COMPUTERIZED CONFERENCING & COMMUNICATIONS CENTER

at

NEW JERSEY INSTITUTE OF TECHNOLOGY

THE EVOLUTION OF A TAILORED COMMUNICATIONS STRUCTURE:
THE TOPICS SYSTEM

BY:

Peter and Trudy Johnson-Lenz

RESEARCH REPORT #14

January 1981

c/o Computer & Information Science Department
New Jersey Institute of Technology
323 High Street, Newark, N. J. 07102

THE EVOLUTION OF A TAILORED COMMUNICATIONS STRUCTURE: THE TOPICS SYSTEM

by

Peter and Trudy Johnson-Lenz

November, 1980

ACKNOWLEDGEMENTS

The TOPICS system in its current form has been designed, developed, and documented by Peter and Trudy Johnson-Lenz with the involvement and assistance of many other people.

Special thanks go to: Participation Systems Inc. (PSI) and their Politechs projects (Legitech, Publictech, and Localtech) and to the Computerized Conferencing and Communications Center at the New Jersey Institute of Technology (CCCC at NJIT) for their initial involvement and assistance; to the Cooperative Extension Service offices at Colorado State University, New Mexico State University, and the University of Arizona for support during the Rapid Growth Communities project; and to the many users of the system who provided invaluable criticism and suggestions.

The TOPICS system grew, in part, out of the concepts and design of the original LEGITECH system previously designed and developed on EIES by Participation Systems Inc. and Johnson-Lenz with support from the National Science Foundation under grants DSI 77-17838 and DSI 77-27940.

The development of the TOPICS system was partially supported by grant number MCS 78-00519 from the Division of Mathematical and Computer Sciences of the National Science Foundation.

OVERVIEW: THE NEED FOR EVOLVING COMMUNICATION STRUCTURES

A computer-based human communication system should be designed for people's use, in response to their perceived needs and communications styles; no single system can meet the needs of all groups and individuals. It might seem that a general electronic mail or computerized conferencing system with a standard set of features should be able to meet most communications needs, in much the same way that the telephone system meets the needs of a wide range of users. However, there are many communications structures found in everyday life, ranging from one-to-many news broadcasts, to the many-to-many patterns of town meetings, from the unstructured and informal gatherings at the local pub, to highly structured meetings using Robert's Rules of Order. Each of these is an example of a specific communications structure appropriate in some circumstances and quite inappropriate in others.

Within a flexible computerized conferencing system such as the Electronic Information Exchange System (EIES), it is possible to tailor the features of the system to the needs of the users, rather than forcing them to adapt their communications behaviors to the system and its limitations. Current concepts and structures such as electronic mail and conferencing will be supplemented in the next decade by an ever-increasing array of specially designed structures to meet specific needs. Hiltz and Turoff (1978) discuss some of the promises and potentials for how human communication via computer will transform the ways we work, play, learn, and govern ourselves. They also discuss in some detail a variety of communications structures designed for group problem-solving and decision-making.

The major question addressed here is how these communications structures evolve. How are they initiated? Where do they lead? What forces govern their evolution? For a structure to be effective, it must meet the needs of the group using it. However, the perceived needs of a group may (and probably will) change over time. This means that as a group's needs change, either as it learns more about the medium or as its situation changes, the communications structure must EVOLVE to match those needs. Thus, the process of designing and implementing a communications structure becomes an ongoing process. Since it is generally recognized that the microelectronics and telecommunications "wave" of change we are now beginning to experience (Toffler, 1980) will transform the very fabric of our society, and since the communications procedures and structures we use in this electronic medium are going to evolve very rapidly in the next two decades, an understanding of the process of this evolution seems critical for our successful transition to a post-industrial, communications-era society.

A model of the ongoing process of design of these structures is introduced in Johnson-Lenz (1980c). Included there is the concept of GROUPWARE -- the integrated, systemic whole made up of a group's processes and procedures, PLUS software to support those processes and procedures. Most specific software structures can be used in a variety of ways, depending on the characteristics of the group and its perceived needs for process. Thus, the system which evolves is not only the computer software but also the process and procedures followed by the group to achieve its purposes, with or without software support; hence the term GROUPWARE.

This paper traces the evolution of a particular communications structure, the

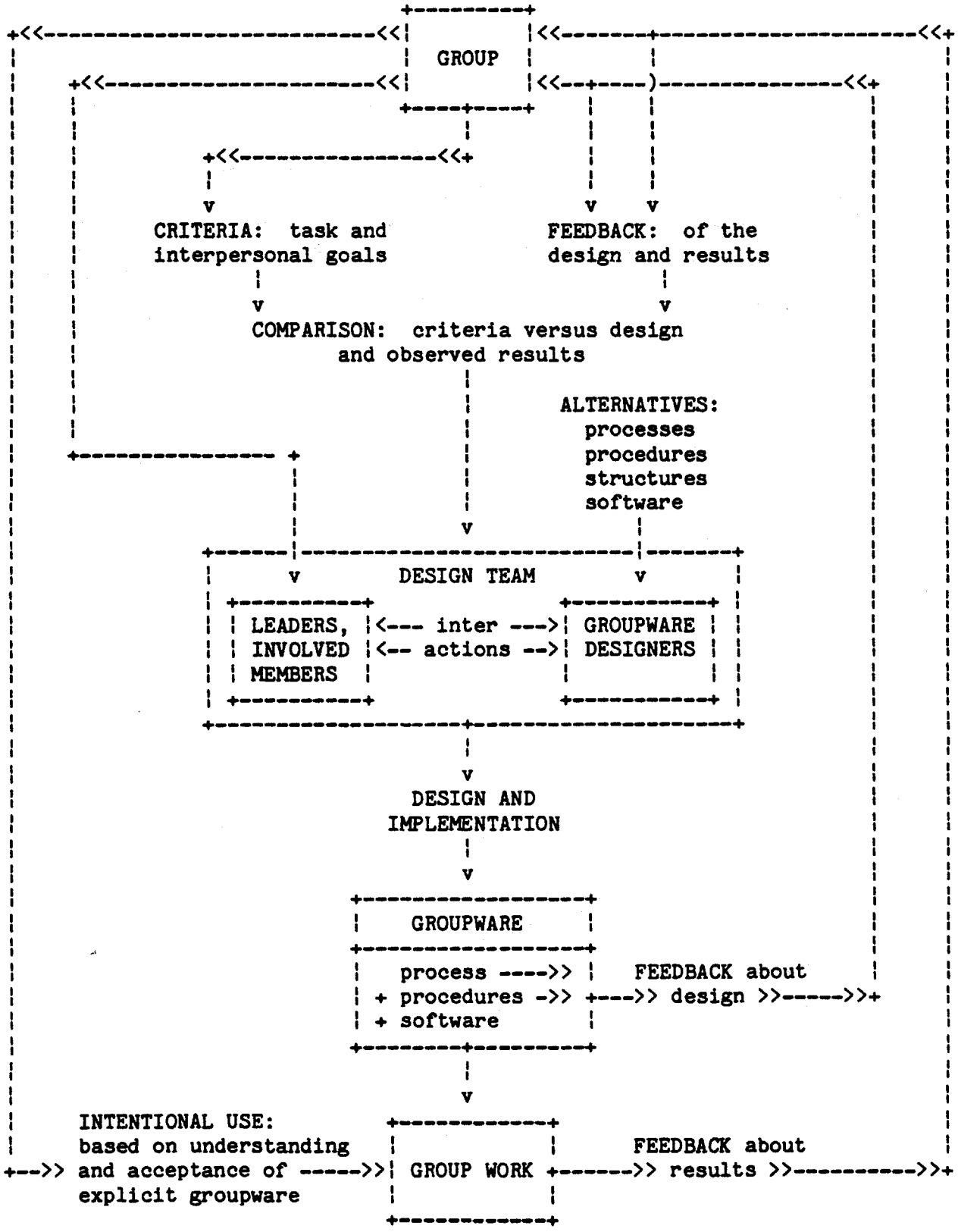
TOPICS system, as well as the evolution of several groups using that system, each with its own unique and evolving groupware supported by the TOPICS software, and each contributing its own unique set of needs to the evolution of the software. The TOPICS system, resident on EIES, was designed and developed by the authors, in collaboration with the groups using it.

TOWARD A THEORY OF GROUPWARE EVOLUTION

MODEL OF THE GROUPWARE DESIGN PROCESS

The process of groupware design, as shown in Figure A, begins with the group providing the design team with criteria by articulating its needs in terms of task and interpersonal goals to be achieved through the work of the group. The design team, consisting of group leaders and involved members plus the groupware designers, then begins by selecting PROCESSES which seems to meet those needs. Based on the processes selected, specific PROCEDURES and STRUCTURES are chosen which will support them. Note that up to this point, no software has been considered -- only the processes and procedures needed by the group, independent of any computer considerations. Then, only after the processes and procedures have been agreed upon, the design team moves on to the phase of selecting from among the array of existing, available software systems or designing and implementing a system tailored to support the specific processes and procedures previously decided upon.

Figure A: DIAGRAM OF GROUPWARE DESIGN PROCESS



At this stage the design team provides the group with feedback about the processes, procedures, and software chosen for its work. Group reactions are integrated into the design, which is then implemented. The group can then begin to learn how to use the groupware. The model continues by including a second opportunity for feedback to/from the group based on actual experience with the groupware.

Note that this entire process focuses on GROUPWARE rather than the software by itself. If the software alone were considered, evaluation of the use of the complete groupware system would be insensitive to either the characteristics of the group itself or the process and procedures being followed by the group. Ineffective group work might well be the result of an inappropriate match between the group's needs and the software selected. It might also be the result of inappropriate procedures being used by the group, software aside. The whole system of process, procedures, and software must therefore be considered when evaluating the experiences of the group in order to assess next steps.

The model of groupware design includes this final feedback loop based on experience with the system to show that the design process is ongoing. However, this model does not explore the dynamic process through which the groupware evolves over time, nor its impact, if any, on the design and evolution of the software system used to support the group's work.

TRADE-OFFS/BALANCES INVOLVED IN THE EVOLUTION OF GROUPWARE

Figure B shows that the process of evolution is driven by the needs of the group using the groupware. As in Figure A, everything always comes back to the expressed needs of the group and its reactions to the system it is currently using. There is no objective set of criteria as to what is appropriate. There is only the subjective base of experience of group members and the consequent design decisions made by the design team.

Figure B: TRADE-OFFS/BALANCES INVOLVED IN THE EVOLUTION OF GROUPWARE

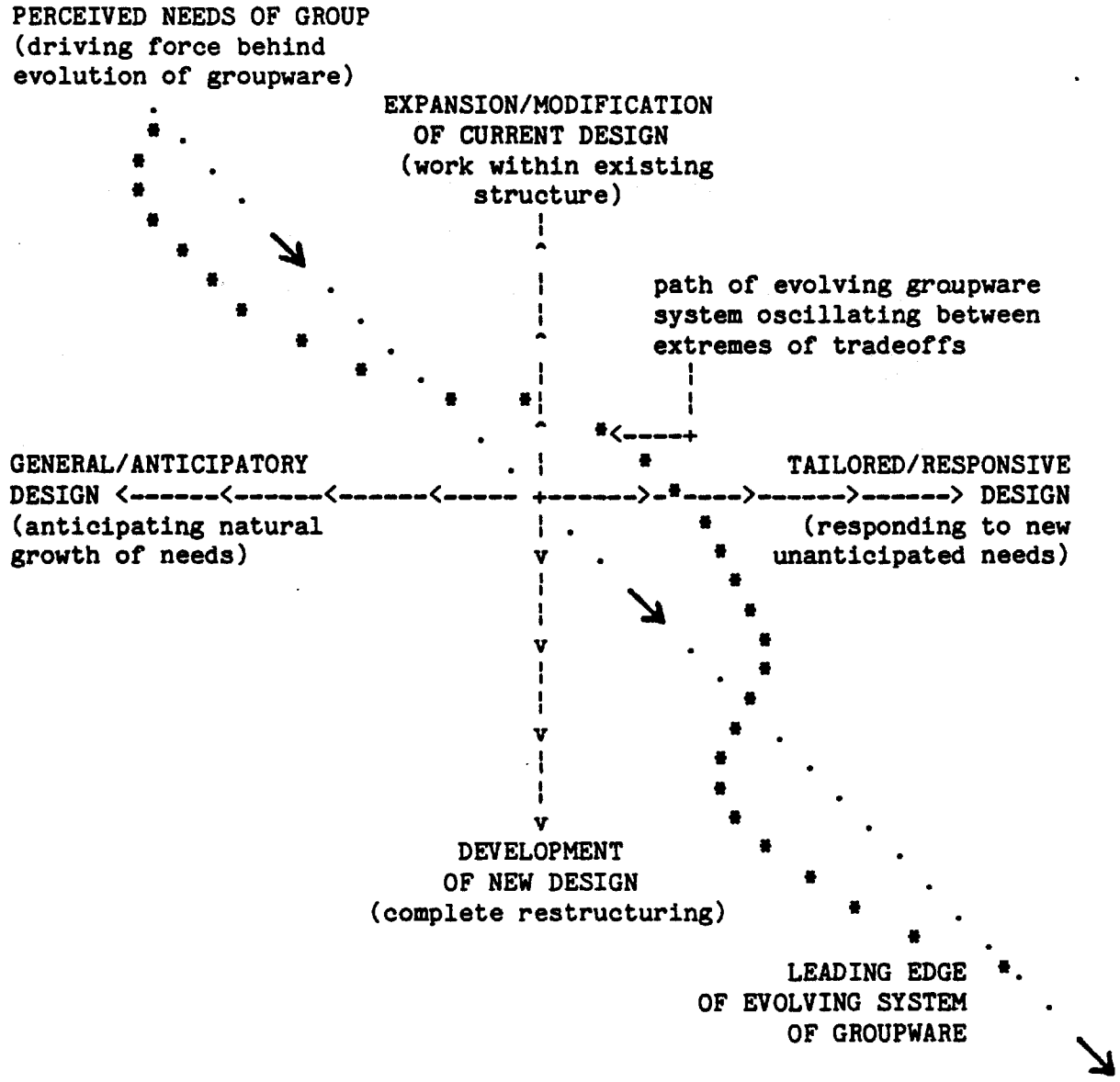


Figure B also shows two axes which cross the diagonal path of groupware evolution. These axes represent the two major trade-offs which must constantly be taken into consideration during the ongoing process of system evolution. While these trade-offs clearly apply to the software, they also apply to the group process and procedures and hence to the entire groupware system as well. The vertical axis represents the trade-off between working within the current system versus going to a completely new design. There are times to continue to work within an existing framework and there are times to abandon it and move on to a new one. This is discussed in more detail below.

The horizontal axis represents the trade-off between a general design and one which is specific (and hence limited) to a particular set of needs. On one hand, the investment of additional initial effort in design and development

can yield a more general system which anticipates the evolving needs of the group even before its members may realize them. This has the advantage that new capacities and procedures are designed and ready to use as soon as they are needed. On the other hand, it is of primary importance that the system be responsive to the emerging needs of the group, even though those needs may not have been anticipated through a general design. Also, unanticipated needs will undoubtedly arise. In some cases it is best to design generally, in anticipation of growing needs, while in others it is best to implement specific procedures without generality simply to retain a high level of responsiveness to unanticipated needs.

The "path of evolving groupware" (the diagonal) oscillates between general and tailored design in a dynamic and unpredictable fashion, never staying very long at one extreme. If too much effort is spent on a general design, the system will not be responsive to current needs and the "pendulum" will move to the other side for balance. Similarly, if the development process is immediately responsive without accumulating a good, general foundation, the overall cost of the system will increase as patches and one-shot developments add to an ever-increasing accumulation of ad hoc inefficiency. The same holds true along the vertical axis. If the current framework for the groupware is stubbornly held to, the emerging unmet needs of the group will eventually force a shift to a totally new system. Similarly, too frequent restructurings into new systems will trigger a conservative response among users seeking some measure of stability and familiarity.

Thus, the path of the evolving system is a constant, unpredictable oscillation between these extremes. The design team must make rather arbitrary decisions between these four paired trade-offs based on experience, feedback from the group, and intuition of future needs. The design process is an evolving, constantly changing art, and the direction of the evolution of the groupware is a compromise or optimization of dynamic forces that guide its overall path toward the realization of the group's self-chosen and evaluated goals and purposes.

LOCATION OF WINDOW FOR EVOLUTION OF GROUPWARE

Figure C: REGIONS OF MATCH BETWEEN GROUP NEEDS AND EVOLVING SYSTEMS: LOCATION OF WINDOW FOR EVOLUTION

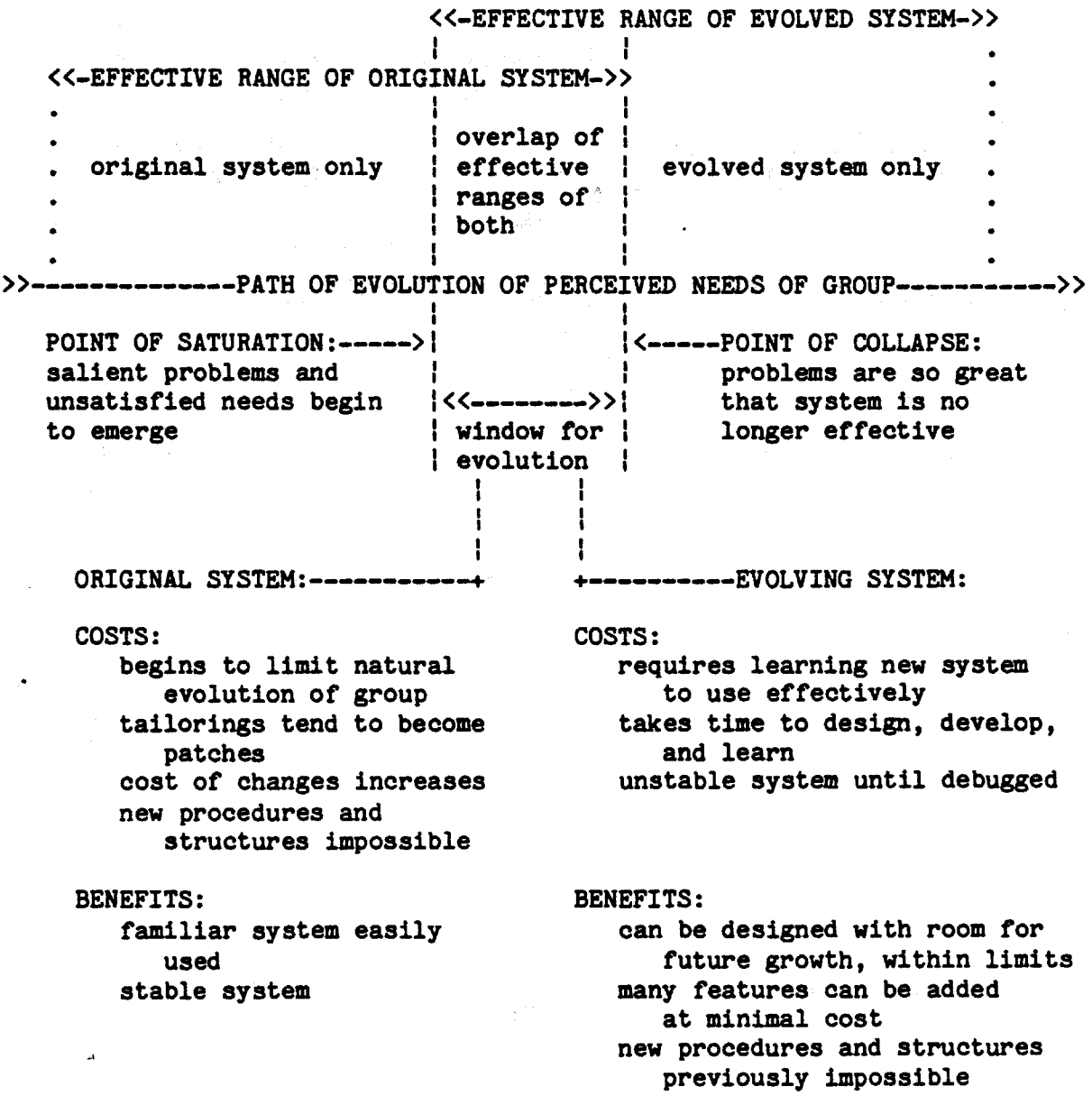


Figure C provides a more detailed view of the dimension involved in the vertical axis from Figure B. The long horizontal arrow represents the same flow of evolution of the needs of the group and its consequent groupware as shown in Figure B. This diagram shows a period at the left during which the original or current system is effective in meeting the group's needs. At the far right is a period during which only an evolved, restructured system will be effective in meeting the group's needs. In the middle is a zone in which both systems are effective: the original system decreases in effectiveness while the evolved system increases in effectiveness. This middle zone is the WINDOW for evolution of the system. It begins with the point of SATURATION at

which the current framework of procedures begins to be exhausted or saturated by users who encounter its inherent limitations and begin to complain about them. The window ends with the point of COLLAPSE at which effective group work within the original structure becomes impossible. Since the group needs to communicate and coordinate its activities during the transition to a new, evolved system, the changeover must begin BEFORE the point of collapse. Hence, this point defines the end of the evolutionary window and in effect, the end of the group's work. Evolution of groupware is not possible after the communication of the group has collapsed.

As the group's needs evolve during passage through the window, the costs (in the broad sense) of continuing to use the original system increase while the benefits decrease. Meanwhile, the costs of the evolved system decrease while its benefits increase. The increasing costs of continuing with the original system include limiting the natural evolution of the group's work, increasing patchiness of the system's design, increasing cost of modifying and maintaining the system to meet the group's needs, and the impossibility of implementing certain procedures and structures ruled out by the design of the system. The decreasing benefits of the original system include familiarity, ease of use, and stability.

The decreasing costs of the evolved system include time and effort required to learn a new system, lack of familiarity with new procedures, and using an unstable system during development. The increasing benefits of the evolved system include capacity for general/anticipatory design, potential for new procedures not previously possible, and low cost of any specific new feature if developed within the context of a total restructuring of the system.

These concepts are certainly not limited to the evolution of groupware systems. de Bivort (1980) has developed a general theory of evolution and evolutionary management -- the deliberate facilitation and management of systems undergoing evolutionary change -- on which some of these ideas about the evolution of systems are based.

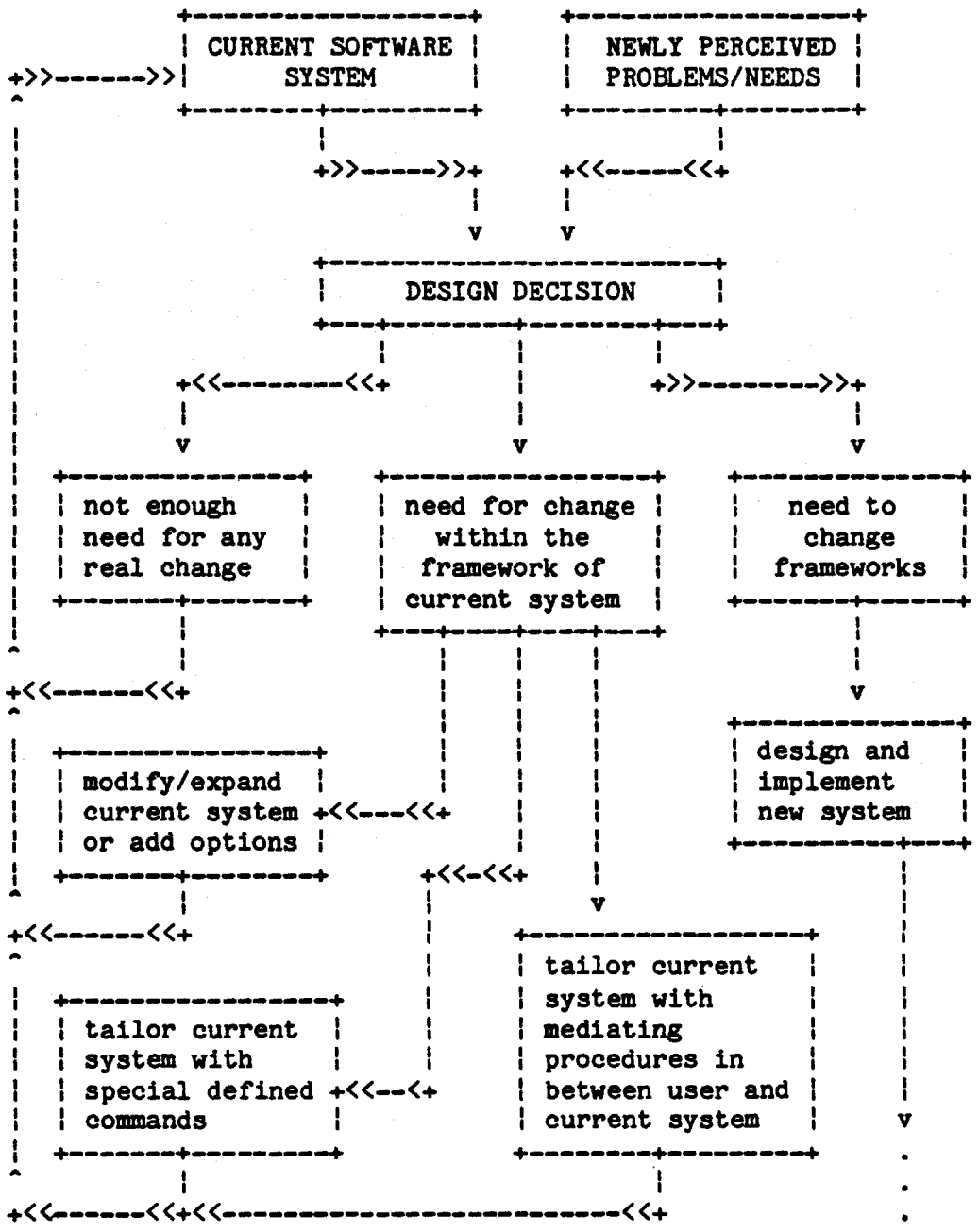
ALTERNATIVES TO THE CURRENT SOFTWARE SYSTEM

During the period that the group is within the evolutionary window, the design team must choose a course of action. Early in the window there are more options than later on. Often such options include modifications to the original system which will prolong its effectiveness, while nevertheless ultimately bringing on eventual restructuring. Thus, some options can be chosen which make the window of evolution longer than it might be otherwise. It should be noted, however, that such options are best chosen during the early part of the window before problems with the current system have become too great and consume a large portion of the group's resources.

Since the major purpose of this paper is to explore the process of software evolution, focusing on the specific example of the TOPICS system, Figure D below shows some of the alternatives available to the software designer confronted with newly emerging group needs somewhere in the middle of the evolutionary window. Before exploring these alternatives in detail, however, it will be helpful to establish a general framework for consideration of software systems.

It is becoming increasingly popular among computer scientists to think of a software system as a framework within which certain operations can be performed on certain objects (or operands). Often this is expressed in terms of a general database which contains objects (e.g., messages; group members, each of whom has a name, address, description; groups; conferences, each of which has members and comments; etc.) and the operations which can be performed on those objects (e.g., add, modify, delete, copy, etc.). This, in turn, can be thought of as a user language in which there are certain nouns (objects) and certain verbs (operations). Thus, from the user's point of view, the software system is controlled by entry of commands which tell the machine to perform certain operations on specified objects: modify a message, add a member, etc. One of the major limiting characteristics of a software system is this "vocabulary" of objects and operations. Only those objects (nouns) included the vocabulary can be manipulated, and then only in those ways specified by the list of verbs. Therefore, any group process or procedure which involves any concepts (or objects) NOT in the vocabulary simply cannot be supported by the software. For example, a system which does not recognize the noun VOTE obviously cannot support formal voting. Similarly, a system with no verb MODIFY (or some functionally equivalent term) will not allow modification of material once entered.

Figure D: ALTERNATIVES TO CURRENT SOFTWARE SYSTEM



Now, within that framework, let us look at the alternatives displayed in Figure D. First, there may not be any pressing need for change, in which case the current system can be used as is until problems motivate further considerations.

Second, the current system can be modified to meet current needs. In many cases features can simply be added within the existing framework. Nouns and or verbs can be added. As long as the general framework can absorb the addition of these new concepts, there is no problem. It is as if a new, relatively independent section of the system is added. The rest of the system

remains the same. Users need only learn about the new features when needed for their work, and the effective life of the system is extended.

However, sometimes new features cannot simply be added, largely because the newly perceived needs of the group warrant a change in EXISTING features. In some cases this presents no problems and the modification can proceed directly. But direct modification of a system often creates more problems than it solves, since some users of the modified feature would have preferred to be able to use it in its unmodified form. One approach that can be taken to "tailor" the system to meet both the needs of those who want the new way and those who want the old is to create options. Options can be user-specific, such as those which support different formats for output for people with different terminals or different command interfaces (menu versus terse commands, for example) but which nevertheless perform the same operations on the database. Options can also be group-specific so that the software functions differently for different groups. In the discussion of the TOPICS system, below, several such options are described which allow the system to support a variety of differing group needs within a single framework.

Third, through the use of user-defined and group-defined commands the very language that people use to express their wishes to the computer can be tailored to a vocabulary appropriate to their particular process. The definition of such commands allows a user or group leader to specify automatic "translation" from a vocabulary familiar to and appropriate to a particular group into a more general and yet less appropriate vocabulary used by the system itself. Several different examples of the use of such defined commands will be discussed below in the context of evolution of the TOPICS system.

Fourth, specially programmed software procedures can be used to mediate the interaction of users with the computer system. Such a procedure "inserts" itself between the user and the general computer software system and processes all inputs from the user to the system, sometimes translating them into the language of the system, sometimes performing additional checks and error-correcting feedback not normally performed by the system, and in other ways "tailoring" the interface between the user and the system. With this approach it is possible to modify the current system to meet new needs without having to completely rewrite the system, simply by writing appropriate mediating procedure(s). The result is user satisfaction and an extension of the useful life of the current system. Several examples of the use of these procedures in the evolution of the TOPICS system are discussed below.

Finally, in some cases, modification/extension of the current system, even through the use of defined commands or mediating procedures, is not enough to make the current system responsive to the needs of the groups using it. The vocabulary of the system is simply too limited or otherwise inappropriate. The framework of the system is too constraining to absorb the evolutionary pressure without a complete reconceptualization and restructuring at a new level of integration not possible within the old framework. In such situations, the evolutionary window cannot be lengthened. The only alternative is a new system. The history of the TOPICS system begins with such a situation in which its precursor system, LEGITECH, could not evolve to support the growing needs of the groups using the system, and a totally new framework, called TOPICS, was designed and implemented as the best

alternative.

SOFTWARE EVOLUTION IN EIES: TOOLS FOR TAILORING

The TOPICS system and its precursor, LEGITECH, were developed by the authors using the high-level programming language, INTERACT, developed by the Computerized Communications and Conferencing Center at the New Jersey Institute of Technology. INTERACT runs on EIES, the Electronic Information Exchange System, which is itself largely written in INTERACT. The language was designed to facilitate the rapid development of tailored communications structures in the electronic medium.

INTERACT includes a very powerful, general, tree-structured and keyword-accessed database capability that supports the design and creation of a wide variety of specially structured databases. Built directly into this database system are features allowing differential access (read only, write, etc.) to various portions of the database. Such access is, in itself, one way of imparting structure to a communications process merely by regulating who can perform which operations on which objects in the database.

INTERACT also includes direct access to basic communications primitives such as a text editor, terminal communication and control, indirect editing (output processor), input and command control (input processor), queuing, and event creation and execution. In the context of a structured programming language, these primitives can be combined into many different communications structures through which group members may add, modify, and delete elements in the shared database, using the system to communicate and do its business.

The design of INTERACT also allows for easy overlaying or interfacing software "beings" in between the user and the basic EIES system. The TOPICS system itself is such a mediating "being" between the basic EIES/INTERACT communications functions and the user. TOPICS tailors this vast potential to the limited needs of TOPICS users. This can be shown with the following diagrams. Normally, a user interacts directly with the computer system:

USER <<------>> EIES

However, the full potential of EIES can be tailored by inserting an INTERACT procedure, such as TOPICS, in between the user and EIES so that the user interacts with TOPICS and in turn TOPICS interacts with EIES:

USER <<------>> PROCEDURE <<------>> EIES

USER <<------>> TOPICS SYSTEM <<------>> EIES

The potentials for creating additional software "beings" to mediate between the user and the system do not end there. As exemplified in the discussion of the TOPICS system that follows, additional layers of procedure can be inserted between the user and previous layers of procedure to further tailor the interaction to the user's needs. For example, the PARTY procedure was used to create an easily used process for conferencing which used the database and features of the TOPICS system in a deliberately simple way appropriate for introducing new users to the medium and for operating in a synchronous,

simultaneous conferencing mode.

USER <<------>> PARTY <<------>> TOPICS SYSTEM <<------>> EIES

In addition to these very powerful features of INTERACT, individual users and groups of users can use the defined command feature on EIES to summarize complex command strings often confusing to new users into a single word, which when entered into the computer by a user, automatically invokes the full set of commands associated with it. This allows different groups to call a particular procedure by different names. For example, the Politechs networks wished to use the name POLITECHS for their software system instead of the general name TOPICS. They defined a command, called +POLITECHS (a + on EIES denotes a command) which was equated to the command +TOPICS so that Politechs users simply typed +POLITECHS without realizing that the computer translated this into +TOPICS for them automatically. Another example of this was using the +PARTY procedure under the command name +HOPES for a group more interested in discussing their hopes for the future than having a party. The software used in both situations was exactly the same. Only the name was changed through the use of defined commands.

EVOLUTION OF THE TOPICS SYSTEM

Figure E: FLOW OF EVOLUTION OF THE TOPICS SYSTEM

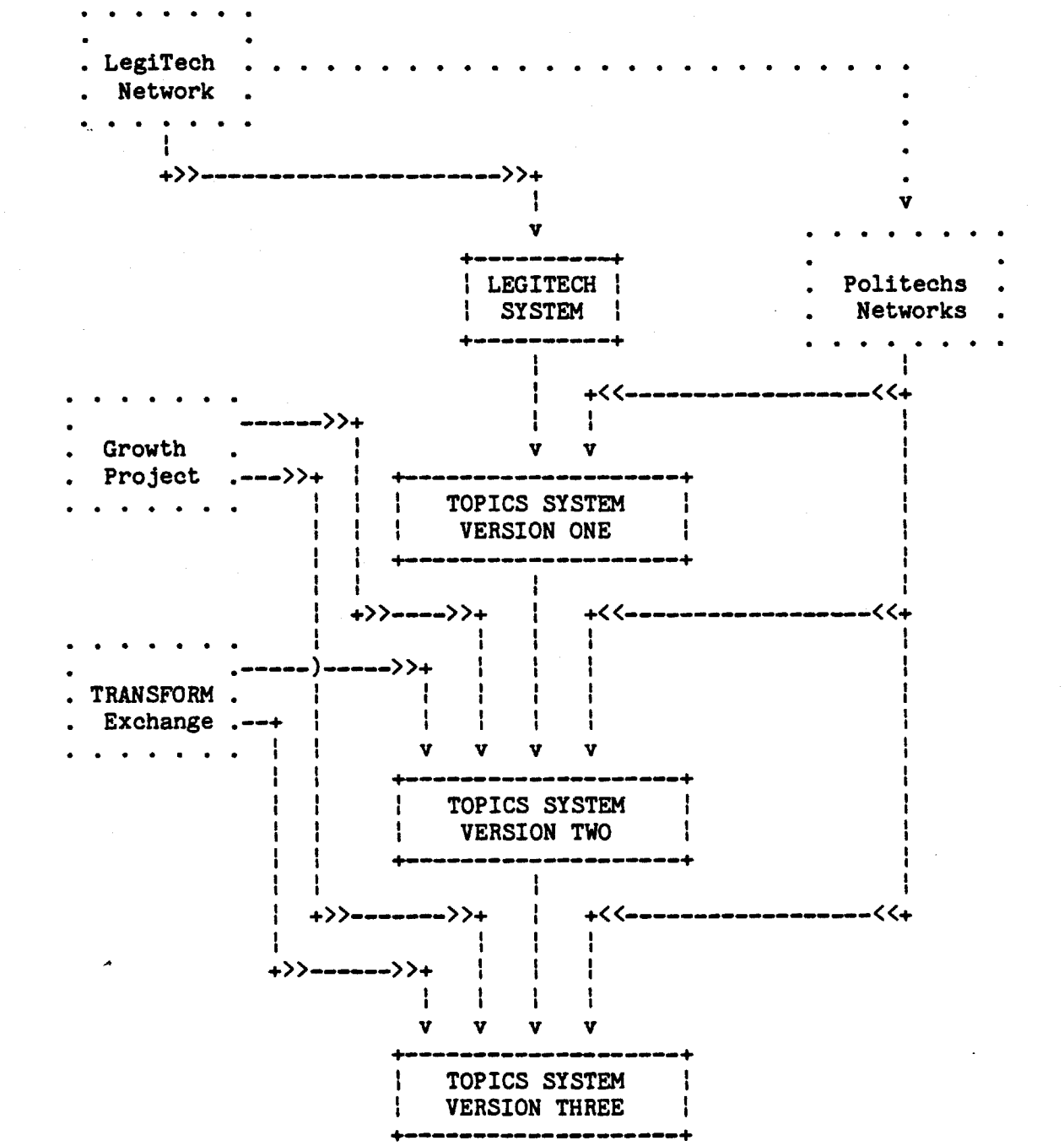


Figure E shows the overall flow of evolution of the TOPICS system. It began in 1978 when Participation Systems Inc. (PSI) received a grant from the National Science Foundation to investigate the potentials of using EIES for establishing a computer-based network among state legislative science research units. PSI, and its president and principal investigator on the project, C. H. (Harry) Stevens, had already developed a print-based exchange network among the legislative research units in several eastern states and had gained experience in working with such groups and their needs. Thus, the LegiTech (Legislative science and Technology) network came to EIES with specific needs for computer support for its work.

LEGITECH CONCEPTS

The authors were invited to be part of the design team for LegiTech's use of computerized conferencing. The design team also included Harry Stevens, Jim Williams, and Murray Turoff (the designer of EIES). LegiTech came to EIES with a well-developed set of concepts in the vocabulary of its groupware processes which worked as follows. Any member of the network with a question poses an INQUIRY to the network. An inquiry is a short, pointed statement of the question. BACKGROUND material on the problem or issue can be added as a response. Any other members of the network who have an answer to the question enter a RESPONSE to the inquiry. In LegiTech, a response could also take the form of a LEAD to another potential source of information, including books, publications, people, and organizations. PSI had been using these inquiry and response concepts in the print-based exchange and so already had the understanding and support of group members for this process and the more detailed procedures that they had already been using. These procedures included specific formats for inquiries, background, responses, and leads. They also included a system for identifying network members and norms and procedures for turning inquiries and associated responses into more well-edited inquiry/responses (later called BRIEFS) for broader circulation in the non-computer-based reaches of the LegiTech network.

The design objectives for the LEGITECH software system were to meet the needs of the LegiTech network at a modest cost and to support the process outlined above. However, as the design team met, concerns began to be raised that some network members would get overwhelmed by inquiries and responses unless some mechanism were included to "filter" the output so that only those responses of interest were automatically delivered by the system. So the objectives were expanded to include some mechanism for filtering the exchange of inquiries and responses.

To implement this latter objective, Jim Williams suggested the notion of a PROFILE for each member, showing those inquiries which were of interest. The computer could use this list to determine whether to deliver a response to any particular user. After some discussion and consideration, it was decided to replace the word PROFILE with SELECTION, and so the concept of SELECTION was added to the vocabulary of inquiries and responses. A user's SELECTION is the list of inquiry topics s/he is interested in and wishes to follow over time.

The final general objective of the development was to keep the cost down, since the budget was quite limited. Using INTERACT, it was possible to design a rather straightforward system using existing EIES notebooks for the database

-- one each for inquiries, responses, leads, and inquiry/responses (briefs). The software so developed was not general, since that would have raised the development cost. It was a one-shot system which worked only for the LegiTech network and no other groups.

FEATURES OF LEGITECH

In late November, 1978, the LegiTech network began using the LEGITECH system for filtered exchange of inquiries and responses. Some of the salient features of the system were:

-- upon entering the exchange system, the user was informed of the number of any waiting items, if any, and was asked a simple question as to whether s/he wanted to ACCEPT WAITING ITEMS (Y/N)?.

-- if the user accepted waiting items, all responses to previously selected inquiries were delivered first, followed by any new inquiries. As each new inquiry was delivered, the brief question (3 lines) was printed out and the user was asked if s/he wished to SELECT it or not. If the user did select it, s/he would receive all responses to it, both those already entered and those which were entered in the future.

-- after accepting waiting items, or if the user chose not to accept them, s/he was taken to LEGITECH CHOICE? where a numeric menu provided access to INQUIRIES, RESPONSES, LEADS, INQUIRY-RESPONSES, and the user's SELECTION. A second menu then allowed the user to GET, DISPLAY, COMPOSE, or MODIFY any of the above. Thus, the user was given the power to perform all necessary operations on the relevant objects in the LEGITECH database. Through these menus a user could compose an inquiry or response, or even modify the contents of his/her selection, if desired.

```
ACCESS TO:
 INQUIRIES           (1)      DO YOU WISH TO:
 RESPONSES           (2)       GET            (1)
 LEADS               (3)       DISPLAY        (2)
 INQUIRY-RESPONSES   (4)       COMPOSE        (4)
 YOUR SELECTIONS     (8)       MODIFY         (5)
 EIES                (9)
LEGITECH CHOICE?
```

-- the activity of the LegiTech network as it used the LEGITECH system automatically created an organized database of inquiries of concern to state legislatures and associated responses to those inquiries.

-- the system used regular EIES notebooks which gave the LegiTech editors special access directly to the database.

-- a member's selection was limited to 28 inquiries, largely to keep the database design simple and to speed up processing of waiting

items.

-- all inquiries, responses, and leads were formatted under strict computer control to conform to formats based on existing group norms.

-- copy commands were provided so that members could copy material of interest with a simple command and then send it as an EIES message.

-- some members were "trapped" into the LEGITECH software when they logged into EIES so that they would not be exposed to the additional demands of learning how to use EIES and LEGITECH all at once.

As part of the development, documentation was written to help users. Since time was short, only a brief two-page introductory summary was written by the authors. This brief document was entered into EIES and could be retrieved by any user by entering ?LEGITECH. The LegiTech project staff wrote more extensive documentation later on, once the project got underway.

PROBLEMS WITH THE LEGITECH SOFTWARE

During the 18 months of the LegiTech project, users reported a variety of dissatisfactions with the LEGITECH software. They noted the lack of a capacity to search for topics of interest by keyword. Although the standard EIES notebook search features were available, they were slow and required a different set of commands to use. Related to this was the lack of an alphabetical index to inquiries and responses. Users also reported confusion about the formats. They were not certain which type of lead format to use in some cases -- organization or person, book or publication.

Probably the most frequently reported problem was frustration with the limit of 28 inquiries in a selection. The software forced users to remove specific inquiries from their full selections before it would allow them to add new ones. Users did not like having to do this housekeeping chore simply to be able to get new responses to new inquiries.

Other problems reported were the desire to give read-only access to some members and confusions about the simultaneous access to the database through LEGITECH and standard EIES notebooks.

During the project period, the LegiTech organizers, PSI, began promoting an expanded version of inquiry/response networking, called Politechs. They conceived of an overlapping set of networks, each of which had some specific group focus such as LegiTech. Johnson-Lenz (1980a) describes some of the directions of the evolution of the LegiTech groupware into the Politechs concept. So, as the LegiTech project matured, PSI's perceived needs expanded for a more flexible system which would support more than a single exchange like LegiTech. PSI's needs had evolved to a point well within the window for evolution of the system (Figure C above); the LEGITECH software had passed the saturation point, as described in the problem list above.

The authors, as developers of the software, also felt that the LEGITECH design

had reached its limits for evolution. The software was not written generally. The code included numerous references to specific EIES notebooks and would have been difficult to convert into a multi-exchange system. INTERACT had also evolved in the meantime, and the advantages of using some of the more powerful new features of INTERACT, plus the clear need for general code, far outweighed the advantages of trying to modify the existing system. It was time for complete restructuring and development of a new system evolved from the original LEGITECH software design.

TOPICS VERSION ONE

CONCEPTS AND FEATURES OF TOPICS I

During the summer of 1979, as the LegiTech project was concluding its first year, the accumulated problems encountered in using the limited LEGITECH software, plus a variety of other considerations, brought Harry Stevens of PSI and the authors together once again to design and develop an improved version of what was becoming to be called an "inquiry/response" system. The design objectives were to address the problems and limitations of the LEGITECH system, to allow for easy creation of new exchanges, to write the system generally with the best coding techniques to maximize the potential for further evolution without redesign, to allow other groups in addition to PSI's Politechs networks to begin experimenting with such a filtered exchange system (and possibly to make it a general feature available on EIES eventually), to make the system as compatible with the old LEGITECH software as possible to reduce the stress of moving to a new system, and to add better facilities for tracing the activity of users. The new software system was called TOPICS.

It was decided that the concepts of leads and briefs (edited inquiry/responses) could be phased out. The formats for leads seemed to create too many problems and the concept could be folded into the concept of a response, of which a lead was a special case. Since briefs were not a central part of the highly interactive inquiry/response process it was decided to drop that concept from the menu and use either an EIES notebook or a special exchange within the TOPICS system for such edited material. Politechs currently uses the Brieftech Exchange for such briefs.

The first concept added to the LEGITECH vocabulary was TOPIC -- the sum of an inquiry and all its associated responses. It was added to provide conceptual clarity when referring to all responses to a particular inquiry. It was also added to suggest the potential for using the new software to support a process somewhat different than inquiry/response. This new process came to be called "mini-conferencing" since the software, in effect, allows any user to begin a computer conference on a topic of his or her choice, all within a given exchange.

The concept of an EXCHANGE, a separate database of inquiries and responses, was also added to the vocabulary. Each exchange contains MEMBERS and GROUPS of members. Furthermore, each member has an assigned LEVEL OF ACCESS to the database. Level 0 is read-only. Level 10 is respond-only (cannot raise topics). Level 20 is regular access (respond plus raise topics). Level 30 allows editing of any material. Level 40 allows one to add and remove members and groups. Level 50 allows one to set the monitor options detailed below. Level 60 is for system implementors/programmers who need special access in order to fix bugs. Each level has the access of all levels with lower numbers. In addition, the concept of LEVEL OF ACCESS brought with it specific roles: EDITOR, GATEKEEPER, and MONITOR.

Since the specific nature of the groupware procedures for Politech's use of the system was not completely firm, and since the software was being designed

to be useful to more than one group, the authors included the concept of MONITOR OPTIONS. These options, controlled by the MONITOR (access level 50), govern the way the system functions. They can also be changed as needed. They include:

-- the maximum number of lines allowed in an inquiry or "topic raiser"

-- whether those who have received but not selected a topic can respond to it

-- whether the exchange is open publicly to all members of EIES or is private and open to specified members only -- and if public, what the default level of access for incoming members is

-- the limit, if any, on the number of topics allowed in a user's selection

-- whether a record is kept showing each time a member authors a topic and/or response and each time a member selects a topic

-- whether the list of keywords is open to new keywords entered by users or is limited to a fixed thesaurus (see discussion of alphabetic index below)

-- the maximum number of keywords that can be assigned to a topic during composition

-- whether an indirect editing command (such as .text) is automatically inserted at the beginning of each item (inquiry or response)

-- the location of the default topic marker -- the number of the first topic delivered to new members, regardless of when they join the exchange

To provide both an alphabetically sorted index of topics and rapid search and retrieval of topics by keyword, an ALPHABETIC INDEX OF TOPICS was added to the design. A user can print either the full alphabetic index or a section it. The index lists all topics by number which are associated with each keyword. Another menu choice allows direct retrieval by keyword. The software maintains an inverted file of all topics associated with each keyword to facilitate rapid search and retrieval.

The same basic structure of LEGITECH -- an initial opportunity to accept waiting items, followed by menus providing complete access to all objects in the database and all operations on those objects -- was maintained, largely to ease the changeover from LEGITECH. Several new objects were added to the menu, including the ALPHABETIC INDEX, MEMBERS OF THE EXCHANGE, MEMBERS OF TOPICS (those who had selected a particular topic), GROUPS OF MEMBERS, and OTHER EXCHANGES.

The single delivery question in LEGITECH (ACCEPT WAITING ITEMS?) was replaced by a series of three questions. The reasons for this were complex. In

LEGITECH all responses were numbered from 1 to N, regardless of the associated inquiry. For clarity of structure and access, this was replaced by a system of numbering all responses from 1 to N WITHIN a given topic. In LEGITECH a single response marker was maintained, showing the next undelivered (waiting) response for each user. In TOPICS, however, this was replaced by a response marker for each topic. As a consequence, the database can handle a much larger array of responses without getting overloaded. However, this also meant that counting to see how many responses in selected topics were waiting for users was more time consuming. Especially during peak loads on the system, this processing caused a noticeable and frustrating pause. So an initial question, REVIEW WAITING ITEMS (Y/N/A)?, was added. Users in a hurry who did not wish to see a count of how many items were waiting could say A (for Accept) and be taken directly to the second question, ACCEPT WAITING ITEMS (Y/N/O)?. This question was much like the LEGITECH question, except that it also allowed entry of the letter O (for Options). If the user entered an O, a third question, DELIVERY OPTION (B/T/R/N/KEY/T#)?, was asked. At this point the user could enter B and get all waiting items delivered in "batch" without any interactive opportunities to select topics (this appealed to users who liked to print out everything at once), T to get waiting topics only, R for responses only, N to skip delivery altogether, a KEYword to get only responses to topics so keyed, or T# to get responses to a specific topic (with number #) only.

In addition to the variety of new features already described above, the TOPICS software allows pagination control so that users can have their printing terminals skip to the top of a new page for each new topic, an option for users to get their own items delivered to themselves if so desired, and private topics which can be sent to specific members of the exchange if a private discussion or inquiry is involved.

The TOPICS system also allows easy retrieval of information about members of an exchange, members of a topic, and various facts about each member, such as selection size and number of topics authored. It also allows users to move their markers back and forth if they want to skip over a certain range of waiting items.

EXPANDED MODULAR DOCUMENTATION FOR TOPICS

As part of the development, the authors wrote 24 pages of more detailed documentation for the system which was necessary to explain all the new features. This new documentation was entered into EIES and made available via ?TOPICS in modules which could be selected from a menu of sections.

TECHNICAL CHANGES

Invisible to the user, many dramatic changes took place underneath the surface of TOPICS during its evolution from LEGITECH. First, the use of EIES notebooks was dropped and the much more flexible database features of the INTERACT tree-structured index system were used for organization and storage of the variety of objects in the expanded database. Even the software and documentation was stored in the index system. This made organization of an efficient, expandable, flexible database possible. Such was not the case with

the limited EIES notebook structure never intended for such uses.

To support the variety of delivery options and varied sequences available to the user for review and delivery of waiting items, the concept of delivery queues was established. If a user chose to accept waiting items without any review beforehand, items were delivered as they were found by the delivery software. However, if a user chose to review waiting items first, they were accumulated in the queue for later delivery to avoid the system having to find them twice. Since the task of computing which responses were waiting involved looking at the user's selection and his or her markers, the efficiency of using the delivery queue is obvious. Its use also made possible a variety of complex delivery capabilities such as delivering a private topic to a newly invited member, even after that user's topic marker had been moved beyond the number of the private topic.

VARIETIES OF GROUPWARE SUPPORTED BY TOPICS I

The Politechs networks were the first to use the new TOPICS I software. They began using it in October, 1979 in parallel with the LEGITECH software. At first they opened up the Publictech Exchange, a public exchange in which new members of the Politechs networks could familiarize themselves with the groupware of Politechs inquiry/response and where small networks could be developed into larger ones which could eventually spin off into their own Exchanges. After several months, the Legitech network moved into the second Politechs Exchange, called, obviously, the Legitech Exchange. To tailor the TOPICS system to the groupware of the Politechs networks, a defined command was used, as explained in an earlier section of this paper, so that Politechs users typed +POLITECHS to get into PUBLICTECH or LEGITECH instead of typing +TOPICS. This helped to create a special atmosphere around the Politechs Exchanges. PSI also published the "Politechs Information Sharing Networks NETWORKBOOK" and held a special Networkshop in October, 1979 as part of the Politechs activities.

Later in 1979, the authors, in collaboration with Robert Theobald, co-covenened their own TOPICS exchange, called the TRANSFORM exchange. The groupware of this exchange differed from that of Politechs in two striking ways. The purpose of TRANSFORM is to bring together a network of people who share the belief that we are in a time of fundamental transformation of ourselves and our society (person, planet, spirit) and wish to work together in an environment of trust and cooperation on transformational issues. All members of the exchange are asked to agree to a covenant which expresses these values and beliefs in order to focus thinking on the transformational emphasis and style of the exchange. This exchange is used as an open "mini-conferencing" system with several topics used for bulletins of news items, many others for discussion of areas such as the impact of religion and science fiction on transformation, and still others for the management of the TRANSFORM exchange itself. Johnson-Lenz (1980b) includes a more detailed discussion of some of the activities in the TRANSFORM group.

Again in collaboration with Robert Theobald, the authors worked on another project which attempted to use the TOPICS software to support the exchange of problems and successes among a network of half a dozen communities in the Southwestern United States undergoing the shared problems of rapid growth due

to the "energy boom." A third variant of groupware using the TOPICS system was developed by a design team for this project which included the authors, Robert Theobald, community development facilitators, and staff of the Cooperative Extension Service in Arizona, New Mexico, and Colorado. The GROWTH exchange used the TOPICS software both to support the exchange of inquiries and responses among participating communities and to hold mini-conferences on project management and other topics of interest.

In addition, the authors have tried other unsuccessful experiments with the TOPICS software, including an attempt to bring together a network of people in the appropriate and community technology areas, and an exchange on networks and networking in which a presentation for a conference was begun.

One of the biggest differences in the way these different groups used the software is their choice of monitor options. The Politechs Exchanges finally chose to prevent users from entering keywords when raising topics. Editors come in afterwards and assign keywords to conform to a standard set of hierarchically structured terms. Meanwhile, the GROWTH and TRANSFORM exchanges allow users to enter whatever keys they choose. The Politechs Exchanges set the limit to 3 lines per inquiry while the others use a limit of 5. The different needs of each group can be met within the general framework through use of these options.

PROBLEMS AND LIMITS OF TOPICS I

Below is a list of problems and limitations of the TOPICS system, version I, reported by users:

-- no pen named or anonymous entries (reported by members of TRANSFORM)

-- no voting features (reported by members of GROWTH and TRANSFORM)

-- uncertainty about highest topic and response numbers when requesting a printout of items (reported by users in all exchanges)

-- index format not good for printed index (reported by the Politechs editor)

-- need for nested and/or Boolean searches (reported by various users)

-- various problems with the delivery queue, such as users reviewing waiting items, not accepting them, and then complaining of having many forgotten but undesired items delivered before being able to receive newer ones (reported by various users)

-- problems with the use of the EIES copy command (&<) in the scratchpad during composition of topics and responses -- not being able to copy in material from EIES into an exchange

-- difficulty finding what users wanted to know in the documentation (reported by various users)

-- documentation lacking examples and a flowchart showing how to get around in the system (reported by GROWTH project members)

-- slowness of system, especially during peak loads

-- desire to trace user activity with greater detail and ease

In addition, the general matter of how to best make use of the keyword index was much discussed by most groups involved. The authors engaged in a long debate with the Politechs editor who believed that the ability to print out a readable index of topics by keyword was of primary importance. Since the index lists keywords in alphabetic order and since the Politechs editor believed it essential that these keywords be listed in a hierarchy, he implemented a keyword system in the Politechs Exchanges which involved the use of multiple keywords such as TRANSPORTATION:AUTOMOBILE:SAFETY with the more general terms in the hierarchy always included in the multiple key.

The authors objected to this approach since it made it difficult to use the automatic keyword retrieval features. They claimed that a user would have to learn the hierarchy to be able to enter the complete, multiple-part key to retrieve something by keyword. They preferred the simple, single keyword approach which allowed direct retrieval on the keyword SAFETY rather than TRANSPORTATION:AUTOMOBILE:SAFETY. The debate was never settled. All the exchanges in which the Politechs editor was responsible for the keyword index used his hierarchical system and had the advantage of having a more easily read and used index. Other exchanges used the approach preferred by the authors and had the advantage of easier search and retrieval without having to read the entire index. The authors/designers sought a solution which would allow easy searches and also provide a good, readable print form of the keyword index. That solution is discussed in CONCEPTS AND FEATURES OF TOPICS II below.

SPECIAL TAILORING OF TOPICS I

During the Spring of 1980, the authors' work with the GROWTH and TRANSFORM groups included desiging and developing a specially tailored procedure within TOPICS to support what came to be called as "intensive exchanges." An intensive exchange is a simple computer conference in which all participants get all comments entered by all other participants. It differs from a conference, however, in that it is INTENSIVE, since it is used in either a synchronous mode in which all participants are on line at the same time (which is overwhelming!) or in a near-synchronous mode where a group of people meet intensively over a period of days, unlike most regular computer conferences which often take months.

The idea of an intensive exchange began when a GROWTH project member articulated a need to have "computer parties" to demonstrate the new technology to participating communities by having simultaneous face-to-face parties with refreshments and friendly company in each of the participating communities, all connected with each other via EIES. Some discussion was held in the GROWTH exchange on the topic of a computer party. Some members expressed concerns over having to use the complexities of the TOPICS system in a community party setting where they did not want to make mistakes. The authors proposed a tailored procedure that simplified the user interface in the following ways. The procedure first asked members some basic questions, such as who is at your party and what issues and problems are of concern to folks in your community. As answers were typed in, the computer automatically delivered the answers to all other participating communities. Once a participating community had answered these initial "ice breaker" questions, they were given the opportunity to begin entering unstructured comments. After each comment was entered, the computer automatically delivered new comments from other communities and then waited for another comment to be entered. The process continued like this until the community logged off. The procedure was simple to use and very effective. People enjoyed the experience a great deal, even though at times the procedure was so overloaded by the simultaneous exchange that it became frustratingly slow. The GROWTH project had two such "computer parties."

At the software level, the PARTY procedure written for the GROWTH project created an intermediate "being" in between the TOPICS system and the user. This "being" asked users the ice-breaking questions and then gave them an easy way to share comments with users in other communities. It recorded all answers and comments within the GROWTH exchange in a pre-assigned topic set aside for that purpose. Thus, all the access and editing features of the TOPICS system were available for further work with the record of the intensive exchange.

Exactly the same procedure was used under the name HOPES by the TRANSFORM exchange when members decided that they needed a ten-day intensive exchange to share their hopes for and interests in the future of the TRANSFORM exchange. In addition to the initial questions which asked about individual hopes and interests, the authors also developed a set of "voting" routines or response/tallies which allowed the facilitators of the intensive to ask all members questions on a scale from 1 through 9 (or less where approriate) and

then provide everyone with feedback of the tallies of these responses. Members were asked if they were excited by the visions for the future of TRANSFORM during the intensive (on a scale from 1 to 7) and also if they felt there was a place for themselves in those visions. Participants gave high positive ratings to both of these questions. The authors were pleased with the results, even though the group's follow through was more disappointing, and they used the opportunity to develop this basic "voting" software which was to later be incorporated into the TOPICS system directly as a standard feature.

TOPICS VERSION TWO

CONCEPTS AND FEATURES OF TOPICS II

Late in the spring of 1980, another modification cycle in the evolution of the TOPICS system was undertaken to address the limitations and problems reported by users to date. Included in this cycle of changes were several new features which had not been part of the system before.

The first new concept added to the TOPICS vocabulary during this cycle was TALLIES, a form of voting or responding to numerically scaled questions. Two forms of the tally were added to the software. The first is a TOPIC/TALLY, which is associated with an entire topic and to which the recipient of a topic is asked to respond EACH TIME any responses to that topic are received. Thus, it is an ongoing tally or straw vote on the entire topic whose results are updated and changed as the flow of responses and discussion in that topic continues. The second form is a RESPONSE/TALLY, which is associated with a particular response only and to which the recipient is asked to respond only when that particular response is delivered. Between the two forms, a variety of voting and response procedures can be set up in an exchange to meet a wide range of group needs.

The alphabetic index to topics by keyword was improved in several ways. First, to increase the usefulness of the printed index and to allow rapid searching via keywords, the concept of RELATIONAL KEYS was added. This means that any keyword can be associated with any number of other keywords in a network of relationships. The possible relationships include BROADER THAN, NARROWER THAN, and RELATED TO. Thus, the editor of an exchange has the power to associate keywords with each other, including arranging them in a true hierarchy using only the BROADER and NARROWER THAN relationships. The Politechs keyword hierarchy can be maintained without having to use the multiple, colonated keys which confounds effective retrieval via keyword. When the keyword index is printed, each keyword is listed, showing all associated topics and all associated keywords so that the reader can find a keyword of interest and then easily see what additional keywords might also be worth looking at. Furthermore, during an interactive search by keyword, following retrieval of a list of topics for one key, the system lists all related keywords so that the searcher can then enter a narrower keyword to reduce the size of the "hit" list or a broader one to expand the list if it is too small. This lays the foundation for BOOLEAN searching as well. When entering a second or third key to refine a search, the user is now asked to specify whether the new key is to be combined with the first in an AND, OR, or NOT relationship. This feature supports much more complex, tailored searches not previously possible in the TOPICS system.

In addition to the variety of features to trace user activity already in the system, it was decided to add what is called the P by I (Persons by Interests) matrix. This matrix lists each member of the exchange as a row and each topic in the exchange as a column. A cell in the matrix can contain any of three things: (1) a one if a member has selected that topic, else zero; (2) the number of responses to that topic received by a member; or (3) the number of

responses to that topic authored by a member. Recent advances in the analysis of social networks have shown that such data is very useful in revealing the underlying social structure of overlapping interests among people (Johnson-Lenz, 1979). Cluster and other analyses of a P by I matrix can yield new understanding about how the group is evolving and who shares interests with whom. This feature in the TOPICS system provides a powerful and yet concise method of data collection for group evaluators to trace a group's activity in an exchange.

The ability to enter PEN and ANONYMOUS topics and responses was another feature added during this modification cycle.

The software was also modified so that any time the system asks for a response or topic number or a list of same (e.g., when displaying topic titles), it always prints out the highest topic or response number used to date. In addition, during delivery, the software now reports the number of associated responses for all new topics before asking the user if s/he wants to select them. Thus, the user has some idea of how much information s/he is asking for.

The procedure for modifying topic and response markers was modified so that it allows users to "clear" their delivery queues of waiting items if they had reviewed waiting items and not had them delivered subsequently.

A new monitor option was added to allow tracing each time a user either enters or leaves an exchange. This made it possible for group facilitators to know much more about when and for how long users were in an exchange.

CHANGES TO DOCUMENTATION FOR TOPICS II

The documentation was completely rewritten during this phase of the system's evolution and nearly doubled in length. The modules were organized into a decimal-numbered hierarchy (such as 1.2 and 1.2.3) so that readers can access the most general modules and then work down the tree into more specific material without having to print it all out.

A flowchart was added which shows the relationship of parts of the system and how a user can get from one place to another. Many users reported that they found this flowchart quite helpful in figuring out where they were in the system. In addition, examples of interactions were added to the documentation to show users just what the printout would look like and how to get the system to do what they wanted.

Basic procedures or "hooks" were added to the software to allow eventual addition of on-line documentation in response to either one or two question marks (? or ??) entered by a user at any question asked by the system. EIES already has such a feature and perhaps sometime in the future similar explanations can be written for TOPICS.

TECHNICAL CHANGES IN TOPICS II

One major problem reported by users was general sluggishness of the system,

particularly during peak use times. The EIES implementors had made some important changes to INTERACT which made it possible to speed up the TOPICS system somewhat. First, instead of using the EIES scratchpad editor directly from within TOPICS, which slowed down entry of text since the mediating procedure had to first process all inputs before sending them on to the editor, the SIMSP (simulate scratchpad) system procedure was installed in the TOPICS software. This procedure temporarily allows the TOPICS mediating "being" to "step aside" while a user enters and/or edits text directly and then calls the TOPICS system back to mediate and control the process once the text is entered. The response time during composition increased noticeably.

One benefit of using the SIMSP procedure is that users can now copy EIES text items (messages, conference comments, etc.) directly into the TOPICS scratchpad for inclusion in topics or responses. Before, using the system developers' privileges during composition meant that only items to which the developers had access could be copied in. By using the SIMSP feature, the user's privileges are in effect during composition instead.

The TOPICS system had been using four logical units to open files on the EIES disk. Up to this time, INTERACT had been limited to four such units. As a consequence, these units had to be shared among many files and the software was constantly opening and closing them to access the variety of files needed to support the full system, which slowed the system down. During this cycle of modifications, the system was modified to use 9 of the 10 logical units now available in INTERACT with a consequent improvement in system performance.

PROBLEMS AND LIMITS OF TOPICS II

Members of the GROWTH project reported that while they liked the tally feature, they found no way to display the tally results without having to re-enter their responses.

Members of all groups reported continued frustrations with the three initial delivery questions and asked for simplification. They also requested some initial explanation of how the delivery questions worked.

The procedures which listed members of an exchange and members of individual topics had bugs which created problems when previous members who had been deleted from EIES but not from the exchange were listed.

The Politechs networks expressed a need to be able to track how many responses had been received by any user. They were developing an experimental procedure for charging users, based on credits for items contributed and debits for items received. To do this, they needed a count of how many responses were received by each member; the system already recorded how many were authored. Politechs wanted software support for this economically oriented groupware.

TOPICS VERSION THREE

CONCEPTS AND FEATURES OF TOPICS III

By fall of 1980, the Politechs networks were preparing for a new year of inquiry/response networking and wanted to incorporate into the software the billing mechanism mentioned above and a simplified version of the initial delivery questions.

The authors undertook a new cycle of modifications. The first change involved adding the automatic count of responses received by each user. These data provided the basic information for the Politechs billing procedures discussed in the section below on SPECIAL TAILORING OF TOPICS III.

The next change involved simplification of the three delivery questions. It was decided to replace the three questions with a single one which asks:

> ACCEPT WAITING ITEMS (yes/no/display/batch/topics/responses/key/help)
> ACCEPT (Y/N/D/B/T/R/K/?)?

A response of Y results in regular interactive delivery. N takes the user directly to the menus. D (for Display) is the equivalent of the old initial question, REVIEW WAITING ITEMS?. Thus, the vocabulary changed from REVIEWing waiting items to DISPLAYing them. B yields "batch" delivery previously hidden away as a delivery option. T gives the user delivery of topics only, and R delivery of responses only. Entering K results in a second question, (KEY/KEYPHRASE/T#)?, to which a user can respond with either a keyword or a specific topic number. If a user enters either one or two question marks, s/he receives some explanation of these delivery alternatives.

To ease the transition to this simpler accept and delivery question for users more familiar with the previous complex of three questions, the software was set up to accept responses in the old three-question format, converting them to the new format. Thus, users who forget that the system has been changed are able to use the old responses. For example, users previously entered +TOPICS,EXCHANGE,A,O,B to get "batch" delivery of waiting items by answering ahead on the REVIEW, ACCEPT, and DELIVERY OPTIONS questions. Under the new system, +TOPICS,EXCHANGE,B is the proper entry, but the software automatically converts A,O,B into just B in case a user forgets the new protocols.

Finally, a new monitor option was added which allows the monitor to specify a set of "default" topics. The default topic marker option had previously been implemented so that the system would automatically set a new member's topic marker to a place specified by the exchange monitor to avoid overwhelming new members by giving them only the more recent topics. However, this created a problem in that important initial topics such as introductions, explanations, and bulletins were skipped over. The new default topics option allows the monitor to specify certain topics which will be delivered to new members, regardless of where their initial topic markers are set.

The welcome lines printed by the system when the user enters an exchange were

modified to compress the printout and to provide the number of the highest topic entered to date so that users can get a quick overview of activity in the exchange each time they enter it.

The documentation was updated to incorporate these new features. The updated documentation is available in print form as GUIDE TO THE TOPICS SYSTEM (Johnson-Lenz, 1980d) which includes a series of smaller guides, each oriented toward a particular role in the system such as a regular user, an editor, a gatekeeper, or a monitor. The on-line version of the same material is still available on EIES under ?TOPICS.

New INTERACT system functions were used for detecting if a particular user still has an active EIES account. Use of these new functions corrected the previously reported problem with deleted users still being in an exchange.

As this paper is being written, these new software features are just being put into use and so it is too early to tell what new problems and/or emerging needs may guide the evolution of the TOPICS system still further. Whereas the LEGITECH system quickly ran into limits, the general foundation on which the TOPICS system was developed appears to be sustaining a much longer evolutionary life. User needs may evolve to the point where the existing framework is too limiting and a total rewrite will need to be considered, but that point has not been reached yet. However, is interesting to note that mediating procedures are already being added on top of the TOPICS system by the Politechs networks to tailor it to their special needs which cannot be satisfied within the framework of the TOPICS system per se, as described in the next section. This indicates that the design is nearing its peak of maturity and development.

SPECIAL TAILORINGS OF TOPICS III

Figure F: TAILORING THE TOPICS SYSTEM:
OPTIONS, DEFINED COMMANDS, SPECIAL INTERACT PROCEDURES

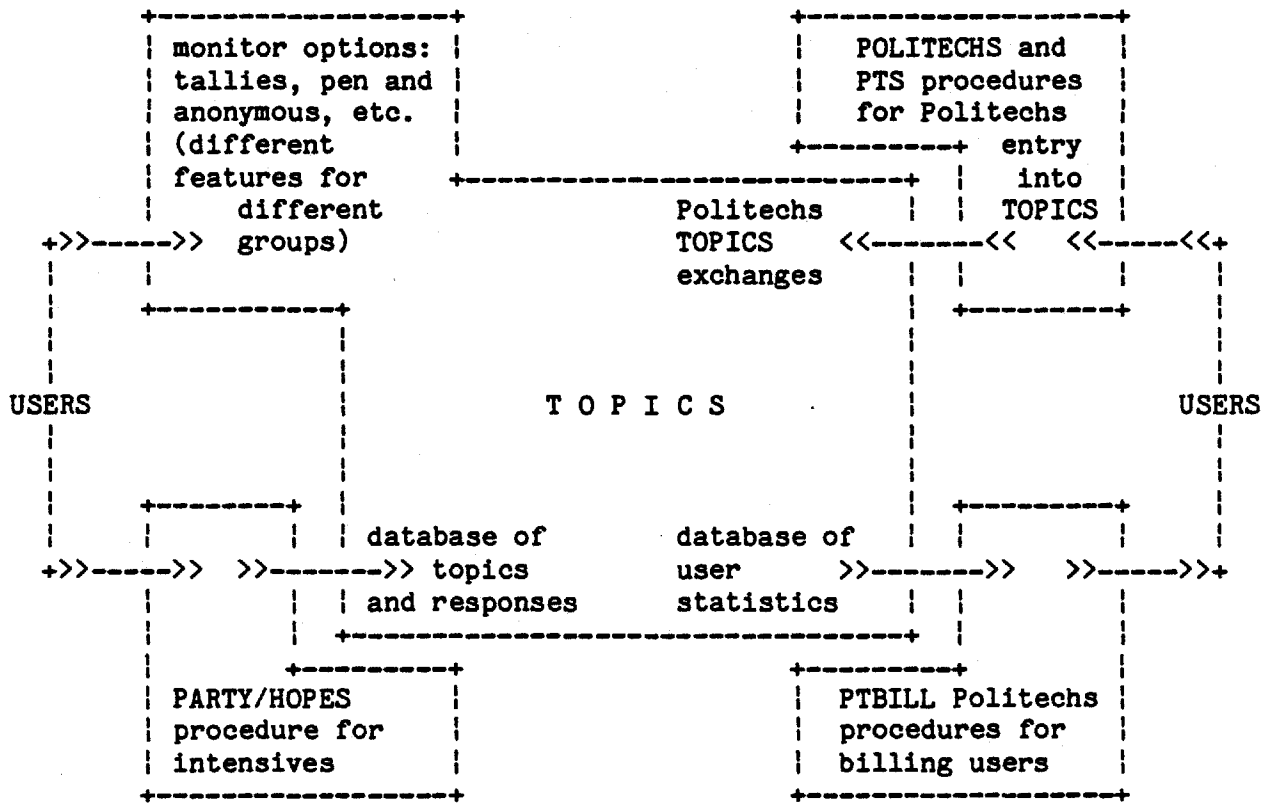


Figure F shows four tailorings of the TOPICS system. The monitor options and the PARTY and HOPES procedures have been discussed in previous sections. During the TOPICS III cycle of modifications of the system, the Politechs networks expressed a need for a procedure to bill network members as previously described. At first, the possibility of building such procedures into the TOPICS system itself was considered. However, the design they wished to incorporate was not sufficiently general to warrant direct inclusion in the general framework of the TOPICS system, and it was therefore decided that the only necessary change to the TOPICS system itself was to make it count responses received, as discussed above. Then the Politechs programmer developed a PTBILL (Politechs Billing) procedure that reads the counts of responses accumulated by TOPICS and computes and prints bills to send to Politechs members. This PTBILL procedure is not a part of the TOPICS system, but is rather a stand-alone procedure which mediates between the TOPICS system and the person doing the billing. The Politechs networks are free to evolve and modify this billing procedure without having to consider the TOPICS system which supplies it with raw data.

Up to the TOPICS III cycle of modifications, the Politechs networks had been using the +POLITECHS command as the equivalent of the +TOPICS command. During

this cycle, they also decided that they wanted to create a special Politechs "front end" to provide users with many options, including going into just one of the Politechs Exchanges, getting waiting items in all Politechs exchanges, getting waiting EIES messages, going into EIES, logging off, or combinations of all of these. To accomplish this, they developed a new +POLITECHS procedure more complex than the previous simple defined command. The same procedure is also available via the briefer +PTS command. Thus, they can create a sophisticated, tailored front end for the TOPICS system without having to concern themselves with the internal structure of that system. This front-end procedure mediates between the user and TOPICS and translates the special +POLITECHS user commands and abbreviations into the language acceptable to TOPICS and EIES.

SUMMARY AND CONCLUSION

We began by describing a model for the design of groupware systems and the software systems nested within them. This model is based on the assumption that the design must be responsive to the perceived needs of groups using the system. These needs define the processes appropriate for effective group work which in turn govern the selection of procedures to support those processes. Finally, the procedures chosen govern the selection or development of software to support them. The model includes two critical feedback loops. The first involves feedback of the design prior to implementation for any last-minute adjustments. The second involves feedback to the design team based on the group's experience with the system. These two feedback loops, constantly driven by the emerging needs of groups, create on ongoing, dynamic process of design.

We continued by amplifying this model to show the balances and trade-offs between the extremes of a general system on one hand and a responsive and tailored system on the other. We also discussed a second trade-off between working within the current framework and completely restructuring the system. The concept of an "evolutionary window" was introduced, during which there is an opportunity for the system to evolve, but after which the very fabric of communication which keeps the group vital is lost as the system becomes dysfunctional and eventually collapses. The changing costs and benefits of the current/original system and the evolved system during this window were presented.

The discussion of the evolutionary window then focused on the software itself and an array of alternative strategies for making the system responsive to user needs. The array included some methods for lengthening the evolutionary window by working within the original system and extending its useful life, including modifications, tailored options, use of defined commands, and use of mediating procedures or "beings" in between the user and the original system.

Before relating the evolutionary history of the TOPICS system, a brief discussion of the flexible, powerful programming and tailoring tools available on EIES through the INTERACT programming language was presented to provide the reader with a basic foundation with which to understand some of the technical details of the TOPICS developments.

THE SUBJECTIVE ART OF GROUPWARE DESIGN

In conclusion, the authors wish to emphasize that a design team's selection of a strategy for coping with an evolving set of needs is somewhat arbitrary. There is no absolute taxonomy of situations, problems, and solutions; rather, a wide range of alternative strategies for evolving systems exist, each with its own costs and benefits. No single answer is the right one. Figure B, showing the trade-offs involved in such evolution, presents a rather unpredictable and dynamic path of the evolution of a system. As long as the design team remains sensitive to the changing needs of the groups using the system, the extremes of an overly general or overly specific system, and the

problems of staying within a dying framework too long or moving too quickly to a new design, the process will remain alive and vital.

More experience with computer-based human communication may yield a more well-organized and less subjective methodology. Already there is emerging consensus about some of the more mechanical and simple aspects of user interfaces. However, there is still the legendary disagreement among users as to which kind of editor is best. The subjective element in preferences for different editors frustrates any attempt to develop a concise, logical theory of editors.

As a consequence, the authors believe that any mature theory of the evolution of software must first be nested within a larger theory of the evolution of groupware, which is in fact, the evolution of social systems. Any such theory will have to come to terms with two broad themes. First, it must address the subjective preferences, norms, and values which drive the evolution of social systems. Second, it must address the rapidly emerging diversity of cultural and social forms and the current epistemological shifts to more relativistic approaches to thinking about ourselves, our societies, and the conduct of life on this planet.

It is the authors' hope that this account of the evolution of the TOPICS software system may open up consideration of the dynamic forces at work in software and groupware evolution, as well as provide an initial conceptual framework for such consideration.

REFERENCES

de Bivort, Lawrence H.
1980 EVOLUTIONARY DEVELOPMENT: THE PRINCIPLES AND PURPOSES OF EVOLUTIONARY MANAGEMENT. Unpublished.

Hiltz, S. Roxanne, and Turoff, Murray
1978 THE NETWORK NATION. Addison-Wesley, Reading, Massachusetts.

Johnson-Lenz, Peter and Trudy
1979 "Speculations on Facilitating Network Structures: Balancing Social Coherence and Individual Liberty." Unpublished.
1980a "Legitech/EIES: Information Exchange and Collective Knowledge Building Among State Legislative Researchers." In Henderson, Madeline M. and Marcia J. MacNaughton (eds.), ELECTRONIC COMMUNICATION: TECHNOLOGY AND IMPACTS. Westview Press, AAAS Symposium Series, Boulder, Colorado.
1980b "Groupware: The Emerging Art of Orchestrating Collective Intelligence." Paper presented at First Global Conference on the Future, Toronto, Ontario.
1980c "Consider the Groupware: Design and Group Process Impacts On Communication in the Electronic Medium." Unpublished.
1980d "Guide to the TOPICS System." Forthcoming Research Report, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark, New Jersey.

Toffler, Alvin
1980 THE THIRD WAVE. William Morrow & Co., New York, New York.