

---

## **Argumentation-based negotiation for collaborative engineering design**

---

Yan Jin\*

Department of Aerospace and Mechanical Engineering,  
University of Southern California,  
3650 McClintock Ave., OHE-430,  
Los Angeles, CA 90089-1453, USA  
Fax: (213) 740-8071 E-mail: yjin@usc.edu  
\*Corresponding author

Mathieu Geslin

Honda R&D Americas, Inc.,  
1900 Harpers Way,  
Torrance, CA 90501-2746, USA  
E-mail: mathieumgeslin@gmail.com

**Abstract:** Designing complex systems requires collaboration among multiple engineers who coordinate to plan tasks, cooperate to resolve dependencies, and co-construct to identify shared objectives and solutions. While collaboration technologies have been developed to date, few can help designers negotiate effectively and reach agreement efficiently. In this paper, we propose an argumentation based engineering negotiation approach that provides a structured framework for designers to specify design situations, compose arguments, and make joint decisions by following various strategies. The details of the proposed approach are described and a case study is presented to demonstrate the effectiveness of the approach.

**Keywords:** collaborative design; negotiation; argumentation; coordination; cooperation; decision-making; co-construction; agreement.

**Reference** to this paper should be made as follows: Jin, Y. and Geslin, M. (xxxx) 'Argumentation-based negotiation for collaborative engineering design', *Int. J. Collaborative Engineering*, Vol. x, No. x, pp.xxx-xxx.

**Biographical notes:** Yan Jin is Professor of Aerospace and Mechanical Engineering in the Viterbi School of Engineering, University of Southern California. He earned his PhD in Naval Engineering from University of Tokyo. His current research interests include design methodology, agent-based collaborative engineering, and computational organisation modelling. He is a recipient of National Science Foundation CAREER Award (1998), TRW Excellence in Teaching Award (2001), and Xerox Best Paper Award (ASME International Conference on Design Theory and Methodology, 2002).

Mathieu Geslin is senior engineer at Honda R&D Americas Incorporated located in Torrance, California. He earned his PhD degree from University of Southern California and did research on argumentative negotiation for collaborative engineering. He also holds an MS in Mechanical Engineering from Ecole Centrale de Lille, France. He is currently engaged in racing car research and development activities.

---

## 1 Introduction

Collaborative engineering design is an activity in which multiple stakeholders are involved. The overall product development task is decomposed into subtasks which are assigned to multiple designers. To develop local design solutions, designers must collaborate with each other to deal with the dependencies between their subtasks. The final design solution is realised by integrating and synthesising the sub-solutions. Depending on the complexity of the design task, design teams may employ different forms of collaboration during design. Following *activity theory* (Leontjev, 1978; Engeström, 1987), we differentiate among three forms of collaboration, namely, coordination, cooperation, and co-construction. *Coordination* is a form of collaboration that calls for preplanning of design tasks and developing interaction rules and policies for designers to follow. Coordination emphasises predefined orders of tasks and roles of participants. It is highly effective for less complex and more routine design tasks where subtasks can be predefined and the composition of the final solution can be pre-determined because most dependencies are *sequential*. *Cooperation* refers to designers' interactions carried out to deal with *reciprocal* dependencies between their subtasks. In this case, one designer's design decisions may influence the decisions of another and vice versa. Designers focus on their shared problems, trying to find mutually acceptable ways to solve them. *Co-construction* is needed for highly complex and original design tasks, for which there is no clear and commonly acceptable way to approach the design. In this case, team members must co-construct subtasks and develop ways to solve the subtasks. Co-construction can result in new design requirements, constraints and evaluation criteria. It is worth mentioning that the above three forms of collaboration often co-exist in one single collaborative design process; one may be more dominant than the others depending on the nature of the design problem.

Research on collaborative design has addressed the issues of *coordination* and *cooperation*, but little has been done to deal with *co-construction*. Process modelling and management based technologies have been developed to support coordination. Program Evaluation and Review Technique (PERT)/Critical Path Method (CPM) (Joseph et al., 1995) and Design Structure Matrix (DSM) (Steward, 1981; Eppinger et al., 1994) were developed for task planning. Design Roadmap Framework (Park and Cutkosky, 1999) was proposed to represent large scale engineering processes. To facilitate both process modelling and process information exchange, Process Interchange Format (PIF) (Lee et al., 1998) and Process Specification Language (PSL) (Michel and Cutting-Decelle, 2004) were proposed. Besides process modelling languages and systems, efforts have been made to support coordination by facilitating *information sharing* among designers. DICE (Sriram and Logcher, 1993) utilises a centralised database system, while Designworld (Genesereth and Katchpel, 1994),

PACT (Cutkosky et al., 1993), and KICAD (Jin and Zhou, 1999; Jin et al., 1999; Jin and Lu, 1998) employs software agents, for coordination support.

*Cooperation* between designers deals with *reciprocal* dependencies. Researchers proposed models and heuristics based methods to support conflict resolution through knowledge representation (Klein, 2000) and rationale capturing (Pena-Mora et al., 1996), and others developed truth maintenance systems to capture and maintain design interdependencies between multiple designers and applied the dependency information to help conflict resolution (Petrie, 1993). Organisational issues were also addressed for effective cooperation among designers (Levitt et al., 1993; Jin and Levitt, 1996). Viewing collaborative engineering not only as a technical problem but also a social one, researchers have proposed to consider social perspectives of designers and developed a socio-technical framework to model and support collaborative design (Lu and Cai, 2001).

Research in Multidisciplinary Design Optimisation (MDO) offers computational solutions for effective *cooperation* among disciplinary designers working on subtasks (Kodiyalam and Sobieszczanski-Sobieski, 2001). Concurrent Sub-Space Optimisation (CSSO) is a non-hierarchic system optimisation algorithm that optimises decomposed subspaces concurrently, followed by a coordination procedure for directing system problem convergence and resolving subspace conflicts. In the Collaborative Optimisation (CO) approach, a complex problem is hierarchically decomposed along disciplinary boundaries into a number of sub-problems that are brought into multidisciplinary agreement by a system-level cooperation process (Kroo et al., 1994). The BLISS method uses a gradient-guided path to reach the improved system design, alternating between the set of modular design subspaces (disciplinary problems) and the system level design space (Sobieszczanski-Sobieski et al., 2000). To make MDO more effective in a dynamic teamwork environment, researchers proposed using game theory protocols to allow designers to solve interactive decision-making issues in collaborative design (Lewis and Mistree, 1998; Chen and Li, 2002; Xiao et al., 2002).

While the research to date has generated useful methods and techniques to solve coordination and cooperation problems where only *individual* and/or *interactive decisions* are involved, little has been done to address the issue of *co-construction* where task dependencies are not fully understood and *decisions must be made jointly*. To overcome this limitation, we propose an *argumentation based engineering negotiation approach* to supporting *cooperation* and *co-construction* in collaborative engineering design. The goal is to develop a negotiation framework that links designers and engineering systems together at decision-level and facilitates cooperation and co-construction among them through negotiation. Negotiation in general is a process in which a *joint decision* is made by two or more parties (Pruitt, 1981). The parties first verbalise contradictory demands and then move towards agreement by a process of agreement-making. For multi-disciplinary collaborative design problems, negotiation is a way for multiple designers to exchange information, acquire knowledge of other designers' perspectives and intents, and identify new opportunities. Therefore, design negotiation has the potential to create new opportunities and new designs.

In this paper, we present a framework of Argumentation-Based Engineering Negotiation, called ABEN, for collaborative design. Built on previous research on negotiation, ABEN is composed of a negotiation protocol, a set of negotiation strategies, a Design Context Model (DCM), and a network of intelligent agents that help human designers and computer systems to follow the negotiation protocol, select negotiation strategies, and make proposals. In Section 2, we review the current negotiation research

and discuss the requirements of engineering negotiation. After that, we describe the ABEN framework in detail in Section 3. In Section 4, a case study is presented to demonstrate the effectiveness of ABEN. Finally, concluding remarks are drawn in Section 5.

## **2 Negotiation and collaborative engineering design**

Negotiation in collaborative design is a process of interaction among designers in which inconsistent goals are reframed, conflicting issues related to functions and constraints are identified and narrowed, and alternative solutions are proposed, attacked and defended, and finally agreements are reached and confirmed. A negotiation framework for collaborative design facilitates such cooperative and co-constructive interactions.

Negotiation has been a research topic in various areas including social psychology where the focus is on developing principles for successful human interactions (Pruitt, 1981; Gulliver, 1979; Toulmins, 1969), Distributed Artificial Intelligence (DAI) of which the goal is to develop protocols and reasoning mechanisms for computers to work together (Bond and Gasser, 1988; Sycara, 1989; Rosenschein and Zlotkin, 1994; Kraus and Lehmann, 1995; Parsons et al., 1998), and decision-theory that promotes decision-theoretic models and analysis of negotiation (Raiffa, 2003).

Gulliver (1979) proposed an eight-phase model of negotiation process that describes the progress of negotiation from the initial recognition of the dispute to some kind of outcome. The eight phases are: search for arena, agenda setting, exploring the field, narrowing the difference, preliminaries to final bargaining, final bargaining, ritual affirmation, and execution. Pruitt (1981) proposed a strategic choice model of negotiation, stating that parties involved in negotiation must make strategic choices at every point in time. The choices include conceding unilaterally, standing firm, or collaborating with other parties in search of a mutually acceptable solution. Toulmin (1969) suggested a simple model of argument structure for negotiation. He proposed that the first step in an argument is for one party to express an opinion, called 'claim'. If the claim is challenged, it has to be defended by 'data' and 'warrant' successively. DAI researchers have developed various methods for distributed cooperative agents, or computer systems, to reconcile their disparities (Bond and Gasser, 1988). Sycara (1989) proposed a negotiation process that uses case-based reasoning mechanism together with a restricted protocol to support agents resolving their goal conflicts. Parsons et al. (1998) developed a negotiation protocol in which agents explore mutual spaces of negotiation and eventually arrive at a mutually acceptable solution by continuously exchanging arguments. Raiffa (2003) proposed taxonomy of group decision-making and suggested negotiation as a way to achieve joint decisions. Extending the multi-objective decision theory and game theory, he examined the dynamics of win-lose win-win and multi-party negotiations and proposed novel approaches for successful negotiation.

Researchers in the field of engineering design attempted to facilitate engineering negotiation by providing information, methodology, and technology supports. Some treated the problem of negotiation in design as an issue of information imprecision and developed formal mathematical models to incorporate the imprecision into design computations (Antonsson and Otto, 1995; Scott, 1997). Others formulated collaborative design problems as games and treated negotiation as a process of playing various types of games, e.g., collaborative, non-collaborative (Lewis and Mistree, 1998).

Viewing negotiation as a conflict resolution process and devising ways to support conflict identification and resolution is another direction of engineering negotiation research (Klein, 2000). CONVINCER (Pena-Mora and Wang, 1998) is a computer program that facilitates the negotiation process in large-scale infrastructure projects by integrating the concepts of game theory and negotiation forms and guiding negotiations toward sustainable outcomes. One common feature of the existing approaches to negotiation in engineering is that they treat negotiation as a process of single level information exchange and conflict resolution and attempt to *reduce* the negotiation problem into a multi-objective optimisation problem so that a convergent solution can be found. Because these approaches usually require prior knowledge of evaluation criteria and available alternatives, they have only limited use for the early stage of engineering design where defining problems and exploring alternative spaces is part of the negotiation process.

In our research on engineering negotiation, we view negotiation not only as a means to resolve discrepancies among designers during *cooperation*, but also as a *process of synthesis* in which designers *co-construct* to develop better understanding of the design tasks, better ways of doing design, and new alternatives as design solutions. We focus more on the negotiation process, protocol, strategies, and mechanisms for generating new ideas, rather than developing computational algorithms. In order to develop such a negotiation framework, we identified following important requirements.

- Deal with semi-structured and ill-structured problems. Design problems are often open-ended and not well represented. In dealing with such problems, human involvement is inevitable. There must be ways to support human-computer interaction for successful and efficient negotiation.
- Allow a variety of engineering issues to be negotiated. Design negotiation is not merely about avoiding conflict of parameter values. It is a means for designers to explore shared understanding of the design problem and make joint decisions for best design. Therefore, the issues must include not only design parameters, but also function requirements, constraints, and design goals and preferences.
- Encourage generating new alternatives. Unlike the negotiations in many other fields where give-and-take is everything, design negotiation must encourage generating new alternatives and reach agreements by expanding alternative spaces.
- Deal with multiple interrelated negotiations. Unlike deal making in business where a negotiation completes when two parties reach an agreement, engineering design often involves multiple interrelated negotiations between different groups of designers. There must be ways to link these negotiations together.

### 3 Argumentation-Based Engineering Negotiation (ABEN)

We take an argumentation-based approach (Jennings et al., 1998) to support engineering negotiation. The idea behind argumentation-based negotiation is relatively simple, i.e., negotiation should involve not only give-and-take, or propose-accept/reject (Smith, 1980); it should allow participants to argue about their positions, their understanding of the issues, and their goals. To apply this simple idea to engineering negotiation, we must determine *what can be argued* and *how to argue effectively*.

In the rest of this section, we first discuss possible situations of engineering negotiation and then present the framework of ABEN, its protocol, strategies, and information models.

### 3.1 *Design conflicts and Zone of Possible Agreement (ZOPA)*

Collaborative design involves making collaborative decisions among design participants. As pointed out by Raiffa (2003), there are two types of group decisions. One is *separate and interactive decision* where the payoff of one individual's decision depends on those of the others. In this *game* situation, decision-makers must interact strategically to achieve their best payoffs. The other type of group decisions is *joint decision* in which participants cooperate to attain joint understanding of the problem and aggregate multiple separate decisions into a joint decision. In this *negotiation* situation, the participants must gain sufficient understanding of others' positions and finally reach a shared agreement.

In collaborative design, the need for joint decisions among designers emerges from the recognition of potential conflicts between subtasks. In our research, we identified the following possible design conflicts.

- *Parametric conflict.* Conflicts among the values of design parameters or attributes of parts or assemblies.
- *Constraint conflict.* Two designers may be in a constraint conflict situation when they cannot find a common ground for their interacting design parameters because they hold on to their conflicting constraints.
- *Design entity conflict.* In this research we use the term *Design Entity (DE)* to represent ideas or solution principles. Two designers may share DE conflicts if their chosen design entities are not compatible.
- *Functional conflict.* If two designers are developing solutions for their subtasks by selecting incompatible Functional Requirements (FR), then they are in functional conflict.
- *Objective conflict.* Design objectives of different designers working on the same project may clash, e.g., 'minimise weight' may be in conflict with 'maximise strength'.

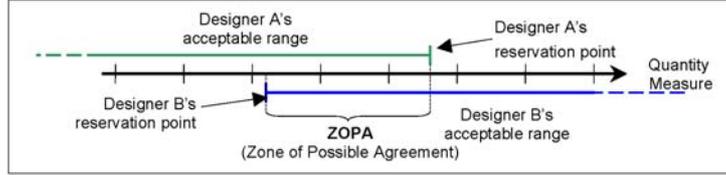
Besides the above design conflicts, there can be management conflicts, e.g., *resource conflict*, and personal conflicts, which will not be considered in this paper.

While most parametric conflicts and constraint conflicts are *quantifiable*, DE conflicts and functional and objective conflicts are usually *qualitative*. Quantifiable conflicts can be solved using the ZOPA model, as shown in Figure 1. If ZOPA can be recognised, then the issue of negotiation becomes how the designers can argue with each other and settle at an agreement point within the ZOPA.

In our research, we extend the concept of ZOPA into situations where design conflicts are qualitative. We hypothesise that for two designers to reach a joint design decision there *must be* a ZOPA between the two. The ZOPA can be at parametric value level, constraint level, functional requirement level, or design objective level, depending on the problem at hand. Lower level ZOPAs tend to be quantitative and higher level ones more qualitative. Furthermore, no ZOPA at lower level (e.g., parameter level) may imply a ZOPA exists at a higher level (e.g., constraint level or function level, or even

objective level). The research challenge here is *how can we devise negotiation protocols, strategies, and design information models that allow designers to recognise ZOPA by going through the levels and negotiate effectively to reach a mutually acceptable joint decision*. We address this question in the following subsections.

**Figure 1** Zone of Possible Agreement (ZOPA) model (see online version for colours)



### 3.2 A general negotiation model

In collaborative design, negotiation involves two or more designers with potential design conflicts who are trying to find ways to reconcile their differences. To clarify the scope of discussion, we introduce a general negotiation model of collaborative design as shown in equation (1).

$$\mathbf{N} = \{\mathbf{D}, \mathbf{I}, \mathbf{P}; \mathbf{S}, \mathbf{T}\} \quad (1)$$

where

$\mathbf{N}$ : Engineering negotiation process

$\mathbf{D} = \{d_1, d_2, \dots, d_n\}$ : A set of *participants*, i.e., designers and/or computer systems

$\mathbf{I} = \{i_1, i_2, \dots, i_m\}$ : A set of types of *issues* that are negotiable

$\mathbf{P} = [\mathbf{r}, \mathbf{q}, \mathbf{a}]$ : A *protocol* composed of communicative *speech acts*  $\mathbf{r} = \{r_1, r_2, \dots, r_1\}$ , negotiation *states*  $\mathbf{q} = \{q_1, q_2, \dots, q_k\}$ , and *strategic actions*  $\mathbf{a} = \{a_1, a_2, \dots, a_s\}$

$\mathbf{S} = \{s_1, s_2, \dots, s_g\}$ : *Strategies* for choosing *strategic actions*; each strategy is composed of a set of *strategic rules*

$\mathbf{T} = \{t_1, t_2, \dots, t_h\}$ : *Tactics* for choosing proposed issue instances; a tactic is composed of a set of *tactic rules*.

For a given collaborative design situation, participants, negotiable issues and negotiation protocol are common knowledge to everyone, while different designers may have their own strategies and tactics. The research question here is *how can we model negotiation issues, devise protocols, and develop strategies and tactics so that the requirements and challenge mentioned above can be met?*

### 3.3 Multi-level issues and design context

In engineering design, negotiation participants may include designers, managers, and customers. A computer system can also be a participant in negotiation. While identifying participants is relatively straightforward, modelling “what are the negotiable issues” and “what kind of arguments can be made during negotiation” may have significant impact on negotiation performance. In our research, we developed a multi-level model to represent negotiation issues, as shown in equation (2).

$$I = \{DO, FR, DE, DC, PV\} \quad (2)$$

where

$DO = \{do_1, do_2, \dots\}$ : Design objectives

$FR = \{fr_1, fr_2, \dots\}$ : Functional requirements

$DE = \{de_1, de_2, \dots\}$ : DE

$DC = \{dc_1, dc_2, \dots\}$ : Design Constraints

$PV = \{pv_1, pv_2, \dots\}$ : Parametric values

In ABEN, issues may range through different levels, from bottom *parametric value* level to top *design objective* level, as shown in equation (2). A lower-level issue is strongly influenced or governed by its related higher-level issues. A design parameter's value must be within a given range due to one or more Design Constraints (DC). Certain DC exist because certain design entities are chosen. The design entities are chosen to satisfy certain FR, and finally functional requirements are introduced to satisfy certain design objectives. A specific session of negotiation starts often when two designers find discrepancies among their lower level issues. To resolve their discrepancies, the designers may 'argue' at that level. If this is not enough, they may move to a higher level to identify related higher-level issues as current negotiation issues. This 'multi-level' negotiation is important for engineering design because the consistency of higher-level issues among participants implies better global design.

To represent the negotiation issues described above, we developed a *Design Context Model* (DCM). DCM is an object-oriented information model that captures design product and process concepts and sub-concepts for designers and computer intelligent agents to describe the design situation and compose negotiation arguments. The details of DCM can be found in Geslin and Jin (2005). In the following, we briefly describe the definitions of the major components of DCM.

*Design Entity (DE)*: Design entities refer to the elements generated during the design process to satisfy certain FR. Examples of such entities are: *solution concepts*, *components*, *assemblies*, and *parts*. A DE is usually characterised by a set of design parameters – i.e., the primitive variables that can be assigned *parametric values*.

*Design Constraints (DC)*: Design constraints specify relations and bounds of design parameters of the overall system or design entities. For a specific design problem, DC can be either given by customers or imposed by designers when new design entities are introduced.

*Functional Requirements (FR)*: Following Suh (1990), FR pertain to the functional domain and are specific requirements that determine the high-level design goals. Doing design is the process of relating the *functional requirements* to the *design entities*.

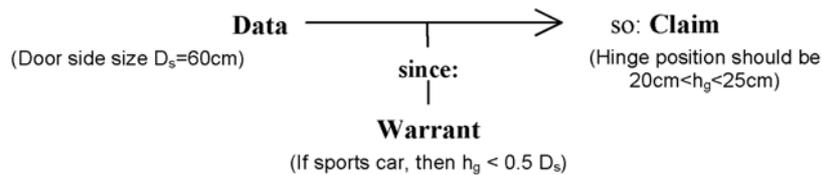
*Design Objectives (DO)*: A design objective is a statement of some aspect associated with the design product that a designer desires to achieve. For example, in designing bicycle frames, *maximise strength* and *minimise weight* can be two important objectives for a designer.

Using the above concepts, a designer can clearly describe the current design situation, the issues that need to be addressed, and the potential conflicts. The next question is how designers can generate powerful arguments to negotiate the issues.

### 3.4 Modelling arguments

One important feature of our ABEN framework is its structured way of representing arguments that designers can make in negotiation. Arguments are the basis of persuasive communication. Designers use arguments during negotiation with the intent of changing the minds of other designers. There are two basic requirements for argument modelling. First, the model should be ‘logical’ so that the reasons behind an argument are traceable. On the other hand, the model should also allow incomplete knowledge and qualitative information. To develop such a model, we turn to Toulmin’s well-accepted argument structure (Toulmin, 1969) in which three important elements of an argument are identified, namely, *claim*, *data*, and *warrant*. As shown in Figure 2, in this structure, argumentation starts with one party expressing an opinion, called ‘claim’. If the claim is challenged, it can be defended by ‘data’, if the ‘data’ alone is still not persuasive enough, then ‘warrant’ can be provided to support the claim and claim-data relationship.

**Figure 2** Toulmin’s (1969) argument structure and example



Following this model, when negotiation starts a designer makes a ‘Claim’ (e.g., “Hinge position  $h_g$  should be  $20\text{ cm} < h_g < 25\text{ cm}$ .”). If the claim is challenged, then the designer adduces ‘Data’ (e.g., “Door size  $D_s = 60\text{cm}$ ”) to defend it. If the challenger is still not satisfied with the data, then a ‘Warrant’ (e.g., “If sports car, then  $h_g < 0.5 D_s$ ”) can be supplied, either voluntarily or at the request of the challenger. A ‘warrant’ can be either a rule that states the relation between claim and data as shown in Figure 2, or a related higher-level issue. In this case, if the challenger starts to challenge the ‘warrant’, i.e., the higher-level issue, the negotiation moves to a higher-level in which the ‘warrant’ becomes a ‘Claim’ and negotiation continues.

In ABEN, the composition of ‘Claim’, ‘Data’, and ‘Warrant’ is supported by the DCM described above. Figure 3 shows a Backus Naur Form (BNF) summary of our argument model.

**Figure 3** BNF of argument model of ABEN

```

<Argument> ::= <Claim>* | <Claim> <Data> | <Claim> <Data> <Warrant>
<Claim>    ::= <Issue> IS <Value> | NOT <Issue> IS <Value>
<Data>     ::= <Issue> HAS <Value>
<Warrant> ::= SINCE <Issue> | SINCE <Rule>
<Rule>    ::= IF <Argument>|<Claim>|<Data>|<Warrant>
             THEN <Argument>|<Claim>|<Data>|<Warrant>
<Issue>   ::= <PV> | <DC> | <DE> | <FR> | <DO> /* See Equation (1)*/
<Value>   ::= Value assignable to corresponding issue
    
```

### 3.5 Speech-Acts and ABEN protocol

The argument model and DCM can help designers compose effective arguments in given design situations. The next question is *what should be the negotiation process in which the arguments can be effectively and efficiently exchanged*. This relates to negotiation protocol.

A negotiation protocol for engineering design should be *flexible* enough to deal with semi-structured or ill-structured design problems in which not all issues and arguments are numerical. On the other hand, it should also be '*formal*' enough so that negotiation efficiency can be achieved and computer support attained. In our research, we developed a *speech-act based argumentative negotiation protocol*. The protocol is based on the abovementioned issue and argument models and a set of verbs chosen from a speech-act dictionary (Ballmer and Brennenstuhl, 1981). Following is a brief description of the speech-acts used in the protocol.

- *Propose <claim>*: Introduce an initial <claim> and initiate the negotiation process.
- *Counter-Propose <claim>*: Introduce a new <claim> going against another *claim* proposed by the other party earlier.
- *Compromise <claim>*: Propose a <claim> as a compromised version of the previous one.
- *Critique NOT <claim> AS <data> (or SINCE <warrant>)*: Introduce a negated <claim> followed by <data> and possibly a <warrant> to justify the negation. The purpose of here is to convey the idea that the <claim> is not suitable to solve the problem for the reasons stated by either the <data> or <warrant>.
- *Defend <claim> AS <data> (or SINCE <warrant>)*: Introduce <data> and/or <warrant> to justify or defend the <claim> challenged by the other party.
- *Agree <claim>*: Declare that an agreement is reached on the <claim> and the party is committed to the agreement.
- *Refine <claim1> WITH <claim2>*: Introduce a new <claim2> whose contents builds upon the last <claim1> passed on by the other party. This SA proves useful in the creative thinking process occurring during integrative negotiation.

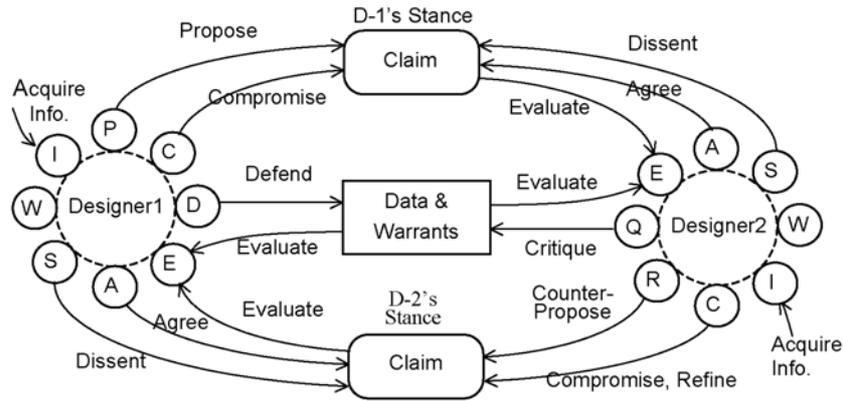
Table 1 presents the adopted speech acts and their acceptable responses.

**Table 1** ABEN Speech-Acts and their responses

<i>Speech act</i>	<i>Acceptable speech act in response</i>
Propose	Critique, counter-propose, agree, dissent, accept
Counter-propose	Critique, counter-propose, agree, dissent, accept
Critique	Defend, counter-propose
Compromise	Agree, dissent, critique
Defend	Agree, compromise, critique, counter-propose, refine
Refine	Refine, agree, counter-propose
Agree	Agree

Figure 4 illustrates the protocol being applied in a two-party negotiation. As shown in Figure 4, our protocol is composed of sets of *speech-acts*, *negotiation states*, and *strategic actions*. The speech-acts serve as performatives for designers to communicate their claims and arguments. Argumentation during negotiation is carried out through supplying <data> and <warrant> for speech acts *Defend* and *Critique*. The strategic actions are the actions that a designer can choose during negotiation. They include all the speech acts. In addition, *Wait*, *AcquireInfo* and *Quit* are three non-speech strategic actions.

Figure 4 Speech-Acts and ABEN protocol



- Speech-Acts:** Propose, Agree, Dissent, Defend, Compromise, Critique, CounterPropose, Refine  
**Negotiation States:** (P)=Proposing, (D)=Defending, (C)=Compromising, (A)=In-Agreement, (S)=In-Disagreement, (Q)=Critiquing, (R)=CounterProposing, (I)=AcquiringInfo, (W)=Waiting  
**Strategic Actions:** Propose, Defend, Compromise, Agree, Dissent, Critique, CounterPropose, Wait, AcquireInfo, Quit

In Figure 4, negotiation starts when D-1 (Designer1) proposes a claim, e.g., “Hinge position should be made  $20\text{ cm} < h_g < 25\text{ cm}$ ”. Upon evaluating this claim, D-2 (Designer2) may decide to *Agree* or *Dissent*. He/she may also request <data> or <warrant>, through *Critique*, to further understand the claim. Being dissented or requested, D-2 supplies <data> and <warrant> to *Defend* his/her claim or stance. Still not satisfied by the <data> and <warrant>, D-2 can *Counter-propose* a new claim, e.g., “Hinge position should be made  $25\text{ cm} < h_g < 30\text{ cm}$ ”. Upon evaluating this new claim, D-1 may *Agree*, *Compromise* (i.e., modify his/her own stance), or *Dissent*, or start it over again by *re-Propose* a new claim. During this negotiation process, a designer may choose to *Acquire* more information including engaging in negotiation with the third party or just *Wait*. Higher level issues used in *Data and Warrant* for *Defend* and *Critique* can be elevated into *Claims* during *Counter-propose* or *re-Propose* actions.

### 3.6 Negotiation strategies and tactics

Given the models of issues, arguments, and the protocol described above, the efficacy of negotiation depends on how the participating designers decide on strategic actions and how they compose proposals and arguments. The former is related to negotiation *strategies* and the latter to negotiation *tactics*. Strategies and tactics together determine

the direction of negotiation: whether to explore the solution space of the current issue, or to identify new issues at the same level, or to move up to a higher level of relevant issues.

Both strategies and tactics are often domain dependent and sometimes task dependent. In our research on automotive design, we developed following strategies.

- *Solution Exploration (SE)*. Try to stick to the current issue and explore its solution space extensively. In following this strategy, designers often *Propose* or *Counter-Propose* new solutions (e.g., new values for a parameter or an attribute). They sometimes defend their solutions by providing Data and Warrants but rarely pick up new issues from other party's arguments for further exploration.
- *Issue Exploration (IE)*. Try to move to, or create, new issues at the same level to avoid conflicts. New issues at the same level are used in arguments during negotiation and they are also picked up by the participants as the current negotiation focus. This way, negotiation is moved from one issue to another.
- *Function Exploration (FE)*. Try to move up to functional requirement level to resolve functional conflicts. Functional requirement is an important concept in engineering design. Designers following this strategy always attempt to make sure the consistency between the negotiating parties at the function level can be maintained.
- *Value Exploration (VE)*. Try to move up to design objective level and redefine design objectives. In addition to solution, issue and functional discrepancies, negotiating parties may have different value systems that govern what they want. Negotiation with VE strategy involves identifying value level conflicts and reconciling them.

Each strategy is implemented as set of IF-THEN rules, of which the execution leads to moving negotiation to a specific direction by selecting specific group of strategic actions and arguments. Strategies SE and IE are more applied for routine design tasks, and FE and VE for non-routine ones.

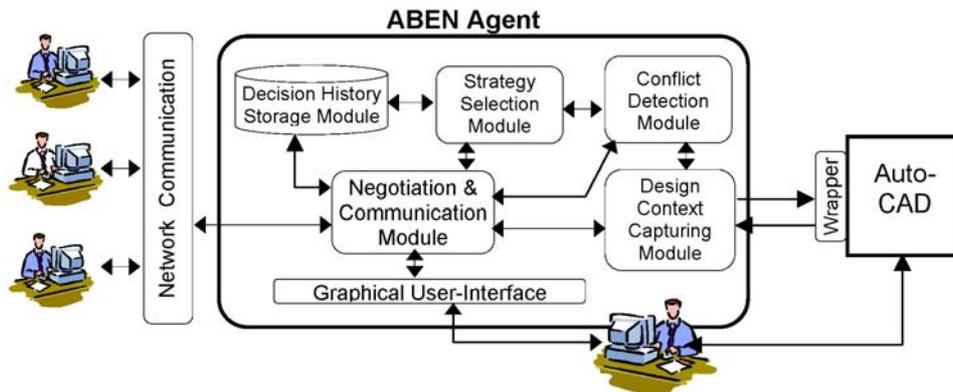
Execution of the first three strategies, SE, IE, and FE allows designers to make *collaborative design decisions* where the designers can find common grounds in FR, issues, and solutions, while maintaining their original preferences. On the other hand, exercising VE strategy promotes designers to make *joint design decisions* in which designers align their objectives and attempt to work out common decisions for all parties with shared objectives and criteria. From a collaboration activity's point of view, the first two strategies, SE and IE, facilitates *cooperation* between designers, whereas the other two, FE and VE, encourage more *co-construction*. Exploration of FR and design objectives may lead to redefinition of the design task and yield innovative design results, as will be shown in the following case study.

#### 4 Case study

The ABEN framework described above is implemented based on AutoCAD and an agent based collaboration support system (Jin and Zhou, 1999). Besides using AutoCAD to perform design tasks, each designer has his or her own computer agent, called ABEN-Agent that facilitates the negotiation process. Figure 5 illustrates the prototype

system architecture. A communication wrapper is attached to AutoCAD for linking to the agent. Following is a brief description of the main modules.

**Figure 5** ABEN system architecture (see online version for colours)



*Design context capturing module.* Collects and stores the relevant design context information connected to decision points. This module monitors design entities in the CAD system and keeps track of the dependencies between the local task and the other designers' activities through the communication module.

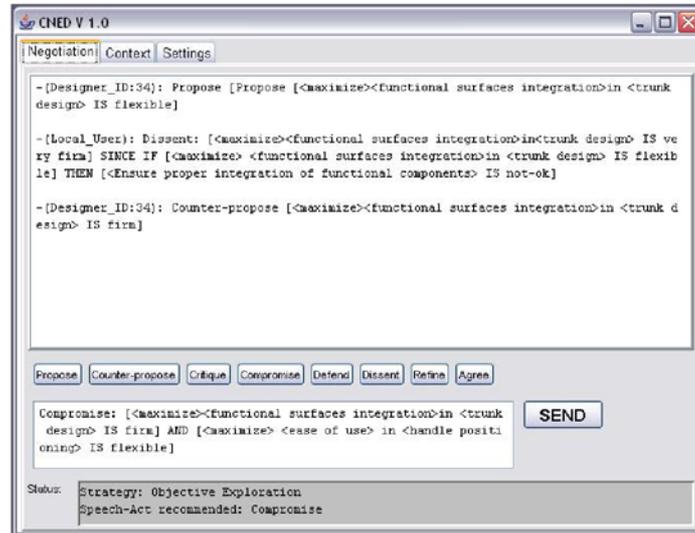
*Conflict detection module.* Identifies uncoordinated and opposing decisions made by different engineers. This module detects inconsistencies directly based on the information collected by the design context capturing module.

*Strategy generation module.* Uses a rule-based approach to generate appropriate strategy based on the design context information collected, the type of conflict identified and the design history (whether the parties have unsuccessfully tried to solve the conflict earlier or not)

*Negotiation and communication module.* Implements the ABEN protocol and argument model. This module is the heart of the negotiation automation, and contains the argument handling mechanism.

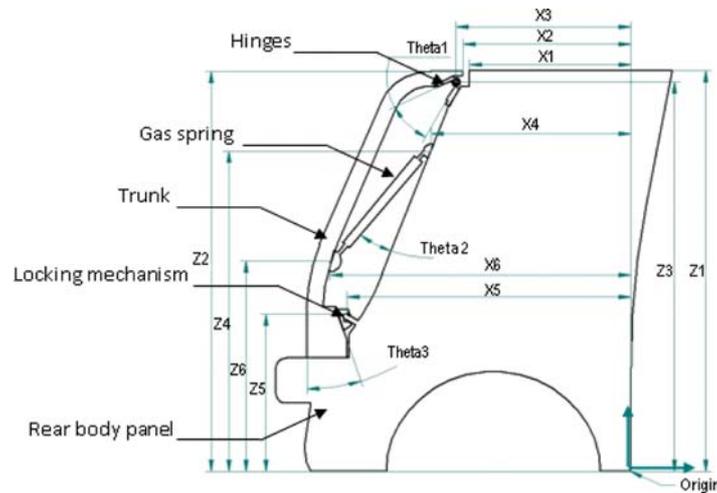
*Decision history storage module.* Keeps track of the design decision history, and determine what strategy can be implemented successfully based on the design decision history, or the way similar situations encountered in the past have been successfully solved.

*Graphical User Interface (GUI).* Provides dialogue boxes for the designer to interact with the ABEN agent and other designers through the agent. Figure 6 is a screenshot of the communication window of the GUI. The top text pane contains incoming and outgoing messages, while the bottom pane consists of a text input window and a status bar that show the outgoing message and current strategy suggested by the agent.

**Figure 6** Negotiation window of the ABEN agent GUI (see online version for colours)

#### 4.1 The design problem

The design problem for our case study is configuration of a hinge mechanism used to open the trunk on a hatchback vehicle. The problem involves three designers each working on a different subtask of the design. Figure 7 illustrates the overall configuration of the trunk system in a 2D drawing that indicates the locking mechanism, gas spring, and hinge in a common plane.

**Figure 7** Configuration of the trunk design problem (see online version for colours)

In this design problem, several requirements must be considered. First, the external shape of the trunk must match the lines determined by the style department. In addition, the trunk needs to incorporate multiple functional surfaces on its inner surface to achieve various functions, such as enclosing the locking mechanism, providing support surfaces for the hinges, as well as fixing points for the gas springs. This challenge is resolved by using a hollow part made of the assembly of an inner and an outer panel. The inner panel contains all the functional surfaces while the outer shell respects the style. The second requirement is related to the ergonomics of the trunk opening, and more specifically the kinematics of the opening movement. The trunk should be easily opened by customers of average size and force; therefore, the forces to apply on the trunk handle at opening and closing must be controlled. These forces depend strictly on the geometry of the system (i.e. where the gas springs are fixed and where the hinges are located), and the force of the gas springs selected. Finally, the proper alignment and match with the rest of the vehicle's body must be achieved. The transitions have to be smooth on the outer parts, while the functional surfaces have to be in proper positions to guarantee that the FR will be addressed.

We chose this design problem because it involves three main parts with high dependencies between them that require designers to cooperate for resolving potential design conflicts and to co-construct for generating new design alternatives. Three designers are involved in this example:

*Designer A* is in charge of completing the trunk design. This task includes designing the inner and outer shells of the trunk, as well as ensuring that the functional surfaces are properly positioned to receive the locking mechanism, hinges, and the gas springs. In addition, the trunk in a closed position must be waterproof, which constitutes another constraint on the model, i.e. the sealing joint must be pressed in a homogeneous way (i.e. the surfaces of the trunk pressing against it must be perpendicular to it). Design A is responsible for X2, Z2, Theta3, X6, and Z6.

*Designer B* is in charge of the kinematics and ergonomics of the trunk. She is responsible for the positioning of the gas springs, the trunk handle (we consider that the grabbing point is the same for both opening and closing), and the hinges. She has to ensure that the forces required to open and close the trunk meet the ergonomics FR. Designer B is responsible for Theta1, Theta2, X3, Z3, X5, Z5, and L1.

*Designer C* is in charge of modelling the rear body of the vehicle, in particular the interface with the trunk, the hinges surface, the alignment of the locking mechanism with the hook fixed on the body, as well as the positioning of the sealing joint. Designer C's responsible parameters are X1, Z1, X4, and Z4.

Table 2 shows the FR and DC considered by each designer for their design tasks.

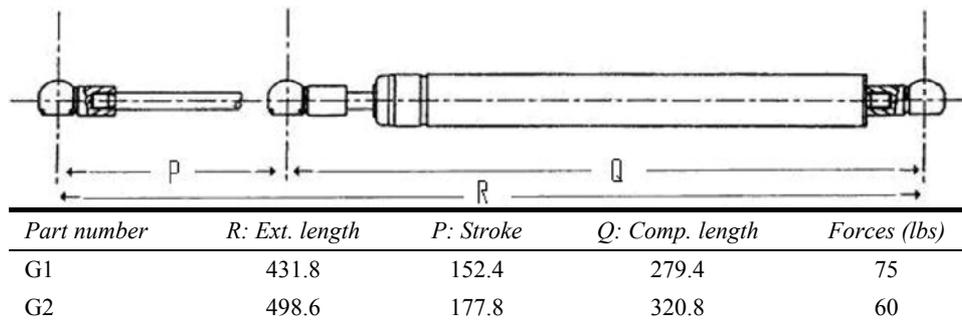
To keep the design problem simple, we further assume that Designer B can choose a gas spring only from the following two alternatives, as shown in Figure 8. Therefore, the only two configurations possible for Designer B are  $L = 279.4$  mm or  $L = 320.8$  mm. The positions of the fixing points  $P_4 = (X_4, Z_4)$  and  $P_6 = (X_6, Z_6)$  are dependent and directly related to the length of the gas spring  $L = \sqrt{(X_6 - X_4)^2 + (Z_4 - Z_6)^2}$ . Designer B must make a selection here, in order to complete his task.

In the following, we present three realist application scenarios of our framework to this problem. The first scenario describes the situation in which potential conflicts are at the parameter value level and ZOPA is attainable. The second scenario illustrates a situation where no ZOPA exists at lower level and higher-level ZOPA must be identified and negotiated. In the last scenario, we describe a *co-constructive* negotiation in which higher-level negotiation yields new design ideas.

**Table 2** Functional requirements and design objectives

<i>Designer</i>	<i>Functional requirements</i>	<i>Design constraints</i>
A	Provide soundproofness Provide clear rear visibility under any conditions Ensure proper integration of functional components Prevent access to interior space when desired for safety of belongings Ensure smooth continuity between body and trunk	Provide opening angle of 85 deg Maintain rear sightline to 12 feet for an average adult driver Maintain trunk weight below 20 lbs
B	Maintain waterproof-ness between car body and trunk lid Provide easy access to rear cargo space	Limit force needed at opening to 40 N Limit force needed at closing to 35 N Limit self-opening angle <sup>1</sup> to 40 deg Limit self-closing angle <sup>2</sup> to 20 deg 1: angle beyond which the trunk needs no external force to open 2: angle beyond which the trunk needs no external force to close
C	Ensure proper integration of functional components Control hinge position to avoid interferences between trunk top part and rear top body Optimise fixation point position to avoid interferences with other parts	Limit displacement of fixation points under loading to 0.05 mm Provide 5 mm adjustment space for hinge positioning

**Figure 8** Suitable gas springs for designer B's task



#### 4.2 Parametric value level negotiation with ZOPA (Scenario 1)

Consider the task of *designer B* of setting up the gas springs in a proper configuration to respect the “provide easy access to rear cargo space” requirement and the DC. We assume that the position of the hinges has been decided beforehand. At parameter value level, i.e., *PV* level in equation (2), *designer B* needs to set proper values for the following parameters:  $X_4$ ,  $Z_4$ ,  $X_6$  and  $Z_6$ . In addition, he needs to select suitable values for the gas spring characteristics, i.e. its length, stroke and force.

*Designer B* sends a request for the creation of point P6 at (510, 295) to *designer A* and point P4 to *designer C*. Let us assume that *designer A* cannot create the required fixation point. He contacts *designer B* to try to resolve the issue. Faced with a situation where no obvious solution is suitable to all the designers at the same time, *designer A*, who is unable to complete the task requested by *designer B*, will contact him to solve the problem.

The objective of both parties is to generate a ZOPA that will help them to solve this qualitative conflict. This ZOPA is actually two dimensional as both x and z coordinates are to be decided upon. Following is the monitored negotiation process with manual editing for easy understanding.

Designer A: Propose [ $X_6$  IS 530 mm] & [ $Z_6$  IS 295 mm]

Designer B: Critique NOT [ $X_6$  IS 530 mm] AS [need more information]

Designer A: Defend [ $X_6$  IS 530 mm] SINCE IF [<DP>  $X_6$  IS 510 mm] AND [ $Z_6$  IS 295 mm] THEN [ $P_6$  IS NOT on Curve C1] AND [<maximise> <functional surfaces integration> in <trunk design> IS not- ok]

Designer B: Counter-Propose [ $P_6$  IS on Curve C1] AND [ $Z_6$  IS >265 mm and <285 mm]

Designer A: Critique [ $Z_6$  IS >265 mm and <285 mm] AS [Need Justification]

Designer B: Defend [ $Z_6$  IS >265 mm and <285 mm] SINCE IF [Gas Spring A1 IS selected] AND [ $Z_6$  IS <265 mm or >285 mm] THEN [Limit force needed at opening to 40N IS not-ok]

Designer A: Compromise [ $P_6$  IS on Curve C1] AND [ $Z_6$  IS >280 mm]

Designer B: Agree [ $P_6$  IS on Curve C1] AND [ $Z_6$  IS >280 mm]

Designer B: Propose [ $Z_6$  IS 280 mm]

Designer A: Agree [ $Z_6$  IS 280 mm].

The parties agreed on a final value determined by the intersection of a curve  $C_1$  and a parallel plane at  $Z_6 = 280$  mm. The agreement is the result of the determination of a ZOPA identified after *designer A* compromises to a  $Z_6$  superior or equal to 280. The ZOPA is thus [280, 285] and at the first iteration, the value of 280 is accepted as an acceptable value for both parties.

In addition, the negotiation process could have taken another direction when *designer B* invokes the DC in a warrant. Indeed, *designer A* could have generated an argument including a claim about this DC, such as:

Designer A: Propose [Limit force opening force needed IS 50N]

He could also have suggested using the other gas spring:

Designer A: Propose [Gas Spring A2 IS selected]

Therefore multiple solutions can be generated depending on which direction the parties are taking in their communication exchange. In the next scenario, we look into non-quantitative issues.

#### 4.3 Objective level negotiation with ZOPA (Scenario 2)

In this scenario we demonstrate how ZOPA at higher and qualitative level can be explored and recognised for reaching an agreement. Let us assume a situation where *designer B* is attempting to fulfill the following design objectives:

<maximise> <ease of use> in <handle positioning>

<maximise> <ease of use> in <gas spring configuration>

Assume further that the only attainable gas spring configuration, after negotiation with *designers A* and *C*, is such that the handle has to be positioned lower than previously expected. As a result the user would have to reach down. Although the forces needed would be maintained within the boundaries set by the DC, the awkward user's arm position renders the maneuver actually more difficult.

In order to solve this deadlock, *designer B* initiates a negotiation process with *designer A* over their respective design objectives, in the hope of influencing other's design objectives to reach a compromise situation. As described above, to facilitate qualitative ZOPA negotiation, we introduce a set of semi-quantitative qualifiers to provide a metric over which we can discuss the respect of unquantifiable elements: *very firm*, *firm*, *flexible*, and *very flexible*

Designer B: Propose [<maximise><functional surfaces integration>in <trunk design> IS flexible]

Designer A: Dissent [<maximise><functional surfaces integration>in <trunk design> IS very firm]  
 SINCE IF [<maximise> <functional surfaces integration>in <trunk design> IS flexible]  
 THEN [<Ensure proper functional integration> IS not-ok]

Designer B: Counter-propose [<maximise><functional surfaces integration>in <trunk design> IS firm]

Designer A: Compromise [<maximise><functional surfaces integration>in <trunk design> IS firm] AND [<maximise><ease of use> in <handle positioning> IS flexible]

Designer B: Critique [<maximise><ease of use> in <handle positioning> IS flexible] AS [Need Justification]

Designer A: Defend [<maximise><ease of use> in <handle positioning> IS flexible] SINCE IF [<maximise><ease of use> in <handle positioning> IS flexible] THEN [<Move trunk handle upwards> IS ok] AND [move P<sub>6</sub> up to Z = 285 mm]

*Designer A* suggests that relaxing the design objective to a lower level of importance will enable the designers to create a new work element, moving the handle position up to Z<sub>6</sub> = 285 mm (previously adopted as a boundary of the ZOPA at the parameter value level), and resolving the issue (the handle reaching a more suitable position). The designers, through the exchange of arguments are able to create the equivalent of a ZOPA by conceding on the stringency of their design objectives. *Designer B* consults *designer C* to determine how high the handle can be moved while maintaining loosely the design objective on forces to apply (*designer C* will move his fixation point P<sub>6</sub> up to maintain somehow the efforts values).

Designer B: Agree and Acquire information: [Designer C]  
 Designer A: *wait*.

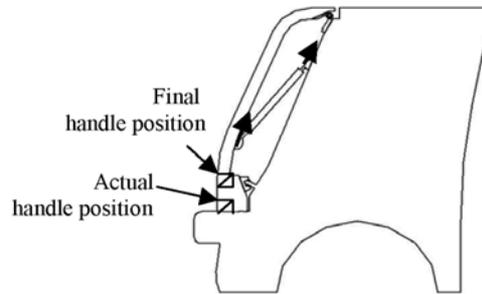
Reaching a satisfactory value for the handle position with *designer C*, *designer B* comes back to finish the process with *designer A*

Designer B: Refine [<maximise> <ease of use> in <handle positioning> IS flexible] and [<Move trunk handle upwards by 110 mm> IS ok]

Designer A: Agree [<Move trunk handle upwards by 110 mm> IS ok]

Designer B: Agree

Figure 9 illustrates the result of the negotiation process over design objectives on the design itself. The gas spring and the handle are both moved up as *designers A* and *B* compromise on their design objectives. The resolution of high-level conflicts led to a solution at the bottom parameter value level.

**Figure 9** Result of negotiation over design objectives (Scenario 2)

#### 4.4 *Co-constructive negotiation for innovative ideas (scenario 3)*

Let us consider the case example used in Scenario 1, and assume that no ZOPA is found through the exploration at the solution level. As a result, another approach has to be used.

Designer A: Propose [ $X_6$  IS 530 mm] AND [ $Z_6$  IS 295 mm]

Designer B: Critique NOT [ $X_6$  IS 530 mm] AS [need more information]

Designer A: Defend [ $X_6$  IS 530 mm] SINCE IF [ $X_6$  IS 510 mm] AND [ $Z_6$  IS 295 mm] THEN [ $P_6$  IS NOT on Curve C1] AND [<maximise> <functional surfaces integration> in <trunk design> IS not-ok]

Designer B: Counter-Propose [ $P_6$  IS on Curve C1] AND [ $Z_6$  IS > 265 mm and < 285 mm]

Designer A: Critique [ $Z_6$  IS >265 mm and <285 mm] AS [Need Justification]

Designer B: Defend [ $Z_6$  IS >265 mm and <285 mm] SINCE IF [Gas Spring A1 IS selected] AND [ $Z_6$  IS <265 mm or >285 mm] THEN [Limit force needed at opening to 40N IS not-ok]

At this point, the parties must try to solve the problem using a strategy that will help them go beyond the purely parameter-related arguments. For this, they are going to successively inquire about the reasons behind their requests and soon deal with the design objectives.

Designer A: Critique [Gas Spring A1 IS selected] AS [Need Justification]

Designer B: Defend [Gas spring A1 IS selected] AS [Inner panel weight  $W_1 = 5.3$  lbs] AND [outer panel weight  $W_2 = 8.5$  lbs] AND SINCE [IF [Total trunk weight IS > 10 lbs and < 20 lbs] THEN [<DP> gas spring force needed IS 75 lbs]]

Designer A: Critique [IF [Total trunk weight IS >10lbs and <20 lbs] THEN <DP>gas spring force needed IS 75 lbs]] SINCE [Need Justification]

Designer B: Defend [IF [Total trunk weight IS >10 lbs and <20 lbs] THEN [gas spring force needed IS 75 lbs]] SINCE [<maximise> <ease of use> in <gas spring positioning> IS very firm]]

Understanding that the problem is related to the weight of the trunk part, the designers can try to generate ideas to reduce this weight:

Designer B: Propose [Inner trunk panel material IS Aluminum] and [Outer trunk panel material is aluminum] AS IF [Inner trunk panel material is Aluminum] AND [Outer trunk panel material is aluminum] THEN [Trunk weight IS reduced]

Designer A: Dissent NOT [Inner trunk panel material IS Aluminum] and NOT [Outer trunk panel material is aluminum] AS IF [Inner trunk panel material is aluminum] AND [Outer trunk panel material is aluminum] THEN [<minimise cost> in <trunk design> IS not-ok]

After trying to modify the material of the part which is also a parameter-level type of solution, and realising that the issue is not resolved, the designers can backtrack the decision making process and reconsider choices made earlier, while looking at functional-level objectives. Thus far, they have clarified that *designer B* needs gas spring A1 to achieve the proper opening angle and to satisfy requirements on the opening and closing forces despite the weight of the trunk and its dimensions.

Designer B: Critique NOT [T1 IS Design a 2-panel assembly] AS IF [T1 IS Design a 2-panel assembly] THEN [<maximise> <ease of use> in <gas spring positioning> IS not-ok]

Designer A: Agree NOT [T1 IS Design a 2-panel assembly]

Designer B: Propose [T1 IS Design a 2-part trunk] AS IF [T1 IS Design a 2-part trunk] THEN [Minimise weight in trunk design IS ok]

Designer A: Refine [T1 IS Design a 2-part trunk] AND [T1-1 IS Design trunk parts] AND [T1-2 IS Design opening mechanisms]

*Designers A* and *B* have put their ideas and objectives in perspective and generated a new solution by backtracking and revising earlier decisions. To conclude, they need to make choices about the downstream decision points: how to design the new parts, and the opening/closing mechanism. Depending on *designer C*'s objectives, and the DC he has to follow, the outcome may differ.

*Solution 1:* Assume that *designer C* has the same constraints as earlier, but that given the weight change of the upper trunk part, a smaller gas spring, fixed lower (i.e. within *designer C*'s range), can perform the desired function. *Designer B* is currently exchanging arguments with *designer A*, but needs to contact *designer C* to ensure that this new configuration will suit his objectives.

```

Designer B Propose [T1-2-1 IS Select gas spring] AND
           [T1-2-2 IS Determine fixation points]
Designer A Agree [T1-2-1 IS Select gas spring] AND
           [T1-2-2 IS Determine fixation points]
Designer B Acquire_Information [Designer C]
Designer A wait

```

At this point, *designer B* contacts *designer C* to negotiate over the possibility of fixing a smaller gas spring in a new position. *Designer C* accepts and *designer B* comes back to conclude the negotiation with *designer A*.

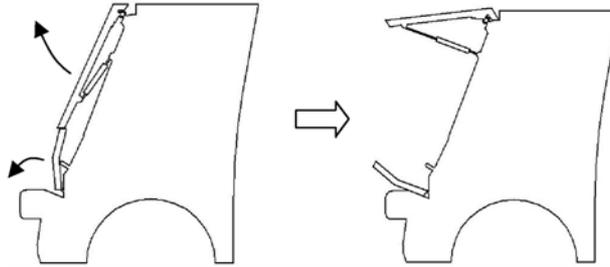
```

Designer B: Agree [T1-2-1 IS Select gas spring] AND
           [T1-2-2 IS Determine fixation points]
Designer A: Agree [T1-2-1 IS Select gas spring] AND
           [T1-2-2 IS Determine fixation points]

```

Figure 10 represents the solution generated by this negotiation process.

**Figure 10** Solution with a 2-part trunk with sliding upper part (Scenario 3)



*Solution 2:* Let us assume that *designer C* needs to work with the same constraints as before, but that no other surface is available to position the gas springs' fixation points. Therefore, he rejects the proposition of *designer B* to position a smaller gas spring on his part. *Designer B* comes back to the negotiation table with *designer A* and attempts to generate another solution.

```

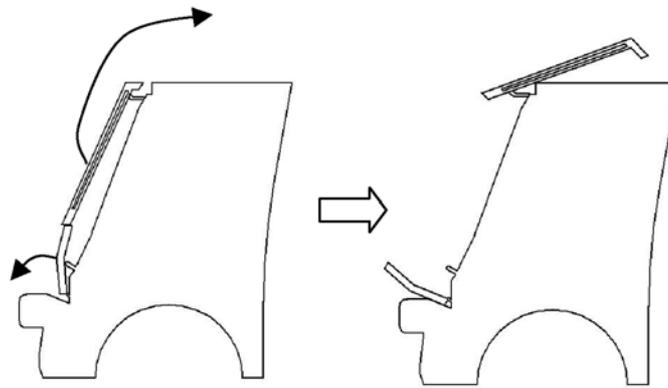
Designer B: Acquire_Information [Designer C]
Designer A: wait
Designer B: Dissent NOT [T1-2-1 Select gas spring] AND
           [T1-2-2 IS Determine fixation points] AS IF
           [T1-2-2 Determine fixation points] THEN
           [maximise fixing point integration in rear
           body design IS not-ok]

```

- Designer A: Compromise [T1-2-1 IS Create sliding mechanism] and [T1-2-2 IS Determine sliding mechanism fixation point]
- Designer B: Agree [T1-2-1 IS Create sliding mechanism] and [T1-2-2 IS Determine sliding mechanism fixation point]

Figure 11 is a representation of such a solution, the top part of the trunk incorporates a cavity on its sides, in which a pin may slide freely and guide the part to stack it above the roof. The bottom part is identical to the previous solution.

**Figure 11** Solution with a 2-part trunk with hinged upper part (Scenario 3)



*Designers A and B*, through the exchange of arguments containing design context information, and design objectives, have been able to overcome their initial parameter-level problem, by generating new design task and solutions, and focusing on sharing the design context information.

## 5 Concluding remarks

Collaborative engineering design involves coordination, cooperation and co-construction among multiple team members. For complex design problems where design subtasks cannot be clearly separated and dependencies are too complex to be sorted out, how to make cooperation and co-construction effective and efficient is a major challenge for the design research community. To meet this challenge, we proposed an ABEN framework that allows designers to fully present their point of views toward the problem, their ways to approach the problem and their potential solutions. The mechanism that supports this kind of high level and flexible design interactions is composed of a multilevel issue model, a structured argument model, a speech-acts based negotiation protocol, and a set of negotiation strategies. The basic idea behind ABEN is that for two designers to reach an agreement on an outstanding issue, there must be a ZOPA of this issue between the designers. If ZOPA cannot be found at the level of the outstanding issue, then the designers should identify relevant issues at a higher level and try to find ZOPA for those higher level issues. Once a ZOPA is found at a higher-level, the agreement can be

reached, and new design alternatives can be developed at lower levels. The recognition of the existence of ZOPA and the flexibility of moving between levels of issues made it possible for ABEN to provide effective support for extensive cooperation and co-construction among designers.

ZOPA is not limited to quantitative issues. Our case study demonstrated that for qualitative issues, there still can be ZOPA. Through passing arguments, designers can eventually find ZOPA and reach agreement. The three scenarios discussed in the case study demonstrated that ABEN is not only effective for simple quantitative ZOPA solutions but also capable of facilitating negotiation on problems with qualitative ZOPAs. Furthermore, it is demonstrated that multi-level negotiation can lead to new design ideas.

One major limitation of our current ABEN system is that it depends on designers to interpret and compose arguments. Agents provide argument templates but little content. This low level automation poses a big communication overhead on designers. Our current research attempts to enrich the DCM so that computer agents will be able to understand the design context and provide assistance for designers by *participating* in negotiation.

## References

- Antonsson, E. and Otto, k.N. (1995) 'Imprecision in engineering design', *ASME Journal of Mechanical Design*, Vol. 117, No. B, pp.25–32.
- Ballmer, T. and Brennenstuhl, W. (1981) *Speech Act Classification: A Study in the Lexical Analysis of English Speech Activity Verbs*, Springer-Verlag, Berlin.
- Bond, A.H. and Gasser, L. (1988) 'An analysis of problems and research in DAI', in Bond, A.H. and Gasser, L. (Eds.): *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, pp.3–35.
- Chen, L. and Li, S. (2002) 'A computerized team approach for concurrent product and process design optimization', *Computer-Aided Design*, Vol. 34, No. 1, pp.57–69.
- Cutkosky, M.R., Engelmores, R.S., Fikes, R.E., Genesereth, M.R., Gruber, T.R. and Mark, W.S. (1993) 'PACT: an experiment in integrating concurrent engineering systems', *IEEE Computer*, Vol. 26, No. 1, pp.28–37.
- Danesh, M. and Jin, Y. (2001) 'An agent-based decision network for concurrent engineering', *Journal of Concurrent Engineering Research and Application*, Vol. 9, No. 1, pp.37–47.
- Engeström, Y (1987) *Learning by Expanding: An Activity-theoretical Approach to Developmental Research*, Orienta-Konsultit Oy., Helsinki.
- Eppinger, S.D., Whitney, D.E., Smith, R.P. and Gebala, D.A. (1994) 'A model-based method for organizing tasks in product development', *Research in Engineering Design*, Vol. 6, pp.1–13.
- Genesereth, M.R. and Katchpel, S.P (1994) 'Software agents', in *Communications of the ACM*, Vol. 37, No. 7, pp.48–53.
- Geslin, M. and Jin, Y. (2005) 'Design context based negotiation for collaborative engineering', Submitted to *17th ASME International Conference on Design Theory and Methodology*, 25–29 September, Long Beach, CA.
- Gulliver, P.H. (1979) *Disputes and Negotiations: A Cross-Cultural Perspective*, Academic Press, New York.
- Jennings, N.R., Parsons, S., Noriega, P. and Sierra, C. (1998) 'On argumentation-based negotiation', *Proceedings of the International Workshop on Multi-Agent Systems*, Boston, USA, pp.1–7.

- Jin, Y. and Levitt, R.E. (1996) 'The virtual design team: a computational model of project organizations', *Computational and Mathematical Organization Theory (Netherlands)*, Vol. 2, No. 3, pp.171–196.
- Jin, Y. and Lu, S. (1998) 'An agent-supported approach to collaborative design', *CIRP Annals*, Vol. 47, No. 1, pp.107–110.
- Jin, Y. and Zhou, W. (1999) 'Agent-based knowledge management for collaborative engineering', *Proceedings of 19th ASME Computers in Engineering Conference*, DETC99/CIE-9022, 12–15 September, Las Vegas, Nevada
- Jin, Y., Zhao, L. and Raghunath, A. (1999) 'ActivePROCESS: a process-driven and agent-based approach to collaborative engineering support', *Proceedings of 19th ASME Computers in Engineering Conference*, DETC99/CIE-9099, 12–15 September, Las Vegas, Nevada, pp.1–11.
- Joseph, J., Phillips, C.R., Davis, E.W.M., Edward, W.D. and Cecil, R.P. (1995) 'Project management with Cpm', *Pert and Precedence Diagramming*, Blitz Pub Co.
- Klein, M. (2000) 'Towards a systematic repository of knowledge about managing collaborative design conflicts', *Proceedings of the International Conference on AI in Design*, Boston, MA.
- Kodiyalam, S. and Sobieszczanski-Sobieski, J. (2001) 'Multidisciplinary design optimization – some formal methods, framework requirements, and application to vehicle design', *Int. J. Vehicle Design (Special Issue)*, pp.3–22.
- Kraus, S. and Lehmann, D. (1995) 'Designing and building a negotiating automated agent', in *Communication Intelligence*, Vol. 11, No. 1, pp.132–171.
- Kroo, I., Altus, S., Braun, R., Gage, P. and Sobieski, I. (1994) 'Multidisciplinary optimization methods for aircraft preliminary design', *AIAA 94-4325, Fifth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 7–9 September, Panama City, Florida.
- Lee, J., Gruninger, M., Jin, Y., Malone, T., Tate, A. and Yost, G. (1998) 'The PIF process interchange format and framework', *Knowledge Engineering Review*, Vol. 13, No. 1, pp.91–120.
- Leontjev, A. (1978) *Activity, Consciousness and Personality*, Prentice-Hall, Englewood Cliffs.
- Levitt, R.E., Cohen, G., Kunz, J., Nass, C., Christiansen, T. and Jin, Y. (1993) 'The virtual design team: simulating how organization structure and information processing tools affect team performance', in Carley, K. and Prietula, M. (Eds): *Computational Organization Theory*, Autumn, Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp.1–18.
- Lewis, K. and Mistree, F. (1998) 'Collaborative, sequential and isolated decisions in design', *Journal of Mechanical Design*, Vol. 120, pp.643–652
- Lu, S. C-Y. and Cai, J. (2001) 'A collaborative design process model in the socio-technical engineering design framework', *Journal of Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Vol. 15, pp.3–20.
- Michel, J.J. and Cutting-Decelle, A.F. (2004) 'The process specification language', *International Standards Organization ISO TC184/SC5 Meeting*, April, Paris.
- Park, H. and Cutkosky, M.R. (1999) 'Framework for modeling dependencies in collaborative engineering processes', *Research in Engineering Design*, Vol. 11, pp.84–102.
- Parsons, S., Sierra, C. and Jennings, N.R. (1998) 'Agents that reason and negotiate by arguing', *Journal of Logic and Computation*, Vol. 8, No. 3, pp.261–292.
- Pena-Mora, F. and Wang, C.Y. (1998) 'Computer-supported collaborative negotiation methodology', *Journal of Computing in Civil Engineering*, Vol. 12, No. 2, pp.64–81.
- Pena-Mora, F., Sriram, R.D. and Logcher, R. (1996) 'Conflict mitigation system for collaborative engineering', *(AI EDAM) Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 9, No. 2, pp.101–124.

- Petrie, C. (1993) 'The Redux' server', *Proceedings of the International Conference on Intelligent and Cooperative Information Systems (ICICIS)*, Rotterdam.
- Pruitt, D.G. (1981) *Negotiation Behaviour— Organizational and Occupational Psychology*, Academic Press, New York.
- Raiffa, H. (2003) *Negotiation Analysis*, Belknap Press, Cambridge, MA, USA.
- Rosenschein, J. and Zlotkin, G. (1994) *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*, MIT Press, Cambridge MA, USA.
- Scott, M. (1997) *Formalizing Negotiation in Engineering Design*, PhD Thesis, California Institute of Technology.
- Smith, R.G. (1980) 'The contract net protocol: high-level communication and control in a distributed problem solver', *IEEE Transactions on Computers*, Vol. C-29, No. 12, pp.1104–1113.
- Sobieszcanski-Sobieski, J., Agte, J. and Sandusky Jr., R. (2000) 'Bi-Level Integrated System Synthesis (BLISS)', *AIAA Journal*, Vol. 38, No. 1, pp.164–172.
- Sriram, D. and Logcher, R. (1993) *The MIT DICE project*, *Computer*, Vol. 26, No. 1, January. pp.64–65.
- Steward, D.V. (1981) 'The design structure system: a method for managing the design of complex systems', *IEEE Transaction on Engineering Management*, EM-Vol. 28, No. 3, pp.71–74
- Suh, N.P. (1990) *The Principles of Design*, Oxford University Press, New York.
- Sycara, K.P. (1989) 'Multiagent compromise via negotiation', in Gasser, L. and Huhns M. (Eds.): *Distributed Artificial Intelligence*, Vol. 2, Pitman Publishing, London, pp.119–137.
- Toulmins, S.E. (1969) *The Uses of Argument*, Cambridge University Press, Cambridge, UK.
- Xiao, A., Choi, H.J., Allen, J.K., Rosen, D. and Mistree, F. (2002) 'Collaborative decision making across digital interfaces', *Proceedings of ASME Design Automation Conference*, Montreal, Canada, DETC2002/DAC-34073.

## Bibliography

- Danesh, M.R. (2001) *A Framework for Value-based Conceptual Engineering Design*, Doctoral thesis, University of Southern California, Los Angeles, USA.
- Engeström, Y. (1997) 'Coordination, cooperation and communication in the courts' *Mind, Culture, and Activity*, Cambridge University Press, Chapt. 28, pp.369–388, Cambridge, UK.
- Eppinger, S.D., Nukala, M.V. and Whitney, D.E. (1997) 'Generalised models of design iteration using signal flow graphs', *Research in Engineering Design*, Vol. 9, pp.112–123.
- Keeney, R.L. (1992) *Value-Focused Thinking, A Path to Creative Decisionmaking*, Harvard University Press, Cambridge, Massachusetts.
- Krothapalli, N.K.C. and Deshmukh, A.V. (1999) 'Design of negotiation protocols for multi-agent manufacturing systems', *International Journal of Production Research*, Vol. 37, No. 7, pp.1601–1624.
- Sandholm, T. (2002) *Algorithm for Optimal Winner Determination in Combinatorial Auctions*, *Artificial Intelligence*, Vol. 135, pp.1–54
- Scott, M.J. and Antonsson, E.K. (1998) 'Aggregation functions for engineering design trade-offs', *Fuzzy Sets and Systems* (U.S.A.) Vol. 99, pp.253–264.
- Sellar, R.S., Batill, S.M. and Renaud, J.E. (1996) 'Response surface based, concurrent subspace optimization for multidisciplinary design optimization', *AIAA 96-0714, 34th Aerospace Sciences Meeting and Exhibit*, Reno, NV.
- Sobieszcanski-Sobieski, J. (1988) 'Optimization by decomposition: a step from hierarchic to nonhierarchic systems', *Proceedings, 2nd NASA/USAF Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, NASA CP-3031, Hampton, Virginia,. Also, NASA TM-101494.