Automation of Cassini Support Imaging Uplink Command Development

Lisa Ly-Hollins[1], Herbert H. Breneman[2], and Robert Brooks[3]
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109, USA

"Support imaging" is imagery requested by other Cassini science teams to aid in the interpretation of their data. The generation of the spacecraft command sequences for these images is performed by the Cassini Instrument Operations Team. The process initially established for doing this was very labor-intensive, tedious and prone to human error. Team management recognized this process as one that could easily benefit from automation. Team members were tasked to document the existing manual process, develop a plan and strategy to automate the process, implement the plan and strategy, test and validate the new automated process, and deliver the new software tools and documentation to Flight Operations for use during the Cassini extended mission. In addition to the goals of higher efficiency and lower risk in the processing of support imaging requests, an effort was made to maximize adaptability of the process to accommodate uplink procedure changes and the potential addition of new capabilities outside the scope of the initial effort.

Introduction

The team on the Cassini Project tasked with commanding and monitoring the science instruments during flight operations is the Instrument Operations (IO) Team. The team consists of several individuals with expertise in the operations of specific science instruments as well as analysis and interpretation of data they return during the primary mission phase (1997-2008), the project staffed this team such that each instrument was the responsibility of at least one, and usually two members. This provided a backup for members who took vacation or sick leave during the long primary operations period that included launch, cruise to Saturn, and Saturn orbital operations. This staffing profile was consistent with the observational complexity and operational demands of a flagship mission to the second largest planet in our Solar System and its cadre of moons.

Upon completion of the primary mission for Cassini, the project was granted an extended mission (XM), running until 2010, during which the spacecraft would continue its observations of Saturn and its numerous moons. It was deemed necessary for the staffing level for XM to continue at its primary mission level to assure the safety of the spacecraft and to support the aggressive science campaigns planned for XM.

However, staffing for an "extended-extended" mission (XXM) following completion of XM was not expected to exceed fifty percent of nominal staffing levels. This reason alone would have called for more efficient operational procedures. Compounded with the fact that several IO team members were going to transition to other projects after XM, team management recognized that a process improvement effort must take place. One component of this process improvement was the use of process automation. Team members were tasked to document the existing manual process, develop a plan and strategy to automate the process, implement the plan and strategy, test and validate the new automated process, and deliver the new software tools and documentation to Flight Operations for use during the Cassini extended mission.

A cardinal rule of automation is to system engineer the process completely prior to beginning any actual development of scripts or software. This was accomplished by holding regularly scheduled IO team meetings at which the processes to be automated were analyzed in great detail. This included analysis of all external interfaces used by the process. Once this system engineering was completed, a plan for accomplishing the automation was created and the task of building the required scripts and software was begun.

---

[1] Software Engineer, Instruments and Science Data Systems, 4800 Oak Grove Drive, Pasadena, CA 91109/230-250.
[2] Software Engineer, Instruments and Science Data Systems, 4800 Oak Grove Drive, Pasadena, CA 91109/230-250.
[3] Software Engineer, Instruments and Science Data Systems, 4800 Oak Grove Drive, Pasadena, CA 91109/230-250.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Until copyright transferred to publisher, (c) 2010 California Institute of Technology. Government sponsorship acknowledged.

The goals of automation for XM were defined as follows.

- Retain resident expert knowledge base in those expert people (people will always be an integral component of such complex and critical processes)
- Capture processes and procedures in automation software (mostly scripts) where and when appropriate.
- Enable standardization of uplink tasks
- Develop interfaces that enable inter-process communication (automated tasks triggering and talking to other automated tasks)
- Maximize the efficiency, responsiveness, accuracy, and availability of the uplink processes and procedures
- Minimize or eliminate manual processing and perform same tasks with automated tasks where and when appropriate.
- Reduce time and workforce needed to perform the uplink tasks where and when appropriate.
- Enable reduced staffing operations by flight team if extended mission funding is reduced from primary mission flight operations funding levels
- Enable rapid response time to anomalous spacecraft behavior

With these goals clearly in mind, the IO team began the analysis of the team's internal procedures to identify those that could benefit from automation and standardization. The team identified several processes and a subset of these was chosen for implementation.

One of these processes was the Support Imaging Generation Process. This process was performed manually during primary mission operations. It was considered wise to do so because the procedure was evolving as flight operations progressed, and performing it manually assured that errors would be caught by the team members performing the task and corrected prior to uplink of commands.

However, after several years of using the process manually, it was considered mature enough to be automated. This automation would use the existing manual process as a model for the development of scripts that would perform the same functions as a human user but do so at much greater speed and without the risk of user input errors. What follows is a detailed description of the evolution of the Cassini Support Imaging Generation Process from a manually performed process to an automated process that permits the team to reduce staffing to the required levels during XXM.

The Uplink Process

Support images (SI) are those images taken by the Imaging Science Subsystem (ISS) cameras, which are requested by other Cassini science instrument teams (typically CIRS, UVIS or VIMS) to aid in the interpretation of their own data. This is as distinct from ISS images taken by the ISS science team for their inherent science value. Early in the Cassini mission, it was decided that when SI was requested for an observation in which ISS was already participating for science reasons, either because it was the lead or "prime" team (in control of the pointing) or just "riding along" with another Team's observation, then the SI would be implemented by the ISS Team at their remote site and integrated into their overall science imaging plan. But when SI was requested for an observation in which ISS was not otherwise involved, then the SI would be implemented by the Cassini Instrument Operations (IO) Team at JPL. This policy has been maintained throughout the tour to date.

The ISS Support Imaging Engineer's role is to take the SI requests from the science team and process them into a standard commanding format which the uplink ground data system is able to understand and interpret into command sequences by which the spacecraft and ISS instrument are controlled.

The uplink commanding for the Cassini instruments is specified through two types of standard human-readable data files. The Spacecraft Activity Sequence File (SASF) is a plain text file specifying, in the case of ISS, the absolute timing of all of the ISS camera system "triggers" (initiation of a set of images) in one uplink sequence load, which typically operates the spacecraft for a period of about five weeks. The specific camera settings for these images are contained in the Instrument Operations Interface (IOI) files. These are plain-text files in keyword=value format, specifying the exposure times, filters, gain states, data compression modes and other camera settings, and the

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology,
under a contract with the National Aeronautics and Space Administration. Until copyright transferred to publisher,
(c) 2010 California Institute of Technology. Government sponsorship acknowledged.

relative timing of the images in the IOI file. One IOI corresponds to one 'trigger' of the camera system and contains the imaging taking place during one 'tracking period' of an observation. Depending on its pointing design, an observation may contain one or more tracking periods. A typical sequence load may contain roughly 15 observations containing support imaging and perhaps about 50 SI-containing tracking periods in total, resulting in 50 SI IOI files. However, the level of SI activity has varied widely from one sequence to another, up to about twice the above figures.

The main tool for selecting the optimum camera parameters for ISS images and building IOI files is the ISS Pre-commanding Tool (ISSPT[2]). This is an IDL program developed by the ISS science team for planning their own science imaging. Taking as input the observation pointing design and desired timing and filters for the support images, and interacting with the user via a graphical user interface (GUI), ISSPT determines the optimum exposure time and other camera settings for each image, taking into account a mathematical model of the camera characteristics, photometric models of the various potential target bodies, and the lighting and viewing geometry at the time the image is to be taken. It also calculates an estimate of the "data volume" (number of bits that the image requires) based on predicted scene complexity and compression mode specified. When it was decided that the Cassini IO group at JPL would be responsible for building commands for support imaging, a version of ISSPT was made available to IO and installed locally on Sun workstations.

The development of the SI automation proceeded as follows:

1. Document the existing manual process and procedures.
2. Capture this process in a diagram or flowchart.
3. Analyze the diagram and identify simple parts that are practical to automate.
4. Develop the coding to automate these processes, test, and debug as required.
5. Repeat 3 and 4 for other readily automatable procedures.

Support Imaging Command Generation Prior to Automation

Prior to automation, the SI engineer had to build, essentially by hand using existing templates as a guide, the SASF file specifying the absolute timing of all the ISS triggers (IOIs) in the sequence. The primary template available was derived from an XML file produced by the Cassini Information Management System (CIMS[1]), an on-line database of observation design parameters. The XML file had to be converted to standard SASF format and then heavily hand-edited to add multiple triggers for observations containing more than one tracking period, and to replace dummy values with correct absolute times, image counts and trigger numbers, among other parameters. The required tracking information was obtained from the Short Form Output File (SFOF), produced by the science team designing the observation's pointing, and available via FTP. Finally, the SASF had to be run through a standard Project constraint-checking program (SEQGEN) prior to being delivered to the Project; this required first the downloading and hand-updating of an "environment" file to use in the SEQGEN run. This entire process is illustrated in Figure 1. In addition to being very time-consuming, the manual nature of most of these steps made the process prone to human error. The SEQGEN run could detect errors in syntax but not in intent.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology,
under a contract with the National Aeronautics and Space Administration. Until copyright transferred to publisher,
(c) 2010 California Institute of Technology. Government sponsorship acknowledged.

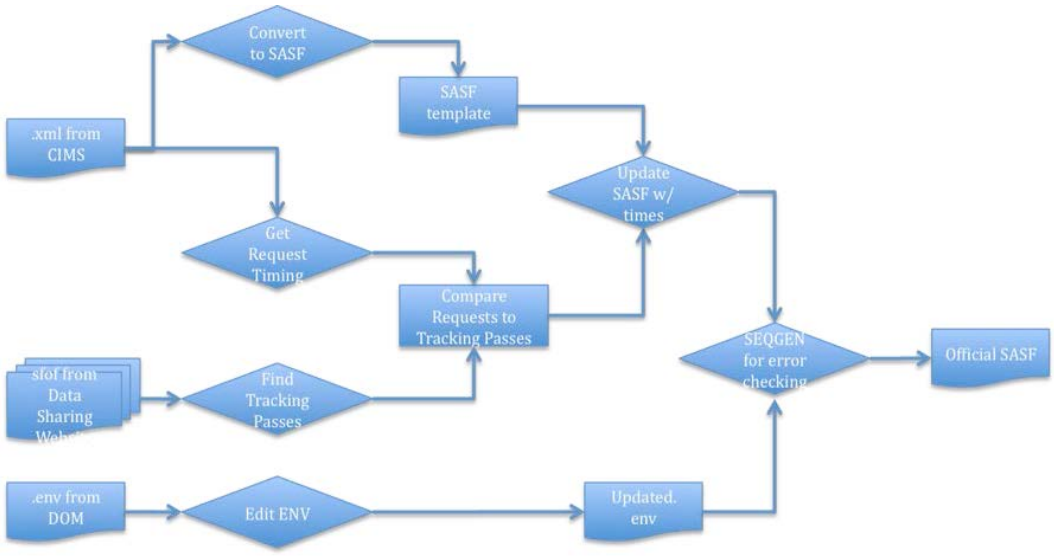


Figure 1: CASSINI Support Imaging SASF Generation Process

Experience early in the mission also showed that using ISSPT to manually build IOIs for support imaging was extremely tedious and labor-intensive. ISSPT can process only one tracking period at a time; hence a separate run of ISSPT is required to generate each IOI. ISSPT required, as input, the files produced by the prime instrument team, which define the pointing for the observation. These files had to be downloaded from an FTP site to the user's local directory on the workstation to be accessible to ISSPT. The operator also had to build by hand, for each tracking period containing support imaging, a "Shutter File" for input to ISSPT. This file specified the relative timing of the images in that period. Working out the optimum timing of the support images in an observation to comply with the requestor's specifications while staying within the tracking periods, which are generally not contiguous, was a non-trivial and time-consuming exercise. Although support imaging consistently used only clear filters and lossy compression, these parameters had to be manually specified in each run of ISSPT. The user also had to check that the total predicted data volume for an observation fell within the amount allocated to it earlier in the sequence planning phase; if this were not the case, re-runs of ISSPT with altered settings for some IOIs might be necessary to solve the problem. The IOI generation process is illustrated in Figure 2. Because of the complexity of the exposure calculations, ISSPT ran quite slowly, with the result that completing the SI IOI generation for even an average sequence easily required many days of work. The entire process from start to finish sometimes had to be repeated several times during the course of the sequence development process, depending on the extent of changes to the design of observations containing SI.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Until copyright transferred to publisher, (c) 2010 California Institute of Technology. Government sponsorship acknowledged.

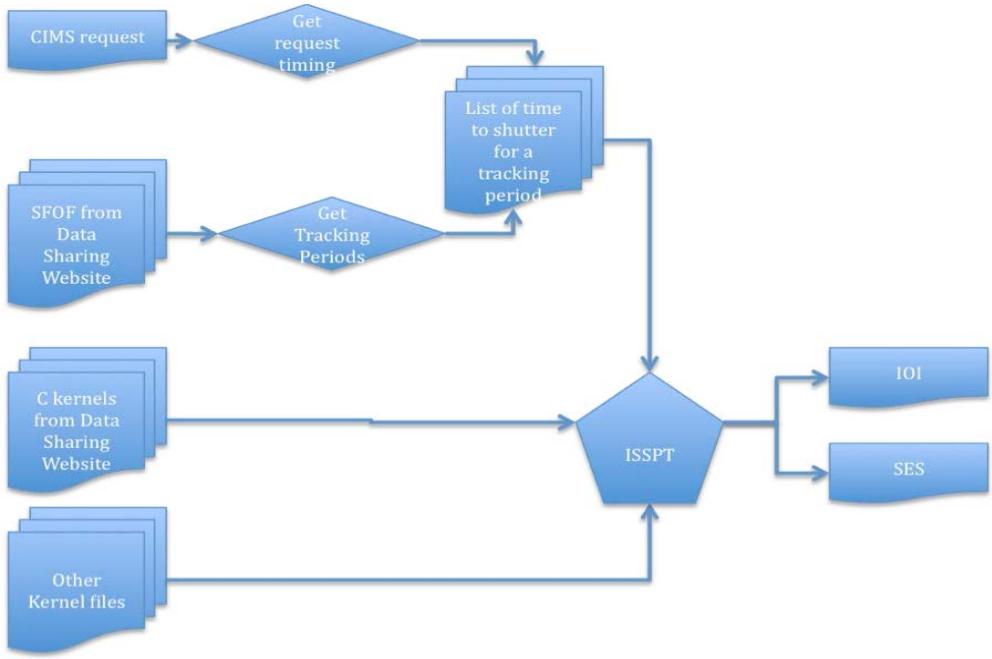


Figure 2: CASSINI Support Imaging IOI Generation Process

It thus became apparent that an effort to automate the process of IOI and SASF generation might result in significant reductions in time, risk, and dollars, particularly for the planned Extended Mission, during which available resources were expected to be significantly reduced. One feature that is apparent from the foregoing description and Figures 1 and 2 is that building the support imaging commands requires bringing together diverse pieces of data from a variety of different sources; much of the complexity of the manual process derives from this fact, and any automation procedure would have to be able to draw on all of these sources of information in a predictable way to produce the desired products.

Breakdown of SI Automation

It was recognized there were many parts of this process that could be automated. The simplest and most easily defined processes were automated first, followed by more and more complex processes, some of which tied into the inputs and outputs of the previously automated processes. In the yellow area in Figure 3 are shown two simple areas where the automation process was begun. First it was necessary to determine the complexity of the automation in

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Until copyright transferred to publisher, (c) 2010 California Institute of Technology. Government sponsorship acknowledged.

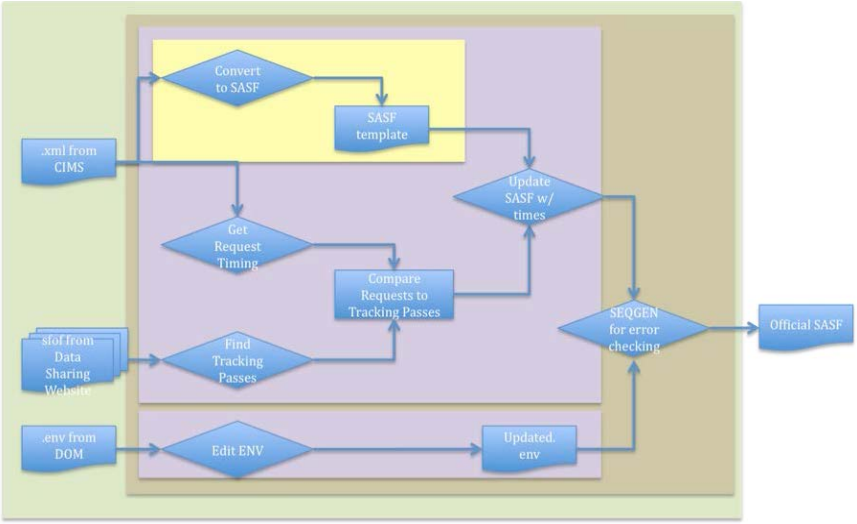


Figure 3: Breakdown of automation process

this area, both within and outside of the system under consideration. For example, it had to be determined whether the right resources were available to solve the problem of automating the conversion of the XML file to an SASF template. If so, then the next question was whether the input (XML) could be converted to a SASF given these resources. In this way, it was determined that the problem in the yellow area is solvable, so the automation could be expanded to other components of the process. The diagram above shows the progression of this expansion. Eventually, it was possible to automate the entire SI process, tying together the SASF and IOI generation (See Figure 4).

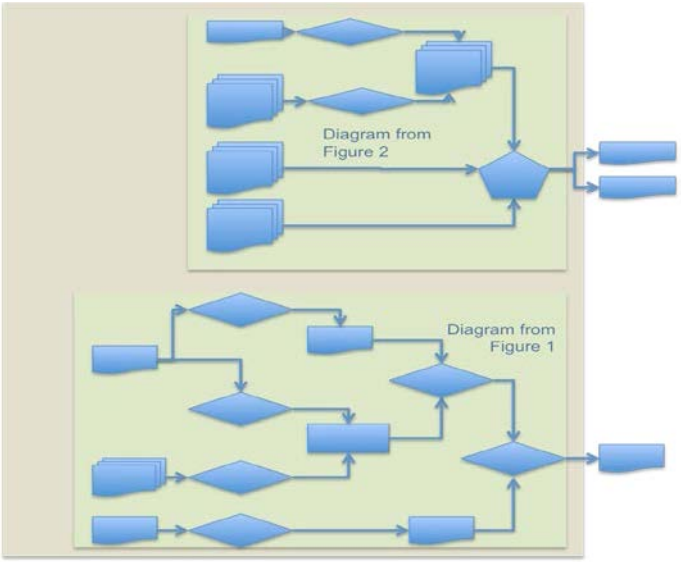


Figure 4: Automated CASSINI Support Imaging Process

Results/Metrics

The table below is a comparison between manual and automated process. The example is taken from 3 requests that consist of 3 tracking periods per request with a total of 60 images.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Until copyright transferred to publisher, (c) 2010 California Institute of Technology. Government sponsorship acknowledged.

Time for Manual Process	% possible error for manual process	Time for Automated Process	% possible error for automation	Description
15 minutes	5%	5 minutes	0%	Gather pointing files
10 minutes	5%	< 1 minute	0%	Gather requester's information
1 hour	10%	< 1 minute	0%	Edit files
1 hour	20%	< 5 minutes	1%	Process 20 images

Working manually, the SI ISS Engineer usually required approximately 15 minutes to gather all pointing input files via FTP download. The engineer had to make sure that the latest version of the pointing files was used, by comparing the file creation dates and times. There was a potential for the engineer to inadvertently choose the wrong (i.e. outdated) pointing files, causing images to be taken improperly or sequencing errors to be issued later by the sequence development software. By comparison, it takes around 5 minutes for the automated process to gather all the pointing files (somewhat dependent on the networking speed between the two systems), and with a high reliability that the correct files are selected.

Another way the engineer has to retrieve input data is by querying the requestor's design information from the observation database. This can take up to 10 minutes for a typical sequence. The automation performs this nearly instantaneously.

The two examples above illustrate that the SI Engineer had to switch between different computer systems in order to gather all the necessary files and input data. Once it was automated, and the required access paths and permissions set up, the different systems were easily accessible by the automation almost instantaneously.

One of the most tedious parts in the manual process was editing input files, temporary template files, etc. This was basically a text editor "cut and paste" operation. This could take an engineer up to 1 hour to complete manually, but with the automation, the editing of each file is trivially easy and almost instantaneous. Errors for manual edits and calculations can occur roughly 10% of the time; again, with the automation in place, the calculations and edits on the files are essentially error-free.

The time it takes to create the IOI files for a single support imaging request is dependent on the number of tracking periods and the number of images being requested. When running this part manually, the engineer must devise a timing plan for the images that meets the requestor's specifications as nearly as possible subject to the breakdown of the observation into tracking periods, between which the target body is not in the field of view and images cannot be shuttered. This process may require several iterations. The timing plan is then put in the form of a text-edited "shutter file" for input to ISSPT. When running ISSPT, many camera settings (filter, compression mode, etc.), as well as output IOI file parameters, must be manually set. Re-runs may be necessary to adjust exposure times, gain states or compression mode to meet data volume constraints. This part of the process could take the engineer around 1 hour to implement per typical request with 3 tracking periods. Under the automation, communication between ISSPT and the automation program isn't through a GUI, but via ISSPT module calls, and the process is much quicker.

In another view, the automated and manual process is compared in Figure 5 and Figure 6.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology,
under a contract with the National Aeronautics and Space Administration. Until copyright transferred to publisher,
(c) 2010 California Institute of Technology. Government sponsorship acknowledged.

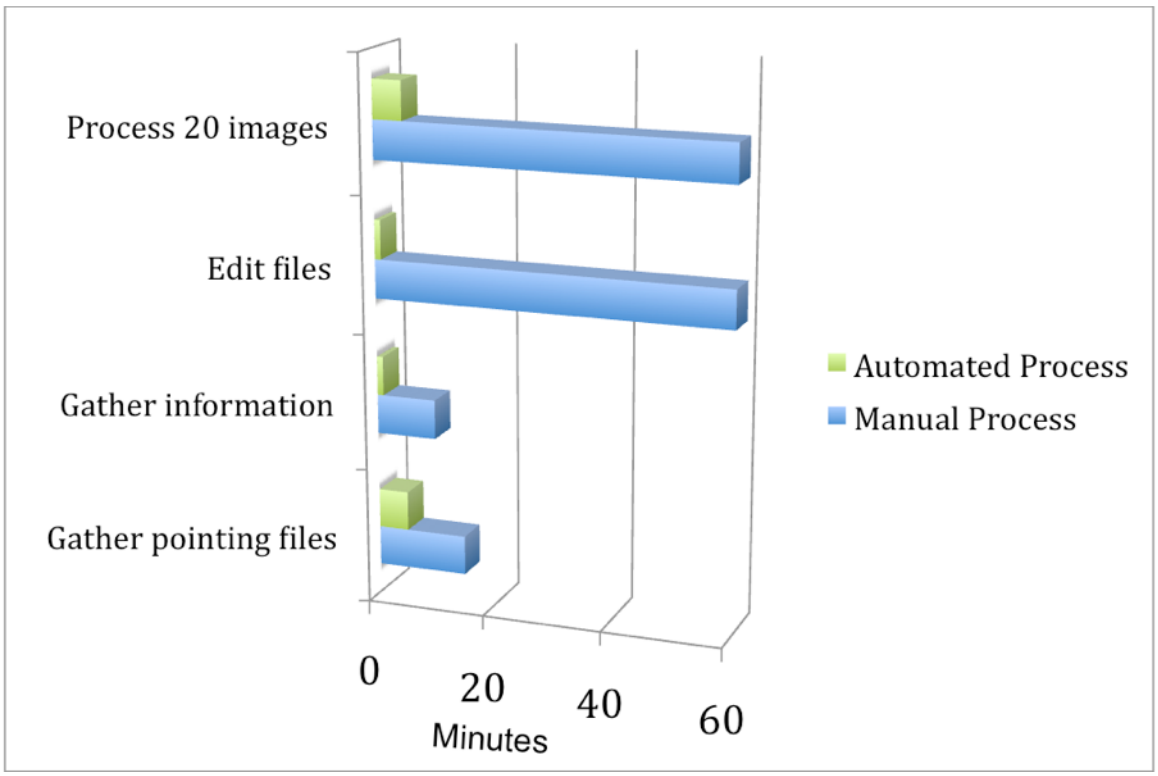


Figure 5

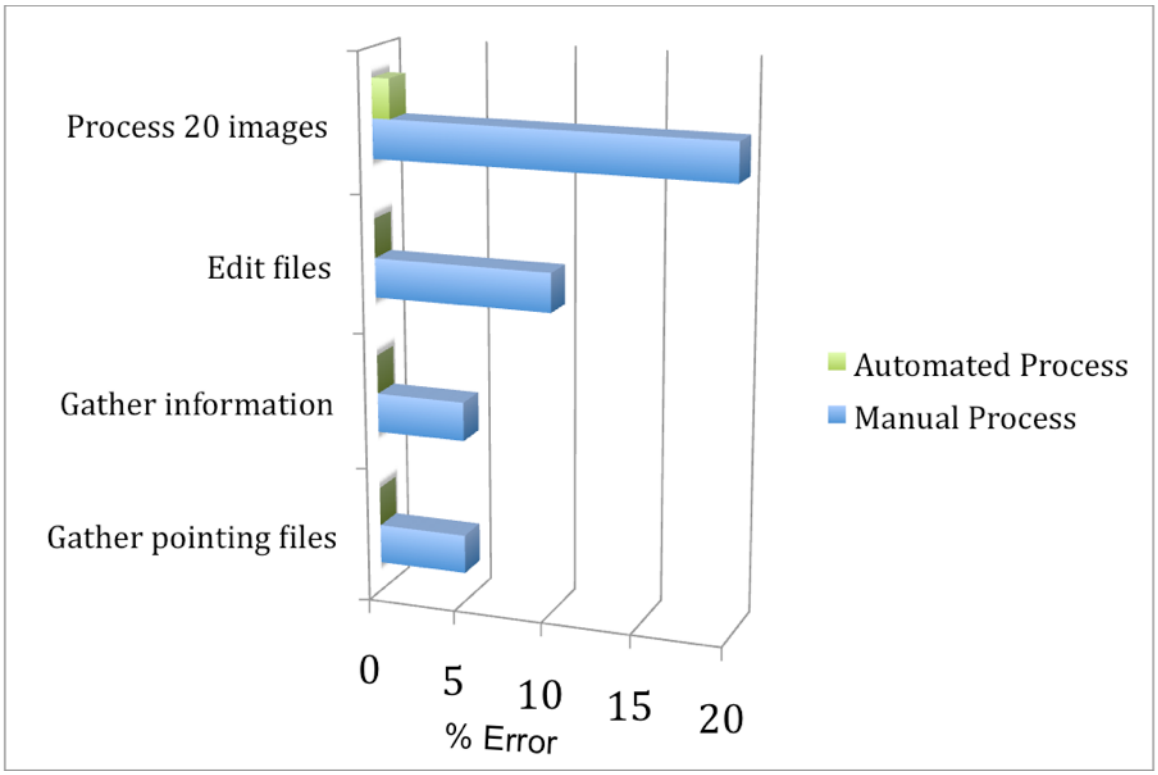


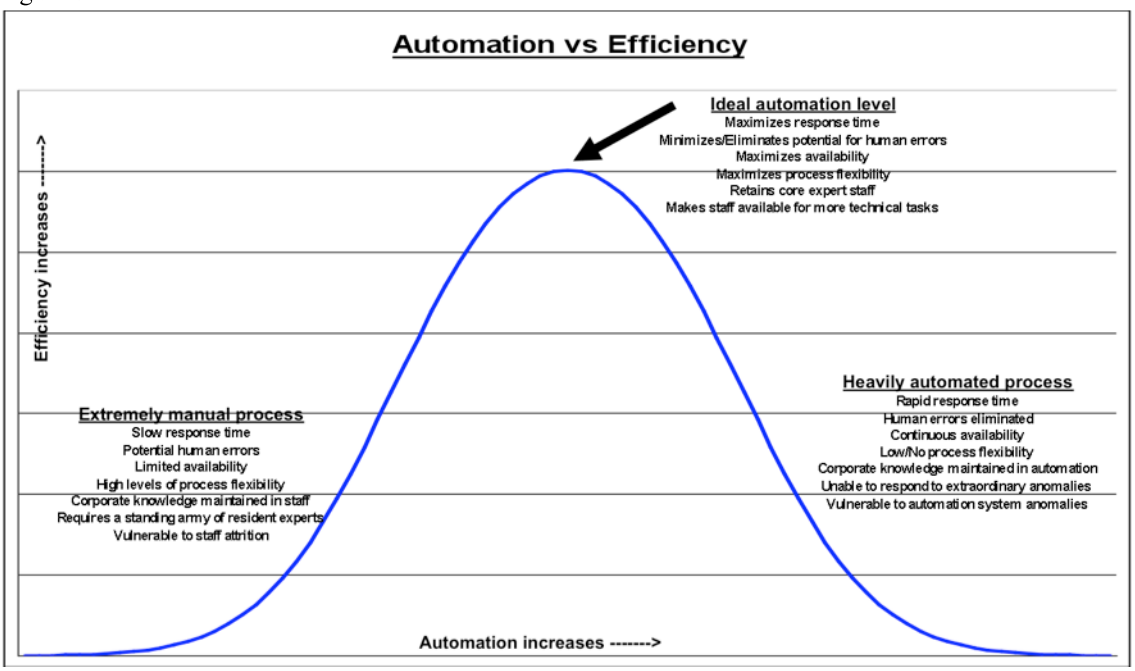
Figure 6

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Until copyright transferred to publisher,
(c) 2010 California Institute of Technology. Government sponsorship acknowledged.

Efficiency

The efficiency of the automation can be summed by a bell curve (see Figure 7). The curve is based on empirical data observed from a Support Imaging Automation example and other automation efforts implemented on several past JPL missions over many years. There are three major levels in the curve: 1) Fully manual process 2) Ideal automation level 3) Heavily automated process.

Figure 7



When the process is completely manual, the process is highly flexible. The advantage of flexibility is that the SI Engineer easily handles anomalies. For example, if pointing files are misplaced, the SI Engineer can search for the files, or if a database name has changed, the SI Engineer can make the necessary name change in the process. This is extremely flexible, but it can be time consuming. On the other hand, there are major costs to a fully manual process.

One of the costs of a fully manual process is the potential for human errors. Human errors can easily occur when the SI Engineer edits files. If caught early, these errors would only waste time, requiring re-edits. If the errors aren't caught, there are risks of images being degraded or lost. Another cost of a fully manual process is a slow response time. The support imaging process described above involves much repetitive work; for instance, creating shutter files. These are just lists of times to shutter images, which can quickly be calculated and written by a computer. Finally, yet another cost of not automating the process is encountered when resident experts leave the project and a new team member has to scramble to gain the same expertise in a short amount of time.

Early in the automation project, the efficacy of automation was positive at all times; the more automation was added, the more efficient and better the process became. But if too many things were automated, the software would end up complicated with too many interdependencies, and keeping track of all the variables for those interdependencies could require more time than actually creating the products. As automation is increased, an optimum efficiency will be reached, and any automation after that point becomes counterproductive. Some of the things that were not automated as part of this effort would have made the system more complicated than it needed to be. They would have added another layer of complexity in the software, without adding enough benefit. The degree of automation was considered 'enough' because it was at or near the top part of the curve.

An ideal automation process is one that sits at the top of the bell curve. There is a point where the efficiency gained from the automation has been maximized, and away that point efficiency is lost, on either side of the curve. In order

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Until copyright transferred to publisher, (c) 2010 California Institute of Technology. Government sponsorship acknowledged.

to know when that point has been reached, the process, inputs, and outputs must be well understood. At the top of the bell curve, all of the below is achieved:

1) Rapid response time
2) Minimum potential for human errors
3) Maximum availability of process execution (not sure what this means)
4) Maximum process flexibility where response to an anomaly is adaptable to changing situations.

When this state has been achieved for a particular automation effort may be hard to pinpoint, but if the process is well understood, this point can be intuitively recognized. In Support Imaging automation, we automated almost everything. One of the things that were not automated was the delivery of the output files to the Project. It was considered desirable for the SI Engineer to look through the products and do a final check and approval before delivering. This would give the SI Engineer the powerful flexibility of saying 'no' or 'yes' to a product.

Testing of Automation Algorithm

The testing and validation of the support imaging automation algorithm was an iterative process. A version of the automation software was installed by the programmer and made available for another IO team member to test. The testing was typically done using past flight imaging sequences as input, since the output from the previous manual runs of ISSPT were already available for comparison. Results of the testing, including any problems encountered, error messages issued, etc., were reported back to the programmer. These were then addressed and a new version of the automation software was released, and the process repeated.

Testing of the automation routine was to verify that:

(a) The process ran to completion, without aborting or hanging up.
(b) All required input files could be accessed and read successfully.
(c) ISSPT was successfully called and executed as required.
(d) All expected output files (SASF and IOIs) were produced and were in the correct format.

The detailed numerical values for the exposure times for individual images were not checked as part of the SI automation testing/validation process. It was assumed that testing done by the ISS science team before releasing ISSPT to their own team members (and to JPL) was adequate to validate the performance of ISSPT. At JPL, only random spot-checks were performed to verify that the exposures seemed reasonable based on past ISSPT runs done without automation.

The first version of the automation routine was delivered for testing in March 2006; testing of subsequent versions has continued intermittently to the present. "S38" (the 38th flight sequence since orbital insertion) was the first sequence used for automation testing.

Many of the problems encountered during the testing were due to hardware and software configuration issues, either due to differences in configuration between the machines used for the coding and the testing, or due to configuration changes that were unexpected by the IO staff or which had unanticipated impacts on the automation algorithm. This included factors such as read-access privileges on files, remote login access to machines storing required data, IDL licensing issues, IDL, ISSPT and workstation O/S version upgrades, password updates, etc. These types of problems, if unresolved, could prevent the automation routine from accessing all of the data files and software required.

Problems were also encountered due to failures of other teams responsible for providing certain data to follow agreed-upon procedures. One example is the pointing data files, which sometimes were delivered late, or to the wrong directory, or in an incorrect format (e.g., a tar bundle rather than individual files). When ISSPT was run manually, the operator could address these types of problems straightforwardly, but encountering these problems during testing of the automation underscored that with automation, it was essential that all teams involved follow agreed-upon procedures.

In addition to the reporting of problems, the automation tester also made note of any potential enhancements to the process that came to light during the testing process. Examples included improvements to the algorithm for spacing SI frames during an observation period, changes to the default names of output files, and several ideas for greater automation of a few remaining manual steps in the process, which would speed the process up even further.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Until copyright transferred to publisher, (c) 2010 California Institute of Technology. Government sponsorship acknowledged.

Many of these proposed changes have been implemented by the programmer on a best-efforts basis. Others may still be implemented in the future if workforce and budget constraints permit.

Conclusion

The Support Imaging Generation Process used by the Cassini Instrument Operations Team was greatly enhanced by the use of automation. It reduced or eliminated the possibility of user input errors during process execution, it made the process much more rapid, it codified the expert knowledge needed to perform the process, and it made possible the staffing reductions dictated by XXM budget constraints. All of this was accomplished while also keeping the process flexible and responsive to changing needs and requirements that will eventually occur on a mission that has been performing its many science data gathering functions since its launch in 1997 and will continue until its propellant resources are depleted in 2017, twenty years after it set out on its journey of discovery.

References

Computer Software

[1]CIMS, Cassini Information Management System, Web Application, Ver. 3.5, California Institute of Technology, Pasadena, CA, 2008.

[2]ISSPT, Imaging Science Subsystem Pre-Commanding Tool, Software Program, Ver. 2.6.7, Cassini Imaging Central Laboratory for Operations, Boulder, CO, 2004.

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Until copyright transferred to publisher, (c) 2010 California Institute of Technology. Government sponsorship acknowledged.