objectives, and change in state (Bercovitch, 1984). The potential negative outcomes of conflict in agent systems could be a poor decision leading to a poor design.

Negotiation is simply the exchange of ideas, information, risk, or value between two concurring agents (Karrass, 1970). The negotiating agents must speak a common language. This is a driving force in current agent research. Negotiation between agents may be implemented by assigning values to different decisions that may be made by the agents. These decisions are then traded as a virtual commodity with the intent to satisfy the local agents and to advance the value of the complete system. The aspiration indicates the intended performance goal.

Bargaining requires both information and disclosure (Kniveton, 1989). The more information available to each participant, the higher value each player will garner. Additionally, with true bargaining, a certain degree of trust must be established through disclosure. With respect to agents, this disclosure takes the form of reliability of information. Should the information be highly reliable, then the agent that provides the information might be considered "trustworthy". Finally, the goals of the individual may not represent the goals of the organization. When these goals do not coincide a representative from the company or organization must negotiate with the participant (Kniveton, 1989). Agents, being computer based, are designed to be exclusively rational; removing many of the hindrances encountered with social problems (Bazerman and Neale, 1992).

2.4 Organization

The organization of assigning tasks to agents and choosing between different solutions is vital to the architecture. Generally, most agent architectures have static organization that allows for agents to dynamically move through the system (Cohen, 1999). However, the final decisions made in the system are made in the same manner regardless of the abstraction of the design. As will be discussed later, an evolving leadership style might be best suited for the changing environment of engineering design.

In social constructs, leadership has been studied as a form of delegation of decision-making, charisma, and structure. With computerized agents, charisma need not be addressed. The leadership type is important in developing the agent architectures because it determines the need for agents to submit to authority or to challenge it. Several structures of leadership have been identified (Table 2).

Table 2. Sociological Leadership Styles

Type	Description	Who Decides	How Selected
Dictator	Single Leader	Dictator	Assumed
Oligarchic	Multiple Leaders	Leader consensus	Appointed
Representative	Multiple Leaders	Reps. Consensus	Elected
Democratic	Divested Leaders	All Consensus	Automatic
Arbitrator	Negotiator	Conflicting agents	Appointed
Hierarchic	Stratified Leaders	Depends	Assumed, Elected, Appointed

Each of these types of leadership has different benefits and drawbacks. Many are already employed in agent architectures, but few have been incorporated together in a single architecture. It is necessary to identify when different types of leadership style might be more beneficial than other leadership styles. The most prevalent method of leadership style in agent architectures is the dictatorial style (decision made by single agent), as in the blackboard (Skarmeas, 1999); while the democratic styles (consensus required for all agents) have gained some popularity (Martin, et al., 1999). Some hybrid static systems include the layered system of Nwana and Ndumu (1998) and InteRRap (Mueller, et al., 1995).

3. LEADERSHIP RULES

Agent architecture has been almost exclusively interested in the development of communication lines between agents. For this reason, only single leadership styles have been used in agent system architectures at any given time (Cohen, 1999). The selection of leadership style also depends upon the design and sub-design decisions. For many scenarios, micromanaging a design by a single agent is impractical. Constraints need to be established on the leadership structure to allow design agents some "free will" in executing decision-making (Barron-Carro and Gordillo, 1999).

Psychologists and sociologists have developed rules guiding the choice of leadership style. A prominent model for leadership decision-making is the Vroom-Yetton model (Vroom and Jago, 1978). This model describes two sets of rules: rules intended to protect the quality of decision and rules to protect the acceptance of the decision. The quality rules include: Leadership Information (LI), Goal Congruence (GC), and Unstructured Problem (UP). These rules are applied when the quality of the decision must be weighed against other factors, such as reliability of information at a given time. The acceptance rules include: Acceptance (A), Conflict (C), Fairness (F), and Shared Acceptance (SA). These rules are used to ensure that the subordinates, or agents will accept the decision made.

The leadership style depends upon the information provided, the reliability of the information, the state of the design, and type of decision. These rules imply that the goals and aspirations of the sub-ordinates are not always in step with the values set forth by the organization. Table 3 illustrates the summation of the different rules of with respect to extracted variables: the quality of the decision (QD), the information available to the leader (IA), the expertise of the leader (EL), the acceptability of the agents (AA), the similarity between agent goal and organization goal (SG), the structure of the problem (SP), the included agents (IA), and the agreement among agents (A). The filled circles indicate a strong association between the metric and the rule and the empty circles indicate a weak association between the metric and rule.

Table 3. Leadership Rules Applied to Agents

Rule	QD	IA	EL	AA	SG	SP	IA	A
LI	●	○	●					
GC	●				●			
UP	●	○	●			○	○	
A	○			●				
C	○			●			○	●
F				●			○	○
SA				●	●		○	●

Some variables describing the design state may be used in determining the application of the rules. The amount of information available in the design is a relative measure of the data available in the design against the initial data provided by the problem statement. The type of information available in the design is a comparison of the input information for the agents against the current set of information in the design. The abstraction level indicates the highest level of representation of the current design and may be a useful tool to gage the progression of the design. The number of agents affected by a possible design change is a measure of the agents that may have primary levels of relations with the changes. The desired stability of the design indicates the level of stability required of an agent in making design decisions. Other variables that might affect leadership style selection may include: the availability of agents with decision making potential, the number of leadership designed agents, the amount of available computing power dedicated to leadership driven decision making, and the amount of human interaction desired. A matrix is constructed relating the rules with the leadership styles considering the influence of the variables on the rules (Table 4).

Table 4. Leadership System Matrix

	LI	GC	UP	A	C	F	SA
Dictator	N	Y	N	N	N	N	N
Oligarchic		Y	N	N	N		N
Representative	Y		Y	Y	N	Y	Y
Democratic	Y	N	Y	Y	Y	Y	Y
Arbitrator			Y	Y	Y		Y
Hierarchic	Y	Y	Y	Y	Y		

4. PROPOSED SYSTEM

The proposed agent architecture is dependent upon the development of software wrappers for the various engineering design and analysis agents. These wrappers will provide the individual agents the capabilities discussed previously, such as the ability to record their own decision making record, other agents' decision making records, and understanding of the overall design. The diverse nature of the different agents needed in mechanical engineering design (stress analysis, tolerance analysis, cost analysis, etc.) dictates that a wrapper must be flexible to be applied to any of the agents.

In addition to wrappers, two new agents are required: a leader agent an a design state modeller agent. With the evolution of the design, the leader agent spawns a new set of leader agents with the same set of design objectives and goals. This new set of leaders forms an oligarchic set. These leaders have limited views of the design and make decisions about the provided changes by their subordinate agents. These decisions need to be negotiated with the other oligarchic leaders. The design state modeller agent is responsible for maintaining a model of the design state based upon the design environment variables. These variables derive directly from the design model, the collection of agents in the design region, and the collection of agents in the leadership region. This model is needed for evaluating the need for changes in the leadership structure. A basic architecture is presented, in which the different leadership styles must operate at varying degrees of abstraction (Figure 1).

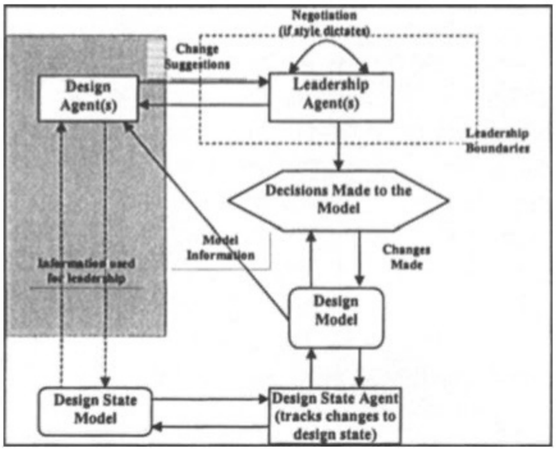


Figure 1. Agent Based Leadership Organization Architecture

There are two main regions where the design agents reside: the leadership region (dashed region at top) and the design agent congregation (shaded region at left). In the leadership region, decisions are accepted and directions chosen. Agents may migrate into this region as the state of the design changes and thus changing the leadership style. The design agents not currently in the leadership structure may suggest changes based upon the information contained in the design model. These agents have knowledge about the design state model; used to assess the leadership configuration. The design state agent tracks the changes in the design state variables creating the design state model.

5. EXAMPLE

While this proposed system is intended for complex agent congregations, a simple example may prove beneficial in illustrating the goal of the system. For this example, a typical mechanical engineering problem will be used: the design of a power transmission system. It is desired to transmit a certain amount of power, at a cost, within a contained volume, and with production capacities. This description of the problem is too abstract for many agents to act upon it. Table 5 illustrates some agents that may be activated at different stages of the design process.

Table 5. Agents for Power Transmission Design

Conceptual Design	Embodiment Design	Detailed Design
Belt design agent	Configuration agent	Assembly agent
Sprocket design agent	Costing agent	Manufacturing agent
Gear design agent	Load analysis agent	
	Parametric assignment agent	

In order to proceed with the design, some initial direction must be made. The leader agent provides the design to a complete set of design agents including belt-design agents, gear-design agents, and pump-design agents. Other design agents, manufacturing, costing, assembly, configuration, etc., do not yet recognize enough information in the system for them to activate. The system is initially dictatorial (Figure 2). As the state of the design changes, the leader agent makes decisions, which may eliminate or activate different agents. In this stage, the Goal Congruence Rule is the governing rule, indicating the need for a single Leader Agent. This state corresponds to the conceptual stage of design.

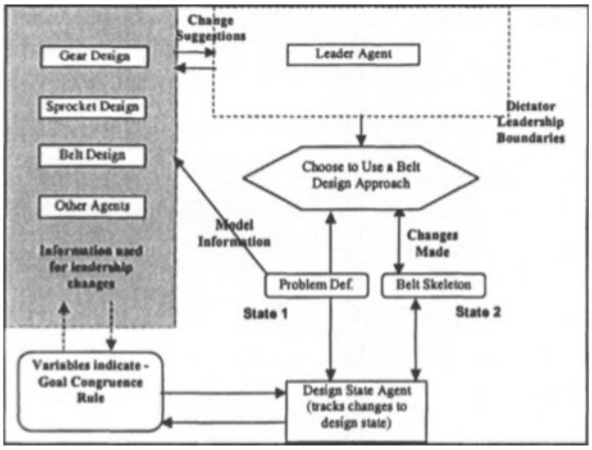

Change Suggestions
Gear Design
Leader Agent
Sprocket Design
Dictator Leadership Boundaries
Belt Design
Choose to Use a Belt Design Approach
Other Agents
Model Information
Changes Made
Information used for leadership changes
Problem Def.
Belt Skeleton
State 1
State 2
Variables indicate - Goal Congruence Rule
Design State Agent (tracks changes to design state)

Figure 2. Agent Architecture for Conceptual Stage

Eventually, the delegation of decision-making tasks may become too great for a representative form. With the design relatively stable, it may prove more beneficial to allow local issues to be determined by the agents directly affected by direct negotiation. Thus, a more democratic method of leadership may be implemented, replacing the representatives in the leadership boundary with the remaining active agents. Assuming the Shared Acceptance Rule is governing the leadership style selection, the architecture configuration for detailed design may be seen in Figure 3.

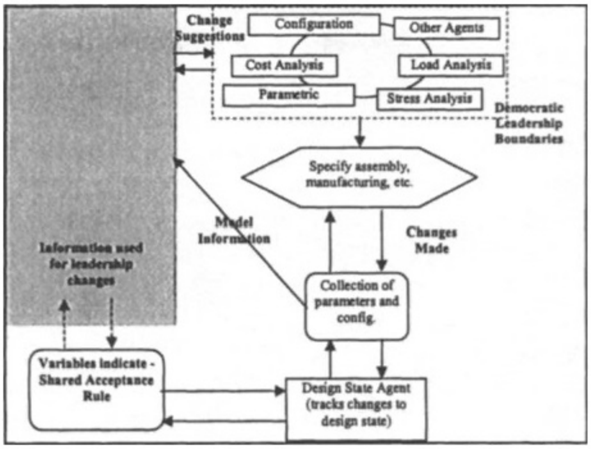

Change Suggestions
Configuration
Other Agents
Cost Analysis
Load Analysis
Parametric
Stress Analysis
Democratic Leadership Boundaries
Specify assembly, manufacturing, etc.
Model Information
Changes Made
Information used for leadership changes
Collection of parameters and config.
Variables indicate - Shared Acceptance Rule
Design State Agent (tracks changes to design state)

Figure 3. Agent Architecture at Detailed Design

6. CONCLUSIONS

This paper presented a general foundation for the issues in agent architecture at a high conceptual level. From these issues, selection of a leadership style is identified as having great influence upon a collaborative design system. The reasoning behind this is derived from the psychological and sociological studies available on organizational structure and leadership theory. A set of generalized rules is presented to aid in the determination of leadership style for different decision scenarios in addition to general agent based architecture for leadership selection. Based upon these rules, a system architecture is proposed to allow for the evolution of the leadership style.

It is important to recognize that the effort necessary for inter-agent negotiation is expensive. In the described leadership style selection method and the associated proposed architecture, the negotiation is kept in check depending upon the leadership style. There is no negotiation between agents in the earliest stages of the design where the quantity of information is low, the quality of information is low, the desired stability of the design is low, the number of agents affected by changes is very high, and the level of abstraction is high. The amount of allowable negotiation increases as the design gains stability. Eventually, the design reaches a stable point (detailed design). At this point, only the agents directly affected by various changes in the design will negotiate to generate a consensus among the entire democratic set of agents.

Considering the current state of research in agent architecture, it is believed that agent leadership organization must be addressed. The architecture presented here is primarily designed for complex agent systems consisting of many agents. A smaller system ought to be designed with the lower levels of leadership in mind, such as democratic, representative, and committee. If agents are truly social in nature, then the leadership structure of organizations as found in society must be evaluated as applied artificially and virtually.

7. REFERENCES

Azmoodeh, M. Davison, R., (1997), "Performance Management of Public ATM Networks – A Scalable and Flexible Approach", *Proc. of the IEEE – Comm. in the 21st Century*, vol. 85, no. 10, pp. 1639-45.

Barber, K., Han, D., Liu, T., McKay, R., Kim, J., Goel, A., Martin, C., (1999), "Sensible Agents in Supply Chain Management: An Example Highlighting Procurement and Production Decisions", *ASME Proceedings of DECT'99*, DETC99/CIE-9078, Las Vegas, NV.

Barron-Carro, O., Gordillo, J., (1999), "Integrating Tools for Manufacturability Analysis of Assemblies", *ASME Proceedings of 1999 DETC'99*, DETC99/DFM-8907, Las Vegas, NV.

Bazerman, M., Neale, M., (1992), *Negotiating Rationally*, The Free Press, New York.

Bercovitch, J., (1984), *Social Conflict and Third Parties – Strategies of Conflict Resolution*, Westview Press, Boulder, CO.

Cohen, P., (1999), "Rational Interaction for Multiagent Systems", http://cse.ogi.edu/CHCC/, Oregon Research Institute, Portland, OR.

Danesh, M., Yan, J., (1999), "ADN: An Agent-Based Decision Network for Concurrent Design and Manufacturing", *ASME Proceedings of DECT'99*, DETC99/DFM-8915, Las Vegas, NV.

Darr, T., Birmingham, W., (1992), "Concurrent Engineering: An Automated Design Space Approach", *Technical Report, CSE-TR-149-92*, University of Michigan, MI.

Deshmukh, A., (1999), "What is an Agent", http://farm.ecs.umass.edu, Fundamental and Applied Research in Multiagent Systems, web master, K. Leibkuchler, University of Massachusetts, Amherst, MA.

Dixon, J., Poli, C., (1995), *Engineering Design and Design for Manufacturing – A Structured Approach*, Field Stone Publishers, Conway, MA.

Eaton, P., Freuder, E., Wallace, R., (1998), "Constraints and Agents – Confronting Ignorance", *AI Magazine – Intelligent Agents, AAAI*, vol. 19, no. 2, pp. 51 – 65.

Edmonds, E., Candy, L., Jones, R., Soufi, B., (1994), "Support for Collaborative Design: Agents and Emergence", *Communications of the ACM*, vol. 37, no. 7, pp. 41-7.

Elkan, C., (1993), "The Paradoxical Success of Fuzzy Logic", *National Conference on Artificial Intelligence*, AAAI, Washington, D.C., pp. 698-703.

Epstein, J., Axtell, R., (1996), *Growing Artificial Societies*, The MIT Press, Cambridge, MA.

Gero, J., Sudweeks, F., (1991), "Artificial Intelligence in Design", *IJCAI*, University of Sudney, Australia.

Goel, A., Liu, T., Barber, K., (1996), "Conflict Resolution in Sensible Agents", *Proceedings of the International Multidisciplinary Conference on Intelligent Systems: A Semiotic Perspective*, Gaithersburg, MD, pp. 80-5.

Hazelrigg, G., (1996), *Systems Engineering: An Approach to Information Based Design*, Prentice-Hall, Upper Saddle River, NJ.

Jin, Y. Zhou, W., (1999), "Agent-Based Knowledge Management for Collaborative Engineering", *ASME Proceedings of DECT'99*, DETC99/EIM-9022, Las Vegas, NV.

Karrass, C., (1970), *The Negotiating Game*, Fitzhemry and Whiteside, Toronto, Canada.

Kniveton, B., (1989), *The Psychology of Bargaining*, Gower Publishing Company, Brookfield, VT.

Krishna, V., Ramesh, V., (1997), "Intelligent Agents for Negotiations in Market Games, Part 2: Application", *IEEE Transactions on Power Systems*, vol. 13, no. 3, pp. 1109-1114.

Lyons, K., Shooter, S., Keirouz, W., Hart, P., (1999), "The Open Assembly Design Environment: An Architecture for Design Agent Interoperability", *ASME Proceedings of DECT'99*, DETC99/DFM-8945, Las Vegas, NV.

Martin, D., Cheyer, A., Moran, D., (1999), "The Open Agent Architecture: A Framework for Building Distributed Software Systems", http://www.ai.sri.com/, Artificial Intelligence Center, SRI International, Menlo Park, CA.

Mueller, J., Pishel, M., Thiel, M., (1995), "Modelling Reactive Behavior in Vertically Layered Agent Architectures", *Intelligent Agents*, M. Wooldridge, N. Jennings (eds.), Springer Verlag, New York, NY.

Nwana, H., Ndumu, D., (1998), "A Brief Introduction to Software Agent Technology", *Agent Technology*, N. Jennings, M. Wooldridge (eds.), Springer-Verlag, New York, NY.

Orelup, M., Dixon, J., Simmons, M., (1987), "Dominic II: More Progress Towards Domain Independent Design by Iterative Redesign", *Proceedings of ASME Winter Annual Meeting*, Boston, MA.

Pahl, G., Beitz, W. (1995), *Engineering Design – A Systematic Approach*, Springer, New York, NY.

Rapoport, A., (1974), *Game Theory as a Theory of Conflict Resolution*, D. Reidel Publishing Company, Dordrecht, Holland.

Resnick, M., (1998), *Turtles, Termites, and Traffic Jams*, The MIT Press, Cambridge, MA.

Skarmeas, N., (1999), *Agents as Objects with Knowledge Base State*, Imperial College Press, River Edge, NJ.

Sundsted, T., (1999), "Agents Talking to Agents", http://www.javaworld.com/ javaworld/jw-09-1998/jw-09-howto.html?idg.net, JavaWorld.

Thomas, L., (1987), "Using Game Theory and Its Extensions to Model Conflict", *Analysing Conflict and Its Resolution – Some Mathematical Contributions*, G. Bennett, (ed.), Claredon Press, Oxford, UK.

Ullman, D., (1992), *The Mechanical Design Process*, McGraw-Hill, St. Louis, MO.

Vroom, V., Jago, A., (1978), "On the Validity of the Vroom-Yetton Model", *Journal of Applied Psychology*, American Psychological Association, vol. 63, no. 2, pp. 151-62.

Yang, X, L. Zheng, Z. Zhang, D. Liu, B. Zhang, T. Liu, (1999), "A Web Based Micro Turning-Process Planning System with Considering Distributed Resources", Proceedings of the 1999 ASME DETC, DETC99/CIE-9073, Las Vegas, NV.