Cooperative Negotiation in Concurrent Engineering Design

Katia P. Sycara,
The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Design can be modeled as a cooperative multi-agent problem solving task where different agents possess different knowledge and evaluation criteria. These differences may result in inconsistent design decisions and conflicts that have to be resolved during design. The process by which resolution of inconsistencies is achieved in order to arrive at a coherent set of design decisions is negotiation. In this paper, we discuss some of the characteristics of design which make it a very challenging domain for investigating negotiation techniques. We propose a negotiation model that incorporates accessing information in existing designs, communication of design rationale and criticisms of design decisions, as well as design modifications based on constraint relaxation and comparison of utilities. The model captures the dynamic interactions of the cooperating agents during negotiations. We also present representational structures of the expertise of the various agents and a communication protocol that supports multi-agent negotiation.

1. Introduction

Issues of industrial productivity are of major economic significance. To attain substantial improvements in industrial productivity the following capabilities are crucial:

- rapid reaction to changes in functional requirements of products and to new technological opportunities
- rapid transition from design concept to product
- production of high quality products at the lowest possible life-cycle (design, manufacturing, testing, operation, maintenance) cost

In order to attain these capabilities, improvements in effectiveness and speed of design are needed. Recent developments in computer science, especially in AI, large scale modeling and simulation, and information systems provide us with unique opportunities to push beyond the present level of computer-aided automation technology and to attain fundamental improvements in industrial productivity. The vision of the next generation of industrial automation technology includes:

- Computer systems for conceptual design and for high level design processes, where a product is designed for functionality, manufacturability, maintainability and economy.
- An infrastructure including design knowledge bases for products and processes that would be widely accessible to researchers and practitioners.

Besides being of major economic significance, design is a challenging task that (1) presents many opportunities for significant scientific breakthroughs in many areas of experimental computer science (e.g., AI, Numerical Methods, Human-Machine Interface), (2) can stimulate links among these computer science areas, and (3) can stimulate links among computer science and other disciplines (e.g., mechanical, electrical, chemical engineering).

The computer field is now at a point where it can provide the intellectual foundations and the technical basis for developing a science and technology of design and manufacturing that will have a dramatic impact on industrial productivity. Furthermore, by building on top of the present state of computing, and by further accelerating research and development in areas of advanced computing which can contribute to improvements in design and manufacturing (AI in particular), we can bring about major gains in industrial productivity. Our challenge is to:

- Develop computer systems that will efficiently generate solutions (autonomously or as design assistants) to a broad range of design problems.
- Represent relevant domain and control knowledge that can be readily incorporated in these systems.
- Formulate specific design problems in a manner that can lead to their efficient solution by these systems.

1.1. Design

Design is the act of devising an artifact which satisfies a useful need, in other words, performs some function. Since our world is full of artifacts, design activity is pervasive. Underlying design tasks is a core set of principles, rules, laws and techniques which the designer uses for problem solving. His expertise lies in his ability to use these techniques to produce a feasible design. The designer's expertise is a consequence of his experience and training, much of which is based on previous exposure to similar design problems. This is particularly true in our domain of interest: engineering design [Pahl & Beitz 84, Gregory 88, Hicks 87].

Typically, the input to the design process is a set of specifications in terms of goals and constraints of the desired artifact. The designer's task is to transform these specifications into a structural description of the artifact in a given language of design descriptions. There are three types of knowledge that are used in design:

1. Knowledge about the domain (e.g., theories, and models)
2. Design descriptions in terms of a language of designs
3. Design specifications

Acquiring and representing this knowledge so that it can effectively guide design generation is a central concern of design research. Typically, a designer reasons in two spaces: the space of design specifications and the space of design structures. To facilitate the automated generation of designs, we must strive to represent design structures, so that parts of the artifact correspond as directly as possible to parts of specifications.

Typically, design problems have multiple interacting goals, and/or complex systems of constraints. To handle the complexity of goal interaction, a design system should be able to decompose design problems into loosely coupled subproblems in order to handle subproblem interactions and combine partial solutions. Good design decompositions are critical since "bad"
choices have strong negative impact on design efficiency. Choosing good decompositions is complicated by the fact that there is no decomposition that is good under all circumstances: The goodness of a decomposition depends on the design goals.

Design systems should not only have the ability to evaluate candidate designs for a particular set of specifications, but also be able to reason from the design specifications to candidate solution structures. This reasoning evolves gradually as successive goals and constraints are identified during the design process. Consideration of a goal or constraint at some point in the design process may result in changes of previous design decisions and/or in recommendations for modification of previously considered goals and constraints. This process is especially complex since the goals and constraints could be at different levels of abstraction, pertain to different design specialties, and express different design tradeoffs (e.g., aerodynamic efficiency versus structural strength for a turbine blade).

Design systems should have the ability to: (a) represent designs in ways that facilitate solution construction, (b) represent design records so as to facilitate explanation and design reuse, (c) represent designs from multiple viewpoints (e.g., aerodynamic and structural perspective for aircraft wing design), and (d) organize large reusable design knowledge bases.

The concept of design record is of central importance in research on design and it is an area where AI can be most useful1. A design record includes descriptions of the problem specifications in terms of design goals and constraints, the solution to the design problem, and the trace of decisions that shows why the solution satisfies the problem specifications. The record should be structured in such a way that it can be effectively used for tasks such as explanation, redesign, and design by analogy.

Because of the complexity of the design activity, in most cases it is impossible for a single designer to carry out the whole design effort alone. As has been ascertained from field studies of design in large organizations (e.g., [Bond 89]), design is a problem solving process among cooperating specialists/perspectives2. In AI terms, this process can be viewed as multi-agent problem solving where each agent is a design expert who has (a) incomplete knowledge of the environment, (b) limited knowledge of the constraints and intentions of other agents, and (c) limited number and amount of resources that are required to produce a system solution. The goals and constraints of the agents are the result of their particular expertise. Thus, each agent has a limited and egocentric view of the problem. A design is a solution to reasoning about multiple conjunctive and possibly conflicting goals.

The process through which a final design is produced is cooperative. Because each expert has insufficient local knowledge to solve the problem by himself/herself, experts have to cooperate. Conflicts arise, however, because of the variety of concerns, knowledge and evaluation criteria that each specialist brings to the design process. It is not easy to resolve these conflicts due to the fact that the experts (1) do not have the same mental model of the design, and (2) they do not "speak the same language". This engenders misunderstandings and long iterations of explanation.

---

1A design record is very similar to a design case in research work that uses case-based reasoning for design.

2In this paper, we use the words "specialist", "perspective", "expert", "party" and agent synonymously.
and attempts to "translate" concerns and knowledge into another's language.

Existing approaches to concurrent design have primarily been concerned with investigating architectures for communication between various perspectives [DICE 89], [Lander 89, Talukdar 88] or on conflict detection [Robinson 87, Sriam 88]. In this paper, we focus on modeling the process of reconciling design decisions and design proposals that arise from the different perspectives during the design process in order to form an acceptable compromise, i.e., the final design. We argue that understanding of the negotiation process in design will enable (1) the development of intelligent and efficient design support systems to aid human designers in their tasks, and (2) progress towards the development of systems that can reason from design specifications towards candidate solution structures.

The design problem has the following characteristics:

- The global system goal is to produce a design that is synthesized from contributions of different expertise, concerns and constraints.

- During the design process, conflicts in the form of constraint violations could arise. If these conflicts are not resolved in a satisfactory manner, infeasible designs will occur.

- Another kind of conflict is over approaches/styles used to achieve a design goal.

- Disparate evaluations of (partial or complete) designs could surface as a result of different criteria used to evaluate designs from different perspectives. Typically, these criteria cannot be simultaneously and optimally satisfied. The design decisions that optimize one set of criteria could conflict with those that optimize another set. If these conflicts do not get resolved in a satisfactory fashion, design suboptimalities occur.

- The system goal is achieved by making the best tradeoffs on conflicting design goals and constraints.

- Because of the presence of conflicting constraints, goals and possibly evaluation criteria, it is impossible for each agent/expert/perspective to optimize the overall design using only local information.

- Backtracking can be a major problem since it may result in invalidating design decisions that other agents have made. Hence the need for computationally efficient multi-agent models.

---

3It has recently been advocated that not only design and manufacturing but also marketing, subcontractors, suppliers and customers should be involved in the design process. This is what happens in Japan, where design and prototype building and testing takes more time on average than in the U.S. but debugging the design takes much less significantly decreasing cycle time from concept to market. This design philosophy and methodology inevitably introduces into the concurrent design process even more participate each with their own goals and evaluation criteria. As a result, the design team is not only faced with possible constraint violations that must be resolved by parameter relaxation and backtracking but also with more complex conflicts involving goals and evaluation criteria.

4We avoid using the word "optimal" since it is not always possible to prove optimality in design. On the other hand, the system goal is not satisfied by the first design that minimally satisfies goals and constraints, but improvements in the design are sought.
As a result of the above characteristics, the final successful design can be viewed as a compromise\(^5\) that incorporates tradeoffs such as cost, ease of manufacturing and assembly, reliability and maintainability. We suspect that such compromises are commonly done implicitly by human design experts tacitly using rules of thumb (e.g., imprecise versions of other agents' perspectives' evaluation functions). Typically, these implicit compromises go unrecorded making it very difficult to trace and avoid suboptimalities in the design. The proposed model allows for explicit recording of design proposals, modifications of design decisions, and associated justifications or objections.

Depending on particular decisions concerning tradeoffs, different designs will be produced. For example, the valve for a water tap could be a metallic threaded part or a plastic plug valve with a hole. There is a tradeoff between the low cost of the plastic valve and the high durability of the metal valve. Such tradeoffs can be accomplished in a rational fashion with the use of negotiation techniques. During the negotiation process, the feasibility and desirability of proposed tradeoffs is evaluated and may result in incremental design adaptations. Despite the difficulty of applying negotiation techniques, recorded goal conflicts and their resolution, namely existing designs, provide a foundation for rationalizing designs.

Negotiating an agreement involves finding a compromise solution for multiple conflicting goals. This is a complicated problem, not amenable to traditional AI planning techniques [Sycara 88a]. The negotiation process is dynamic and formulating it as a search problem is inadequate since there is no well-defined goal or search operators. The negotiation process itself is a search of a dynamic problem space where an agent's beliefs about another agent's beliefs over the cycle of proposals continuously changes the space being searched. What was not a solution at one point becomes a solution at a later point as new constraints enter the process.

A negotiation model must somehow reason about alternatives that include "suitable" values for each issue on which an agent will agree. For continuum-valued issues the available choices are infinite. Hierarchical decomposition of the problem into smaller subproblems each of which is easier to solve may not be suitable, since a compromise-solution may be a "package" whose parts are strongly interconnected and interacting. These difficulties are compounded by the absence of a coherent set of constraints that alone could guide search through the space of all possible settlements.

To address these shortcomings of traditional AI and expert systems, we have integrated reasoning from previous designs, case based reasoning (CBR), symbolic and numeric constraint propagation, and the use of multi attribute utilities. In contrast to other work [Werkmam 89] where different perspectives propose values for a design attribute and an arbiter evaluates

---

\(^5\)Pruitt [Pruitt 81] has identified two types of negotiation which are used by expert human negotiators to seek acceptable solutions and which may be applicable to machine agents: (a) compromise negotiation where each party makes concessions on its demands to facilitate agreement, and (b) integrative negotiation where the most important goals of each party are used to form innovative solutions, relinquishing, if necessary, secondary goals. In our view, both these negotiation types result in compromise solutions, in other words in partial goal satisfaction. Moreover, in typical negotiations a goal of secondary importance to one agent could be of primary importance to another because of different local evaluation functions. My use of the word "compromise" encompasses both these negotiation types. For more detail on different types of negotiation solutions and strategies, and mechanisms to automatically generate them, see [Sycara 87a].
and selects among the proposed values (i.e., there is no interaction among the perspectives), our model captures the full complexity and dynamics of negotiation by representing and modeling the communication and interactions that occur during negotiation in design. A communication protocol that supports negotiation is presented in section 8. The negotiation model that we present can be used for negotiation at each design stage, negotiation between design teams, and negotiation involving both humans and machines.

In the framework of the Design by Hindsight project at Carnegie Mellon, we are investigating the integration of Case-Based Reasoning, Qualitative Reasoning and Constraint Propagation in the domain of mechanical design [Sycara 89a]. Negotiation as a method for resolving constraint violations and achieving design optimization is a part of the project. A parallel effort is the development of a Manufacturing System Architecture [Sycara 89b] that will allow machine and human agents in the manufacturing domain (e.g., designer, process planner, scheduler, facility layout) to communicate and negotiate in order to integrate decision making within the manufacturing enterprise.

Although the proposed model allows for automating negotiation, the interface enables human designers to participate in the process. Users can register their reaction (acceptance or rejection) to proposals, express their objections to particular pieces of a proposed design, give the reason for a particular objection, and input their utilities with respect to various issues.

The rest of the paper is organized as follows: Section 2 presents rationale for the usefulness of accessing past designs and design decisions as part of the design process. Section 3 presents requirements of a model for negotiation. Section 4 presents a simple example of negotiation in concurrent design. Section 5 presents our negotiation process model and the reasoning methods used in the model. Section 6 presents the conceptual vehicle for representing agents’ goals and expertise. Section 7 presents the agents’ interactions during negotiation. Section 8 presents a detailed protocol for negotiation and section 9 presents concluding remarks.

2. Use of Design Cases

From the standpoint of AI, designs can be viewed as solutions to problems with multiple conjunctive and possibly conflicting goals. In this regard, mechanical design is not different from most other design disciplines. However, in many design disciplines, such as software design and circuit design, designs can be characterized as collections of weakly interacting functional modules, each of which implements one of the functional requirements. On the other hand, good mechanical designs are often highly integrated, tightly-coupled collections of interacting components [Suh et al. 78, Ulrich and Seering 88]. A simple and obvious correspondence between specific functional requirements of the artifact and individual components in the design does not usually exist. This has the consequence that there is no obvious decomposition of the overall function into subfunctions. For example, in a can opener the circular blades perform the function of holding the can, rotating it and cutting off the top. Ironically, these are also the major functions of the entire can opener. It is not possible to identify specific features of the can opener or its blades which perform each of the functions independently.

Because of the highly coupled nature of mechanical designs, design cases that capture good working solutions to component interactions provide a good basis for performing design. Cases
are very important in engineering design. In fact almost all design efforts [Mark 90] do make use of previous design experience to provide comparisons, warn of pitfalls, and sometimes furnish components or structures for the new design. But the use of this experience is limited to the individual designer's ability to retrieve and determine the relevance of past design experiences (mostly his own) to the new design problem. Cases are the primary way in which engineering students are taught to design. This is because there are no general algorithms for design. Typically, students are exposed to numerous cases and examples which illustrate how complex problems are solved. Even when an entry-level engineer joins a design office, an important part of his training involves going through the design records of previous projects. Although the engineering design community recognizes the importance of cases in problem solving, the use of precedent cases in Computer-Aided Engineering (CAE) tools has been completely ignored. This is not because the CAE research community is not aware of the ubiquity of case based reasoning in design, but because they have not had access to the right techniques.

A typical CAE tool for design includes a geometric modeling system and a standard set of analytic tools for tasks such as finite-element and boundary-element analysis. Over the last five years, design tools are being extended to include design heuristics. These heuristics come in several forms: as rules, as constraints and as recommendations. It is only very recently that a third aspect of the design process, the use of past cases, is beginning to be recognized in the Design Automation literature [Mostow 85, Ullman & Dietterich 87], [Huhns 87], [Mostow & Barley 87, Navinachandra et.al. 87]. Almost all organizations have records-on paper or in CAD systems-of the detailed specifications and results of previous design efforts. These records have limited usefulness, however, due to the fact that they typically do not record the design decisions that led to the final design, alternatives that were perhaps considered and abandoned, or the rationale for those decisions. Design "cases" on the other hand, besides indexing designs in terms of relevant features, also incorporate memories of design processes including the relevant decisions and their justifications.

Using previous cases in coming up with solutions to a new problem is called case-based reasoning. The central idea of Case-Based Reasoning (CBR) is that reasoning is done by recalling previous appropriate cases rather than reasoning from "first principles". Complete solutions for conjunctive goals (i.e. previous designs and design rationale) that have been used with success in the past are stored in memory in the form of cases, so they can be accessed and used in the future. Failures and failure reasons are also stored so that they can be used to predict and avoid future failures. If features in the past situation that gave rise to a failed solution are also present in the current situation, then the failed solution should not be tried. Thus, previous failures help the problem solver avoid potential present failures. If repairs are also stored in memory6, upon discovery of a failure, an appropriate repair can be found. In that case, the repair is stored along with the associated failure. The repair can be applied if the same failure happens or can be predicted [Sycara 87b, Sycara 88b].

In our model, cases are organized hierarchically in memory around important concepts in the problem domain. In order to perform CBR, cases need to be retrieved in terms of conceptual similarity. The basic idea behind conceptual similarity between two concepts is that they share salient attributes. Salient attributes are the ones that allow a reasoner to make inferences that

---

6Repair of a failure is available either when a similar failure case is available, or via direct user feedback.
will be useful for the problem solving task at hand. For example, a stool, a chair and an armchair, though different as to their structural features that are visibly prominent, are similar in that they can be used to deliver the function of "support for sitting". Although differences in attributes such as color, size and materials in the three objects may be more numerous than the similarities, they are not salient and so they are not considered in a case retrieval of designs of artifacts that can provide sitting support.

In contrast to expert systems that build solutions from scratch and discard them at the end of problem solving, in CBR the solution and solution context are integrated into the case memory so that they can be reused. Thus, learning is central to CBR. A case-based reasoner learns new design decisions during problem solving. It learns to predict failures so it can avoid them in the future. It learns repairs so that it can apply them if similar problems occur or can be predicted. As the case memory is enriched with new experiences, a case-based reasoner can refine its problem solving strategies and improve its performance. In practical terms, CBR alleviates the knowledge acquisition bottleneck that plagues expert systems since (a) experts are much better at remembering specific cases than rules, and (b) new cases are acquired as a by product of problem solving.

3. Requirements of a Model for Negotiation

Negotiation is a process in which the parties iteratively exchange proposals and proposal justifications until an agreement is reached. The final design is a "package" whose parts are strongly interconnected and interacting. The negotiation process exhibits several characteristics that give rise to various requirements for computer-assisted negotiation.

- **Negotiation** involving multiple agents with multiple conflicting goals/issues/assertions is usually a lengthy and iterative process. The parties start by having conflicting goals/issues/assertions and whose distance has to be narrowed gradually to zero. Therefore, a negotiation model must be **iterative** rather than **one shot**.

- After each round of proposals the agents give feedback to each other about which parts of a proposal (partial or final design) they agree or disagree on. Hence, a negotiation model needs to be able to receive and evaluate **feedback** about a proposition.

- In order to arrive at agreement, design proposals must be modified. Hence a negotiation model must have the ability to **propose suitable modifications**.

- Since final agreement is reached through narrowing the difference in the proposals of the parties, a negotiation model must have a way of **predicting/evaluating** whether each new proposal indeed narrows these differences.

- Reaching an agreement through negotiation entails that each of the parties must modify partially or totally some of their goals and proposals. A good grounding for such modifications is justifications and arguments in support of or against proposed modifications. Hence, a negotiation model needs to have a component that generates **justifications and arguments**.

In design, negotiation occurs recursively at all levels and stages of design from conceptual design through embodiment to detailing. This model of design by negotiation implies an
organizational structure that emphasizes product rather than function. The design is done by a number of design teams each of which contains various specialists and is responsible for producing a part of the desired artifact. For example, a detailed study of aircraft design [Bond 89] has found that aircraft design proceeds by the cooperation of specialists, each of which have their own model of the design. Design decisions are negotiated by the specialists among themselves. In aircraft design there are many specialists each with their own technology and language. For example, there are aerodynamicists who use surface models and equations, there are maintainability engineers, concerned with access, disassembly and replacement, there are hydraulic engineers, stress engineers, and thermodynamic experts. They do not understand each other’s specializations, but they have to collaborate to produce a single design acceptable to each one of them. Each team operates within constraint ranges set by other teams. Failure to reach agreement on the design of a part that is the responsibility of one team is a problem not only for the particular team but for the manufacturing organization as a whole. Such failure must be communicated to other teams and resolved by negotiation.

Negotiation enters the design process at the following points:

- When different relevant specialists have made conflicting recommendations regarding values of attributes of a design.
- When an attribute value proposed by one specialist makes it infeasible for another specialist to offer a consistent set of values for other attributes.
- When a design decision made by one expert adversely affects the decision optimality of other experts.
- When alternate approaches can achieve similar functional results.

There are two models of design by cooperating experts/perspectives: (1) one perspective generates the design and the others critique it, and (b) individual perspectives generate different partial designs which are exchanged among the participants, evaluated and "merged". Regardless of the adopted scenario, the final design is the result of a negotiation process.

4. A Negotiation Example

Consider, for example, the process of designing a turbine blade. Some of the dominant perspectives are aerodynamics, structural engineering, manufacturing and marketing. The blade design team operates within constraint ranges specified by the aircraft engine team. The blade design team incorporates various concerns. The concern of aerodynamics is aerodynamic efficiency; for structural engineering it is reliability and safety; for manufacturing, it is ease and cost of manufacturing and testing; for marketing it is overall cost and customer satisfaction. The two variables of concern in a turbine blade that we consider in this simple example are: (a) root radius, and (b) blade length. From the perspective of structural design, the bigger the root radius, the better since it decreases stress concentration. From the perspective of aerodynamics, the smaller the root radius, the better, since it increases aerodynamic efficiency. Concerning the length of the blade, from the point of view of structural design, the shorter the blade, the lower the tensile stresses; from the point of view of aerodynamics the longer the blade, the better the aerodynamics. On the other hand, if the blade is shorter, it makes for a lighter engine which is a desirable characteristic for aerodynamic efficiency. Thus, we see that the aerodynamics expert needs to make tradeoffs internal to its perspective. From the point of view of marketing,
aerodynamic efficiency lowers the cost of operation of the aircraft, thus making it more attractive to customers. From the point of view of manufacturing, it is easier to manufacture shorter blades with bigger root radii.

The following is a simplified example dialogue of the various concerned perspectives in an attempt to arrive at a mutually satisfactory turbine blade design:

Aerodynamics (possibly reasoning from existing designs) suggests particular values \( x \) and \( y \) for length and root radius of the blade. The suggested \( x \) and \( y \) values are within acceptable constraint ranges.

Structural engineering evaluates these values from its point of view and suggests values \( x' \) and \( y' \) where \( x' < x \) and \( y' > y \) (i.e., shortening the length and increasing the root radius) to increase safety.

Aerodynamics says that the values structural engineering suggested would considerably decrease aerodynamic efficiency.

Structural engineering counters that shorter blade makes engine lighter, thus also increasing efficiency.

Marketing says aerodynamic efficiency sells the product since it is less costly to operate.

Manufacturing supports structural by saying it is easier to manufacture short blades with big root radius.

Aerodynamics suggests that the materials engineering expert could try to investigate new materials that make the blade lighter, thus alleviating weight considerations.

Manufacturing says that new materials take lots of time to test and debug.

Structural engineering adds that new materials may introduce safety hazards that could go undetected.

This example illustrates the exchange of proposals for values of length and radius of the blade as well as the exchange of arguments and justifications in critiquing the proposed values. The nature of the resulting design will depend on (a) the ranges of various artifact constraints, (b) which constraints can be relaxed and in what ways, (c) the relative importance of various artifact-dependent goals (e.g., aerodynamic efficiency), (d) relative importance of various artifact-independent goals (e.g., safety), and (e) the way in which particular variable values contribute to the achievement of the goals.

5. The Negotiation Process

The negotiation process consists of three main tasks: generation of a proposal, generation of a counterproposal based on feedback from dissenting parties, and communication of justifications and supporting evidence. An initial compromise is generated and presented to each expert/perspective. Each agent evaluates the proposal from their point of view and register their reactions (evaluations, objections and suggestions). The process terminates when all concerned
agents accept a proposed design.

Each agent during negotiation engages in the following activities:

- **recommending design decisions**, i.e. designs that express potentially acceptable compromises and tradeoffs of the parties. In our work, we are investigating the generation of design recommendations using a combination of case-based reasoning, use of utilities and constraint propagation techniques.

- **justifying recommendations**. Often the agents, guided by their own values and criteria, cannot recognize why a proposed design may be the best under the circumstances. In order for a design proposal to become intelligible and have an increased chance of being accepted, justifications must also be communicated. These justifications help an agent assess the desirability of a design decision from his point of view. These justifications can be derived from running numerical models or accessing previous appropriate designs.

- **exploring feasible alternatives** so as to optimize the proposed compromise. A memory for past designs provides a rich repository of such alternatives.

- **modifying a rejected compromise** to make it more acceptable to the rejecting party without making it unacceptable to the party that had previously accepted it. This is done using previous cases and modification rules. A modified candidate design is evaluated using an appropriate criterion of improvement.

The input to the negotiation process is the set of conflicting goals and violated constraints of the various design agents and the context of the design (e.g., constraints that have been handed to the design team from others). The final output is either a single consistent set of design decisions that have been agreed upon by the agents, or an indication of failure if the negotiating parties did not reach agreement within a particular number of proposals. The final output is reached through iterations of the following tasks: (a) proposal of an initial set of design decisions, (b) justification and critiquing of the proposal and (c) repair and improvement of a rejected proposal. These tasks are performed using knowledge of existing designs and their characteristics, knowledge of physical laws and constraints, traces of design decisions made so far and goal graphs (see section 6) of the participating agents. The agents could have access to a common case base of previous designs but because of their specialized knowledge each one accesses and evaluates a design from a particular view. Use of previous design cases offer a reasoner (a) suggestions of how tradeoffs and resolutions have been made in the past, (b) failure avoidance advice (since failures are recorded in the design case), (c) possible modifications and repairs to design suboptimalities.

Figure 5-1 presents an agent’s actions in the negotiation process.

In the figure, the square rectangles represent the negotiation planning processes: plan generation, plan evaluation, plan presentation to the parties, communications of justifications and arguments, plan modification, and memory update (either with successful or failed cases). The “Generate Plan” rectangle is shaded to indicate that it is the starting point of the process. Each negotiation task, i.e. generation of an initial proposal, argumentation, and proposal modification recursively uses a subset of the problem solving processes, namely plan generation, plan evaluation, plan presentation to the parties, and (if needed) plan modification. The word "plan" is used to denote a set of design decisions (i.e. a partial or final design) and it applies to a
Figure 5-1: The Negotiation Process

differential entity for each task. For proposal generation, “plan” refers to a design proposal, for argumentation, “plan” refers to a set of justifications, for proposal modification, “plan” refers to a set of design decision modifications. The rectangles with the rounded corners in figure 5-1 denote the methods, CBR or use of multi-attribute utilities (Preference Analysis), used in each process. The ellipses denote conceptual inputs to a process and the diamonds represent decision points.

Negotiation is performed through integration of case-based reasoning [Kolodner et al. 85, Sycara 87a] and use of multi-attribute utilities and constraint relaxation. These methods are employed in all negotiation tasks, namely in generation of an initial proposal, repair of a rejected proposal to formulate a counterproposal, and communication of justifications and objections. The integration of heuristic and analytic methods makes a system both robust and flexible. The problem solver does not break down when heuristic methods fail. In addition, the problem solver has the flexibility to use whichever method type is more natural to the particular problem-solving stage in which it is engaged.
5.1. Case Based Negotiation

Negotiation is a multi-agent, iterative process. Because design decisions are typically tightly coupled, modification of one design decision may necessitate undoing of previous decisions. This can have major ripple effects and render the process very inefficient. Thus, efficiency has increased importance in multi-agent systems. Reasoning from previous design cases can result in major efficiency gains in negotiating design decisions. The use of previous cases is beneficial for negotiation because:

- Negotiation is a complex process involving many steps. A case-based reasoner can learn complex sequences of negotiation steps that it can re-use rather than reason from scratch for a set of conflicting and inconsistent goals and decisions which it had resolved in the past.

- Negotiation is a process involving multiple conflicting goals. Traditional AI planners (e.g. [Sacerdoti 75, Sussman 75]) reason about each individual goal and then deal with any incompatibilities as they arise. This method, besides being inefficient, compels planners to recreate and then debug the same mistakes rather than avoid them altogether using previous experiences.

- Negotiation is an ill-defined domain with uncertain and incomplete knowledge. There are no well-specified goals and operators. Hence traditional AI planning methods cannot be used effectively.

- There is no strong domain model for negotiation. Therefore, reusing sets of decisions that have worked in the past is efficient.

- Many problems during negotiation occur with some regularity (e.g. a particular tradeoff among the same variables). Hence, resolutions of these problems can be remembered and re-used.

- Previous negotiation experiences may point out important issues that may have been overlooked in the current negotiation.

- Case-based inference minimizes the need for information exchange during negotiation, thus minimizing communication overhead.

- Anticipating and avoiding problems through reasoning from past failures helps the agents minimize the exchange of proposals that will be rejected.

- If the repair of a past failure is also stored in memory, computation by each agent is minimized.

- Cases provide successful (and failed) goal and constraint relaxations.

The case based reasoning process consists of the following steps: (1) retrieve appropriate cases (or case pieces) from memory, (2) select the most appropriate case(s) from those retrieved, (3) compare similarities and differences of the current and the retrieved cases, (4) construct a baseline solution, (5) evaluate the baseline solution for applicability to the current problem, (6) based on the evaluation, apply modifications to the baseline solution to fit the current situation, and (7) verify (if possible) the correctness of the solution. If verification fails, then debugging must be applied. The debugging process itself could also use case based reasoning to find and apply appropriate repairs. Although the case based reasoning steps have been presented as sequential, they could be interleaved. For more details on case based negotiation, see [Sycara r.].
5.2. Utility Based Negotiation

Another negotiation method that is integrated with case-based reasoning is the use of utilities [Keeney 76, Sycara 88c] and constraint relaxations. Preference Analysis is based on Multi-Attribute Utility Theory [Keeney 76] and is used in our model as the underlying formalism for portraying the parties' preferences. Utility theory models the process through which a decision maker evaluates a set of alternatives, so that he can choose the best one. It has also been used in aiding a decision maker to structure his problem in such a way that evaluation of the alternatives is easily accomplished [Whitmore 74, Keeney 75]. We concentrate on the ways that utility theory can be exploited in a multi-agent setting to: (1) generate potentially acceptable solutions to be proposed to the parties, (2) measure the quality of a modification to a rejected settlement, and (3) determine the effectiveness of justifications and arguments.

Each agent has a utility associated with each issue in the negotiation, i.e. a particular design decision. A design is a vector whose elements are the design attributes of concern. Each particular design alternative is a vector with specific values for each of the design attributes. Since each attribute value has an associated utility for each agent, the combination of these utilities results in an overall utility for the agent associated with a particular design proposal. An agent chooses the design alternative that maximizes the overall utility from its point of view. The utilities portray the possible tradeoffs that can be made among conflicting sets of values to arrive at an acceptable compromise. Knowledge of an agent's utilities is helpful to himself as well as other agents in making proposals for alternative design decisions. Utilities incorporate tradeoffs that could be derived from mathematical models, performance information captured by statistics and subjective judgements of the specialists. Because of the subjective elements in the utility calculations, different agents may have different utilities for the same design decision. Hence proposed design alternatives may not be immediately agreed upon by all agents.

Since many of the values that enter the calculation of a design alternative are continuous-valued, there are in general an infinite number of alternative designs. However, range subdivision of the values of the relevant attributes, application of constraints and sensitivity analysis can be used to arrive at finite sets of alternatives that designers can select from. Given a set of alternatives, various criteria can be used to determine the overall best alternative. The selection criterion most often cited in the literature (e.g., [Keeney 76, Navinchnanda 87]) is Pareto optimality. Another criterion that we have found useful for negotiation is to select the alternative that combines Pareto optimality with equity. In other words, the criterion maximizes the joint payoff of the agents and minimizes the payoff difference [Sycara 88c].

Use of utilities is particularly suited to problems involving multiple goals not all of which can be simultaneously and entirely satisfied. If an agent has some knowledge of another agent's utilities, he can model the tradeoffs that the other agent is willing to make among the many issues and goals in the negotiation and could predict which compromise the second agent will be most willing to accept. In case of proposal rejection, the payoff of possible counterproposals can be calculated. Based on such calculations, an agent can predict which counterproposal represents a relative improvement (i.e. giving an overall higher payoff) over the rejected one. Without an ability to predict which counterproposal constitutes an improvement, the negotiating agents could blindly exchange offers that could never converge to an acceptable compromise.

The criterion used in our work is that a counterproposal is an improvement if it increases the rejecting party's payoff by a greater amount than it (possibly) decreases the payoff of the parties
that had previously agreed to the compromise. Upon proposal rejection, possible counterproposals are generated by modification of the rejected alternative. The criterion is used by each agent to evaluate possible counterproposals that the agent is contemplating. If a contemplated counterproposal conforms to the criterion, it is proposed. If it does not, subsequent modifications could be performed through (a) case-based reasoning, (b) constraint relaxation methods, and (c) traversal of goal graphs (see section 6).

6. Agent Goals and Expertise

The exchange of proposals and justifications lies at the heart of negotiation. It is the process used to cohere the decisions of the parties and guide the process toward solution convergence. We claim that in order to negotiate effectively, agents need the ability to (a) represent and maintain models of the knowledge and goals/beliefs of other agents, (b) reason about other agents' goals/beliefs, and (c) influence other agents' beliefs and intentions through the exchange of justifications and arguments. The information communicated is intended to "convince" the recipient agent to shift his negotiating position so as to narrow the parties' differences to achieve final agreement. In a nutshell, the process of communication of justifications and arguments can be described as follows: an agent reasons about another agent using its own (largely inexact) model of that other agent, finds as many ways as the model will allow to affect the other agent's outcomes, and uses them selectively to influence the other agent.

The knowledge needed to perform generation of arguments and justifications in the design domain is (a) previous designs, and (b) an agent's belief and preference structure. The belief structure of an agent consists of a collection of goals/beliefs, goal importance, amount and feasibility as well as relationships among goals. We use the word "goal" and "belief" to indicate what is commonly thought as beliefs (e.g., that safety is a good attribute of an aircraft), abstract goals, such as increasing marketability, reducing operational costs, and also design attributes, such as length, particular material types, connections types etc. In other words, an agent's expertise is encoded as part of his belief and preference structure (to the extend that such expertise can be represented). We represent an agent's belief structure as a directed acyclic graph where each node represents an agent's goal. Edges of the graph linking two goals represent the relationship between goals in terms of how one affects (positively or negatively) the achievement of the other. For example, aerodynamic efficiency positively affects lower operation costs. Associated with each node is:

- a sign (+ or -) that denotes the desirability of an increase or decrease in that goal
- the amount by which the attribute/goal should be increased or decreased
- the importance that the agent attaches to the goal
- the feasibility as perceived by the agent of achieving the goal

The values in the goal graph are set based on the constraints on the design and company policy. For example, after an accident whose cause was traced to a defective engine, the importance of the safety goal increases for the marketing agent. Directed edges connect subgoals to the higher level goals to which they contribute. A contribution value is associated with each directed edge denoting the contribution of the subgoal to the higher level goal. Contribution values range from -100% to +100%. A positive value means that the subgoal supports the achievement of the higher level goal by the denoted percentage. A negative
contribution value has the interpretation that the subgoal is detrimental to the higher level goal. Sink nodes\textsuperscript{7} are the highest level goals of an agent.

Figures 6-1 and 6-2 present a partial view of the belief structure of the structural and aerodynamic perspectives. For visual simplicity we have indicated in the figure only the edges connecting particular nodes, and the appropriate sign.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{structure.png}
\caption{Partial belief structure of structural engineering}
\label{fig:structure}
\end{figure}

By traversing a goal graph one can answer the following queries:
- Which goals are supported by a set of design decisions?
- Which design decisions are justified by a set of goals?

A path from node $X$ to node $Y$ in a goal graph constitutes a causal/justification chain that provides an explanation of the change in $Y$ in terms of the change in $X$, assuming no other change has occurred in the rest of the graph. For example, from the point of view of structural engineering, decreasing the length of the blade, Blade-Length(-), decreases tensile stresses, Tensile-Stress(-), which results in structural soundness, Structural-Soundness(+). In turn, an

\textsuperscript{7}Sink nodes have no out-going edges
increase in structural soundness increases reliability, resulting in increased safety and contributing to increased marketability of the blade.

In addition to an agent's beliefs, the representation includes an estimate of its utilities for each attribute in the goal graph. Utilities express the preference structure of an agent. Moreover, utilities express the tradeoff structure among various attribute values associated with an alternative design. The (possibly nonlinear) utilities of individual attributes are combined to give an overall utility of an alternative. Being able to compare different alternatives enables a reasoner to choose the alternative that affords the maximum payoff. An integration algorithm [Sycara 89c] traverses the belief structure to determine which way goal values should be moved to increase payoff and thus the acceptability of a resolution. Moreover, goal graph traversal allows an agent to discover alternative design decisions that support important goals thus leading to innovative designs.

7. Agent Interactions

A central task in negotiation is communication. Local computation of the agents resulting in suggestions or evaluation of various design proposals is interleaved with communication to other agents of the results of local computations. Conversely, an agent uses communicated feedback of other agents as input in its local computations. An interesting issue is the vocabulary used by the agents for communication. As has been stated in the introduction, design specialists do not share
vocabulary or exact understanding of each other's models and problem solving processes. This complicates intelligible communication among them. We are currently investigating the issue of shared understanding and communication among specialists. Our hypothesis is that specialists utilize terms in an intermediate shared vocabulary so that they can be intelligible to others. So, although each specialist's expertise is private to him, the intermediate vocabulary is the medium for making public relevant portions or results of the expertise for intelligible communication. In the example of designing a turbine blade, although the marketing expert might not understand the concepts of Axial-Velocity or Swirl-Coefficient, (see, figure 7-1), he understands the concepts of Blade-Efficiency and Structural-Soundness and how they relate to his own high level goal of marketability. Blade-Efficiency and Structural-Soundness are examples of terms in the intermediate vocabulary used for intelligible communication among the design specialists. In Figure 7-1, the shaded portions indicate the private expertise of the aerodynamics and structural specialists, whereas the unshaded portion indicates terms to indicate goals and issues in the shared, public vocabulary.

The messages that the negotiating agents exchange contain the following information:

- The proposed design
- Justifications of design decisions
- Agreement or disagreement with a proposal
- Requests for additional information, such as with which issue in the proposed design the agent disagrees.
- Reasons for disagreement.
- Utilities/preferences of the agents associated with disagreed upon issues.

Since different agents evaluate designs using different evaluation criteria, the information communicated by an agent to others cannot be simply its decisions. It needs to communicate justifications of its own design decisions and proposed design changes. If challenged, an agent must communicate arguments in support of its decisions and justifications. This capability can be used to provide agent-tailored explanations and suggest alternative design decisions from the viewpoint of each agent.

Proposals and supporting justifications are used by an agent, the persuader as a means to dynamically change the utilities associated with various decisions and outcomes of another agent, the persuadee, so as to increase the willingness of the persuadee to accept a proposal. This, in turn, improves the efficiency of convergence to a global solution. By observing reactions to the proposed rationale and justifications, the persuader can update and correct its model of the persuadee, thus refining its planning and argumentation knowledge. In our work, generating proposals, counterproposals and justifications is based on integration of goal graph search, use of multi-attribute utilities, and availability of a case memory of experiences with similar negotiations [Sycara 87a].

---

8In describing the behavior of multi-agent systems it is often awkward to describe agent interactions in terms of "the first agent", the "second agent" etc. We employ the terms persuader and persuadee to distinguish between the agent that is presenting a proposal, justification or argument and the one to whom the proposition is being made. Of course, during the course of the interaction, two agents alternatively may play the role of persuader and persuadee.
Belief and belief modification in the model is based on the conjunctive goals of the agents and their interactions. A belief involves the correspondence between a state (a possible design) and the other agents' design decisions. Group knowledge focuses on the facts of the case: proposals, counterproposals, negotiation context etc. If agreement on a design were obtainable by inference from these facts, negotiation would be unnecessary. Such is not the case, however, since besides quantitative information available to the specialists, qualitative information involving assumptions and approximations as well as different evaluations of the assumptions and the resulting designs can be present. The negotiation process itself is a search of a dynamic problem space where an agent's beliefs about other agents' beliefs and hence feasible solutions continuously changes the space being searched. What was not an acceptable solution at one point becomes a solution at a later point as more information is generated and becomes intelligible through agent interactions possibly making some agents re-evaluate their positions.

During negotiation an agent's belief structure is updated based on his reactions to presented new information and proposals. In this way, an agent's model is refined and corrected dynamically. This functionality is important since (a) it is not possible for an agent to have a correct and detailed belief model of another, and (b) beliefs are not static but change with external circumstances and an agent's experiences.

If a reasoner decides to change another agent's beliefs, he needs guidance as to (a) what kinds of changes he wants to effect, and (b) how to do it. In addition, he needs to be able to make predictions regarding how a change in knowledge and beliefs will affect acceptability of a proposed design change.

We claim that a party's satisfaction with a proposition expresses his willingness to accept the proposition. Hence, if an agent could manipulate another agent's utilities (resulting in manipulation of that agent's payoff), he would be able to affect predictably the outcomes of the second agent. Convincing an agent to change his evaluation and increase his willingness to accept a design decision is modeled as producing a justification to increase the payoff of the proposition. The elements of a design decision are a subset of the information that appears in the agent's belief structure. Hence, the task of a persuader can be viewed as finding the most effective justification/argument that will increase a persuadee's payoff. Since a persuadee's payoff can be approximated by a linear combination of his utilities, the payoff can be increased by either changing the importance (coefficient) the persuadee attaches to an issue, or changing the utility value of an issue.

The argumentation goals of a persuader express what in the beliefs and outcomes of a persuadee he wants to influence. To accomplish the argumentation goals, argumentation strategies are used. Based on a utility view of argumentation, there are two argumentation goals that could be used in the model:

1. Change the importance of a persuadee's goal/issue
2. Change the persuadee's perception of an issue's value

Changing the importance that a party attaches to an issue reflects the intuitive notion that satisfaction with a thing is a function not only of the intrinsic value of the thing, but also of the importance that one attaches to it. Changing the importance of an issue translates into changing the corresponding issue's importance in the belief structure. The change in value of a point on
an individual utility curve for an issue can be interpreted as a change of the party's assessment of the value of that issue. This corresponds to changing the "amount" parameter A in a persuadee's belief structure. In the utility theory model, changing a party's assessment of the value of an issue is equivalent to changing the party's satisfaction curve at that value. Consider the situation where a designer objects to an increase of 20 inches in length of a variable as "too high". In the utility theory formulation, this can be translated as "payoff(20) = LOW". (i.e., the payoff with respect to the altered design would be some low value). In the agent's belief structure, this assessment is represented as LENGTH(+, A<20inches)^9 (i.e. the designer wants to have a length increase of at most some amount less than 20inches). Convincing the designer that this increase is not so high changes the length attribute to LENGTH(+, A=20inches). Correspondingly, this attribute change results in raising the designer's (payoff(20)>LOW).

The argumentation strategies used to accomplish the argumentation goals determine how argument generation is done. Two argumentation strategies can be used to accomplish the first goal (change the importance of an issue):

(a) indicate a change (increase or decrease) in the contribution of the present goal to a higher level goal of the persuadee. For example, the justification of a proposed decrease in turbine blade length that structural engineering could present (see section 4) is that such a design decision increases structural soundness of the blade, which in turn increases safety. This argument will be effective if the safety goal is of great importance for the organization.

(b) indicate a change in the feasibility of the proposed goal. For example, the arguments made by manufacturing (see section 4) that the development of new materials take a very long time to test and debug is meant to decrease the importance of new materials as an option for the blade design.

If a persuadee disagrees with a proposed argument, the reasons for the disagreement are analyzed for new information that could alter subsequent argumentation, such as new information about the persuadee's concerns. If the analysis reveals that the persuader had some incorrect notions regarding the beliefs and preferences of the persuadee, the appropriate updates are made to the persuadee's model. In addition, updates to the persuader's argumentation goals and strategies may be needed.

When the argumentation goal of the persuader is to change the persuadee's perception of an issue's value, two argumentation strategies can be used. One is to find a counterexample from past similar designs. The second is to find evidence from successful design practices of the manufacturing organization itself or competitors regarding the value of the attribute. For both strategies, the argument generation algorithm involves search of the design case memory. For example, if marketing rejects a particular design feature as costing "too high", the model could search for evidence showing that the company's competitors have successfully marketed designs with the same or more costly features. The justification for this type of argument is "appeal to prevailing practice".

The heuristic that is used to generate a persuasive argument if the persuader uses the first strategy is as follows:

---

^9We adopt the convention of only presenting the values of goal parameters (e.g., amount, sign, feasibility) that are relevant to the discussion at hand without worrying what value the rest of the parameters have.
Retrieve past designs with respect to the present issue.
IF a past design is found where the persuadee's proposal had been implemented and was deemed a failure,
THEN point out this failure
ELSE use the second argumentation strategy

The heuristic that is used to generate a persuasive argument if the persuader uses the second strategy is as follows:

1. Retrieve past experiences of similar persuadees (e.g., competitors) with respect to the same design issue.

2. Collect the ones whose value for the issue is appropriate.

3. Present them to the persuadee.

8. The Negotiation Protocol

We present in detail the communication protocol that supports negotiation that we are currently investigating in our work. For simplicity, the protocol is presented for two agents, agent1, who initiates an initial design and agent2, who evaluates the design and possibly generates a counterproposal. The protocol generalizes to more than one agent that evaluates and suggests modifications.

1. Agent1 communicates to agent2 a design proposal, as well as arguments and justifications in support of the proposal.

2. Agent2 uses the arguments and justifications communicated by agent1 to possibly modify its goal graph (e.g., change importance of goals, including possibly abandoning goals).

3. Agent2 evaluates the proposal from its point of view (using its constraints and utilities).

4. If the proposal satisfies agent2's local constraints and gives it payoff above a threshold, it communicates ACCEPT to agent1.

5. If not, agent2 generates a counterproposal by whatever problem solving means it has at its disposal (e.g., CBR, constraint relaxation).

6. Agent2 evaluates the counterproposal. If the counterproposal gives agent2 payoff above the threshold, agent2 communicates to agent1:
   • The PORTION/ISSUES of the proposal that have been modified
   • The REASON for modifying the previous proposal (e.g., value1 violates some of the agent2's hard constraints, a set of proposed values does not contribute enough to
higher level goals of agent2).

• The COUNTERPROPOSAL and its PAYOFF.

• ARGUMENTS and JUSTIFICATIONS in favor of the counterproposal.

7. If the counter-proposal does not give agent2 payoff above the threshold, agent2 goes to step 5.

8. If agent2 has exhausted all counterproposals it can generate through the methods of step 5, it traverses its goal graph to see whether there is another way to satisfy its higher level goals.
   • If there is, it generates a counterproposal and goes to step 6.
   • If there is not, it communicates FAILURE to agent1 (who now has to generate a modification and/or look for alternative ways in its goal graph).

9. Concluding Remarks

Design can be viewed as a multi-agent problem solving process involving multiple conjunctive and potentially conflicting goals. The agents have different expertise (e.g., mechanisms, hydraulics, assembly, testing) that propose and evaluate designs using different, possibly conflicting criteria. The process of negotiation is used to propose and examine design decisions involving various tradeoffs. Negotiation is performed recursively at all stages of design and involves different design teams. We have proposed a negotiation model based on (a) knowledge of previous designs, (b) communication of design rationale, justifications and objections to proposed design decisions, (c) constraint propagation and relaxation, and (d) traversal of goal graphs.
References

[Bond 89] Bond, A., and Ricci, R.
Cooperation in Aircraft Design.

[DICE 89] DICE: Initiative in Concurrent Engineering.
Red Book of Functional Specifications for the DICE Architecture.

[Gregory 88] Gregory, S.A.
The Boundaries and Internals of Expert Systems in Engineering Design.

[Hicks 87] Hicks, T.G.


[Keeney 75] Keeney, R.L., and Nair, K.
Decision Analysis for the siting of nuclear power plants-The relevance of multiattribute utility theory.

[Keeney 76] Keeney, R.L. and Raiffa, H.
Decisions with Multiple Objectives.

A Process Model of Case-Based Reasoning in Problem Solving.

[Lander 89] Lander, S., and Lesser, V.
A Framework for Cooperative Problem-Solving Among Knowledge-Based Systems.

[Mark 90] Mark, W.
A Design Memory Without Cases.

[Mostow 85] Mostow, J.
Toward Better Models Of The Design Process.


[Sycara 87b] Sycara, K.
Finding creative solutions in adversarial impasses.

[Sycara 88a] Sycara, K.
Resolving Goal Conflicts via Negotiation.

[Sycara 88b] Sycara, K.
Patching Up Old Plans.

[Sycara 88c] Sycara, K.
Utility Theory in Conflict Resolution.

[Sycara 89a] Sycara, K. and Navinachandra, D.
Integrating Case-Based Reasoning and Qualitative Reasoning in Engineering Design.

[Sycara 89b] Sycara, K., and Marshall C.
Towards an Architecture to Support Integration of Decision-Making in Manufacturing.

[Sycara 89c] Sycara, K.
Argumentation: Planning Other Agents’ Plans.

[Sycara r.] Sycara, K.
Negotiation Planning: An AI Approach.

Concurrent design, simultaneous engineering and distributed problem solving.


Function Sharing in Mechanical Design.
[Werkman 89] Werkman, K., and Barone, M. 
Evaluating Alternative Connection Designs Through Multiagent Negotiation. 

Experimental Determination of community preferences for water quality-cost alternatives. 
# Table of Contents

1. Introduction  
   1.1. Design  
2. Use of Design Cases  
3. Requirements of a Model for Negotiation  
4. A Negotiation Example  
5. The Negotiation Process  
   5.1. Case Based Negotiation  
   5.2. Utility Based Negotiation  
6. Agent Goals and Expertise  
7. Agent Interactions  
8. The Negotiation Protocol  
9. Concluding Remarks
List of Figures

Figure 5-1:  The Negotiation Process  12
Figure 6-1:  Partial belief structure of structural engineering  16
Figure 6-2:  Partial belief structure of aerodynamics  17
Figure 7-1:  Partial view of private expertise and shared communication vocabulary  19