MMS: An Electronic Communication System For Emergency Management Organizations

Henning Boje Andersen
Henrik Garde
Verner Andersen

Risø National Laboratory
Cognitive Systems Group
4000 Roskilde, Denmark

Abstract

The paper outlines the main features of an electronic communication system, MMS, designed as a dedicated coordination tool supporting Emergency Management efforts. The design of the MMS, which has been motivated by analysis of communication and coordination problems observed during EM actions, provides users with an overview of the follow-up history of any EM message and of unanswered requests and alarms.

Background

The MMS (Message Management System) is a system which is designed to support electronic communication of Emergency Management (EM) organisations. While a precursor of the system was initiated in a project led by Risø National Laboratory and sponsored by the European Community under the Esprit Programme (Esprit Project 2322 ISEM: Information Technological Support for Emergency Management; 1989-1992), the present version of the system, is the result of efforts mainly within a project of the Eureka Programme: MEMBrain (Decision Support Integration Platform for Major Emergency Management; 1993-98).

The system has been designed, first, as a general-purpose electronic mail system, and second, as an e-mail system with extended features especially dedicated for use by EM organisations and other types of organisations and companies operating in *safety critical domains*. In the following we describe some of the major features that distinguish MMS from most other e-mail systems as well as the motivation for their introduction. But we leave out features that can be expected to be present in any modern electronic mail system.

Decision makers involved in EM, including their aides and team leaders, have several needs in regard to support of communication and procedural information: Put in general terms, they need *timely and perspicuous* information. Thus, they need to have information about the status and development of "physical" events (past, current, and forecasted), about actions and procedures that are dictated, or perhaps suggested, by preparedness plans, about the status of current EM efforts and relevant EM resources, and about the reactions by the public and the press. While it is obviously important to provide information that helps to establish decision makers' awareness of the "objective" emergency situation, in the sense of the physical conditions that are endangering lives, environment and property, the importance of providing information about *the flow and status of EM-coordination* is sometimes overlooked. However, a communication system dedicated to supporting EM-efforts should, we believe, aid in conveying information about EM agents' intentions and plans, commands, acknowledgements of commands as well as their absence, and, in general, a picture of "who is doing what where and when".

It is this requirement - supporting the provision of an *overview of EM-coordination* - more than anything else which has prompted the design of the specific features of the MMS, in particular the division of messages into *types* such as REQUESTs and OKs and their linking and graphic 2D-display. However, before we sketch how these and other features work in the MMS, let us describe in slightly more detail some of the problems that motivated the design of the system.

Some problems concerning communication and procedural activities associated with EM efforts

Based on interviews with EM experts and surveys of reports of drills and real accidents or disasters, we have concluded that a number of problems seem to arise in connection with establishing an overview of how commands and requests are being followed up.

There are two sides to this. First there are the needs as they are defined from a sender's point of view, for instance, a busy decision maker and his aides who have dispatched a number of commands, requests, and alarm messages to a possibly large number of units need to know who has acknowledged and carried out the actions requested and who has not (yet). Conversely, if we define the needs from the receiver's point of view, we may think of a busy EM organisational unit, having received a possible large number of messages which have not yet been dealt with - in this case the unit needs to know what are the unacknowledged commands or requests to this unit waiting for them to react or respond to.

Besides the problem of keeping track of commands and responses to these, there are a number of other communication and coordination problems which to some extent can be lightened by proper systems support. But space does not permit us to go further into these matters in this paper. Here we merely give a list of such problems:

- A given sequence of messages from one organisational unit to another may be misunderstood by the receivers. This may happen when an initial message has not been opened or has been overlooked; then, when subsequent messages arrive that allude implicitly to the lost message, they are liable to be misinterpreted.
- A brief declarative message meant as a command by the sender can be interpreted mistakenly by the receiver as a piece of information.
- Decision makers and personnel at operating centres may become overwhelmed by "book-keeping" when they try to keep track of responses to their commands (requests) issued - i.e. commands not yet responded to, commands responded to but not yet completed and commands already completed.
- The meaning of a given message will sometimes be misunderstood if the message is not seen in the context of other messages to which it is related. The personnel of an operating centre who typically must deal with a great number of messages may lose track of the context of a given incoming message unless they are able quickly to tie it to the relevant preceding messages.
- Finally, while decision makers and their aides are usually familiar with the details of the relevant preparedness plans, there are often problems in communicating quickly alarms and requests to all relevant units as well as in determining if receivers have actually received, or acknowledged receipt of, such messages.

Specific MMS features

The following features have been implemented in the MMS design in order to overcome or alleviate problems in keeping track of coordination and communication:

- Initial messages are distinguished into REQUESTs and NOTEs. REQUESTs are messages that require a reply from receivers; NOTEs are all other initial messages.
- When receiving a REQUEST, a user will be prompted to send a reply. There are three types of replies he may send: an OK! message that contains no text; an ANSWER message that contains any free text composed by the user; and another REQUEST which then in turn will prompt its receiver for a reply.
- Message linking and classification into typed messages is based on the users' own classification of messages - the "smart" features of the system are governed by rules that operate entirely on message "envelope" properties most of which result from user inputs.
- Links among messages can be seen in a graphic tree display (see fig. 3) which also reveals different types of responses - providing, for instance, a quick view of who has failed to acknowledge a command or complete an action requested.
- A reminder function can be activated when sending an urgent REQUEST. The reminder alarm will alert the sender and/or the receiver after a given interval chosen by the sender depending on any of a range of criteria to be fixed by the sender.
- A set of status attributes is constantly updated for any message so that a sender, when looking at a list of messages, may see at a glance whether a message has arrived at the receiver's mailbox, whether it has been opened by the receiver, and, when applicable, what type of reply has been returned.
- A library of *structured messages* is supplied allowing users to fill out and submit standard reports with least possible effort.
- Integration between the MMS and the preparedness plan (PP) allowing users to keep an integrated overview of commands and acknowledgements and the status of current tasks as dictated by the PP.

Illustration of some MMS features

In fig. 1 is displayed the entry window of the e-mail system as it appears when the user has

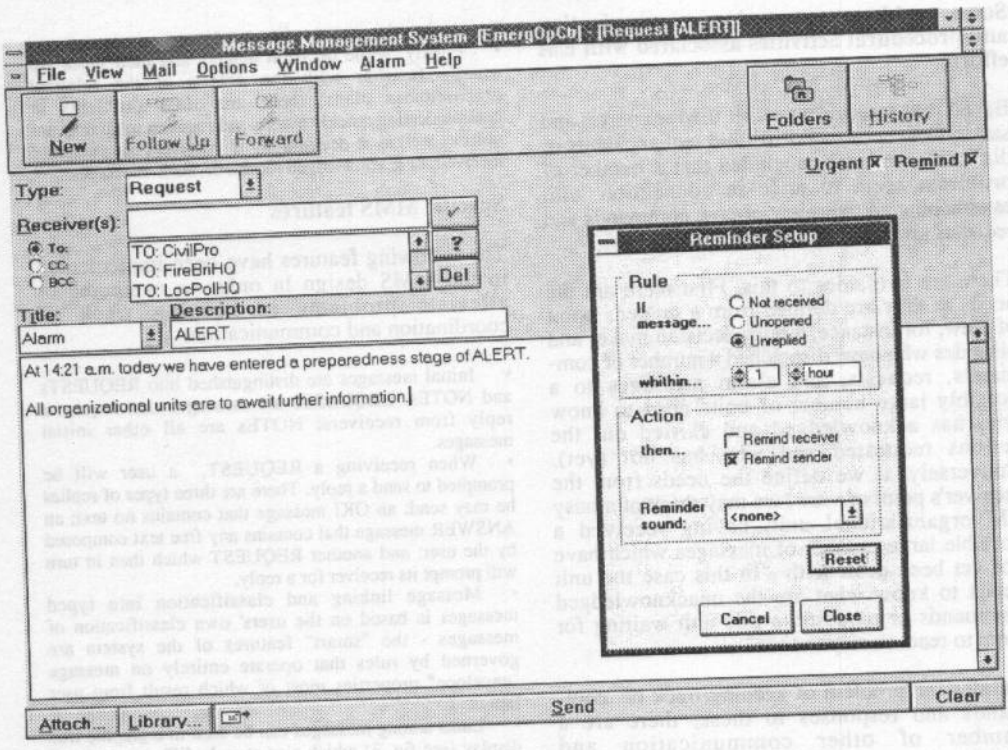

Figure 1

clicked on the *New* button - i.e., invoking the command "create new message".
The user has to mark the type as either a NOTE or REQUEST. While a NOTE is an ordinary message a REQUEST is a message that will prompt the receiver for a reply. So, a REQUEST is a message that the receiver is expected to respond to and possibly act on. We can also see in fig. 1 that in this particular case the user - Emergency Operating Centre - has started his MMS application in an alarm situation that has been set in an accompanying module carrying a preparedness plan. The specific receivers may be picked as individuals or groups from a pre-written list of possible receivers. Furthermore, the reminder function (see fig. 1) may be invoked by the user allowing him to set a reminder alarm after a selected interval in case the receiver has not opened, or has not replied to the sender's REQUEST. The alarm can be set to notify either the sender or the receiver or both of them.

In replying to an initial message the receiver of this message may open a similar window for reply by clicking the 'follow-up' button. He will now be able to view the initial message while creating his reply. He is offered three types of replies: he can either send a no-text OK! message back, or he may send an ANSWER if he needs to add some text. Finally, he may choose to respond to the REQUEST he has received by another REQUEST - the latter option is to be chosen, of course, if he wants the sender to do something or if he just wants the sender to acknowledge his message. The reason why we have included a no-text OK! type of reply is, as explained above, that decision makers need to know precisely who among a possibly very large group of receivers has not reacted to a command.

In fig. 2 we are once again back at the original sender's screen, the Emergency Operating Centre. By using the alarm option, the user may for instance jump directly to the preparedness plan module (PP), or by setting the alarm mode and specifying the kind of alarm, he may fetch

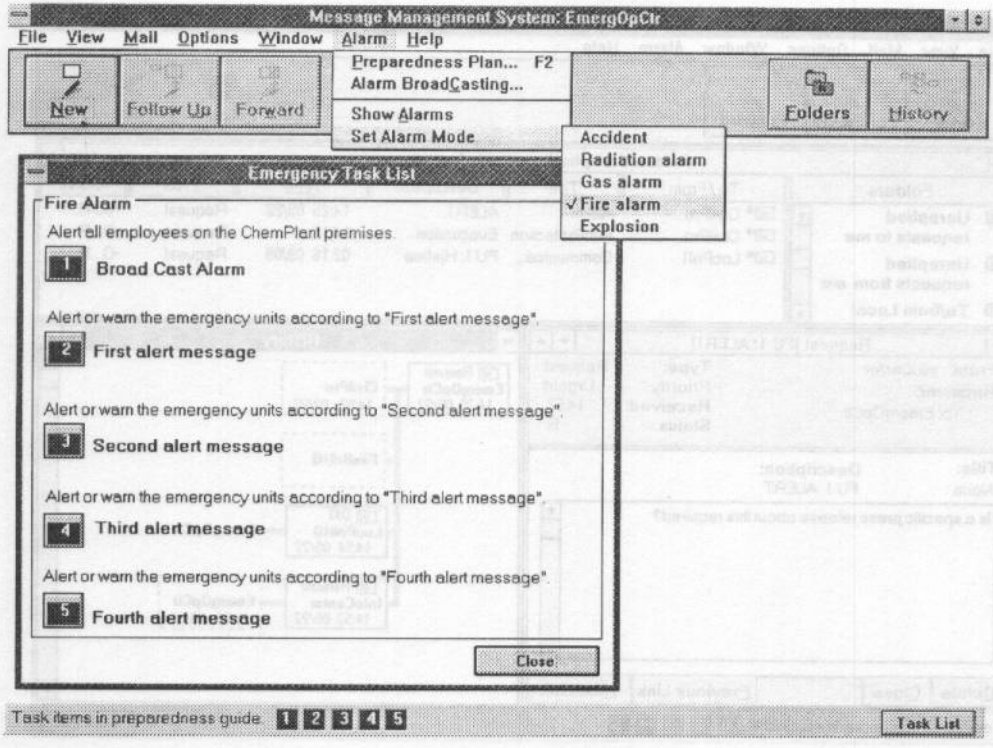

Figure 2

from the PP the specific list of actions to be carried out in the situation at hand. A task list, a generic version of which is shown in fig. 2, may either be a checklist for himself or it could be forwarded, if needed, to other participants or groups of participants in the emergency organisation.

In fig. 3 the scenario is continued and the sender has opened his folder *Unreplied requests from me*, finding in the folder his own original message sent out at 14:25. By clicking on the History button on the top tool bar, the user opens a tree-like display of the *history of the message*. The *History* function provides a picture that is automatically updated by the MMS displaying the flow of communication linked to the original message. By clicking on the boxes in the History diagram the user may make the selected messages apear in the bottom left cornor.The original message is symbolised by the box labelled "Request / EmergOpCtr / 14:25 09/22". The request has been sent to four receivers symbolised by four successor boxes. The top successor box, representing the receiver "CivilPro", is surrounded by a dotted line indicating that the receiver has not yet sent a reply; but it contains a time stamp that indicates that the receiver has opened the message at 14:57. The next box, denoting the receiver "FireBriHQ", is dotted and contains no time stamp, so our user can see that this receiver has not yet opened the original message. Then comes the receiver "LocPolHQ" who has replied by sending an OK! - this, recall, is the no-text reply option. To the right of LocPolHQ's box symbolising his OK! message the successor branch indicates to whom the OK! was sent (in this case this is superfluous information, in other cases it is not so.) Finally, the fourth receiver, the "InfoCenter", has sent a counter-request at 14:52 as is displayed here. Again, the dotted box to the right of InfoCenter's counter-REQUEST box symbolises that the receiver of this second REQUEST, EmergOpCtr, has opened the message at 15:00 and has not yet

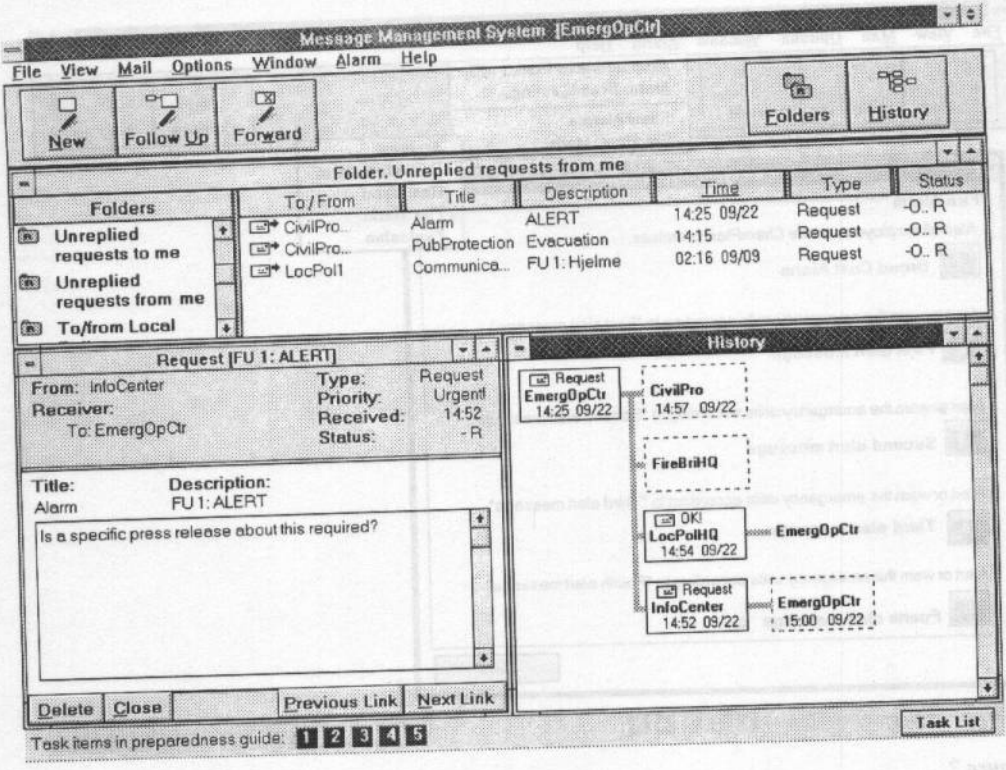

Figure 3

replied to it. Furthermore, it may be seen from this figure that after the task items have been identified, the checklist is maintained by the MMS and displayed at the bottom of all subsequent screens, indicating which tasks have been completed and which are yet to be carried out. At this stage, the five actions remain to be carried out, whereas in fig. 4 we can see that the 2nd action has been completed.

The MMS is supplied with a library of *structured messages*, each of which is configured in a "Structured Message Authoring Module". Fig. 4 gives an example of such a pre-written message form. This message form is, we imagine, a message that will be sent out by a certain chemical processing plant when an accident of a certain severity has occurred. The example chosen is the first alert message sent out. In this structured message form, every slot can be filled out automatically by the system on the basis of available data, including real time data provided from external modules. However, the slots with associated combo-boxes have to be filled out by the user. So, the *time* slot is filled out by the time stamp of the first alarm; the *date* comes from the machine clock. The *type of accident* is selected by the user from a list including an optional free text line. The *number of injured persons* is filled out by the user, and he may choose from a list, including an "unknown" option, or write free text. The slots indicating the *number of employees* and *guests* present are filled out by data from an automatic personnel recording system in use in several high-hazard industries or high security installations. Finally, a small weather station in use at the plant provides real time data for the *wind force* and the *wind direction* slots, while the user has to be the judge of the type of *precipitation* that prevails. In designing structured messages, great care has to be exercised in dividing the types of information

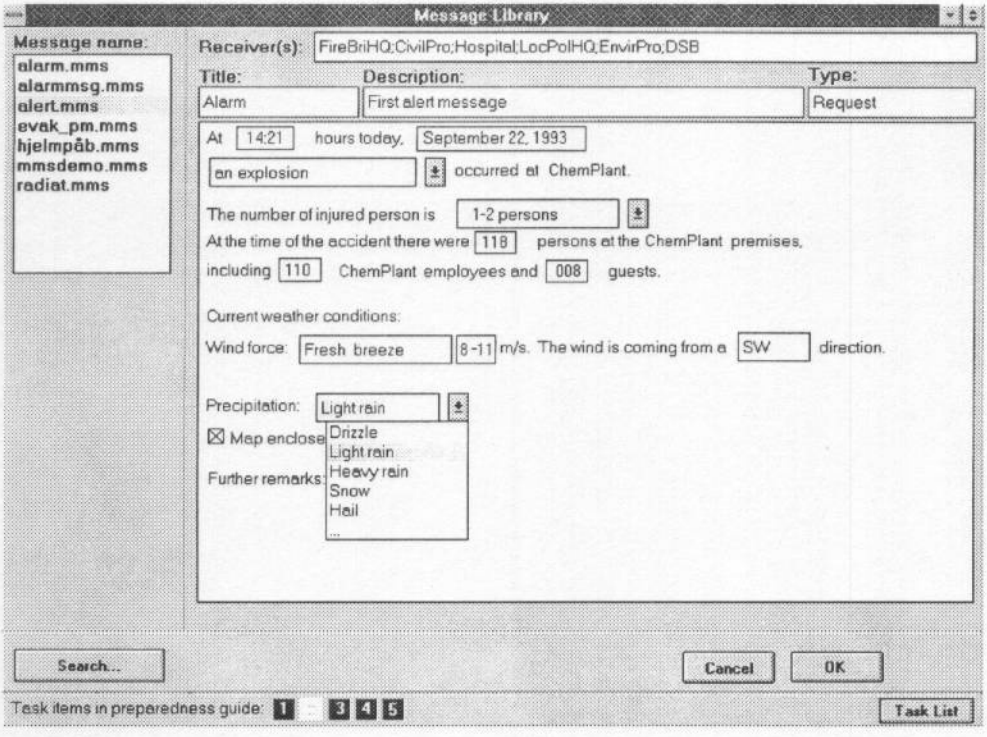

Figure 4

which sensibly and reliably can be supplied by machines and those which requires human operators to interpret and to exercise common sense.

Implementation details

The current version of MMS is a prototype that runs on top of Microsoft Mail and is implemented in Visual Basic. Current plans include implementing the MMS in an operational version for Windows NT, and porting it to the UNIX operating system. Ongoing efforts will extend the features that allow setting up structured messages and overview of work flow. Finally, as a consequence of requests expressed by field personnel, a simplified version of the MMS using *touch sensitive screens* will be implemented. The planned activities will be funded in part by Eureka project MEMBrain.