Generic Distributed Mission Control Center for Student Satellites

Kristian Edlund, Martin Green, Martin Kragelund, Axel Michelsen, and Rasmus Stougaard
Group 733 (04gr733@space.aau.dk) - December 2004

Department of Control Engineering, Aalborg University, Denmark

***Abstract*— This paper deals with the development of a generic mission control center for use with SSETI Express [1] and AAUSAT-II [2], which are student satellite projects by the European Space Agency and Aalborg University, respectively. The mission control center will be distributed through the Internet utilizing ground stations in Aalborg and Svalbard to establish communication with these satellites. The objective is to allow tele operation through a graphical user interface from a remote location. Stanford University offers an advanced system for distributing ground stations [3]. However, as this is still under development it is not relied upon in this project.**

The design utilizes the open source database management system MySQL for storing all communication with the satellites. The database design was complicated by the differences between the communication protocols for SSETI Express and AAUSAT-II and a generic design of the mission control center was required. Java was chosen as implementation language as it provides convenient networking features such as Remote Method Invocation, which was used for distributing the graphical user interface through the Internet. Furthermore, the portability of Java programs allows the graphical user interface to run on most hardware platforms and operating systems.

The implementation comprises a graphical user interface which allows generic definition of tele commands and telemetry formats based on the SSETI Express communication protocol. It supports manual and automated communication through multiple ground stations provided that appropriate drivers are implemented. During integration at the European Space Research and Technology Centre (ESTEC) a communication link to the SSETI Express satellite was established, and telemetry was successfully received and stored in the database.

Defining the tele commands and telemetry in the database provided a generic framework for the communication protocol. For the mission control center to function with AAUSAT-II, reimplementation of some protocol specific parts are required. However, the developed mission control center framework can be reused.

***Index Terms*— Satellite, Distributed Systems.**

I. Introduction

An important part of building a student satellite is establishing communication from the Earth, which is usually achieved utilizing a radio link from a ground station. In addition to establishing a communication link the operator must be able to communicate with the satellite by transmitting tele commands and receiving telemetry. These are essentially bit patterns and in order to provide a more intuitive interface for the operators the Mission Control Center (MCC) allows communication with the satellite to be handled at a higher level of abstraction.

The MCC is part of two student satellite projects. The Student Space Exploration & Technology Initiative Express (SSETI Express) is a satellite project managed by the European Space Agency (ESA) and developed and constructed as a collaboration between universities all over Europe [1]. The satellite is an $80\,kg$ micro-satellite made for educational purposes. The mission objectives for SSETI Express are to deploy three CubeSat [4] pico-satellites, take pictures of the Earth, act as a test bed and as a technology demonstration for hardware of the complementary project - the European Student Earth Orbiter (ESEO), and function as a radio transponder for radio amateurs [1].

The other satellite project is AAUSAT-II, which is a $1\,kg$ CubeSat pico-satellite also developed for educational purposes by students at Aalborg University [2]. The satellite carries two scientific experimental payloads; an Attitude Determination and Control System (ADCS) developed at Aalborg University and a gamma ray detector supplied by Danish Space Research Institute (DSRI). The technical mission objectives for AAUSAT-II are; establishing one-way communication, establishing two-way communication, detumbling the satellite using the ADCS, and detecting gamma ray bursts [2].

As the MCC is part of both projects it must be capable of interfacing both satellites, with interfacing to SSETI Express considered as the highest priority, due to the imminent launch in June 2005. As of this writing the launch date for AAUSAT-II will be in late 2005.

Two ground stations are available for SSETI Express; one in Aalborg, Denmark and another in Svalbard, Norway. The ground station at Svalbard is primarily servicing NCUBE-2, a Norwegian pico-satellite being deployed by SSETI-Express. Thus, usage of that ground station is limited by NCUBE-2. A system for managing multiple ground stations, Mercury Ground Station system, already exists. It is implemented as a test bed for the Ground station Markup language. The developers expect to have the basic operation of Mercury ready in 2004 [3]. As there still seems to be problems with the stability of the networking functionality, the MCC will not rely on it for interfacing ground stations even though it offers advanced features.

This paper will provide an overview of the requirements for a generic MCC followed by a description of the design chosen under consideration of the requirements. The description of the design will include an elaboration of some of the key elements in the design and complete the description with a protocol summary. Subsequently, the results achieved from the MCC implementation and from preliminary tests will be presented, and finally the results are discussed and a conclusion on the MCC development is drawn.

A. Requirements

A use case diagram delineating the relationship between actors and privileges is depicted in Figure 1.

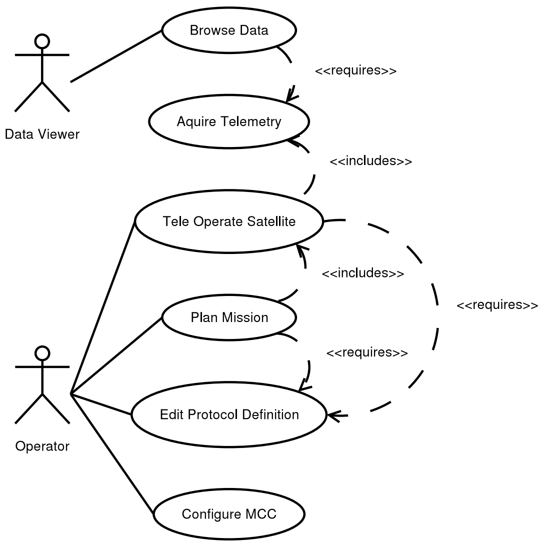

Fig. 1. Use case diagram for the MCC.

The MCC must accommodate data viewers who are interested in telemetry from satellites and operators who are responsible for tele operating the satellites. In order for the MCC to be generic, operators must have access to configure the MCC and modify protocol definitions. As the operations team for the SSETI Express mission is not situated near the physical ground station, the MCC must facilitate remote access.

To ease the workload of the operators the MCC must provide functionality for planning a mission by creating flight plans for upload during a future satellite pass. This automates parts of the tele operation of a satellite, hereby eliminating the need for continuous presence of operators. The MCC must facilitate an authentication scheme to prevent multiple users from accidentally issuing conflicting tele commands and to provide some level of security against unauthorized individuals.

The data viewer must have access to browse telemetry through an intuitive interface distributed through the Internet so that scientific data obtained during a satellite mission is publicly available. Since data integrity must be maintained, the data viewer must under no circumstances have access to modify telemetry through this interface.

Since two ground stations are presently available and more might be incorporated at a later stage, the MCC is required to be scalable with regard to the number of supported ground stations. Furthermore, the MCC should provide a reliable and fault tolerant platform for interacting with the satellite.

II. Design

The overall structure of the MCC design is based on the requirements previously described and is depicted in Figure 2. The structure comprises a Mission Control Server (MCS),

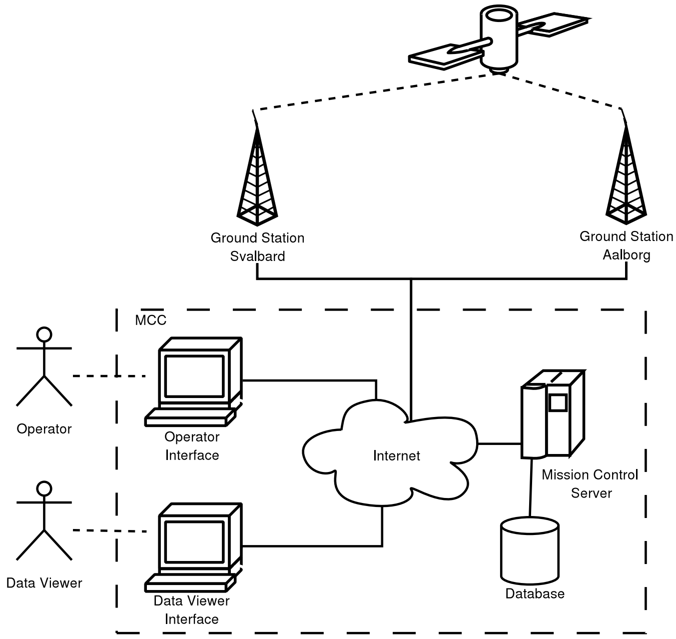

Fig. 2. Overall structure of the MCC.

which is the main computer in the MCC running the operating system Debian. The MCS handles communication with a database, ground stations and user interfaces.

All data transmitted to and received from the ground stations and configuration data for the MCC is stored in the database, which utilizes the open source Database Management System (DBMS) MySQL. A simple scheme of periodic backup is invoked to ensure data integrity.

The operator has access to control the satellite using the Operator Interface (OI), which is designed as a thin client program. Thus, most of the program logic and data handling is performed in the MCS. The Java programming language is chosen for implementing the MCS and the OI.

The Data Viewers Interface (DVI) allows read only access for the public to browse telemetry using a standard web browser. However, the task of designing and implementing the DVI has been assigned to another team of students in the SSETI Express project and will therefore not be further addressed in this paper.

A. Radio Link and Communication Protocol

The communication channel to the satellite is half duplex and it is not possible to sense a collision in the medium. Hence, the only way to ensure communication reliability is to receive acknowledge on transmitted packages. The impossibility of collision detection, dictates the use of collision avoidance algorithms.

The radio link utilizes the AX.25 protocol [5], which is used for packet radio by radio amateurs and supports both connection oriented and connectionless communication. The connectionless part of AX.25 is used for SSETI Express and AAUSAT-II by encapsulating data packages in unnumbered information frames. An example of this is shown in Figure 3, where a tele command is contained in the *Info* field of the AX.25 frame.

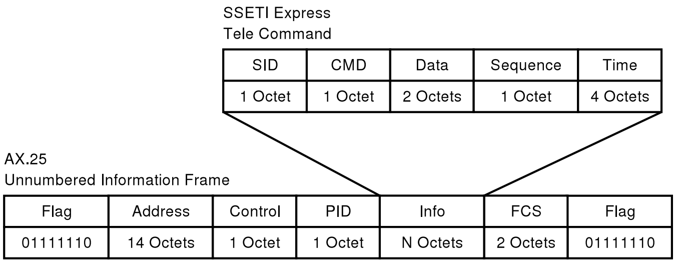

Fig. 3. AX.25 encapsulation of a SSETI Express tele command.

The architecture of the protocol used by SSETI Express differs from that used in AAUSAT-II. As depicted in Figure 3 the tele commands used for controlling SSETI Express consist of five fields.

- *SID* is a system identifier used for indicating which satellite subsystem the tele command is affiliated with.
- *CMD* is the tele command for the selected subsystem.
- *Data* is the parameters for the tele command.
- *Sequence* indicates the execution sequence for tele commands with equal timestamps.
- *Time* is a timestamp with a resolution of $1\,s$ indicating when the tele command should be executed.

SID	MID	Length	Data	Time
1 Octet	1 Octet	1 Octet	0-64 Octets	4 Octets

Fig. 4. Telemetry package used in the SSETI Express protocol.

Telemetry packages from SSETI Express contain the following fields as depicted in Figure 4.

- *SID* is a system identifier used for indicating which subsystem in the satellite the tele command is affiliated with.
- *MID* is a measurement identifier.
- *Length* indicates the length of the *Data* field in octets.
- *Data* is the measured data.
- *Time* indicates the time of measurement with a resolution of $1\,s$.

Using the connectionless part of AX.25 the connection orientation must be handled in a higher protocol layer for the connection to be reliable. The communication scheme for both satellites rely on a master/slave concept, with the satellite as slave, i.e., the satellite only transmits if dictated to do so by the MCC. The advantage of placing responsibility for connection reliability on ground is evident when considering accessibility. If communication to the satellite fails it is nearly impossible to repair faulty onboard software, in contrast to systems on earth which are easily accessible and changing the software in these is a manageable task.

SSETI Express and AAUSAT-II uses different acknowledgement schemes. SSETI Express transmits an acknowledge in response to every received package, whereas the protocol used for communicating with AAUSAT-II provides an acknowledge flag. When AAUSAT-II receives a package containing an enabled acknowledge flag the satellite will respond with an acknowledge on successfully received packages. Use of this scheme dictates the need for a way to flush the acknowledge buffer, which is done by setting another flag indicating that the buffer must be flushed. Both acknowledgement schemes are reliable and both place the responsibility for connection maintenance with the MCC, but the AAUSAT-II scheme increases the connection efficiency.

Facilitating the acknowledgement scheme used for SSETI Express, the MCC includes a retransmission scheme design as depicted in Figure 5.

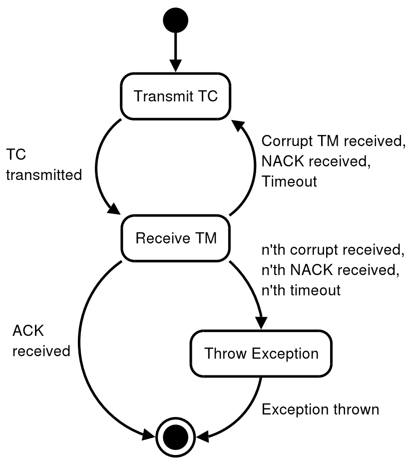

Fig. 5. State diagram for retransmission.

When a tele command has been transmitted the MCC will await an acknowledge. If the acknowledge is not successfully received within a certain time limit the MCC will retransmit the tele command until an acknowledge is received, or an unacceptable number of unsuccessful retransmissions has occurred in which case an exception is thrown.

B. Distributed Ground Stations

The support of multiple ground stations in the MCC introduces problems as the use of each ground station must be coordinated and the interfaces to different ground stations are most likely not identical. The ground station in Aalborg is controlled by requesting it to track a certain satellite; when the satellite is in range data can be transmitted and received. If errors occur during transmission the ground station will issue error messages.

To handle the various ground station interfaces a driver must be implemented for each station. These drivers must have the same interface in order for the MCC to utilize every ground station in a similar manner. The MCC is designed in such a way that the task of allocating usage of ground stations is manually controlled by the operator and a multiplexing layer distributes communication to the designated drivers.

C. Tele Operation of Student Satellites

A substantial part of a MCC is the communication patterns that emerge when the operator interacts with a satellite. The interaction between the MCC, SSETI Express, and the operator is illustrated in the state diagram in Figure 6.

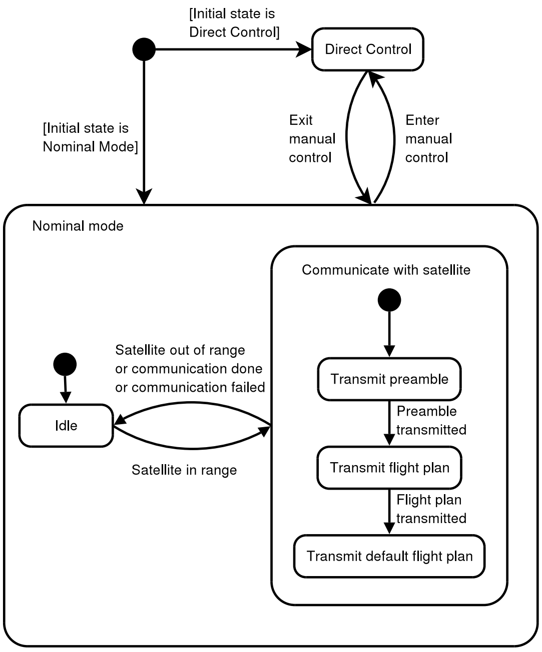

Fig. 6. State diagram of overall states.

The control of the satellite is divided into two main modes. *Direct control* disables all forms of automated control, giving the operator manual control of the communication between satellite and MCC.

Nominal mode facilitates automated control providing the operator with the possibility to plan a mission and automating the scheduled communication when a radio link can be established with the satellite. The autonomous communication between MCC and the satellite is decomposed into three different flight plan sub-categories.

- *Preamble* consists of tele commands that will be transmitted subsequent to establishment of the radio link to the satellite, e.g., time synchronization, download of alarms, and mission critical telemetry.
- *Flight Plan* consists of tele commands that should be communicated to the satellite during the upcoming pass and subsequently deleted, e.g., instructions for scientific experiments to be executed.
- *Default Flight Plan* consists of tele commands that should preferably be executed as often as possible, but are not crucial to the continued operation of the satellite, e.g., the download of scientific data.

Editing the three flight plans cannot take place while communication with the satellite, this is only possible when the satellite is out of range and the MCC enters the *idle* state, or if the operator enables direct control.

In direct control mode the operator decides how to handle transmission errors, and when retransmission fails in nominal mode the MCC will return to idle state until the next pass of the satellite.

Performing tasks such as downloading alarms from the satellite requires the operator to issue several tele commands and analyze the telemetry received to determine the course of further interaction. To aid the operator when performing these tasks the MCC design supports the automation of these tasks through executable macros.

D. Database design

For preservation of communication data the MCC utilizes a database. This database also contains information concerning ground stations and related communication logs of control messages from these, thus facilitating the gathering of statistical information concerning the performance of each ground station. Furthermore tele command sequences for the three flight plans used in nominal mode communication are also stored in the database.

To restrict the operator to only transmit predefined tele commands definitions of these are stored in the database. Combining these with the definitions of telemetry provide a basis for interpreting the communication with the satellite.

Figure 7 depicts an entity relationship diagram for the database without attribute definitions and entities are described in the following.

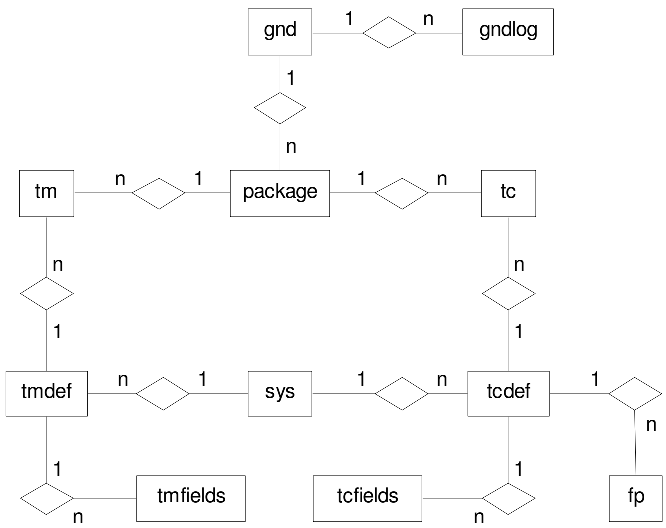

Fig. 7. ER-diagram for the database without attribute definitions.

- *package* holds the raw packages that has been transmitted to or received from the satellite. When data is transmitted/received each package is ascribed an identification number, a timestamp and a package type, either transmit or receive.
- *gnd* contains information about ground stations. This facilitates the addition of extra ground stations and editing information affiliated with each ground station. The relation to *package* is made for statistical reasons, hereby providing the possibility to track from which ground station a packet was transmitted or received.

- *gndlog* holds status and error messages from the ground stations defined in *gnd*.
- *tm* and *tc* contains decoded packages transmitted to or received from the satellite. The reason for dividing the decoded data packages into two different entities is the dissimilarity in the format of the packages.
- *sys* holds information concerning satellite subsystems.
- *tmdef* and *tcdef* contains information used for interpretation of telemetry and telecommands, hence the relation to *tm*/*tc* and *sys*.
- *tmfields* and *tcfields* hold definitions of data fields in tele commands and telemetry.
- *fp* contains the sequence of tele commands in the preamble, the flight plan and the default flight plan.

E. Distributed Operator Interface

The OI is designed as a graphical user interface and is distributed through the Internet. Implementing the MCC in Java offers the advantage that the resulting program will be portable, thus allowing the OI to run on any hardware with the Java runtime environment installed. Furthermore, Java provides the possibility of using Remote Method Invocation (RMI), which is a communication protocol that allows the sharing of objects between multiple machines on a network.

Only input checking is performed locally in the OI and tele commands are communicated to the MCS using RMI resulting in a thin client.

The MCC is designed as a single user system as this will eliminate the risk of several operators accidently merging flight plans, hereby avoiding the need for developing a system with version control and complicated access control. The authentication scheme requires the operator to provide a valid username and password combination for identification and security against destructive individuals with no access rights. As a further security measure communication between the OI and MCS utilizes a secure connection.

F. Protocol Summary

The collective description of the communication path from the OI through the MCC and the ground station to satellite can be surveyed by examining the protocol stack, presented in Figure 8.

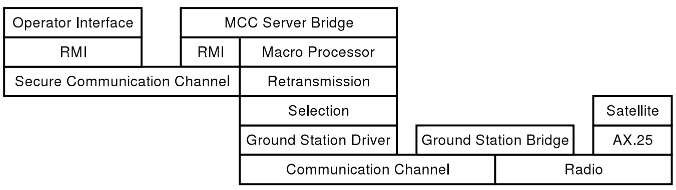

Fig. 8. Protocol stack for communication with the satellite.

The tele commands issued by the operator using the *Operator Interface* are transmitted through the Internet to the *MCC Server Bridge* utilizing *RMI*. From the *MCC Server Bridge* tele commands are transmitted in accordance with the state diagram in Figure 6.

All tele commands are passed to the *Macro Processor* which filters and interprets macros. If the tele command is not a macro it is passed directly to the *Retransmission* layer, which handles retransmission of packages as described in Section II-A, hereby facilitating a reliable communication channel.

The *Selection* layer serves as multiplexer, distributing communication from the *Retransmission* layer to the chosen *Ground Station Driver*, which then transmits the tele command through the communication channel utilized by the affiliated ground station. The *Ground Station Bridge* handles repackaging of the tele command and the *Radio* transmits it to the satellite as an *AX.25* package.

III. Implementation and Test

Presently the Mission Control Center is ready for testing with the SSETI Express satellite. However, some modifications must be applied prior to use with the AAUSAT-II satellite, due to the fact that some features are specifically designed and implemented for the SSETI Express protocol.

By implementation of the OI access is provided to the functionality of the MCC through nine panels. The panels can be selected and detached from the main window using tabbed navigation. To aid the operator, hints for usage are displayed when the mouse pointer hovers over an item.

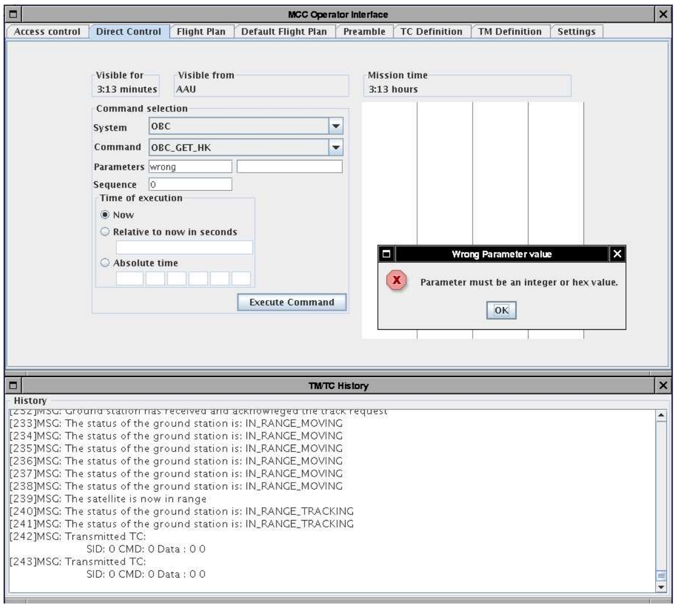

Fig. 9. Direct control panel with alert box and detached history panel below.

Figure 9 shows a screenshot of the OI with the direct control panel selected in the main window and the history panel detached below. Feedback from the MCC is presented to the operator through the history panel and alert boxes. User input is checked for consistency with the tele command definition and erroneous input triggers an alert box.

For each subsystem the operator can define a set of valid tele commands, using the tele command definition panel depicted in Figure 10. Tele commands can be added or, as is the case in the figure, existing tele commands can be edited. Only predefined tele commands can be issued to the satellite.

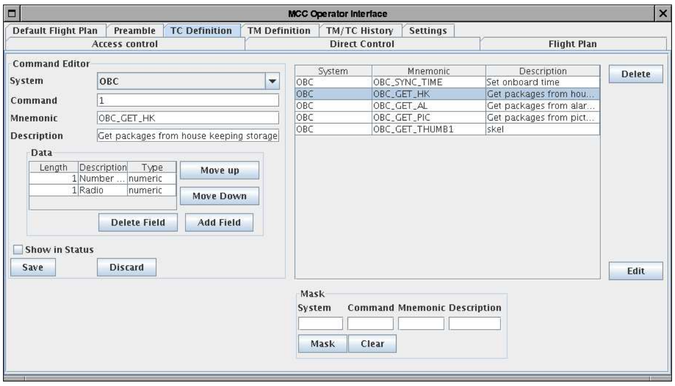

Fig. 10. The tele command definition panel.

Proper macro execution requires all issued tele commands are received correctly by the satellite. To facilitate this an algorithm retransmitting all unacknowledged packages was implemented. Retransmission can be disabled, this will disable simultaneous use of macros. The following macros have been implemented in the MCC.

- Synchronizing time using the MCC system clock, such that the operator is not required to enter the desired timestamp.
- Download of alarms and subsequent removal from the satellite after successful download.
- Download of housekeeping data and subsequent removal from the satellite after successful download.
- Download of picture data and subsequent removal from the satellite after successful download.

The MCC supports automatic transmission of predefined flight plans as prescribed in Section II-A. Thus the MCC is capable of communicating with the satellite without constant supervision by the operator.

The settings panel controls the use of flight plans as depicted in Figure 11. Other options which can be changed from this panel are the use of retransmission, direct control, which satellite to track, and what ground station to use for the radio link.

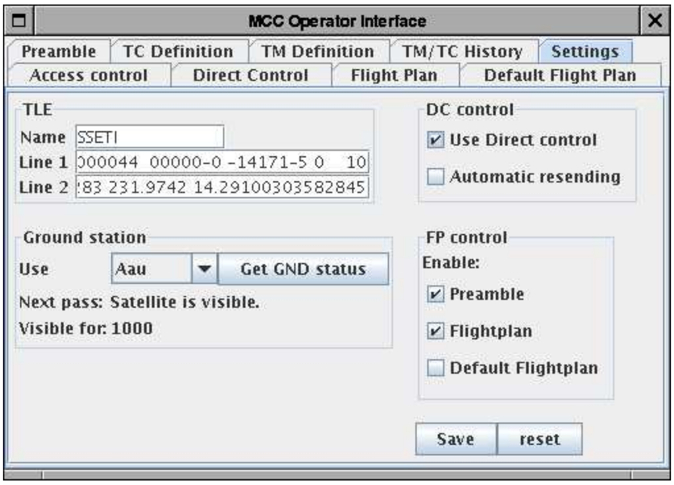

Fig. 11. The settings panel.

Telemetry formats can be defined by the operators in a manner similar to defining tele commands. When publishing the received telemetry these definitions are meant to be utilized by the DVI for presenting the data in a more comprehensible format.

Additional ground stations can be incorporated in the system by creating drivers for these and patching the MCC software with these. As of this writing only the ground station in Aalborg is supported as the driver for the Svalbard ground station is still in progress.

A. *Testing the Mission Control Center*

The development of the MCC has been structured so that the functionality of each component has been verified prior to further integration with other components.

A preliminary test was conducted at the European Space Research and Technology Centre (ESTEC) where the first successful communication between MCC, ground station and satellite was attained. However, some transmissions were unsuccessful which was not unexpected since the three systems were under development. The implementation errors discovered at ESTEC were corrected and a second preliminary test of the MCC and satellite was performed as depicted in Figure 12. The MCC communicated with the satellite through

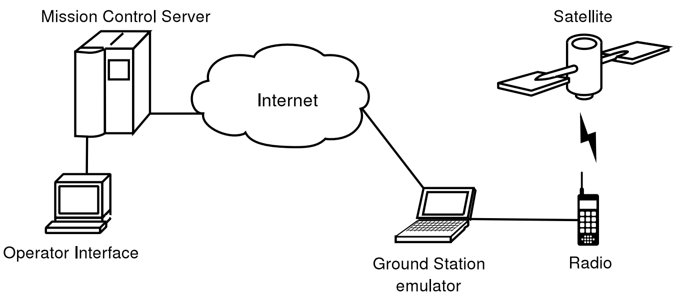

Fig. 12. Communication path for the second preliminary test.

the Internet and a ground station emulator situated at ESTEC. This revealed that the MCC was able to transmit predefined tele commands to the satellite, receive telemetry and store all communication data in the database.

Further functional tests are required as the previous tests are not considered sufficient. The functional test should be structured such that the functionality of the MCC will be verified and the test should include the satellite and ground station. This could be performed using the scenario depicted in Figure 13, where a bridge reroutes data from the serial port on the ground station at Aalborg University to a modem situated at ESTEC.

IV. Discussion and Future Work

The MCC is developed to generically handle communication with a satellite, automating some of the operators' tasks. By enforcing the strategy of predefining tele commands, erroneous definitions of tele commands are more likely to be corrected as they will probably be discovered during functional testing. The MCC implements an interface for

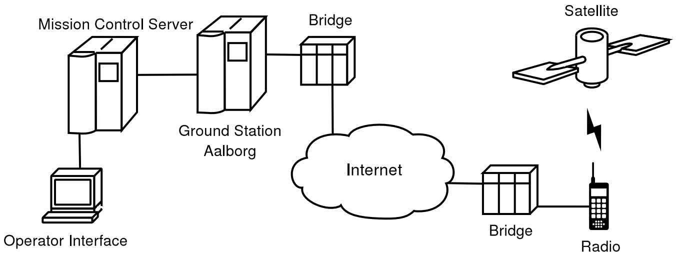

Fig. 13. Functional test of the MCC.

ground stations preparing the MCC for support of satellite links through several ground stations. This will require drivers to be implemented for each additional ground station, i.e., the MCC is generic regarding the addition of ground stations.

The task of selecting ground station in the MCC is manually controlled by the operators. This task could be eased by introducing a scheduler for the usage of the ground stations, even though this would require full knowledge of the ground station usage restrictions for every ground station included in the MCC. The scheduling functionality could be incorporated by merging the MCC with the Mercury Ground Station system, this system, however, is at an experimental stage.

The database has been implemented to give a generic framework for the communication protocol as the definition of tele commands and telemetry can be dynamically defined. This implementation has a limitation in the sense that the communication has to conform to a certain syntax, in this case the syntax of the SSETI Express communication protocol.

The protocol for SSETI Express uses a relatively simple acknowledgment scheme, however, it is not well suited for ground stations such as the Aalborg University ground station which has a receive/transmit switching time of $150\,ms$. The acknowledge scheme for AAUSAT-II implements a transport protocol that enables communication between the MCC and the satellite with fewer occurrences of receive/transmit-switching.

Furthermore, for the MCC to function with the AAUSAT-II, minor parts of the MCC implementation has to be modified. This includes the implementation of the DVI as the re-usability and availability of the DVI developed by the SSETI Express DVI team cannot be relied on. A generic MCC would facilitate definition of communication protocols, which is not the case with the developed MCC, as it cannot be used for AAUSAT-II without modifications.

On the basis of the knowledge generated in the process of developing the MCC a generic mission control center could be developed. However, the generic MCC would have to incorporate a tool for modelling the protocol, e.g. a tool supporting SDL, hereby increasing the complexity of the MCC. On the other hand, if a standardized protocol for student satellite communication is developed, the need for a protocol description language is eliminated. Either of the two possibilities will result in a generic MCC without further additions to the MCC design.

A. *Conclusion*

This paper has elaborated on the development of a generic mission control center for student satellites. The developed MCC is capable of communicating with SSETI Express through a ground station and a generic ground station interface preparing the MCC for interfacing with other ground stations has been developed. Furthermore, a transport layer responsible for retransmitting datagrams lost on the physical media is developed to maintain a reliable satellite connection.

To facilitate the control of the MCC and the tele operation of a satellite a distributed graphical user interface has been developed. To further reduce the workload of the operators a macro processor has been developed which automate some of the tasks the operators have to perform. The operation of the satellite has to some extent been automated by designing and implementing a state machine that automatically transmits a predefined flight plan every time the satellite is in range, thereby eliminating the need of continuous supervision from the operator.

A database design containing both data from satellite communication and definitions of tele commands and telemetry was developed. This database design restricted the operator to only transmit predefined tele commands. Additionally the definitions facilitated interpretation of data logged from the communication with the satellite.

Preliminary functional verification indicated that a two way communication to the satellite could be successfully established utilizing a radio link provided by a ground station.

The mission control center is developed to form a protocol stack, that allows for changes in parts of the software without affecting the overall structure of the program. Even though the developed MCC can not communicate with other satellites than SSETI Express it incorporates a generic framework that requires only minor changes in order for the MCC to be able to communicate with other satellites.

References

[1] Student Space Exploration & Technology Initiative
http://www.sseti.org
Visited: December 8th 2004.

[2] AAUSAT-II
http://aausatii.aau.dk
Visited: December 8th 2004.

[3] James W. Cutler
Ground Station Markup Language
Proceedings of the IEEE Aerospace Conference Big Sky, Montana, March 2004.

[4] CUBESAT Design Specifications Document Revision VIII August, 2003
http://cubesat.calpoly.edu/_new/documents/cubesat_spec.pdf
Visited: December 15th 2004

[5] AX.25 Amateur Packet-Radio Link-Layer Protocol
Version 2.2 November 1997,
http://www.tapr.org/tapr/html/Fax25.html
Visited: December 8th 2004.