Lagarde_2002

PROPOSED COMMUNICATION SOFTWARE TO RESEARCH COMMUNICATION STRATEGIES BETWEEN MARS AND EARTH

Jean Lagarde[†] **JimOliver**[‡]
[†]Mr. Lagarde is a Mars Society volunteer living in California. jlagarde@bigfoot.com
[‡]Mr. Oliver is a Mars Society volunteer living in Colorado. jimoliver@dessica.com

INTRODUCTION

The software described in this paper is an attempt at providing an analog integrated mission database and interplanetary communication system to be used in the operation of the Mars Analog Research Stations set up by The Mars Society. The system is meant to store, organize and transmit both mission data and meta-data (data *about* the mission for the purpose of analog research). The transmittal capabilities of the system are primarily meant for communication between the Mars crew and Earth-based Mission Support, but it could be used for mission communication with other parties as well. Examples of the types of data that could be handled by the system are daily reports, technical support requests, various consumable logs, samples and other scientific databases, waypoints, data processing support requests, and crew comments (critiques) about the mission. The system was developed as a volunteer effort. It is implemented in the Zope application server, with most of the code written in the Python scripting language. The software status is considered early beta. The focus of this paper is not on the implementation details of the software, but on its functionality and its potential usefulness to the Mars analog research program.

MISSION SUPPORT COMMUNICATION IS PART OF THE ANALOG RESEARCH

Many aspects of analog research have been looked at on a day-to-day basis so far, including the constraints of the analog spacesuits, the constraints of the habitat, the use of individual rovers, the use of scientific instruments within the hab, etc. There has been an effort to maintain interactions with a mission support crew during all missions at the Flashline Mars Arctic Research Station (FMARS) and the Mars Desert Research Station (MDRS), but the quality of the analogy has been too low to reach useful conclusions on this aspect of the analog research in our opinion. The type of tool described in this paper is suggested as a way to allow to better research the benefits of various modes of communication between the Mars crew and mission support, and as a way to accumulate other useful data about the missions.

THE NEED FOR MORE STRUCTURED COMMUNICATION AND DATABASES

Besides communication to and from Mars being itself an analog activity that would take place on a real mission, this channel can also be used as the vehicle to accumulate meta-data about the analog operations, i.e. data that will help to analyze the analog operations and allow to reach

Copyright © 2003 by Jean Lagarde. Published by The Mars Society with permission

quantifiable conclusions *about* the effectiveness of tools and procedures used during these operations. Such data would not necessarily be required on a real Mars mission and so collecting it is not necessarily itself an analog operation (although it is in fact probable that such data will be collected during a real mission for the same reasons as we need to collect it during analog operations, i.e. for mission analysis), but given the fundamental goal of the analog research to evaluate various tools and procedures, such meta-data is necessary to us. The unstructured way in which meta-data has been collected at the FMARS and MDRS so far (in a mostly anecdotal manner) does not permit easy after-the-fact analysis of the operations, and is therefore an important reason to seek better communication and data organization tools.

Note that our tool is not proposed as the only way to accumulate mission data. Other approaches, such as the time lapse photography and other techniques used by Bill Clancey of NASA Ames [Clancey01], are also required.

Communications between Mars crews and mission support have so far been almost entirely done by email (with files such as data tables, photos, and audio reports as attachments), with some rare attempts at semi-real-time audio-video messaging at FMARS. The type of communications that have occurred through the email channel have included equipment support requests (which is a realistic analog communication), one-way reporting from the hab to mission support, and a few attempts at involving mission support in mission planning (mostly for extra vehicular activity (EVA) planning). The equipment support requests are perhaps a good analog of those that would occur on a real mission, but the quantity of reporting to and active involvement of mission support in the scientific exploration activities seem lacking. The emails are even less conductive to post-analysis of operations at a meta level; they have basically been archived with no analysis done (a number of groups have expressed interest in analyzing these emails, but nothing has actually been done to the authors' knowledge).

The lack of involvement of mission support into the scientific exploration activities may be due in part to the lack of scientific expertise of these volunteers, irrespective of their overall professionalism and motivation to perform, but these scientific support services would have been hard to provide by mission support crews of any experience because at any time they have had very limited data available to them from the mission to that point (i.e. their situational awareness about the mission was low). The lack of mission data provided to mission support is probably due in part to the current use of email as the communication medium. Sending email is a burden on the crew, especially sending large emails with attachments over an unreliable Internet link (e.g. Starband) so information is not sent as often as it should, and the lack of structure provided by email does not permit mission support crews to quickly find past information. An example of the high latency of the information provided to mission support is that the waypoints table (maintained at the hab in Microsoft Excel format) was sent to mission support only four times during the first MDRS season, at about one month intervals, whereas locally at the hab, the waypoint information was usually updated after every EVA.

THE NEED FOR A SUSTAINED STRUCTURED COMMUNICATIONS EFFORT

There have been sporadic uses of more advanced communication methods at FMARS, by researchers associated with the Mars Haughton Project, but these infrequent trials, although they can help test the technology, cannot answer the long-term operational research questions. To answer those questions, a given communication method should be used continuously for a whole season.

THE OVERALL VISION FOR THE PROPOSED TOOL

Although it was created primarily as a means of communication with mission support, integrating communications more tightly with information management *within* the hab is considered beneficial, and so such a goal is included in the overall vision.

The goals are as follows:

- Data should need to be entered only once. By having the "internal" mission database of the Mars crew shared with mission support, there is no need to duplicate data for the purpose of transmittal.
- At least part of the data produced by the crew (e.g. waypoints) should be continually transmitted to Earth with low latency, without requiring a lot of effort from the Mars crew.
- The tool should permit to experiment with various levels of data organization, from highly unstructured (similar to email), to very structured (like a database).
- The system should be expandable, without programming, to allow for the storage of new kinds of information as they are needed. The creation of new data tables should be simple enough to allow mission support to build them with a reasonable amount of planning.
- The tool should use an intuitive interface that allows anyone to use it with only minimal training.
- The tool should require no more, and possibly less effort than email.
- The tool should allow a distributed (i.e. not co-located) and part-time mission support crew.
- The system should allow for site-specific updating to occur, whereby various remote locations would only receive the data that they require.
- The system should simulate Mars-Earth communication delays (this may be a minor requirement depending on the type of information being transmitted).
- The transmission layer should be robust; it should allow the transmittal of large files without requiring the crew to detect failures and re-initiate transmission themselves, i.e. the system should automatically handle unreliable data channels such as Starband (this only at the application level; improvements to the TCP/IP transport layer, such as those provided by the Interplanetary Internet [Cerf01], have not been considered in this project so far).
- Historical data should be searchable as much as the data's internal structure permits, in order to facilitate post-mission analysis (at the meta level), but also as a research tool within the context of the mission.

DESCRIPTION OF THE CURRENT IMPLEMENTATION

History

This software was developed following discussions on the MDRSinfo Yahoo! Group, one of the groups created by Anna Paulson while she was leading the effort to prepare for the first MDRS season. During the analysis of the FMARS 2001 field season, it was decided that a system should be developed that would allow for storage, retrieval and searching of habitat communications.

Mission support had tried several methods of storing the emails from the habitat, but none proved satisfactory. Clearly, a more structured way of recording information was needed. Although this system was not the direct result of this analysis, the lessons learned from 2001 were used as a guide to the development process. A tentative concept emerged from these discussions, and for the purpose of having something ready by season start, Jean Lagarde put together a very simple prototype implemented in Zope[1] in early December 2001. Later in December and January, Jim Oliver made some important improvements to the prototype (more accurately, he preserved the overall vision but almost rewrote the whole application from scratch, with major new features). Later, Jean Lagarde provided more incremental additions to the software, both adding new features, and improving the underlying code (using existing Python[2] libraries, streamlining the code, etc.). This was a successful example of cooperative software development by two persons who had never met. The software was hosted on an interim server for potential users to try in late January, and was moved to the Chapter's web site provided by Cyberteams in late February. It is still up and running there. No changes to the software were made since March 2002, in part because there did not seem to be enough commitment by others within the analog project to try the tool at the time, and because the authors were busy with other projects. The Mars Society of Canada installed the software on their own server in June 2002, but have not begun their use of it at the time of this writing.

For a while the discussions on MDRSinfo included cooperation with Noel Carrascal, who was working on his own prototype proposal for a similar tool, this one implemented in Java. The MDRSinfo group offered to test his tool as well, and Randall Severy from Cyberteams offered to install his prototype on the Cyberteams server, but the group stopped hearing about Noel at about that time.

Tool Walkthrough

The tool is accessed through any recent graphical Web browser. There are too many screens to show them all in this paper, and the intent herein is to provide a general idea, rather than a user manual. Feel free to communicate with the authors to get first-hand access to the tool and help files. The main menu of the tool, shown in Figure 1, is shown on every pages in the web browser, but has been removed from some of the figures below to save space (all the figures are grouped at the end of the paper).

The tool allows to define and use tables and forms, such as the ones in Figure 2 and Figure 3. The example in the figures reproduces one of the daily reports submitted by the MDRS crew. The two figures illustrate that more than one form can be created for the same table. The form in Figure 2 is a summary form that includes only a few fields and allows to quickly browse through the records. Any form has two optional configurable links, “V” and “E”, that can point to a different view or edit form for the record, respectively. Figure 3 shows the form found by following the “E” link in this example. The figures illustrate the overall database look, the variety of fields offered (simple text fields, large text fields, selection lists, embedded files (the reports), and embedded images), the availability of multiple embedded sub-records (the Report, File, and Image subtables), the hyperlinks provided to embedded files, and the thumbnails that were automatically generated for the images. Working backwards through the design steps, Figure 4 shows how the layout of the Figure 3 form was defined. Figure 5 shows how the table itself was defined. In the table definition form, there are some “List” data types. These types can be defined by the users as required, as shown in Figure 6, and can allow either single or multiple selections to be made.

Using the “people” list type defined in Figure 6 to work our way forward through the process, the list type was used in Figure 5 as the type of the field “SubmittedBy” (the same list type could be reused to define other fields in other tables), then in Figure 4, the “SubmittedBy” field was included in the “Edit Log Book” form, and its position was defined as the second item on the first row (by default, because its position values were left “0”). Finally, in Figure 3, we see that the “SubmittedBy” field is indeed the second field on the first row of the form, and because it is a list type, it is presented as a selection list. Note that in daily use, only the forms in Figures 2 and 3 would be used by the crews. The other forms, in Figures 4 to 6, are used to define the tables and forms, and are more rarely used; they can, nonetheless, be used by the crew during their mission if they want to define a new table or change the layout of a form, for example.

On top of the form and table edit forms there is a “station updates” area. These allow to determine if and how changes are propagated from the originating station (e.g. on Mars), to one or more other station (e.g. Mission support). The three possible types of updates are:

- Scheduled: When a station is marked as receiving one or more scheduled updates, it will receive these updates at a fixed period defined by the users elsewhere within the interface.
- On-demand: When a station is marked as receiving one or more on-demand updates, these updates will be sent only when a user explicitly commands the update in the “update stations” screen (Figure 7). This provides users with complete control of when updates are sent.
- Automatic: When a station is marked as receiving automatic updates, modified or new records will be transmitted as soon as the modification or addition is made. If many records are edited in a row, then many small updates will be sent to the receiving station at the end of each edit, rather than collecting the changes for a single transmittal later on, as is done for the scheduled and on-demand update types.

Note that even form definitions are transmitted between stations using updates (for a station to use data from another station, it helps – although it is not necessary – to also get the forms that were defined for the table as well). Similarly, user-defined list-type enumerations are also transmitted between stations through updates. When a table has “(none)” selected for all three update types (as in the example), then the data in that table will never be transmitted (which could be useful for some purely internal data that is only of local use).

Going back to the main menu shown in Figure 1, the three left-most columns in the main menu deal with defining and using tables, list types, and forms. The “update stations” link leads to a page, shown in Figure 7, where stations that are marked for “on-demand” update of one or more form or table can be manually sent their updates, if any. That page provides an estimate of the amount of data that will be transmitted (data is actually sent compressed, but the estimate currently shows uncompressed size, and is therefore quite conservative). “Incremental update” is the normal way to command a transmittal of data that has changed since the last update of that receiving station, and “full update” is used to resend all data marked for transmittal to that receiving station, whether it has been previously sent or not. The intent of that last option is as a way to resynchronize the two databases if the need ever arises, but the size of data to be transmitted in a full update rapidly grows to a rather large number, so that option as implemented is currently of little value.

The “scan wormhole” menu option brings the page (Figure 8) that looks for updates from other stations and sends scheduled and automatic updates. That page currently uses the auto reload feature of HTML as a way to automatically check for updates to receive or send on a regular basis, and so the intent is for that page to be permanently selected in a browser window (which could be minimized out of sight). A better implementation of this periodic look for updates ought to be implemented using Python threads.

The software does provide a lot of flexibility in setting up either unstructured or very structured data. On the one extreme, the tool could be set up with a single table with mail-header type fields, a single text field and multiple file field, in order to duplicate the functionality of emails with attachments. On the other extreme, a multitude of specialized tables could be created, for any given task or information type, each with a very rigid field structure.

With the proposed system, the crew rarely explicitly sends messages to mission support. Rather, the mission database (which is also the communication system) is configured with tables that correspond to types of mission data or tasks, and some of these tables are pre-configured to be automatically kept in synchronization with one or more clone databases on Earth. Communication to specific individuals is achieved through an automatic email notification system, which is also configurable.

For example, rather than “sending” a maintenance request, the crew could enter or update a problem report in the mission database. The database is set up to automatically propagate that update or addition to the clone database on Earth, and to send a notification email to one or more members of the engineering support team. The notification email would contain a hyperlink to the specific database record that was updated or added. In response, the Earth engineer would update the same problem report record with their proposed solution (which provides much better traceability than sending a separate reply email) and that record update is now propagated the other way to synchronize the Mars mission database.

In any given transmission, the system does not transmit the whole table, but only the record(s) that changed. The current system, however, always retransmit files and images attached to a transmitted record, even when these large chunks of data have not changed. It is an obvious required improvement to retransmit these files only when they changed, and even possibly to transfer these changes more economically using, e.g., rsync rather than FTP.

Figure 9 illustrates at a very top level the configuration of a complete multiple station system. All stations are FTP and Email clients, with data transferred over FTP, and small automated emails used to coordinate the communications. The “wormhole” is a central FTP and Email server that does not permanently store any data, but through which all communications are buffered.

Key Problems

When filling out text fields in one of the tables, our goal of having the crew enter data only once is achieved, i.e. they enter it both in their own mission database and queue the information for transmittal at the same time. However, when transmitting files or images (as part of table records) the files must be "uploaded" to our software from somewhere else in the local file system, and so there is an unnecessary duplication. Ultimately, we would like to leave these files where they are in the hab's filesystem, and to possibly store them in the same filesystem hierarchy at the receiving end, but even conceptually this may not be entirely straightforward.

There is presently a protection for records against update by any other than the originating station. This solves the usual database problem of handling multiple concurrent updates to the same record (i.e. we simply do not allow the situation to occur), but this is not satisfactory for those tables where parts should be filled by one station, and other parts by another (e.g. Problem reports requiring an answer from mission support), and probably other similar usage patterns.

There is no method currently to search through the old data to retrieve specific information. While this is very important, we wanted to get the general data structure finished and tested before adding the search capability. This way we would not have to re-write the search engine if a change was needed in the system's basic structure.

There are many other lesser issues that need to be resolved. As stated in the introduction, this software can be considered at the early beta stage at most. At this point we need to do some field trials with real users to help determine which of these features are most needed (and if there are any we had not thought of yet.)

Fringe Benefit: Automatic Generation Of Public Reports For The Mars Society Web Site

Although this was not the main reason for the development of this system, note that the structured nature of this system would allow to automatically generate the daily report pages for the public web site, requiring a lot less volunteer manpower to maintain the pages, and providing lower latency from transmittal from the hab to posting on the web. The main problem with an automatic system would the the need to review the information before posting to ensure that the contents are acceptable for publication. Another minor problem would be the need to manually adjust the layout of some content, e.g. images, to provide a more visually pleasing result. Both of these problems can be solved by having the page's HTML be automatically generated, but allowing a volunteer to edit the resulting HTML before the page is posted. The intent would be that the automatically generated HTML would seldom have to be changed, but could if need be.

OTHER POSSIBLE SYSTEMS

Noel Carrascal, a Mars Society member from New York, has worked on a similar system to the one described in this paper, that one based on Java. The authors have currently lost track of this effort, which seems to be stopped at the moment.

The standard communication method of email will always be available, and will probably always be of some use. On a real mission, email-type communication could be provided at the very least as a vehicle for private personal messages to family and friends.

Systems using completely different concepts could be used, either to replace or to complement the type of system proposed here. For example, the Concept Map Tools (CMAP) developed at the University of West Florida[3], and used by the Ames Center for Mars Exploration (CMEX)[4] to explore Mars exploration issues, could be used as a way to organize the exchange of ideas between the Mars crew and scientists on Earth.

Any of these application-level systems, including the one we propose, could eventually be tested with the Interplanetary Internet protocols [Cerf01] rather than with the standard TCP/IP Internet protocols.

INTEREST WITHIN THE MARS ANALOG RESEARCH COMMUNITY SO FAR

The software that we developed is not very polished and the authors are already aware of many required improvements. However, this software was developed as a volunteer effort, and at this stage is is unlikely that much more time will be spent on it until the authors receive some active interest from others in trying it out. Moreover, many of the improvements currently considered are dependent on assumptions about the usage patterns of the system, and the usage patterns will be better known only once the system has been used in its current state for a little while.

The Rocky Mountain Mars Society, the chapter which logged the most direct mission support time so far for our analog stations, had expressed some interest in such a system, but never got around to trying it out. The Mars Society of Canada, who have volunteered to provide part of mission support for the 2002 season of FMARS, has installed our software on one of their server, and intend to experiment with the system for future mission support tasks. They have proposed to “replay” the email messages exchanged in previous FMARS and MDRS seasons through our system (defining the required, tables, lists, and forms in the process), as a way to analyze that archived data and as a way to evaluate the potential usefulness of our system. At the time of this writing, they have not started this project yet. Other groups have expressed interest.

ACKNOWLEDGMENTS

The authors would like to thank the participants of the MDRSinfo Yahoo! Groups for their participation in discussing the concepts of this tool, Randall Severy of Cyberteams for allowing us to use his server to run our tool, and the Mars Society of Canada, the Rocky Mountain Chapter, and the Northern California Chapter for their interest and feedback.

REFERENCES

[Clancey01]: William J. Clancey, Field Science Ethnography: Methods for Systematic Observation on an Expedition, NASA Ames Research Center, June 2001 (http://ic.arc.nasa.gov/ic/publications/pdf/2001-0258.pdf)

[Cerf01]: Vint Cerf et al, Interplanetary Internet (IPN): Architectural Definition, Interplanetary Special Interest Group of the Internet Society, May 2001 (http://www.ipnsig.org/reports/memo-ipnrg-arch-00.pdf)

NOTES

1. A web application server. See www.zope.org.
2. Python is the main scripting language used in Zope. See www.python.org.
3. http://cmap.coginst.uwf.edu/
4. http://cmex.arc.nasa.gov/CMEX/Map%20of%20Maps.html

Figures

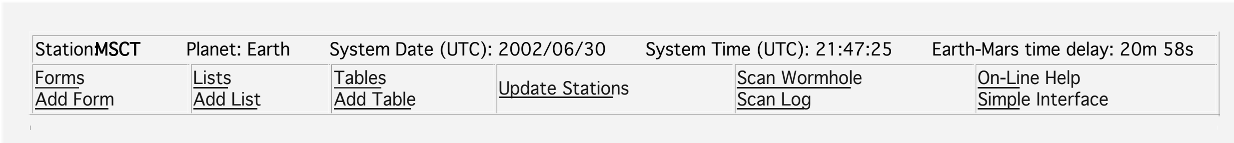


Figure 1 - The main menu, present at the top of every page.

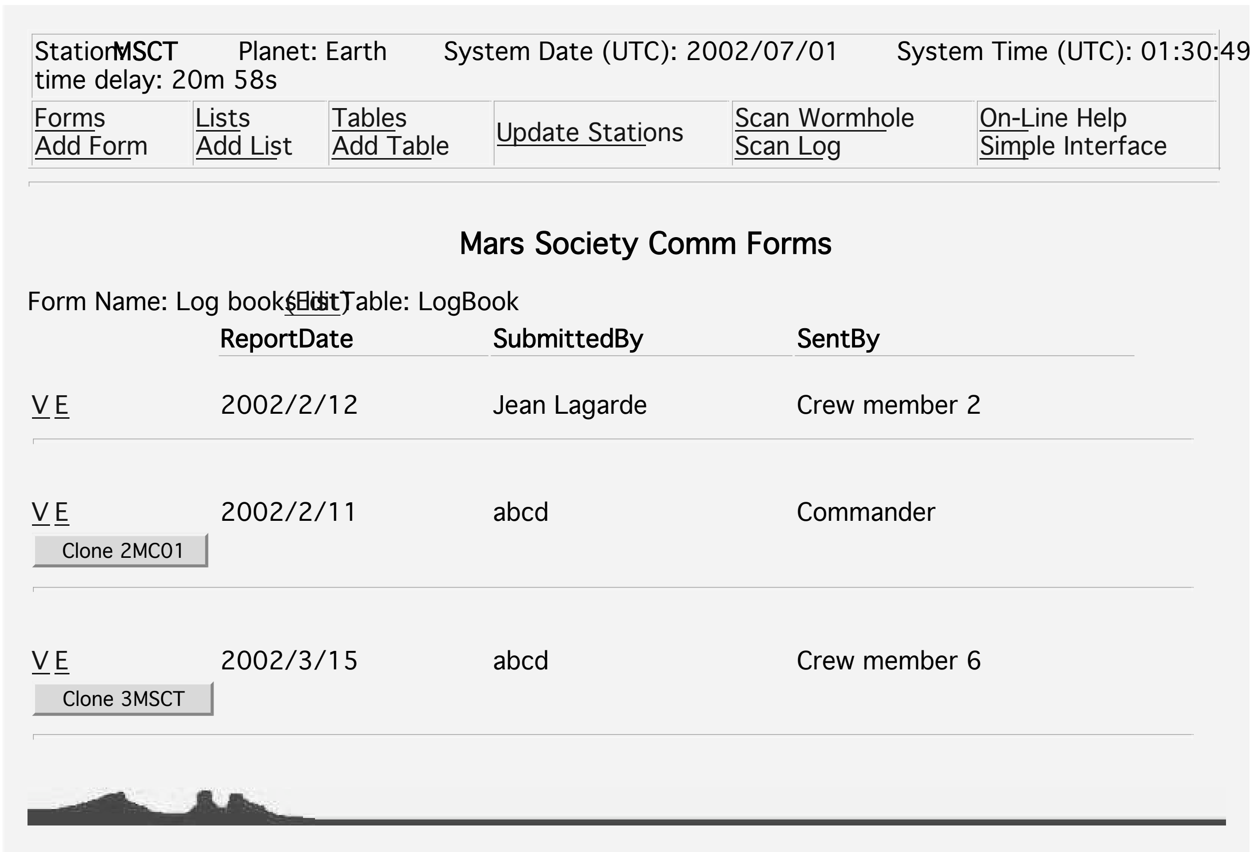


Figure 2 - Example of a form with view and edit links to other forms

Form Name: Edit log book (Edit) Table: LogBook

ReportDate
2002/2/11

SubmittedBy
abcd

SentBy
Commander

OverallComments
We are having com problems that have delayed transmittal of voice report. We are fine. EVA today travelled 7 km to NNW. lots of sampes and data. Frank arrived today with keyboard and UPS. Voice and written reports and photos to follow.

ReportIntro	Report
Bob Zubrin's Status Report	Bob Zubrin's Status Report11Feb.html Delete
Engineering report from Frank Schubert	Engineering report from Frank Schubert.ht Delete
Add New Subrecord:	
	Browse...

FileCaption	File
	Add: Browse...
Add New Subrecord:	

ImageCaption	Image
EVA team leaving hab	img_1880.jpg Delete

ImageCaption	Image
EVA team on plain between local ridge and Skyline Rim. RZ	img_1896.jpg Delete
Add New Subrecord:	

Figure 3 - Example of a form in action

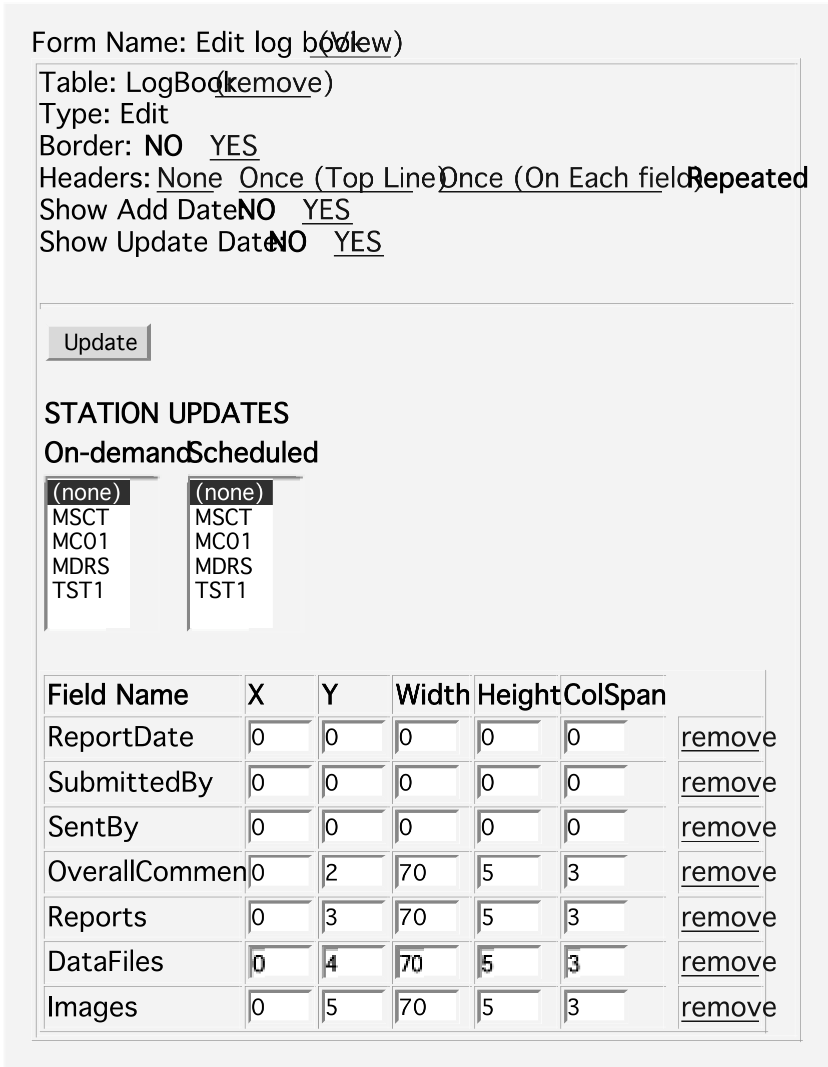
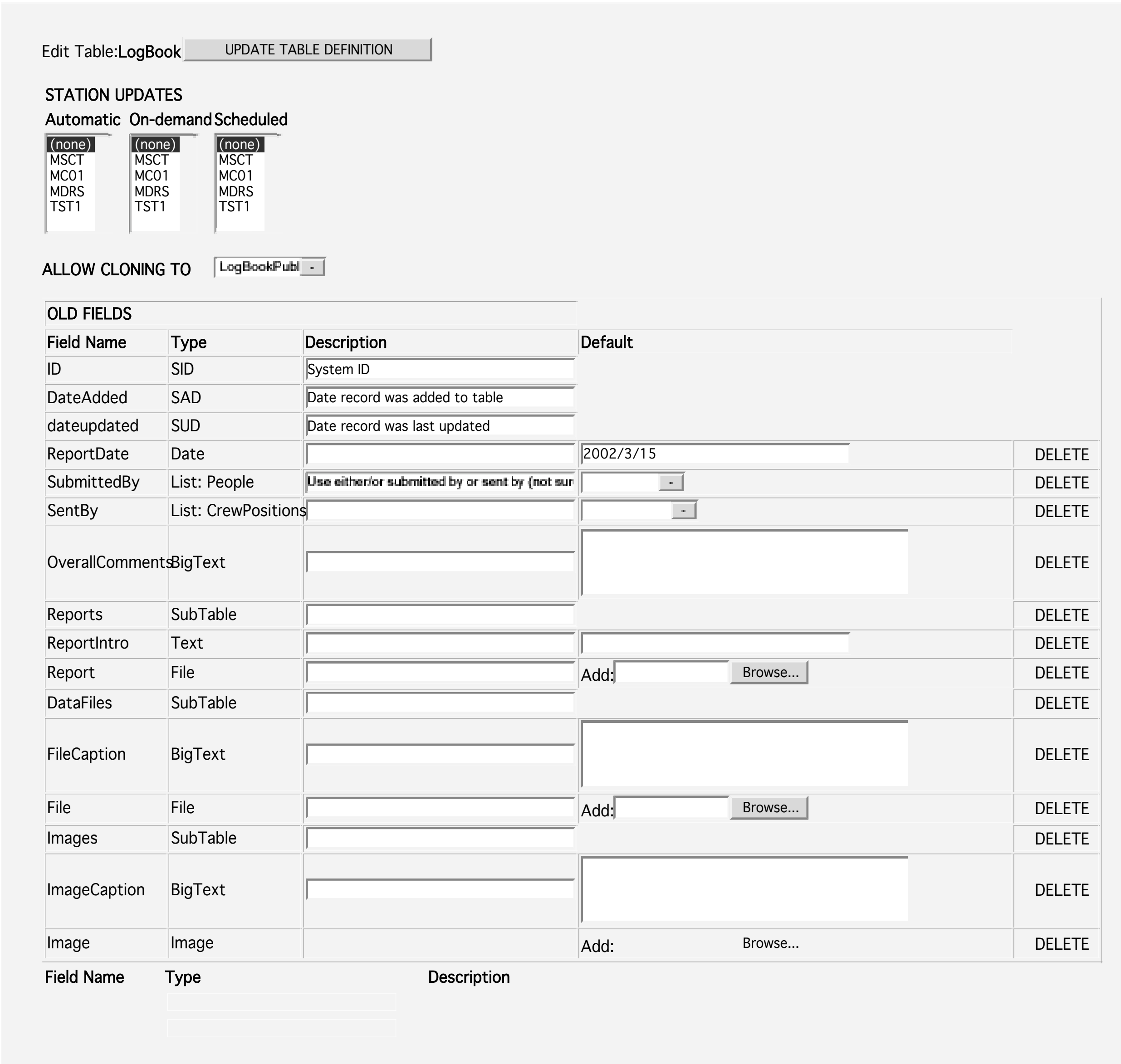


Figure 4 – Example of a form layout definition



Edit Table:LogBook UPDATE TABLE DEFINITION

STATION UPDATES

Automatic	On-demand	Scheduled
(none)	(none)	(none)
MSCT	MSCT	MSCT
MC01	MC01	MC01
MDRS	MDRS	MDRS
TST1	TST1	TST1

ALLOW CLONING TO LogBookPubl

OLD FIELDS

Field Name	Type	Description	Default	
ID	SID	System ID		
DateAdded	SAD	Date record was added to table		
dateupdated	SUD	Date record was last updated		
ReportDate	Date		2002/3/15	DELETE
SubmittedBy	List: People	Use either/or submitted by or sent by (not sur		DELETE
SentBy	List: CrewPositions			DELETE
OverallComments	BigText			DELETE
Reports	SubTable			DELETE
ReportIntro	Text			DELETE
Report	File		Add: Browse...	DELETE
DataFiles	SubTable			DELETE
FileCaption	BigText			DELETE
File	File		Add: Browse...	DELETE
Images	SubTable			DELETE
ImageCaption	BigText			DELETE
Image	Image		Add: Browse...	DELETE

Field Name Type Description

Figure 5 – Example of a table definition form

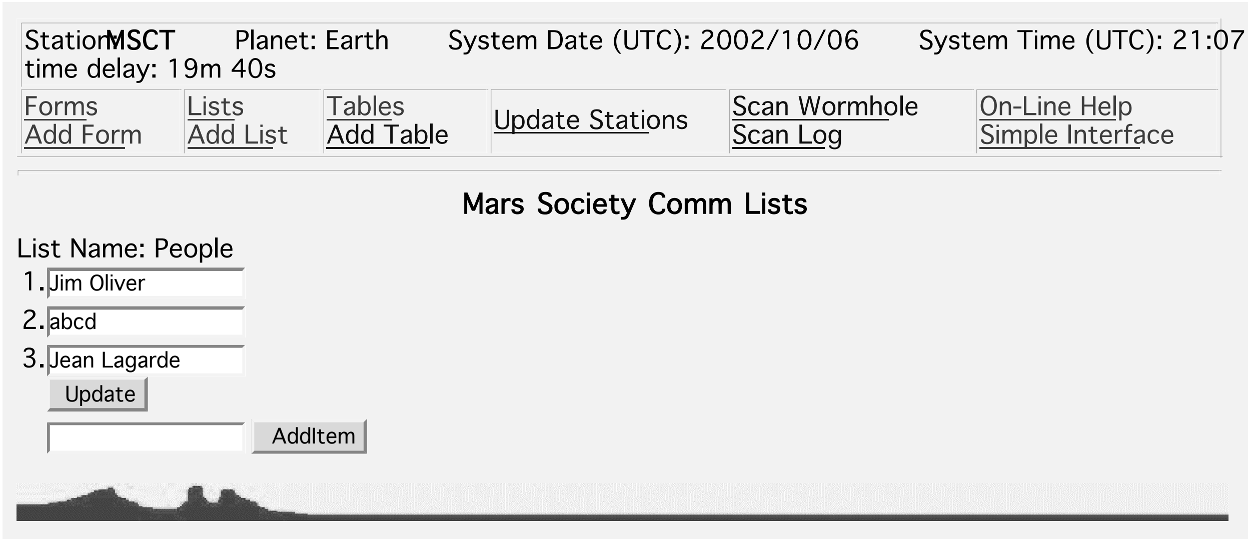


Figure 6 - List type definition screen

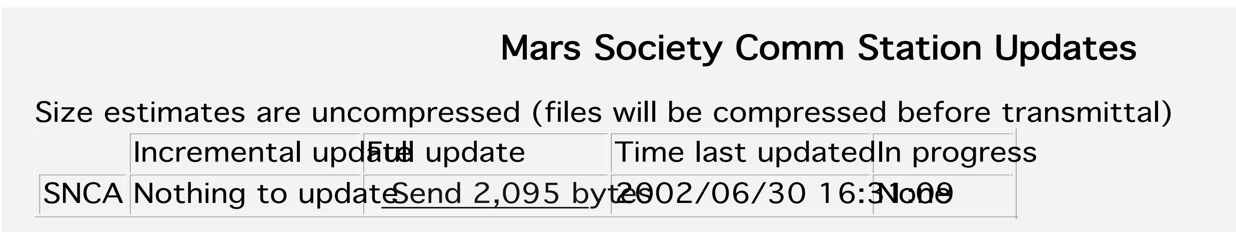


Figure 7- Screen used to command on-demand updates

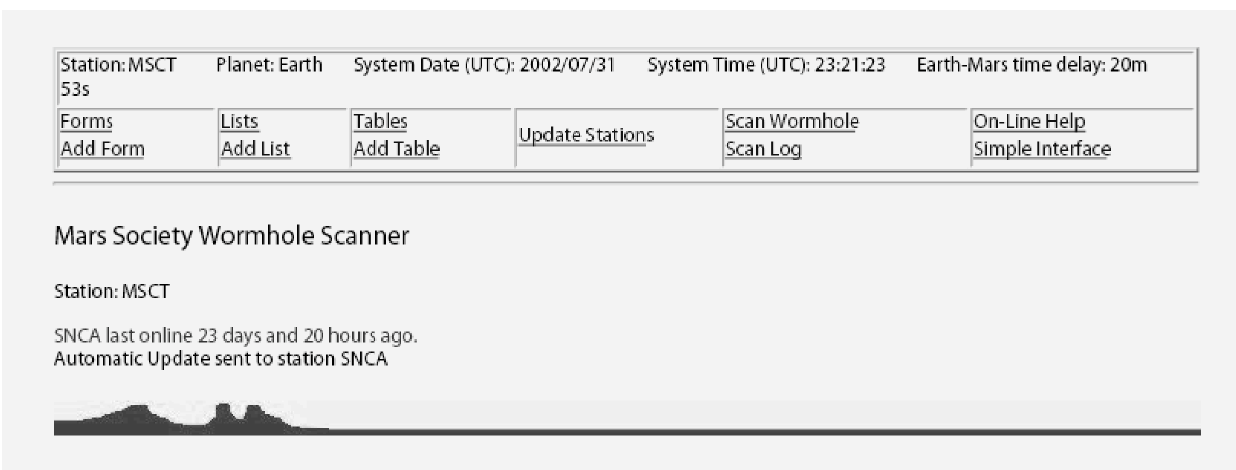


Figure 8 - Auto reload screen used to send and receive updates unattended

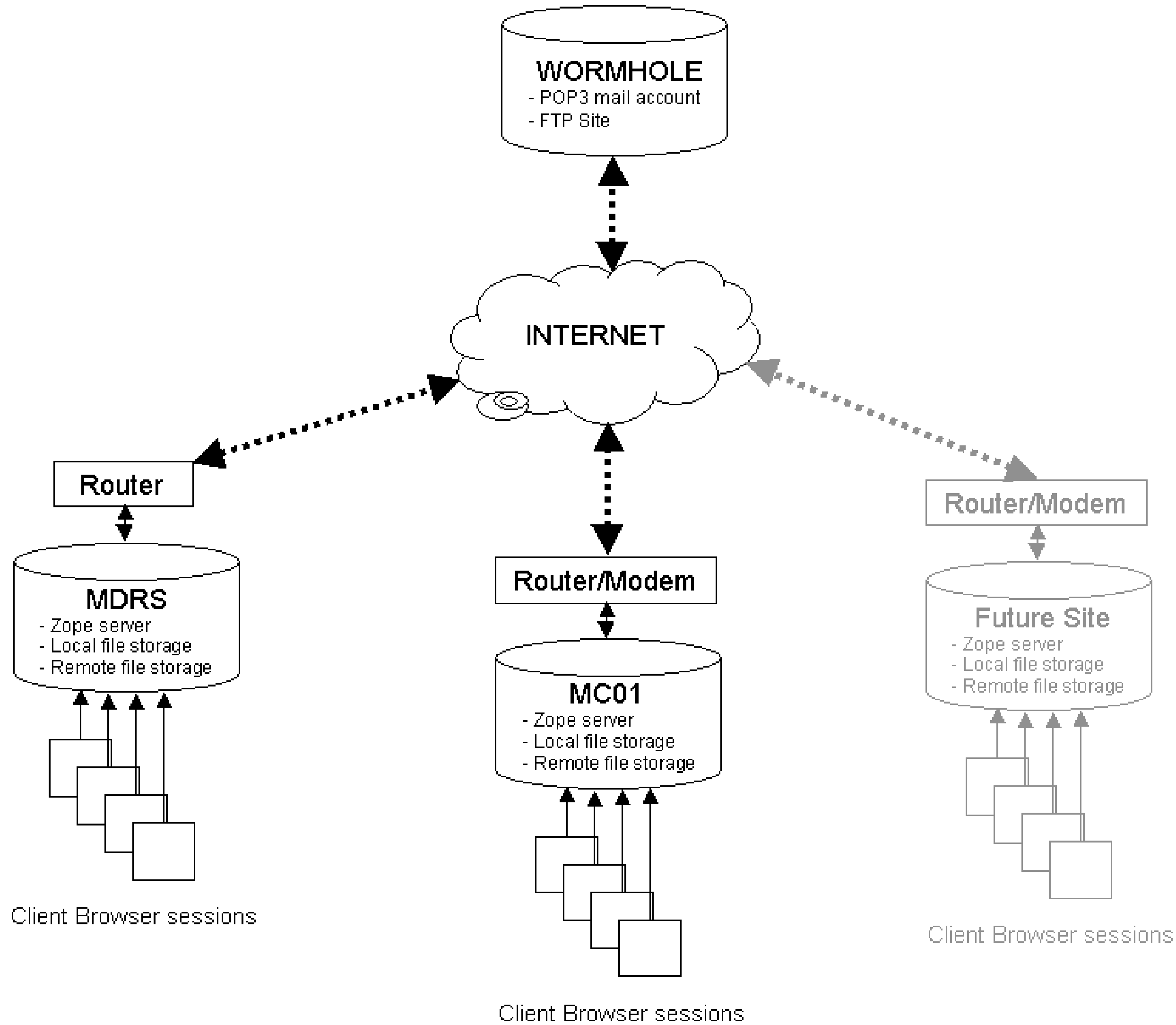


Figure 9 - Top level communication architecture