UNIVERSAL PAYLOAD INFORMATION MANAGEMENT

Ralph B. Elmore
Email: ralph.elmore@msfc.nasa.gov

Lockheed Martin Space Operations, Integrated Technology Solutions (ITS)
for the
National Aeronautics and Space Administration (NASA)
George C. Marshall Space Flight Center (MSFC)
Flight Projects Directorate
Ground Systems Department

Introduction

As the overall manager and integrator of International Space Station (ISS) science payloads, the Payload Operations Integration Center (POIC) at Marshall Space Flight Center has a critical need to provide an information management system for exchange and control of ISS payload files as well as to coordinate ISS payload related operational changes. The POIC's information management system has a fundamental requirement to provide secure operational access not only to users physically located at the POIC, but also to remote experimenters and International Partners physically located in different parts of the world. The Payload Information Management System (PIMS) is a ground-based electronic document configuration management and collaborative workflow system that was built to service the POIC's information management needs.

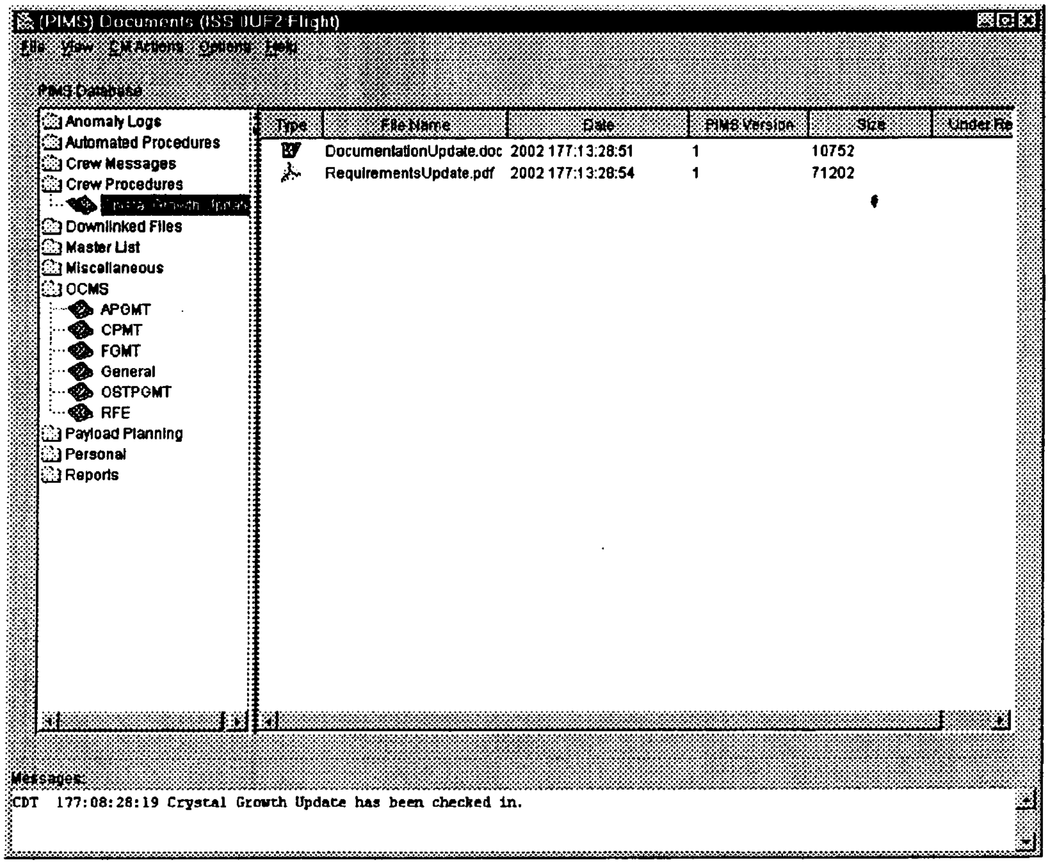
This paper discusses the application components that comprise the PIMS system, the challenges that influenced its design and architecture, and the selected technologies it employs. This paper will also touch on the advantages of the architecture, details of the user interface, and lessons learned along the way to a successful deployment. With PIMS, a sophisticated software solution has been built that is not only universally accessible for POIC customer's information management needs, but also universally adaptable in implementation and application as a generalized information management system.

PIMS Applications and Capabilities

PIMS consists of three main applications that are part of the POIC's arsenal of Web based applications. The "Documents" application is an electronic document configuration management system used by the POIC as a controlled repository of payload operations files. The "Operations Change Request" (OCR) application is a workflow system for managing and approving changes needed during real time payload operations. The "To Do List" application is similar to an email client and provides access to user specific collections of current OCR and Document tasks requiring action. Further details of these applications are explored in the paragraphs that follow.

Documents

The Documents application provides consolidated access to a central repository of the approved versions of documents and files for a specific mission. The appearance of the Documents application is similar to graphical file system traversal application, with the documents and files being organized under folders (see Figure 1 - Documents Application Main Window). A notable difference is that files reside in a document container rather than directly under a folder. A document container may hold any number of files. It is also used to associate additional metadata, such as a title, description, and an email notification list. The document container is the object under primary configuration management control (i.e. if a file change is desired, a revision is performed on the document to change the file). Multiple versions of documents and files are maintained, with checkout and check-in features supported for revision. Document and file revisions are performed in isolation on an in-work copy of the document. The in-work copy is manipulated from the user's To Do List. A document revision will not be publicly accessible until published by the editing user. When new versions become available, the system will automatically generate email notifications of version availability to a document specific list of designated addresses.



- Anomaly Logs
- Automated Procedures
- Crew Messages
- Crew Procedures
  - Crystal Growth Update
- Downlinked Files
- Master List
- Miscellaneous
- OCMS
  - APGMT
  - CPMT
  - FGMT
  - General
  - OSTPGMT
  - RFE
- Payload Planning
- Personal
- Reports

Type	File Name	Date	PIMS Version	Size	Under Re
W	DocumentationUpdate.doc	2002 177:13:28:51	1	10752	
	RequirementsUpdate.pdf	2002 177:13:28:54	1	71202	

```
CDT  177:08:28:19 Crystal Growth Update has been checked in.
```

Figure 1 - Documents Application Main Window

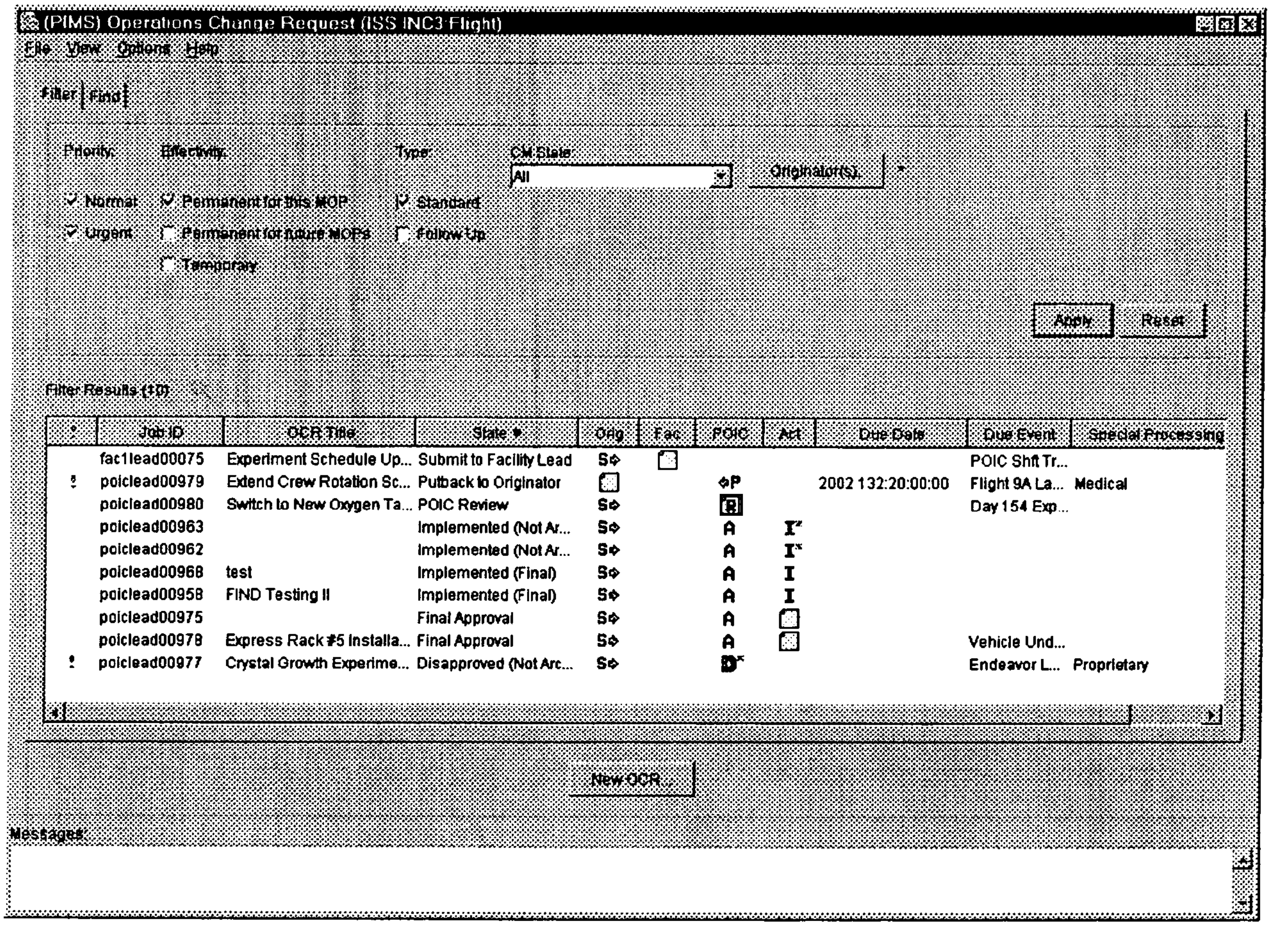
The Documents application is capable of transferring files between the PIMS repository and the machine running the PIMS client (for import, export and integrated viewing), as well any machine running an FTP server (import and export only). The Documents application is also used to transfer files from the PIMS repository to a Johnson Space Center (JSC) accessible drop-box for spacecraft uplink. A background polling process also monitors external facility (e.g. JSC) drop-boxes and will automatically retrieve designated files and place them in the PIMS repository.

There are also legacy POIC applications that require access to PIMS services. An API library interface exists to support programmatic storage and retrieval of files as well as access to document and file information.

Other features built into the Documents application for operating on the PIMS repository include capabilities to compare files, perform a byte swap of files, perform a string search against stored files, and the ability to bulk import files, folders, and documents from an object definition file.

OCRs

The OCR application provides a collaborative, electronic change request system for coordinating the approval and implementation of real time changes desired for spacecraft payload operations. The OCR application provides the capability to create, view, manipulate, and query information concerning OCRs (see Figure 2 - OCR Application Main Window). The application's real-time status displays are maintained with dynamic display updates. The OCR application encapsulates a configurable workflow process which routes OCRs through an entire life cycle of development, approval, implementation, and archival. All OCR workflow processes are coordinated through and routed to individual users via the "To Do List" application.



Priority: ☑ Normal ☑ Urgent

Effectivity: ☑ Permanent for this MOP □ Permanent for future MOPs □ Temporary

Type: ☑ Standard □ Follow Up

CM State: All

Filter Results (10)

!	Job ID	OCR Title	State ▾	Orig	Fac	POIC	Act	Due Date	Due Event	Special Processing
	fac1lead00075	Experiment Schedule Up...	Submit to Facility Lead	S◆	□				POIC Shft Tr...	
!	poiclead00979	Extend Crew Rotation Sc...	Putback to Originator	□		◆P		2002 132:20:00:00	Flight 9A La...	Medical
	poiclead00980	Switch to New Oxygen Ta...	POIC Review	S◆		R			Day 154 Exp...	
	poiclead00963		Implemented (Not Ar...	S◆		A	I			
	poiclead00962		Implemented (Not Ar...	S◆		A	I			
	poiclead00968	test	Implemented (Final)	S◆		A	I			
	poiclead00958	FIND Testing II	Implemented (Final)	S◆		A	I			
	poiclead00975		Final Approval	S◆		A	□			
	poiclead00978	Express Rack #5 Installa...	Final Approval	S◆		A	□		Vehicle Und...	
!	poiclead00977	Crystal Growth Experime...	Disapproved (Not Arc...	S◆		D			Endeavor L...	Proprietary

Figure 2 - OCR Application Main Window

The underlying workflow that supports operations change requests is a mix of both fixed and dynamic steps. In its simplest form, an operations change request starts with an originator, is approved by a lead, and implemented by a capable individual. As required and depending on the change, an OCR approval can be much more involved. It can be routed through hierarchies of approval with optional review phases at each tier of the hierarchy. Reviewers can coordinate subordinate reviews with team members or other individuals within an area of expertise. Likewise, an OCR implementation can be constructed with complex implementation sequences consisting of sequential, parallel, hierarchical, and independent task actions. Refer to Figure 3 - OCR Workflow Diagram for complete details of the OCR workflow process.

PIMS OCR Workflow

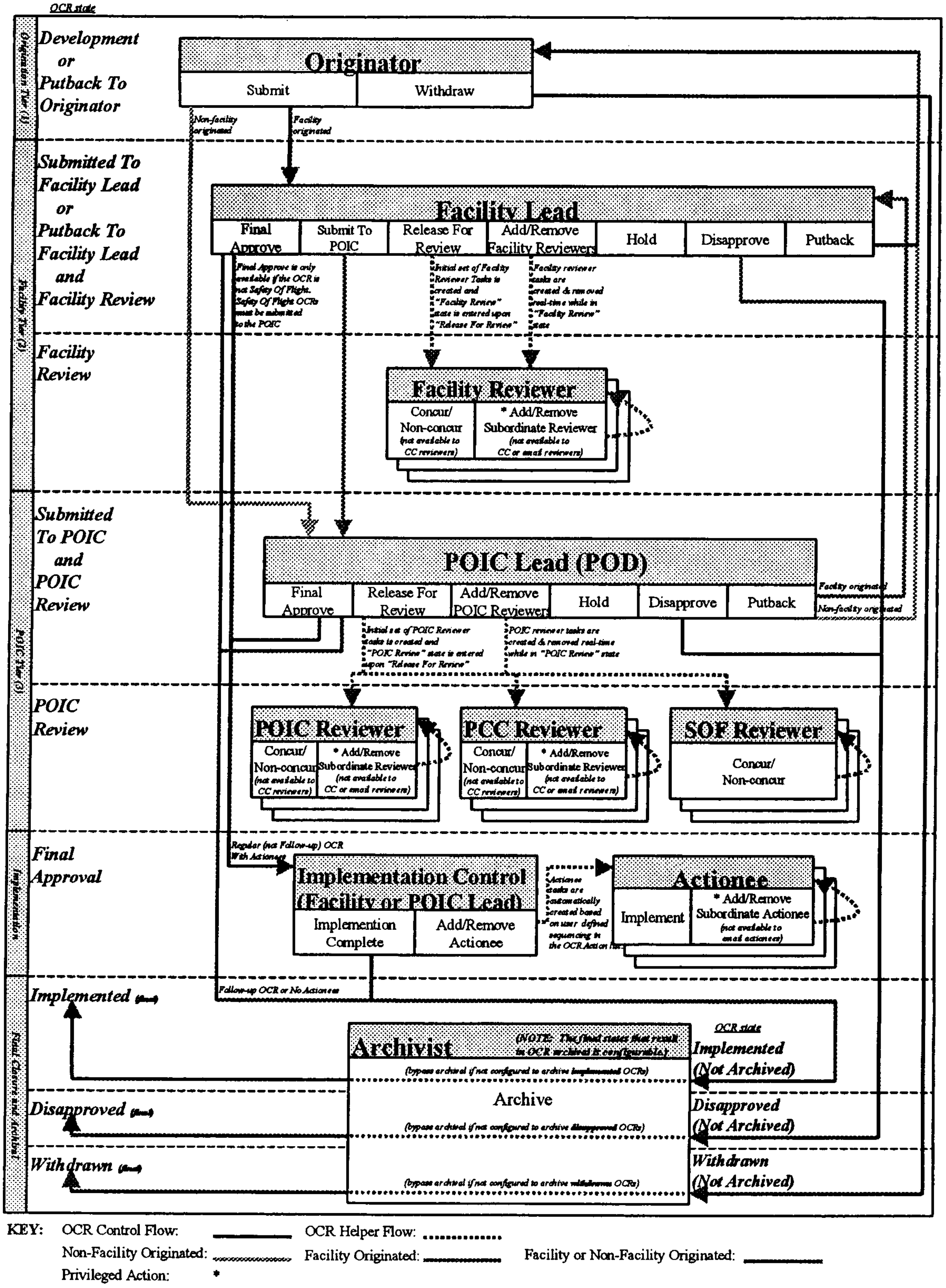

OCR state
Origination Tier (1)
Facility Tier (2)
POIC Tier (3)
Implementation
Final Closure and Archival
Development or Putback To Originator
Originator
Submit
Withdraw
Non-facility originated
Facility originated
Submitted To Facility Lead or Putback To Facility Lead and Facility Review
Facility Lead
Final Approve
Submit To POIC
Release For Review
Add/Remove Facility Reviewers
Hold
Disapprove
Putback
Final Approve is only available if the OCR is not Safety Of Flight. Safety Of Flight OCRs must be submitted to the POIC
Initial set of Facility Reviewer Tasks is created and "Facility Review" state is entered upon "Release For Review"
Facility reviewer tasks are created & removed real-time while in "Facility Review" state
Facility Review
Facility Reviewer
Concur/ Non-concur (not available to CC reviewers)
* Add/Remove Subordinate Reviewer (not available to CC or email reviewers)
Submitted To POIC and POIC Review
POIC Lead (POD)
Final Approve
Release For Review
Add/Remove POIC Reviewers
Hold
Disapprove
Putback
Facility originated
Non-facility originated
Initial set of POIC Reviewer tasks is created and "POIC Review" state is entered upon "Release For Review"
POIC reviewer tasks are created & removed real-time while in "POIC Review" state
POIC Review
POIC Reviewer
Concur/ Non-concur (not available to CC reviewers)
* Add/Remove Subordinate Reviewer (not available to CC or email reviewers)
PCC Reviewer
Concur/ Non-concur (not available to CC reviewers)
* Add/Remove Subordinate Reviewer (not available to CC or email reviewers)
SOF Reviewer
Concur/ Non-concur
Final Approval
Regular (not Follow-up) OCR With Actionees
Implementation Control (Facility or POIC Lead)
Implemention Complete
Add/Remove Actionee
Actionee tasks are automatically created based on user defined sequencing in the OCR Action list
Actionee
Implement
* Add/Remove Subordinate Actionee (not available to email actionees)
Follow-up OCR or No Actionees
Implemented
Disapproved
Withdrawn
Archivist
(NOTE: The final states that result in OCR archival is configurable.)
(bypass archival if not configured to archive implemented OCRs)
Archive
(bypass archival if not configured to archive disapproved OCRs)
(bypass archival if not configured to archive withdrawn OCRs)
OCR state
Implemented (Not Archived)
Disapproved (Not Archived)
Withdrawn (Not Archived)
KEY:
OCR Control Flow:
OCR Helper Flow:
Non-Facility Originated:
Facility Originated:
Facility or Non-Facility Originated:
Privileged Action: *

Figure 3 - OCR Workflow Diagram

To further illustrate the OCR workflow, the following flow diagram (Figure 4 - Sample OCR Workflow) is a hypothetical flow of a non-facility originated OCR (e.g. it was originated by a POIC member) that was routed through the entire workflow process. This OCR was:

[1] submitted to the POIC lead,
[2] released for POIC review,
[3] sent on by one POIC reviewer for subordinate review to two additional reviewers,
[4] subsequently approved by the POIC lead,
[5] implemented with a root sequence of 2 parallel steps (labeled 1) followed by 2 sequential steps (labeled 2 and 3),
[6] as part of the implementation of step 2, was routed through 2 subordinate sequential steps (labeled 2.1 and 2.2),
[7] implementation was then declared complete by the POIC lead, and finally
[8] archived to fully complete the OCR life cycle.

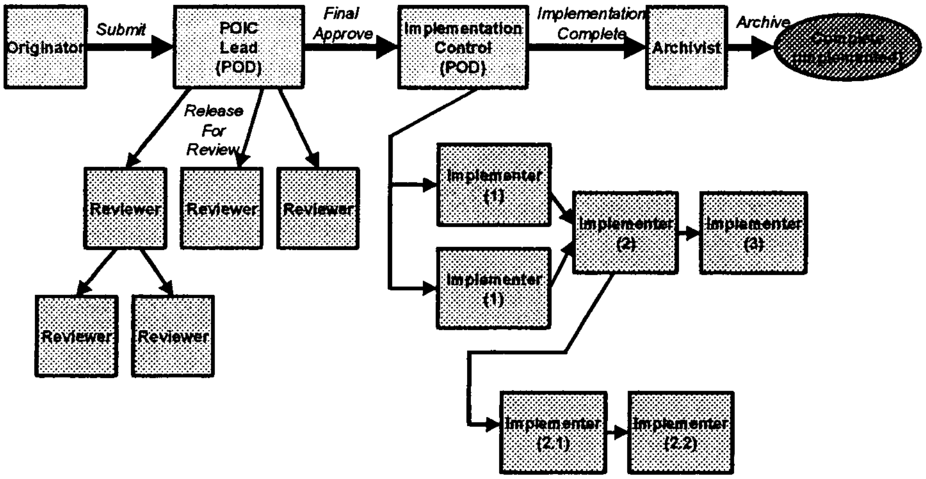
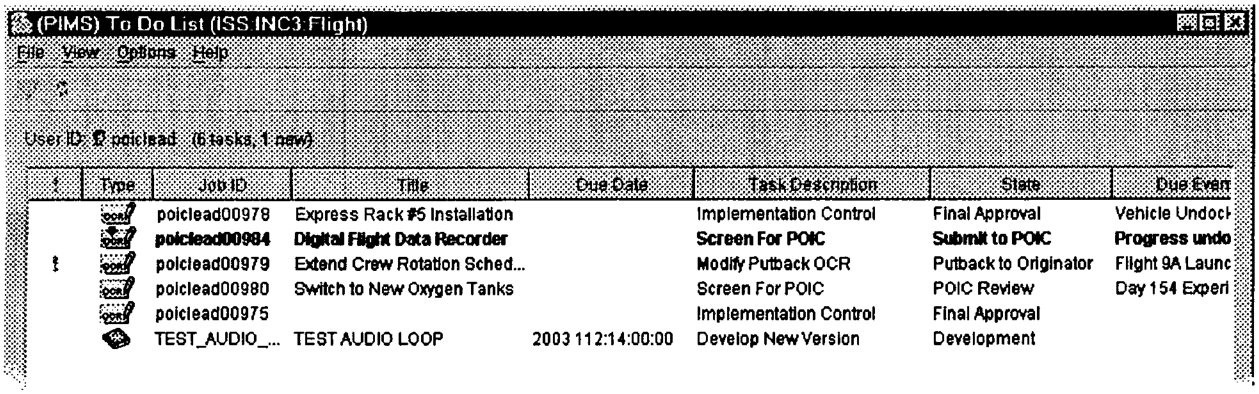


Figure 4 - Sample OCR Workflow

While an OCR is in process, the capability exists for users to store supporting and reviewer attachments of any file type to an OCR. Email messages can be sent to users with notification of OCR progress through the development, approval, and implementation process. Certain actions, such as OCR review or action implementation by non-PIMS users, can be coordinated through automated, system-generated email messages. Lastly, each user involved in the OCR workflow has a log area in which they can permanently record any information concerning their task implementation as part of the OCR.

To Do List

As previously stated, the "To Do List" is an email-like application that provides a list from which users view and operate on current workflow tasks they are required to perform relating to the approval of a Document or an OCR (see Figure 5 - To Do List Application Main Window). The "To Do List" task display is dynamic. Tasks can be automatically added or removed by workflow actions at any time. "Complete Task" screens brought up from the "To Do List" focus the user on the action required and restrict the user's actions to only what is appropriate for the task. Surrogate tasks can involve non-PIMS users in an OCR review or its implementation utilizing email.



User ID: poiclead (6 tasks, 1 new)

!	Type	Job ID	Title	Due Date	Task Description	State	Due Even
	OCR	poiclead00978	Express Rack #5 Installation		Implementation Control	Final Approval	Vehicle Undocl
	OCR	**poiclead00984**	**Digital Flight Data Recorder**		**Screen For POIC**	**Submit to POIC**	**Progress undo**
!	OCR	poiclead00979	Extend Crew Rotation Sched...		Modify Putback OCR	Putback to Originator	Flight 9A Launc
	OCR	poiclead00980	Switch to New Oxygen Tanks		Screen For POIC	POIC Review	Day 154 Experi
	OCR	poiclead00975			Implementation Control	Final Approval	
		TEST_AUDIO_...	TEST AUDIO LOOP	2003 112:14:00:00	Develop New Version	Development	

Figure 5 - To Do List Application Main Window (partial)

Design and Architecture

Challenges

Typical application development challenges existed with the PIMS system. These included industry common issues such as operations concept, external interface, requirement, and technology flux. PIMS also had varying degrees of involvement from the user community. Early in the process there was limited user involvement, with some categories of "power" users being excluded altogether. Beyond these common issues, there were notable application specific challenges in the areas of accessibility, security, and user interface needs.

The heart of PIMS is its centralized storage facility that provides full transaction support to ensure data integrity. The POIC must provide secure access to the centralized PIMS repository and related services to its entire POIC user base. The user base includes individuals physically located at the POIC as well as remote users and facilities. Some users have dedicated connections, while others must access PIMS over the Internet. As such, a controlled access web-based user interface is a fundamental requirement of the system and a major challenge.

Another challenge centered on the Graphical User Interface (GUI) requirements of the PIMS system. It is expected to have the look and feel, input validation, and relative response of a native application (reference previous examples are included in this paper: Figure 1 - Documents Application Main Window, Figure 2 - OCR Application Main Window, and Figure 5 - To Do List Application Main Window). The PIMS GUI is expected to function similar to existing POIC developed applications and commonly used COTS products. Further complicating the GUI are requirements to enforce data interrelationships and input validations real time, with each mouse-click and keystroke. It was therefore not possible to rely on traditional, HTML based, form submittal and validation. Additional GUI challenges are desires for it to function similarly on multiple platforms and for it to process real time updates such that the graphical information display always shows current information.

The PIMS system, while being engineered for web accessibility, also has requirements to interface with legacy systems written primarily in C. These legacy systems required an API library interface that could be linked into the legacy application executable.

The PIMS system is required to support many file transfer interfaces. Among the possible components exchanging of files with the centralized PIMS storage mechanism are:

1. PIMS Web GUI client user workstations.
2. Any FTP server controlled by a PIMS Web GUI client. This is known as a 3-point transfer.
3. Legacy applications, through the aforementioned API library interface.
4. The POIC drop-box, for file exchange with interfacing facilities.
5. The PIMS POIC drop-box polling program, which pulls and imports files from POIC interfacing facilities.

Other design challenges included automated system emails for both status notification and workflow involvement and PDF report generation capabilities.

Selected Technologies

In order to address these outlined design challenges, several modern architectural programming technologies were used.

Java was selected as the primary programming language. It is portable, allowing for platform independence, and web friendly, as it can be run from a browser. The swing component library of Java allows the creation of the required sophisticated GUI. Additionally, enterprise level component libraries exist for Java to support database interaction, email, messaging, and XML parsing.

A Structured Query Language (SQL) based database with ACID (Atomicity, Consistency, Isolation, and Durability) compliance was chosen as the central repository. ACID compliance ensures transaction and data integrity. SQL compliance allows the use of the Java Database Connectivity (JDBC) interface for access from Java. In addition, a SQL based database accessed through a standard JDBC programming interface allows great flexibility in choosing or switching database vendors.

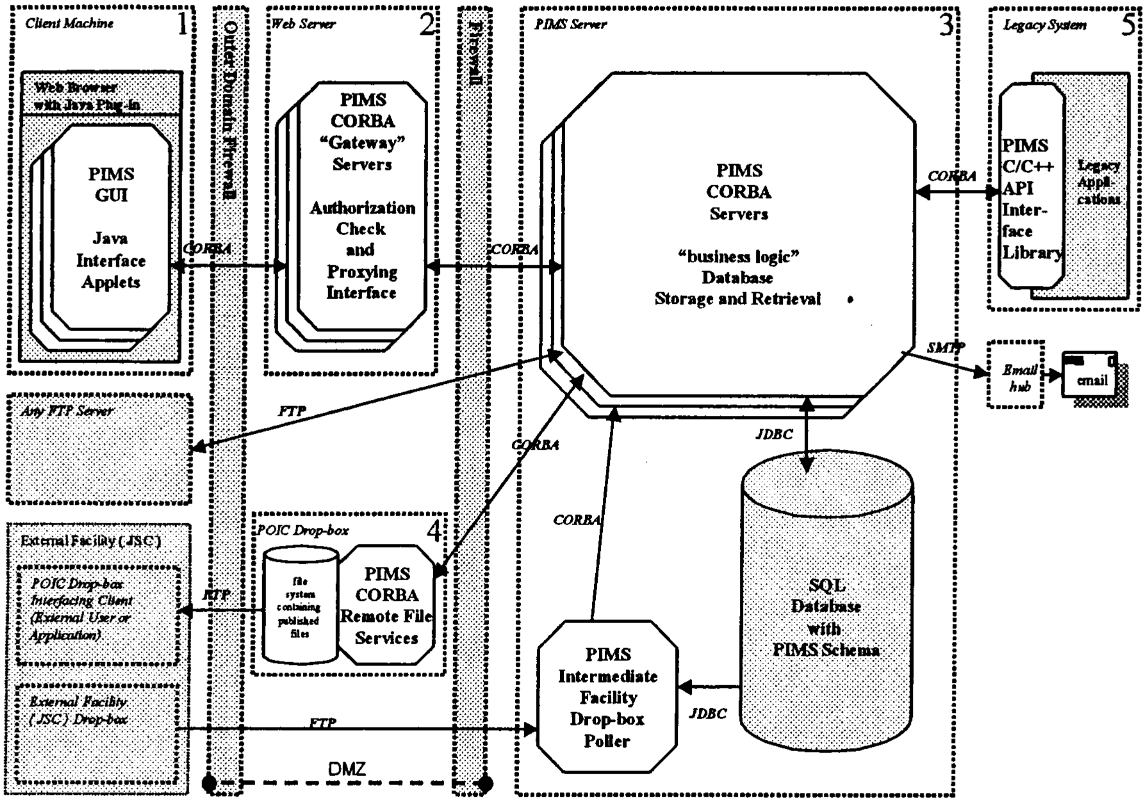
Common Object Request Broker Architecture (CORBA) was selected as the distributed object backbone of the PIMS architecture. The CORBA ability to pass complex objects between systems satisfies tier-to-tier communication needs. The inter-language capability of CORBA by using Interface Definition Language (IDL) facilitated the integration of legacy software and applications.

For security, several technologies were chosen. These technologies are part of the overall POIC web infrastructure of which PIMS is a component and not directly built into the PIMS software. A Virtual Private Network (VPN) is utilized for network security. The VPN provides user and host based authentication. In addition, the encryption capabilities on the VPN "pipe" eliminate encryption needs for individual application connection mechanisms. Object signing allows certification of the client software. Common web security measures such as the use of HTTPS, certificates, password authentication, and account privileges provide additional layers of security.

Finally, as a general rule, an effort was made to ensure that PIMS utilized ubiquitous technologies and file formats based on standards. Sticking with standards makes the system adaptable and allows it to easily interface with other products. Some examples include the use of an integrated File Transfer Protocol (FTP) client for exchanging files between the PIMS repository and non-PIMS systems. Simple Mail Transfer Protocol (SMTP) is used for all PIMS generated emails. Extensible Markup Language (XML) is used for object import into the "Documents" application. Portable Document Format (PDF) is used as the format of all generated reports. Lastly, Zip format is used for archives generated through the OCR archival process.

Platform and Component Architecture

The following diagram (Figure 6 - PIMS Architecture Diagram) illustrates the platform and component architecture of the PIMS system.



Client Machine
1
Web Browser with Java Plug-in
PIMS GUI
Java Interface Applets
Outer Domain Firewall
CORBA
Web Server
2
PIMS CORBA "Gateway" Servers
Authorization Check and Proxying Interface
Firewall
CORBA
PIMS Server
3
PIMS CORBA Servers
"business logic" Database Storage and Retrieval
Legacy System
5
PIMS C/C++ API Inter-face Library
Legacy Appli-cations
CORBA
SMTP
Email hub
email
Any FTP Server
FTP
CORBA
JDBC
CORBA
External Facility (JSC)
POIC Drop-box Interfacing Client (External User or Application)
FTP
POIC Drop-box
4
file system containing published files
PIMS CORBA Remote File Services
SQL Database with PIMS Schema
PIMS Intermediate Facility Drop-box Poller
JDBC
External Facility (JSC) Drop-box
FTP
DMZ

Figure 6 - PIMS Architecture Diagram

As can be seen in this diagram (Figure 6 - PIMS Architecture Diagram), PIMS employs a multi-tiered web-based architecture with a centralized SQL database accessed via Java JDBC drivers. Separate GUI (diagram box 1), gateway (diagram box 2), and business logic layers (diagram box 3) have the advantage of separating the presentation from the implementation. Even though "thin clients" (strictly a served GUI with limited front-end logic) are not used, this multi-tiered architecture still has many the advantages of a thin client system. Business logic can be changed without impacting the clients.

PIMS client applications (diagram box 1) run as Java applets hosted by a Web browser and are part of a larger POIC web application suite. The Java virtual machine is provided by the required Java plug-in. All code runs from pre-installed, signed Java JAR files. Client users maintain persistent connections to the web server. However, all service requests are transaction oriented. Hosting in the Web browser also provides a key PIMS feature, file viewing. Integrated file viewing is accomplished by instructing the browser to open local URLs, thus invoking the native viewing application.

The gateway tier (diagram box 2) is primarily a security layer. The gateway is located in a controlled demilitarized zone (DMZ) network area. The PIMS gateway tier software further enhances security by authenticating user requests on a transactional basis and by proxying authenticated traffic to the secured business logic layer.

The heart of the system is the PIMS Server or business logic tier (diagram box 3). It implements nearly all application specific control logic. It also provides all interfaces with the SQL database.

Additional tiers include the legacy application PIMS API interface library (diagram box 5) and the POIC drop-box (diagram box 4) mechanism used to export files for retrieval by other facilities. Additionally, the diagram shows the drop-box polling program used to automatically import files and information from the drop-boxes of external facilities into PIMS.

Architectural Advantages

Numerous architectural advantages have already discussed in *Selected Technologies* and *Platform and Component Architecture*. However, it is worthwhile to both point out and re-iterate several key features:

1. This PIMS architecture is modular, scalable and highly distributable. It can be run on one box or distributed across many. Running multiple instances of CORBA servers can increase throughput.
2. The PIMS architecture is not platform dependent. Pervasive use of Java allows easy porting.
3. The PIMS architecture is adaptable. Use of pervasive, standards-based technologies allow easier integration into other environments and internal implementation swapping.
4. The PIMS architecture, as deployed at the POIC, is secure.

Lessons Learned

Over the course of developing the PIMS system, several important lessons were learned. First, iterative development was absolutely essential to the success of the system. With the challenges of requirements, concepts, interface, and technology flux, it was imperative to get capability to the users early so that they can begin the discovery and re-evaluation process while additional capability is being added. Further, it provides opportunities to recover from poor architectural component decisions and software implementations. Iterations provide necessary opportunities to continuously enhance and even re-invent the product to best meet the needs of both users and developers.

It is critical that end users, including those who are not necessarily driving the requirements, have direct communication with developers throughout the development process. An open communication channel keeps the users informed and contributing to the design. Users can contribute to the priorities of the next round of enhancements. End user communication also facilitates user satisfaction as they have contributed to the design throughout the iterative process. Minor software changes can often provide a major user savings in terms of time and satisfaction. An open developer-to-user dialog and user community surveys help identify both these minor changes with large user enhancement. It also helps to identify missed requirements and engineering opportunities that could further enhance a user's ability to perform their job.

It is important for developers to constantly monitor and evaluate new technologies. An emerging technology may suggest new and better solutions than originally planned, significant cost savings possibilities, or simply directions to steer the product on the next iteration.

By sticking with ubiquitous technologies based on standards, the system has maximum flexibility, adaptability, and acceptance. This approach makes it easy to replace costly or poorly performing implementations with alternatives. It also provides better interoperability with external COTS products and the desktop environments of a disparate user community.

Infrastructure and utility level "Off-the-shelf" (OTS) components proved more useful and viable than adaptation of large scale, generalized "all-in-one" products that tend to dictate business logic. Early on, it was recognized the POIC needed a "document configuration management and workflow" system. There are commercial product suites that generically perform these functions. Initial incremental design and deployments attempted to tailor and integrate one of these large-scale proprietary products. This resulted in several problems:

- Generally speaking, the required integration code and the resultant system architecture overly complex.
- Limitations of the software prevented evolution of the integrated solution to better meet changing customer needs.
- Startup and recurring maintenance costs were high.
- Performance could not be optimized to a level acceptable for real time operational use.
- Fragmentation of the system components occasionally compromised data integrity on failures.

Ultimately, large-scale "all-in-one" products were abandoned for PIMS. Their use is not advised unless operations concepts, processes, requirements, and user interface needs can be easily adapted to conform to the selected tool. Integration of infrastructure and utility OTS components that had no influence on business logic or requirements were more rewarding for the PIMS system. Examples of these types of products include implementations for SQL database, CORBA, task specific libraries (XML parsing, e-mail, PDF report generation, FTP, etc.), and virus scanning.

Even good selection of OTS components comes with a potentially overlooked maintenance issue. The interrelationships of upgrading hardware, operating system, or any component must be examined in the context of the total system. For example: It may not be possible to upgrade to a new version of Java because the database or CORBA solution does not yet support it.

Conclusion

PIMS is an information management system currently supporting the POIC's operational needs as the overall manager and integrator of ISS science payloads and experiments. It has been operational since March 2001. Even after PIMS became operational, the iterative development process has continued. Iterative improvements have significantly reduced system startup and recurring operational costs. Additionally, the system has adapted to better meet user needs with both new and altered capabilities. User satisfaction has continuously improved.

Sophisticated PIMS Java GUIs are capable of running over the Internet on virtually any common platform from a Web browser. Java GUIs enabled the clients to have the desired native application look and feel. PIMS is secure, scalable, adaptable, and configurable to alternative platforms, infrastructure implementations, and environments. These features are partly made possible by its multi-tier distributed object architecture built with common standards-based, ubiquitous technologies.

The generalized capabilities for change request processing and document configuration management make the system adaptable as an information repository and change request approval system not only for the POIC, but also for alternative purposes. In summary, PIMS provides the POIC universal information management system of generalized capabilities that are widely accessible and highly adaptable.

Universal Payload Information Management

Introduction

- The Payload Operations Integration Center (POIC) at Marshall Space Flight Center (MSFC) manages the operations of science and research experiments aboard the International Space Station.
- The POIC has a critical need to provide information management services for exchange and management of ISS payload files as well as to coordinate ISS payload related operations changes.

Introduction

- The POIC's information management system has a fundamental requirement to provide universal, secure access not only to users physically located at the POIC, but also to remote experimenters and international partners.

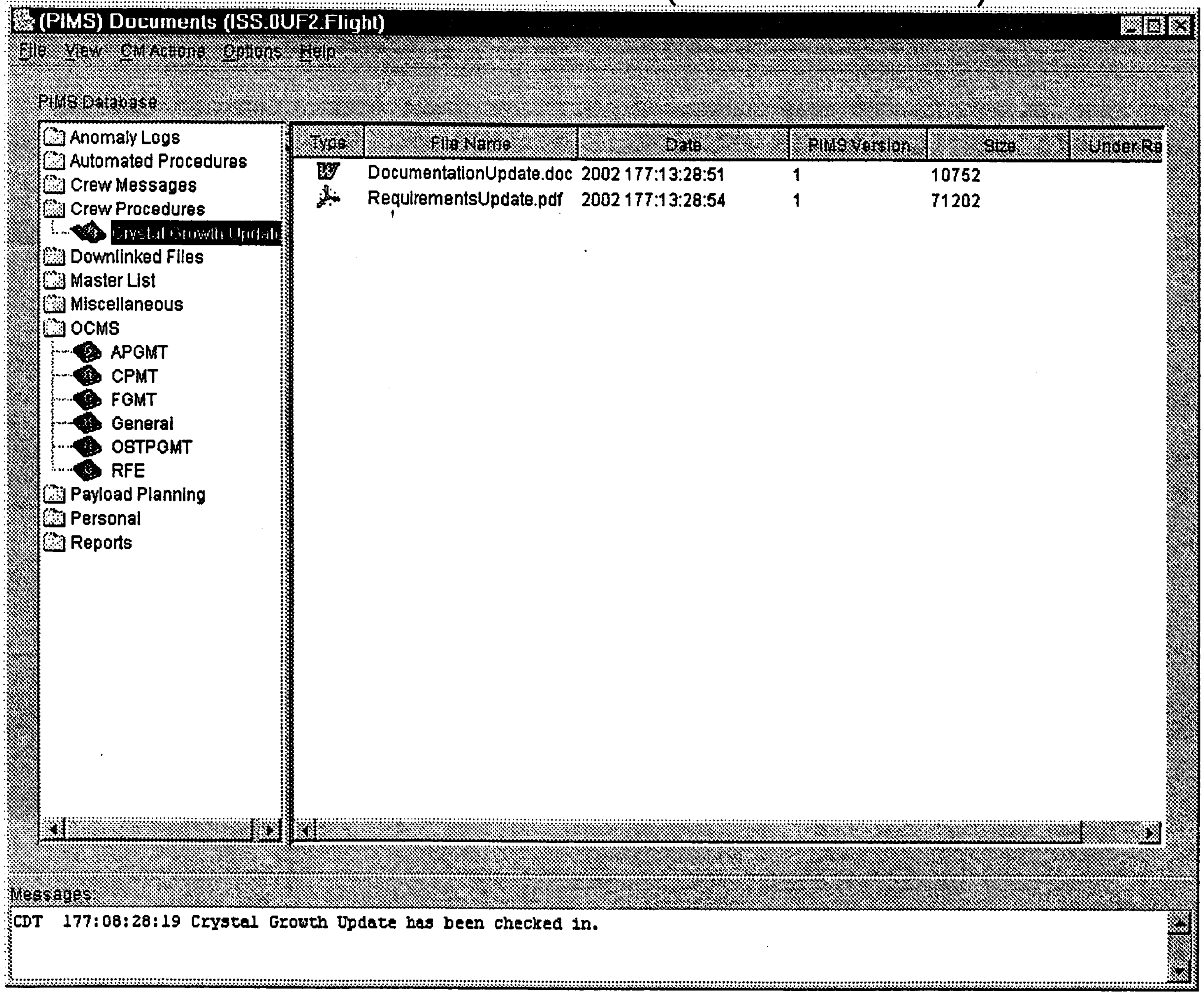
- The Payload Information Management System (PIMS) is a ground based electronic document configuration management and workflow system built to satisfy the POIC's information management needs.

PIMS
Applications and Capabilities

Applications: Documents

- PIMS provides a configuration managed documentation repository that allows for centralized document control, storage and retrieval. Supplemental information and multiple versions are maintained for each document. Other capabilities include:
  - ❖ An integrated capability to retrieve and store files to/from any third-party platform accessible through a standard FTP connection.
  - ❖ User definable email distribution lists to initiate automatic, system-generated, email notifications of new document version availability.
  - ❖ The ability to publish files to an FTP accessible drop-box. This mechanism also supports file uplinks to the spacecraft.
  - ❖ An automated, configurable polling mechanism allows PIMS to retrieve and store files obtained from the drop-boxes of external facilities (e.g. JSC) so that they are available through the document repository to PIMS users.
  - ❖ A programmatic interface that allows other applications to store and retrieve data to and from PIMS.
  - ❖ Other capabilities include file comparison, file byte swapping, string searching of stored files, and bulk import.

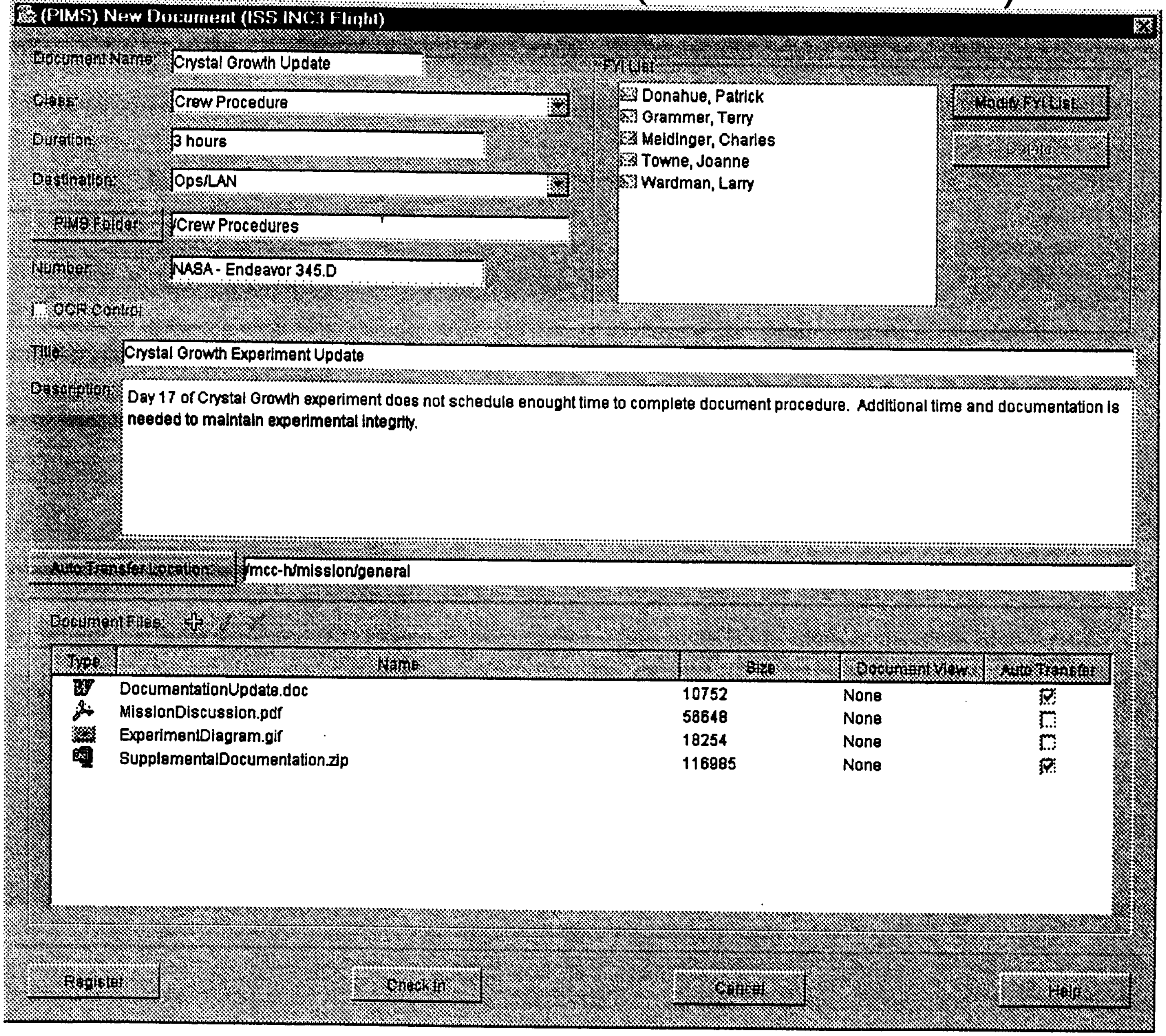
Documents (Main Window)



Type	File Name	Date	PIMS Version	Size	Under Re
W	DocumentationUpdate.doc	2002 177:13:28:51	1	10752	
	RequirementsUpdate.pdf	2002 177:13:28:54	1	71202	

```
CDT  177:08:28:19 Crystal Growth Update has been checked in.
```

Documents (New Document)



(PIMS) New Document (ISS INC3 Flight)

Document Name: Crystal Growth Update

Class: Crew Procedure

Duration: 3 hours

Destination: Ops/LAN

PIMS Folder: /Crew Procedures

Number: NASA - Endeavor 345.D

□ OCR Control

FYI List:

- ☒ Donahue, Patrick
- ☒ Grammer, Terry
- ☒ Meidinger, Charles
- ☒ Towne, Joanne
- ☒ Wardman, Larry

Title: Crystal Growth Experiment Update

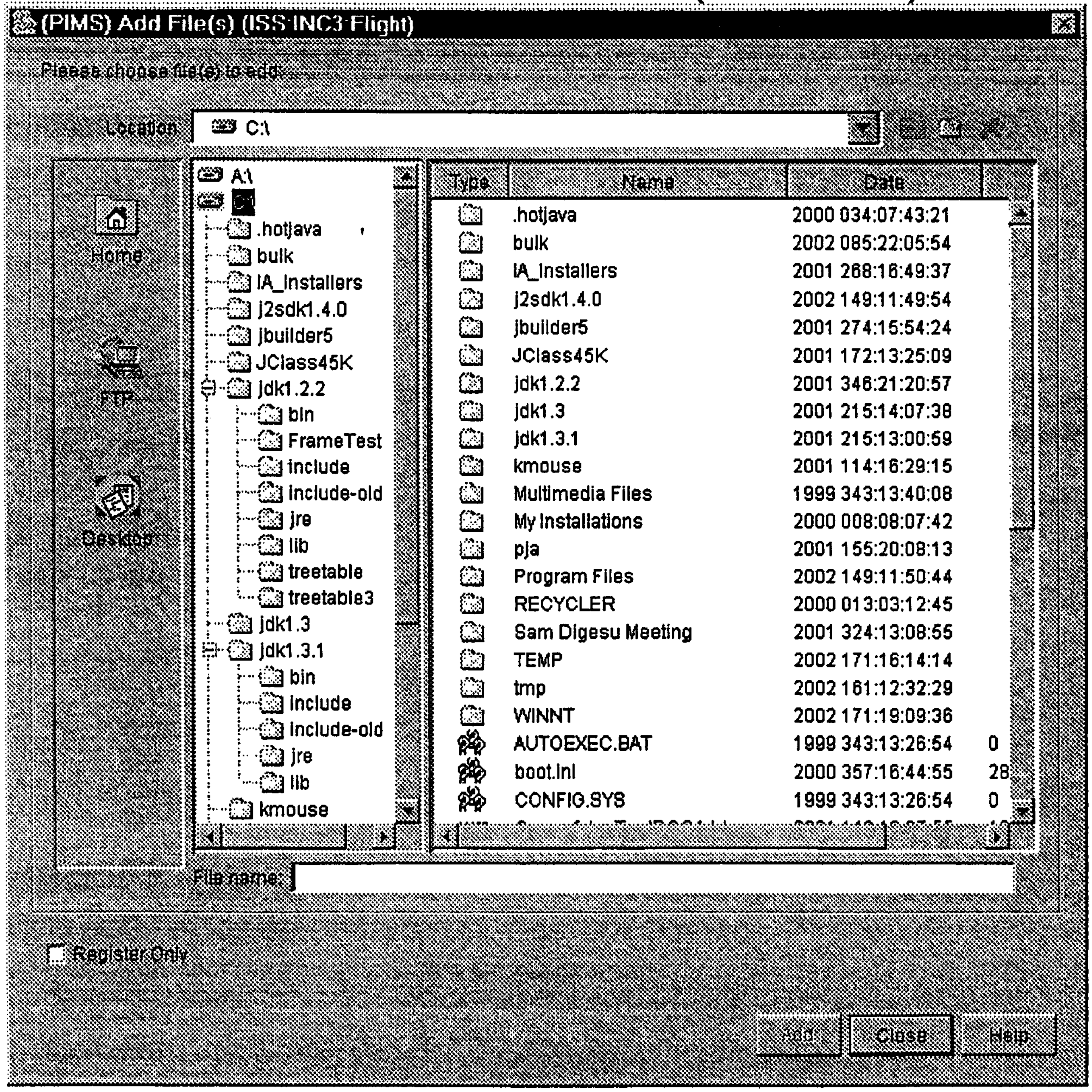
Description: Day 17 of Crystal Growth experiment does not schedule enought time to complete document procedure. Additional time and documentation is needed to maintain experimental integrity.

Auto Transfer Location: /mcc-h/mission/general

Document Files:

Type	Name	Size	Document View	Auto Transfer
	DocumentationUpdate.doc	10752	None	☑
	MissionDiscussion.pdf	58648	None	□
	ExperimentDiagram.gif	18254	None	□
	SupplementalDocumentation.zip	116985	None	☑

Documents (Add File)



(PIMS) Add File(s) (ISS:INC3:Flight)

Please choose file(s) to add:

Location: C:\

```
A:\
C:\
 .hotjava
 bulk
 IA_Installers
 j2sdk1.4.0
 jbuilder5
 JClass45K
 jdk1.2.2
   bin
   FrameTest
   include
   include-old
   jre
   lib
   treetable
   treetable3
 jdk1.3
 jdk1.3.1
   bin
   include
   include-old
   jre
   lib
 kmouse
```

Type	Name	Date	
	.hotjava	2000 034:07:43:21	
	bulk	2002 085:22:05:54	
	IA_Installers	2001 268:16:49:37	
	j2sdk1.4.0	2002 149:11:49:54	
	jbuilder5	2001 274:15:54:24	
	JClass45K	2001 172:13:25:09	
	jdk1.2.2	2001 346:21:20:57	
	jdk1.3	2001 215:14:07:38	
	jdk1.3.1	2001 215:13:00:59	
	kmouse	2001 114:16:29:15	
	Multimedia Files	1999 343:13:40:08	
	My Installations	2000 008:08:07:42	
	pja	2001 155:20:08:13	
	Program Files	2002 149:11:50:44	
	RECYCLER	2000 013:03:12:45	
	Sam Digesu Meeting	2001 324:13:08:55	
	TEMP	2002 171:16:14:14	
	tmp	2002 161:12:32:29	
	WINNT	2002 171:19:09:36	
	AUTOEXEC.BAT	1999 343:13:26:54	0
	boot.ini	2000 357:16:44:55	28
	CONFIG.SYS	1999 343:13:26:54	0

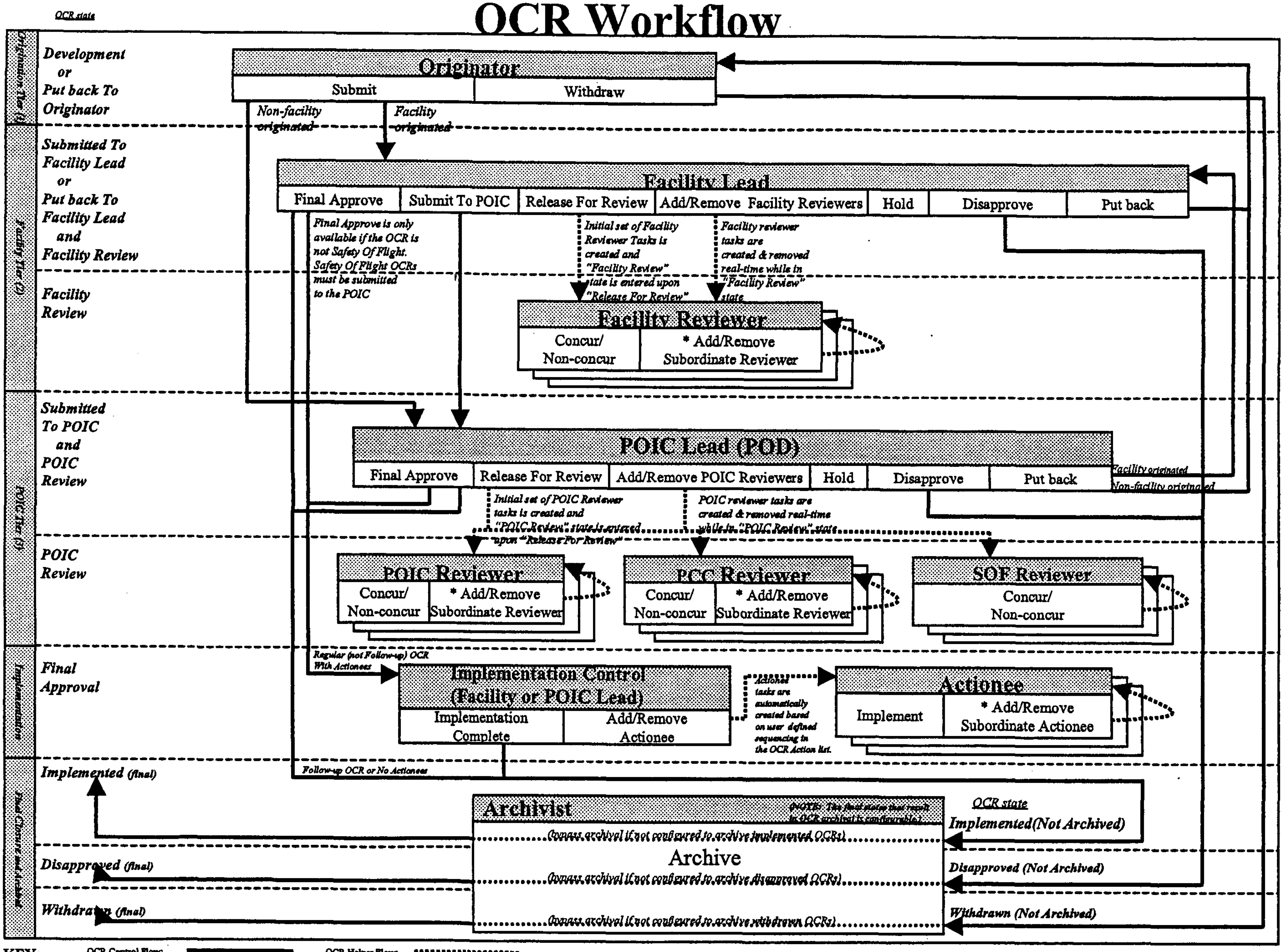
File name:

□ Register Only

Applications: OCRs

- PIMS provides a collaborative, electronic change request approval system for coordinating the approval and implementation of real time changes desired for spacecraft payload operations. This system is known as Operations Change Requests (OCRs) and provides the following:
  - ❖ The capability to create, view, manipulate, and query information concerning OCRs. Real-time status displays are maintained with dynamic display updates.
  - ❖ A configurable workflow process which routes OCRs through developers, multiple tiers of reviewers, approvers, implementers, and OCR archival. Individual user coordination of workflow tasks occurs via the "To Do List".
  - ❖ The capability for users to store supporting supporting and reviewer attachments with an OCR.
  - ❖ Email messages can be sent to users with notification of OCR progress through the development, approval, and implementation process. Certain actions, such as OCR review or action implementation by non-PIMS users, can be coordinated through automated, system-generated email messages.
  - ❖ Each user in the workflow has a log area in which they can permanently record any information concerning their task implementation as part of the OCR.

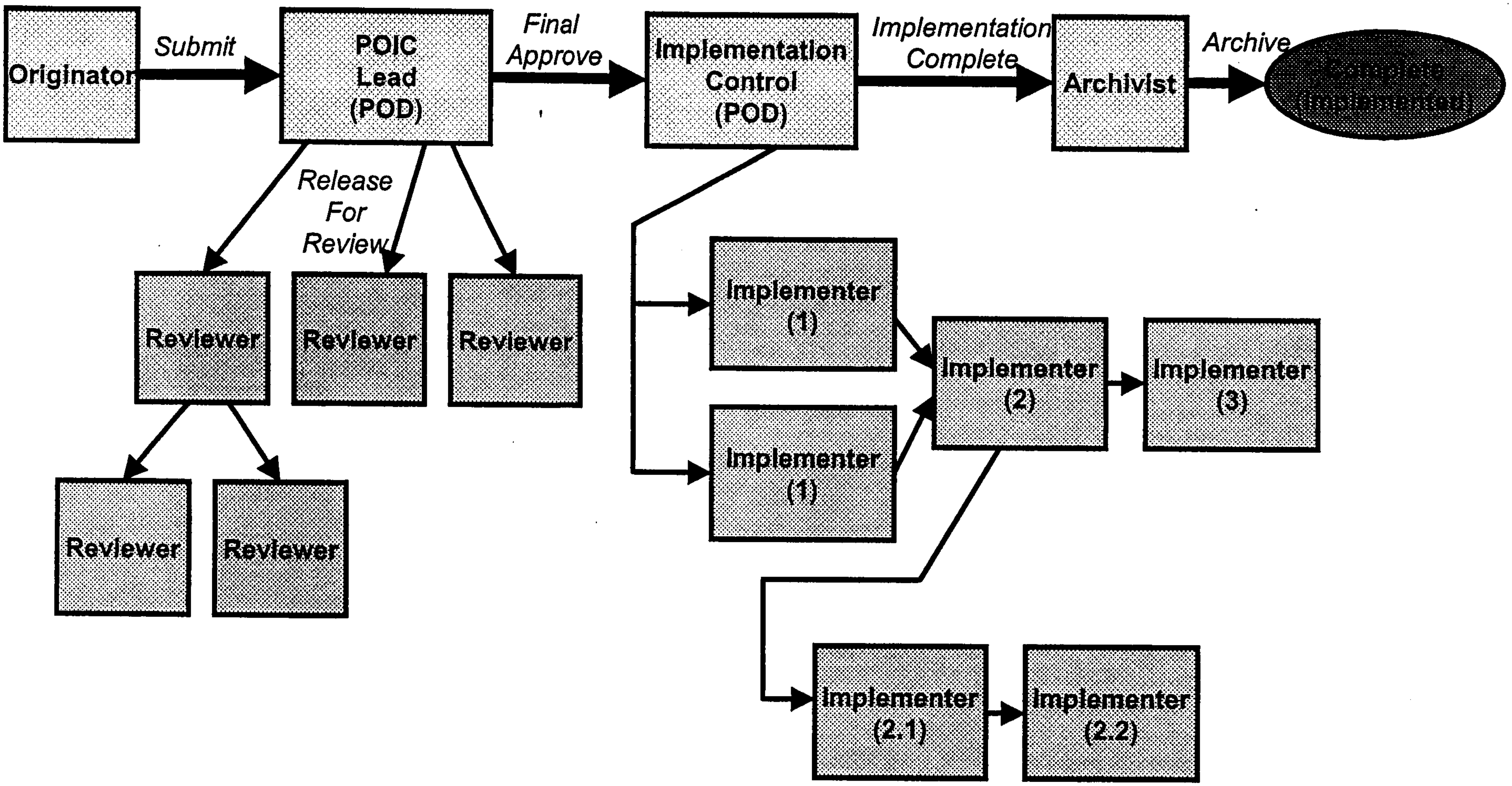
OCR Workflow


OCR state
Origination Tier (1)
Facility Tier (2)
POIC Tier (3)
Implementation
Final Closure and Archival
Development or Put back To Originator
Submitted To Facility Lead or Put back To Facility Lead and Facility Review
Facility Review
Submitted To POIC and POIC Review
POIC Review
Final Approval
Implemented (final)
Disapproved (final)
Withdrawn (final)
Originator
Submit
Withdraw
Non-facility originated
Facility originated
Facility Lead
Final Approve
Submit To POIC
Release For Review
Add/Remove Facility Reviewers
Hold
Disapprove
Put back
Final Approve is only available if the OCR is not Safety Of Flight. Safety Of Flight OCRs must be submitted to the POIC
Initial set of Facility Reviewer Tasks is created and "Facility Review" state is entered upon "Release For Review"
Facility reviewer tasks are created & removed real-time while in "Facility Review" state
Facility Reviewer
Concur/ Non-concur
* Add/Remove Subordinate Reviewer
POIC Lead (POD)
Final Approve
Release For Review
Add/Remove POIC Reviewers
Hold
Disapprove
Put back
Facility originated
Non-facility originated
Initial set of POIC Reviewer tasks is created and "POIC Review" state is entered upon "Release For Review"
POIC reviewer tasks are created & removed real-time while in "POIC Review" state
POIC Reviewer
Concur/ Non-concur
* Add/Remove Subordinate Reviewer
PCC Reviewer
Concur/ Non-concur
* Add/Remove Subordinate Reviewer
SOF Reviewer
Concur/ Non-concur
Regular (not Follow-up) OCR With Actionees
Implementation Control (Facility or POIC Lead)
Implementation Complete
Add/Remove Actionee
Actionee tasks are automatically created based on user defined sequencing in the OCR Action list.
Actionee
Implement
* Add/Remove Subordinate Actionee
Follow-up OCR or No Actionees
Archivist
(NOTE: The final states that result in OCR archival is configurable.)
Archive
(bypass archival if not configured to archive implemented OCRs)
(bypass archival if not configured to archive disapproved OCRs)
(bypass archival if not configured to archive withdrawn OCRs)
OCR state
Implemented (Not Archived)
Disapproved (Not Archived)
Withdrawn (Not Archived)

KEY:

OCR Control Flow: ▬▬▬
Non-Facility Originated: ▬▬▬
Privileged Action: *
OCR Helper Flow: ••••••••
Facility Originated: ▬▬▬
Facility or Non-Facility Originated: ▬▬▬

Example OCR Workflow



Originator

Submit

POIC Lead (POD)

Final Approve

Implementation Control (POD)

Implementation Complete

Archivist

Archive

Complete (Implemented)

Release For Review

Reviewer

Reviewer

Reviewer

Reviewer

Reviewer

Implementer (1)

Implementer (1)

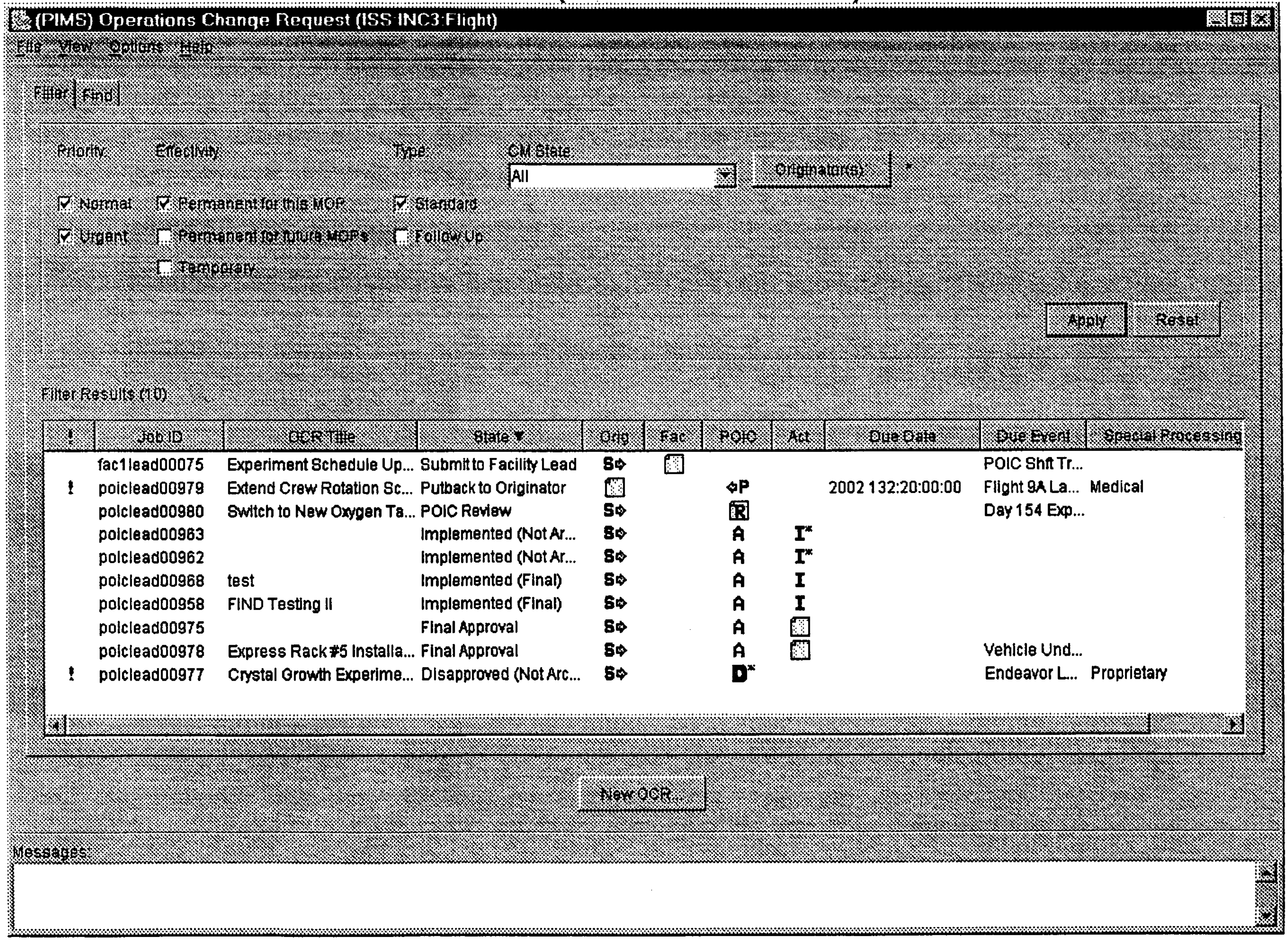
Implementer (2)

Implementer (3)

Implementer (2.1)

Implementer (2.2)

OCR (Main Window)



Priority:
☑ Normal
☑ Urgent

Effectivity:
☑ Permanent for this MOP
☐ Permanent for future MOPs
☐ Temporary

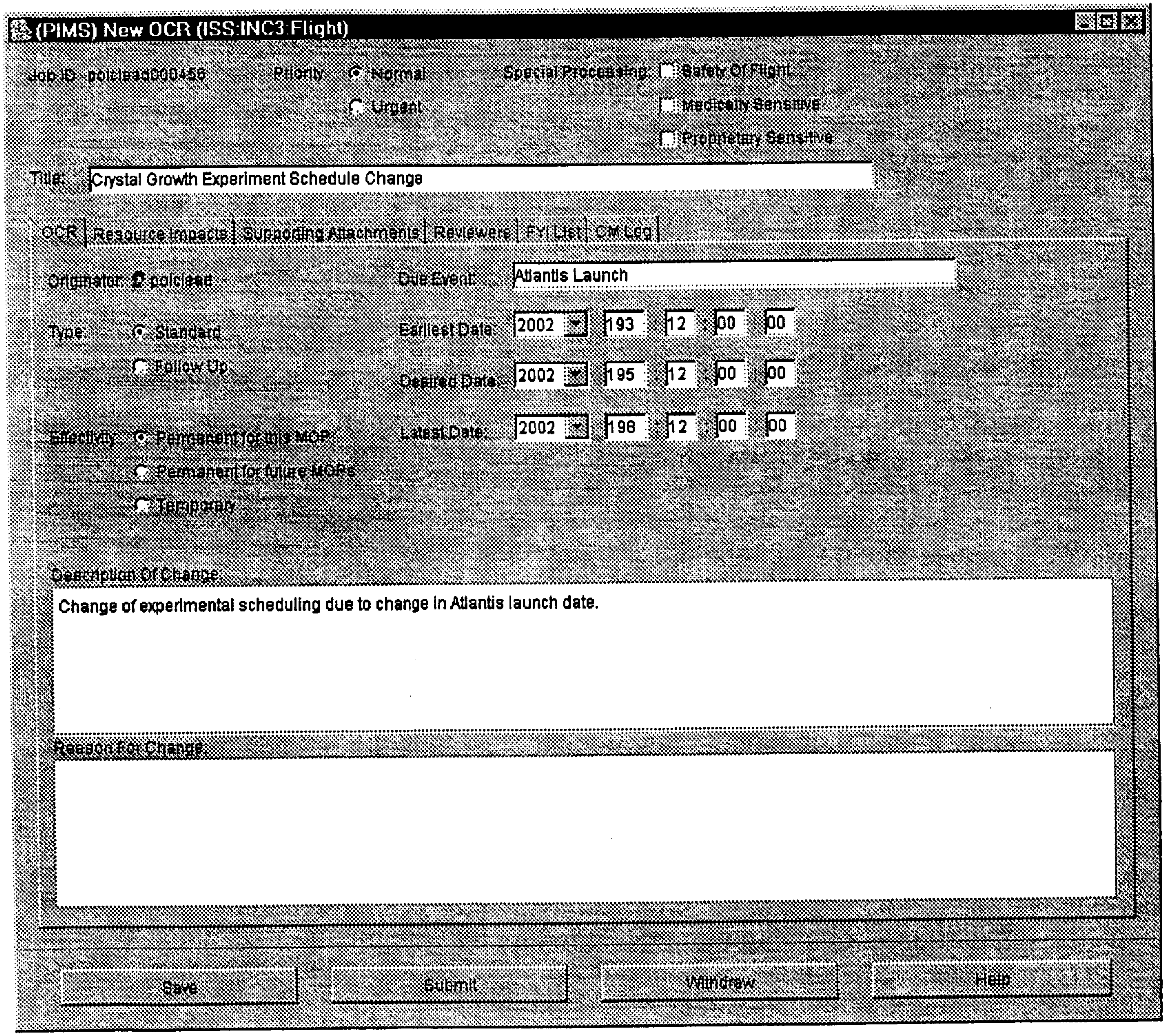
Type:
☑ Standard
☐ Follow Up

CM State: All

Filter Results (10)

!	Job ID	OCR Title	State ▼	Orig	Fac	POIC	Act	Due Date	Due Event	Special Processing
	fac1lead00075	Experiment Schedule Up...	Submit to Facility Lead	S⇨	□				POIC Shft Tr...	
!	poiclead00979	Extend Crew Rotation Sc...	Putback to Originator	□		⇦P		2002 132:20:00:00	Flight 9A La...	Medical
	poiclead00980	Switch to New Oxygen Ta...	POIC Review	S⇨		R			Day 154 Exp...	
	poiclead00963		Implemented (Not Ar...	S⇨		A	I*			
	poiclead00962		Implemented (Not Ar...	S⇨		A	I*			
	poiclead00968	test	Implemented (Final)	S⇨		A	I			
	poiclead00958	FIND Testing II	Implemented (Final)	S⇨		A	I			
	poiclead00975		Final Approval	S⇨		A	□			
	poiclead00978	Express Rack #5 Installa...	Final Approval	S⇨		A	□		Vehicle Und...	
!	poiclead00977	Crystal Growth Experime...	Disapproved (Not Arc...	S⇨		D*			Endeavor L...	Proprietary

OCR (New OCR)



(PIMS) New OCR (ISS:INC3:Flight)

Job ID: poiclead000456

Priority: ● Normal ○ Urgent

Special Processing: □ Safety Of Flight □ Medically Sensitive □ Proprietary Sensitive

Title: Crystal Growth Experiment Schedule Change

OCR | Resource Impacts | Supporting Attachments | Reviewers | FYI List | CM Log

Originator: poiclead

Due Event: Atlantis Launch

Type: ● Standard ○ Follow Up

Earliest Date: 2002 193 : 12 : 00 : 00

Desired Date: 2002 195 : 12 : 00 : 00

Latest Date: 2002 198 : 12 : 00 : 00

Effectivity: ● Permanent for this MOP ○ Permanent for future MOPs ○ Temporary

Description Of Change:

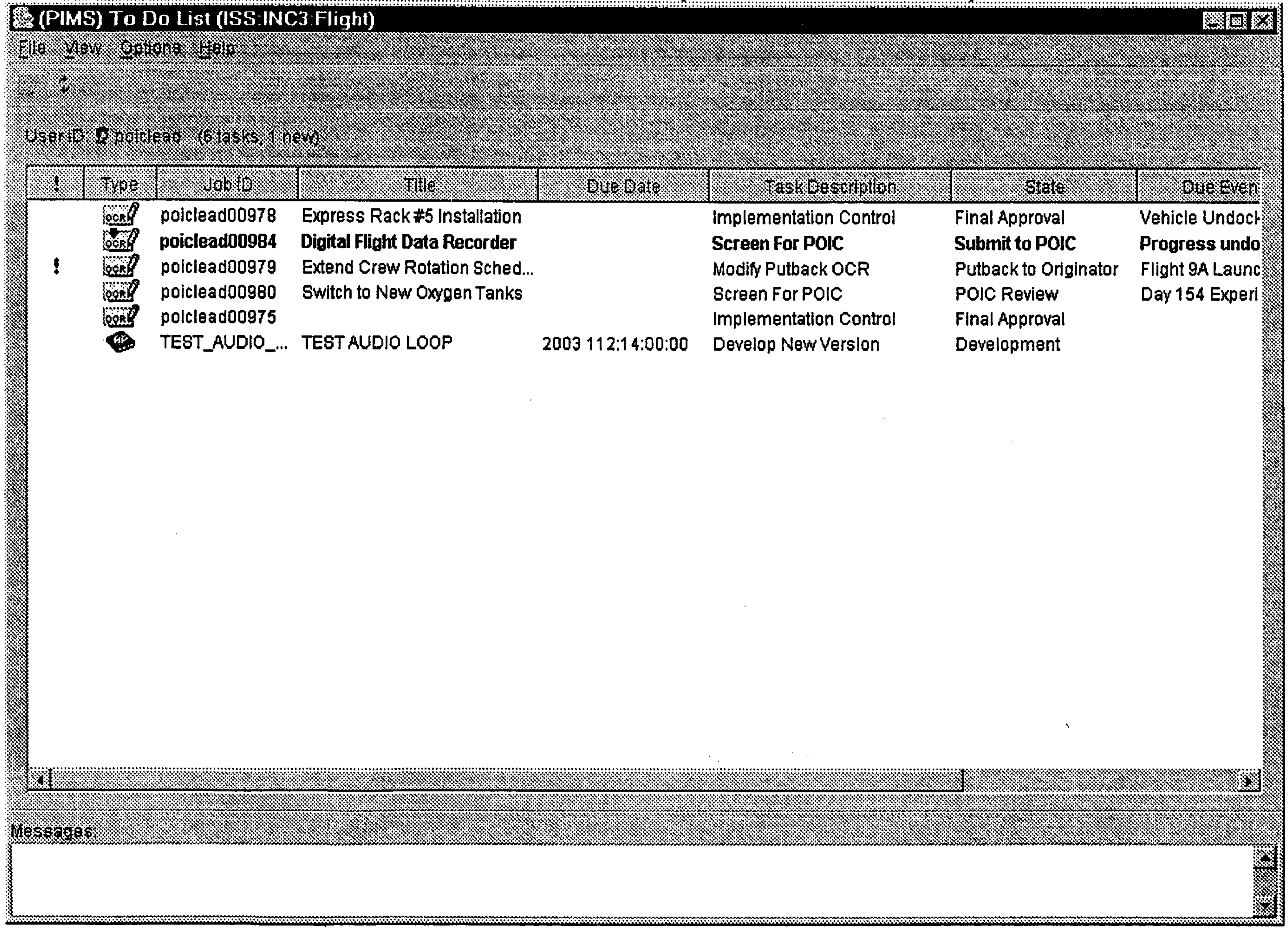
Change of experimental scheduling due to change in Atlantis launch date.

Reason For Change:

Applications: To Do List

- The “To Do List” is an email-like application that provides a list from which users view and operate on current workflow tasks they are required to perform relating to the approval of a Document or OCR.
  - The task display is dynamic. Tasks can be automatically added or removed by workflow actions at any time.
  - “Complete Task” screens brought up from the “To Do List” focus the user on the action required and restrict the user’s actions to only what is appropriate for the the task.
  - Surrogate tasks can be used to involve non-PIMS users in an OCR review or implementation utilizing email.

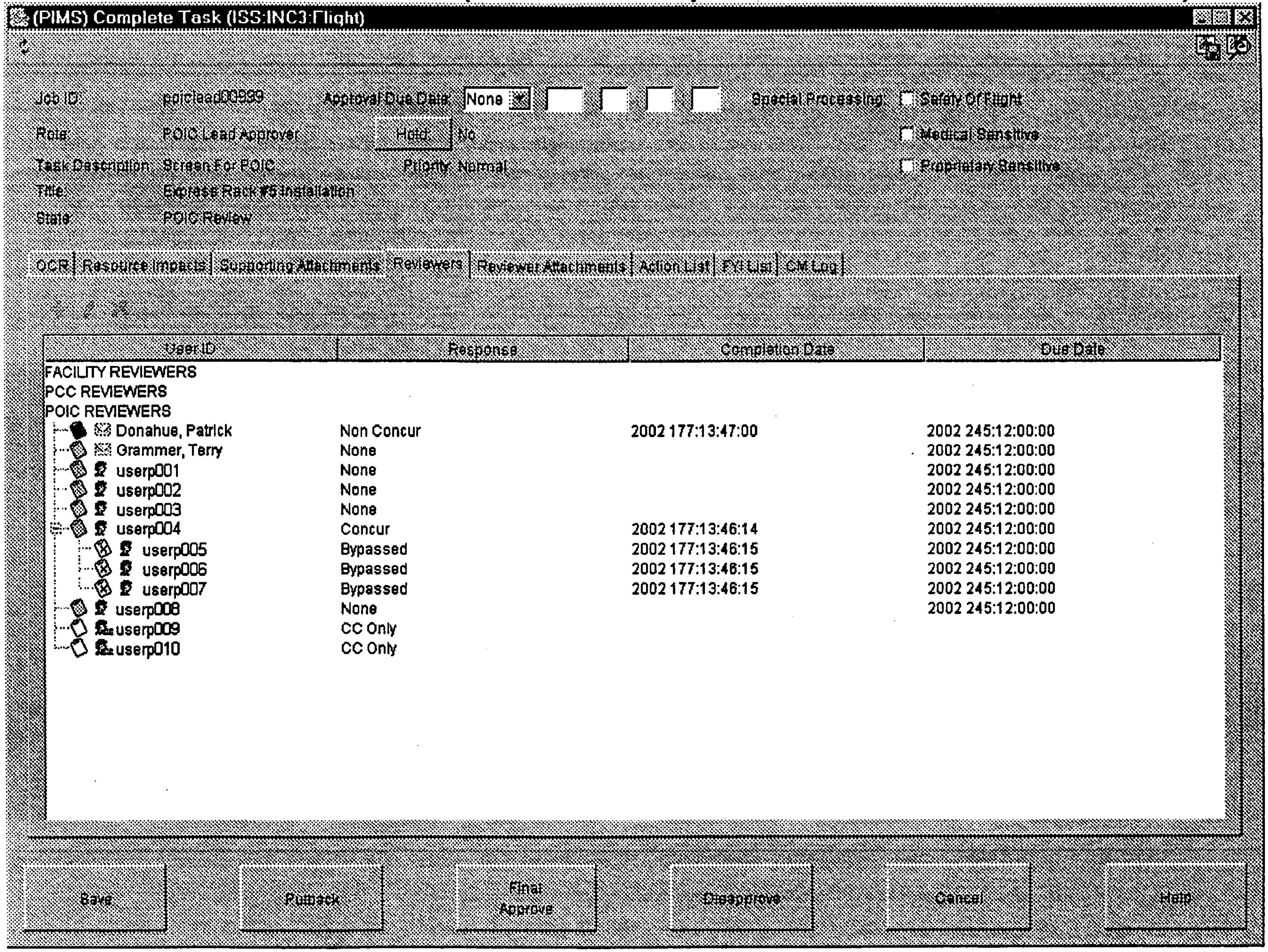
To Do List (Main Window)



User ID: poiclead (6 tasks, 1 new)

!	Type	Job ID	Title	Due Date	Task Description	State	Due Even
	OCR	poiclead00978	Express Rack #5 Installation		Implementation Control	Final Approval	Vehicle Undocl
	OCR	**poiclead00984**	**Digital Flight Data Recorder**		**Screen For POIC**	**Submit to POIC**	**Progress undo**
!	OCR	poiclead00979	Extend Crew Rotation Sched...		Modify Putback OCR	Putback to Originator	Flight 9A Launc
	OCR	poiclead00980	Switch to New Oxygen Tanks		Screen For POIC	POIC Review	Day 154 Experi
	OCR	poiclead00975			Implementation Control	Final Approval	
		TEST_AUDIO_...	TEST AUDIO LOOP	2003 112:14:00:00	Develop New Version	Development	

To Do List (OCR Complete Task – Reviewers)



Job ID: poiclead00999

Approval Due Date: None

Special Processing: □ Safety Of Flight

□ Medical Sensitive

□ Proprietary Sensitive

Role: POIC Lead Approver

Hold: No

Task Description: Screen For POIC

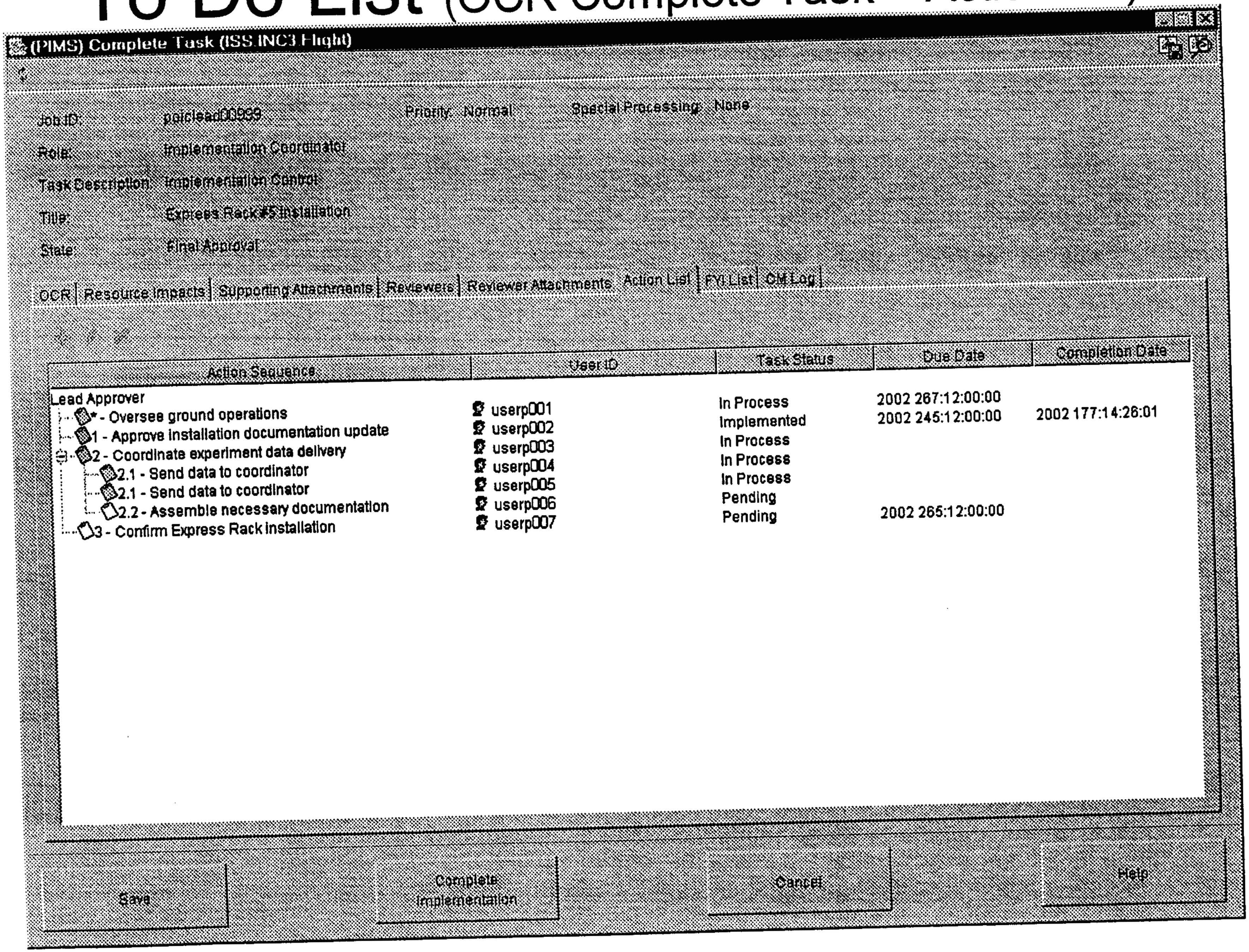
Priority: Normal

Title: Express Rack #5 Installation

State: POIC Review

User ID	Response	Completion Date	Due Date
FACILITY REVIEWERS			
PCC REVIEWERS			
POIC REVIEWERS			
Donahue, Patrick	Non Concur	2002 177:13:47:00	2002 245:12:00:00
Grammer, Terry	None		2002 245:12:00:00
userp001	None		2002 245:12:00:00
userp002	None		2002 245:12:00:00
userp003	None		2002 245:12:00:00
userp004	Concur	2002 177:13:46:14	2002 245:12:00:00
userp005	Bypassed	2002 177:13:46:15	2002 245:12:00:00
userp006	Bypassed	2002 177:13:46:15	2002 245:12:00:00
userp007	Bypassed	2002 177:13:46:15	2002 245:12:00:00
userp008	None		2002 245:12:00:00
userp009	CC Only		
userp010	CC Only		

To Do List (OCR Complete Task – Actionees)



(PIMS) Complete Task (ISS.INC3 Flight)

Job ID: poiclead00999 Priority: Normal Special Processing: None

Role: Implementation Coordinator

Task Description: Implementation Control

Title: Express Rack #5 Installation

State: Final Approval

OCR | Resource Impacts | Supporting Attachments | Reviewers | Reviewer Attachments | Action List | FYI List | CM Log

Action Sequence	User ID	Task Status	Due Date	Completion Date
Lead Approver				
* - Oversee ground operations	userp001	In Process	2002 267:12:00:00	
1 - Approve installation documentation update	userp002	Implemented	2002 245:12:00:00	2002 177:14:26:01
2 - Coordinate experiment data delivery	userp003	In Process		
2.1 - Send data to coordinator	userp004	In Process		
2.1 - Send data to coordinator	userp005	In Process		
2.2 - Assemble necessary documentation	userp006	Pending		
3 - Confirm Express Rack installation	userp007	Pending	2002 265:12:00:00	

Design and Architecture

Design Challenges

- Secure access is required for remote users, including authorized internet users.
- User Interface must mimic look and feel of other existing POIC and typical COTS applications, and thus requires a native application appearance. See previous examples.
- User Interface displays and underlying data have sophisticated inter-relationships to be enforced during data entry.
- System is used for collaborative real time operations. User interaction delays must be minimal and information status displays must be dynamic.
- Centralized data storage is required, with data integrity ensured.
- System interfaces with legacy software systems on different hardware and written in different languages.

Design Challenges

- Fundamentally, the system requires generic capabilities (document management and change request workflow processing). However, there are POIC specific needs. In addition to the generic capabilities listed previously, the POIC requires:
  - "Meta-data" collection of additional POIC required information associated with documents and files.
  - Multiple file exchange mechanisms with the central storage, including integrated drop box exports for file exchanges with other facilities. Spacecraft uplinks utilize this mechanism.
  - Email notification of events.
  - PDF report generation capabilities.
  - Change request workflow full life cycle process that is multi-tiered, part fixed, but dynamically extensible.

Design Challenges

- End-user operations concepts, requirements, and needs are constantly evolving throughout design and development of the system.
- Information Technology, especially in the areas of internet capable applications, as well as hardware have also been evolving.
- Several categories of system users exist, many of whom are not directly contributing to the requirements and design.

Selected Technologies

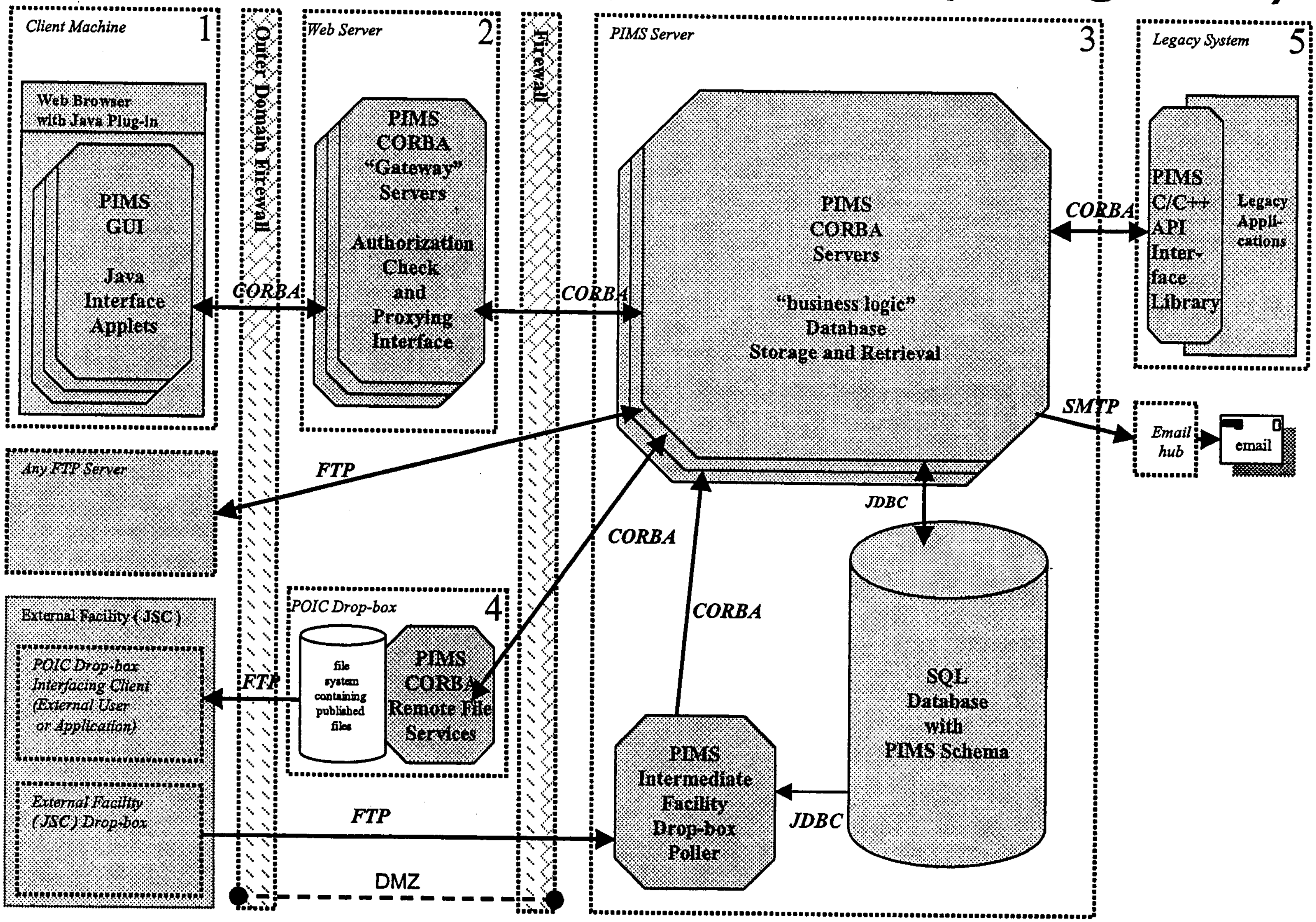
- Java programming language
  - Portable - allows platform independence.
  - Web friendly - can be invoked from a browser.
  - Supports the sophisticated Graphical User Interface (GUI) requirements.
  - Support for key features through standard J2EE API extensions (database access, email, messaging)

- SQL Database with full “ACID” compliant transaction support
  - Ensures data integrity.
  - A SQL standard database allows for :
    - Use of Java Database Connectivity (JDBC) standard interfaces.
    - Use of SQL compliant query and management tools.
    - Ability to change databases with minimal impact.

Selected Technologies

- Common Object Request Broker Architecture (CORBA)
  - Provides distributed communication backbone for object exchange.
  - Inter-language capability allows easier integration with legacy components in different languages.
  - CORBA Event Service provides display update distribution mechanism.
- Virtual Private Network (VPN) for network level security
  - Provides user and host based access authentication
  - Encryption on the VPN "pipe" eliminates encryption needs for individual application connection mechanisms.
- Object signing, https, certificates, privileges, password authentication, and software features add additional security.
- As a rule, utilize ubiquitous technologies and file formats based on standards.
  - Some examples not previously noted include: FTP (file transfer), SMTP (e-mail), XML (object import), PDF (reports), and ZIP (archives)

PIMS POIC Architecture (Diagram)


Client Machine
1
Web Browser with Java Plug-In
PIMS GUI Java Interface Applets
Outer Domain Firewall
CORBA
Web Server
2
PIMS CORBA "Gateway" Servers
Authorization Check and Proxying Interface
Firewall
CORBA
PIMS Server
3
PIMS CORBA Servers
"business logic" Database Storage and Retrieval
CORBA
Legacy System
5
PIMS C/C++ API Inter-face Library
Legacy Appli-cations
SMTP
Email hub
email
Any FTP Server
FTP
JDBC
CORBA
CORBA
SQL Database with PIMS Schema
External Facility (JSC)
POIC Drop-box Interfacing Client (External User or Application)
FTP
POIC Drop-box
4
file system containing published files
PIMS CORBA Remote File Services
External Facility (JSC) Drop-box
FTP
PIMS Intermediate Facility Drop-box Poller
JDBC
DMZ

Architecture

- PIMS employs a multi-tiered web based architecture with a centralized database.
- PIMS client applications run as pre-installed Java applets hosted by a Web browser as part of a larger POIC web application suite. The Java virtual machine is provided by Java plug-in.
- Client users maintain persistent connections to the web server. However, all service requests are transaction oriented.
- The main part of the PIMS architecture is 3-tier:
  - Tier 1 - Client application graphical user interface (1)
  - Tier 2 – Gateway. Authentication and proxying for each transaction. (2)
  - Tier 3 – Business logic and all database communication. (3)
- Legacy systems communicate with PIMS via an API library. (5)
- Tier-to-tier communication occurs utilizing CORBA.
- Database communication occurs utilizing the Java JDBC driver.

Architecture Advantages

- Architecture is scalable
  - The entire architecture can be run on one box or distributed across many.
  - Multiple instances of CORBA servers can be run to increase throughput.
- Architecture is not platform dependent
  - Pervasive use of Java on client and server makes the system easily portable.
- Use of standards based technologies allows easy implementation swapping. Some examples:
  - Database implementation can be easily changed due to JDBC interface and SQL.
  - Messaging Service implementation can be changed to any JMS compliant implementation.
  - CORBA implementation can be changed.
- Though thin clients are not used, still has the advantages of multi-tiered systems.

Lessons Learned

Lessons Learned

- Iterative development is absolutely essential.
  - Must get some capability to the users so that they can begin the discovery and re-evaluation process. Ensure that users are aware of the evolutionary development process.
  - Must have development windows to refine development decisions and even infrastructure.
- Keep the users involved, even those that are not necessarily the mouthpieces for system requirements.
- Constantly monitor and evaluate industry trends and related new technologies.
  - An emerging technology may suggest new and better solutions than originally planned, significant cost savings possibilities, or simply directions to steer the product on the next iteration.
- Stick with solutions based on ubiquitous technologies and standards.
  - Maximizes flexibility, adaptability, and product acceptance.

Lessons Learned

- Infrastructure and utility level "Off-the-shelf" (OTS) components proved more useful and viable than adaptation of large scale, generalized "all-in-one" products that tend to dictate business logic.
  - Early on, it was recognized that the POIC needed a "document configuration management and workflow" system. There are commercial product suites that generically perform these functions. Initial incremental design and deployments attempted to tailor and integrate one of these large scale products. This resulted in several problems:
    - Generally, the code required to integrate the OTS component and the resultant system architecture was overly complex.
    - Limitations of the software prevented evolution of the integrated solution with customer needs.
    - Startup and recurring maintenance costs were high.
    - Performance could not be optimized to a level acceptable for real time operational use.
    - Fragmentation of the system components occasionally compromised data integrity on failures.

Lessons Learned

- Use of OTS components (continued):
  - **Ultimately, large scale "all-in-one" products were abandoned for PIMS.**
  - Integration of infrastructure and utility OTS components that had no influence on business logic or requirements were far more rewarding for the PIMS system. Examples of these types of products include:
    - Database
    - CORBA
    - Task specific libraries (XML parsing, E-mail, PDF report generation, FTP)
    - Virus scanning
- Even good selection of OTS components comes with often overlooked maintenance issues. The interrelationships of upgrading hardware, operating system, or any component must be examined in the context of the total system. For example:
  - It may not be possible to upgrade to a new version of Java because the database or CORBA solution does not yet support it.

Conclusion

Conclusion

- PIMS is an information management system currently supporting the POIC's operational needs as the overall manager and integrator of ISS science payloads and experiments. It has been operational since March 2001.
- Even after PIMS became operational, the iterative development process has continued. It has significantly reduced its system startup and recurring operational costs. It has adapted to better meet user needs with both new and altered capabilities. User satisfaction has continuously improved.
- By using a multi-tier distributed object architecture built with common standards-based, ubiquitous technologies, PIMS is highly scalable and adaptable to alternative platforms and infrastructure implementations.

Conclusion

- Sophisticated Java GUIs give the client a native look and feel, and are capable of running on virtually any common platform from a Web browser.
- Generalized capabilities for change request processing and document configuration management make the system adaptable as an information repository and change request approval system for alternative purposes.
- In summary, PIMS provides the POIC universal information management system of generalized capabilities that are widely accessible and highly adaptable.

Questions and Answers