Development and Use of a Web-based Automated Command Request Application in a Distributed Operations Environment for the Cassini Saturn Mission

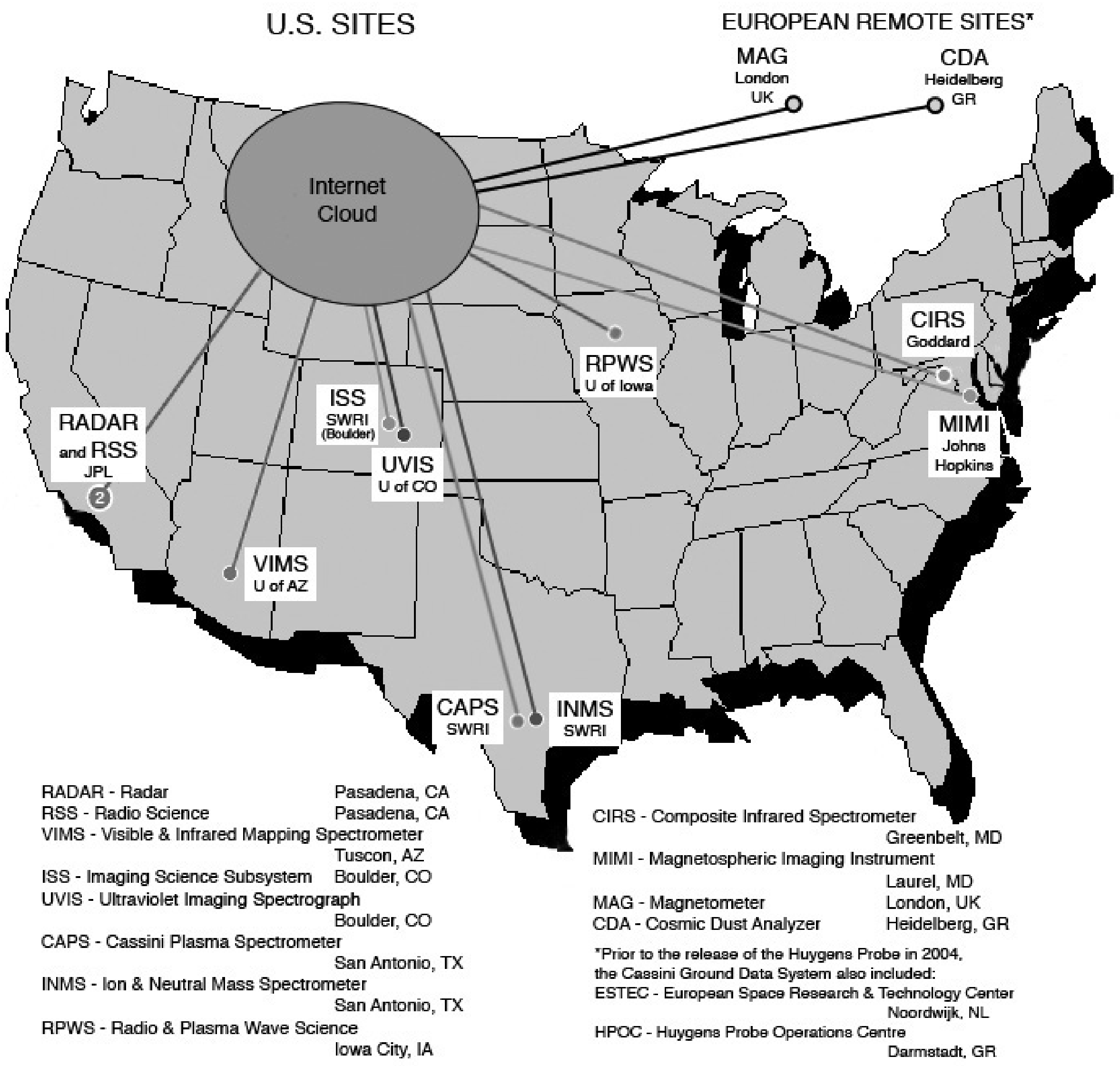
Caroll Wong[*], David Doody[†], William M. Heventhal III[‡], and John Ibanez[§]
Jet Propulsion laboratory, California Institute of Technology, Pasadena, CA 91109, USA

The Ground Data System for the Cassini Saturn Mission is a distributed environment that supports the efforts of operations teams at 10 US and 2 European sites. The Mission Controller, also known as the Ace, in the Mission Support and Services Office (MSSO) at the Jet Propulsion Laboratory (JPL), is responsible for radiating command (CMD) files to the spacecraft which were built by the Sequence Virtual Team (SVT) based on information obtained from various operations teams. Following the launch of Cassini-Huygens in 1997, information needed for processing and radiating each CMD file was assembled on a paper Command Request Form (CRF), initiated within JPL or faxed in from a remote site, and carried by hand from office to office for additional information, validation, and approval. Following approval of the CRF and radiation of the CMD files to the spacecraft by the Ace, the paper CRF forms were stored in a binder. As the Saturn Mission progressed and required an increasing number of CMD file transmissions, MSSO explored the idea of automating the command request process by using a Web-based electronic CRF (eCRF). The principal benefits of the eCRF were the increased efficiency and convenience of electronic generation, tracking and storage of information regarding CMD files; and, the convenience of allowing authorized team members from operations sites worldwide to interactively view and/or contribute to a command request in progress, or search for information regarding previously stored command requests, without the need for having the paper forms in front of them. Additional features not possible with a paper form, such as automatic insertion of selected data, error checking, highlighting of required data fields, and automatic email notification, were incorporated. The result was a tool that gradually earned the confidence of an ever-wary user community, and has been highly reliable since deployment. The eCRF was in use during the critical events of the Saturn Orbit Insertion and the Huygens probe release. The tool is flexible enough to accommodate many different platforms, continues to evolve to meet user needs, and provides increased efficiency and cost savings over use of the paper CRF. This paper discusses the evolution of the eCRF software from conception through deployment, challenges encountered and lessons learned, as well as current status and refinements.

I. Overview

THE Ground Data System for the Cassini Saturn Mission is a distributed environment that supports the efforts of operations teams at 10 US and 2 European sites (see Figure 1 below). The Mission Controller, also known as the Ace, in the Mission Support and Services Office (MSSO) at the Jet Propulsion Laboratory (JPL), is responsible for radiating command (CMD) files to the spacecraft, which were built by the Sequence Virtual Team (SVT) based on information obtained from various operations teams.

[*] Cassini MSSO Systems Engineer, 4800 Oak Grove Drive/MS 230-301.
[†] Cassini MSSO Flight Operations Lead, 4800 Oak Grove Drive/MS 230-301and AIAA Senior Member.
[‡] Cassini Uplink Operations Manager, 4800 Oak Grove Drive/MS 230-101.
[§] Cassini MSSO Flight Operations, 4800 Oak Grove Drive/MS 230-301.



RADAR - Radar	Pasadena, CA
RSS - Radio Science	Pasadena, CA
VIMS - Visible & Infrared Mapping Spectrometer	Tuscon, AZ
ISS - Imaging Science Subsystem	Boulder, CO
UVIS - Ultraviolet Imaging Spectrograph	Boulder, CO
CAPS - Cassini Plasma Spectrometer	San Antonio, TX
INMS - Ion & Neutral Mass Spectrometer	San Antonio, TX
RPWS - Radio & Plasma Wave Science	Iowa City, IA
CIRS - Composite Infrared Spectrometer	Greenbelt, MD
MIMI - Magnetospheric Imaging Instrument	Laurel, MD
MAG - Magnetometer	London, UK
CDA - Cosmic Dust Analyzer	Heidelberg, GR

*Prior to the release of the Huygens Probe in 2004, the Cassini Ground Data System also included:

ESTEC - European Space Research & Technology Center	Noordwijk, NL
HPOC - Huygens Probe Operations Centre	Darmstadt, GR

Figure 1. Cassini Ground Data System

Following the launch of Cassini-Huygens in 1997, information needed for processing and radiating each CMD file was assembled on a paper Command Request Form (CRF), initiated within JPL or faxed in from a remote site. These CRFs were carried by hand from office to office for additional information, validation, and approval. Following approval of the CRF at the Command Approval Meeting (CAM) and radiation of the CMD files to the spacecraft by the Ace, the paper CRF forms were stored in a binder.

As the Saturn Mission progressed and required an increasing number of CMD file transmissions, MSSO explored the idea of automating the command request process by using a Web-based electronic CRF (eCRF). The principal benefits of the eCRF were the increased efficiency and convenience of electronic generation, tracking and storage of information regarding CMD files; and, the convenience of allowing authorized team members from operations sites worldwide to interactively view and/or contribute to a command request in progress, or search for information regarding previously stored command requests, without the need for having the paper forms in front of them.

The development of the eCRF required rigorous configuration management, extensive research and requirements gathering from all Cassini offices, and user testing and feedback. The security of eCRF access via the Web was of paramount concern, requiring proper authentication and authorization for all users involved in the command request process, from initiation through approval (electronic "signing" of the eCRF) and uplink to spacecraft. Additional features not possible with a paper form, such as automatic insertion of selected data, error checking, highlighting of required data fields, and automatic email notification, were incorporated. The result was a tool that gradually earned confidence from an ever-wary user community, and has been highly reliable since deployment.

The eCRF was deployed in September, 2003, and was in use during the critical events of the Saturn Orbit Insertion and the Huygens probe release. The tool is flexible enough to accommodate many different platforms, continues to evolve to meet user needs, and provides increased efficiency and accuracy over the use of the paper CRF.

II. The Cassini Command Request Form

A. Purpose

The purpose of the Cassini Command Request Form, whether paper (CRF) or electronic (eCRF), is to identify a particular file of CMD data and to convey all the essential information about that file to all parties involved and eventually to the Ace who will uplink the file to the spacecraft. Before placing the CMD file on the Command System and transmitting the CMD file to the spacecraft, the Ace makes certain that the proposed uplink conforms to all the instructions and constraints stipulated on the CRF, and that the form is properly authorized.

A subset of CMD data files, identified as "Library Commands", are files which are used repeatedly or are kept available for contingencies during specific time periods. Another paper form, an abbreviated form of the CRF called a library "Ticket", was used to authorize each uplink of such files based on a previously authorized CRF.

Note that the CRF does not actually manipulate the CMD data, that is, the bits that are radiated to the spacecraft. The CRF only contains information about the CMD data.

B. Users

The Cassini team members involved in the CRF process are: 1) Command initiators – comprised of Spacecraft Operations (SCO), Navigation (NAV), Uplink Operations (ULO), and Instrument Operations (IO) team members; 2) Command file builders – comprised of ULO team members; 3) Mission Controllers (Aces) – comprised of MSSO team members; and, 4) CAM chairpersons – comprised of ULO team members.

All of the above team members are part of the Cassini SVT, led by a ULO team member known as the Cassini SVT Lead (SVTL), equivalent to "Flight Director" in other missions.

III. Automating the Command Request Form

A. Goals

The goals in developing the eCRF application were: 1) to provide increased efficiency and cost savings over the paper CRF by automating manual tasks; 2) to provide automatic notification to users of required action and command file status; 3) to provide automatic, reliable and consistent backup of each eCRF; 4) to provide immediate access to an eCRF in progress ("view") or a stored eCRF ("search") by all Cassini team members, and to generate reports based on user defined parameters; 5) to provide the automatic insertion of selected data; 6) to eliminate the possibility of certain human errors by value checking selected data; 7) to offer off-site teams (Distributed Operations) the convenience of initiating a command without the need to fax a paper form, and to interactively view and/or contribute to a command request in real time during the CAM; 8) to provide for the convenient re-use of previously authorized eCRFs for the handling of Library Commands; 9) to distribute CRF information, when desired, without the need to make copies of a paper form; 10) to obviate the need for large volumes of stored paper CRF files and library Tickets.

MSSO envisioned that the achievement of these goals would streamline the Cassini command request process, ensure the accuracy of command request data, and provide workload relief to the SVT members.

B. Challenges

The primary challenges in developing the eCRF application were: 1) to define and duplicate the sequence of processes involved in the paper CRF, its initiation from, and progress through, the various Cassini offices; 2) to define and duplicate the sequence of processes involved in the library Ticket; 3) to duplicate all the information gathering possibilities of the paper CRF and library Ticket; 4) to provide a means of accessing, assembling and validating such mission critical information via the Web (including a secure means of validating and approving the eCRF by means of electronic "signatures") in compliance with the strictest of security requirements as stipulated by JPL and International Traffic in Arms Regulations (ITAR).

All of these challenges needed to be met while conforming to Cassini's rigorous software configuration management requirements.

C. Requirements

To address the above goals and challenges, MSSO performed extensive research on automated tracking systems and online security, and extensive requirements gathering among all Cassini offices.

The requirements gathering process included: 1) questionnaires delivered to key flight operations team members in each office; 2) personal interviews with current frequent users and expected new users; 3) group meetings with all involved offices; and, 4) cross checking of information obtained by questionnaires, interviews and meetings with information contained on a large number of previously approved CRF forms.

Of particular importance was to precisely define the processes used by each office with regard to command requests, to define the interactions between the offices, and to define which personnel were allowed the authority to take certain required actions. Interviews with the key Cassini team members were vital in order to assemble the necessary information regarding inter-office and intra-office command request procedures, and to define the responsibilities, privileges and authorities of the various team members involved in the process. Given the complexity and variety of command requests, there were many possibilities for the contribution, revision and approval of information on the CRF. Therefore, information was obtained not only from current and expected CRF users, but also by close examination of information contained on previously completed and approved CRFs.

After compiling all the information obtained by research, interviews and group meetings, MSSO arrived at a list of 34 basic requirements and desired features which met the goals and challenges of the eCRF, and began to develop the application.

IV. Developing the Electronic Command Request Form

A. Design

1. User Interface

Appearance: Early in the requirements gathering process, each office expressed the desire that the eCRF mirror as closely as possible the paper CRF. An eCRF was designed in Portable Document Format (PDF) format which allowed exact duplication of the front and back sides of the paper CRF. The only differences in the visual appearance were the addition of pull-down menus and push buttons for certain actions, the addition of a Status ("Initiate", "Radiate", "Uplinked", etc.) data field, and the automation of a data field entitled "Log Number", which was a unique tracking number assigned each time a user initiated and Submitted an eCRF.

Data Input: The PDF eCRF offered identical data input capabilities to the paper CRF, with the addition of user conveniences such as automatic data insertion in selected fields, cutting and pasting data from one eCRF to another, highlighted (not grayed out) data fields to indicate allowed data entry, the titles of required data fields indicated in red color, and a dialog box for Cassini team members to communicate with each other.

2. Accessibility, Authentication and Authorization

The eCRF application is accessible via a Web browser, allowing authenticated users to view the eCRF from their own desktop computers. Any Cassini team member with a valid Cassini Web user account may view the eCRF and search the eCRF database; however, only properly authorized users are allowed to interact with a command request in progress or initiate an eCRF.

Security is of the utmost concern when providing Web access to a mission critical process such as eCRF, and Cassini security protocols follow the strict requirements of ITAR.

Access Control List (ACL): An ACL was established containing the username of each Cassini flight team member involved in the command request process. The ACL defines how specific users are allowed to input, validate or approve information on the eCRF. Once a team member logs in to eCRF under his/her username, only the allowed interactions are possible.

Electronic “Signatures”: An important part of the paper CRF process was the requirement that authorized personnel review, validate, and indicate approval of specific command request information by “signing off” in the proper space on the form. SVT Leads, Systems Leads, Subsystems Leads, Instrument Team Leads and Aces are required to approve and “sign” the appropriate area of the eCRF at one time or another prior to or during CAM. To accommodate this requirement in an electronic format, the eCRF provides a means for electronic signatures.

If a user is required to sign a particular signature field on the eCRF, then that user selects the appropriate signature field, and a new pop-up authentication window appears, prompting the user to enter his/her username and password. Upon successful entry, the username appears in the signature field. The ACL determines whether or not an authenticated user is authorized to enter a signature on the eCRF, and which signature field that user is authorized to “sign” for. For example, the "TELECOM" signature field on the form may be signed only by a user who has been authorized as a "Telecom" member.

Note that even though users have been authenticated upon log-in to the Cassini Web server, if the same user selects a signature field, then a new authentication process must occur for the purpose of “signing” the form. This “double check” of user identity ensures that only properly authenticated and authorized users are allowed to sign the designated signature fields on the eCRF.

3. *Process*

The step-by-step processes for a New Command Request and a Library Command were precisely defined. MSSO grouped these processes into Phases.

Figures 2 and 3 show the Phases associated with a New Command Request Process and a Library Command Request Process, and the automatic email notifications that are sent to specific user(s) indicating required actions upon the completion of each Phase.

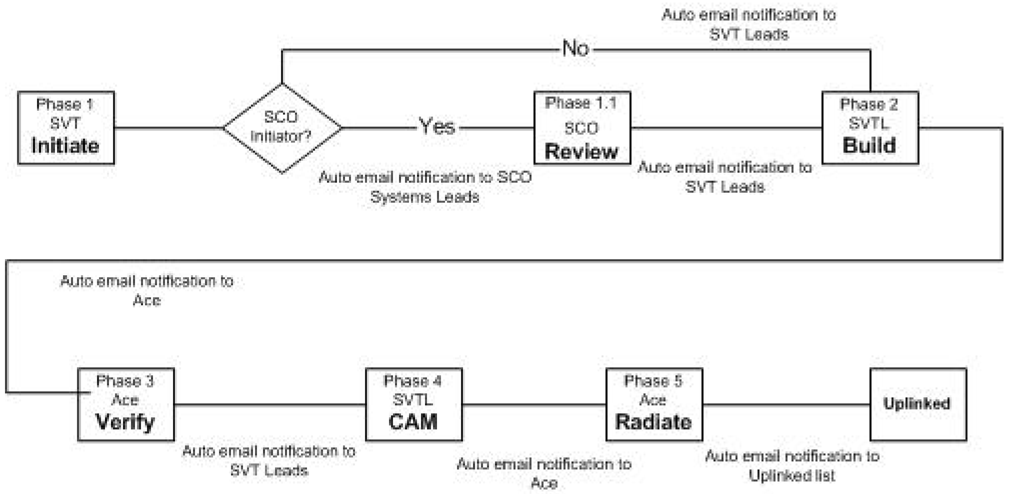

Auto email notification to SVT Leads
No
Phase 1 SVT Initiate
SCO Initiator?
Yes
Phase 1.1 SCO Review
Phase 2 SVTL Build
Auto email notification to SCO Systems Leads
Auto email notification to SVT Leads
Auto email notification to Ace
Phase 3 Ace Verify
Phase 4 SVTL CAM
Phase 5 Ace Radiate
Uplinked
Auto email notification to SVT Leads
Auto email notification to Ace
Auto email notification to Uplinked list

Figure 2. New Command Request Process

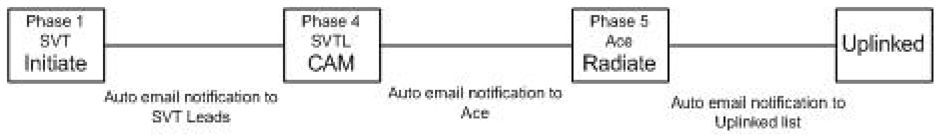

Phase 1
SVT
Initiate
Phase 4
SVTL
CAM
Phase 5
Ace
Radiate
Uplinked
Auto email notification to SVT Leads
Auto email notification to Ace
Auto email notification to Uplinked list

Figure 3. Pre-approved Library Command Request Process

4. System Architecture

Figure 4 shows the eCRF System Architecture. The eCRF application interfaces with the Lightweight Directory Access Protocol (LDAP) server which provides user authentication, and the Cassini Send Mail server which provides email notification. A Sybase database server is allocated for the storage of the eCRF and any related files. All servers are maintained by MSSO.

No new hardware purchases were required to accommodate the eCRF application.

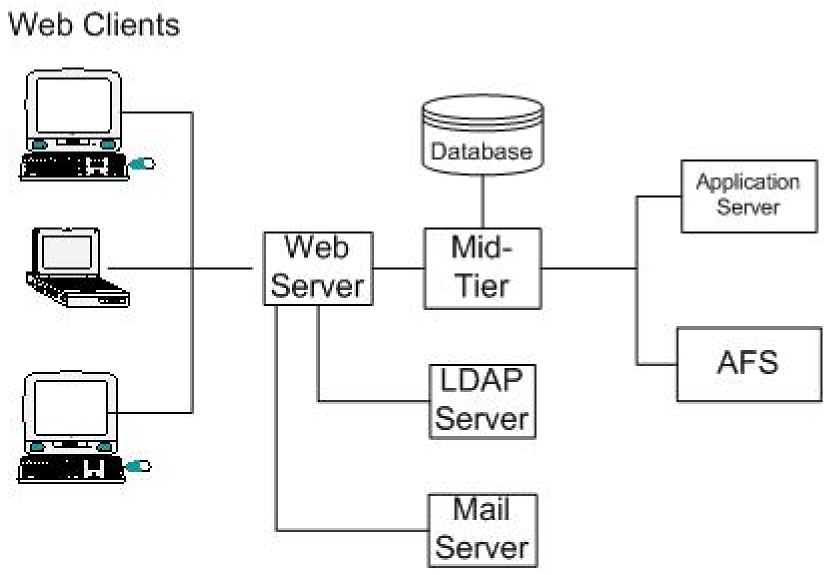


Figure 4. eCRF System Architecture

5. Backup

Automatic full backup of the eCRF database occurs daily. The designated workaround in event of system failure is the use of a paper CRF, exactly as used prior to the deployment of eCRF. If the initiation of a command request were to be required while the eCRF system was down, then that request would be initiated on a paper CRF, and the data would be re-entered on an eCRF by the Ace as soon as the system was back on line.

Failover: MSSO maintains backup servers in the event of primary server failure.

When enough of the initial requirements were met to get the eCRF on line with essential command processes, the tool was piloted in the test environment.

B. **Piloting**

1. Testing the Preliminary Prototype

A preliminary test was performed in March 2003 with a few key users who accessed the eCRF in the Verification and Validation (VNV) environment using a variety of Web browsers. This idea behind this test was to evaluate the appearance, accessibility and usability of the PDF eCRF.

The preliminary test results indicated that the PDF format was incompatible with some of the Web browsers in use, and was prohibitively slow for some other browsers, even though they were properly configured and compatible. Since it was essential for the eCRF to be properly viewable across all platforms used by Cassini team members, the decision was made to sacrifice the requirement that the eCRF precisely mirror the paper form in favor of the requirement that the eCRF be accessible and usable by all team members, and resemble the paper form as closely as possible.

The solution was to provide the eCRF in HyperText Markup Language (HTML) format, with the layout of data fields resembling the paper form, including separating the single HTML form into two areas labeled “Front” and “Back”, with the top part corresponding to the front side of the paper CRF and the bottom part corresponding to the back. Another preliminary test was performed using this HTML eCRF, with satisfactory results. At this point, a large-scale user test was in order.

2. Testing the Revised Prototype

The Test Plan for the eCRF application included: 1) functional testing to ensure that the individual components of the software performed as specified; 2) User Acceptance Testing (UAT) to ensure that the software performs as desired by the user; and, 3) failover testing to ensure adequate back-up for the eCRF system.

Functional testing was performed by MSSO testers, using Solaris, PC and Mac platforms.

UAT was performed by Cassini team members using PC, Mac, Solaris, and Linux platforms.

All known CRF users were contacted by email with an announcement of the eCRF, and a request to view and test the software and respond to an attached questionnaire. An eCRF User Guide was made available.

In addition, MSSO identified the 38 most frequent command initiators (by reviewing 400 previous commands spanning an 18 month period), and 24 expected new frequent users (by personal inquiry). These frequent users were contacted by phone, in addition to the email notice, with a request to test the application.

MSSO offered a personal hands-on walk through of the eCRF for all Cassini team members.

Three fictitious CAMs were held, which were attended by SVT Initiators, SVT Leads, Aces and remote instrument team members. During these fictitious CAMs, every Phase of the eCRF process was tested, from initiation through pending radiation, which included secure login and the addition of electronic signatures, and simultaneous viewing and updating of the eCRF (using the Refresh feature) by Distributed Operations. These fictitious CAMs allowed end-to-end data flow to be tested, and real time command request operations mimicked following the specifically defined command request process.

Testing continued across a three month period, and substantial user feedback was received. This feedback led to the specification of 32 additional requirements for the eCRF. Many of these new requirements were cosmetic in nature, but some additional data fields and automated functions were desired for increased user convenience. MSSO was able to implement 29 of these new requirements in the first release of eCRF.

V. Deploying the Electronic Command Request Form

The eCRFv1 was deployed in September 2003. A one month “soak” period was designated, during which any command requests already in progress using the paper CRF were completed using the paper form, and any new requests could be initiated using either the paper form or the eCRF. Following the soak period, new requests were initiated by using the eCRF only.

Following its introduction in the Operations (Ops) environment, the eCRF gradually earned the confidence of an ever-wary user community, being highly reliable since deployment.

VI. Security and Reliability

The eCRF application has been highly secure and reliable since deployment. To give an idea of the demand placed on the system, consider that during the nearly three years since the deployment of eCRF in September 2003, a total of 1060 command requests have been initiated, processed and stored using the electronic form. Only once did the SVT need to revert to the backup paper CRF to initiate a command due to a network problem, and on that instance the eCRF was back online in time to be used for the CMD uplink. The eCRF has been 100% reliable during CMD file uplink.

VII. Enhancements to eCRFv1

The eCRFv1 software was placed under Cassini configuration control upon delivery. In accordance with Cassini's rigorous configuration management requirements, if a user desires a new feature to be incorporated in eCRF, a formal Engineering Change Request must be submitted and reviewed by authorized members of the Cassini team. Approval of such a request then drives the development, delivery, and installation of a new version of eCRF.

The eCRF application has continued to evolve to meet user needs, and several new versions have been deployed.

1) eCRFv1.1

The eCRFv1.1 provided an interface for Automated Sequence Processor (ASP) commands, a set of CMD files initiated and built by the instrument teams.

ASP commands were designed to be used by the Instrument Teams during the Cassini Tour Phase which began in June, 2004. ASP automates a generation of non-interactive, instrument internal real-time commands which control the various science data collecting instruments on board the spacecraft. Since the ASP process only allows the transmission of a pre-approved list of commands, and so that the instrument teams may operate the instruments autonomously, ASP commands do not require a CAM.

Since the eCRF software code is of a modular design, it was a relatively simple task to implement a new command request process to accommodate ASP commands. Figure 5 shows the ASP Command Request Process.

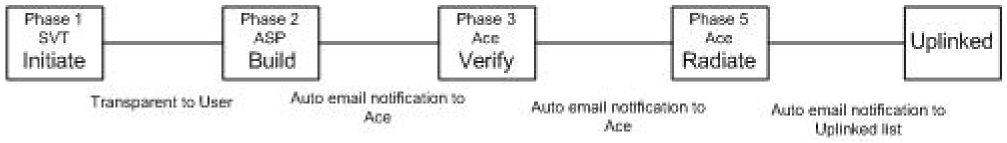


Figure 5. ASP Command Request Process

2) eCRFv1.2

The eCRFv1.2 provided an interface for Maneuver Automation Software (MAS) commands, a subset of CMD files built by the NAV and SCO teams for conducting Orbit Trim Maneuvers (OTMs). Cassini depends on carefully targeted gravity-assist Titan flybys each Saturn orbit in order to remain on the Saturn system tour, and three OTMs are executed for each orbit: the first OTM about 3 days after the targeted encounter, the second near apoapsis, and the third maneuver 3 days before the next targeted encounter.

Since the NAV and SCO teams use MAS to automatically generate OTM data, these MAS commands do not need to proceed through the Initiate and Build Phases of eCRF. Thus, the MAS interface implemented in eCRFv1.2 automatically checks for MAS generated data, automatically populates this data into a new eCRF, and automatically notifies the Ace that the action of Phase 3 Ace Verify is required.

Again, the modular design of the eCRF software code provided for the relatively simple implementation of a new command request process to accommodate MAS commands. Figure 6 shows the MAS Command Request Process.

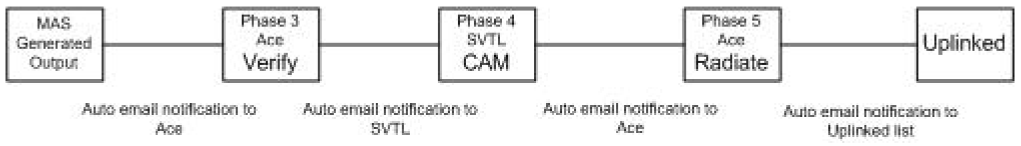

MAS Generated Output
Phase 3 Ace Verify
Phase 4 SVTL CAM
Phase 5 Ace Radiate
Uplinked
Auto email notification to Ace
Auto email notification to SVTL
Auto email notification to Ace
Auto email notification to Uplinked list

Figure 6. MAS Command Request Process

3) eCRFv1.3

The eCRFv1.3 provided additional user conveniences, such as automatically emailing an electronic copy of the eCRF approved at CAM to the Ace to be used as a backup in the event that the eCRF system is down when a CMD file needs to be radiated.

4) eCRFv1.4

The eCRFv1.4 provided additional enhancements to the New Command Request Process and Library Command Request Process, by allowing SVT Leads to "send back" an eCRF from the CAM Phase to SVTL Build Phase.

5) eCRFv1.4.1

The eCRFv1.4.1 provided bug fixes that resulted from a Web server software upgrade. When MSSO upgraded its Web servers from iPlanetv6.0SP5 to v6.1SP4, the "Cancel" button on the eCRF form did not function properly on some browsers. An adjustment in the java code was required.

VIII. Current Status of eCRF

The eCRFv1.4.1 is currently in use by Cassini. As noted, this application has been used to initiate over 1060 command requests since deployment in September 2003, has been highly reliable since deployment, and has provided increased efficiency and accuracy over the use of the paper form.

IX. Lessons Learned

We would like to share some of our experiences in developing and deploying the eCRF:

1) Obtain buy-in from senior management early in the planning process.

This is especially important in the development of an intra-office tool, as it helps to ensure that the various offices are as attentive as possible to the needs of the design team in obtaining information and feedback, helps in resolving potentially conflicting requirements, helps unify the individual objectives of the various offices, and, in short, helps ensure the success of the application from conception through deployment.

2) Help the customers understand their needs.

Where a manual process is in place, all the details of the procedure may be well known but not specifically documented. Questions aimed at understanding the manual process in detail, and how these processes might translate into an automated system, can help the users define exactly what their requirements are in order for the automated system to function as expected.

3) Piloting is important.

MSSO piloted a preliminary PDF eCRF prototype to determine whether the PDF format would function properly in the Cassini user environment. A preliminary test conducted by a small group of users revealed that the PDF form did not perform as desired, and the eCRF was re-designed in HTML format. The decision was made to forego the initial requirement of mimicking the paper CRF exactly in lieu of a format that functioned as required, yet still resembled the paper form closely enough to provide a desired level of familiarity for the user. It was beneficial to make this decision early in the development process.

Once the HTML eCRF was implemented, MSSO piloted this revised prototype to conduct comprehensive product testing. During piloting, new requirements were identified and implemented into the prototype for further testing.

4) Design to accommodate change – use a modular and flexible design.

In replacing a complex series of manual processes, all the requirements desired by a large number of potential users may not be identified at the outset. So, in the interest of moving forward, use a modular design that meets the basic requirements, but is flexible enough to accommodate changes as new requirements are identified. Get a basic prototype operating in test environment, and then encourage users to provide feedback which can be implemented by the design team.

For instance, during user testing of the HTML eCRF prototype, an additional Phase in the command request process (SCO Review) was requested. Since the eCRF code was of a modular design, the addition of a new phase was a relatively simple task. In fact, using the incremental prototyping method described above, 29 other new requirements were implemented during the test phase. Due to the modular design of eCRF, MSSO was able to accommodate these requirements without re-designing the code.

5) Design configuration files outside of the compiled code.

As much as possible, design elements such as drop-down menus, tables, ACLs, etc., outside of the compiled code. The eCRF configuration files designed outside of the compiled code, which allowed MSSO SAs to modify these types of files without affecting the compiled code.

6) Actively solicit user feedback during the test phase.

Given the work load of team members involved in a project such as Cassini, feedback regarding the user testing of a new application may not be received by the development team as promptly as may be desired based solely on email notifications and questionnaires. Personal requests and active walk throughs are more effective.

In the case of eCRF, MSSO took the initiative to personally contact key users and walk them through the test, as well as offering a hands-on open walk through to all Cassini users at lunch time. Three “fictitious CAMs” were held, during which every Phase of the eCRF process was tested. Vital feedback was obtained using these active methods.

7) Begin with the familiar.

When automating a manual process that has been in place for some time, it is important to maintain as much similarity as possible between the new process and the old. This way the benefits of the new process are obvious right away; then, during prototype testing, more innovative elements can be introduced.

8) Continue moving forward as requirements are defined.

It is important to maintain momentum when developing and delivering a new application that impacts several offices. As potential users become enthusiastic about the idea of automating a previously manual process, and begin to conceive the many possibilities that automation offers, desired requirements may continually be presented to the design team. Rather than wait for the “complete” set of requirements to be defined, get the application up and running with essential processes. Many of the “bells and whistles” can be added later as the application gains widespread use.

In the case of eCRF, two command group interfaces, ASP and MAS, were delayed until the release of eCRFv1.1 and eCRFv1.2, respectively. This gave the MAS and ASP teams additional time to specify their interface requirements and to test the interfaces after their requirements had been implemented. In addition, user conveniences continued to be added to eCRF as time and budget allowed.

9) Conduct additional testing for each Web server upgrade.

When using a Web based application, server software upgrades may result in performance anomalies using some browsers. For example, when MSSO upgraded its Web servers from iPlanetv6.0SP5 to v6.1SP4, the “Cancel” button on the eCRF form did not function properly using some browsers. This was discovered during testing prior to the upgrade. A new release of eCRF, Version 1.4.1, was required to fix this anomaly.

X. Conclusion

The development and use of a Web-based automation tool to replace manual processes offers many advantages. In the case of Cassini’s eCRF application, the automatic distribution of the electronic form from office to office, the ability to search and view the eCRF database at the touch of a button, the ability of Distributed Operations to submit requests electronically rather than by fax, and many other user conveniences, have streamlined the Cassini command request process by reducing the work hours involved. The implementation of automatic data insertion, data value checking, and the automatic double-checking of required data fields has reduced the possibility of human error.

While the development of the eCRF presented many challenges and required consistent effort through development and testing to win the confidence of an ever-wary user community, the results of increased convenience, accuracy, and reduction in workload over the previously used paper CRF has proven to be a welcome enhancement to the Cassini command request process.

The eCRF serves as a pioneering model at JPL for the use of a Web-based automated command request application, and MSSO has shared information with other JPL flight projects planning to develop a similar Web-based tool.

Acknowledgments

The eCRF application was developed by the Cassini Mission Support and Services Office at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the national Aeronautics and Space Administration (NASA). The Cassini Program funded this task.