



MISSOURI
S&T

CENTER FOR TRANSPORTATION INFRASTRUCTURE AND SAFETY

A Web-based Intelligent Collaborative Bridge Design System

by

Man Zheng



**NUTC
R203**

**A National University Transportation Center
at Missouri University of Science & Technology**

Disclaimer

The contents of this report reflect the views of the author(s), who are responsible for the facts and the accuracy of information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program and the Center for Infrastructure Engineering Studies UTC program at the University of Missouri - Rolla, in the interest of information exchange. The U.S. Government and Center for Infrastructure Engineering Studies assumes no liability for the contents or use thereof.

Technical Report Documentation Page

1. Report No. NUTC R203	2. Government Accession No.	3. Recipient's Catalog No.		
4. Title and Subtitle A Web-based Intelligent Collaborative Bridge Design System	5. Report Date September 2007		6. Performing Organization Code	
	7. Author/s Man Zheng		8. Performing Organization Report No. 00016751	
9. Performing Organization Name and Address Center for Transportation Infrastructure and Safety/NUTC program Missouri University of Science & Technology 223 Engineering Research Lab Rolla, MO 65409		10. Work Unit No. (TRAIS)		
		11. Contract or Grant No. DTRT06-G-0014		
12. Sponsoring Organization Name and Address U.S. Department of Transportation Research and Special Programs Administration 400 7 th Street, SW Washington, DC 20590-0001		13. Type of Report and Period Covered Final		
		14. Sponsoring Agency Code		
15. Supplementary Notes				
16. Abstract This project is aimed at graduate research training of students interested in pursuing careers in transportation areas. Financial support will be provided to recruit eight new graduate students interested in pursuing their doctoral degrees in transportation areas each year. These students can pursue their doctoral studies in any department at MST. In departments where a master's degree is the highest degree awarded, students pursuing their master's degree with thesis option will be considered. Areas as stated in the goals, interests and objectives of State Departments of Transportation and Missouri Department of Transportation in particular will be considered for support in this project.				
17. Key Words Design, economic development, education, finance, Future Transportation Professionals, management, materials, operations, safety, security, policy		18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.		
19. Security Classification (of this report) unclassified		20. Security Classification (of this page) unclassified	21. No. Of Pages 23	22. Price

A Web-based Intelligent Collaborative Bridge Design System

INTRODUCITON

This system provides a web-based intelligent collaborative design environment to facilitate the bridge designers to cooperate with each other while working at different locations at the same time. Each designer communicates with the rest though the system interface shown as figure 1.

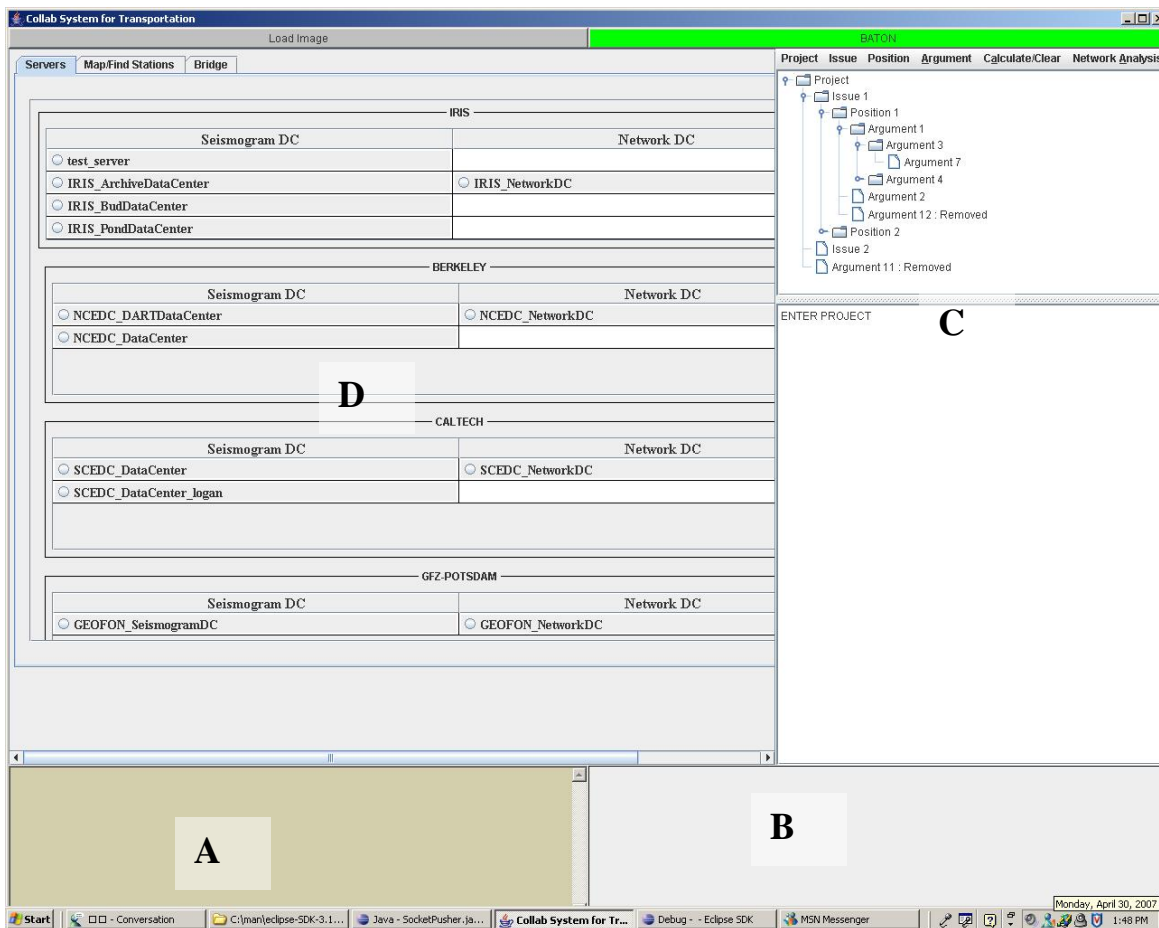


Figure 1 the system interface

The system interface consists of four parts, the part A is the chatting window, designers can exchange their opinions through chatting with each other; part B is the drawing window, it allows designers to simply draw drafts and the rest of designers are able to see the drafts spontaneously. In Part C designers share some drafts of bridge designs as shown in figure 2A and obtain seismograms data from seismograms servers and minimum and maximum value of a specific time unit for a certain time period. Part D manages the arguments from different

designers and applies argumentation based conflict resolution theory that we developed to choose the best option for a bridge design issue. The server is in charge of multiple system interfaces. On the server side, it manages client communication, concurrent access to design objects, and argumentation network

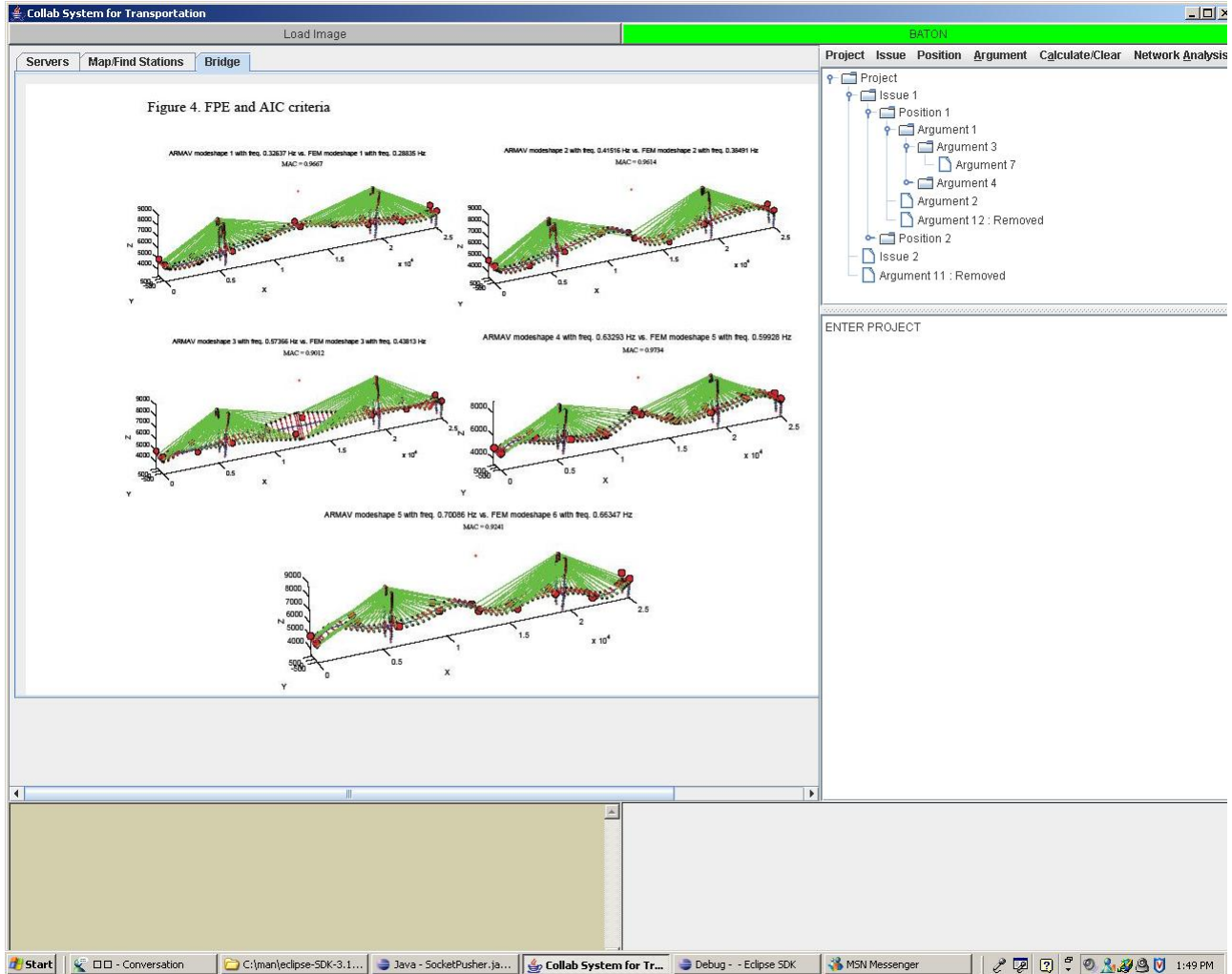


Figure 2A design drafts for a design issue in part C

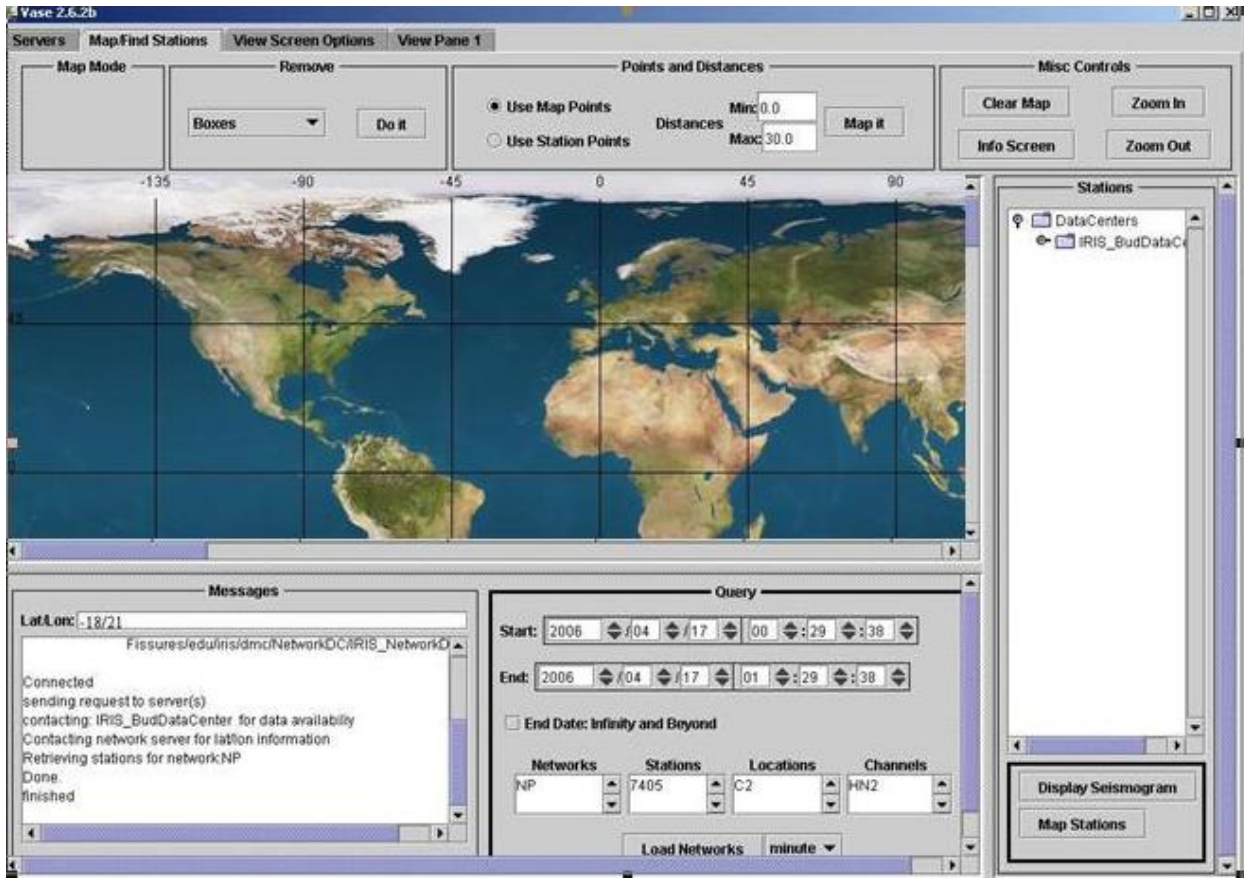


Figure 2B input window for seismogram data in part C

GENERAL USAGE ON BRIDGE DESIGN PART OF THE SYSTEM

Seismograms Servers

The opening screen as shown in part D Figure 1 presents you with a listing of Seismograms and Network servers grouped by institution. Scrollbars will appear as necessary to allow selecting servers which are not displayed. The column with a heading of Seismogram DC lists the waveform servers. The Network DC column lists servers for station information such as latitude and longitude.

Which Seismogram Server to Use?

As of October, 2005, the features that are available in various seismogram servers are summarized in Table 1. To have a complete picture of the seismic data available in the U.S., all the servers servicing the U.S. seismic stations included in the IRIS website are discussed. The IRIS BudDataCenter includes the seismograms from the Bill Emerson Memorial Cable-stayed Bridge. In the following discussions, an emphasis will be placed on this server including the seismic data required for this project.

Table 1 Main Features in Various Servers

<p>Iris Server BudDataCenter (stored the recorded data from the Bill Emerson Memorial Cable-stayed Bridge)</p>	<p>Use the BudDataCenter for those stations which are streaming data to the DMC. There is approximately a 6 week moving window of data available in the bud. This varies slightly from station to station. Use this server if you want to have seismograms streaming onto your computer</p>
<p>Iris Server PondDataCenter</p>	<p>This server accesses waveform data going back over ten years. This collection is obtained by gathering waveforms in a two hour window from earthquakes of magnitude 5 and above. Recent events are collected from the Bud, and as more station report, their data is included. This older data sets have many more stations reporting than recent ones.</p>
<p>Iris Server ArchiveDataCenter</p>	<p>This server is an offline system which is not supported by Vase.</p>
<p>Berkeley Servers NCEDC_DataCenter</p>	<p>This server is similar to the Bud server. However the data starts at approximately 2001/10 and continues up to two days before the current day. Stations located in Northern California and Southern Oregon.</p>
<p>CalTech Servers SCEDC_DataCenter</p>	<p>Similar to the BudDataCenter. This server will stream seismograms from stations located in Southern California.</p>
<p>SC Servers SCEPPSeismogramDC</p>	<p>Similar to the BudDataCenter. This server will stream seismograms from stations located in South Carolina.</p>

Map/Find Stations

Clicking the "Map/Find Stations" tab in figure 2B will prompt you with the Map/Find Stations screen. At the bottom right side of the screen in figure 2B is a query section.

Query

The query section as shown in Figure 3 allows one to search for load networks/stations. First, select and enter a start/end time, or specific networks, stations, locations and channels. Wildcard may be used as a shortcut. To use an unspecified end time, in other words to download continuously, click on the "End Date: Infinity and Beyond". Then, press the "Load Networks" button to query the DHI for stations and channels that meet your search criteria. If a specific search area has been specified on the map, the search results will only include those stations that lie within the latitude/longitude point/distance or freehand shape drawn on the map. On the right side of the "Load Networks" button, a combo box shown "minute" in Figure 3 is used to select the observation time duration (e.g., minute or hour) for the selected peak responses. If the time duration is specified as "hour", the software will find the hourly minimum and the maximum responses within the time window of interest. Otherwise the computer software will by default extract the minimum and the maximum response values every minute.

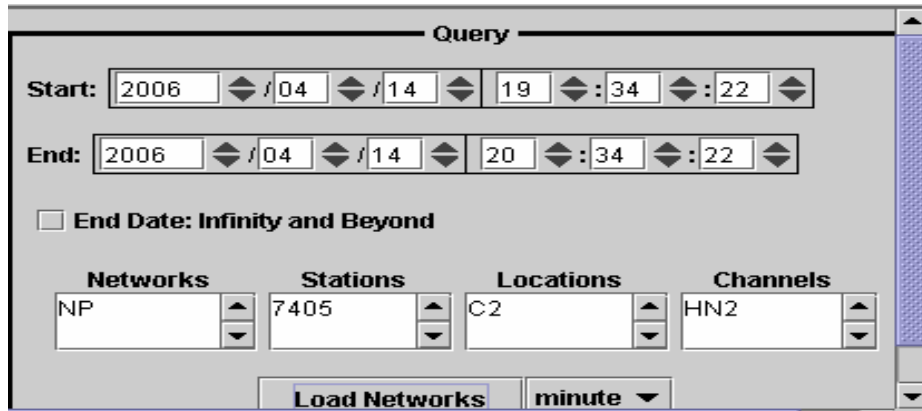


Figure 3 Query Section

As an example, the minimum and the maximum acceleration values are to be determined every minute in a one hour window from 19:34:22 to 20:34:22 on April 14, 2006. In this case, the Start Day and Time were specified in Figure 2 as April 14 at 19 hours, 34 minutes, and 22 seconds. Correspondingly, the End Day and Time were April 14 at 20 hours, 34 minutes, and 22 seconds. The network (NP), Station (7405), Location (C2), and Channel (HN2) are selected in this example. By clicking on any triangle in Figure 3, one can change the data and time by one unit. In this query system, one can select individual channels or a station for all channels at the station. Similarly one can select a network to display all stations and channels within that network or click on a data center for all networks, stations and channels. The query process is recorded in the message panel as illustrated in Figure 4. Note that the latitude and longitude information (-9/ -18) does not necessarily correspond to the location of the station selected in Figure 3. They correspond to the last location of the cursor on the map.

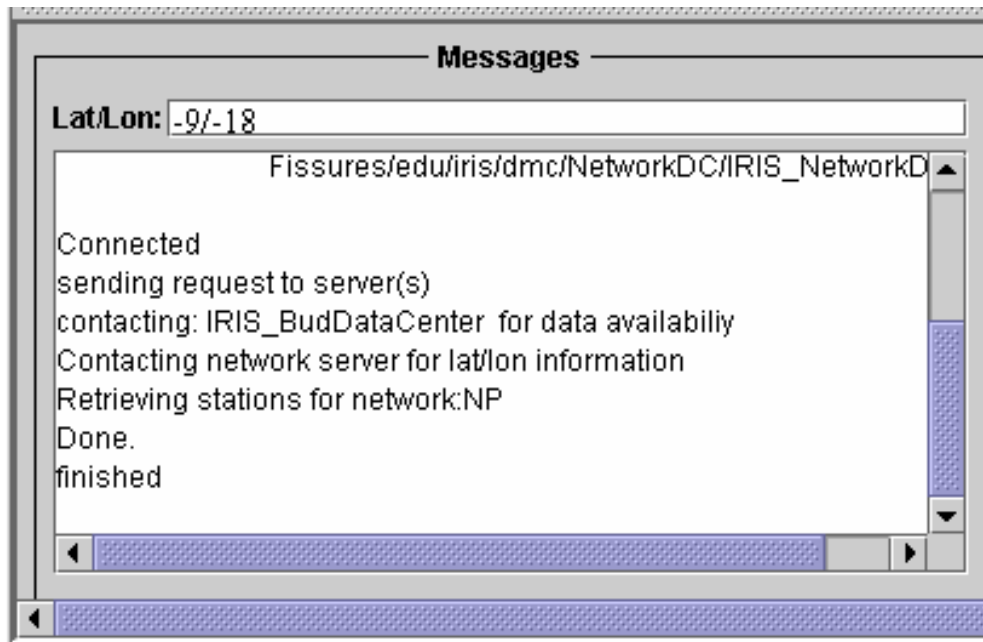


Figure 4 Query Results

Displaying Seismograms

To display the seismograms, click on the "Display Seismograms" button in Figure 2B. A new tabbed pane appears as shown in Figure 5 and the downloading process is initiated.

Each time the "Display Seismogram" button is clicked on, a new tabbed pane will be created with the name "View Pane n" where "n" is an increasing integer. Figure 5 shows View Pane 1.

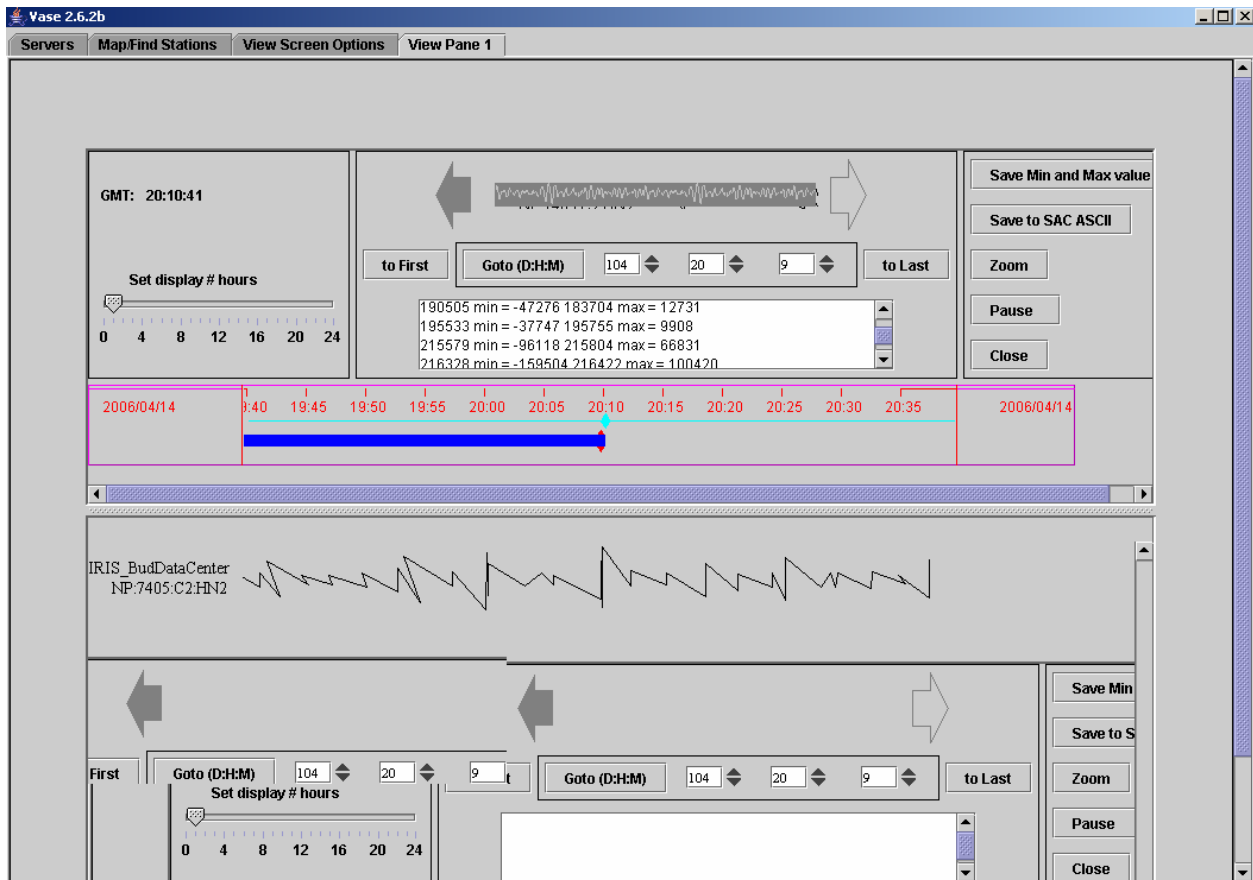


Figure 5 View Pane 1

The top center section of View Pane 1 is zoomed in as presented in Figure 6. This section gives some information and control areas. For example, if one would like to view the minimum and the maximum acceleration values on April 14 at 20 hours and 9 minutes (GMT), the number of day (104) and hour (20), and minute (9) are selected in Figure 5. Note that April 14 is the 104th day of 2006. The minimum (min) and maximum (max) values every minute and their corresponding time (first and third columns in Figure 6) are showed in the print-out pane. The acceleration is represented by the count of samples. How many samples correspond to 1g is to be determined in the near future.

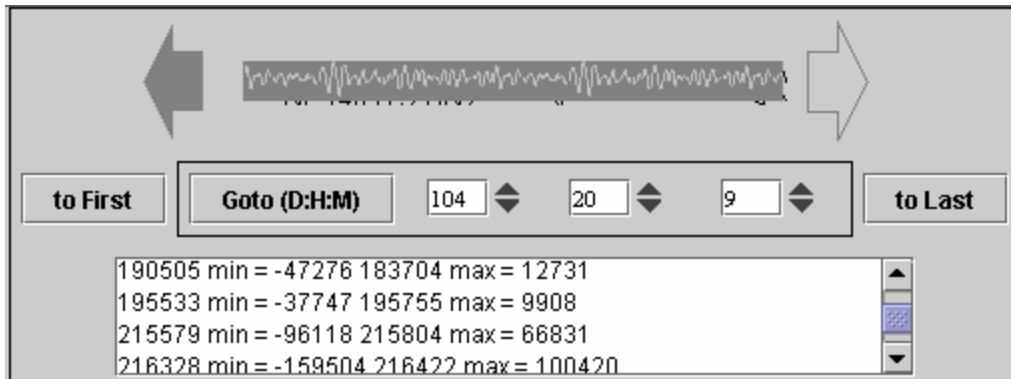


Figure 6 Heading Information in View Pane 1

In Figure 6, the left- and right-arrows allow one to view the previous or next pages of a long stream of the downloaded data within the time window. Alternatively, one can also jump to the beginning or end of the seismogram by using “to First” or “to Last” buttons or to any specific time by using the “Goto (D:H:M)” button.

The scrolling bar with a seismogram animation in the middle of Figure 5 is an indicator that the program is receiving data. As data arrives the message to that effect is displayed in the listing.

To the bottom and right of Figure 5 are several option buttons. A complete set of the downloaded data may be saved to a disk in the SAC ASCII format. Figure 7 is a screen shot illustrating the change of the min and max value every minute (observation time duration) with the time window.

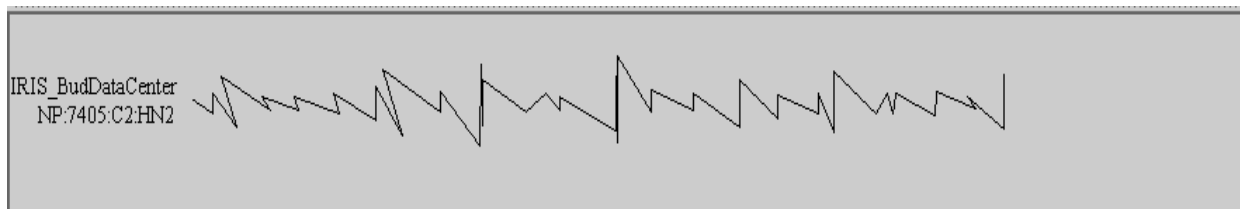


Figure 7 Waveform of min and max Values

Instead of the complete downloaded data, one may select and save individual traces into a disk or zoom in the traces. Once either one of the "Save Min and Max value" buttons is clicked on, a popup screen appears as presented in Figure 8, from which one can choose the right directories on the local disk to store the data.

One may type in the directories or scroll the bar and click on the one selected in the listing. In this way, data can be saved in different drives. Click on the ".." entry to go up one level. When an appropriate directory is selected, press the "Accept" button to initiate the saving of the data in the hard disk.

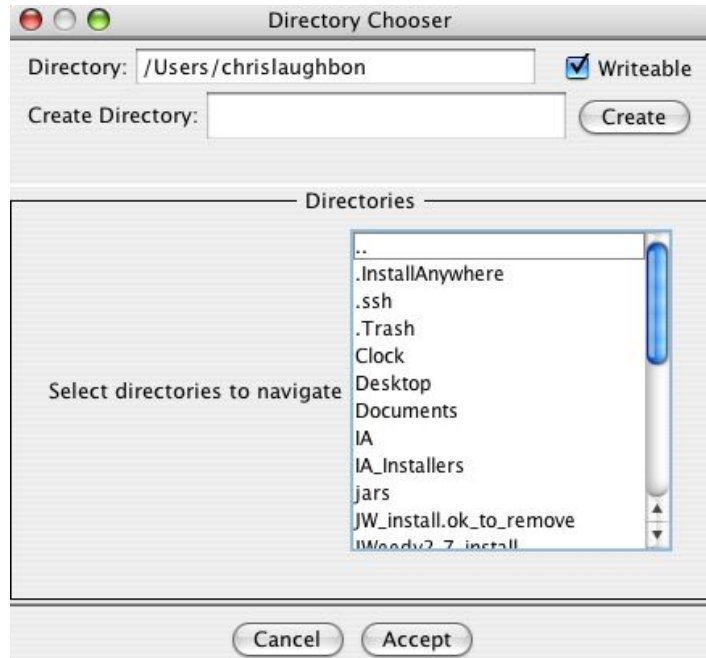


Figure 8 Directory Chooser

Saved Data

	A	B	C	D	E	F	G	H	I
1	1815 min =	-44268	905 max =	7976					
2	21242 min =	-46383	19710 max =	19126					
3	35515 min =	-76818	35503 max =	35144					
4	45542 min =	-60957	45547 max =	26418					
5	52107 min =	-48012	60162 max =	12491					
6	66561 min =	-76747	69306 max =	32889					
7	83974 min =	-56091	72205 max =	16004					
8	91124 min =	-103721	91025 max =	64426					
9	104823 min =	-48543	105350 max =	22513					
10	119480 min =	-64466	119478 max =	30819					
11	131998 min =	-236851	131769 max =	155553					
12	132632 min =	-176992	132637 max =	162273					
13	144534 min =	-47599	145131 max =	16743					
14	160187 min =	-105237	159952 max =	66294					
15	175000 min =	-47668	176596 max =	10630					
16	189832 min =	-166233	189845 max =	108878					
17	192308 min =	-81621	193047 max =	30205					
18	214703 min =	-87061	214675 max =	50180					
19	227975 min =	-151958	227934 max =	109872					

Figure 9 saved data

Finally, the data can be saved in an Excel file. The data format of the preliminary results is shown in Figure 9. Column A represents the time corresponding to the min value listed in Column B. Column C denotes the time instant for the max value in Column D.

INSTRUCTION TO ARGUMENTATION MANAGEMENT PART

A Web-based Intelligent Collaborative Engineering Design Environment

This collaborative bridge design system is based on a prototype of a web-based intelligent collaborative system for engineering design is being developed. It is based on client-server architecture, as shown in figure 10. 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.

In the collaborative design process, when a conflict is detected, an argumentation based conflict resolution session will be initiated. A design issue concerning the conflict is raised first in the session. After multiple design alternatives are generated from designers, arguments can then be proposed to either support or oppose the design alternatives or arguments themselves. Our system will help to identify the alternative most favored by all designers considering all arguments to resolve the conflicts.

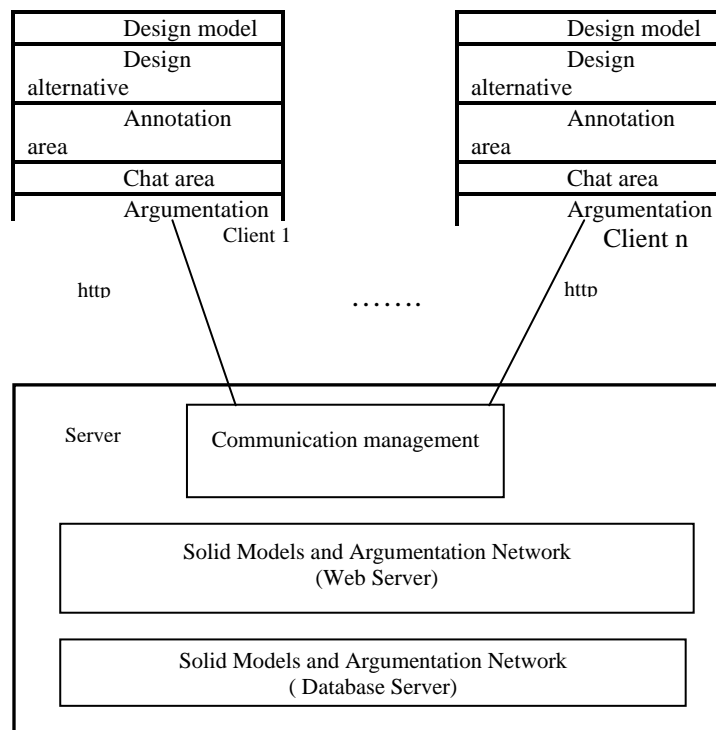


Figure 10. Architecture for a Web-Based Intelligent Collaborative Engineering Design Environment

The argumentation framework of this conflict resolution system is an extension of the informal IBIS model of argumentation. We have developed a computational argumentation method for collaborative bridge design based on our preliminary work on software design rationale capturing. It will help to achieve consensus among stakeholders and identify the most favorable design alternative through argumentation by computing favorability of individual design alternatives from all arguments in the argumentation network in an uncertain environment based on fuzzy logic.

The components of the proposed design argumentation model for collaborative engineering design includes stakeholders, requirements, conflicts, design issues, parts, alternatives, arguments, and decisions. We view collaborative design as the process of negotiating the resolution of design issues through dialog between the stakeholders. The dialog for a given design issue is represented by the alternatives that are related to a design issue, and the arguments for or against each alternative. Resolution of a design issue is represented by a decision that selects an alternative which is the most favored.

The conflict resolution window for the conflict resolution module is shown in part C figure 1. Control Panel: The Control Panel has five menus viz. Project, Issue, Position, Argument and Calculation/Clear. Each menu had submenus which perform unique actions on the respective argumentation elements viz. Project, Issue, Position and Argument.

The basic argumentation elements are project, issues, positions and arguments. Project forms the root node, followed by issues i.e. the conflicting design issues that are generated under a particular project. Under issues are positions i.e. the design alternatives which can solve the issue. Under positions are arguments and every argument can have any number of arguments under it. The tree structure is so designed that a designer at a time can work on any sub-tree of the complete argumentation tree. This helps the designer to concentrate on a specific part of the argumentation. The argumentation tree is not too large and as the fuzzy inference engine is used to solve the conflict, design decisions can be made without any difficulty.

The design dialog for a design issue is captured as a weighted directed graph called a dialog graph. The nodes denoted by a circle 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. Position node contains the name of the stakeholder posting the position and the text of the position. Each Argument node contains the name of the stakeholder posting the argument, the text of the argument and a weight value. The weight attached to an argument is known as the Argument strength. It is the measure of an arguments degree of attack or support of either a position or another argument in the position dialog graph. 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. It is easy to use linguistic labels to denote the strength of an argument over another argument or a position instead of a real number value. By doing so fuzzy inference can be used to evaluate a position. A position node contains a label associated with it which gives the measure of the position's strength relative to the strengths of the arguments under it. This measure is known as the favorability factor of the position.

Argument reduction through fuzzy inference

In figure 11, we can see some arguments attached to some other arguments, by a label to denote the degree of support or attack on the arc going between arguments, other than directly attached to the position. For example, A3 medium attack(MA) A1 and A5 strong support A3. Argument reduction means reducing the arguments which are not directly connected to the position in order to have them directly connected to the position i.e. Argument A3 which is posted as an argument that attacks argument A1, actually attacks the position P after reduction.

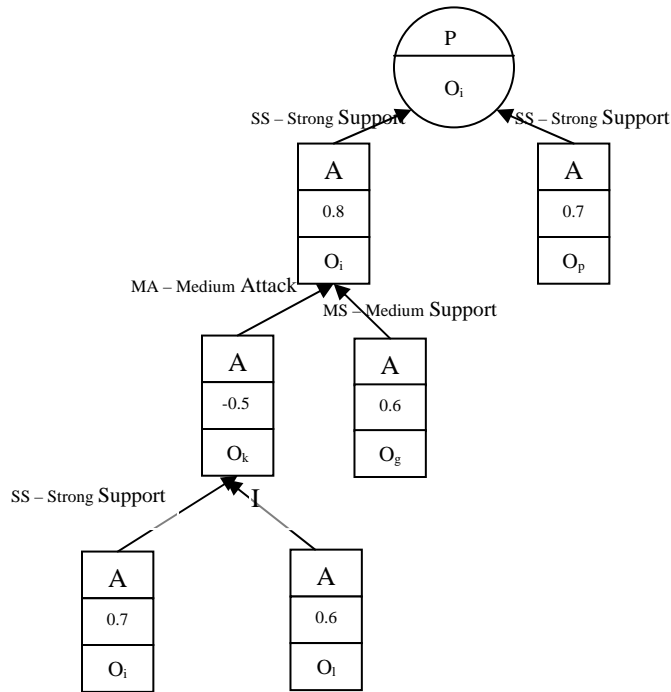


Figure 11 Position Dialog Graph

There are four possible General Argumentation Heuristic Rules that can be formulated as follows,

- General Argumentation Heuristic Rule 1: If argument B supports argument A and argument A supports position P, then argument B supports position P.
- General Argumentation Heuristic Rule 2: If argument B attacks argument A and argument A supports position P, then argument B attacks position P.
- General Argumentation Heuristic Rule 3: If argument B supports argument A and argument A attacks position P, then argument B attacks position P.
- General Argumentation Heuristic Rule 4: If argument B attacks argument A and argument A attacks position P, then argument B supports position P.

As the linguistic labels used are Strong Support (SS), Medium Support (MS), Indecisive (I), Medium Attack (MA) and Strong Attack (SA), the above four General Argumentation Heuristic Rules can be extended to obtain twenty-five Argumentation Heuristic Rules shown in figure 12.

Consider an instance where the strength of the level-1 argument is Strong Attack and that of the level 2 argument is Medium Support then the reduced strength of the level-2 argument will be Medium Attack as shown by the entry in column 3 and row 5.

A fuzzy inference engine has been built to infer the reduced strengths of the arguments as discussed later in this section. Using this fuzzy inference engine we can reduce a given Position dialog graph into one in which all the argument nodes are directly attached to the position node. Consider the example in figure 11, where we have arguments occurring at level 3.

First, the argument nodes at level 3 are reduced, i.e. their reduced strengths are computed using the fuzzy inference engine and are attached to the argument node at level 1, which is on the path from the argument node to the position node. Hence from level 3 the arguments come to level 2. It is shown in figure 13.

	SS	MS	I	MA	SA
SS	SS	MS	I	MA	SA
MS	MS	MS	I	MA	SA
I	I	I	I	I	I
MA	MA	MA	I	MS	MS
SA	SA	MA	I	MS	SS

SS: Strong Support
 MS: Medium Support
 I: Indecisive
 MA: Medium Attack
 SA: Strong Attack

Figure 12 Argumentation Heuristic Rules

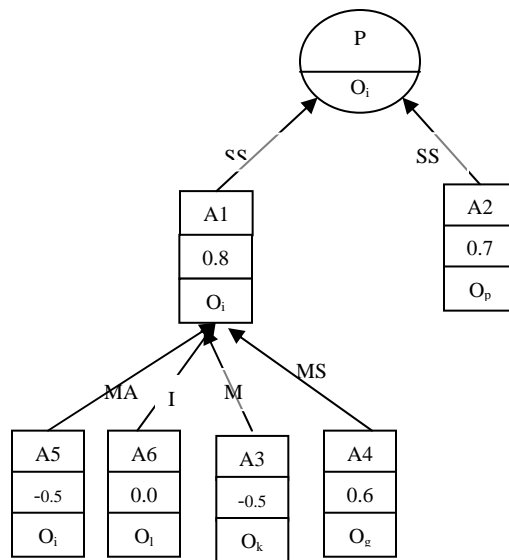


Figure 13 Position Dialog graph after one level reduction

Now there is one level of arguments which are not directly attached to the position and hence argument reduction has to be performed once again to have the reduced position dialog graph which will have all the arguments directly attached to it. The arguments at level 2 are reduced using the fuzzy inference engine and attached directly to the position node as shown in figure 13.

So far the procedure of argument reduction has been discussed. The fuzzy inference engine takes in two inputs and gives one output. The inputs are the weights or 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 that had to be reduced.

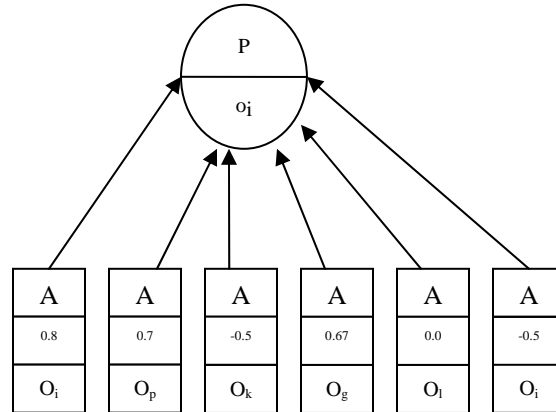


Figure 13. Position dialog graph after complete reduction

Charactering of linguistic variable through Fuzzy Membership Functions

Fuzzy membership functions are used to characterize quantitatively linguistic systems represented as fuzzy set, such as strong attack, The fuzzy membership function chosen for this system is the piecewise linear trapezoidal function. Several membership functions are defined by using a,b,c,d to denote the four vertexes of the trapezoids. Five membership functions have been defined for the five fuzzy sets. The five fuzzy sets are, Strong Attack (SA: a = -1, b = -1, c = -0.8, d = -0.5), Medium Attack (MA: a = -0.8, b = -0.6, c = -0.4, d = -0.2), Indecisive (I: a = -0.3, b = 0, c = 0, d = 0.3), Medium Support (MS: a = 0.2, b = 0.4, c = 0.6, d = 0.8) and Strong Support (SS: a = 0.5, b = 0.8, c = 1, d = 1). Figure 14 shows the five membership functions for the above five linguistic terms.

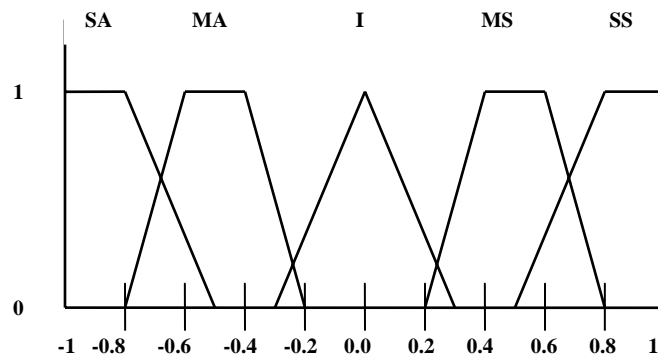


Figure 14. Five Membership functions

Fuzzy Inference Rules

Fuzzy inference rules combine two or more 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 rules, also termed later as the Argumentation rules are presented in Table 1 in the beginning of this section. The fuzzy or argumentation rules are stored and represented through the use of fuzzy association memory (FAM) matrix. Figure 15 shows the two-dimensional FAM matrix used. Here there are two inputs X and Y. Each input variable has five input sets associated with it, which have been labeled as “SS”, “MS”, “I”, “MA”, and “SA”. It is not necessary that both the inputs belong to the same input set. The output variable, Z also has five output sets which are same as the five input sets. Each FAM matrix entry is an output fuzzy set that is the output fuzzy set of the fuzzy rule. For example, the shaded part in the figure 5 represents the rule: “If X is Strong Attack (SA) and Y is Strong Support (SS), then Z is Strong Attack (SA).” SA is a fuzzy set that would typically represent a range of values near -1.

Fuzzy System and Defuzzification

The system associated with the FAM matrix is shown in figure 15. In this case we have two input variables, X and Y, with associated fuzzy sets SS, MS, I, MA and SA. Figure 16 shows how the membership functions may look for these sets.

$\begin{matrix} x \\ y \end{matrix}$	SS	MS	I	MA	SA
SS	SS	MS	I	MA	SA
MS	MS	MS	I	MA	SA
I	I	I	I	I	I
MA	MA	MA	I	MS	MS
SA	SA	MA	I	MS	SS

Figure 15. The Fuzzy Association Memory (FAM) matrix

The membership functions for the fuzzy sets SS, MS, I, MA and SA is denoted by FSS, FMS, FI, FMA and FSA respectively. A particular value x of the input variable X then has membership degrees FSS(x), FMS(x), FI(x), FMA(x) and FSA(x). For example, with the trapezoidal membership functions shown in figure 16 and a value $x = -0.7$, we would have:

$$\begin{aligned} FSS(-0.7) &= 0.0 \\ FMS(-0.7) &= 0.0 \\ FI(-0.7) &= 0.0 \\ FMA(-0.7) &= 0.5 \\ FSA(-0.7) &= 0.67 \end{aligned}$$

Similarly, a particular value for y of the input variable Y would have membership degree values FSS(y), FMS(y), FI(y), FMA(y) and FSA(y). The value y = 0.6 as shown in figure 16 would result in

$$\begin{aligned} \text{FSS}(0.6) &= 0.33 \\ \text{FMS}(0.6) &= 1.0 \\ \text{FI}(0.6) &= 0.0 \\ \text{FMA}(0.6) &= 0.0 \\ \text{FSA}(0.6) &= 0.0 \end{aligned}$$

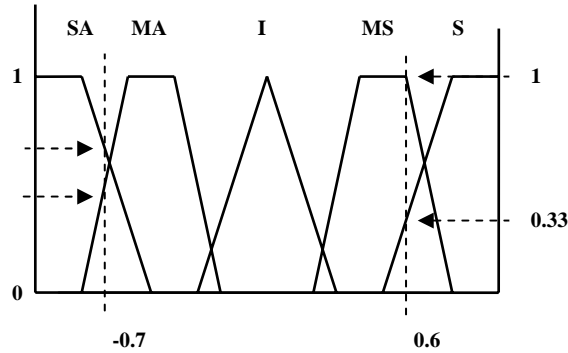


Figure 16. Membership Degrees

Consider $x = -0.7$ and $y = 0.6$ are presented to the system as values of the input variables X and Y. A weight 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 SS. The weight w_1 associated with the entry would be computed as:

$$\begin{aligned} w_1 &= \min [\text{FMA}(-0.7), \text{FSS}(0.6)] \\ &= \min [0.5, 0.33] \\ &= 0.33 \end{aligned}$$

Only those FAM matrix entries which have nonzero membership-function values for both X and Y will have nonzero weights associated with them. The shaded entries in the figure 17 show the four activated rules for the values in the example. In addition to w_1 , there are three more nonzero weights.

$$\begin{aligned} w_2 &= \min [\text{FMA}(-0.7), \text{FMS}(0.6)] \\ &= \min [0.5, 1.0] \\ &= 0.5 \end{aligned}$$

$$\begin{aligned} w_3 &= \min [\text{FSA}(-0.7), \text{FSS}(0.6)] \\ &= \min [0.67, 0.33] \\ &= 0.33 \end{aligned}$$

$$w_4 = \min [\text{FSA}(-0.7), \text{FMS}(0.6)]$$

$$= \min [0.67, 1.0]$$

$$= 0.67$$

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:

$$\text{Output} = \frac{(w1 \cdot MA + w2 \cdot MA + w3 \cdot SA + w4 \cdot MA)}{(w1 + w2 + w3 + w4)}$$

$$= \frac{(0.33 \times (-0.5) + 0.5 \times (-0.5) + 0.33 \times (-1) + 0.67(-0.5))}{(0.33+0.5+0.33+0.67)}$$

$$= -0.59$$

x y	SS	MS	I	MA	SA
SS	SS	MS	I	MA	SA
MS	MS	MS	I	MA	SA
I	I	I	I	I	I
MA	MA	MA	I	MS	MS
SA	SA	MA	I	MS	SS

Figure17. The Fuzzy Association Memory(FAM) matrixII

Incorporation of Priority of Designer into Intelligent Argumentation

Each designer is also assigned a priority. The priority value ranges from 0 to 1. The higher priority a designer has, the more powerful his/her argument is. A priority represents a designer'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 designer is not changed no matter where this designer's argument is moved to in the network. We present two methods to incorporate priority into an argumentation network as discussed below.

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:

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

where w_i is strength of argument i and p_i is priority of the designer who raises argument i . As an example, a reduced final argumentation network [1] is shown in figure 18. Assume that the priority of designer 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.

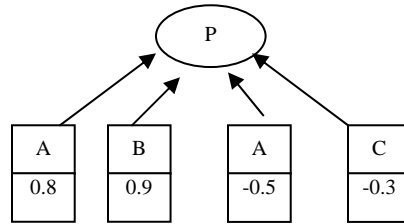


Figure 18. The Highest Level Where Every Argument Directly Connects to the Position

Reassessment of Argument's Strength Based on Designer'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 designer 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 19.

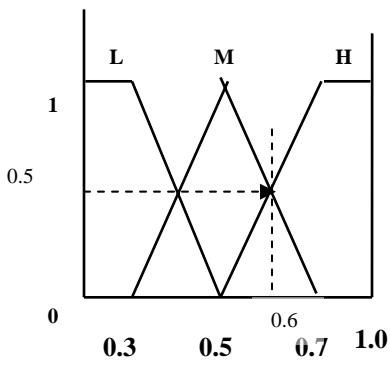
Using this fuzzy inference engine, we can incorporate priority and strength to revise the strength of an argument. Fuzzy membership functions are used to quantitatively characterize linguistic labels, such as low priority. In our previous research work, the fuzzy membership function chosen for the weight 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.

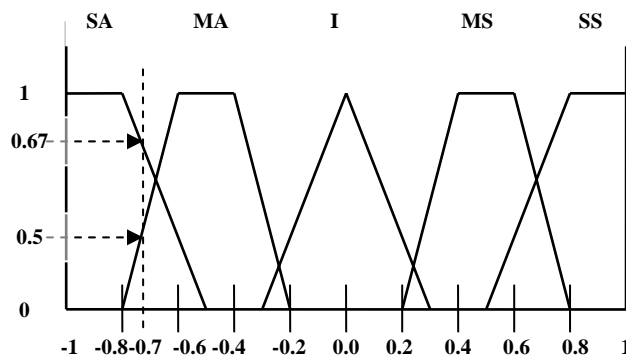
	H	M	L
SS	SS	SS	MS
MS	SS	MS	I
I	I	I	I
MA	SA	MA	I
SA	SA	SA	MA

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

Figure 19. Argumentation Heuristic Rules



(A)



(B)

Figure 20. (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 21. 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 21 represents the rule: “If X is Strong Support (SS) and Y is L (low priority), then Z is Medium Support (MS).”

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:

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

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

Figure 21. FAM Matrix

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 22 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 22 illustrates the

membership value for the priority input. The strength w_1 associated with the entry would be computed as:

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

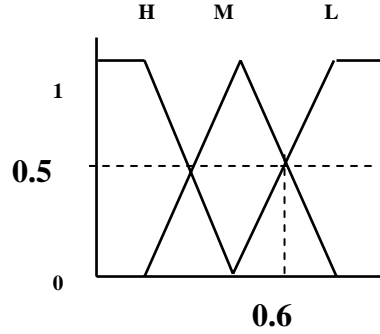


Figure 22. Membership Value for Priority Input

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 23 show the four activated rules for the values in the example. In addition to w_1 , there are three more nonzero weights.

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

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

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

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:

$$\begin{aligned} \text{Output} &= \frac{(w_1 \cdot MA + w_2 \cdot MA + w_3 \cdot SA + w_4 \cdot MA)}{(w_1 + w_2 + w_3 + w_4)} \\ &= -0.89 \end{aligned}$$

Conflict resolution by computing favorability

The favorability factor of a position is a value which gives the strength of the position. It is calculated by taking the sum of the strengths of arguments obtained by performing reduction on the ones which were not directly connected to the position. Such a measure allows the designers in a design deliberation to compare positions based upon the argument strength.

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

Figure 23. The Fuzzy Association Memory

In order to resolve conflicts and identify a good design concept, multiple design alternatives are usually developed and explored, and the one which satisfies all stakeholders' requirements to the highest degree is selected. These alternatives are known as positions. The designers would argue over each position giving their arguments and respective weights. In order to resolve the conflict, i.e. to decide which position is the best design alternative, we need to calculate the favorability factor for each position. The position with the maximum favorability factor is the best design option for the conflicting design issue.

At every point in the argumentation, the designers can view which position has the maximum favorability factor and can post their arguments accordingly. For example, a designer may observe that the favorability factor of a given position to which he is supporting is low. He may then decide to post a Strong Support on that position or a strong Attack on the argument having a Strong Attack on the position.

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 designer are contradictory among themselves. In a complicated collaborative design environment with a number of designers, the self-conflicting problem could take place frequently. What is even worse is that they are not easy to detect in many cases. A designer 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 designer has some self-conflicting arguments in the network, then no matter how powerful this designer 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 24, the owner of argument A1 is O1, 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 O1, then A1 and A5 are a pair of self-conflicting arguments of owner O1.

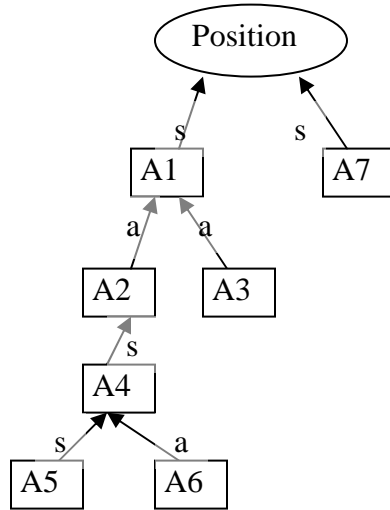


Figure 24. A Simple Example to Illustrate Self-conflicting

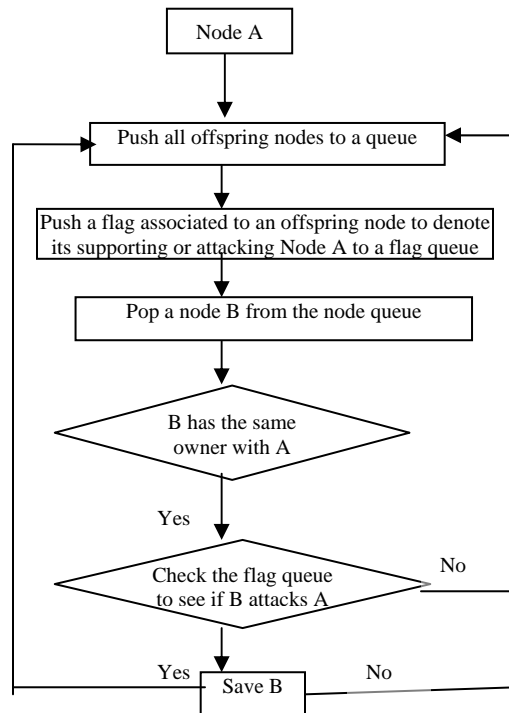


Figure 25. The Algorithm to Resolve the Self-conflicting

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.

CONCLUSION

The collaborative bridge design system provides a working environment for bridge designers to work spontaneously at different locations, discussing and sharing design options, design drafts and resolve the conflict among designers and help make the best design decision. At the same time, they are able to connect to the seismograms server to obtain seismogram data and the peak value of a specific time unit during a certain period of time.