ABSTRACT
With the growing use of electronic mail come challenges in how to aid users in handling ever-increasing volumes of email. Firsthand experience as well as the systematic study of users’ email patterns suggest that a large fraction of messages are parts of larger transactions. We have implemented an approach toward structuring messages that is intended to help users carry out some of these transactions. Our approach has been to make message structure both general-purpose and optional. Two studies support the applicability and the acceptance of the messaging model. The system, which provides capabilities that are largely absent from conventional email systems, has been deployed in an internal trial.

Keywords
email, structured messaging, media choice

EMAIL AS TRANSACTIONS
Electronic messaging has recently experienced an increase in popularity and is now commonly employed by both businesspeople and consumers for a wide variety of tasks. Because electronic messaging systems automate the delivery and storage of information, they simplify the tasks of communicating and of keeping track of communications. While much of the literature treats messaging as the conveyance of information from one party to one or more other parties, in truth, messages are often used as parts of larger, more complicated exchanges. For example, a message that invites you to a meeting is likely to request a reply indicating your availability. The sender of that message has not only conveyed the desire for your attendance, but has also initiated a process that could continue up until the time of the meeting. During that time, you may or may not respond to the initial message. If you do respond, the sender may compile the responses from several people, decide upon a time and re-respond with the new meeting time, possibly including an additional RSVP. You may change your availability in the meantime, and the process could begin over. If you choose not to respond to any of the messages, the sender may initiate a reminder. All of these steps related and they are all done with messages.

We have used the term “transaction” to refer to exchanges of this sort: a series of activities between two or more parties that follows through to the completion of some planned goal. Transactions, like messages, involve interactions between people and are often asynchronous. But, they differ from the traditional messaging paradigm in two ways:

First, transactions have temporal continuity. Once initiated, transactions continue to exist until they are either finished, terminated or “timed-out”. While ongoing, a transaction always has a definable set of “next” actions, so that the transaction has some state, even when no overt activity is taking place. In workflow applications, these actions are defined by formal process rules of the organization. In informal transactions, there are similar actions, although they are often implicit. [3]

The second aspect of transactions that differs from traditional messaging is that the potential end-states are pre-defined, as are the processes for reaching them. The financial transaction, for example, of withdrawing money from a savings account does not ever lead to winning a large sum of money, because that end-state is not part of the defined transaction. While in informal transactions, the end-states may be less clearly stated, the possibilities are bounded [11]. One way to view this aspect of transactions is in terms of different kinds of control. At the level of individual communications, control in a transaction passes back and forth between participants in that any participant can usually decide whether to respond, when to respond, and how to respond. But, in terms of the transaction as a whole, the initiator often maintains a large degree of control over the transaction. By having decided the potential end-states, the initiator has put bounds on the kinds of actions that can take place within the transaction.

We have used the term “transaction” in favor of related linguistic terms such as “conversation” or “adjacency pair.”
or Clark’s [3] “joint project” because it carries fewer theoretical expectations. Our intent has not been to match the complexity of natural language, but instead, to use linguistic concepts as a guide for designing system features. As a result, our prototype Action Email system has not dealt with some frequent linguistic complications; e.g., there is no embedding of requests and, since exchanges are limited to a single question and answer, “local structure” is emphasized [2].

Transactions and Conventional Messaging

Anecdotally, users of email and voice messaging systems often engage in transactions. However, there has been little empirical investigation on the use of current electronic messaging systems as vehicles for transactions by either businesspeople or consumers. One loose metric for transactional messaging is the frequency of replying to email. Again, little has been published regarding the frequency or reasons for messaging users replying to messages they receive, though Sproull and Kiesler [12] report that business people spend considerable time typing replies to messages. We have analyzed the responses of electronic mail users in various ways. For example, Milewski, Millen and Smith [10] have analyzed the outgoing mail of 10 electronic mail users in a technical research setting and have found that more than one-third of those messages sent out contained replies, either by virtue of their content or because there was a “Re:” in the subject line. These characters, in many mailers, indicate that the outgoing message is a reply to a received message, and therefore, may be part of a larger interaction. An additional group of messages were judged as “requests” for information, so that, in total, more than half of email could be construed as transaction-based.

Empirical data on transactions in voicemail is even less common. It is difficult, in most voicemail systems, to automatically detect replies, and content analysis of messages is difficult. Anecdotally however, many voicemail messages contain a simplistic request for a callback.

In summary, a significant proportion of messages are associated with a transaction between users. While these number of transactions may not currently be overwhelming, they are performed within systems with minimal support for transactions. A system that automates and facilitates everyday transactions could act as an enabler and increase the number and complexity of transactions.

Transaction Support in Messaging Systems

In order to support transactions and make them convenient for users, a messaging system must provide several things. These include:

- **An uncomplicated way to compose transactions** -- The composition of transactions is substantially more complicated than that of creating traditional messages.

In addition to specifying the addressing and content information, transaction-composition needs to specify which responses are available to the recipient; how to process replies; how to report status as the transaction proceeds; what kinds of content changes are contingent on events and recipient actions; what to do if the recipient never responds, needs to alter their previous response, etc.. Some newer mail and workflow systems have implemented this support via complicated programming languages. For example, Microsoft Exchange™ accepts programs written in Visual Basic Script while Lotus Notes™ takes Lotus Script Programs. Because these general programming languages require some experience to use, these programs are generally written by programmers, and seldom by end users. Some systems provide pre-written, application-specific templates. These have generally not received great use, possibly because such a large number of task-specific templates is needed for a general system.

- **A simple way for respondents to reply to transaction-based messages** -- Automatic reply and forwarding facilities have been common for a long while in most electronic mail and some voice messaging systems. However, since the reply content is traditionally free-text, the originator of the transaction has no control over the kind of response, or the endstates of the transaction. For example, while the originator might expect a yes/no response, s/he might receive a lengthy explanation. A useful alternative for both text and voice messaging is “buttons” and menus and other response-structuring techniques [7]. Such message structure often simplifies both the task of responding for the recipient and the automated processing of replies for the sender. This is important since a successful collaborative tool must provide benefits to all parties [6]. Systems based on structured responding can have the disadvantage of being inflexible in the face of contextual changes [13], but users can benefit from the use of “semistructure” and careful consideration of the tradeoffs involved in structuring [8, 15].

- **Management and reporting of the status of multi-step transactions** -- Some newer electronic mailers have begun displaying “threads” that relate messages and replies in such a way that emphasizes their relationships. Usually this is done simply by bundling messages and replies that have identical “subject” lines, although Lotus Notes Mail and newer versions of Netscape mail actually maintain links between messages. But, there is generally no compilation of replies, and certainly no way for the originator to know who has not responded. To do more than simple threading, the messaging system needs to be able to automatically process responses based on their content.

- **Dynamic message content** -- Since the state of a transaction can depend both on events, and on how
members of the transaction respond, there is often a need to have the content of individual messages change between the time it is sent and when it is read by recipients. Consider the transaction of enlisting speakers at monthly meetings. The sender could compose a message asking for volunteers to speak at any of the next five meetings. Recipients reading the message would see five choices until someone signs up for one. Subsequent recipients would see the remaining four choices, and so on until all slots are filled. Recipients would see different message content, depending on when the message is read. Traditional messaging systems provide no support for dynamic content.

In general, traditional messaging systems have implemented a limited communications model and, while there is growing support for transactions, there is clearly room for better support. Indeed, two key characteristics of traditional electronic messaging are antithetical to transaction support. First, messages are traditionally treated as independent objects; there is no simple way of bundling together messages that are part of the same transaction. Secondly, once a message is sent, there is traditionally little or no control over its content, in the event, for example, that the sender wishes to change it, or if some event has occurred that makes the existing content obsolete.

The following sections describe an Action Email system that explores the feasibility supporting transactions more fully.

**ACTION EMAIL SYSTEM**

Action Email is a prototype messaging system what we have created to explore the feasibility and usefulness of supporting transactions in messaging. A few constraints have guided the design and architecture of the system. The system must be standards-based: no proprietary client software. The user interface must be very lean: we would sacrifice generality of solution for ease of learning. There must be benefits to recipients as well as senders. Finally, it would be application-agnostic: there were no a priori assumptions about the topic or content of the transactions.

**Design of the System**

The Action Email system comprises three main portions.

1. **Message database** – contains message contents, list of recipients, and replies to date.
2. **Web server** – provides the user interface for composing and replying to messages as well as checking transaction status.
3. **Email transport** – for notification of new messages and replies.

In order to send an Action Email message, the sender accesses a web page that presents the message composition screen (Figure 1). The sender types a body for the message, enumerates the anticipated set of possible responses, addresses and sends the message. The web server then stores the message in the message database and notifies recipients of the new message via email.

**Message Composition**

Action Email is not intended to replace traditional email systems; instead it is an adjunct to be used at the sender’s discretion. Users are free to use the email client of their choice, however certain clients that are configurable (e.g., Microsoft Internet Mail) can display a button that invokes the Action Email capability, making the tool ready-to-hand. Figure 1 shows the main features of the Action Email message composition screen. There is a place for the sender to specify a message subject, just as in conventional email. This subject will be used when recipients are notified of a new Action Email message. Below the subject is a large area for typing the body of the message. This area is labeled Question to emphasize that the transaction is being structured as a question/answer dialogue. Next to the question is a set of items labeled Possible Responses. By default, these are filled in with Yes, No, and Other, but the sender may type in any set of desired responses. If the sender includes the special response called “Other,” recipients will be able to fill in their own response if none of the specified responses are judged to be adequate. Below this list is a selection: Allow One Response or Allow Any Number. This indicates whether recipients will be forced to choose one response from the list (as for a Yes/No question, for example) or whether they will be able to choose any number of responses (as in the Figure 1 example). Finally, there is an option to allow extra space for recipients to type freeform comments.

**Pointer Message**

When the user completes the message and presses Send, the message is stored in the message database. The Action Email server then sends a pointer message to each recipient. This is a short email message that informs the recipient of a new Action Email message, identified by
sender and subject. Also included in the pointer message is a URL to retrieve the message. Recent mailers that support “clickable” URLs make retrieval of the message particularly convenient.

**Message Reply Screen**

When the message URL is requested, the recipients see a form similar to Figure 2. The original message is displayed along with user interface elements which facilitate a quick response. Radio buttons are used for exclusive choice messages; checkboxes are used for multiple choice messages as in the example. If the sender has allowed a freeform comment area, then that is also displayed to the recipient.

![Figure 2. Message Reply Screen](image)

The recipient marks the appropriate response(s) and optionally types a comment. The recipient may elect to have a copy of the response sent to himself via regular email so that it can be filed with other related messages. If the recipient requests that message URL again, then the original message and the recipient’s response(s) are displayed, but the recipient is not allowed to “re-answer” the question.

**Dynamic Content**

The Action Email architecture has been designed to support dynamic content. While in theory, any server-based messaging system can do so, the combination of server-based messaging and structured response objects in messages is particularly useful. Since recipients store only pointer messages on their local equipment, the Action Email system can change the content of the server-resident messages based on automatic processing of the others’ structured replies. For those recipients that have not yet read their messages, the content, in effect, changes between the time the message is “delivered” and when it is read.

For an efficient implementation of dynamic content, the message storage must be centralized, as in our server-based architecture. Though computational email systems such as ATOMICMAIL [1] could render forms for recipient, the content of that form is necessarily limited to the content at the time it was sent.

Since the ability to have message content change in midstream fundamentally changes the messaging model, we have found it a challenge to integrate dynamic content into our message composition procedures in a way that is easy-to-use and yet general purpose. This is still an area for continuing research.

**USING TRANSACTIONAL MESSAGING**

Two fundamental questions needed to be addressed in order to predict the efficacy of a system such as the Action Email system: 1. would it have broad applicability? and 2. would it have broad acceptance? We conducted studies to address each question.

**Study 1: Range of application**

The outboxes were analyzed for eight email users in a communications research organization. Messages came from an assortment of commercial email systems. Outboxes were chosen over inboxes because they more clearly reflect user-to-user transactions, undiluted by what can be a large number of organizational announcements and corporate directives. While the study was limited to users with at least 100 outbox messages, the actual size of the outbox maintained or archived by our sample ranged from 100 to 1100. Since the investigation required content analysis of messages, users were permitted to edit their outboxes and omit any messages they felt were too personal for analysis. Moreover, several users maintained several systems (e.g. home-office and office); outboxes were combined across systems. Approximately one hundred outbox messages from each user were selected for analysis. Because of the editing and multiple systems, these messages were not necessarily successive.

![Figure 3. Message Status Screen](image)
Messages were manually scored on several dimensions. One dimension was whether the message was requesting a response (initiating a transaction), providing a response (continuing a transaction) or neither (e.g., an FYI message). Those messages that were judged to be part of a transaction were then scored as to whether or not the originator of the request could have anticipated the list of possible responses.

Messages for each user were rated by one of two observers. Because the judgment for several of the measures above are subjective, interobserver reliability was assessed by having both observers separately rate 30 messages for three different users. There was 93% overall agreement between observers and agreement on individual categories varied from 83% to 100%.

The results showed a high percentage of messages to be transaction-related. 55% of messages were judged to contain requests or answers (or both). This overall pattern of results was found across all users. For example, the mean proportion of requests and answers was 21% and 28% respectively, while the standard deviations were both 6%. Thus more than half of these users’ outgoing messages were part of email transactions.

Of these transactional messages, 44% were judged to fit into our structure; i.e., the list of possible responses could be anticipated by the sender. Thus about a quarter of all outgoing messages (44% of 55%) were judged to be of a type that would fit into our transaction structure. By far the prevalent structure for those messages was “allow one response” from the list of possible responses rather than “allow any number” (90% vs. 10%). This may be surprising since so many workplace tasks would seem to benefit from multiple-choice structures. (e.g., “during which of the following times are you free for a meeting?...”). Indeed, the first version of Action Email only supported “allow one answer”; the option to allow any number of answers was added in response to the request of trial users. Since current email systems provide no automated processing or organization of replies, and since the sender may find it quicker to construct a request for free-text answer, it may be that any advantage to the “allow any number” type is reduced without tools that exploit the structure.

Study 2: Social Acceptance
Study 2 explored users’ perceptions of the richness of structured response objects as communication media. Media richness is characterized as "rich" or "lean" based on the immediacy of the feedback, the ease of conveying a personal focus through the message, the multiple cues conveyed through the message, and the ability to include natural language in the message. Daft and Lengel [4] ranked different media in ascending order with face-to-face communication, the richest medium, at the top of the list.

Their theory suggests that "good" managers would choose a rich medium for messages or situations that are highly equivocal (interpretation of the task is necessary) and a leaner medium for messages and situations that are not equivocal. There are several other theories of media choice including critical mass theory [9] and social influence models [5].

Regardless of theory, different factors or dimensions affect an individual's media choice. Personal, important or ambiguous situations may lead an individual to choose one medium over another. Personal messages are ones that involve private, delicate, or sensitive information which can be associated with strong feelings or emotions [14]. Important messages involve information that is of special value to either the sender or the recipient of the message. Importance is perhaps the most difficult of these dimensions to define because perceptions of importance greatly differ from individual to individual, and perhaps from sender to recipient. Ambiguous messages involve ideas which are vague, confusing and have the potential of being interpreted in more than one way.

We constructed a study in which 125 respondents answered each of eight questions with their choice of a structured response (e.g., Yes/No) or freeform text. After all eight questions had been answered, the respondents rated the ambiguity, importance, and personal nature of each question on a five-point scale. Also, various pieces of demographic data were collected about the respondents.

We constructed a logistic regression model to predict the media choice (structured response vs. freeform text) as a function of the ratings and demographic data. Overall, 83% of questions were answered with a structured response. However, there were factors which modulated the choice by question and respondent. The perceived ambiguity of the question was a reliable predictor (p<.01) with the strongest effect of the three per-question dimensions. The personal nature of the question was also a reliable predictor (p<.05) with a reduced strength of effect; the perceived importance of the question was not a factor in media choice. Another question-related factor found to influence media choice was the wording of structured responses (structured Yes/No responses were chosen more often vs. freeform text than structured Agree/Disagree responses). Per-respondent factors that influenced choice were Internet experience, amount of time spent surfing the web, and sex of respondent (those with more Internet experience, less time per week spent surfing, and males all had an increased tendency to choose freeform text).

Thus structured responses are considered to be broadly socially acceptable, but special care should be given when posing potentially ambiguous or personal questions.
CONCLUSIONS

Structured messaging and the model of email as transactions may provide many benefits to users. Roughly a quarter of outgoing messages analyzed from a set of users could have been handled in the simple transactional structure of a single question accompanied by a list of possible responses. We have built a system to support this transactional model and studied its social acceptability. Action Email gives senders a general way to structure transactional messages, but still permits flexibility for both sender and respondents. Since users are not required to install or use any particular client software, our system could have immediate benefits to a user community. We have used the system internally, continually refining the user interface to provide maximum benefit to both senders and recipients.

Dynamic content provides a powerful new capability, accompanied by the double-edged sword of a new messaging model. Further research in how to communicate this model to users could substantially enhance group communication through transactional messaging.

ACKNOWLEDGMENTS

The authors gratefully acknowledge Beatrice Camino and David Millen for their significant contributions to the design and analysis of Action Email. Bruce Hanson’s ingenuity led directly to the original system design. Greg Blonder provided a steady stream of valuable feedback and encouragement. Finally, anonymous GROUP’97 reviewers substantially increased the quality of our submission by helping place it in the context of related work.

REFERENCES


