Implementation of an interactive conference system

by THOMAS W. HALL

Language and Systems Development, Inc.
Silver Spring, Maryland

INTRODUCTION

This paper discusses a specific type of decision making through the use of a timesharing computer facility. The technique is variously called “conferencing” or “Delphi conferencing.”* In conferencing, a computer would serve as a data collection and routing device which enables a geographically scattered group of experts on some subject (the conferees) to conduct remotely those discussions and/or referendums that might occur at a conventional, face-to-face conference. A computerized conference system must then handle the mechanics of running such a remote conference. Conference systems have been studied recently and advantages of this type of system have been discussed:** for instance, the remote accessibility itself; the possible anonymity of the respondents as a control over personality factors; and the capability of participating in the conference at one’s convenience rather than at a precise time. The purpose of this paper is to present aspects of implementing a conference system. These aspects are of three types: functional (conferee oriented); control (conference chairman oriented); and implementation (program requirements to provide control and functional capabilities). Following this discussion of these three aspects, some features of an actual conference system implementation are reviewed.

FUNCTIONAL ASPECTS

The functional aspects of a conference system consist of what the user of the system can actually do. The user of a conference system—denoted a “respondent”—requires:

1. The ability to access a data base available to all the respondents. This data base is a representation of the activities of a conference—consisting principally of the texts of the things being presented for consideration by the respondents.
2. The ability to contribute to the activities of the conference in prescribed ways—expressing opinions, proposing new topics for consideration, and recording quantitative judgments on topics, e.g., by voting yes or no on a proposal.

In moving from these requirements to actually implementing a conference system, we see that what distinguishes an implementation of a conference system from a user’s point of view is then:

1. How accurately the system’s data base structure can mimic the conference situation the user needs; and
2. What the limits on his use of the system are—exactly what he can and cannot do.

From this point, it is possible to proceed to implement an almost endless variety of conference systems. What follows in this paper is a description of a system that has been implemented. Access to a prototype of this system was made available to a group of respondents—the subject being the conference itself. As a result of this experiment, the following functional system was designed and implemented.

Access

To participate in a conference, a respondent must first contact the computer. This normally entails using a remote device such as a Teletype(R) to call the phone number tying into the computer's timesharing system.
Next, the user has to provide the identification and accounting information the monitor system needs to allow him to sign on. Then the user has to indicate that he wishes to use the conference system rather than some other program. (The accessing procedure up to this point has been known to intimidate some first-time computer users, but it is just a fixed sequence of things to do ritualistically.) When the conference system is finally reached it takes control from then on. The conference system asks for a short code word which the respondent has been given by whoever arranged the conference. This code first of all allows the conference system to tell which conference the user is participating in—the system is capable of conducting several conferences at once, each with its own data base. The code also uniquely identifies the respondent within the conference so that his activities can be accounted for separately from all other respondents. Finally, this unique identification allows the system to give different privileges to different users. A given respondent may have any or all of the following privileges:

1. viewing the items, messages and votes (see the following pages for definition of these quantities;)
2. voting on items;
3. adding items or adding messages.

Interaction

When the conferencing system accepts the user’s code, it offers the user his choice of seeing an abstract of the subject of the conference, or of using a tutorial program explaining the system, or of proceeding to the heart of the interaction. When he has proceeded to the main interaction, he is given the choice of:

1. viewing summary information;
2. viewing items or messages;
3. voting on items;
4. adding items or messages (or amending those he has already added.)

Items, messages, and votes are the basic types of data handled by the conference:

1. Items are blocks of concise textual data which are to be presented to the respondents for their consideration and evaluation. Four types of items are recognized:

   a. Proposals—proposed actions or policies of which the respondents are asked to judge the desirability;
   b. Comments—discussion points which may be evaluated by the respondents for their importance;
   c. Facts—quantitative points of information which may be of interest in the discussion—respondents may evaluate facts on their pertinence to the subject of the conference;
   d. Estimates—requests for numeric estimates from the respondents—again the respondents may vote on the relevance of such estimates.

As noted, proposals, comments, facts, and estimates are evaluated by respondents according to the primary criteria of desirability, importance, pertinence, and relevance, respectively. Additionally, items can be graded by the respondents on secondary criteria—for example: agreement with a comment, feasibility of a proposal, confidence in an estimate, or impact of a fact. Items can optionally be associated with one another, linking together items on the same subject.

2. Messages are blocks of textual data which are not voted on by respondents and may be used for such purposes as pure discussion or to make comments about or clarifications of items. Messages may optionally be linked or associated with items, also.

3. Votes are the responses of the conference users to the items of the conference. Votes on the primary and secondary criteria are recorded on a scale of 1 to 5, with the exact vote distribution being made available to the respondents only after fifty percent of the vote is in and only after the respondent who requests to see the votes has himself already voted. User responses to estimate items are also part of the voting; the number, average, and standard deviations of the estimates being recorded, along with the number of estimates above the mean—a rough measure of skewness. As a result of the respondents’ votes, items are segregated according to those which have been accepted, those which have been rejected, and those which are still pending.

The process of adding items and evaluating those which other respondents have added constitutes the essential user interaction.

CONTROL ASPECTS

The control aspects of a conference system consist of the overall control mechanisms for a conference. In
the line of controlling a conference, the following capabilities must be examined:

1. organizing a conference;
2. exercising dynamic control over the proceedings of the conference;
3. examining the results of the conference;
4. disbanning the conference at its conclusion.

Again in our particular implementation, these capabilities appear in concrete form. Instead of identifying himself to the conference system as a regular user, one can take the role of a conference chairman or “monitor.” Obviously, the effort of finding people to participate in the conference and the task of guiding the conference proceedings along a coherent path still tax the human resources of the conference monitor. The conference system only takes care of the mechanics.

Setting up a conference

A user first creates the conference of which he is to be the monitor. In doing so, he establishes a monitor code or “key” which is required before he or anyone else can act as the monitor for that particular conference. Creating a conference entails:

1. establishing a list of respondents—supplying code words for their access to the system and indicating which permissions (votes, adding items, etc.) are accorded to each respondent;
2. providing the abstract of the conference's subject;
3. possibly providing an initial set of items for the consideration of the respondents.

Modifying the conference

Occasionally during the life of a conference, the monitor may have to intervene in one of the following ways:

1. to alter or completely purge inappropriate items or messages or to rearrange the association links between items into a more correlated structure;
2. to change the list of respondents—adding new ones or deleting old ones;
3. to communicate important information to the respondents in the form of a “monitor message” which is presented to each respondent the next time he uses the system.

Analyzing the conference

In addition to the vote totals made available to the respondents, the monitor of a conference can view the individual voting records of the respondents and can summarize the voting of various groups of respondents.

Deleting the conference

At the conclusion of a conference, the monitor directs the conference system to eliminate the conference from the computer system—reclaiming whatever storage the conference utilized.

IMPLEMENTATION ASPECTS

So far, this paper has discussed what a particular conference system does. This is a major part of implementing the abstract idea of a conference system—the designing part. Now, we turn to the actual realization of the design—making a working system to run a conference in the manner previously described. This part of the job—which is the part usually thought of as implementation—involves several areas of general concern which can be considered independently of a particular system.

Hardware and monitor systems

The nature of a conferencing situation as described up to this point makes evident some initial requirements.

1. Communications

To serve its intended purpose, the conference system must be convenient to geographically scattered users. Optimally, the system should accept interaction with the user from whatever remote device he finds most accessible. In practical terms this is best accomplished by such means as a dial-up Teletype arrangement as used by most commercial timesharing services. Where possible, higher speed devices including graphics display terminals, should be accommodated by the system for the user’s convenience.

2. Operation under a Timesharing System

At any one time, the conference system may be in use by no respondents or by several. In any case, the vast majority of the computer’s time would be spent
idle if it had to run only this one system. This problem and many others are solved by writing the conference system as a subsystem of a standard timesharing system. This subjects the user to variations in response time due to totally unrelated uses of the timesharing computer, but this disadvantage is insignificant compared to the advantages. With a standard timesharing system, the desired communications capabilities should be already included; the job of dividing time between respondents becomes the monitor system’s task instead of the conference system’s; and the power and flexibility of the timesharing system’s software become available.

**Required features**

Having made the decision to operate the conference system under a timesharing monitor, one must require the monitor system to allow all the operations absolutely essential to the running of a conference.

1. **Common Access**

First of all, the respondents must be able to run the program or programs constituting the conference system. If possible, the conference system might be placed among the standard components of the monitor system to make access to it as easy and natural as possible. Ideally, the conference system should be a re-entrant program so that several respondents could be served by the single copy of the conferencing program. Second, the monitor system must allow the concurrent use of a common area of mass storage (a file) by several respondents. Access to this area must be direct in the sense that what a user writes there can immediately be read by another user—i.e., all users agree as to what is in the file at any given time. This essentially requires unbuffered input/output to be used. In addition to providing common access to a file, the monitor system must also be able to limit it as noted in the next point.

2. **Exclusive Access**

When two or more user programs in a timesharing situation independently desire to change the contents of a common file, some means must be used to insure that their attempted changes do not interfere with one another. This is done in practice by guaranteeing one user program at a time exclusive access to the area he wishes to change. Three techniques are available for this purpose: first, a program might arrange to raise its priority sufficiently to guarantee that it can get done what it wants to get done without interruptions; second, the program might request the monitor system to temporarily grant it exclusive use of the entire file; and, third, the program might ask the monitor to grant it exclusive use of just the portion of the file that had to be modified. Any or all of these features may be available for use in a given timesharing system.

**Programming Systems**

In actually programming a conference system, one must choose a language and programming system which offers access to all the required features mentioned previously. It is of no use if the monitor system offers exclusive access to common files if the language chosen for the conference system cannot make use of this feature. It is sufficient to choose a language which can communicate with the presumably omnipotent assembly language. It is not completely unreasonable to write the conference system entirely in the assembly language, but use of a higher level language suggests itself for several reasons:

1. **Availability**

Oddly enough, many timesharing systems offer the remote user no access to assembly language programming. The philosophy behind this seems to be twofold: first, the timesharing user is not seen as requiring any of the features available only in assembly language; and, second, access to all the features open to assembly language programs may be hazardous to the stability of the system. Conversely, higher level languages such as BASIC and FORTRAN are made commonly available.

2. **Programming Ease**

Unless the programmer(s) assigned to code a conferencing system are of the extremely rare breed of systems programmers who prefer assembly language coding, the time required for implementing a conference system in an appropriate higher level language should be considerably less. Also, mistakes in higher level language coding are apt to be less troublesome—either because fewer mistakes are made or because they are more easily found and corrected with the help of the diagnostic aids accompanying more advanced programming systems.
3. Use of Packaged Features

In using a high level language, one can often make use of powerful features built into the language which would be difficult to rival in assembly language. The most important of these are the input/output operations which are often the most difficult part of assembly language level programming to master. Also, the higher level language should offer capabilities for manipulating string data and for data formatting chores.

4. Efficiency

Clearly, the execution of a conference system carefully written in assembly language would be more efficient in terms of computer usage than that of one written in a higher level language. However, this would appear to be of negligible interest for two reasons: first, the saving in effort of implementing a system in a higher level language should counterbalance the execution differences; and, second, the nature of a conference system is such as to make it entirely input/output oriented or “I/O bound,” waiting for the user’s input nearly continuously. This means that the response time for a user—which is the most important feature of such an interactive system—will not be appreciably affected by the inefficiencies in computing due to the higher level language. It also means that the dollar cost due to use of the computer’s processor will probably be smaller than the cost of calling up the computer in the first place.

FEATURES OF AN ACTUAL IMPLEMENTATION

To explore the requirements for implementing a conference system further and to show how the required features come into use in practice, the techniques used in an actual implementation of a conference system* should be examined.

Choice of computers

All the features necessary for implementation of a conference system are readily available on the UNIVAC 1108** system operating under the standard UNIVAC monitor system, Exec 8. The important features of the Exec 8 system for this purpose are: its timesharing capabilities—allowing remote users access to the full resources of the system, including the assembly language; and its flexible file handling capabilities.

Programming language

On the UNIVAC 1108 used, an extended version of the BASIC language, XBASIC,*** was available for use. The theoretical justifications for writing a user conference system in a higher level language, or a user programming system centered around such a language, have already been stated. Writing the conference system in XBASIC made available the powerful features of the XBASIC language, such as string, vector, and data file manipulation capabilities. Even more significantly, using the XBASIC system made writing the conference system much easier. In fact, the system was developed entirely from remote terminals—a rather remarkable occurrence: using one remote user oriented system (XBASIC) to develop another (the conference system). One final consideration in using XBASIC was that XBASIC is not a completely static system—if new features were seen to be necessary for such tasks as writing a conference system, they could have been added to the language readily to produce a better programming language at the same time. As a result of the conference system project, a re-entrant version of XBASIC specifically streamlined for running other user systems was created.

Data structures

1. Files

The data base for each conference (there can be up to 26 active simultaneously) is stored on two random access mass storage files. Names for these files are generated by the conference system and all control of these files is done internally by the system without the user's having to know anything about Exec 8 file handling. This internal allocation, use and release of files under program control without bothering the user with facility assignment or “data definition” control

---

* Implementation of the conference system described was done by the author for Language and Systems Development, Inc., 8121 Georgia Avenue, Silver Spring, Maryland, 20910. The work was partially supported by the Department of the Army, Corps of Engineers (Contract DACA31-70-C-0088).

** UNIVAC is a registered trademark of the Sperry Rand Corporation.

*** XBASIC is a proprietary processor developed by Language and Systems Development, Inc. See "XBASIC for the Univac 1100 Series Computers," Language and Systems Development, Inc. (1969)
cards is extremely desirable in a user oriented system. Two files are used by the conference to conserve storage. Since Exec 8 offers dynamically expanding files, the conference system only uses the space in these files as it needs to, using them as two pushdown lists. The two files are automatically placed by the conference system on the best (fastest) type of mass storage available at the time—ranging from 4.3 millisecond average access time for the fastest to 92 milliseconds average time for the slowest.

2. Data Items

The storage used for the conference system's files is actually only randomly accessible down to 28 word blocks. This presents the challenge of fitting the required data efficiently into convenient multiples of 28 word blocks. In most cases, this proved fairly easy. The primary simplifying assumption permitting the use of fixed size blocks of storage is that textual items have a maximum length (360 characters.) This is actually a requirement imposed in the design of the system to force brevity on the respondents, not a requirement imposed by the implementing programmer.

(For sample data block formats, see Figure 1.)

3. Program Structures

All together, the conference system requires somewhat more than 2000 lines of XBASIC program to function. However, only segments of the system are actually required by a user at any one time. To keep the respondents from tying up core storage with unused portions of the system, there are actually several programs comprising the system. When the user requests a feature of the system implemented by a program other than the one he is currently executing, the current program deactivates itself and calls in the new program (program "chaining"). The various programs are organized so as to minimize the number of chaining operations required. As an additional feature of the system, a respondent who is not entitled to vote is chained to an interaction program which does not contain code to perform voting—saving some amount of memory space.

(For organization of the program chains, see Figure 2.)

Control mechanisms

1. User Directives

The user directs the conference system by making choices at various junctures in the logic of the system.
The system indicates what the available choices are (or can be asked to skip the explanations for experienced users) and then requests the user's decision in the form of a numeric choice. The system does recognize such things as "YES" and "NO" choices; but for the more complicated multiple-choice decisions, making numeric choices is the best method in the long run—being faster to type with less chance of error than attempting to use keywords. Input is taken by the system in string form and decoded by the program so that it can recognize errors and act on them without subjecting the user to the monitor system's error messages about such things as improper numeric formats.

2. Common File Management

The careful control of access to the common data file of a conference is extremely important in the implementation of the conference system. The problem is complicated by the way items are stored in the files. To keep an upper bound on the amount of file storage used by a conference, only the fifty most recent items are kept available. The items are stored in the file in a circular manner—the fifty-first item is written into the area formerly used by the first item. The conversion from external numbers of items to their internal positions in the file is handled fairly easily with modular arithmetic. The trouble comes when the user asks for the first item as it is being outdated and replaced by the fifty-first. Other conflicts arise with two users attempting to update the vote totals at one time or two users attempting to add an item at the same time.

As indicated in the section on Functional Aspects, several methods are available for resolving conflicts in use of common files. In the Exec 8 system, one can either ask for exclusive rights to a file—if someone else is using it, the request is rejected and must be repeated; or one can ask for exclusive use of a small block of the file, "locking" it—if someone else attempts to reference this block, his request is automatically delayed until the lock is removed. This second method is by far the better—it offers much finer control over the file's use; and it turns out to be over thirty times faster to use, due to the way Exec 8 operates. The feature of locking and unlocking portions of the file is used to direct access to the common file via a method which could be called "key block" control. For example: to add an item, a user's conference program must lock the key block necessary for this function—namely the directory block containing a pointer to the next available place in the file. The user then adds his item, updates the directory block and unlocks it. During this time, other users can be accessing other portions of the file freely.

Obviously, the more key blocks are locked and the longer they are locked, the slower the system will respond to other users. Several devices are used to circumvent this:

a. The functions of key blocks are distinct so that, for example, a user adding an item only locks out other users attempting to do the same thing.

b. All preparations for changing the file are made before the locking is done; then the file is changed as rapidly as possible and unlocked. The principal rule here is to never have a key block locked while waiting for input from a respondent.

c. Control of key blocks is not required by all operations which just read information.

Note that (c) implies that the problem of attempting to read an item which has just been outdated has not yet been resolved. This problem is handled by having each item or similar type of record explicitly contain the external item number to which it corresponds. When the system reads an item, it checks to be sure that the item number in the data read matches the one it was expecting—if the check fails, the system knows the desired item has been outdated and tells the user so. Happily, this checking also protects the conference system against failures of the computer system. If the computer stops after an item has been changed but before the changes are made in the key directory blocks, the check for outdated items should discover the discrepancy.

EXAMPLES OF INTERACTION

The following pages are copies of actual sessions using the conference system. In these examples, the information in lower case letters or following question marks represents input supplied by the user. All of the details of the interaction and the meanings of some of the choices offered by the system are not discussed in this paper. The examples are presented only to give the reader the flavor of the interaction.

The first example covers the setting up of a conference (in this case with only one respondent—the creator of the conference). The second explores the entering of an item into the conference, voting on the item, and seeing the results of the voting. The final example shows the monitor having the results of the conference printed (the results being very sparse in this case).

The remote device used for the interaction was a Teletype Model 37. The cooperation of the Teletype Corporation in adapting this device for use with the computer system was greatly appreciated.
DELPHI CONFERENCE SYSTEM AT 141830 ON 123170
PLEASE TYPE YOUR CODE? monitor

DO YOU WISH TO:
SET UP A CONFERENCE (1)
DELETE A CONFERENCE (2)
MODIFY A CONFERENCE (3)
ANALYZE A CONFERENCE (4)
RETURN TO MAIN SYSTEM (+)
TERMINATE (-)
MONITOR CHOICE:? 

BY WHAT CODE SHOULD YOU BE KNOWN (5 SYMBOLS)? twh
ALL YOUR CODES SHOULD BEGIN WITH THE LETTER A
YOUR CODE IS AUTOMATICALLY ATWH
TYPE CODES FOR OTHER PARTICIPANTS.
WHEN ASKED 'VERIFIED?', HIT ONLY RETURN KEY IF OK, ANY Symbol IF NOT.
WHEN ASKED 'CODE:' AND YOU HAVE NO MORE CODES, HIT JUST THE RETURN KEY.
BUT FIRST, SUPPLY INFORMATION FOR YOURSELF:
WHAT CLASSIFICATION CODE? 11
LIST PERMISSIONS: VOTE(1), ADD ITEM(2), ADD MESSAGE(3):? 1,2,3
HIT RETURN IF FULL EXPLANATION IS TO BE GIVEN WHEN THIS CODE FIRST LOGS IN. TYPE ANY SYMBOL IF NOT. no
TYPE ANY TWO LINES OF INFO. FOR FUTURE REFERENCE:
? this is alphabetic information to identify this respondent.
? VERIFY: CODE= ATWH PERMISSIONS= 1 2 3
? CLASSIFICATION= 11
THIS IS ALPHABETIC INFORMATION TO IDENTIFY THIS RESPONDENT.

VERIFIED?
CODE:? 

IS THE 'WAIT' CHOICE ALLOWED. RETURN=YES, ANY SYMBOL=NO? no
THE SYSTEM WILL KEEP A BACKUP TAPE ASSOCIATED WITH THIS CONFERENCE.
SUPPLY A TAPE REEL NAME TO BE USED FOR THIS PURPOSE. IF YOU DO NOT
HAVE A TAPE ALREADY, HIT JUST THE RETURN KEY AND THIS SYSTEM
WILL REQUEST ONE FROM THE OPERATOR. TAPE NUMBER:? none
TYPE FOUR LINES OF SUBJECT DEFINITION.
FOR THE FIFTH LINE, TYPE YOUR NAME AND HOW TO REACH YOU.
? this is a demonstration conference.
? it has no real subject.
?
? this would be the monitor's name and phone number.

THIS IS A DEMONSTRATION CONFERENCE.
IT HAS NO REAL SUBJECT.
THIS WOULD BE THE MONITOR'S NAME AND PHONE NUMBER.

VERIFIED?
DO YOU WANT A 'MONITOR MESSAGE' NOW.
HIT RETURN IF YES, ANY SYMBOL IF NO.? no
CONFERENCE SUCCESSFULLY CREATED.
MONITOR CHOICE:? +
DELPHI CONFERENCE SYSTEM AT 145603 ON 123170
PLEASE TYPE YOUR CODE? atwh
DO YOU WISH
  AN EXPLANATION (1)
  SUBJECT DEFINITION (2)
  LONG FORM (3)
  SHORT FORM (4)
CHOICE? 3

DO YOU WISH TO:
  VIEW SUMMARIES (1)
  VIEW ITEMS (2)
  VIEW MESSAGES (3)
  VIEW VOTES (4)
  VIEW AND VOTE ON ITEMS (5)
  VOTE (6)
  ADD AN ITEM OR MESSAGE (7)
  MODIFY AN ITEM (8)
  WAIT (9)

MODE CHOICE:? 7
ITEM OR MESSAGE MUST FIT IN SIX LINES OF 60 CHARACTERS.
HIT RETURN KEY WHEN EACH LINE IS COMPLETED AND WAIT
FOR THE QUESTION MARK BEFORE BEGINNING NEW LINES.
YOU MUST SUPPLY SIX LINES EVEN IF THEY ARE PUT IN
BLANK BY HITTING THE RETURN KEY AT THE BEGINNING OF THE LINE.
ADD ITEM OR MESSAGE IN NEXT SIX LINES. END OF LINE IS HERE
* this is a test item for demonstration purposes only.
? treat this item as an estimate, when asked for your
? estimate, supply an arbitrary number within the given
? limits.
?
?

THIS IS A TEST ITEM FOR DEMONSTRATION PURPOSES ONLY.
TREAT THIS ITEM AS AN ESTIMATE, WHEN ASKED FOR YOUR
ESTIMATE, SUPPLY AN ARBITRARY NUMBER WITHIN THE GIVEN
LIMITS.

IS THE ABOVE CORRECT AS STATED:
YES(1), NO(2), OR NULLIFY(0)
CONTINUE CHOICE:? yes
INDICATE ASSOCIATION WITH EXISTING ITEM (NOT
MESSAGE) BY SUPPLYING EXISTING ITEM NUMBER.
IF NO ASSOCIATION, ENTER ZERO(0).
ASSOCIATION CHOICE? 0
INDICATE TYPE:
  PROPOSAL (1)
  COMMENT (2)
  FACT (3)
  ESTIMATE (4)
  MESSAGE (5)
TYPE CHOICE:? 4
WHAT IS THE LOWER BOUND FOR ESTIMATES? 0
WHAT IS THE UPPER BOUND? 100
INDICATE SECONDARY EVALUATION SCALE:
  NO SECOND SCALE (0)
  CONFIDENCE (1)
  AGREEMENT (2)
  FEASIBILITY (3)
  IMPACT (4)
  PROBABILITY (5)
  ARBITRARY (6)
SCALE CHOICE:? 0
TYPE: E SEC. SCALE: NONE ITEM >
LB: 0 UB: 100
ARE YOU SATISFIED WITH THE ABOVE CHOICES:
YES(1), NO(2), OR NULLIFY(+).
CONTINUE CHOICE:? 1
ITEM OR MESSAGE ENTERED: 123170 AT 151122
3E:
THIS IS A TEST ITEM FOR DEMONSTRATION PURPOSES ONLY.
TREAT THIS ITEM AS AN ESTIMATE. WHEN ASKED FOR YOUR
ESTIMATE, SUPPLY AN ARBITRARY NUMBER WITHIN THE GIVEN
LIMITS.
0<3
DO YOU WISH TO ADD OR MODIFY ANOTHER ITEM OR MESSAGE
YES(1) OR NO (2).
CONTINUE CHOICE:? 2
DO YOU WISH TO:
  VIEW SUMMARIES (1)
  VIEW ITEMS (2)
  VIEW MESSAGES (3)
  VIEW VOTES (4)
  VIEW AND VOTE ON ITEMS (5)
  VOTE (6)
  ADD AN ITEM OR MESSAGE (7)
  MODIFY AN ITEM (8)
  WAIT (9)
MODE CHOICE:? 5
DO YOU WISH ITEMS PRESENTED BY:
  LIST ORDER (1)
  SINGLY BY NUMBER (2)
  ASSOCIATIONS (3)
  THOSE NEW OR MODIFIED (4)
  THOSE ACCEPTED (5)
  THOSE SIGNIFICANT (6)
  THOSE PENDING (7)
  THOSE INSIGNIFICANT (8)
  THOSE REJECTED (9)
ORDER CHOICE:? 2
ITEM:? 3
Implementation of Interactive Conference System

3E:

This is a test item for demonstration purposes only. Treat this item as an estimate. When asked for your estimate, supply an arbitrary number within the given limits.

0 < 3
You have not yet voted on this item.

PER: Last vote: 3 Present choice: 63

Estimate between 0 and 100

No last choice. Present choice: 63

ITEM: 3

3E:

This is a test item for demonstration purposes only. Treat this item as an estimate. When asked for your estimate, supply an arbitrary number within the given limits.

0 < 3

Code: (1) (2) (3) (4) (5) (6) AVE

PER: 0 0 1 0 0 0 3.0

N: 1 A: 63.00 AA: 0

SD: .00 LE: 63.00 HE: 63.00

PER: Last vote: 3 Present choice: 1

Estimate between 0 and 100

Last choice: 63 Present choice: 65

(Note: that vote changes the item's status.)

ITEM: +

Do you wish to:

View summaries (1)
View items (2)
View messages (3)
View votes (4)
View and vote on items (5)
Vote (6)
Add an item or message (7)
Modify an item (8)
Wait

Mode choice: 1

Activity summary

There are no messages.

Items: 1 to 3

Types: P: 0 C: 1 F: 0 E: 2

Status: A: 1 S: 0 P: 2 I: 0 R: 0 Purged 0

Active voters 1
Active viewers 0
Total logins 7
Total votes 1
Total vote changes 2
Vote changes from no judgement 0
No judgements 0

Do you wish:

Association map (1)
Item summary (2)
Your vote summary (3)
Return to mode choice (+)

Summary choice: -
**DELPHI CONFERENCE SYSTEM AT 151230 ON 123170**

**PLEASE TYPE YOUR CODE? monitor**

---

**DO YOU WISH TO:**
- SET UP A CONFERENCE (1)
- DELETE A CONFERENCE (2)
- MODIFY A CONFERENCE (3)
- ANALYZE A CONFERENCE (4)
- RETURN TO MAIN SYSTEM (+)
- TERMINATE (-)

**MONITOR CHOICE: ? 4**

**WHAT IS YOUR MONITOR CODE? atwh**

---

**ANALYSIS OF DELPHI CONFERENCE NUMBER 1**

**DO YOU WISH INFORMATION ABOUT EACH RESPONDENT TO BE PRINTED?**
- YES (1) OR NO (2): ? 1

**WHAT ARE THE LIMITS ON FIRST DIGIT OF CLASSIFICATION CODE:**
- TYPE LB,UB: ? 1,9

**WHAT ARE THE LIMITS FOR THE SECOND DIGIT:** ? 1,9

**CODE: ATWH -- VOTING RESPONDENT**

**INFORMATION SUPPLIED BY MONITOR:**
- THIS IS ALPHABETIC INFORMATION TO IDENTIFY THIS RESPONDENT.
- NUMBER OF LOGINS = 7
- PERMISSIONS = 7 (PACKED FORMAT)

**ITEM NUMBER 1 VOTING RECORD:**
- HAS NOT VOTED ON THIS ITEM.

**ITEM NUMBER 2 VOTING RECORD:**
- HAS NOT VOTED ON THIS ITEM.

**ITEM NUMBER 3 VOTING RECORD:**
- LAST VOTE ON SCALE 1 = 1
- LAST VOTE ON SCALE 2 = 0
- NO. VOTE CHANGES ON PES = 2
- NO. VOTE CHANGES ON SES = 0
- NUMBER OF CHANGES FROM NJ = 0
- LAST ESTIMATE = 65

**TOTAL AVERAGES FOR THIS GROUP:**
- NUMBER OF CODES COUNTED = 1
- TOTAL LOGINS = 7
- VOTE CHANGES ON SES = 2
- VOTE CHANGES ON PES = 0
- VOTE CHANGES FROM NJ = 0

**ITEM AVE. ON PES AVE. ON SES**
- 3 1.00 NO SES

**END OF SUMMARY ******

**TIME: 1.522**

---

From the collection of the Computer History Museum (www.computerhistory.org)
ACKNOWLEDGMENTS

This paper is an outgrowth of work done on the subject of conference systems by Dr. Murray Turoff of the Systems Evaluation Division, Office of Emergency Preparedness, Washington, D.C. The author worked with Dr. Turoff in this area during the year of 1970. The design of the user's interaction with the conference system cited in the paper is due to Dr. Turoff—the expansion of the design into a complete conference system and the implementation of this system being the author’s concern.

The cooperation of the author’s associates at Language and Systems Development, Inc., should also be acknowledged—their ideas on systems programming being, as always, useful.