FCDA: A framework for collaborative distributed multidisciplinary design

T. Khedro, M. R. Genesereth and P. M. Teicholz
Center for Integrated Facility Engineering
Standford University
Stanford CA 94305
USA

Abstract
A framework for collaborative distributed multidisciplinary design called, the Federation of Collaborative Design Agents (FCDA) is presented. This framework is an extension of an integration technology called Agent-Based Software Engineering. The focal entity of this framework is a Multiagent Design Team (MDT) that comprises a collection of Design Agents (e.g., design software with certain communication capabilities) and a Design Specialist, which together perform specific design tasks. Multiagent design teams are autonomous and form an organization structured based on a Federation Architecture. Every multiagent design team surrenders its autonomy to a system program called Facilitator, which coordinates the interaction among design agents in the federation. Facilitators can be viewed as representatives of one or more teams that facilitate the exchange of design information and knowledge in support of the design tasks they perform. In the federation, design specialists collaborate by exchanging design information with others via their design agents, and by identifying and resolving design conflicts by negotiation. In addition to a discussion of the framework's primary aspects, its realization in an integrated distributed environment for collaborative building design is described.

1 Introduction

Multidisciplinary design is a process carried out by a group of geographically-dispersed designers who are making extensive use of various types of design software to accomplish a common goal, which is the design and engineering of a viable landmark. These designers are typically specialists who concurrently perform a variety of specific design tasks and make independent design decisions. Each of the design specialists has a limited knowledge of the design environment and process, has a limited knowledge of the constraints and requirements of other design specialists, and uses special design software to perform specific design tasks.

In such a heterogeneous design environment, in which there is no single designer with a global view of the design process, distributed multidisciplinary design is an incremental process: a global design solution is arrived at by the interleaving of distributed design tasks and information exchange among designers and their software. Designers make independent decisions about some design aspects during the process and a complete design solution is formed by partial solutions being incrementally merged.

This view of the distributed multidisciplinary design process has a number of characteristics:

- The design problem is generally an under-constrained problem insofar many solutions (i.e., design alternatives) exist that satisfy the objective design requirements and design procedures.

[1]Research Assistant, Department of Civil Engineering, Stanford University, Stanford, CA 94305.
[2]Associate Professor, Department of Computer Science, Stanford University, Stanford, CA 94305.
[3]Research Professor, Department of Civil Engineering, Stanford University, Stanford, CA 94305.

These requirements and procedures are considered hard design constraints that must be satisfied and cannot be altered in order to achieve a safe and sound design.

- In performing their design tasks, designers typically satisfy their hard design constraints to ensure correct and safe design first, and then attempt to optimize the design according to their view by satisfying other criteria, which are considered soft design constraints. Soft design constraint are those constraints that could be modified during the design process and are not essential for achieving a globally-satisfiable design (e.g., design aesthetic, structural component arrangement, etc.). Soft constraints are generally subjective and typically represents a designer's limited view of the design process.

- To achieve their goals, designers concurrently and asynchronously make design decisions that are based on the performance of various design tasks, and they communicate them in an incremental fashion. Because of the limited knowledge of designers, it is possible for a designer to make decisions about shared or related design information that do not satisfy other designers' constraints.

- Design conflicts among designers arise from one designer not satisfying some constraints of other designers. In the process of conflict resolution, designers can relax their soft design constraints to reach a common acceptable design while hard design constraint cannot be relaxed.

In this paper, we first introduce the FCDA framework for collaborative multidisciplinary design. Then, we describe the realization of the framework in the collaborative building design process. Finally, we conclude the paper with a brief summary and a sketch of future work.

2 The Federation of Collaborative Design Agents Framework

In this section, we introduce the FCDA framework for collaborative distributed multidisciplinary design. This framework is an extension of an integration technology called Agent-Based Software Engineering (Genesereth 1992). At the heart of this framework, there are five components that describe our approach to collaborative design. First, we formally describe multiagent design teams and their attributes. Second, we discuss the organizational structure of multiagent design teams based on the federation architecture. We next discuss the communication of information and knowledge among design agents. Then we describe the coordination among multiagent design teams. Finally, we present our collaboration strategy for collaborative multidisciplinary design.

2.1 Multiagent Design Teams

In the FCDA framework, a multiagent design team is defined as a collection of sophisticated design software, called design agents, and a human design specialist who together perform specific tasks of the design process. In a multiagent team, the design specialist is responsible for performing specific design tasks by interacting with his/her design agents and for making various design decisions. Each multiagent design team has its own autonomy, which signifies its independence from other multiagent design teams in the environment. From the perspective of a design specialist, the specialist's design agents are not autonomous and operate under his/her control. Design agents can be viewed as the specialist's assistants in performing some design tasks. From the perspective of design agents that belong to the same or different multiagent design teams, other design agents are autonomous since all agents' operations are independent.

2.1.1 Design Agents

A design agent is a sophisticated design software that performs specific design tasks and communicates its design information and knowledge in a standard language called Agent Communication Language (ACL) according to a set of specifications that characterize its

activities. In addition, a design agent possesses other important capabilities, which allow its design specialist to exchange and negotiate design information and changes of design tasks as well as design decisions with other design specialists in other teams.
In a multiagent design team, the specifications of all of the team's design agents collectively characterize the role of the team in the environment. The specifications of a design agent, which describe the agent's role , consist of the following:

- *Interests:* A design agent's interests consist of the design information the agent desires to receive. These interests are specified in terms of the ACL messages, which precisely specify the design entities and the attributes in which the agent is interested. Once the interests of a design agent are declared, it will appropriately receive design information when this information is generated by any other design agent.

- *Perspectives:* A design agent's perspectives specify the attributes and formats in which the design agent desires to receive and send design information. The idea of perspectives is related to the different views of the same design information various design agents may have. The use of perspectives gives different design agents flexibility in expressing their view of the design information.

- *Behaviors:* Behaviors of a design agent describe the agent's activities in a multiagent design team. One important aspect of an agent's behaviors is the type of information the design agent generates or communicates when it receives some other design information. This partly characterizes some of the tasks the design agent performs.

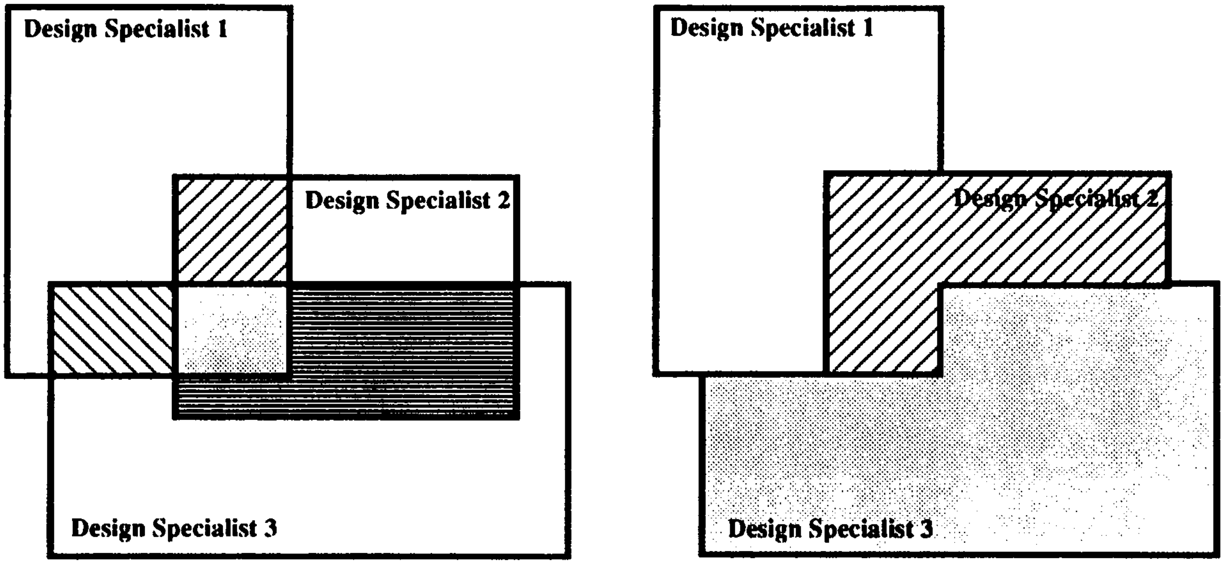
2.1.2 Design Specialists

A design specialist is a human designer who is responsible for ensuring that his/her part of the design requirements and constraints in the design process are met by his/her performance of specific design tasks and appropriate design decisions. During the design process, design specialists, who belong to different multiagent design teams, exercise different levels of authority over the design tasks they perform and the design requirements they must satisfy. Typically, a design specialist exercises authority by making design decisions over the design information in which he/she is interested. Generally, there are a number of design specialists who share design information over which they make decisions. In the FCDA framework, the interests and authority of a design specialist are defined as follows:

- *Interests:* The interests of a design specialists consist of design information he/she would like to know about. In essence, the interests of a design specialist are the combined interests of all of the specialist's design agents.

- *Authority:* A design specialist's authority over some design information is the ability to make design decisions about this information. A final design decision is a decision that concludes a disagreement or conflict among design specialists. An important aspect of the FCDA framework is that the authorities of different design specialists are disjoint despite the possible overlapping of design specialists' interests. In other words, there is always one design specialist who can make a final design decision about some design information.

Figure 1 shows design information in which three design specialists are interested and the specialists' authorities. While the interests of the three specialists are overlapping, it is apparent that the authorities of the three specialists do not overlap. As will be discussed later, this is an important aspect of our framework, which is related to the adopted conflict resolution strategy.



(a) Overlapping of Design Specialists' Interests *(b) Authority of Design Specialists*

Figure 1 - Design Specialists' Interests and Authorities

2.2 Organizational Structure of Design Specialists and Design Agents

A group of multiagent design teams comprise an organization whose purpose is to perform some set of design tasks in an attempt to achieve a common goal, which is the design of an artifact satisfying certain requirements. In performing their design tasks, multiagent design teams have to interact according to their organizational structure while obeying some behavioral constraints (Pattison 1987). The organizational structure defines the ways in which multiagent design teams are divided to perform design tasks and coordination is achieved (Mintzberg 1979). It specifies how different teams should be connected to communicate in a way that maximizes the efficiency of communication. Behavioral constraints are those constraints that determine how different teams can communicate and respond to each other in the organization. It is important to note that behavioral constraints are typically based on the strategy adopted by the organization to accomplish its goals to the best of its interests. For this reason, behavioral constraints on the process by which the organization's goals are achieved emphasize tradeoffs between such features as topicality, production costs, robustness, completeness, and quality (Pattison 1987). For example, in the fast-track building design process, we may insist that the strategy produce high quality design as quickly as possible, thus emphasizing maximal values for topicality (i.e., short response time) and quality (i.e., functionally, behaviorally, and aesthetically satisfying solutions), at the expense, perhaps, of construction costs.

In the FCDA Framework, the organization of multiagent design teams is a distributed design environment where design specialists and design agents of different teams are geographically dispersed. In this environment, all design agents are integrated in a Federation Architecture on which the organizational structure of the environment is based. As multiagent design teams interact with each other in the design process, they obey certain behavioral constraints. It is important to note that behavioral constraints are different from design agents' behaviors, which are a part of design agents' specifications.

2.2.1 Federation Architecture

The federation architecture describes the way design agents communicate in an integrated environment. In this architecture, design agents, which are autonomous, interact via special agents called Facilitators. Facilitator are system programs that employ domain-independent automated reasoning mechanisms for facilitating and coordinating the interaction of design agents during the design process. Figure 2 illustrates an example of a federation architecture in which there are three facilitators, one with two design agents and two with three design agents apiece. Those eight design agents belong to four different multiagent design teams. In this architecture, design agents communicate only with their local facilitators, and facilitators communicate with each other. In effect, the agents form a "federation" in which they surrender their communication autonomy to the facilitators, hence, the name of the architecture (Genesereth 1992). One of the primary advantages of the federation architecture is improved flexibility in integrating design agents, which allows different agents not to know about other existing design agents in the environment.

Facilitators represent special types of design agents whose roles are to facilitate and coordinate the interaction among multiagent design teams by coordinating the interaction of design agents within an single team as well as across different teams. Facilitators' specifications are the union of all specifications of all design agents to which they are connected. From the perspective of other facilitators, a facilitator can be viewed as a representative of the design agents working as a group on closely-related tasks of a common design. For this reason, it is important for a group of design agents whose interests are closely related to be connected to a single facilitator. The federation architecture provides a basis for structuring design agents according to their interests in the way that provides the most effective communication.

2.2.2 Formal Behavioral Constraints

Formal behavioral constraints can be viewed as organizational regulations that help the organization achieve its goals according to its strategy. Although design agents enjoy autonomy in the federation architecture, their behavior is constrained for the sake of effectiveness in achieving their goals. In the FCDA framework, multiagent design teams yield to the following formal behavioral constraints:

- Design specialists have to register their design agents. In the registration phase, a design agent sends its specifications to the facilitator so the agent can be informed about the design information of interests to it. Once a design agent is registered, it has to be registered continuously until the design is complete. It is important to note that being registered does not imply continuous design work by a multiagent design team. As will be discussed later, facilitators allow registered design agents to be inactive while still maintaining their registration.

- When a design agent receives design information, the information must be reviewed by the design specialist, who then decides on the appropriate response.

- Once a design specialist makes a design decision and commits design information or changes locally to his/her design agent, the specialist has to broadcast the information by sending it to the agent's facilitator.

- A design specialist respects the authority of other specialists and makes final design decisions only about the design information he/she has authority over.

2.3 Communication of Design Information and Knowledge

Communication is perhaps the most important aspect in organizations. It has been argued that communication is essential for cooperation in the quest for achieving an organization's goals (Werner 1989). The question that is often asked is what sort of communication is needed to achieve cooperation? What communication language and protocol are required in order for design agents to communicate cooperatively? Equally important are the details exchanged during communication.

In the FCDA framework, design agents communicate via a message sending and receiving protocol called Agent Communication Language (ACL). ACL is a language proposed for a communication standard among disparate software. It provides a unified format by which design agents unambiguously exchange complex expressions describing various design aspects. This language follows the recommendations of the various committees involved in the Knowledge Sharing Initiative sponsored by Darpa (Neches 1991). In the federation architecture, communication among design agents is undirected. Design agents communicate their design information in the form of messages without specifying destinations or addresses for those messages. It is the responsibility of the facilitators to determine the appropriate recipient agents of the messages and forward the messages accordingly.

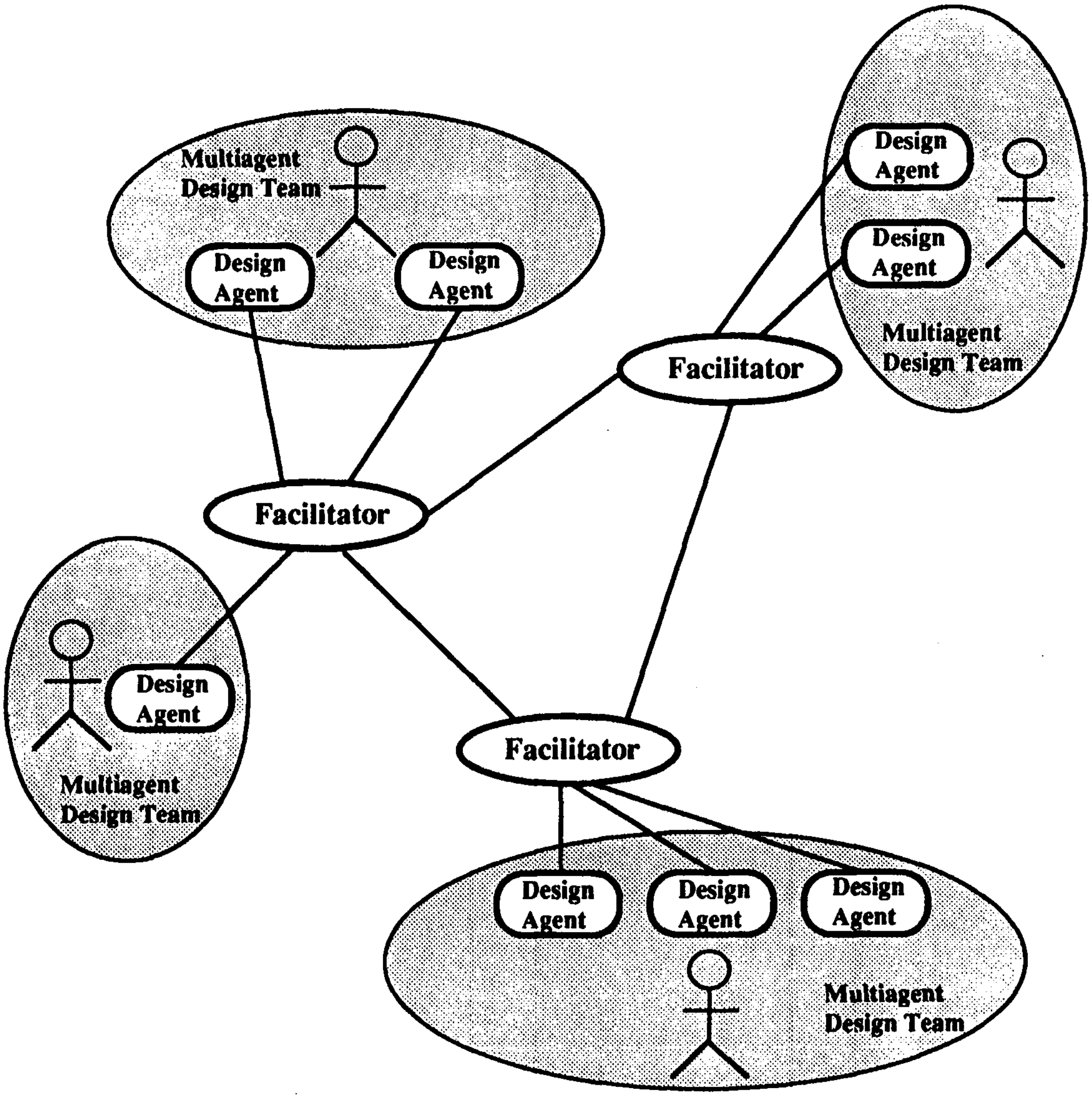


Figure 2 - Organization of Multiagent Design Teams in a Federation Architecture

2.3.1 Agent Communication Language

ACL defines a message sending and receiving language that is based on a logic-based language called Knowledge Interchange Format (KIF) and a message protocol called Knowledge Query Manipulation Language (KQML). This language provides for the communication of design information and changes partially and incrementally. Each message in ACL consists of a KQML message type and a KIF sentence enclosed in parentheses. For example, consider the following ACL message:

```
(assert (= (width obj1) 12))
```

In this ACL message, `assert` is the message type that communicates the addition of the declarative KIF sentence `(= (width obj1) 12)`. This KIF sentence is expressing that the width of the object `obj1` is equal to `12` inches.

- *Message Types:* In addition to communicating design information, the KQML message types indicate different communication modes (e.g., declarative, interrogative, imperative) (Finin 1991). A `retract` message is used to communicate a retraction of a declarative sentence. An `ask` message is used to query whether or not a sentence is true, and an `unask` message is used to cancel a query. An `achieve` message signals that the specified sentence is to be made true, and a `forget` message signals that a specified sentence is no longer a goal. Finally, there is a `reply` message for use in transmitting answers to previous messages. In addition to these basic messages, there are numerous other special cases.

- *Knowledge Interchange Format:* KIF is intended for the interchange of knowledge among agents. It has declarative semantics, is logically comprehensive, and provides for the representation of meta-knowledge (Genesereth 1990). This language is a prefix version of first-order predicate logic, with various extensions to enhance its expressiveness. It is far more expressive than many other languages used for communication standards. It provides for the communication of constraints, negations, disjunctions, rules, quantified expressions, and so forth.

- *Design Vocabulary:* In order for a group of design agents to communicate about a design domain, they must consistently use those words that refer to the objects, functions, and relations (e.g., various types of product design components, geometry and attributes of components, relationships among components, etc.) appropriate to the design domain disciplines. These words, which constitute a design vocabulary expressed in KIF as object constants, function constants, and relation constants, will have precise meanings in the particular design domain under consideration. Using this design vocabulary, design agents can exchange design information and knowledge consistently (Khedro 1993). It is important to note that it is possible to have different vocabularies used different disciplines of design domain. This necessitates vocabulary translation among communicating design agents using the different vocabularies, as will be highlighted later.

2.3.2 Design Agent Communication Protocol

Design agents possess special communication capabilities that allow them to communicate dynamically with their facilitators. Typically there are two phases in the operation of a design agent: its registration phase and its normal operation of message sending and receiving.

- *Agent Registration:* In agent registration, a design agent signals a connection request to the facilitator. Once its connection request is granted by facilitator, the design agent establishes a communication channel and sends its name and specifications so that it becomes identifiable by the facilitator in the environment.

- *Message Sending:* Once the design specialist instructs his/her design agent to communicate design information or changes he/she made, the design agent sends its design information in as a sequence of ACL messages through its communication channel with the facilitator.

- *Message Receiving:* As it performs its design tasks, a design agent has the ability to monitor its communication channel with a facilitator to check for arriving messages. Upon receiving messages, a design agent informs the design specialist of the arriving messages, upon which the specialist can decide to have the information received and propagated.

2.4 Coordination of Design Agents

Generally, coordination is viewed as the act of managing interdependencies among various types of activities in an organization (Malone 1991). In our research, coordination is considered to be the effort of facilitation the exchange of design information among design agents in a way that supports their performing their design tasks in a timely manner. This facilitation requires capturing design knowledge to reason about various types of design tasks and their relationships.

In the FCDA framework, facilitators coordinate the interaction of design agents by receiving messages and distributing them to design agents according to their interests, by scheduling agent activities, and by performing different types of vocabulary translations. Most of these functions performed by the facilitator are based on agents' specifications communicated by the design agents at the time of registration. Other types of vocabulary translation are performed to facilitate the execution of design tasks performed by autonomous design agents. This translation is done by the capture domain-specific knowledge about different multidisciplinary design tasks and their interrelationships in the form of KIF theories. A KIF theory is a collection of a special form of KIF sentences called axioms, which express various aspects of design knowledge. Provided with these KIF theories, the facilitator performs reasoning to infer implicit design information received from design agents and expresses the information explicitly for the use of other agents. The facilitator also reasons about the relationships among different levels of design abstractions and different design models describing the same physical entity. In this way, the facilitator is capable of identifying appropriate design changes and of forwarding them to appropriate design agents.

The operation of the facilitator in an environment is continuous. At each point of its operation, the facilitator checks for agent registration and for messages with already registered agents. When the facilitator detects that an agent is requesting registration, it executes its registration protocol for accepting this registration. Every message received by the facilitator is processed in an automated fashion, which can result in the forwarding of a number of appropriate messages. In processing a message, the facilitator first identifies the message type and then performs a perspective translation according to the design agent's perspective of the information contained within the message. Next, the facilitator reasons about this information to infer from it appropriate other information. Then, the facilitator attempts to find the agents interested in the information and message type. Once all interested design agents are identified, the facilitator prioritizes the order in which messages will be forwarded to the interested agents. This scheduling of message sending is based on the design agents' behaviors. Finally, in the order determined, the facilitator performs a perspective translation for every design agent and forwards appropriate messages to it.

2.5 Collaboration of Multiagent Design Teams

Collaboration is viewed as the act of individuals cooperating in solving a common complex problem because of their self-interest (Durfee 1987a). Collaboration takes on many different forms, but it is important to realize that individuals collaborate when it is for their mutual advantage. In a multidisciplinary distributed design environment, design specialists who belong to different multiagent design teams communicate design information and changes via their design agents to perform various design tasks to achieve their common goals. When communicating information via a design agent, a specialist expects any information forwarded to other agents in his/her team. However, information received by design agents in other teams may not be expected by their design specialists. It is often the case that information received by other design specialists partly violate some of their design constraints and results in a design conflict.

In addition to communication, the essence of collaboration is, in our view, how design specialists go about identifying and resolving their design conflicts. In the section below, we describe how design conflicts are detected and resolved via a negotiation strategy that minimizes backtracking to previous design decisions during the design process.

2.5.1 Detection of Design Conflicts

A design conflict arises from a design decision made by a design specialist based on his/her limited view of the design requirements and design process. When a design specialist in a multiagent design team makes a design decision and communicates the new design information or changes, other design specialists in other teams receive appropriate design information as a result. The design information newly received by a design specialist causes a design conflict when it does not satisfy some of the hard or soft design constraints the recipient design specialist wants to satisfy. This is when a design conflict is detected.

Design conflicts, in our framework, are considered to be of two types: critical and non-critical. Critical design conflicts are those design conflicts that results from design information that violates hard.design constraints. Non-critical design conflicts are those conflicts that result from design information that violates the soft design constraints design specialists desire to satisfy as they optimize their design solution locally.

2.5.2 Resolution of Design Conflicts

Conflict resolution is an essential requirement for collaboration in autonomous, intelligent, interacting systems (Adler 1989). In planning, much of the research on conflict resolutions has focused on strategies that involve intelligent, interacting systems exchanging information about their goals and plans in order to resolve their conflicts and generate a globally acceptable plan (Durfee 1987b, Jin 1990). In addition, the role of negotiation has been emphasized as the focal point for conflict resolution in distributed problem solving for different domains including concurrent multidisciplinary engineering (Laasri 1990, Levitt 1991, Werkman 1990).

Our research work has focused on developing a strategy for resolving design conflicts that includes negotiation and minimizes backtracking to previous design decisions. In this strategy, conflict resolution is carried out by the design specialists with the help of their design agents. Depending on the type of design conflict (i.e., critical or non-critical), on who has authority over the design information that causes the conflict, and on who detects the design conflict, a negotiation takes place in an attempt to resolve conflicts. In the strategy, critical design conflicts are always resolved because they result from the violation of hard design constraints, which must be satisfied in order have a satisfiable design. Non-critical conflicts, however, are not always resolved. Typically, one of the design specialists needs to relax his/her violated soft design constraints in order for an agreement to be reached.

2.5.3 Negotiation

Negotiation is a process of resolving conflicts between two conflicting, intelligent systems in which they attempt to develop or refine their plans jointly so that the goals of each are satisfied. In the course of negotiation, intelligent, interacting systems typically develop understanding of each other at the goal level, and thus can find complex solutions involving trade-offs and novel approaches to solving shared problems that neither could have recognized independently (Adler 1989).

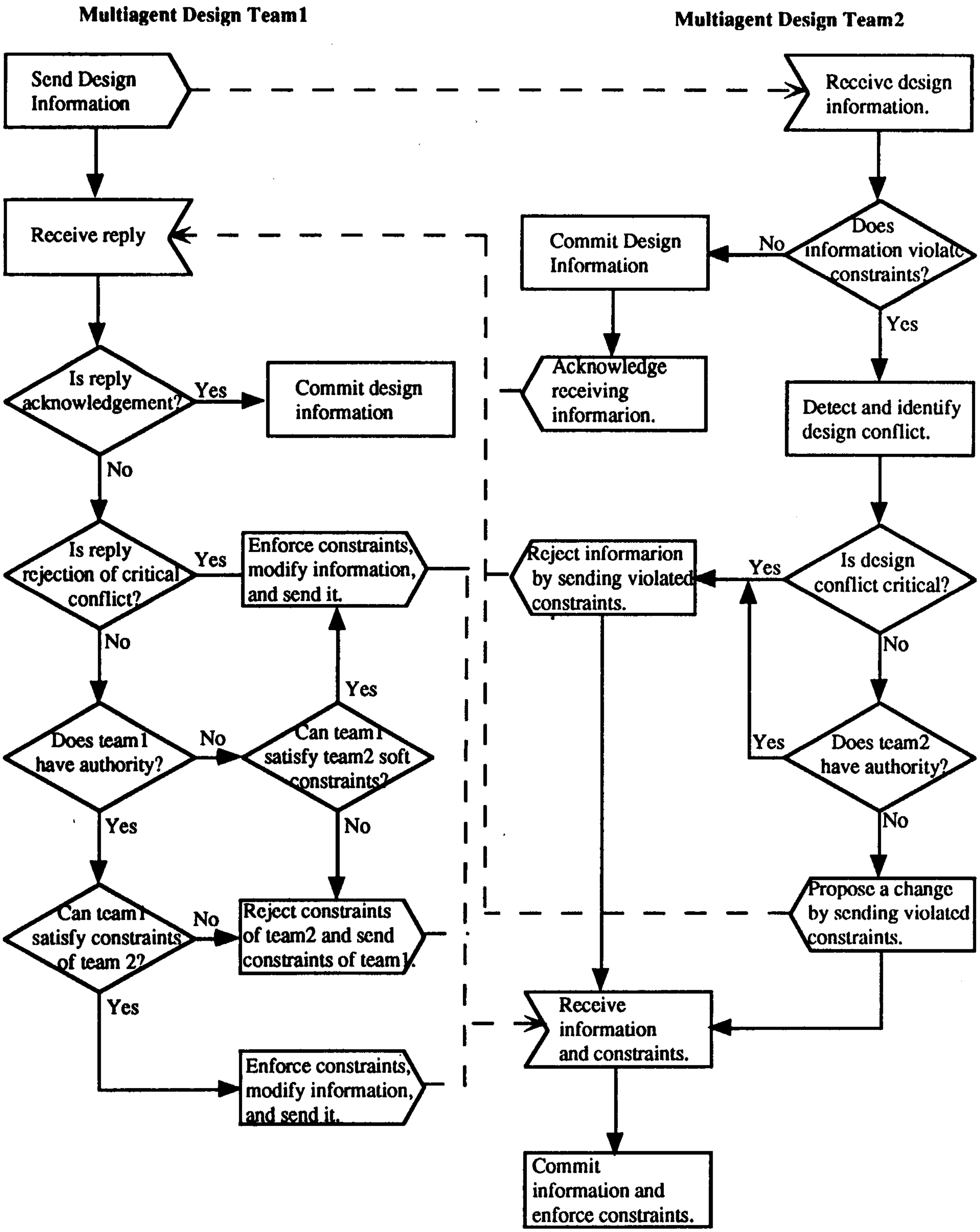


Figure 3 - A Flowchart for Conflict Detection and Resolution by Negotiation

In our framework, negotiation takes place among multiagent design teams by the teams communicating design constraints that have resulted in design conflicts. Figure 3 shows a flowchart that illustrates the process of negotiation between two multiagent design teams. The dashed arrows indicate indirect communications (via facilitators) between the two teams. Negotiation starts after Team 2 detects a design conflict as a result of receiving design information that violates some design constraints. If the design conflict is critical, then Team 2 rejects the received information by sending the hard design constraints that have been violated. In cases in which the design conflict is not critical, but Team 2 has authority over the information received, Team 2 can still reject the received information by sending the violated soft constraints. If the design conflict is not critical and Team 2 does not have authority, Team 2 can only propose a modification to the received design information by sending the violated soft constraints.

After sending the design information, Team 1 expects a reply if any of this information has been forwarded to other teams by the facilitators. Once a reply is received, Team 1 examines whether the reply is an acceptance of the design information. If it is, Team 1 commits the design information locally. Otherwise, Team 1 checks what conflict type has been sent by Team 2. If it is a critical conflict, Team 1 modifies the design information in a way that satisfies the hard design constraints received from the other team and sends the modifications. If the conflict is non-critical and Team 1 does not have authority, it modifies the information to satisfy the received soft constraints only if it does not thereby violate its own hard constraints. Otherwise, it will reject the soft constraints received from Team 2. When the conflict is non-critical, but Team 1 has authority, Team 1 attempts to satisfy the other team's soft constraints while still satisfying its own hard and soft constraints. If Team 1 succeeds in doing so, it will modify the information. Otherwise, it will reject the soft design constraints proposed by Team 2. This process of negotiation ends when Team 2 receives the final reply from Team 1, which typically includes either modifications to the design information or constraints of Team 1 that have to be satisfied by Team 2.

3 An Environment for Collaborative Building Design

In parallel with studying the fundamental research aspects of the FCDA framework, we have been developing an integrated distributed environment for collaborative building design. We have developed this environment by integrating eight systems that perform various aspects of design and planning of buildings. Most of these systems have been implemented at different times by different people, and there has been no communication among their developers. The systems have been adapted to be design agents by the implementation of shell programs that added agents' capabilities. As a result, these systems have been enabled to register with the facilitator and to communicate in ACL using appropriate design vocabularies particular to the system's design task. In this section, we briefly describe the environment architecture and highlight its capabilities.

3.1 Environment Systems and Architecture

The integrated design environment consists of the four design systems ARCH-CIFECAD, BERT, FOOTER, and CIFECAD, of a planning system, OARPLAN, a scheduling system, Scheduler, a cost estimating system, Estimator, and a database management system, ORACLE. Figure 4 shows the organization of the environment as five multiagent design teams and shows how these systems are connected in a federation architecture.

ARCH-CIFECAD is an extension of the Autocad system that is particularly intended to support the architectural schematic design of buildings by providing an object model with a library of architectural building components that facilitates the design in a 3-D environment. BERT is a structural design system that performs conceptual structural design for buildings and generates a steel structural system (Fuyama 1992). FOOTER is a system that performs design for reinforced concrete footings for buildings. CIFECAD is a similar extension of Autocad with a library of structural building components. OARPLAN is a planning system that generates project plans based on facility description (Darwiche 1989). OARPLAN needs a description of the structural system that is to be constructed in terms of the system's physical components. Scheduler is a scheduling system that generates necessary construction operations based on a high-level plan and personnel and equipment allocations (Intellicorp 1990). Estimator is a system that produces cost estimates for steel structural systems and reinforced concrete footings.

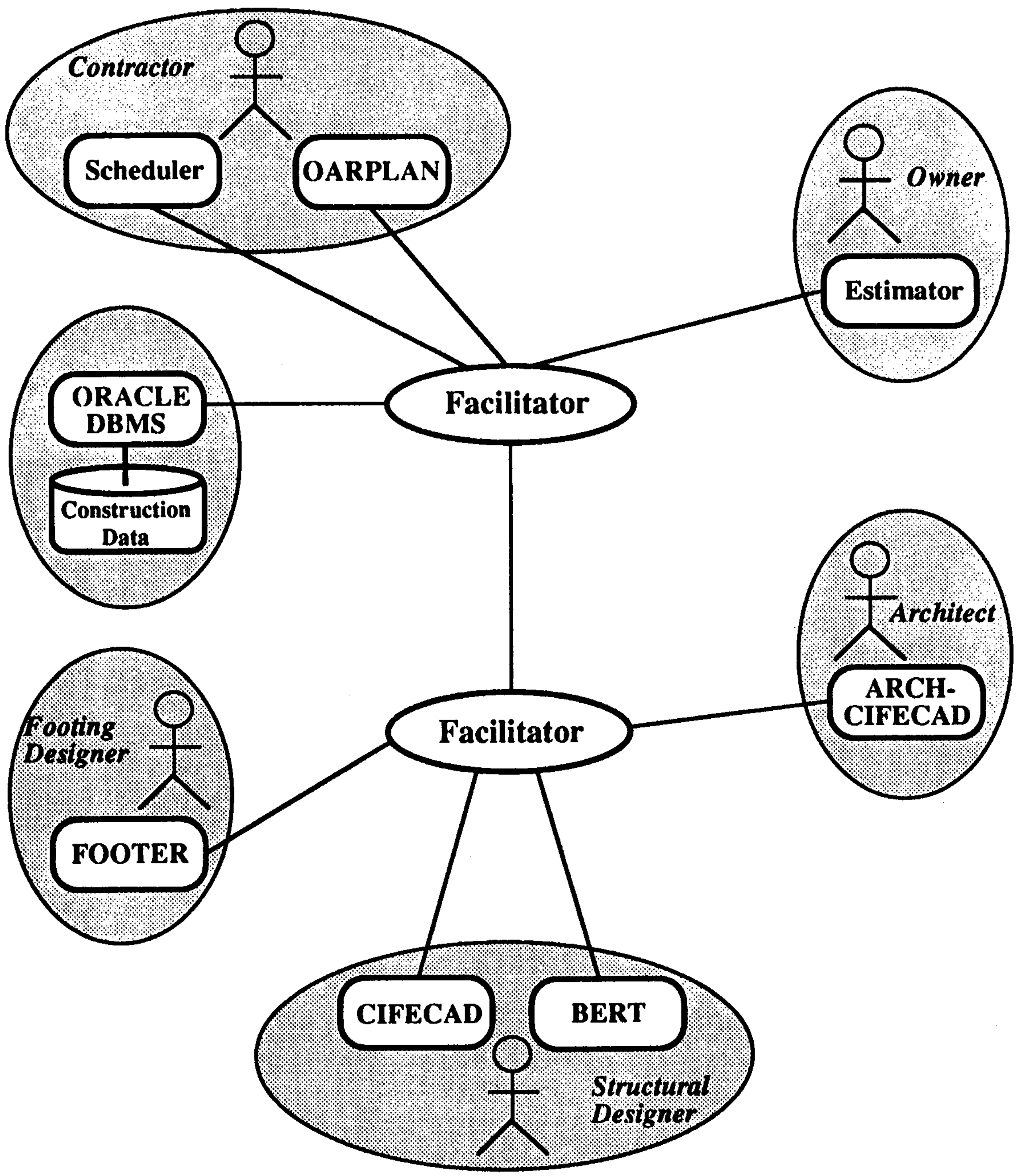


Figure 4 - Organization of Multiagent Building Design Teams

The eight systems and the two facilitators of the integrated environment run on seven different computers: two DEC workstations, three Sun workstations, and two Macintoshes. Figure 5 illustrates the implementation architecture.

3.2 Environment Capabilities

In the environment, five design specialists can perform various design tasks concurrently and exchange design information as well as changes. The process of building design typically starts with the architectural schematic design performed by the architect using ARCH-CIFECAD. Once the architect communicates the design information to the facilitator, the structural designer receives general information about the building within BERT. Using BERT, the structural designer designs the building's structural system by selecting design sections for all structural components. When the structural designer communicates this information, CIFECAD receives the geometric design information and displays the structural system in a 3-D mode. Meanwhile, the footing designer receives appropriate information about the first-story columns, including the loads their are resisting. Using FOOTER, the footing designer designs for reinforced concrete footings. At the same time, the contractor receives information about the relationships among various structural components in OARPLAN and, using Scheduler, generates a construction plan and then a schedule for construction operation. In parallel with the footing design and construction planning and scheduling activities, the owner, using Estimator, receives the necessary information about the structural component design sections. After obtaining the labor and material unit costs from the contractor's database via the ORACLE DBMS, he/she can produce a cost estimate for the structural system.

When footing design information is communicated, both the contractor and owner multiagent design teams get updated. As a result, a new cost estimate is produced and a new plan and schedule are generated. During the process, as design specialists make design changes and communicate them, the appropriate teams are informed, and necessary design and planning tasks can be performed again.

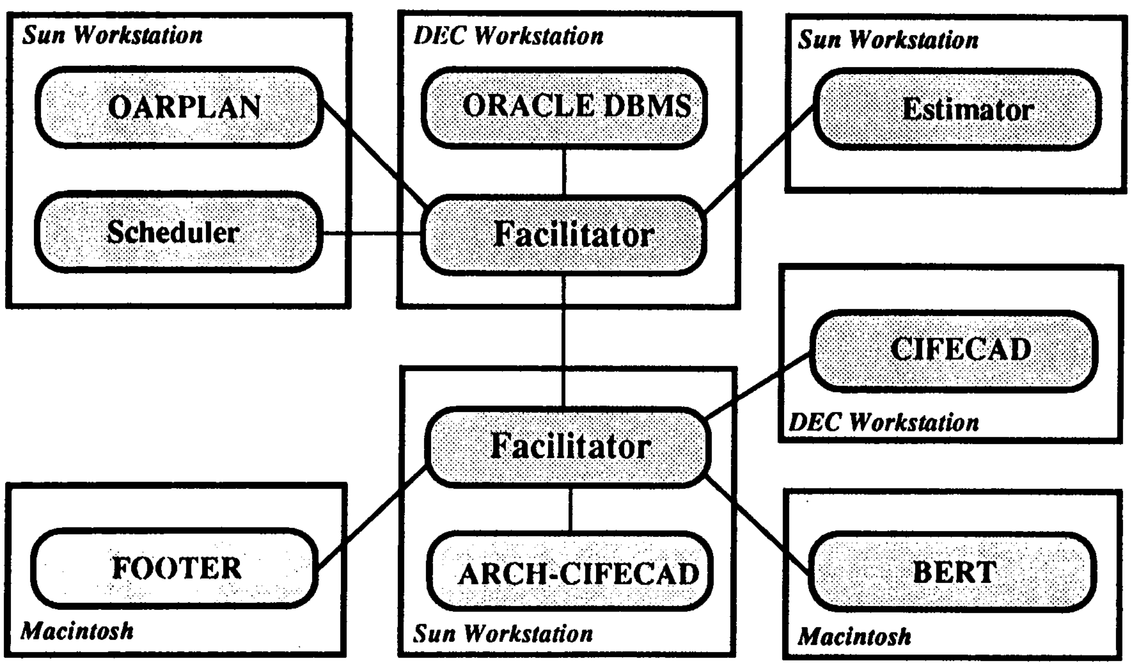


Figure 5 - Implementation Architecture of Environment

4 Summary and Future Work

This paper has discussed a framework for collaborative distributed multidisciplinary design, called the Federation of Collaborative Design Agents (FCDA). The framework is based on the notion of multiagent design teams as groups of design specialists and design agents, the latter of which are design software that communicates in a standard language called Agent Communication Language (ACL) and have certain capabilities. Multiagent design teams are structured based on a federation architecture in which design agents interact via system programs called facilitators and obey formal behavioral constraints. In the federation architecture, communication among design agents takes place through ACL messages. Coordination among design agents is achieved by facilitators that perform a variety of functions to facilitate the exchange of design information in support of their design tasks. Collaboration among design specialists who belong to different multiagent design teams is achieved by the exchange of design information and changes as well as by the detection and resolution of design conflicts in a particular negotiation strategy, which minimizes backtracking to previous design decisions.

The current realization of this framework, the integrated environment for collaborative building design, demonstrates various aspects of design agent communication and facilitators' coordination. The realization of design specialists' collaboration is our next step. In this effort, we need to identify appropriate techniques for allowing design agents to assist design specialist in detecting design conflict, in locally managing design information and changes, and in enforcing and communicating design constraints. In addition, we are exploring effective strategies for resolving design conflicts that are the result of concurrent changes.

5 Acknowledgments

This research has been supported by the Center for Integrated Facility Engineering at Stanford University. The support is gratefully acknowledged.

6 References

Adler, M. R. Davis, A. B. Weihmayer, R. and Worrest, R. W. (1989). Conflict-Resolution Strategies for Non-hierarchical Distributed Agents. Research Notes in Artificial Intelligence, Distributed Artificial Intelligence, *Morgan Kaufmann Publishers, Inc.*

Darwiche, A., Levitt, R. E., and Hayes-Roth, B. (1989). OARPLAN: Generating Project Plans by Reasoning about Objects, Actions, and Resources. *AI EDAM*, 2.

Durfee, E. H., Lesser, V. R., and Corkill, D. D. (1987a). Cooperation Through Communication in a Distributed Problem Solving Network. Research Notes in Artificial Intelligence, Distributed Artificial Intelligence, *Morgan Kaufmann Publishers, Inc.*

Durfee, E. H. and Lesser, V. R. (1987b). Using Partial Global Plans to Coordinate Distributed Problem Solvers. *Proceedings of the Tenth IJCAI*, 857-883.

Finin, T. and Wiederhold, G. (1991). An Overview of KQML: A Knowledge Query Manipulation Language. Research Report, *Department of Computer Science, Stanford University, Stanford, CA.*

Fuyama, H. (1992). Computer Assisted Conceptual Steel Design of Steel Buildings. *Department of Civil Engineering, Stanford University, Stanford, CA*, PhD Dissertation.

Jin, Y. and Koyama, T. (1990). Multiagent Planning Through Expectation Based Negotiation. *Proceedings of the 10th AAAI International Workshop on Distributed Artificial Intelligence.*

Genesereth, M. R. et al. (1990). Knowledge Interchange Format. *Department of Computer Science, Stanford University, Stanford, CA*, Report Logic-90-4.

Genesereth, M. R. (1992). An Agent-Based Framework for Software Interoperability. *Proceedings of Darpa Software Technology Conference*, 359-366.

IntelliCorp, Inc. (1990). Scheduler Booster Module: User and Reference Manual.

Khedro, T., Genesereth, M. R., and Teicholz, P. M. (1993). Agent-Based Framework for Integrated Facility Engineering. *Engineering with Computers*, in press.

Laasri, B. Laasri, H. amd Lesser, V. R. (1990). Negotiation and Its Role in Cooperative Distributed Problem Solving. *Proceedings of the Tenth AAAI International Workshop on Distributed Artificial Intelligence.*

Levitt, R. E., Jin, Y., and Dym, C. L. (1991). Knowledge-Based Support for Management of Concurrent, Multidisciplinary Design. *AIEDAM*, **5**:77-95.

Malone, T. W. and Crowston, K. (1991). Towards an Interdisciplinary Theory of Coordination. MIT Sloan School Working Paper #3294-91-MSA, *Massachusetts Institute of Technology, Cambridge, MA.*

Mintzberg, H. (1979). The Structure of Organizations. *Prentice-Hall Inc.*

Neches, R., et al. (1991). Enabling Technology for Knowledge Sharing. *Artificial Intelligence Magazine*, **12**.

Pattison, H. E., Corkill, D. D., and Lesser, V. R. (1987). Instantiating Descriptions of Organizational Structures. Research Notes in Artificial Intelligence, Distributed Artificial Intelligence, *Morgan Kaufmann Publishers, Inc.*

Werkman, K. (1990). Knowledge-Based Model of Negotiation using Shareable Perspectives. *Proceedings of the Tenth International Workshop on Distributed Artificial Intelligence.*

Werner, E. (1989). Cooperating Agents: A Unified Theory of Communication and Social Structure. Research Notes in Artificial Intelligence, Distributed Artificial Intelligence, *Morgan Kaufmann Publishers, Inc.*