



Missouri University of Science and Technology
Scholars' Mine

Computer Science Faculty Research & Creative Works

Computer Science

01 May 2007

Management of an Intelligent Argumentation Network for a Web-Based Collaborative Engineering Design Environment

Xiaoqing Frank Liu

Missouri University of Science and Technology, fliu@mst.edu

Man Zheng

Ganesh K. Venayagamoorthy

Missouri University of Science and Technology

Ming-Chuan Leu

Missouri University of Science and Technology, mleu@mst.edu

Follow this and additional works at: https://scholarsmine.mst.edu/comsci_facwork



Part of the [Aerospace Engineering Commons](#), [Computer Sciences Commons](#), [Electrical and Computer Engineering Commons](#), and the [Mechanical Engineering Commons](#)

Recommended Citation

X. F. Liu et al., "Management of an Intelligent Argumentation Network for a Web-Based Collaborative Engineering Design Environment," *Proceedings of 2007 International Symposium on Collaborative Technologies and Systems*, Institute of Electrical and Electronics Engineers (IEEE), May 2007.

The definitive version is available at <https://doi.org/10.1109/CTS.2007.4621732>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Computer Science Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

Management of an Intelligent Argumentation Network for a Web-Based Collaborative Engineering Design Environment

Xiaoqing (Frank) Liu¹, Man Zheng¹, Ganesh K Venayagamoorthy³, and Ming Leu²

¹Department of Computer Science, University of Missouri-Rolla, Rolla, MO-65401, USA

²Department of Mechanical and Aerospace Engineering, University of Missouri-Rolla, Rolla, MO-65401, USA

³Real-Time Power and Intelligent Systems Laboratory, Department of Electrical & Computer Engineering, University of Missouri-Rolla, Rolla, MO-65401, USA

ABSTRACT

Conflict resolution is one of the most challenging tasks in collaborative engineering design. In our previous research, a web-based intelligent collaborative system was developed to address this challenge based on intelligent computational argumentation. However, two important issues were not resolved in that system: priority of participants and self-conflicting arguments. In this paper, we develop two methods for incorporating priorities of participants into the computational argumentation network: 1) weighted summation and 2) re-assessment of strengths of arguments based on priority of owners of the argument using fuzzy logic inference. In addition, we develop a method for detection of self-conflicting arguments. Incorporation of priority of participants and detection of self-conflicting arguments have strengthen the capability of managing intelligent argumentation network for the web-based collaborative engineering design system developed in our previous research.

KEYWORDS: Computational Argumentation, Fuzzy Logic, Inference Engine, Conflict Resolution, Self-Conflicting, Priority Assessment

1. INTRODUCTION

A web-based collaborative engineering design system enables people to have discussions together at the same time while working simultaneously in different locations. It large facilitates, modern product design is a complex process involving multiple roles such as designers, manufacturers, suppliers, and customer representatives. In

our previous intelligent collaborative engineering design network, intelligent argumentation [1] was used to resolve design conflict effectively. However, several important issues in intelligent argumentation were not addressed in the system. One of them was that priorities of stakeholders (participants) were not considered. A participant who is more experienced and knowledgeable should carry more weight in conflict resolution based on argumentation. It is necessary to incorporate priority of participants into management of the intelligent argumentation network for conflict resolution in the collaborative engineering design system.

Another important issue in the intelligent argumentation network management is self-conflicting. Sometimes in a complex network a few participants are very active to offer their opinions. For example, participant 1 has an argument A in the network. Participant 2 has an argument B which supports argument A. Participant 3 attacks argument B with argument C. At this point, participant 1 may find that argument C is reasonable and he (she) will support argument C. Unfortunately, it turns out that participant 1 indirectly attacks himself. In a complicated network that has a large number of arguments and participants, this kind of self-conflicting is hard to detect manually. Automatic detection and removal of self-conflicts will significantly improve the robustness of the network.

The main contribution of this paper is to incorporate priority of participants to a web-based intelligent collaborative system to improve the accuracy and effectiveness of the previous developed system and to detect self-conflicting arguments to order to improve the robustness of the network. This paper addresses the above two issues. The paper is organized as follows. Section 2 reviews related works. Section 3 gives a brief introduction

of our previous intelligent collaborative engineering design system based on intelligent argumentation. Section 4 explains how to incorporate priority into the system. Section 5 describes the self-conflicting argumentation and how to detect it. In section 6, we present quantitative analysis of an argumentation network. In section 7 we discuss implementation of the new features in our system in Java.

2. RELATED WORK

Philosopher Stephen Toulmin [2] developed a very influential model of argumentation that has guided the development of software tools and systems that are intended to support the detection and resolution of conflicts in many knowledge domains. Sillince [3] proposed a more general argumentation model. His model is a logic model where dialogs are represented as recursive graphs and both rhetoric and logic rules are used to manage the dialog and to determine when the dialog has reached closure. Alexander [4] described the incorporation of Toulmin's approach into a software product (Teleologic DOORS) that represents features of arguments in a visual hierarchy to aid the analysis of positions taken by proponents and opponents of particular design requirements. The biggest challenge with these systems is that the sizes of their argumentation networks are often too large to comprehend and therefore it is very difficult to use them to help make design decisions.

Priority has been used to resolve conflicts for a long time [5] [6] in practice. However, incorporation of priority into an argumentation network remains challenging. In [6], Belnap first pointed out that self-conflicting arguments should not result in defeating other arguments. In later research such as [7] and [8], self-conflicting was not considered as a positive factor.

3. ARGUMENTATION BASED CONFLICT RESOLUTION IN THE COLLABARATIVE ENGINEERING DESIGN ENVIRONMENT

We have developed an intelligent collaborative engineering design system based on argumentation [1]. This design environment is based on the client-server architecture. On the client side, the system provides user interfaces for solid modeling, annotation, whiteboards for design alternatives, argumentation based conflict resolution, and chat rooms for real-time information exchange. On the server side, it manages client communication, concurrent access to design objects, and argumentation network. In the intelligent argumentation

subsystem for conflict resolution, the dialog for a design issue is captured as a weighted directed graph called a dialog graph [8], as shown in figure 1. The nodes denoted by circles are *Positions* i.e. the alternatives and the nodes denoted by rectangles are *Arguments*. Arcs represent a relationship (attack or support) from the originating argument node to the terminating argument or position node. The weight assigned to an argument is the Argument strength. It is the measure of an argument's degree of attack or support of either a position or another argument in the design dialog graph [8]. The weight value is a real number between -1 and 1. A positive number denotes support and a negative number denotes attack while zero denotes indecision. The strength of the argument is viewed as a fuzzy set and linguistic labels are used to represent the strength. We use linguistic labels Strong Support, Median Support, Indecisive, Medium Attack and Strong Attack to denote the strength of an argument or a position. A fuzzy inference engine is developed for argument reduction. The fuzzy inference engine has two inputs and one output. The inputs are the strengths of the argument to be reduced and the argument right above it. The output of the fuzzy inference engine is the reduced strength of the argument. We reduced the complexity of the network level by level using a fuzzy inference engine to the point where every argument under a position connects to it directly. Then we compute the favorability factor of each position by summing up every current weight of these arguments. Figure 2 shows that we acquire the favorability factors of two positions from solar car design example on our system platform. The position with the maximum favorability factor is the best design option.

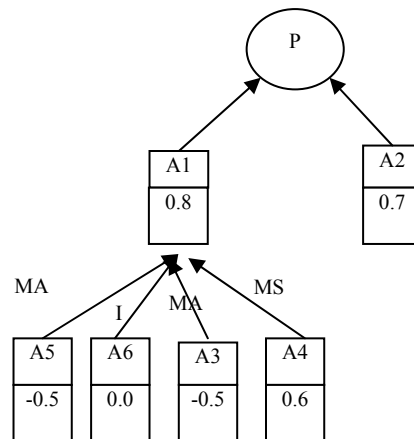


Figure 1. Argumentation Network

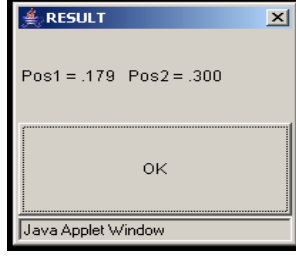


Figure 2. Favorability Factor

4. INCORPORATION OF PRIORITY OF PARTICIPANT INTO INTELLIGENT ARGUMENTATION

Each participant is assigned a priority. The priority value ranges from 0 to 1. The higher priority a participant has, the more powerful his/her argument is. A priority represents a participant's authority in a collaborative work. In our previous research, arguments move up in the argumentation network in the process of argumentation reduction. It is reasonable to assume the priority value of each participant is not changed no matter where this participant's argument is moved to in the network. We present two methods to incorporate priority into an argumentation network as discussed below.

4.1. Weighted Summation

Weighted summation is a simple and easy-to-understand way to assess the impact of priority on the final favorability factor. In our previous research, we summed up all the final strengths of arguments to get the favorability factor. Now the favorability can be computed as a weighted sum of strengths of arguments with priority as follows:

$$\text{Favorability} = \sum_{i=1}^m p_i \times w_i \quad (1)$$

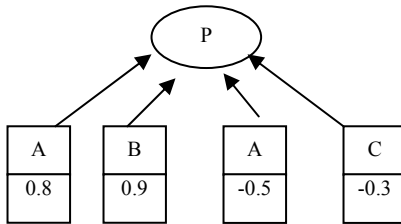


Figure 3. The Highest Level Where Every Argument Directly Connects to the Position

where w_i is strength of argument i and p_i is priority of the participant who raises argument i . As an example, a reduced final argumentation network [1] is shown in

figure 3. Assume that the priority of participant A is 1, the priority of B is 0.7, and the priority of C is 0.5. The favorability of position P calculated using equation 1 is 0.78.

4.2. Reassessment of Argument's Strength Based on Participant's Priority

Another technique to incorporate priority into an argumentation network of the collaborative engineering design system is to re-assess the strength of an argument based on the priority of the participant who raises the argument. It is based on the following priority re-assessment rules:

- General Priority Re-assessment Heuristic Rule 1: If the owner of argument A has a higher priority, the strength of this argument should be higher than it is.
- General Priority Re-assessment Argumentation Heuristic Rule 2: If the owner of an argument has a lower priority, the strength of this argument should be lower than it is.

As the linguistic labels used for the degrees of supporting and attacking are Strong Support (SS), Medium Support (MS), Indecisive (I), Medium Attack (MA) and Strong Attack (SA), and the linguistic labels for priority are high (H), medium (M) and low (L), the above two General Argumentation Heuristic Rules can be extended to fifteen Argumentation Heuristic Rules shown in figure 4.

	H	M	L
SS	SS	SS	MS
MS	SS	MS	I
I	I	I	I
MA	SA	MA	I
SA	SA	SA	MA

SS: Strong Support
 MS: Medium Support
 I: Indecisive
 MA: Medium Attack
 SA: Strong Attack
 H: high priority
 M: medium priority
 L: low priority

Figure 4. Argumentation Heuristic Rules

Using this fuzzy inference engine, we can incorporate priority and strength to revise the strength of an argument. Fuzzy membership functions are used to quantitatively characterize linguistic labels, such as low priority. In our previous research work, the fuzzy membership function chosen for the weight of strength is the piecewise linear trapezoidal function. The five fuzzy sets are strong attack, medium attack, indecisive, strong support, and medium support.

The fuzzy membership function chose for representing priority is also the piecewise linear trapezoidal function. The three fuzzy sets are Low, Medium and High, and the membership functions are shown in figure 5A. Figure 5B shows the five membership functions for the above five linguistic terms.

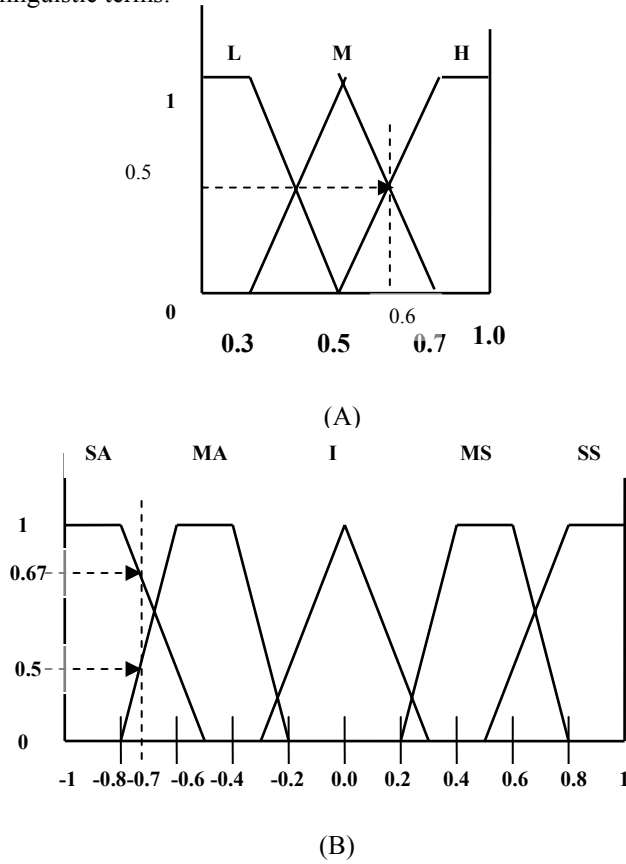


Figure 5. (A) Three Membership Functions for Priorities; (B) Five Membership Functions for Weights

Fuzzy inference rules combine two input fuzzy sets and associate with them an output set. The input sets are combined by means of operators that are analogous to the usual logical conjunctives “and”, “or”, etc. The fuzzy argumentation rules are stored and represented by a fuzzy association memory (FAM) matrix as shown in figure 6.

There are two inputs X and Y. The priority input variable (Y) has three input sets associated with it, which are labeled as “H”, “M”, “L”. The argument strength input variable (X) has five fuzzy sets associated with it, which have been labeled as “SA”, “MA”, “I”, “MS”, and “SS”. The output variable, Z, also has five output sets which are same as the argument strength input sets. Each FAM matrix entry is an output fuzzy set associated with a fuzzy rule. For example, the shaded part in figure 6 represents the rule: “If X is Strong Support (SS) and Y is L (low priority), then Z is Medium Support (MS).”

	H	M	L
SS	SS	SS	MS
MS	SS	MS	I
I	I	I	I
MA	SA	MA	I
SA	SA	SA	MA

Figure 6. FAM Matrix

The membership functions for the fuzzy sets SS, MS, I, MA and SA are denoted by F_{SS} , F_{MS} , F_I , F_{MA} and F_{SA} respectively. A particular value x of the input variable X then has membership degrees $F_{SS}(x)$, $F_{MS}(x)$, $F_I(x)$, $F_{MA}(x)$ and $F_{SA}(x)$. For example, with the trapezoidal membership functions shown in figure 5B and a value $x = -0.7$, we would have:

$$\begin{aligned}
 F_{SS}(-0.7) &= 0.0 \\
 F_{MS}(-0.7) &= 0.0 \\
 F_I(-0.7) &= 0.0 \\
 F_{MA}(-0.7) &= 0.5 \\
 F_{SA}(-0.7) &= 0.67
 \end{aligned}$$

Similarly, a particular value for y of the input variable Y would have membership degree values $P_H(y)$, $P_M(y)$, $P_L(y)$. The value $y = 0.6$ as shown in figure 7 would result in

$$\begin{aligned}
 P_H(0.6) &= 0.5 \\
 P_M(0.6) &= 0.5 \\
 P_L(0.6) &= 0.0
 \end{aligned}$$

Consider $x = -0.7$ and $y = 0.6$ as values of the input variables X and Y . A strength value is assigned to each entry in the FAM matrix by taking the minimum of the

membership function values associated with that entry. Now consider the FAM matrix entry corresponding to X, a member of the fuzzy set MA, and Y, a member of the fuzzy set M. Figure 7 illustrates the membership value for the priority input. The strength w_1 associated with the entry would be computed as:

$$\begin{aligned} w_1 &= \min [F_{MA}(-0.7), P_M(0.6)] \\ &= \min [0.5, 0.5] \\ &= 0.5 \end{aligned}$$

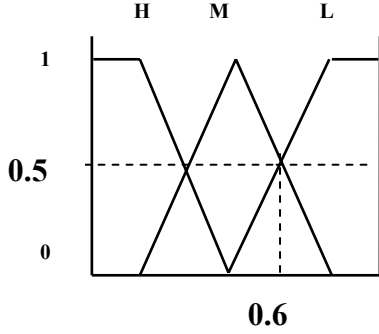


Figure 7. Membership Value for Priority Input

Only those FAM matrix entries which have nonzero membership-function values for both X and Y will have nonzero strengths associated with them. The shaded entries in the figure 8 show the four activated rules for the values in the example. In addition to w_1 , there are three more non-zero weights.

$$\begin{aligned} w_2 &= \min [F_{MA}(-0.7), P_H(0.6)] \\ &= \min [0.5, 0.5] \\ &= 0.5 \end{aligned}$$

$$\begin{aligned} w_3 &= \min [F_{SA}(-0.7), P_M(0.6)] \\ &= \min [0.67, 0.5] \\ &= 0.5 \end{aligned}$$

$$\begin{aligned} w_4 &= \min [F_{SA}(-0.7), P_H(0.6)] \\ &= \min [0.67, 0.5] \\ &= 0.5 \end{aligned}$$

The output variable Z also has five fuzzy sets associated with it, i.e. SS, MS, I, MA and SA. Specific values are assigned to these fuzzy sets, i.e. SS = 1, MS = 0.5, I = 0, MA = -0.5 and SA = -1. The system output is computed as follows:

$$\begin{aligned} \text{Output} &= \frac{w_1 \times MA + w_2 \times SA + w_3 \times SA + w_4 \times SA}{w_1 + w_2 + w_3 + w_4} \\ &= -0.875 \end{aligned}$$

	H	M	L
SS	SS	SS	MS
MS	SS	MS	I
I	I	I	I
MA	SA	MA	I
SA	SA	SA	MA

Figure 8. The Fuzzy Association Memory

5. DETECTION OF SELF-CONFLICTING ARGUMENTS

The robustness of an argumentation network is fundamental to making a convincing decision over multiple positions. However, the self-conflicting problem may hamper the robustness of the whole network and cause negative consequences.

The existence of self-conflicting arguments means that several of arguments of a participant are contradictory among themselves. In a complicated collaborative design environment with a number of participants, the self-conflicting problem could take place frequently. What is even worse is that they are not easy to detect in many cases. A participant often gets involved in self-conflicting arguments. The existence of self-conflicting is such a big issue in a collaborative design environment that it is often difficult to obtain a convincing decision.

If a participant has some self-conflicting arguments in the network, then no matter how powerful this participant is, his arguments will provide some unaccountable and confusing information instead of positively contributing to the argumentation process.

Here is a simple example. In a network as shown in figure 9, the owner of argument A1 is O_1 , A2 attacks A1, A4 supports A2, A5 support A4, therefore we can easily conclude that A5 attacks A1, now if the owner of argument A5 is also O_1 , then A1 and A5 are a pair of self-conflicting arguments of owner O_1 .

In this simple example, it is easy to detect where the self-conflicting is. However, in a huge network with many self-conflicting arguments, they cannot be easily detected by just looking through the network. We divide the self-conflicting problem into two categories. The first one is one-to-one self-conflicting, which represents two obviously contradictory arguments of one owner. The second is multiple self-conflicting. Multiple self-

conflicting represents a more complicated relationship in that a few arguments of one owner are conflicting with each other. This kind of self-conflicting is extremely difficult to find out. It is necessary that we develop an effective algorithm to detect and remove self-conflicting arguments no matter what kind of self-conflicting it is. Using an algorithm shown in figure 10, by traversing all offspring argument nodes of argument node A, we can detect many self-conflicting arguments.

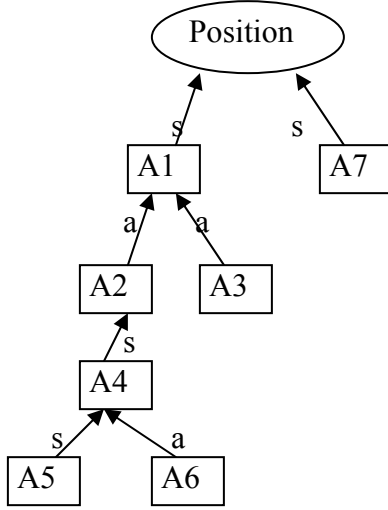


Figure 9. A Simple Example to Illustrate Self-conflicting

6. QUANTITIVE ANALYSIS OF ARGUMENTATION NETWORK

Normally when people start participating in a large complicated argumentation network, they do not even know where to start. Therefore, it is necessary to provide a little statistical information of the network to help comprehend the complicated network. In this paper, we proposed to provide two types of statistical information about an argumentation network: owner-oriented and argument-oriented.

Owner-oriented information indicates participation of each participant and its relation with other owners. It shows how many arguments one participant owns and which group this participant belongs to. Figure 11(A) shows in an example how our system presents owner-oriented information. Argument-oriented information shows which arguments are popular. Normally, a popular argument has many more follow-up arguments supporting or attacking it. Figure 11(B) shows in the same example how our system presents argument-oriented information.

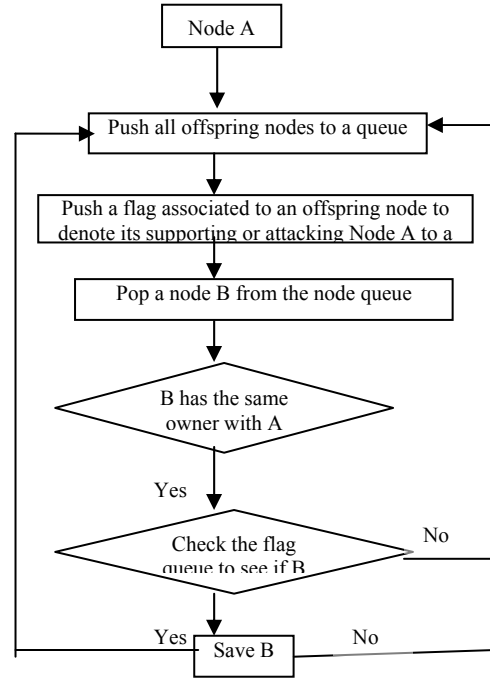


Figure 10. Algorithm to Resolve the Self-conflicting

9 has 1 follow-up arguments

7 has 3 follow-up arguments

6 has 5 follow-up arguments

3 has 7 follow-up arguments

2 has 1 follow-up arguments

1 has 10 follow-up arguments

(A)

GroupD (1) has the following participants:
hut(1);

GroupC (4) has the following participants:
man(3); linsay(1);

GroupB (6) has the following participants:
redbull(1); man(3); robin(1); peter(1);

GroupA (2) has the following participants:
superman(1); spiderman(1);

GroupH (1) has the following participants:
tim(1);

groupH (1) has the following participants:
abc(1);

(B)

Figure 11. (A) Argument-oriented Information; (B) Participant Oriented Information

7. IMPLEMENTATION OF THE INTERNET BASED INTELLIGENT ARGUMENTATION SYSTEM

The proposed priority assessment method, self-conflicting argument detection technique, and quantitative analysis tools have been implemented in our collaborative engineering design system based on arguments.

The conflict resolution window is shown in figure 12. The newly added functionalities have been added to the menu and the proposed features have been implemented in the intelligent argumentation subsystem.

The added functionalities have greatly strengthened the capability of intelligent argumentation for our collaborative design system.

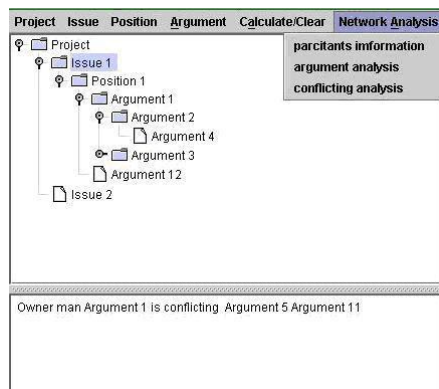


Figure 12. Improved Conflict Resolution Window

8. CONCLUSION

The main contribution of this paper includes several significant improvements to the argumentation model and subsystem of our previous intelligent collaborative engineering design system. Firstly, we incorporate the priority of participant to the argumentation model using two different techniques: weighted summation and priority-based argument strength re-assessment. Secondly, we develop an effective approach for detecting of self-conflicting arguments. In order to help participants understand argumentation in a large complex argumentation network, we provide several analytical tools. In the future, we plan to conduct several field studies to demonstrate the effectiveness of these methods and tools.

ACKNOWLEDGEMENT

This research is supported by the Intelligent Systems Center and University Transportation Center in the University of Missouri-Rolla.

REFERENCE

- [1] Liu, X. F., Raorane, S., Zheng, M., and Leu, M. C., "An Internet Based Intelligent Argumentation System for Collaborative Engineering Design," *Proceedings of the 2006 IEEE International Symposium on Collaborative Technologies and Systems, (CTS 2006)*, Las Vegas, Nevada, May 14-17, 2006, pp. 318-325.
- [2] Toulmin, S. E., *The Uses of Argument*, University Press, Cambridge, UK, 1958.
- [3] Sillence, J., "Intelligent Argumentation Systems: Requirements, Models, Research Agenda, and Applications," *Encyclopedia of Library and Information Science*, New York, USA, 1997, pp. 176 – 217.
- [4] Alexander, I., "Modeling Argumentation, Toulmin-Style," Retrieved April 15, 2005, <http://easyweb.easynet.co.uk/~iany/consultancy/papers.htm>.
- [5] Saaty, T. L., *Fundamentals of Decision Making and Priority Theory with the Analytical Hierarchy Process*, RWS publications, Pittsburgh, PA, 2000.
- [6] Belnap, N. D., Jr. "A Useful Four-Valued Logic," In *Modern Uses of Multiple-Valued Logic*, M. Dunn and G. Epstein, editors, 1977, pp. 8–41.
- [7] Bochman, A., "Propositional Argumentation and Causal Reasoning," *International Joint Conference on Artificial Intelligence*, 2005, pp. 388-393.
- [8] Bochman, A., "Collective Argumentation," *Proceedings of the Workshop on Non-Monotonic Reasoning*, 2002.