

5. MLCI - Management Logistics CSCI

The Management Logistics CSCI (MLCI) for Release B consists of: the Configuration Management Services (Baseline Manager, Software Change Manager, Change Request Manager, the Software License Manager, the Software Distribution Manager, the Inventory/Logistics/Maintenance (ILM) Manager, the Training Manager, and the Policies and Procedures Manager). The Management Logistics context is provided in Figure 5-1.

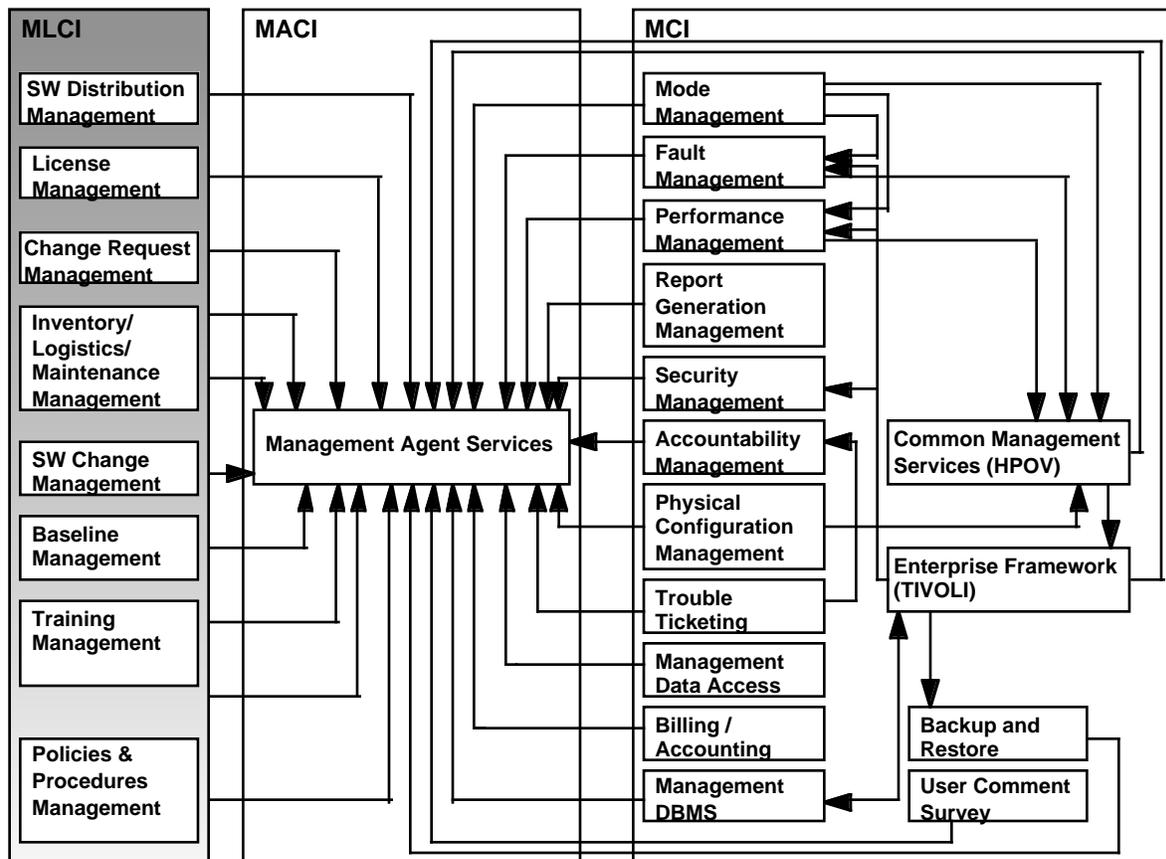


Figure 5-1. Management Logistics CI Context

5.1 Management Logistics Overview

For Release B, the Management Logistics Configuration Item (MLCI) implements: the Configuration Management Services [Baseline Manager, Software Change Manager, Change Request Manager, the Software License Manager, the Software Distribution Manager, the Inventory/Logistics/Maintenance (ILM) Manager, the Training Manager, and the Policies and Procedures Manager]. The MLCI provides tools with which M&O staffs at the DAACs, EOC, and

SMC: track deployed ECS baselines and control changes to the hardware and software that comprise them; distribute software and control the commercial off-the-shelf (COTS) products' licenses; track and control the inventory of ECS equipment, COTS software and documentation; track use of consumable items and logistics supplies; track preventive and corrective maintenance performed on ECS equipment; track training and certification of M&O personnel; and, maintain and distribute information on ECS policies and procedures.

The Configuration Management Services maintain electronic stores of baseline data, software, and system change requests that enter the operational environment, making them and a variety of reports available for system maintenance and operations activities. They accept ECS and algorithm software and non-real time configuration management data from formatted files or via operator interface. M&O staffs, sustaining engineers, and AIT teams rely on CM data stores to make, track and audit configuration changes and to help enforce ECS CM rules. They also use CM to produce formatted files containing change requests, site baseline records, software, documentation, and reports that can be made available for distribution system-wide via CSS services such as e-mail, ftp, Tivoli/Courier Service, and the ECS bulletin board.

For Release B, the MLCI design includes three service from Release A and five new services:

- Baseline Manager - COTS application that will help the DAACs, EOC, and SMC M&O staff manage ECS baseline records. The Baseline Manager, XRP-II, is a COTS application which is an interactive product. It will facilitate the tracking of baselines but will require development of reports, tailoring of input forms, scripts to provide an interface to *ClearCase* and *DDTS*, and triggers for system management activities.
- Software Change Manager - A software application that will help the DAACs, EOC, and SMC M&O staffs organize and partition software, control software changes and versions and assemble sets of software for release purposes. *ClearCase*, a COTS product, will perform the Software Change Manager functions. *ClearCase* provides extensive software library management facilities but requires development of reports, triggers to implement policies and provide an interface with *DDTS* and the Baseline Manager (XRP-II), scripts for frequently recurring operations, and views for providing access to related files.
- Change Request Manager - A software application that will enable the DAACs, EOC, and SMC staffs to register and keep track of configuration change requests (CCR), non-conformance (NCR), and deficiency reports (DR) electronically. The *Distributed Defect Tracking System (DDTS)*, a COTS product, will perform the Change Request Manager functions. *DDTS* provides interactive functionality for tracking CCRs and NCRs but will require customization of forms and reports, tailoring of rules and event flow logic to the ECS environment, and scripts for interfacing with *ClearCase* and the Baseline Manager (XRP-II).
- Software Distribution Manager - a software application that will distribute software upgrades, toolkit software, and accompanying documentation. This distribution can be done either automatically or through a repository (bulletin board). Tivoli/Courier, a COTS product, will perform the Software Distribution Manager functions.
- Software License Manager - a software application that will control and manage the distribution of licenses across ECS sites. For commercial (COTS) licensed products the server configuration and type/number of clients able to access the particular software

product will be governed by the terms of the software licensing agreement. FLEXIm and iFORLS, COTS products, will perform the Software License Manager functions.

- Inventory/Logistics/Maintenance Manager - a software application that will satisfy all requirements for inventory (including property management), logistics, and maintenance. It will track and maintain all of the key data pertaining to ECS equipment, COTS software, documentation and supplies. It will enable the M&O Staff control ECS assets. It will track the on-hand and back-ordered supplies (both consumables and spares) that are used to support operation and maintenance of ECS equipment. The supplies that are managed by this application include such items as: power supplies, toner cartridges, monitors, computer paper and spare interface cards (Line Replaceable Units - LRUs). Also, it will track both corrective and preventive maintenance actions. Normally corrective maintenance will be initiated by the trigger of a trouble ticket. A corrective maintenance job will be initiated to manage the series of actions to remedy the problem: troubleshoot a failed component; replace a failed component; call a site maintenance technician or a third-party vendor (contract maintenance). A preventive maintenance job will be managed in a similar manner except that the remedies (cleaning, adjusting, or other preventive measure) will be scheduled to cause minimum disruption. The functional requirements of the ILM Manager will be satisfied by a COTS product. The COTS selected will be completed by CDR.
- Training Manager - a software application that will make available information on the certification and training status of any ECS operations and maintenance personnel, the availability of any training courses or prescribed training routines to be used by personnel, and other training related information.
- Policies and Procedures Manager - a software application that will provide ready access to the policies that are established by ESDIS. Procedures are also similarly published and distributed to enable ECS personnel to know how to perform procedural actions. The reuse of the document data server, ingest, client, and the use of the Bulletin Board will satisfy the functions of the Policies and Procedures Manager.

Each of these managers is described and discussed further in the following sections.

5.2 Management Logistics Context

MLCI's sole interface to other ECS CSCIs for the AM1/Landsat 7 release is through the MSS' Management Agent CSCI (MACI). Events along this interface (see Figure 5.2-1) permit system management of configuration management applications through monitoring of faults and transfer of historical event data for audits and fault analysis.

5.3 Configuration Management (CM) Object Model

In the following sections, the CM object model is shown in three views, each capturing the distinct aspects of one service manager's capabilities. While some object classes appear in multiple views, for simplicity each view shows only those operations and attributes that are pertinent to the capabilities being described in the view.

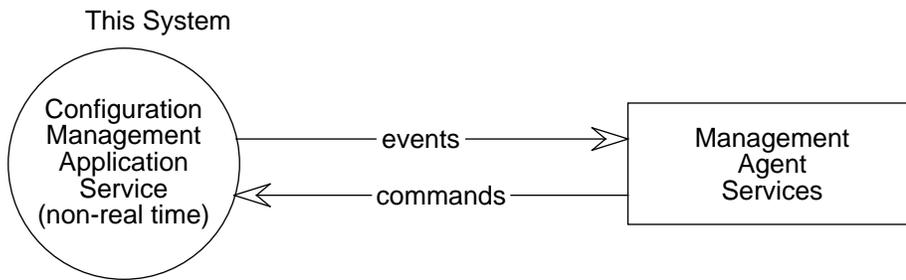


Figure 5.2-1. Management Logistics Context Diagram

5.3.1 Baseline Manager View

The Baseline Manager software, XRP-II, helps the DAACs, the EOC, and the SMC CM specialists maintain records that describe what comprises baselined operational system configurations (see Figure 5.3-1). These records identify the versions of hardware and software items that baselines contain as well as item interdependencies. They also track the identity and location of devices and subsystems the items comprise, maintain chronological histories of the change requests and resource changes new baselines incorporate, and maintain tracability of baselines to their predecessors and their associated system releases. XRP-II at the DAACs and EOC is used to maintain data about baselines deployed to and established at the site, while the application at the SMC maintains consolidated data about baselines system-wide. The software also produces a variety of reports, including a master index of the resources comprising ECS baselines system-wide, which can be accessed remotely or be posted to a configuration management folder on the ECS Bulletin Board for global viewing. The DAACs, EOC, and the SMC can exchange baseline records and reports electronically via formatted files.

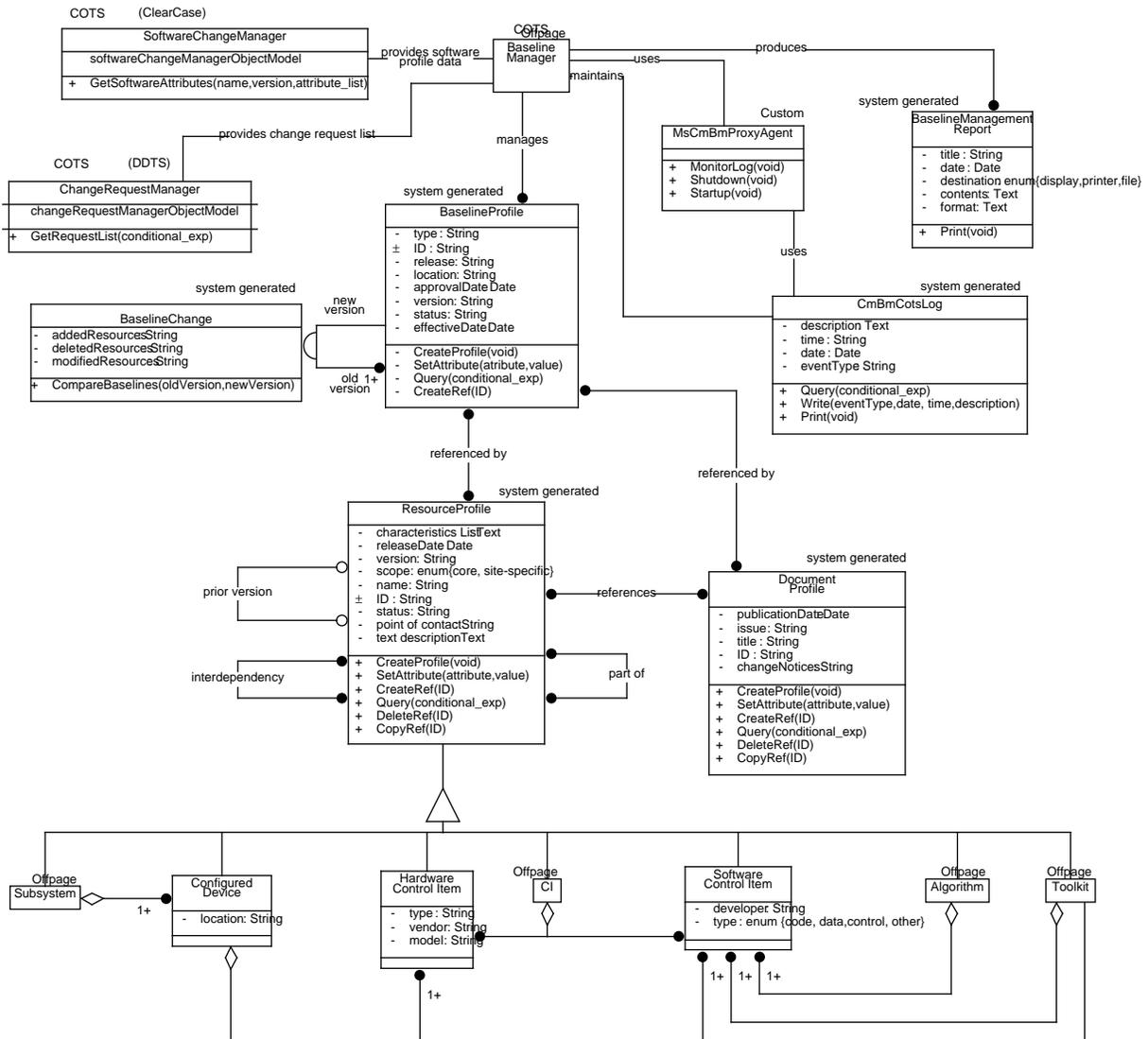


Figure 5.3-1. Baseline Manager Object Model

5.3.1.1 Algorithm Class

Parent Class: ResourceProfile

Public: No

Distributed Object: No

Purpose and Description:

This class is a specialization of the ResourceProfile class. It provides a capability to identify and characterize operational algorithms.

Attributes:

All Attributes inherited from parent class

Operations:

All Operations inherited from parent class

Associations:

The Algorithm class has associations with the following classes:
None

5.3.1.2 BaselineChange Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides a capability to compare two system baselines and identify what system resources changed.

Attributes:

addedResources - This attribute represents the list of the resources included in a baseline that were not in its predecessor.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

deletedResources - This attribute represents the list of the resources not in a baseline that were in its predecessor.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

modifiedResources - This attribute represents the list of the resources whose version in a baseline differs from that in the baseline's predecessor.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

Operations:

CompareBaselines - This operation builds a list of the difference in resources included in two specified baselines.

Arguments:oldVersion,newVersion

Return Type:Void

Privilege:Public

Associations:

The BaselineChange class has associations with the following classes:

None

5.3.1.3 BaselineManagementReport Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides the capability to retrieve and format historical system configuration details for display, storing, or printout.

Attributes:

contents - This attribute represents the data compiled and presented in a report.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

date - This attribute represents the date a report is effective.

Data Type:Date

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

destination - This attribute represents the device to which a report is sent.

Data Type:enum{display,printer,file}

Privilege:Private

Default Value:

Constraints:
Non Persistent Flag:False

format - This attribute represents the data defining the structure of a report.

Data Type:Text
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:

title - This attribute represents the name of a report.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:

Operations:

Print - This operation produces a report at the set destination.

Arguments:void
Return Type:Void
Privilege:Public

Associations:

The BaselineManagementReport class has associations with the following classes:

Class: BaselineManager produces

5.3.1.4 BaselineManager Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class is a service manager that controls configuration profiles of deployed system resources and the hierarchical organizations of resources constituting operational system baselines.

Attributes:

None

Operations:

None

Associations:

The BaselineManager class has associations with the following classes:

Class: CmBmCotsLog maintains

Class: BaselineProfile manages

Class: BaselineManagementReport produces

Class: ChangeRequestManager provideschangerequestlist

Class: SoftwareChangeManager providessoftwareprofiledata

Class: MsCmBmProxyAgent uses

5.3.1.5 BaselineProfile Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides a capability to uniquely identify system baselines and the versions of deployed resources that comprise each baseline.

Attributes:

ID - This attribute represents the code used to name a type of baseline.

Data Type:String

Privilege:Protected

Default Value:

Constraints:

Non Persistent Flag:False

approvalDate - This attribute represents the date a baseline is formally sanctioned by an approval authority.

Data Type:Date

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

effectiveDate - This attribute represents the date a baseline is placed in production.

Data Type:Date

Privilege:Private

Default Value:

Constraints:
Non Persistent Flag:False

location - This attribute represents a site at which a baseline is deployed.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:False

release - This attribute represents the identity of the ECS system release with which the baseline is associated.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:False

status - This attribute represents the code that identifies whether a baseline is in test, in production, or inactive.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:False

type - This attribute represents the classification that distinguishes among baselines according to user-specifiable designations.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:False

version - This attribute represents the current revision to an established baseline.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:False

Operations:

CreateProfile - This operation builds a single record of baseline data and stores it in the baseline manager database.

Arguments:void

Return Type:Void

Privilege:Private

CreateRef - This operation creates and stores a reference between two objects in accordance with database referential integrity rules.

Arguments:ID

Return Type:Void

Privilege:Private

Query - This operation retrieves specified objects and attribute values from the baseline manager database.

Arguments:conditional_exp

Return Type:Void

Privilege:Private

SetAttribute - This operation assigns a named attribute a specified value.

Arguments:atribute,value

Return Type:Void

Privilege:Private

Associations:

The BaselineProfile class has associations with the following classes:

Class: BaselineProfile -

Class: BaselineManager manages

Class: DocumentProfile referencedby

Class: ResourceProfile referencedby

5.3.1.6 CI Class

Parent Class:ResourceProfile

Public:No

Distributed Object:No

Purpose and Description:

This class is a specialization of the ResourceProfile class. It provides a capability to identify ECS Configuration Items to which deployed hardware and software belong.

Attributes:

All Attributes inherited from parent class

Operations:

All Operations inherited from parent class

Associations:

The CI class has associations with the following classes:

None

5.3.1.7 ChangeRequestManager Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class is a service manager that controls change request data and manages operator-initiated actions associated with it.

Attributes:

changeRequestManagerObjectModel

Operations:

GetRequestList - This operation obtains values for specified attributes of specified resource change requests.

Arguments:conditional_exp

Return Type:Void

Privilege:Public

Associations:

The ChangeRequestManager class has associations with the following classes:

Class: BaselineManager provideschangerequestlist

5.3.1.8 CmBmCotsLog Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides the capability to maintain a record of error and library modification events.

Attributes:

date - This attribute represents the date on which an event occurred.

Data Type:Date

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

description - This attribute represents the narrative that explains the nature of an event.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

eventType - This attribute represents the classification of a system event according to source and required action.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

time - This attribute represents the time at which an event occurred.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

Operations:

Print - This operation sends an entry from the Baseline Manager log to an output device.

Arguments:void

Return Type:Void

Privilege:Public

Query - This operation reads an entry from the Baseline Manager log.

Arguments:conditional_exp

Return Type:Void

Privilege:Public

Write - This operation adds an entry to the Baseline Manager log.

Arguments:eventType,date, time,description
Return Type:Void
Privilege:Public

Associations:

The CmBmCotsLog class has associations with the following classes:

Class: BaselineManager maintains
Class: MsCmBmProxyAgent uses

5.3.1.9 ConfiguredDevice Class

Parent Class:ResourceProfile
Public:No
Distributed Object:No
Purpose and Description:

This class is a specialization of the ResourceProfile class. It provides a capability to identify and characterize individual, deployed configured devices, and it identifies the hardware and software items to comprise each.

Attributes:

location - This attribute represents the place at a site where an integrated operational ECS device can be found.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:

Operations:

All Operations inherited from parent class

Associations:

The ConfiguredDevice class has associations with the following classes:
Subsystem (Aggregation)

5.3.1.10 DocumentProfile Class

Parent Class:Not Applicable
Public:No
Distributed Object:No

Purpose and Description:

This class provides a capability to maintain key, technical information about deployed system documentation. It tracks issues of system documents and relates them to specific system baselines and resources.

Attributes:

ID - This attribute represents the code that uniquely identifies system documentation.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

changeNotices - This attribute represents the list of document change notices included in the current edition of a document.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

issue - This attribute represents the nomenclature used to distinguish among versions of a single edition of a document.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

publicationDate - This attribute represents the date associated with a document.

Data Type:Date

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

title - This attribute represents the nomenclature that distinguishes document volumes from one another.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

Operations:

CopyRef - This operation creates and stores a set of references between two objects identical to the references between one of the objects and a third.

Arguments:ID

Return Type:Void

Privilege:Public

CreateProfile - This operation builds a single record of baseline data and stores it in the baseline manager database.

Arguments:void

Return Type:Void

Privilege:Public

CreateRef - This operation creates and stores a reference between two objects in accordance with database referential integrity rules.

Arguments:ID

Return Type:Void

Privilege:Public

DeleteRef - This operation removes a reference between two objects in accordance with database referential integrity rules.

Arguments:ID

Return Type:Void

Privilege:Public

Query - This operation retrieves specified objects and attribute values from the baseline manager database.

Arguments:conditional_exp

Return Type:Void

Privilege:Public

SetAttribute - This operation assigns a named attribute a specified value.

Arguments:attribute,value

Return Type:Void

Privilege:Public

Associations:

The DocumentProfile class has associations with the following classes:

Class: BaselineProfile referencedby

Class: ResourceProfile references

5.3.1.11 HardwareControlItem Class

Parent Class:ResourceProfile

Public:No

Distributed Object:No

Purpose and Description:

This class is a specialization of the ResourceProfile class. It provides a capability to identify and characterize individual, deployed hardware items.

Attributes:

model - This attribute represents the identifier for a single type of item of a manufacturer's product line.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

type - This attribute represents the classification that distinguishes hardware components according to function.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

vendor - This attribute represents the name of a hardware component's manufacturer.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

Operations:

All Operations inherited from parent class

Associations:

The HardwareControlItem class has associations with the following classes:

CI (Aggregation)

ConfiguredDevice (Aggregation)

5.3.1.12 MsCmBmProxyAgent Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides a system management interface for lifecycle services, event reporting, and instrumentation.

Attributes:

None

Operations:

MonitorLog - This operation provides event data for the application manager.

Arguments:void

Return Type:Void

Privilege:Public

Shutdown - This operation results in termination of Baseline Manager server processes.

Arguments:void

Return Type:Void

Privilege:Public

Startup - This operation results in the start of Baseline Manager servers and associated file system setup tasks.

Arguments:void

Return Type:Void

Privilege:Public

Associations:

The MsCmBmProxyAgent class has associations with the following classes:

Class: BaselineManager uses

Class: CmBmCotsLog uses

5.3.1.13 ResourceProfile Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides a capability to maintain key, technical information about deployed

hardware and software resources that both identifies and characterizes them.

Attributes:

ID - This attribute represents the code that uniquely identifies a resource profile.

Data Type:String

Privilege:Protected

Default Value:

Constraints:

Non Persistent Flag:

characteristics List - This attribute represents the collection of information describing key aspects of a profiled resource's configuration.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

name - This attribute represents the nomenclature used to identify a profiled resource.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

point of contact - This attribute represents the name of a designated individual associated with the profiled resource.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

releaseDate - This attribute represents the date the profiled resource was released to production.

Data Type:Date

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

scope - This attribute represents the classification of a profiled resource according to the number of sites where it is deployed.

Data Type:enum{core, site-specific}

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

status - This attribute represents the code that identifies the stage to which a profiled resource has reached in its lifecycle.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

text description - This attribute represents the narrative that describes a profiled resource.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

version - This attribute represents the current revision to a profiled resource.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

Operations:

CopyRef - This operation creates and stores a set of references between two objects identical to the references between one of the objects and a third.

Arguments:ID

Return Type:Void

Privilege:Public

CreateProfile - This operation builds a single record of baseline data and stores it in the baseline manager database.

Arguments:void

Return Type:Void

Privilege:Public

CreateRef - This operation creates and stores a reference between two objects in accordance with database referential integrity rules.

Arguments:ID
Return Type:Void
Privilege:Public

DeleteRef - This operation removes a reference between two objects in accordance with database referential integrity rules.

Arguments:ID
Return Type:Void
Privilege:Public

Query - This operation retrieves specified objects and attribute values from the Baseline Manager database.

Arguments:conditional_exp
Return Type:Void
Privilege:Public

SetAttribute - This operation assigns a named attribute a specified value.

Arguments:attribute,value
Return Type:Void
Privilege:Public

Associations:

The ResourceProfile class has associations with the following classes:

Class: ResourceProfile interdependency
Class: ResourceProfile partof
Class: ResourceProfile priorversion
Class: BaselineProfile referencedby
Class: DocumentProfile references

5.3.1.14 SoftwareChangeManager Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class is a server manager that controls a software library and manages the execution of operator initiated events associated with it.

Attributes:

softwareChangeManagerObjectModel

Operations:

GetSoftwareAttributes - This operation obtains values for specified attributes of specified library files.

Arguments:name,version,attribute_list

Return Type:Void

Privilege:Public

Associations:

The SoftwareChangeManager class has associations with the following classes:

Class: BaselineManager providessoftwareprofiledata

5.3.1.15 SoftwareControlltem Class

Parent Class:ResourceProfile

Public:No

Distributed Object:No

Purpose and Description:

This class is a specialization of the ResourceProfile class. It provides a capability to identify and characterize individual, deployed software items.

Attributes:

developer - This attribute represents the organization that developed the profiled resource.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

type - This attribute represents the classification that distinguishes among the type of information contained in a profiled, software component resource.

Data Type:enum {code, data,control, other}

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

Operations:

All Operations inherited from parent class

Associations:

The SoftwareControlItem class has associations with the following classes:

- Algorithm (Aggregation)
- CI (Aggregation)
- ConfiguredDevice (Aggregation)
- Toolkit (Aggregation)

5.3.1.16 Subsystem Class

Parent Class:ResourceProfile

Public:No

Distributed Object:No

Purpose and Description:

This class is a specialization of the ResourceProfile class. It provides a capability to identify and characterize individual, deployed subsystems and the configured devices that comprise each.

Attributes:

All Attributes inherited from parent class

Operations:

All Operations inherited from parent class

Associations:

The Subsystem class has associations with the following classes:

None

5.3.1.17 Toolkit Class

Parent Class:ResourceProfile

Public:No

Distributed Object:No

Purpose and Description:

This class is a specialization of the ResourceProfile class. It provides a capability to identify and characterize versions of ECS toolkits. It also identifies the ECS devices on which each toolkit resides.

Attributes:

All Attributes inherited from parent class

Operations:

All Operations inherited from parent class

Associations:

The Toolkit class has associations with the following classes:
ConfiguredDevice (Aggregation)

5.3.2 Software Change Manager View

ClearCase, the Software Change Manager, provides version control for custom software, data, and documentation files (see Figure 5.3-2). It stores these files in its software library. At the SMC the *ClearCase* library will contain the master copy of all software that is deployed to the sites. *ClearCase* libraries at the DAACs will contain the software deployed to the site as well as files and data created when tailoring the software for the local operating environment.

ClearCase's library organizes the files and automatically assigns new file version identifiers whenever a change is checked in. Associations can be created so related files and their attributes can be collectively read, archived, checked in and out, and reported on.

Views are the means by which operators select the file versions on which to work. Operations on an object in a view are regulated through use of access profiles that map userids against operations the userids are authorized to perform. (Restrictions can be incrementally tightened as a version is promoted in stages to production.) *ClearCase* checks out (signs out) files to be modified to reduce the risk of losing changes and helps detect conflicting changes when people work on a version in parallel. *ClearCase* also constructs builds and generates build records so that builds can be re-created.

ClearCase interfaces with XRP-II and *DDTS*. In response to a *GetSoftwareAttributes* request, it exports to XRP-II attributes that describe software, data, and documentation in its libraries. The interface with *DDTS* consists of a *GetVerification* request to evaluate a conditional expression for verifying that change to a file version has been approved.

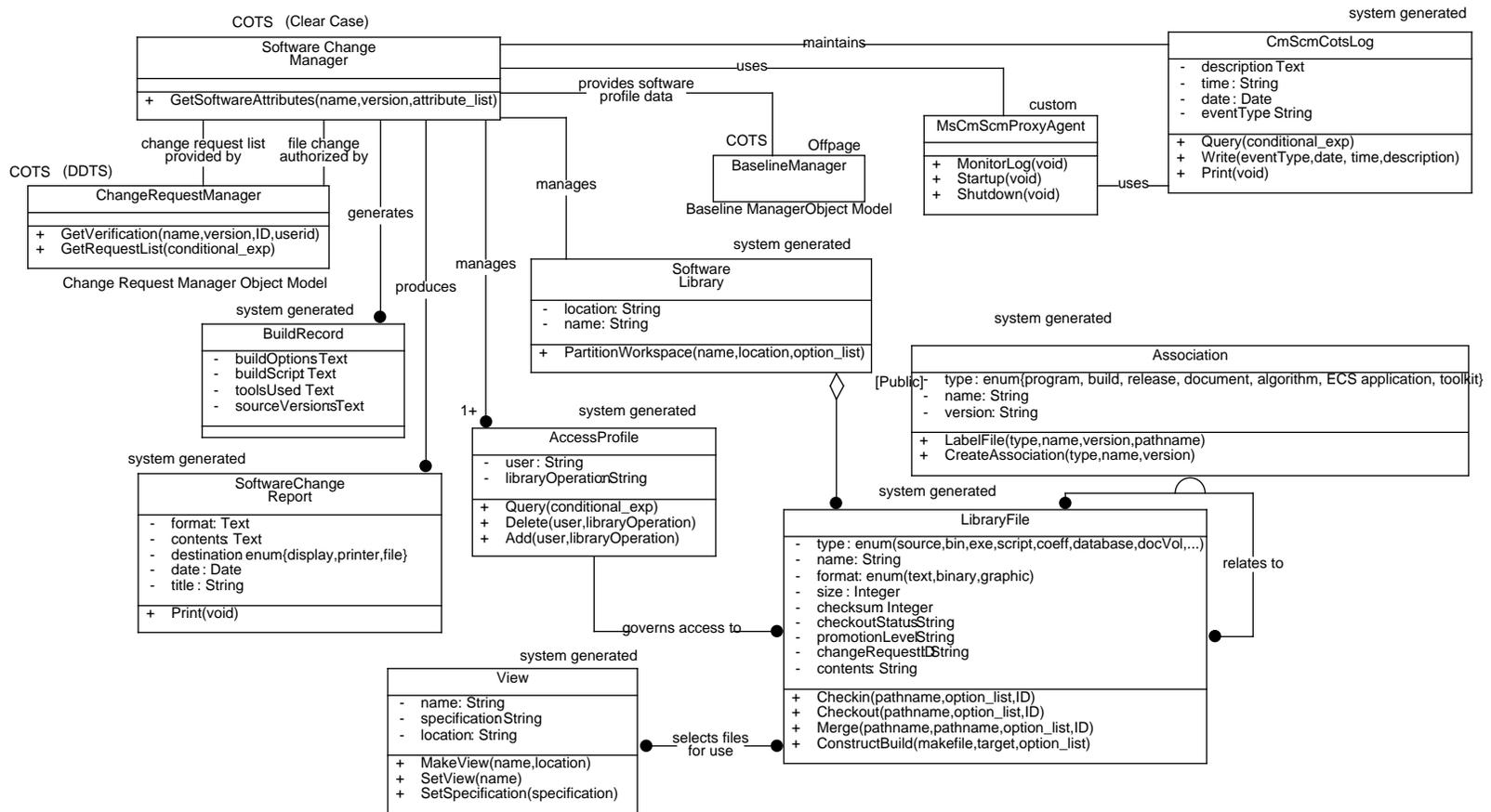


Figure 5.3-2. Software Change Manager Object Model

5.3.2.1 AccessProfile Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides the capability to maintain permissions that authorize individuals and groups to perform specific library operations.

Attributes:

libraryOperation - This attribute represents the action performed on library files.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

user - This attribute represents the UNIX logon identifier or group name to which a set of permissions can be assigned.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

Operations:

Add - This operation creates a record describing a specified operation that a specified individual or group can perform on a specified system entity.

Arguments:user,libraryOperation

Return Type:Void

Privilege:Public

Delete - This operation removes a record describing a specified operation that a specified individual or group can perform on a specified system entity.

Arguments:user,libraryOperation

Return Type:Void

Privilege:Public

Query - This operation retrieves a record describing a specified operation that a specified individual or group can perform on a specified system entity.

Arguments:conditional_exp

Return Type:Void

Privilege:Public

Associations:

The AccessProfile class has associations with the following classes:

Class: LibraryFile governsaccessto

Class: SoftwareChangeManager manages

5.3.2.2 Association Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides the capability to tag related library files in order to facilitate group operations.

Attributes:

name - This attribute represents the nomenclature for an individual collection of system data.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

type - This attribute represents the classification that distinguishes among groups of related library files on which operations can be performed collectively.

Data Type:enum{program, build, release, document, algorithm, ECS application, toolkit}

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

version - This attribute represents the rev level of a group of related resources.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

Operations:

CreateAssociation - This operation creates and stores the specification for a named association type.

Arguments:type,name,version

Return Type:Void

Privilege:Public

LabelFile - This operation establishes an association between two or more library files.

Arguments:type,name,version,pathname

Return Type:Void

Privilege:Public

Associations:

The Association class has associations with the following classes:

None

5.3.2.3 BaselineManager Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class is a service manager that controls configuration profiles of deployed system resources and the hierarchical organizations of resources constituting operational system baselines.

Attributes:

None

Operations:

None

Associations:

The BaselineManager class has associations with the following classes:

Class: SoftwareChangeManager providessoftwareprofiledata

5.3.2.4 BuildRecord Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides the capability to audit and record the files, tools, text and options used in constructing a build.

Attributes:

buildOptions - This attribute represents the collection of information about management options invoked in executing the build.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

buildScript - This attribute represents the listing of the text in a build script.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

sourceVersions - This attribute represents the collection of information identifying the version of source files used in constructing a build.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

toolsUsed - This attribute represents the collection of information identifying the version of tools used in constructing a build.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

Operations:

None

Associations:

The BuildRecord class has associations with the following classes:

Class: SoftwareChangeManager generates

5.3.2.5 ChangeRequestManager Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class is a service manager that controls change request data and manages operator-initiated actions associated with it.

Attributes:

None

Operations:

GetRequestList - This operation obtains values for specified attributes of specified resource change requests.

Arguments:conditional_exp

Return Type:Void

Privilege:Public

GetVerification - This operation evaluates a specified conditional expression on resource change request records and returns a status code.

Arguments:name,version,ID,userid

Return Type:Void

Privilege:Public

Associations:

The ChangeRequestManager class has associations with the following classes:

Class: SoftwareChangeManager changerequestlistprovidedby

Class: SoftwareChangeManager filechangeauthorizedby

5.3.2.6 CmScmCotsLog Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides the capability to maintain a record of error and library modification events.

Attributes:

date - This attribute represents the date on which an event occurred.

Data Type:Date

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

description - This attribute represents the narrative that explains the nature of an event.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

eventType - This attribute represents the classification of a system event according to source and required action.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

time - This attribute represents the time at which an event occurred.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

Operations:

Print - This operation sends an entry from the Software Change Manager log to an output device.

Arguments:void

Return Type:Void

Privilege:Public

Query - This operation reads an entry from the Software Change Manager log.

Arguments:conditional_exp

Return Type:Void

Privilege:Public

Write - This operation adds an entry to the Software Change Manager log.
Arguments: eventType, date, time, description
Return Type: Void
Privilege: Public

Associations:

The CmScmCotsLog class has associations with the following classes:
Class: SoftwareChangeManager maintains
Class: MsCmScmProxyAgent uses

5.3.2.7 LibraryFile Class

Parent Class: Not Applicable
Public: No
Distributed Object: No
Purpose and Description:

This class provides the capability to hold and control versions of software, documentation, control data and test data.

Attributes:

changeRequestID - This attribute represents the collection of resource change request identifiers used in authorizing checkin of a new library file.

Data Type: String
Privilege: Private
Default Value:
Constraints:
Non Persistent Flag:

checkoutStatus - This attribute represents the code that identifies whether or not a library file is in the process of being modified.

Data Type: String
Privilege: Private
Default Value:
Constraints:
Non Persistent Flag:

checksum - This attribute represents the sum of the number of bits in a library file.

Data Type: Integer
Privilege: Private
Default Value:
Constraints:
Non Persistent Flag:

contents - This attribute represents the collection of data stored in a library file.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

format - This attribute represents the classification that distinguishes among library files based on the way its contents are encoded

Data Type:enum(text,binary,graphic)

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

name - This attribute represents the nomenclature for an individual collection of system data.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

promotionLevel - This attribute code identifies the lifecycle stage that the contents of a library file has reached.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

size - This attribute represents the number of bytes a file contains.

Data Type:Integer

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

type - This attribute represents the classification of a library file according to kind of contents.

Data Type:enum(source,bin,exe,script,coeff,database,docVol,...)

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

Operations:

Checkin - This operation stores a new version of a file in the library.

Arguments:pathname,option_list,ID

Return Type:Void

Privilege:Public

Checkout - This operation makes a copy of a library file available for modification.

Arguments:pathname,option_list,ID

Return Type:Void

Privilege:Public

ConstructBuild - This operation executes a build script, performing one or more target rebuilds, and creates an audit record of the build.

Arguments:makefile,target,option_list

Return Type:Void

Privilege:Public

Merge - This operation forms new contents of a library file by combining two or more sets of changes that had been made to that file in parallel.

Arguments:pathname,pathname,option_list,ID

Return Type:Void

Privilege:Public

Associations:

The LibraryFile class has associations with the following classes:

Class: AccessProfile governsaccessto

Class: LibraryFile relateto

Class: View selectsfilesforuse

SoftwareLibrary (Aggregation)

5.3.2.8 MsCmScmProxyAgent Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides a system management interface for lifecycle services, event reporting, and instrumentation.

Attributes:

None

Operations:

MonitorLog - This operation provides event data for the application manager.

Arguments:void

Return Type:Void

Privilege:Public

Shutdown - This operation results in termination of Software Change Manager server processes.

Arguments:void

Return Type:Void

Privilege:Public

Startup - This operation results in the start of Software Change Manager servers and associated file system setup tasks.

Arguments:void

Return Type:Void

Privilege:Public

Associations:

The MsCmScmProxyAgent class has associations with the following classes:

Class: CmScmCotsLog uses

Class: SoftwareChangeManager uses

5.3.2.9 SoftwareChangeManager Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class is a server manager that controls a software library and manages the execution of operator initiated events associated with it.

Attributes:

None

Operations:

GetSoftwareAttributes - This operation obtains values for specified attributes of specified library files.

Arguments:name,version,attribute_list

Return Type:Void

Privilege:Public

Associations:

The SoftwareChangeManager class has associations with the following classes:

Class: ChangeRequestManager changerequestlistprovidedby

Class: ChangeRequestManager filechangeauthorizedby

Class: BuildRecord generates

Class: CmScmCotsLog maintains

Class: AccessProfile manages

Class: SoftwareLibrary manages

Class: SoftwareChangeReport produces

Class: BaselineManager providessoftwareprofiledata

Class: MsCmScmProxyAgent uses

5.3.2.10 SoftwareChangeReport Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides the capability to format and order selected, software library data for display, storing, or printout.

Attributes:

contents - This attribute represents the data compiled and presented in a report.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

date - This attribute represents the date a report is effective.

Data Type:Date

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

destination - This attribute represents the device to which a report is sent.

Data Type:enum{display,printer,file}

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

format - This attribute represents the data defining the structure of a report.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

title - This attribute represents the name of a report.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

Operations:

Print - This operation produces a report at the set destination.

Arguments:void

Return Type:Void

Privilege:Public

Associations:

The SoftwareChangeReport class has associations with the following classes:

Class: SoftwareChangeManager produces

5.3.2.11 SoftwareLibrary Class

Parent Class:Not Applicable

Public:Yes

Distributed Object:No

Purpose and Description:

This class provides the capability to organize and manage the physical placement of collections of files containing ECS custom software, scientific software, and associated test data, control data, and documentation.

Attributes:

location - This attribute represents the site at which the software library is located.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

name - This attribute represents the nomenclature for an individual collection of system data.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

Operations:

PartitionWorkspace - This operation configures and administers the software library's structure.

Arguments:name,location,option_list

Return Type:Void

Privilege:Public

Associations:

The SoftwareLibrary class has associations with the following classes:

Class: SoftwareChangeManager manages

5.3.2.12View Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class maintains sets of specifications operators use to select the version of library objects on which to work.

Attributes:

location - This attribute represents the network region, host, and local pathname at which the view is stored.

Data Type:String

Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:False

name - This attribute represents the nomenclature for an individual collection of system data.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:False

specification - This attribute represents a conditional expression for qualifying library objects for inclusion in a view.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:False

Operations:

MakeView - This operation creates a default view, assigning it a specified name.

Arguments:name,location
Return Type:Void
Privilege:Public

SetSpecification - This operation updates the specification contained in a named view.

Arguments:specification
Return Type:Void
Privilege:Public

SetView - This operation selects the library files that satisfy the named view's specification and sets them as the context for library operations.

Arguments:name
Return Type:Void
Privilege:Public

Associations:

The View class has associations with the following classes:
Class: LibraryFile selectsfilesforuse

5.3.3 Change Request Manager View

DDTS, the Change Request Manager, maintains proposals for changing the configuration of the ECS and tracks their implementation (see Figure 5.3-3). CM specialists interact with the application to compose, register, and track the status of Resource Change Requests electronically. Several types of change requests exist, such as configuration change requests, non-conformance reports, deviations, and waivers. Amendments to change requests are versions of the initial request, and each request may include electronic attachments such as supporting data and impact assessments. Multiple data stores can exist for change requests, and change requests may be mailed among them electronically. Originators are notified when events occur. Reports containing lists of change requests, their descriptions, and their status are produced such that they can be posted to the ECS Bulletin Board.

DDTS interacts with *XRP-II* and *ClearCase*. It responds to `GetRequestList` with resource change request data that satisfy requester-specified criteria, and it answers `GetVerification` requests by returning a status reflecting the result of evaluating a requester-supplied conditional expression.

5.3.3.1 Attachment Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides a capability to maintain lists of files that contain support information associated with specific change requests.

Attributes:

name - This attribute represents the nomenclature used to identify a profiled resource.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

Operations:

SetName - This operation records a specified name for a specified attachment.

Arguments:name

Return Type:Void

Privilege:Public

Associations:

The Attachment class has associations with the following classes:

Class: ResourceChangeRequest includes

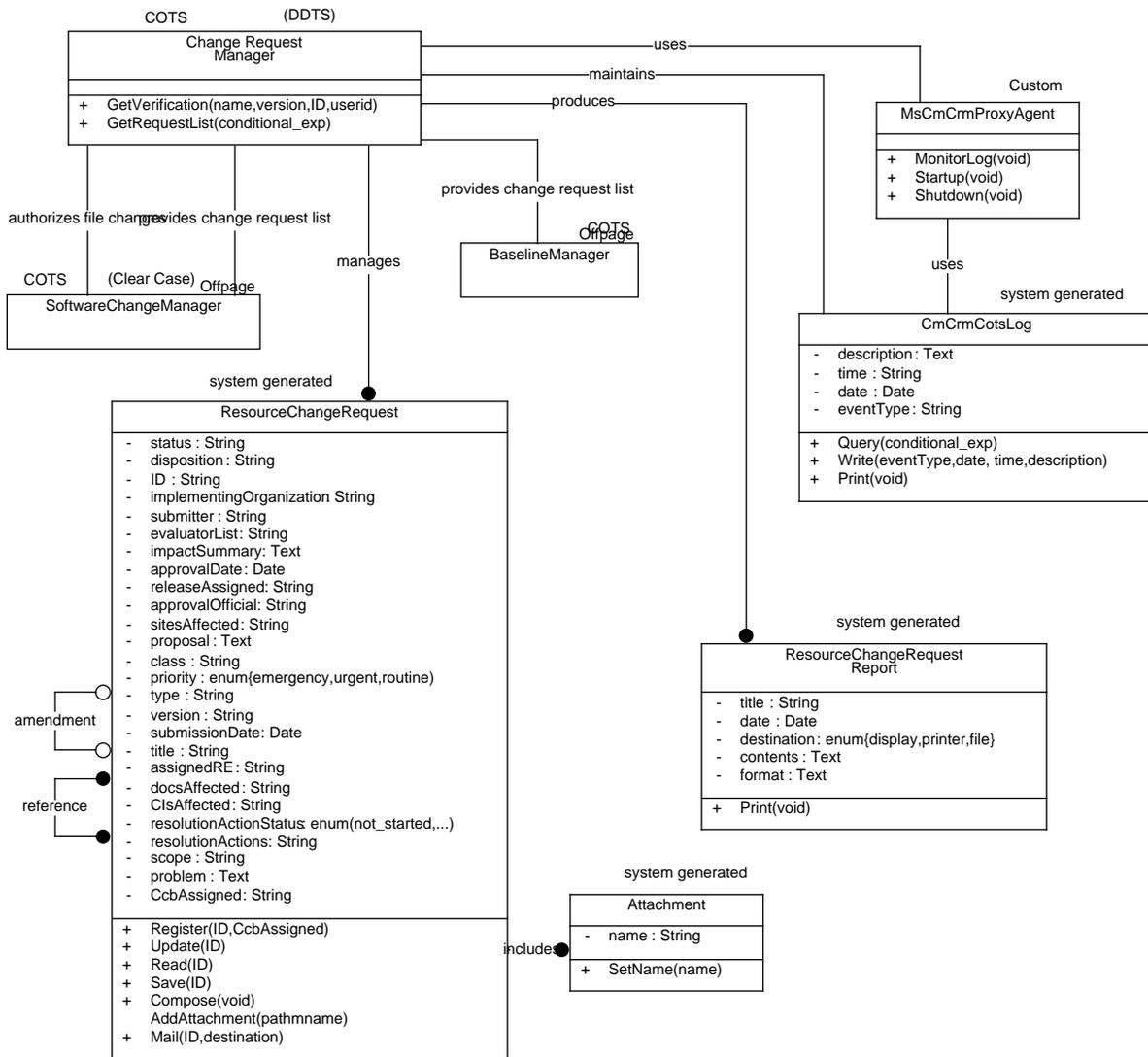


Figure 5.3-3. Change Request Manager Object Model

5.3.3.2 BaselineManager Class

Parent Class: Not Applicable

Public: No

Distributed Object: No

Purpose and Description:

This class is a service manager that controls configuration profiles of deployed system resources and the hierarchical organizations of resources constituting operational system baselines.

Attributes:

None

Operations:

None

Associations:

The BaselineManager class has associations with the following classes:

Class: ChangeRequestManager provideschangerequestlist

5.3.3.3 ChangeRequestManager Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class is a service manager that controls change request data and manages operator-initiated actions associated with it.

Attributes:

None

Operations:

GetRequestList - This operation obtains values for specified attributes of specified resource change requests.

Arguments:conditional_exp

Return Type:Void

Privilege:Public

GetVerification - This operation evaluates a specified conditional expression on resource change request records and returns a status code.

Arguments:name,version,ID,userid

Return Type:Void

Privilege:Public

Associations:

The ChangeRequestManager class has associations with the following classes:

Class: SoftwareChangeManager authorizesfilechanges

Class: CmCrmCotsLog maintains

Class: ResourceChangeRequest manages

Class: ResourceChangeRequestReport produces

Class: BaselineManager provideschangerequestlist

Class: SoftwareChangeManager provideschangerequestlist

Class: MsCmCrmProxyAgent uses

5.3.3.4 CmCrmCotsLog Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides the capability to maintain a record of error and library modification events.

Attributes:

date - This attribute represents the date on which an event occurred.

Data Type:Date

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

description - This attribute represents the narrative that explains the nature of an event.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

eventType - This attribute represents the classification of a system event according to source and required action.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

time - This attribute represents the time at which an event occurred.

Data Type:String

Privilege:Private

Default Value:
Constraints:
Non Persistent Flag:False

Operations:

Print - This operation sends an entry from the Change Request Manager log to an output device.

Arguments:void
Return Type:Void
Privilege:Public

Query - This operation reads an entry from the Change Request Manager log.

Arguments:conditional_exp
Return Type:Void
Privilege:Public

Write - This operation adds an entry to the Change Request Manager log.

Arguments:eventType,date, time,description
Return Type:Void
Privilege:Public

Associations:

The CmCrmCotsLog class has associations with the following classes:

Class: ChangeRequestManager maintains
Class: MsCmCrmProxyAgent uses

5.3.3.5 MsCmCrmProxyAgent Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides a system management interface for lifecycle services, event reporting, and instrumentation.

Attributes:

None

Operations:

MonitorLog - This operation provides event data for the application manager.

Arguments:void

Return Type:Void

Privilege:Public

Shutdown - This operation results in termination of server processes.

Arguments:void

Return Type:Void

Privilege:Public

Startup - This operation results in the start of Change Request Manager servers and associated file system setup tasks.

Arguments:void

Return Type:Void

Privilege:Public

Associations:

The MsCmCrmProxyAgent class has associations with the following classes:

Class: ChangeRequestManager uses

Class: CmCrmCotsLog uses

5.3.3.6 ResourceChangeRequest Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides a capability to maintain records that describe and status proposed changes to ECS resources.

Attributes:

CIsAffected - This attribute represents the list of the configuration items whose configuration is affected by a proposed system change.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

CcbAssigned - This attribute represents the name of a configuration control board having authority to approve the resource change request.

Data Type:String

Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:False

ID - This attribute represents a unique identifier for a resource change request.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:False

approvalDate - This attribute represents the date a final decision was made concerning a proposed system change.

Data Type:Date
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:

approvalOfficial - This attribute is the name of the individual whose decision is reflected in the proposed change's disposition.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:

assignedRE - This attribute represents the responsible engineer designated to analyze and/or implement a proposed system change.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:

class - This attribute is the classification that distinguishes change requests according to management level needed for approval.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:

disposition - This attribute represents the final decision made by a designated approval

official concerning a proposed system change.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:False

docsAffected - This attribute represents the list of the system documents affected by a proposed system change.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

evaluatorList - This attribute is collection of names of organizations designated to assess the impact of a proposed system change.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

impactSummary - This attribute is the information that summarizes assessments of the impact of a proposed change.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

implementingOrganization - This attribute is the name of the organization assigned to implement a proposed change.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

priority - This attribute is the urgency with which a proposed change is needed.

Data Type:enum{emergency,urgent,routine}

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

problem - This attribute represents the narrative description of the circumstances justifying a configuration change.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

proposal - This attribute is the narrative description of the proposed change.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

releaseAssigned - This attribute represents the code for the ECS release in which a proposed change is targeted for implementation.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

resolutionActionStatus - This attribute represents the stage reached in accomplishing an action associated with resolving a system non-conformance.

Data Type:enum(not_started,...)

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

resolutionActions - This attribute represents the list of actions associated with resolving a system non-conformance.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

scope - This attribute represents the range of system baselines affected by a proposed change.

Data Type:String

Privilege:Private

Default Value:

Constraints:
Non Persistent Flag:

sitesAffected - This attribute is the collection of names of ECS sites affected by a proposed change.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

status - This attribute identifies the stage a proposed change has reached in its lifecycle.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

submissionDate - This attribute represents the date a proposed change was first registered.

Data Type:Date

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

submitter - This attribute is a name of the individual who registers a proposed change.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

title - This attribute is the nomenclature used to identify the proposed change.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

type - This attribute is the classification that distinguishes among change requests according to form used.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

version - This attribute is the current revision/amendment to a proposed change.

Data Type:String

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

Operations:

AddAttachment

Arguments:pathmname

Compose - This operation obtains and compiles information to be stored in a change request record.

Arguments:void

Return Type:Void

Privilege:Public

Mail - This operation uses Unix mail utility to send a notification of a CCR/NCR/DR to a specified address.

Arguments:ID,destination

Return Type:Void

Privilege:Public

Read - This operation gets the contents of a stored change request record for display.

Arguments:ID

Return Type:Void

Privilege:Public

Register - This operation forwards a change request record to the SMC.

Arguments:ID,CcbAssigned

Return Type:Void

Privilege:Public

Save - This operation stores a change request record in a sequential file.

Arguments:ID

Return Type:Void

Privilege:Public

Update - This operation stores a new or modified change request record in the change request database.

Arguments:ID

Return Type:Void
Privilege:Public

Associations:

The ResourceChangeRequest class has associations with the following classes:

Class: ResourceChangeRequest amendment
Class: Attachment includes
Class: ChangeRequestManager manages
Class: ResourceChangeRequest reference

5.3.3.7 ResourceChangeRequestReport Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides the capability to retrieve and format change request details and metrics for display, storing, or printout.

Attributes:

contents - This attribute represents the data compiled and presented in a report.

Data Type:Text

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

date - This attribute represents the date a report is effective.

Data Type:Date

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

destination - This attribute represents a device to which a report is sent.

Data Type:enum{display,printer,file}

Privilege:Private

Default Value:

Constraints:

Non Persistent Flag:

format - This attribute represents the data defining the structure of a report.

Data Type:Text
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:

title - This attribute represents a name of a report.

Data Type:String
Privilege:Private
Default Value:
Constraints:
Non Persistent Flag:

Operations:

Print - This operation produces a report at the set destination.

Arguments:void
Return Type:Void
Privilege:Public

Associations:

The ResourceChangeRequestReport class has associations with the following classes:

Class: ChangeRequestManager produces

5.3.3.8 SoftwareChangeManager Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class is a server manager that controls a software library and manages the execution of operator initiated events associated with it.

Attributes:

None

Operations:

None

Associations:

The SoftwareChangeManager class has associations with the following classes:

Class: ChangeRequestManager authorizesfilechanges

Class: ChangeRequestManager provideschangerequestlist

5.4 Configuration Management Dynamic Model

(The scenarios that follow conform to the policies and methodologies contained in 194-102-MG1-001, Configuration Management Plan for the ECS Project.)

5.4.1 Ingest Algorithm

This scenario traces the actions associated with receipt and tracking of scientific software at the sites. The scenario is depicted in Figure 5.4-1.

5.4.1.1 Beginning Assumptions

The algorithm is new.

5.4.1.2 Interfaces with Other Subsystems and Segments

None.

5.4.1.3 Stimulus

An algorithm delivery package is received by file transfer at a DAAC from an SCF and is ready to be stored in the DAAC's EOSDIS software library.

5.4.1.4 Participating Classes From the Object Model

SoftwareChangeManager

View

SoftwareLibrary

ChangeRequestManager

LibraryFile

CmScmCotsLog

BuildRecord

AccessProfile

Association

BaselineManager

BaselineProfile

ResourceProfile

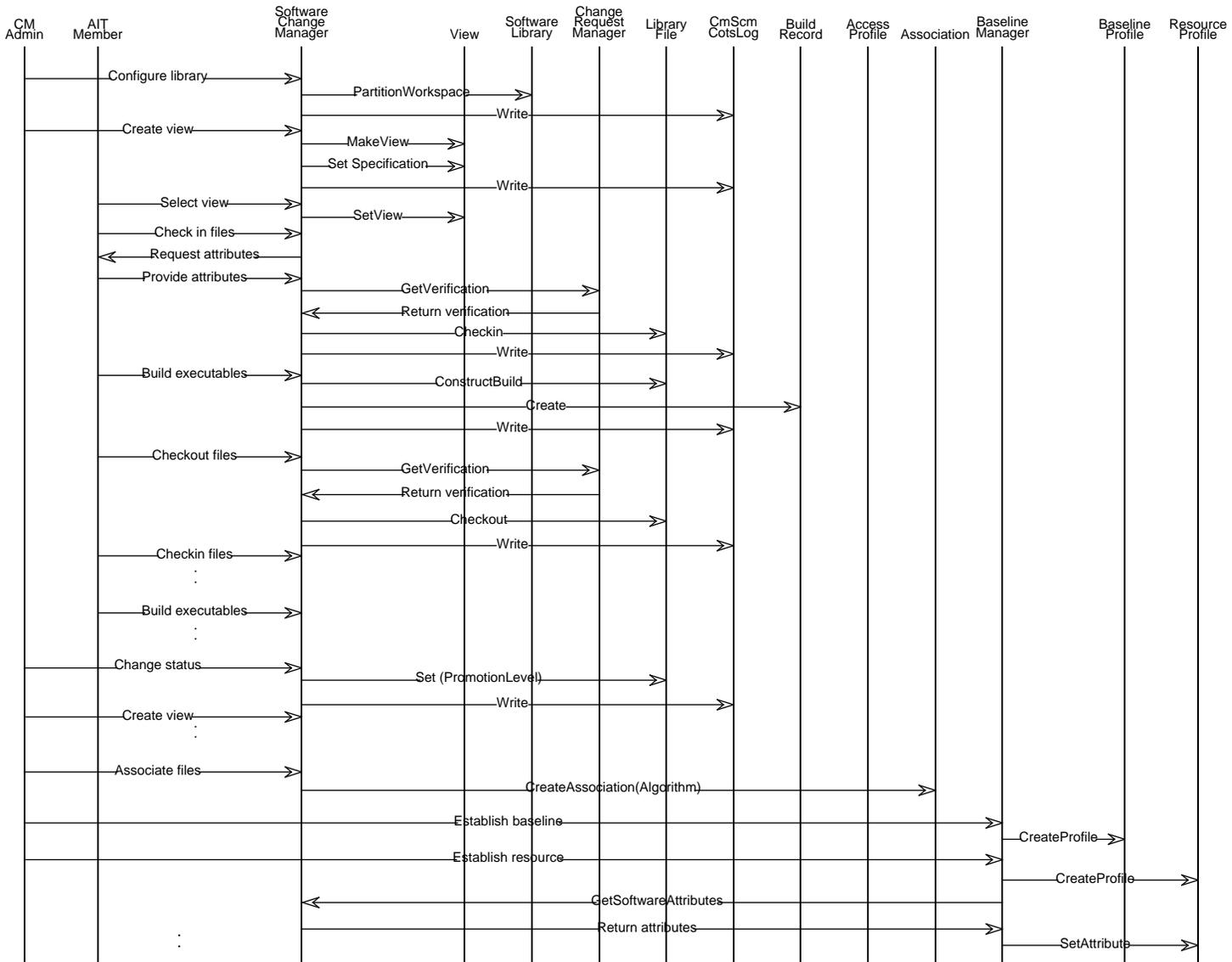


Figure 5.4-1. Software Change Management Scenario: Ingest Algorithm

5.4.1.5 Beginning System, Segment and Subsystem State(s)

SDP Toolkit exists in site software library.

Delivery package contains files to be stored and controlled.

A Resource Change Request (of type CCR) has been prepared by the site, staffed, approved for implementation at the DAAC, and registered in the SMC's Configuration Change Request (CCR) database.

5.4.1.6 Ending State

Algorithm files are stored in library and secured from change by other than CM staff.

Records exist with which to re-create builds.

Site and SMC baseline data reflect the new algorithm.

Configuration change request is closed.

5.4.1.7 Scenario Description

The CM specialist interacts with *ClearCase*, the Software Change Manager, to configure the site library, establishing branches, elements, and a view for the algorithm files and access permissions for algorithm integration team (AIT) members. The library change events are logged.

AIT members use *ClearCase* to select the view and check algorithm files (e.g., code, calibration coefficients, controls) into the site library, supplying attribute and change request information as requested.

ClearCase sends *DDTS*, the Change Request Manager, a request to verify change request information and receives the verification.

The library file is checked in, and the event is logged.

AIT members direct *ClearCase* to build executables configured with the SCF version of the SDP Toolkit. *ClearCase* creates build records and stores them for reference in re-creating the builds. Library change events are logged.

Integration testing reveals that software changes are needed. AIT members, using their view, have *ClearCase* check out specific source files. After first verifying access, the files are made available for modification.

The modified file is checked in (described earlier), except the CCR number supplied is only checked against the *DDTS* database if *ClearCase* finds it differs from what was used during check-in of the previous version of the code. The checkin operation assigns new version identifiers to changed files. Library change events are logged.

AIT members again construct builds, this time configured with the DAAC version of the SDP Toolkit.

As the algorithm progresses in this way through stages to production, authorized persons change the life cycle status for algorithm files to reflect their promotion.

Upon approval of algorithm turnover to production, the CM specialist uses XRP-II to establish a new, operational baseline record in the DAAC's Baseline Manager database and requests that a

new algorithm resource profile be established using data in the software library. XRP-II then requests attributes from *ClearCase*.

The CM specialist interacts with *ClearCase* to create a release partition in the library for the frozen algorithm files, copy them there, lock them and label them with an algorithm identifier. A view is created with which the files can be accessed.

The CM specialist directs *DDTS* to update the resource change requests to reflect completed actions.

5.4.2 Record New Site Baseline

This scenario traces actions associated with recording baseline changes that introduce new configured devices at the sites that implement expanded system functionality. The scenario is depicted in Figure 5.4-2.

5.4.2.1 Beginning Assumptions

None.

5.4.2.2 Interfaces with Other Subsystems and Segments

None.

5.4.2.3 Stimulus

ECS sites' hardware and software have to undergo change due to hardware upgrade and software revision.

5.4.2.4 Participating Classes From the Object Model

Baseline Manager

HardwareControlItem

SoftwareControlItem

ConfiguredDevice

DocumentProfile

BaselineProfile

BaselineChange

Change Request Manager

BaselineManagementReport

5.4.2.5 Beginning System, Segment and Subsystem State(s)

Two resource change requests (of type CCR) that propose changes to the operational baselines at all sites have been recorded in the *DDTS* database and approved for implementation by appropriate CCBs. One CCR covers replacing a platform of the ECS hardware suite with a more capable model. The other CCR covers a new COTS application for use with the hardware.

Formatted files containing descriptions of the new resources and associated documentation has been received at the SMC.

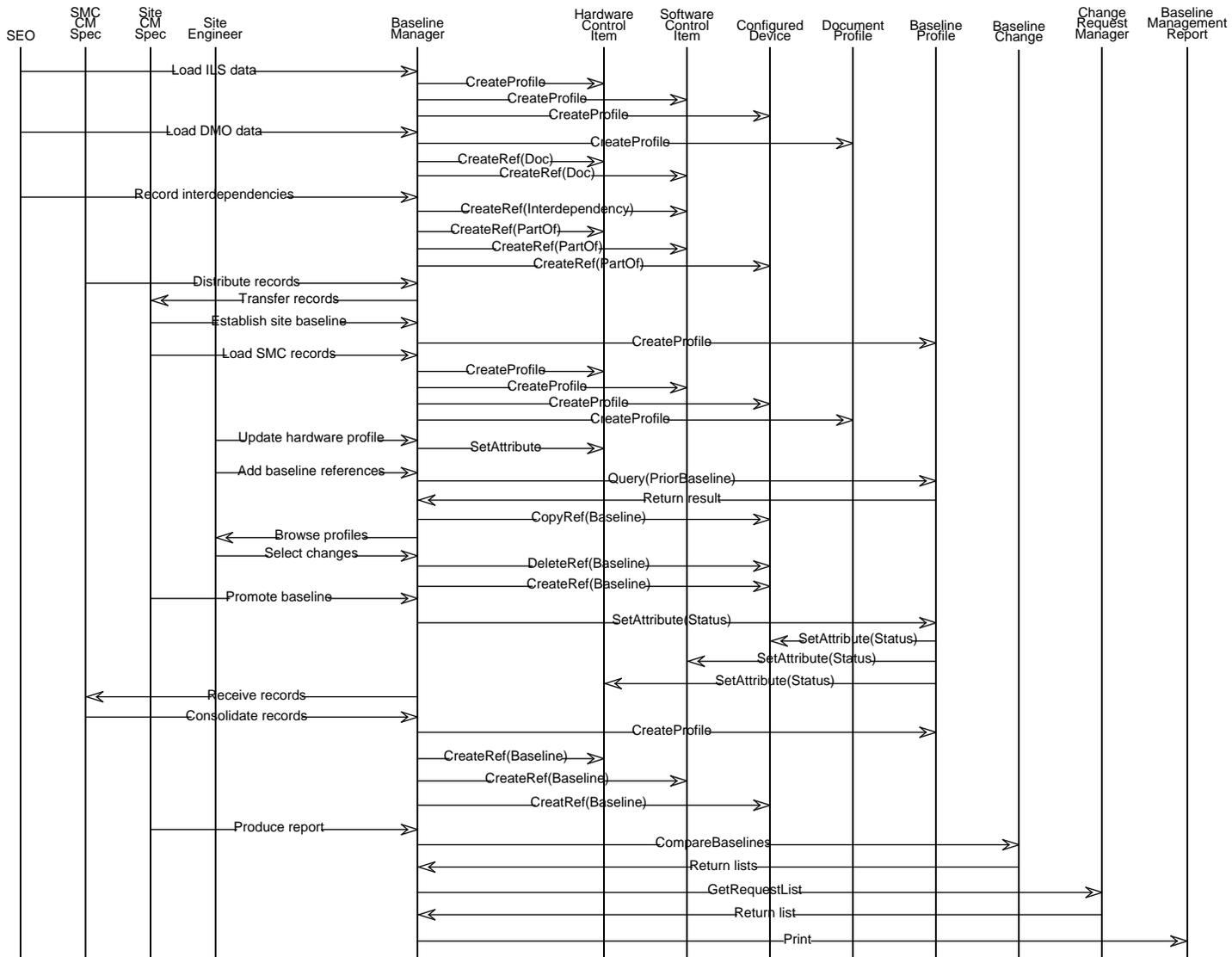


Figure 5.4-2. Baseline Management Scenario: Record New Site

Views exist that provide access to operations baselined versions of software in the library.

5.4.2.6 Ending State

Site baseline data has been updated to reflect the implemented change.

SMC baseline data has been updated to reflect the new baselines at the sites.

Resource change requests have been closed out.

5.4.2.7 Scenario Description

SEO staff member directs XRP-II to load new hardware and software control item and configured device profile data from formatted files provided by the ILS Office.

SEO staff member loads new document profiles from formatted files provided by DMO.

Based on information available in the resource and document profiles, XRP-II automatically establishes references that associate the new resources with appropriate system documents.

SEO staff member interacts with XRP-II to adjust resource interdependencies stored in the database.

Based on the identifiers for the new devices, XRP-II automatically establishes references that specify where they fit in the hierarchical organization of the ECS.

An SMC CM specialist distributes XRP-II data changes to the sites. (Method contingent on tool selection.)

Site CM specialist interacts with XRP-II to establish a profile record for a new site baseline.

Site CM specialist interacts with XRP-II to merge profile data from the SMC.

Once sites' engineers receive the new resources, they use XRP-II to add any previously unknown information (perhaps serial numbers) to the resource profiles.

Each site's engineer establishes references between the new site baseline profile and the resources it includes. XRP-II automatically selects prior baseline's resources and lets the engineer browse resource profiles when selecting changes.

When appropriate, the site CM specialist changes the baseline profile's status to reflect production. XRP-II automatically updates the status of corresponding resources the baseline includes and freezes the baseline-related records.

Sites' CM specialists send their new baseline records to the SMC.

SMC CM specialist generates consolidated reports suitable for posting to the ECS Bulletin Board. XRP-II calls on DDTS to provide the relevant CCR list.

5.4.3 Evaluate System Enhancement Request

This scenario traces the actions associated with registering, assessing, and recording status of proposed changes to ECS resources. The scenario is depicted in Figure 5.4-3.

5.4.3.1 Beginning Assumptions

None.

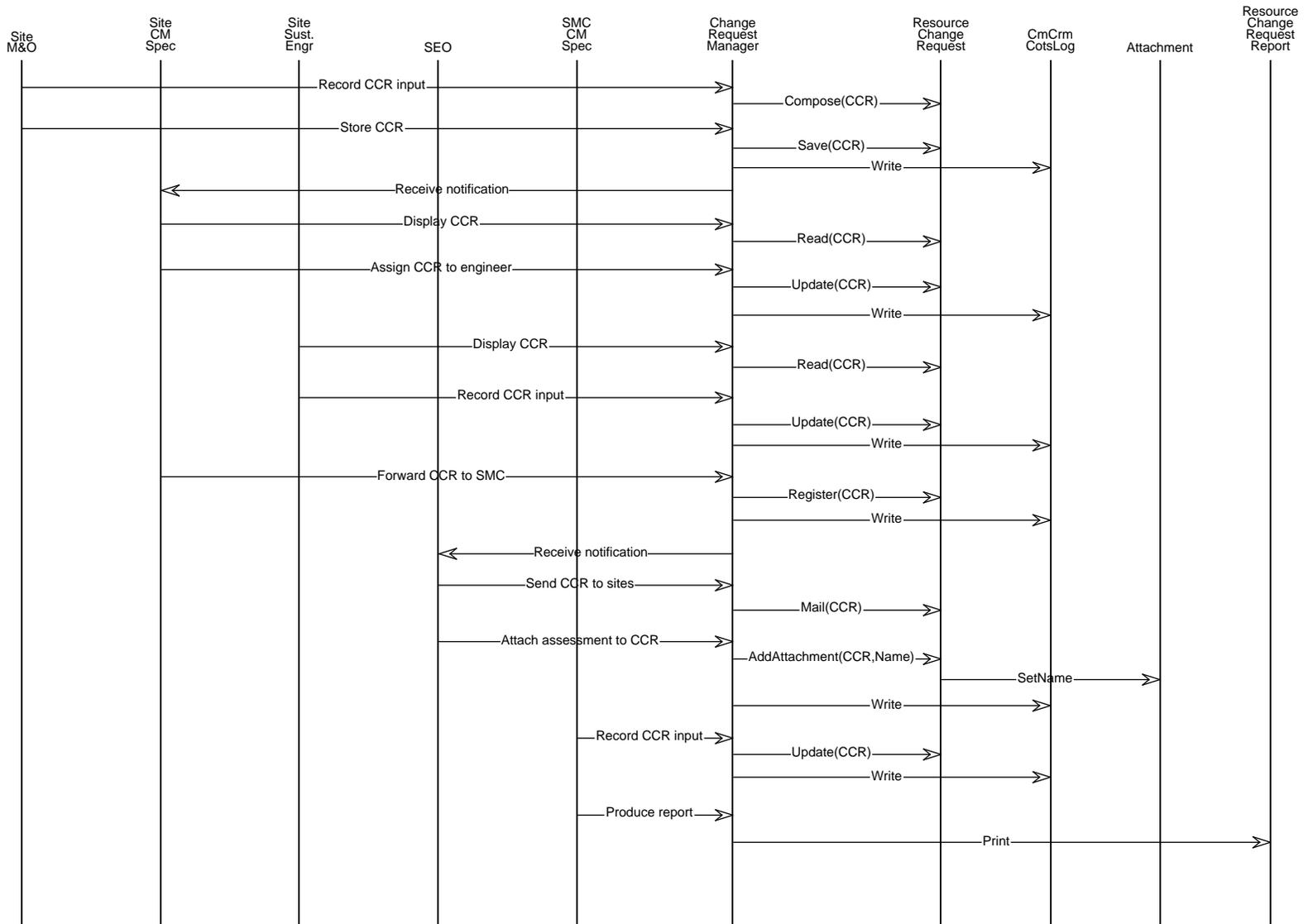


Figure 5.4-3. Change Request Management Scenario: Evaluate System Enhancement

5.4.3.2 Interfaces with Other Subsystems and Segments

CSS

5.4.3.3 Stimulus

A member of the user community recommends to a site that the ECS might be modified in some specific way.

5.4.3.4 Participating Classes From the Object Model

Change Request Manager

ResourceChangeRequest

CmCrmCotsLog

Attachment

ResourceChangeRequest Report

5.4.3.5 Beginning System, Segment and Subsystem State(s)

None.

5.4.3.6 Ending State

A change request record exists in change request database and contains the request's description, its impact summary, and its disposition by the CCB.

The change request record references associated assessments and assessment requests.

The change request and its current status are made available for posting to the ECS bulletin board.

5.4.3.7 Scenario Description

An M&O staff member interacts with *DDTS* to compose an electronic resource change request (in this case, a configuration change request (CCR)). The staff member directs *DDTS* to save the request. *DDTS* saves the CCR and notifies the site CM administrator of the presence of the CCR.

Site CM Administrator reviews the CCR for completeness, and assigns it to the site sustaining engineer for evaluation.

Sust. Engr. evaluates CCR, determines that the CCR has system-wide impact, and enters his/her analysis information into the CCR record. Sustaining Engineer recommends site CCB approves sending CCR to the sustaining engineer organization (SEO) at the SMC for assessment and to ESDIS CCB for approval action.

At site CCB's direction, the site CM specialist directs *DDTS* to send the CCR to the SEO at the SMC. *DDTS* mails the request to the SMC CM specialist and the SEO is notified.

An SEO staff member directs *DDTS* to mail the CCR to all site sustaining engineers for site assessment. As impact assessments are received, the SEO staff member directs *DDTS* to attach them to the change request.

SEO reviews assessments and sends copy of CCR and its recommendation to ESDIS for approval.

At CCB direction, the SMC's CM specialist uses *DDTS* to record CCR disposition, identity of the organization assigned responsibility for implementation, and to update the change request's status.

The SMC's CM specialist directs *DDTS* to produce a report that lists all of the active CCRs and make it available for posting to the ECS Bulletin Board for system-wide viewing.

5.5 Configuration Management Structure

5.5.1 Baseline Manager CSC

This CSC provides all the functionality necessary to record and track what resources constitute ECS operational baselines system-wide. The CMAS object classes it includes are shown in Table 5.5-1.

The Baseline Manager CSC is a transaction-oriented, data-intensive application that will likely perform many more queries than updates. Data store operations are relatively infrequent and accomplished largely interactively by a CM specialist using a keyboard alone for direct data entry or together with files containing formatted data from other applications such as *ClearCase* or a Baseline Manager at a different site.

Table 5.5-1. Baseline Manager Components

CMAS Object Class	Implementation
BaselineManager	COTS (XRP-II)
BaselineProfile	System generated
BaselineChange	System generated
ResourceProfile	System generated
DocumentProfile	System generated
Subsystem	System generated
ConfiguredDevice	System generated
HardwareControlItem	System generated
CI	System generated
SoftwareControlItem	System generated
Algorithm	System generated
Toolkit	System generated
CmBmCotsLog	System generated
BaselineManagementReport	System generated
MsCmBmProxyAgent	Custom

XRP-II, a COTS application, will perform the Baseline Manager functions. The CSC consists of COTS clients and servers for which the following CSUs will be developed:

- Profile Definitions - customization of COTS-provided specifications for baseline manager's records, for baseline profiles, document profiles, and resource profiles
- Profile Input Forms - customization of COTS-provided screens and triggers to accommodate customized profile records
- Profile Import Script - custom script to facilitate adding formatted file data into the baseline manager database

- Profile Export Script - custom script to facilitate producing formatted file data from the database
- Profile Update Triggers - custom coding in native COTS language that implements procedural rules and efficiencies such as automatic carry forward of resource profile links when a new baseline profile is created; specific triggers with identified when a COTS product is selected
- XRP-II Pre-defined Reports - custom coding in native COTS language or customization of COTS-provided specifications for the following menu-selectable, pre-defined reports:
 - Configured Articles List
 - Baselined Documents Report
 - As-Built Resource List
 - Baseline Manager Profiles Report
 - Baseline Change Report

The Release A Overview Design Specification (305-CD-004-001) describes these reports.

- System Management Triggers - custom coding in native COTS language to invoke MACI-provided objects that report application events and handle system management callbacks.
- XRP-II Proxy Agent - customization of Management Agent CSCI's proxy agent. C++ code for interacting with application management agents.

5.5.2 Software Change Manager CSC

This CSC provides all the functionality necessary to version control files containing custom software, data, and documentation. The CMAS object classes it includes are shown in Table 5.5-2.

Table 5.5-2. Software Change Manager Components

CMAS Object Class	Implementation
SoftwareChangeManager	COTS (<i>ClearCase</i>)
SoftwareLibrary	System generated
LibraryFile	System generated
Association	System generated
View	System generated
AccessProfile	System generated
BuildRecord	System generated
CmScmCotsLog	System generated
SoftwareChangeReport	System generated
MsCmScmProxyAgent	Custom

As indicated previously, *ClearCase*, a COTS application, will perform the Software Change Manager function. This client/server application provides extensive library management facilities, but requires development of the following CSUs:

- Library Installers - customization of COTS-provided scripts and specifications to initially configure a software library, including establishing physical file storage locations, creating storage directories, and defining standard attributes that will be maintained for files stored in the library.
- Library View Files - customization of COTS-provided specifications for initial and default views of library objects. This include defining commands to set up a view(s) for group use, naming the view(s), specifying the location for the view storage directory, creating view storage directory, and setting up view for a default set of files.
- File Checkin Trigger - custom coding in native COTS language to enforce file change policies. This trigger monitors the “check-in of files” oriented operations. The trigger executes a script which ask for a CCR/NCR/DR number that's associated with the file being checked in, calls DDTS to verify the number (if not previously verified), evaluates the code returned from DDTS, allows check-in of the file if the return code indicates number is valid, rejects check-in of the file if number is invalid and provides reason for the rejection.
- File Checkout Trigger - custom coding in native COTS language to enforce file change policies. This trigger monitors the “check-out of files” oriented operations. The trigger executes a script which ask for a CCR/NCR/DR number that's associated with the file being checked out, calls DDTS to verify the number and its association with the requested file, evaluates the code returned from DDTS, allows check-out of the file if the return code indicates number is valid, rejects check-out of the file if number is invalid and provides reason for the rejection.
- XRP-II Interface Script - custom coding in native COTS language to service calls from the Baseline Manager. This script uses the parameters passed from the Baseline Manager to locate specified software file(s) in the software library, gets the file(s) attributes values and place values in a formatted file.
- *ClearCase* Pre-defined Reports - custom coding in native COTS language or customization of COTS-provided specifications for the following menu-selectable, pre-defined reports:
 - Software Library Objects Report
 - Software Library Builds Report
 - Software Library Version Tree List
 - Software Library Registered View List
 - Software Library View Specification List
 - Software Library Checkouts List
 - Software Library Event History List

The Release A Overview Design Specification (305-CD-004-001) describes these reports.

- System Management Triggers/Scripts - custom coding in native COTS language to invoke MACI-provided objects that report application events and handle system management callbacks.
- *ClearCase* Proxy Agent - customization of Management Agent CSCI's proxy agent. C++ code for interacting with application management agents.

5.5.3 Change Request Manager CSC

This CSC provides all the functionality necessary to compose, submit, coordinate, and track status of proposals to change ECS resources. The CMAS object classes it includes are shown in Table 5.5-3.

Table 5.5-3. Change Request Manager Components

CMAS Object Class	Implementation
ChangeRequestManager	COTS (Distributed Defect Tracking System -- DDTS)
ResourceChangeRequest	System generated
Attachment	System generated
CmCrmCotsLog	System generated
ResourceChangeRequestReport	System generated
MsCmCrmProxyAgent	Custom

Like XRP-II, *DDTS*, the Change Request Manager, is an interactive, transaction-oriented application. This client/server application requires development of the following CSUs:

- Resource Change Request Definitions - customization of COTS-provided specifications for resource change request records.
- Resource Change Request Input Forms - customization of COTS-provided screens and triggers to accommodate customized resource change requests.
- GetVerification Script - custom coding in native COTS language to service calls for evaluations of specified conditional expressions on resource change request records.
- GetRequestList Script - custom coding in native COTS language to service calls for values for specified attributes of specified resource change requests.
- Mail-related Triggers - custom coding in native COTS language that implement notification policies.
- *DDTS* Pre-defined Reports - Custom coding in native COTS language or customization of COTS-provided specifications for the following menu-selectable, pre-defined reports:
 - Change Request Report
 - Change Request List
 - Change Request Metrics Report

The Release A Overview Design Specification (305-CD-004-001) describes these reports.

- System Management Triggers - Custom coding in native COTS language to invoke MACI-provided objects that report application events and handle system management callbacks.
- *DDTS* Proxy Agent - customization of Management Agent CSCI's proxy agent. C++ code for interacting with application management agents.

5.6 MLCI Management and Operation

5.6.1 System Management Strategy

MLCI's configuration management CSCs require administration common to transaction-oriented applications that maintain repositories of data. Administrative operations include startup, shutdown, backup, performance tuning, and data maintenance. Scripts that facilitate performing these operations are available to local operators or system administrators via the CSCs, UNIX shell, and -- in the case of startup and shutdown - via the Fault Manager.

MLCI's servers operate continuously under normal circumstances. Each MLCI CSC uses an event log, maintained internally by the COTS application, that identifies and timestamps errors and changes to management data when they occur. These logs can be read, printed, and archived in a manner that can be tailored to local management policies. Selected events are also automatically reported to other, local MSS applications via MSS' management agents.

In addition to the COTS applications, MLCI contains metadata, configuration parameters, and mechanisms that implement system management policies. Installation will provide an initial configuration for distribution of databases, database structures, and default specifications for system resources, access controls, reports and data views. Generally, these can be customized to enhance performance and to accommodate growth. Data stores will require periodic maintenance to remove obsolete or unwanted records, references, and work files and to archive historical data. As usage expands, specifications for data viewing and control will need adjusting and storage may need reconfiguring. The latter might include establishing new data partitions, adjusting and moving data stores across devices, and resizing buffer caches where possible in response to performance indicators.

5.6.2 Operator Interfaces

All MLCI CSCs will offer operators both graphical and command line interfaces as provided by their respective COTS vendors and described in their user manuals. In some cases, custom menus and scripts will be added to facilitate accomplishing repetitive or complex actions. _

In addition, operators can invoke select management options (such as startup and shutdown) via a common management user interface. This is provided by the Fault Management Application Service and is described in Section 4.1.

5.6.3 Reports

MCI provides a variety of standard MSS fault, configuration, accounting, performance, and security reports needed to support application management ECS-wide. These are described in the Release A Overview Design Specification (305-CD-004-001). In addition to these, each MLCI COTS application will produce:

- a) event history reports for diagnosing problems and auditing database changes;
- b) access profile reports for controlling use of the application.

5.7 Software Distribution Management

5.7.1 Software Distribution Management Overview

Software Distribution Manager provides the capability for the M&O Staff to distribute ECS software, database information, and software documentation for ECS products and services as well as commercial software. These distributed products are tracked to insure delivery and maintain an up-to-date baseline. Tivoli Courier, a COTS product, will satisfy the functions for the Software Distribution Manager.

The service will employ a software distribution process to package, track, and distribute software, databases, and documentation for delivery to destinations at both ECS and ECS-connected sites. The process entails maintaining version-controlled repositories for toolkit software, software upgrades, and documentation. The CSS Bulletin Board Service will be utilized to provide access to the toolkit repository and information. Both automatic and operator-assisted/controlled distribution services will be provided.

The Tivoli/Courier management application helps the M&O Staff distribute and manage software across a multi-platform network. It provides an efficient method for distributing, installing, and controlling software across the network. Typically, users working on a networked system maintain a large stock of software applications and tools. Tivoli/Courier enables a centralized software management capability to add new applications, update existing software with newer versions, and synchronize software on distributed systems.

5.7.2 Software Distribution Management Context

Figure 5.7-1 contains the context diagram for Software Distribution Management.

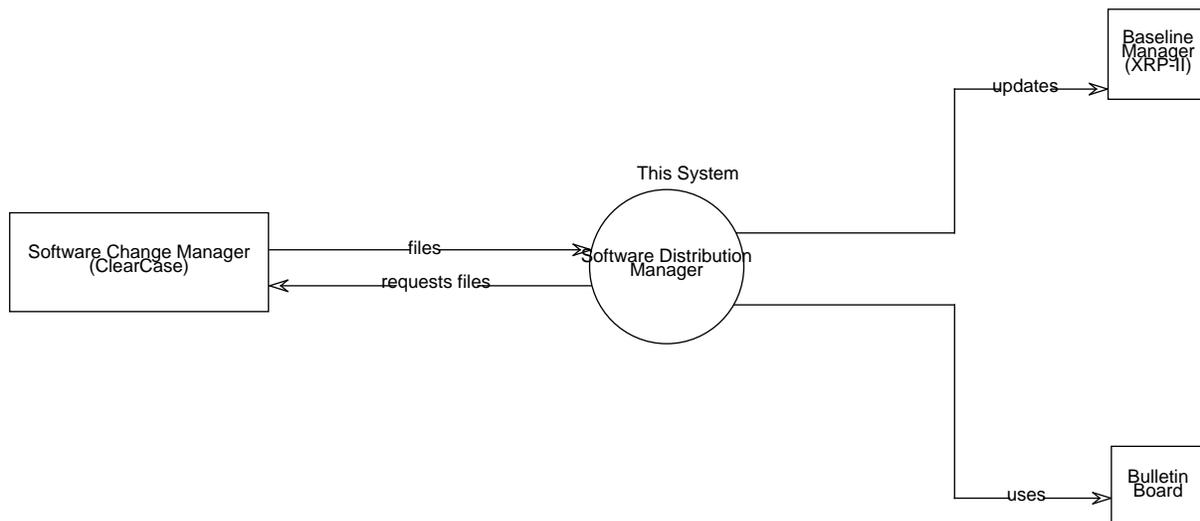


Figure 5.7-1. Software Distribution Management Context Diagram

5.7.3 Software Distribution Management Object Model

Figure 5.7-2, the Object Model for Software Distribution Management depicts the major classes and their associations with one another. These are described below:

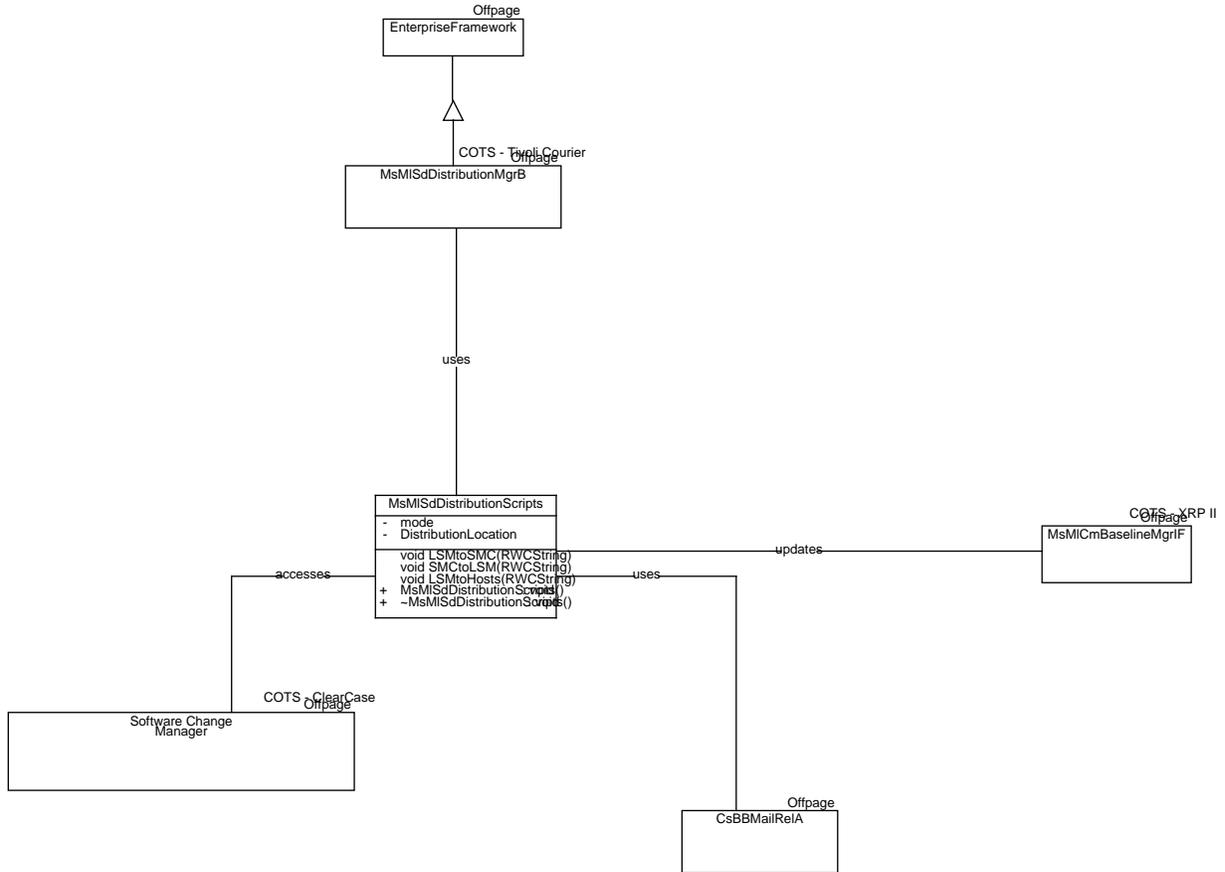


Figure 5.7-2. Software Distribution Management Object Model

5.7.3.1 CsBBMailRelA Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

The MsBulletinBoardB class represents the posting to the bulletin board of the policies and procedures.

Attributes:

None

Operations:

None

Associations:

The CsBBMailRelA class has associations with the following classes:

Class: MsMISdDistributionScripts uses

5.7.3.2 EnterpriseFramework Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

EnterpriseFramework is the Tivoli COTS product that performs enterprise wide services: System Administration (Tivoli/Admin), Software distribution (Tivoli/Courier), performance monitoring (Tivoli/Sentry) and fault correlation (Tivoli/Enterprise Console). The framework also acts as the integrated desktop for Maintenance and Operations, integrating other administrative functions such as Sybase database administration, system backup/restore, and DCE Cell administration.

Attributes:

None

Operations:

None

Associations:

The EnterpriseFramework class has associations with the following classes:

None

5.7.3.3 MsMICmBaselineMgrIF Class

Parent Class:Not Applicable

Attributes:

None

Operations:

None

Associations:

The MsMICmBaselineMgrIF class has associations with the following classes:

Class: MsMISdDistributionScripts updates

5.7.3.4 MsMISdDistributionMgrB Class

Parent Class:EnterpriseFramework

Public:No

Distributed Object:No

Purpose and Description:

This Software Distribution Manager class controls the distribution of software packages project-wide.

Attributes:

All Attributes inherited from parent class

Operations:

All Operations inherited from parent class

Associations:

The MsMISdDistributionMgrB class has associations with the following classes:

Class: MsMISdDistributionScripts uses

5.7.3.5 MsMISdDistributionScripts Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This Site List class holds a list of both ECS sites and non-ECS sites that may receive software packages.

Attributes:

DistributionLocation - This attribute represents the locations where the software is to be distributed.

Data Type:
Privilege:Private
Default Value:

mode - Represents the mode such that software can be distributed automatically to a particular test directory structure to be executed in a designated mode.

Data Type:
Privilege:Private
Default Value:

Operations:

MsMISdDistributionScripts - This is the constructor of the class.

Arguments:
Return Type:void
Privilege:Public

void LSMtoHosts - This operation represents a script that will distribute software from the LSM to the local Hosts.

Arguments:RWCString

void LSMtoSMC - This operation represents a script that will distribute software from the LSM to the SMC.

Arguments:RWCString

void SMCtoLSM - This operation represents a script that will distribute software from the SMC to the LSM.

Arguments:RWCString

~MsMISdDistributionScripts - This is the destructor of the class.

Arguments:
Return Type:void
Privilege:Public

Associations:

The MsMISdDistributionScripts class has associations with the following classes:

Class: SoftwareChangeManager accesses
Class: MsMICmBaselineMgrIF updates
Class: CsBBMailRelA uses
Class: MsMISdDistributionMgrB uses

5.7.3.6 SoftwareChangeManager Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class is a server manager that controls a software library and manages the execution of operator initiated events associated with it.

Attributes:

None

Operations:

None

Associations:

The SoftwareChangeManager class has associations with the following classes:

Class: MsMISdDistributionScripts accesses

5.7.4 Software Distribution Management Dynamic Model

This scenario traces the actions associated with the transferring of a software package by a Configuration Manager for distribution to an ECS site. The scenario is depicted in Figure 5.7-3.

5.7.4.1 Beginning Assumptions

The Configuration Manager has already been authorized to access the distribution system.

5.7.4.2 Interfaces with Other Subsystems and Segments

Software Change Manager (ClearCase)

5.7.4.3 Stimulus

The Configuration Manager has an assembled software package that needs to be distributed

5.7.4.4 Participating Classes From the Object Model

MsMISdDistributionMgrB (Tivoli Courier)

Software Change Manager (ClearCase)

5.7.4.5 Beginning System, Segment and Subsystem State(s)

Normal Operations

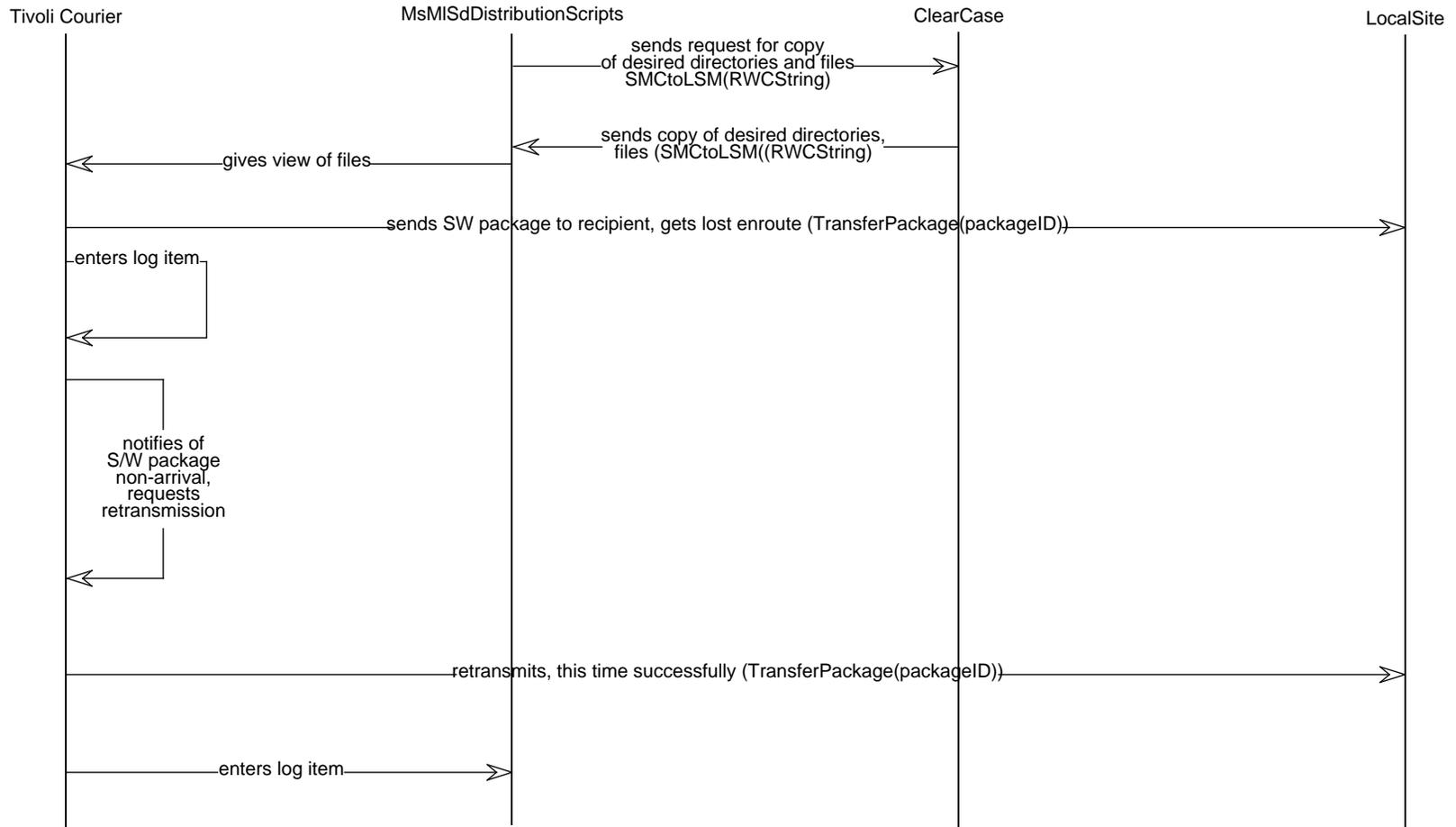


Figure 5.7-3. Software Distribution Scenario: Unsuccessful Software Package Transfer

5.7.4.6 Ending State

The Configuration Manager has successfully transmitted the software package to another Site.

5.7.4.7 Scenario Description

- The M&O Configuration Manager sends a message to an M&O Staffer notifying of the plan to transfer the software package (via E-mail or phone).
- The receiver sends a return message to the M&O Configuration Manager acknowledging receipt of the message (optional).
- The M&O Configuration Manager executes MsMISdDistributionMgrB(Tivoli Courier).
- MsMISdDistributionMgrB(Tivoli Courier) requests for a distribution of MsMISdDistributionScripts SMCToLSM(RWCString).
- MsMISdDistributionScripts (SMCToLSM(RWCString)) executes script to retrieve files from ClearCase.
- MsMISdDistributionScripts (SMCToLSM(RWCString)) retrieves the files from ClearCase.
- MsMISdDistributionScripts allows MsMISdDistributionMgrB(Tivoli Courier) a view of the files.
- MsMISdDistributionMgrB(Tivoli Courier) sends software package to recipient. However, the site never receives the package.
- MsMISdDistributionMgrB(Tivoli Courier) logs the event.
- MsMISdDistributionMgrB(Tivoli Courier) prepares for re transmit.
- MsMISdDistributionMgrB(Tivoli Courier) re transmits the software package to recipient and is successful.
- MsMISdDistributionMgrB(Tivoli Courier) logs successful transmission.

5.7.5 Software Distribution Management Structure

This service provides all the functionality necessary to distribute, track, and record software updates, toolkit software, and documentation distributed to authorized users ECS system-wide. The object classes it includes are shown in Table 5.7-1.

Table 5.7-1. Software Distribution Manager Components

Object Class	Implementation
MsMISdDistributionMgrB	COTS/Custom (Tivoli Courier)
MsMISdDistributionScripts	Custom

5.8 Software License Management

5.8.1 Software License Management Overview

The Software License Management service provides the capability for the M&O Staff to monitor and administer the licensing aspects of COTS software.

This service maintains current licensing information on all COTS products managed by ECS. This information includes product identification, server location, licensing provisions such as expiration date, and number of users. The service supports the creation, installation, reinstallation, update, and distribution of licenses and associated provisions for all ECS servers.

The Software License Management service monitors and collects utilization statistics on licensed software; detects and notifies M&O staff of significant licensing events including grants, releases and imminent expirations; and reports on collected statistical information to assist M&O Staff in analyzing trends in licensed product usage.

The COTS software chosen for the Software License management, FLEXlm and iFORLS, provides reliability in the event of a license server node failure, provides optional enforcement of expiration dates for demonstration copies and maintenance releases, supports floating and node-locked licenses, and provides transparent reconnection of applications when their daemon process becomes unavailable, including conditions of license server node failure.

5.8.2 Software License Management Context

Figure 5.8-1 contains the context diagram for the Software License Management.

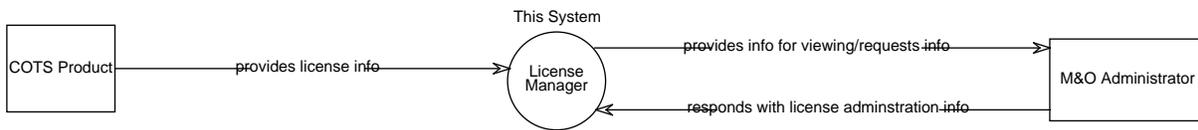


Figure 5.8-1. Software License Management Context Diagram

5.8.3 Software License Management Object Model

Figure 5.8-2, the Object Model for Software License Management Service, depicts the major classes and their associations with one another. These are described below:

5.8.3.1 EcAgCOTSManger Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

this abstract class embodies the characteristics and functionality of a manager object responsible for managing a single COTS process. It encapsulates all MSS management application functions into a single class. The COTS proxy agent developer is responsible for inheriting from this class and specializing it towards the COTS process to manage.

Attributes:

None

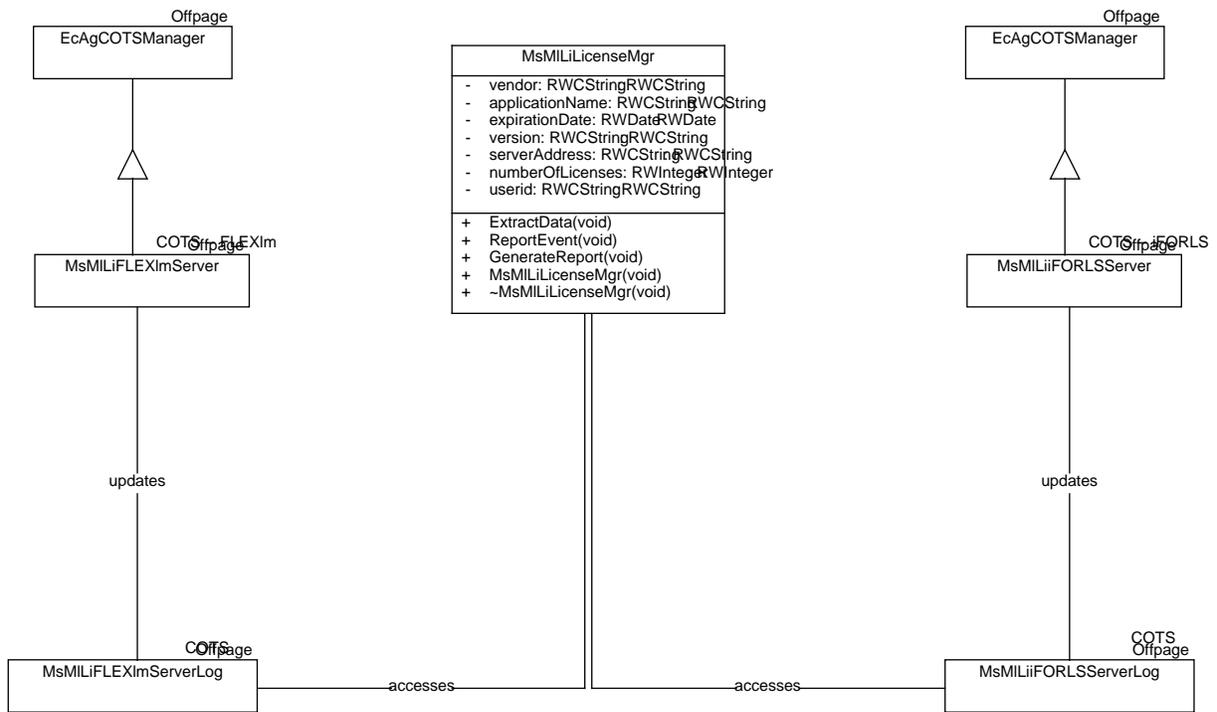


Figure 5.8-2. Software License Management Object Model

Operations:

None

Associations:

The EcAgCOTSMManager class has associations with the following classes:

None

5.8.3.2 MsMILiFLEXImServer Class

Parent Class:EcAgCOTSMManager

Public:No

Distributed Object:No

Purpose and Description:

The MsMILiLicenceB class contains the attributes and operations to control the number of licenses available as well as the control of the distribution of the licenses.

Attributes:

All Attributes inherited from parent class

Operations:

All Operations inherited from parent class

Associations:

The MsMILiFLEXImServer class has associations with the following classes:

Class: MsMILiFLEXImServerLog updates

5.8.3.3 MsMILiFLEXImServerLog Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

The MsMILiLicenceB class contains the attributes and operations to control the number of licenses available as well as the control of the distribution of the licenses.

Attributes:

None

Operations:

None

Associations:

The MsMILiFLEXImServerLog class has associations with the following classes:

Class: MsMILiLicenseMgr accesses

Class: MsMILiFLEXImServer updates

5.8.3.4 MsMILiLicenseMgr Class

Parent Class:Not Applicable

Attributes:

applicationName

Data Type:RWCString

Privilege:Private

Default Value:

expirationDate

Data Type:RWDate

Privilege:Private

Default Value:

numberOfLicenses

Data Type:RWInteger

Privilege:Private

Default Value:

serverAddress

Data Type:RWCString

Privilege:Private

Default Value:

userid

Data Type:RWCString

Privilege:Private

Default Value:

vendor

Data Type:RWCString

Privilege:Private

Default Value:

version

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

ExtractData

Arguments:void

Return Type:Void

Privilege:Public

GenerateReport

Arguments:void

Return Type:Void

Privilege:Public

MsMILiLicenseMgr

Arguments:void

Return Type:Void

Privilege:Public

ReportEvent

Arguments:void

Return Type:Void

Privilege:Public

~MsMILiLicenseMgr

Arguments:void

Return Type:Void

Privilege:Public

Associations:

The MsMILiLicenseMgr class has associations with the following classes:

Class: MsMILiFLEXImServerLog accesses

Class: MsMILiiFORLSServerLog accesses

5.8.3.5 MsMILiiFORLSServer Class

Parent Class:EcAgCOTSManager

Public:No

Distributed Object:No

Purpose and Description:

The MsMILiLogB class contains the attributes and operations for maintaining a log for the Software License Management Service.

Attributes:

All Attributes inherited from parent class

Operations:

All Operations inherited from parent class

Associations:

The MsMILiiFORLSServer class has associations with the following classes:

Class: MsMILiiFORLSServerLog updates

5.8.3.6 MsMILiFORLSServerLog Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

The MsMILiReportGeneratorB class contains the attributes and operations necessary to perform the required reports for the Software License Management Service.

Attributes:

None

Operations:

None

Associations:

The MsMILiFORLSServerLog class has associations with the following classes:

Class: MsMILiLicenseMgr accesses

Class: MsMILiFORLSServer updates

5.8.4 Software License Management Dynamic Model

This section discusses typical operational usage of the Software License Management service by license administration personnel on the M&O staff.

5.8.4.1 Retrieve Denied License Requests

This scenario traces events associated with the M&O Staff retrieving a report for the number of COTS products that have been denied licenses in the current day. See Figure 5.8-3.

5.8.4.2 Beginning Assumptions

The M&O Staff has an indication that licenses are being denied and they are at their threshold for number of users for a particular COTS products.

5.8.4.2.1 Interfaces with Other Subsystems and Segments

None

5.8.4.2.2 Stimulus

A M&O staff notices that there has been a denial on several COTS products.

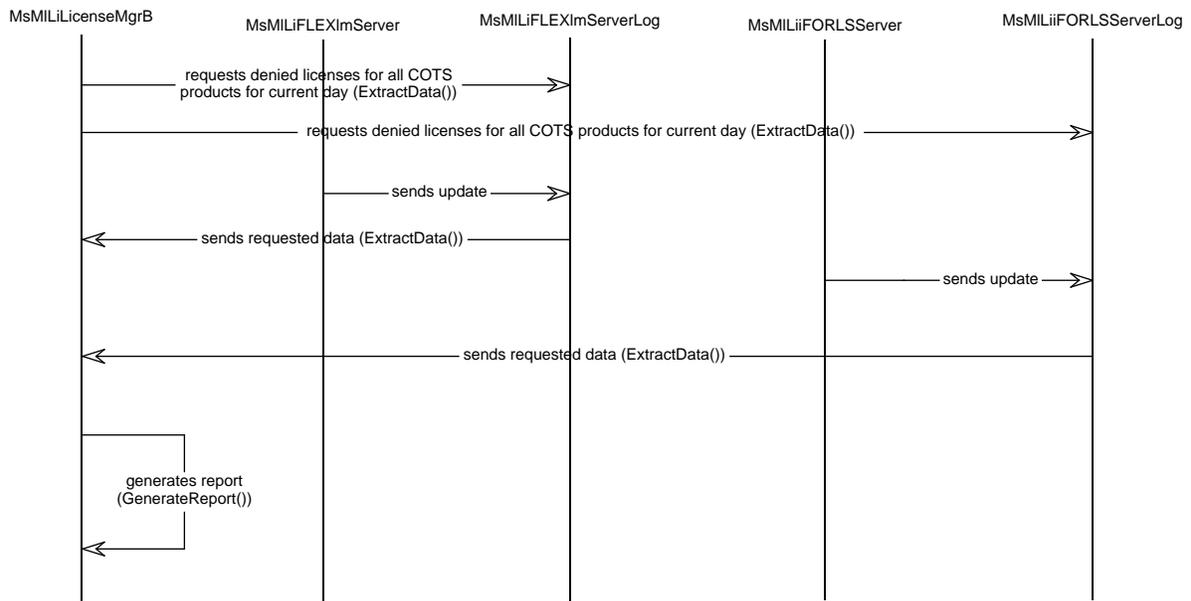


Figure 5.8-3. Software License Scenario: Request Denied License

5.8.4.2.3 Participating Classes From the Object Model

MsMILiLicenseMgrB

MsMILiFLEXImServer

MsMILiFLEXImServerLog

MsMILiFORLSServer

MsMILiFORLSServerLog

5.8.4.2.4 Beginning System, Segment and Subsystem State(s)

The system is in a normal operational state.

5.8.4.2.5 Ending State

The requested report is printed out.

5.8.4.3 Scenario Description

- MsMILiLicenseMgrB requests from both MsMILiFLEXImServerLog and MsMILiFORLSServerLog for denied licenses for current day (ExtractData()).
- MsMILiFLEXImServerLog receives update from MsMILiFLEXImServer.
- MsMILiFLEXImServerLog sends requested information to MsMILiLicenseMgrB (ExtractData()).
- MsMILiFORLSServerLog receives update from MsMILiFORLSServer.

- MsMILiiFORLSServerLog sends requested information to MsMILiLicenseMgrB (ExtractData()).
- MsMILiLicenseMgrB generates a report (GenerateReport()).

5.8.5 Software License Management Structure

This service provides the functionality necessary to monitor and administer COTS licenses. The object classes it includes are shown in Table 5.8-1.

Table 5.8-1. Software License Manager Components

Object Class	Implementation
MsMILiLicenseManagerB	Custom
MsMILiFLEXServer	COTS (FLEXIm)
MsMILiFLEXServerLog	COTS (FLEXIm)
MsMILiiFORLSServer	COTS (iFORLS)
MsMILiiFORLSServerLog	COTS (iFORLS)

5.9 Training Management

5.9.1 Training Management Overview

The Training Management Service supports training programs for ECS employees, as well as certification programs for ECS M&O personnel. The purpose of this Service is to ensure that each employee receives the proper amount and type of training in accordance with their job description. The Service will be managed by members of the M&O staff by utilizing a Sybase database.

The Training Management Service receives training requirements, specific to each job description, and resource requirements from M&O personnel. Based on these requirements and the availability of the resources, which is determined by making a request to the Resource Planner, the Training Service supports the development and maintenance of training materials (through OA tools) and schedules. The database containing the training information will be local to the SMC, as well as to each DAAC, thus allowing them to retrieve the information they require. This database will be automatically replicated by Sybase at a user-specified interval.

Training courses are grouped into curricula for the M&O personnel certification process. In order for an M&O employee to become certified in a particular subject area, the curriculum in that subject area must be completed and a certain level of proficiency must be met. The Training Service maintains and updates each M&O employee's certification, as they must be recertified every so often for reasons such as a change in technology, the certification has expired, or possibly a lack of adequate performance. The Service also supports course evaluations, which are available for review and processing.

5.9.2 Training Management Context

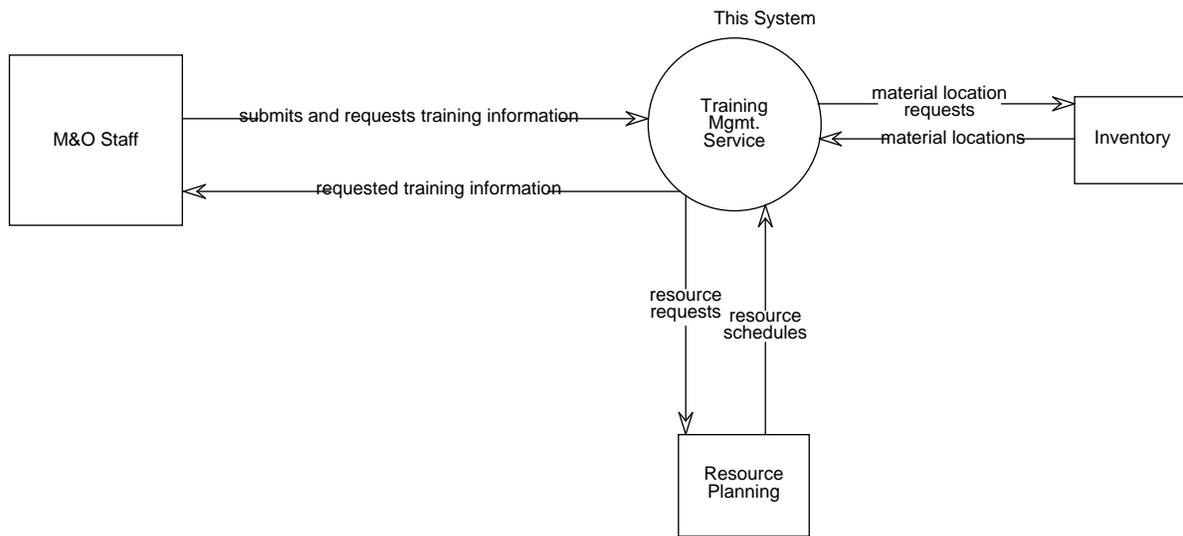


Figure 5.9-1. Training Management Context Diagram

5.9.3 Training Management Object Model

5.9.3.1 EcPfManagedServer Class

Parent Class:Not Applicable

Public:Yes

Distributed Object:No

Purpose and Description:

This is the container class that starts up the event Manager, table Manager, monitor, port monitor, discoverer, subagent configuration, static buffer, and the deputy gate. This class also starts a thread that triggers scheduled events (i.e. polling ECS application's performance metrics).

Attributes:

None

Operations:

None

Associations:

The EcPfManagedServer class has associations with the following classes:

None

5.9.3.2 MsAcUsrProfile Class

Parent Class:Not Applicable

Public:Yes

Distributed Object:No

Purpose and Description:

Attributes:

None

Operations:

None

Associations:

The MsAcUsrProfile class has associations with the following classes:

None

5.9.3.3 MsMITrCertSkillsCatalogB Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class contains the catalogs that further explain the certification procedure.

Attributes:

catalogID - This attribute represents a unique value by which each catalog will be identified.

Data Type:RWCString

Privilege:Private

Default Value:

function - This attribute represents the function for which the Skills Catalog was written.

Data Type:RWCString

Privilege:Private

Default Value:

skillsRequired - This attribute represents the skills each M&O trainee is required to have in order to be certified in a certain subject area.

Data Type:RWCString

Privilege:Private

Default Value:

taskDescription - This attribute represents a brief description of the tasks the M&O trainees need to perform according to their subject area.

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

DisplayCatalog - This operation displays the catalog information to the screen.

Arguments:catalogID

Return Type:EcTVoid

Privilege:Public

MsMITrCertSkillsCatalogB - This operation is the constructor for this class.

Arguments:

Return Type:Void

Privilege:Public

PrintCatalog - This operation prints the particular catalog.

Arguments:catalogID

Return Type:EcTVoid

Privilege:Public

UpdateCatalog - This operation updates the catalog information.

Arguments:catalogID

Return Type:EcTVoid

Privilege:Public

~MsMITrCertSkillsCatalogB - This operation is the destructor for this class.

Arguments:

Return Type:Void

Privilege:Public

Associations:

The MsMITrCertSkillsCatalogB class has associations with the following classes:

Class: MsMITrCertificationB isexplainedin

5.9.3.4 MsMITrCertificationB Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class maintains a record of certification for each M&O Employee.

Attributes:

certificationDate - This attribute represents the date on which each trainee became certified; this attribute will be used as a reference as to when each trainee needs to be recertified.

Data Type:RWDate

Privilege:Private

Default Value:

certificationMethod - This attribute represents how the trainee was certified; e.g., test, paper, small project, and the like.

Data Type:RWCString

Privilege:Private

Default Value:

expirationDate - This attribute represents the date when the certification expires for each M&O trainee.

Data Type:RWDate

Privilege:Private

Default Value:

recertificationReason - This attribute represents a very brief explanation as to why the trainee was recertified.

Data Type:RWCString

Privilege:Private

Default Value:

traineeID - This attribute represents which trainee is being certified.

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

CheckCurriculum - This operation calls the VerifyCurr operation to check if a trainee has completed the correct curriculum.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL: // if (certify = TRUE and skills have been checked) then

// confirm that the correct curriculum has been completed

// (VerifyCurriculum where skills match subjectArea)

CheckSkills - This operation checks the Catalogs for a trainee's required skills.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL:

DisplayCertInfo - This operation displays the certification information to the screen.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL:

DisplayConfirmation - This operation displays a confirmation to the screen that a task has been completed successfully.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL:

MsMITrCertificationB - This operation is the constructor for this class.

Arguments:

Return Type:Void

Privilege:Public

PDL: No PDL

PrintCertInfo - This operation prints the certification information.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL: No PDL

UpdateCertInfo -

Arguments:traineeID

Return Type:EcTVoid

Privilege:Public

PDL:

~MsMITrCertificationB - This operation is the destructor for this class.

Arguments:

Return Type:Void

Privilege:Public

PDL: No PDL

Associations:

The MsMITrCertificationB class has associations with the following classes:

Class: MsMITrCertSkillsCatalogB is explained in

Class: MsMITrCurriculumB is satisfied by

Class: MsMITrTraineeB receives

5.9.3.5 MsMITrCourseB Class

Parent Class: Not Applicable

Public: No

Distributed Object: No

Purpose and Description:

This class maintains a list of courses in which employees may be trained, along with their descriptions.

Attributes:

courseDescription - This attribute represents a brief description of the information covered in each training course.

Data Type: RWCString

Privilege: Private

Default Value:

courseID - This attribute represents a unique value by which each course will be identified.

Data Type: RWCString

Privilege: Private

Default Value:

courseListID - This attribute represents a unique value by which each type of course list will be identified.

Data Type: RWCString

Privilege: Private

Default Value:

courseTitle - This attribute represents the titles of courses which are available for the training of employees.

Data Type: RWCString

Privilege: Private

Default Value:

credits - This attribute represents the number of credits a trainee can earn for each course.

Data Type: EcTInt

Privilege: Private

Default Value:

deliveryMethod - This attribute represents how each course is administered; e.g., on-the-job, classroom, computer-based.

Data Type:RWCString

Privilege:Private

Default Value:

subjectArea - This attribute represents the subject for which the course is being administered.

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

AddCourse - This operation allows for the addition of newly developed courses into the current list.

Arguments:

Return Type:EcTVoid

Privilege:Public

ConfirmVacancy - This operation calls the ComputeVacancy operation in the MsMITrCourseLocationB class to confirm whether the course is available on the selected date.

Arguments:

Return Type:EcTVoid

Privilege:Public

DeleteCourse - This operation allows for the deletion of courses from the current list.

Arguments:courseID

Return Type:EcTVoid

Privilege:Public

DisplayConfirmation - This operation displays whether the date has been confirmed.

Arguments:startDate

Return Type:EcTVoid

Privilege:Public

DisplayCourseInfo - This operation displays the information relevant to the selected course.

Arguments:courseTitle, courseID

Return Type:EcTVoid

Privilege:Public

DisplayCourseList - This operation displays the proper course list corresponding to the

operator's selection.
Arguments:courseListID
Return Type:EcTVoid
Privilege:Public

MsMITrCourseB - This operation is the constructor for this class.
Arguments:
Return Type:Void
Privilege:Public

PrintCourseInfo - This operation prints the information for the particular course.
Arguments:courseID
Return Type:EcTVoid
Privilege:Public

ProduceCost - This operation asks TrainingCost to calculate the training cost of a particular course.
Arguments:
Return Type:EcTVoid
Privilege:Public

RequestCourseList - This operation asks the operator which course list s/he would like to display.
Arguments:
Return Type:EcTVoid
Privilege:Public

RequestVerification - This operation requests verification of either the title or the date.
Arguments:courseTitle, startDate
Return Type:EcTVoid
Privilege:Public

UpdateCourse - This operation updates the information for the particular course.
Arguments:courseID
Return Type:EcTVoid
Privilege:Public

~MsMITrCourseB - This operation is the destructor for this class.
Arguments:
Return Type:Void
Privilege:Public

Associations:
The MsMITrCourseB class has associations with the following classes:
Class: MsMITrTraineeB attends

Class: MsMITrCurriculumB is contained in
Class: MsMITrCourseLocationB is held in
Class: MsMITrCourseB is required by
Class: MsMITrMaterialB uses
MsMITrMgrB (Aggregation)

5.9.3.6 MsMITrCourseLocationB Class

Parent Class: Not Applicable

Public: No

Distributed Object: No

Purpose and Description:

This class maintains additional information about each training course which is dependent on where the course is administered.

Attributes:

completionDate - This attribute represents the date when each training course ends at a particular location.

Data Type: RWDate

Privilege: Private

Default Value:

courseCost

Data Type: EcCurrency

Privilege: Private

Default Value:

currentEnrollment - This attribute represents the number of trainees currently enrolled in a particular training course.

Data Type: EcTInt

Privilege: Private

Default Value:

enrollmentLimit - This attribute represents the maximum number of students allowed in a training course depending on the location.

Data Type: EcTInt

Privilege: Private

Default Value:

instructorName

Data Type: MsMITrInstructorB

Privilege: Private

Default Value:

resourcesRequired - This attribute represents the resources that this course requires to be in use at the time of training.

Data Type:RWCString

Privilege:Private

Default Value:

startDate - This attribute represents the date when each training course begins at a particular location.

Data Type:RWDate

Privilege:Private

Default Value:

vacancy - This attribute represents the number of remaining spaces in a training course in which trainees may be enrolled; this attribute is derived by subtracting the number of trainees currently enrolled in a course (found on the Schedule) from the EnrollmentLimit.

Data Type:EcTInt

Privilege:Private

Default Value:

This is a Derived Attribute.

Operations:

ComputeVacancy - This operation calculates the number of seats left in a particular course, given the course, its enrollment limit, and its current enrollment.

Arguments:currentEnrollment, enrollmentLimit

Return Type:EcTInt

Privilege:Public

MsMITrCourseLocationB - This operation is the constructor for this class.

Arguments:

Return Type:Void

Privilege:Public

~MsMITrCourseLocationB - This operation is the destructor for this class.

Arguments:

Return Type:Void

Privilege:Public

Associations:

The MsMITrCourseLocationB class has associations with the following classes:

Class: MsMITrCourseB isheldin

5.9.3.7 MsMITrCurriculumB Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class maintains a list of training courses, which are grouped together according to subject. M&O Personnel must complete a curriculum of courses in order to become certified.

Attributes:

courseID - This attribute represents the courses included in this particular curriculum.

Data Type:RWCString

Privilege:Private

Default Value:

curriculumDescription - This attribute represents a brief description of which courses are included in each curriculum and what subject area each curriculum satisfies.

Data Type:RWCString

Privilege:Private

Default Value:

curriculumID - This attribute represents a unique value by which each group of courses will be identified.

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

DisplayCurriculum - This operation displays the particular curriculum information to the screen.

Arguments:curriculumID

Return Type:EcTVoid

Privilege:Public

MsMITrCurriculumB - This operation is the constructor for this class.

Arguments:

Return Type:Void

Privilege:Public

PrintCurriculum - This operation prints the selected curriculum.

Arguments:curriculumID

Return Type:EcTVoid
Privilege:Public

UpdateCurriculum - This operation updates the particular curriculum.

Arguments:curriculumID
Return Type:EcTVoid
Privilege:Public

VerifyCurrComplete - This operation checks that the trainee has completed the curriculum corresponding to their job description.

Arguments:
Return Type:EcBoolean
Privilege:Public

~MsMITrCurriculumB - This operation is the destructor for this class.

Arguments:
Return Type:Void
Privilege:Public

Associations:

The MsMITrCurriculumB class has associations with the following classes:

Class: MsMITrCourseB iscontainedin

Class: MsMITrCertificationB issatisfiedby

5.9.3.8 MsMITrDatabaseIFB Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Persistent Class:True

Purpose and Description:

This class represents the database where the training information will be stored and referenced.

Attributes:

None

Operations:

MsMITrDatabaseIFB - This operation is the constructor for this class.

Arguments:
Return Type:EcTVoid
Privilege:Public

RetrieveData - This operation extracts data from the database.

Arguments:

Return Type:RWCString

Privilege:Public

SetDatabase - This operation initializes the database with training information.

Arguments:

Return Type:EcTVoid

Privilege:Public

StoreData - This operation stores data into the database.

Arguments:

Return Type:EcTVoid

Privilege:Public

~MsMITrDatabaseIFB - This operation is the destructor for this class.

Arguments:

Return Type:EcTVoid

Privilege:Public

Associations:

The MsMITrDatabaseIFB class has associations with the following classes:

Class: MsMITrMgrB accesses

5.9.3.9 MsMITrEvaluationB Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class maintains the course evaluations completed by the trainees.

Attributes:

comments - This attribute represents any comments the trainee makes on the evaluation.

Data Type:RWCString

Privilege:Private

Default Value:

completionDate - This attribute represents the date that the training course was completed by the trainee.

Data Type:RWDate

Privilege:Private

Default Value:

courseID - This attribute represents the course for which the evaluation is being written.
Data Type:RWCString
Privilege:Private
Default Value:

evaluationID - This attribute represents a unique value by which each evaluation will be identified.
Data Type:RWCString
Privilege:Private
Default Value:

instructorID - This attribute represents the instructor who taught the course for which this evaluation is being written.
Data Type:RWCString
Privilege:Private
Default Value:

traineeID - This attribute represents the trainee who has completed a particular evaluation. This field will be optional.
Data Type:RWCString
Privilege:Private
Default Value:

Operations:

DisplayEvaluation
Arguments:evaluationID, courseID
Return Type:EcTVoid
Privilege:Public

MsMITrEvaluationB - This operation is the constructor for this class.
Arguments:
Return Type:Void
Privilege:Public

PrintEvaluation - This operation prints the particular evaluation.
Arguments:evaluationID
Return Type:EcTVoid
Privilege:Public

StoreEvaluation - This operation stores the evaluation in a private directory reserved for each course.
Arguments:evaluationID

Return Type:EcTVoid

Privilege:Public

~MsMITrEvaluationB - This operation is the destructor for this class.

Arguments:

Return Type:Void

Privilege:Public

Associations:

The MsMITrEvaluationB class has associations with the following classes:

Class: MsMITrTraineeB submits

5.9.3.10MsMITrInstructorB Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class contains all the instructors for the training courses.

Attributes:

instructorEmployer - This attribute represents the name of the company by whom the instructor is employed.

Data Type:RWCString

Privilege:Private

Default Value:

instructorFirstName - This attribute represents the first name of the instructor.

Data Type:RWCString

Privilege:Private

Default Value:

instructorID - This attribute represents a unique value by which each instructor will be identified.

Data Type:RWCString

Privilege:Private

Default Value:

instructorLastName - This attribute represents the last name of the instructor.

Data Type:RWCString

Privilege:Private

Default Value:

instructorMidInit - This attribute represents the middle initial of the instructor.

Data Type:RWCString
Privilege:Private
Default Value:

Operations:

AddInstructor - This operation adds an instructor to the database.

Arguments:
Return Type:EcTVoid
Privilege:Public

DeleteInstructor - This operation removes an instructor from the database.

Arguments:instructorID
Return Type:EcTVoid
Privilege:Public

MsMITrInstructorB - This operation is the constructor for this class.

Arguments:
Return Type:Void
Privilege:Public

UpdateInstructorInfo - This operation updates the information for a particular instructor in the database.

Arguments:instructorID
Return Type:EcTVoid
Privilege:Public

~MsMITrInstructorB - This operation is the destructor for this class.

Arguments:
Return Type:Void
Privilege:Public

Associations:

The MsMITrInstructorB class has associations with the following classes:
MsMITrCourseLocationB (Aggregation)

5.9.3.11MsMITrInventoryIFB Class

Parent Class:Not Applicable
Public:No
Distributed Object:No
Purpose and Description:

This class contains the results of searching the inventory for particular training materials.

Attributes:

invLocation - This attribute holds the location of materials in inventory.

Data Type:RWCString

Privilege:Private

Default Value:

itemNumber - This attribute holds the item numbers of training materials.

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

GetMaterialItem - This operation returns the item number of the particular material being searched.

Arguments:materialID

Return Type:RWCString

Privilege:Public

GetMaterialLocation - This operation returns a value indicating where the material is located in inventory.

Arguments:itemNumber

Return Type:RWCString

Privilege:Public

MsMITrInventoryIFB - This operation is the constructor for this class.

Arguments:

Return Type:Void

Privilege:Public

~MsMITrInventoryIFB - This operation is the destructor for this class.

Arguments:

Return Type:Void

Privilege:Public

Associations:

The MsMITrInventoryIFB class has associations with the following classes:

Class: MsMITrMaterialB uses

5.9.3.12MsMITrMStaffIFB Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class provides the interface to the members of the M&O staff who will have access privileges to the training database.

Attributes:

catalogID - This attribute contains values that uniquely identify the Certification Skills Catalogs.

Data Type:RWCString

Privilege:Private

Default Value:

materialID - This attribute represents a unique value by which the training materials will be identified.

Data Type:RWCString

Privilege:Private

Default Value:

requirementID - This attribute contains values that uniquely identify the training requirements.

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

GetCatalog

Arguments:taskDescription

Return Type:EcTVoid

Privilege:Public

GetMaterials - This operation returns the unique identifiers of all the training materials required for a particular subject area.

Arguments:subjectArea

Return Type:EcTVoid

Privilege:Public

GetRequirements - This operation returns the unique identifier of all requirements associated with the ECS program area.

Arguments:programArea

Return Type:EcTVoid

Privilege:Public

MsMITrMOSstaffIFB - This operation is the constructor for this class.

Arguments:
Return Type:Void
Privilege:Public

~MsMITrMOSstaffIFB - This operation is the destructor for this class.

Arguments:
Return Type:Void
Privilege:Public

Associations:

The MsMITrMOSstaffIFB class has associations with the following classes:

Class: MsMITrMgrB uses

5.9.3.13MsMITrMaterialB Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class maintains a list of all materials associated with each training course.

Attributes:

courseID - This attribute represents the course for which the material is needed.

Data Type:RWCString

Privilege:Private

Default Value:

location - This attribute represents the place (site) where each material item is currently located.

Data Type:RWCString

Privilege:Private

Default Value:

materialID - This attribute represents a unique value by which each training material will be identified.

Data Type:RWCString

Privilege:Private

Default Value:

materialTitle - This attribute represents the title of each training material associated with each training course.

Data Type:RWCString

Privilege:Private

Default Value:

quantity - This attribute represents the number of copies currently in print of each training material.

Data Type:EcTInt

Privilege:Private

Default Value:

type - This attribute represents the type of training material; e.g., textbook, student workbook, overhead slides, and the like.

Data Type:RWCString

Privilege:Private

Default Value:

version - This attribute represents the current version number of each material item for each course in order to keep the courses up to date.

Data Type:EcTInt

Privilege:Private

Default Value:

Operations:

DisplayMaterial - This operation displays the training material to the screen.

Arguments:materialID

Return Type:EcTVoid

Privilege:Public

DisplayMaterialList - This operation displays a list of available materials for a particular course.

Arguments:courseID

Return Type:EcTVoid

Privilege:Public

LocateMaterial - This operation asks the Inventory to locate the particular material.

Arguments:materialID

Return Type:RWCString

Privilege:Public

MsMTrMaterialB - This operation is the constructor for this class.

Arguments:

Return Type:Void

Privilege:Public

PrintMaterial - This operation prints the particular training material.

Arguments:materialID

Return Type:EcTVoid
Privilege:Public

UpdateMaterial - This operation updates the particular training material.

Arguments:materialID
Return Type:EcTVoid
Privilege:Public

~MsMITrMaterialB - This operation is the destructor for this class.

Arguments:
Return Type:Void
Privilege:Public

Associations:

The MsMITrMaterialB class has associations with the following classes:

Class: MsMITrCourseB uses

Class: MsMITrInventoryIFB uses

5.9.3.14MsMITrMgrB Class

Parent Class:EcPfManagedServer

Public:No

Distributed Object:No

Purpose and Description:

This class controls the overall operation of the training management function. Whatever or whoever wants to access the training management system does so through this class.

Attributes:

siteECS - This attribute allows a user to query for training information on a particular ECS site.

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

ConfirmScheduleUpdate - This operation calls UpdateSchedule in the MsMITrScheduleB class to update the schedule and return a confirmation that the schedule has been updated.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL:

DisplayConfirmation - This operation displays the confirmation that the schedule has been updated.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL:

DisplayMainMenu - This operation displays the main menu screen of the training management service database.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL: No PDL

Exit - This operation exits the system.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL: No PDL

GenerateGroundEvent - This operation generates a ground event when training is scheduled for purposes of resource planning and scheduling and mode management.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL: No PDL

MsMITrMgrB - This operation is the constructor for this class.

Arguments:

Return Type:Void

Privilege:Public

PDL: No PDL

RequestCourse - This operation asks the operator if s/he would like to schedule another course for the current trainee.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL:

RequestExit - This operation asks the operator if s/he would like to exit the system.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL:

RequestResources - This operation sends a request to Resource Planning to plan certain resources to be utilized for training.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL: No PDL

RequestTrainee - This operation asks the operator if s/he would like to schedule another trainee for training.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL:

~MsMITrMgrB - This operation is the destructor for this class.

Arguments:

Return Type:Void

Privilege:Public

PDL: No PDL

Associations:

The MsMITrMgrB class has associations with the following classes:

Class: MsMITrDatabaseIFB accesses

Class: MsMITrScheduleB isupdatedby

Class: MsRgRepWriterB isusedby

Class: PIRpResourceReservation isusedby

Class: MsMITrMOStaffIFB uses

5.9.3.15MsMITrScheduleB Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class maintains a current schedule of trainees, which courses these trainees are attending, where the courses are being held, and when the courses were taken.

Attributes:

completionDate - This attribute represents the date when a trainee is scheduled to complete a training course.

Data Type:RWDate

Privilege:Private

Default Value:

courseID - This attribute identifies the course that a trainee is scheduled to attend.

Data Type:RWCString

Privilege:Private

Default Value:

location - This attribute represents the place where the scheduled course is to be administered.

Data Type:RWCString

Privilege:Private

Default Value:

resources - This attribute represents the resources required to be taking a scheduled course.

Data Type:RWCString

Privilege:Private

Default Value:

startDate - This attribute represents the date the trainee is scheduled to begin attending a training course.

Data Type:RWDate

Privilege:Private

Default Value:

traineeID - This attribute identifies the employee currently scheduled to attend a training course.

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

DisplaySchedule - This operation displays the training schedule to the screen.

Arguments:

Return Type:EcTVoid

Privilege:Public

MsMITrScheduleB - This operation is the constructor for this class.

Arguments:

Return Type:Void

Privilege:Public

PrintSchedule - This operation prints the training schedule.

Arguments:

Return Type:EcTVoid
Privilege:Public

UpdateSchedule - This operation updates training schedules.

Arguments:
Return Type:EcTVoid
Privilege:Public

~MsMITrScheduleB - This operation is the destructor for this class.

Arguments:
Return Type:Void
Privilege:Public

Associations:

The MsMITrScheduleB class has associations with the following classes:

Class: MsMITrMgrB isupdatedby

5.9.3.16 MsMITrTraineeB Class

Parent Class:MsAcUsrProfile

Public:No

Distributed Object:No

Purpose and Description:

This class maintains a current record of all employees project-wide who require training or have already completed training.

Attributes:

creditsEarned - This attribute represents the number of course credits each trainee has accumulated.

Data Type:EcTInt

Privilege:Private

Default Value:

employerID - This attribute represents the company by whom each trainee is employed.

Data Type:RWCString

Privilege:Private

Default Value:

traineeID - This attribute represents a unique value by which each trainee will be identified.

Data Type:RWCString

Privilege:Private

Default Value:

traineeListID - This attribute represents a unique value by which each type of trainee list will be identified.

Data Type:RWCString

Privilege:Private

Default Value:

Operations:

AddTrainee - This operation allows for the addition of new trainees to the current list.

Arguments:

Return Type:EcTVoid

Privilege:Public

PDL: No PDL

DeleteTrainee - This operation allows for the removal of trainees from the current database.

Arguments:traineeID

Return Type:EcTVoid

Privilege:Public

PDL: No PDL

DisplayInfoList - This operation displays a selection list of topics of trainee information.

Arguments:traineeListID

Return Type:EcTVoid

Privilege:Public

PDL:

DisplayTraineeInfo - This operation displays the information for the specified trainee to the screen.

Arguments:traineeID

Return Type:EcTVoid

Privilege:Public

PDL:

DisplayTraineeList - This operation displays the selected trainee list to the screen.

Arguments:traineeListID

Return Type:EcTVoid

Privilege:Public

PDL:

MsMITrTraineeB - This operation is the constructor for this class.

Arguments:

Return Type:Void

Privilege:Public
PDL: No PDL

PrintTraineeInfo - This operation prints the information about the particular trainee.

Arguments:traineeID
Return Type:EcTVoid
Privilege:Public
PDL: No PDL

RequestOrgName - This operation asks the user to enter an organization name.

Arguments:
Return Type:EcTVoid
Privilege:Public
PDL:

RequestTraineeList - This operation asks the operator which trainee list s/he would like to display.

Arguments:
Return Type:EcTVoid
Privilege:Public
PDL:

RequestVerification - This operation asks the operator to verify whether the trainee name is correct.

Arguments:
Return Type:EcTVoid
Privilege:Public
PDL:

UpdateTrainee - This operation updates the information in the database for the specified trainee.

Arguments:traineeID
Return Type:EcTVoid
Privilege:Public
PDL: No PDL

~MsMITrTraineeB - This operation is the destructor for this class.

Arguments:
Return Type:Void
Privilege:Public
PDL: No PDL

Associations:

The MsMITrTraineeB class has associations with the following classes:

Class: MsMITrCourseB attends

Class: MsMITrCertificationB receives
Class: MsMITrEvaluationB submits
MsMITrMgrB (Aggregation)

5.9.3.17MsMITrTrainingCostB Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class contains the cost of the training courses, both per student cost and total cost.

Attributes:

courseCost - This attribute represents the cost of the training course on a per-student or credit basis.

Data Type:EcCurrency

Privilege:Private

Default Value:

courseID - This attribute represents the course for which the cost will be determined.

Data Type:RWCString

Privilege:Private

Default Value:

currentEnrollment - This attribute represents the number of trainees currently enrolled in a particular training course.

Data Type:EcTInt

Privilege:Private

Default Value:

totalCost - This attribute represents the calculated total cost for the particular training course, derived from the course cost and the current enrollment in the course.

Data Type:EcCurrency

Privilege:Private

Default Value:

This is a Derived Attribute.

Operations:

ComputeTotalCost - This operation computes the total cost for the particular training course.

Arguments:courseCost, currentEnrollment, courseID

Return Type:EcCurrency

Privilege:Public
PDL: // for the particular courseID
// totalCost = courseCost * currentEnrollment

DisplayCost - This operation displays the calculated cost to the screen.

Arguments:totalCost
Return Type:EcTVoid
Privilege:Public
PDL: // if (cost has been calculated)
// display the cost in a view box

MsMITrTrainingCostB - This operation is the constructor for this class.

Arguments:
Return Type:Void
Privilege:Public

PrintCost - This operation prints out a report of the training cost.

Arguments:
Return Type:EcTVoid
Privilege:Public

~MsMITrTrainingCostB - This operation is the destructor for this class.

Arguments:
Return Type:Void
Privilege:Public
PDL: No PDL

Associations:

The MsMITrTrainingCostB class has associations with the following classes:
None

5.9.3.18MsRgRepWriterB Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This class is the COTS report writer product associated with the management DBMS.

Attributes:

None

Operations:

None

Associations:

The MsRgRepWriterB class has associations with the following classes:

Class: MsMlTrMgrB is used by

5.9.3.19 PIRpResourceReservation Class

Parent Class: Not Applicable

Public: No

Distributed Object: No

Purpose and Description:

The MsMgSchedulingMgrB class represents the scheduling for the Software Distribution Management Service.

Attributes:

None

Operations:

None

Associations:

The PIRpResourceReservation class has associations with the following classes:

Class: MsMlTrMgrB is used by

5.9.4 Training Management Dynamic Model

This scenario traces the actions associated with scheduling a trainee for a training course. The scenario is depicted in Figure 5.9-3.

5.9.4.1 Beginning Assumptions

An M&O operator has already been authorized to access and update the training system. All necessary training information has already been entered into the database. The operator has entered the system's Main Menu.

5.9.4.2 Interfaces with Other Subsystems and Segments

None.

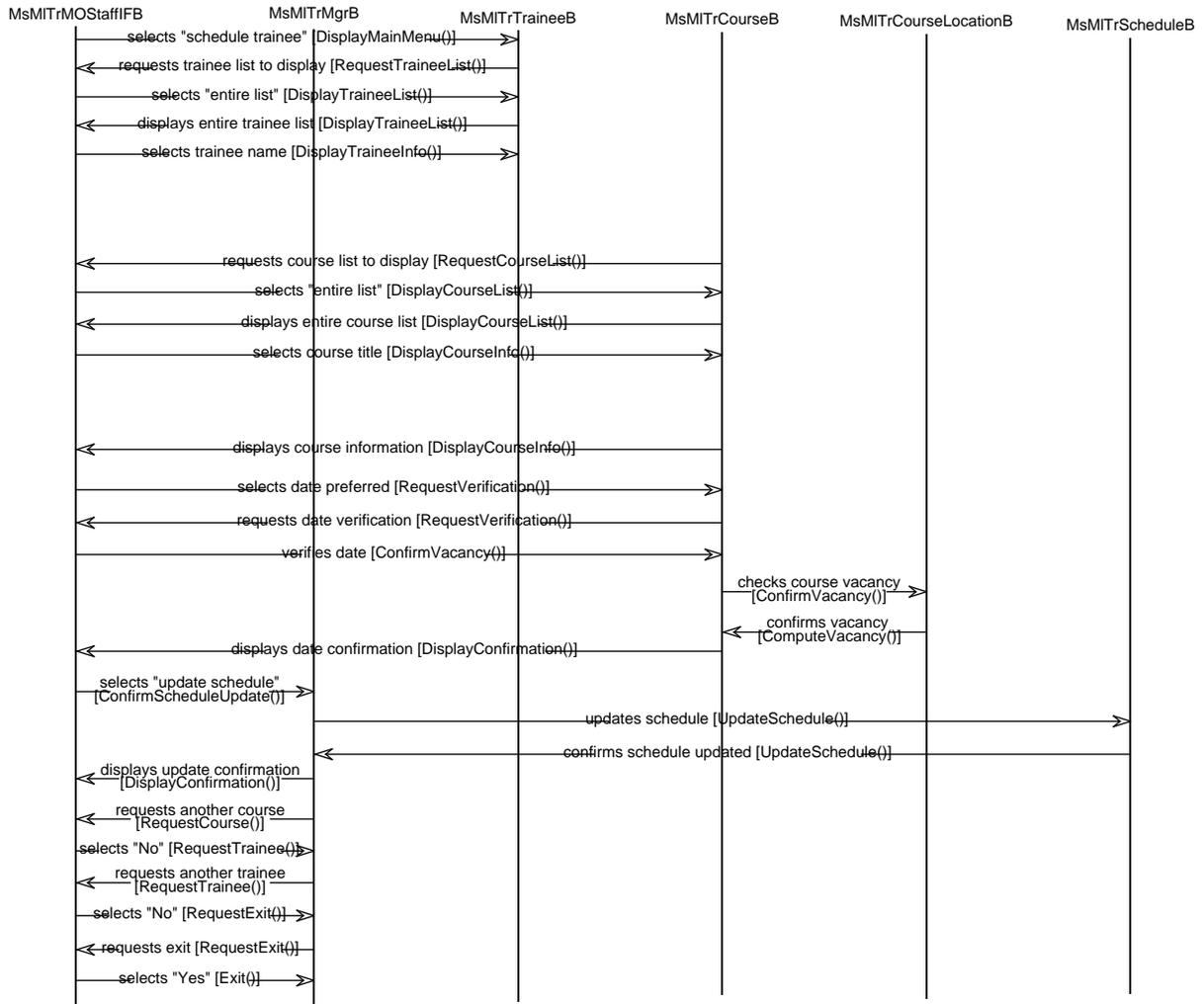


Figure 5.9-3. Training Scenario: Scheduling a Trainee for a Course

5.9.4.3 Stimulus

A training course is necessary to support a trainee's training requirements.

5.9.4.4 Participating Classes From the Object Model

MsMITrMOSstaffIFB

MsMITrMgrB

MsMITrCourseLocationB

MsMITrScheduleB

MsMITrTraineeB

MsMITrCourseB

5.9.4.5 Beginning System, Segment and Subsystem State(s)

MsMITrMgrB displays a Main Menu.

5.9.4.6 Ending State

MsMITrScheduleB is updated and MsMITrMgrB exits.

5.9.4.7 Scenario Description

- MsMITrMOSstaffIFB selects "Schedule Trainee" from the Main Menu. [DisplayMainMenu()]
- MsMITrTraineeB requests which trainee list to display. [RequestTraineeList()]
- MsMITrMOSstaffIFB selects "Entire List". [DisplayTraineeList()]
- MsMITrTraineeB displays entire list of trainees alphabetically. [DisplayTraineeList()]
- MsMITrMOSstaffIFB selects trainee name to be scheduled. [DisplayTraineeInfo()]
- MsMITrCourseB requests which course list to display. [RequestCourseList()]
- MsMITrMOSstaffIFB selects "Entire List". [DisplayCourseList()]
- MsMITrCourseB displays entire list of courses. [DisplayCourseList()]
- MsMITrMOSstaffIFB selects title of course to be scheduled. [DisplayCourseInfo()]
- MsMITrCourseB displays all information relevant to the selected course. [DisplayCourseInfo()]
- MsMITrMOSstaffIFB selects preferred date to be scheduled. [RequestVerification()]
- MsMITrCourseB requests date verification. [RequestVerification()]
- MsMITrMOSstaffIFB verifies date. [ConfirmVacancy()]
- MsMITrCourseB checks, through MsMITrCourseLocationB, whether the course is vacant on that date. [ConfirmVacancy()]
- MsMITrCourseLocationB confirms course vacancy. [ComputeVacancy()]
- MsMITrCourse displays confirmation. [DisplayConfirmation()]
- MsMITrMOSstaffIFB selects "Update Schedule". [ConfirmScheduleUpdate()]
- MsMITrMgrB updates MsMITrScheduleB. [UpdateSchedule()]
- MsMITrScheduleB confirms updates. [UpdateSchedule()]
- MsMITrMgrB displays confirmation. [DisplayConfirmation()]
- MsMITrMgrB requests another course for the same trainee. [RequestCourse()]
- MsMITrMOSstaffIFB selects "No". [RequestTrainee()]
- MsMITrMgrB requests another trainee to be scheduled. [RequestTrainee()]
- MsMITrMOSstaffIFB selects "No". [RequestExit()]
- MsMITrMgrB requests to exit. [RequestExit()]
- MsMITrMOSstaffIFB selects "Yes". [Exit()]
- MsMITrMgrB exits gracefully. [Exit()]

5.9.5 Training Management Structure

This service provides the functionality necessary to manage training schedules for the ECS project, certification of M&O employees, training course cost information, and evaluation of training courses. The object classes it includes are shown in Table 5.9-1.

Table 5.9-1. Training Manager Components

Object Class	Implementation
MsMITrMgrB	Custom
MsMITrMOSStaffIFB	Custom
MsMITrScheduleB	Custom
MsMITrTraineeB	Custom
MsMITrCourseB	Custom
MsMITrCourseLocationB	Custom
MsMITrTrainingCostB	Custom
MsMITrEvaluationB	Custom
MsMITrCurriculumB	Custom
MsMITrCertificationB	Custom
MsMITrCertSkillsCatalogB	Custom
MsMITrInventoryIFB	Custom
MsMITrInstructorB	Custom
MsMITrRequirementsB	Custom
MsMITrMaterialB	Custom

5.10 Policies and Procedures Management

5.10.1 Policies and Procedures Management Overview

The Policies and Procedures Management service includes providing methods to perform ready access and lookup to the policies and procedures established by ESDIS. Policies and procedures, when applied uniformly, ensure that all M&O Staff actions will be standard across sites. Policies are instituted to determine how M&O Staff should manage a situation. Procedures are established to communicate to M&O Staff how to perform procedural actions. The functions of the Policies and Procedures Manager will be satisfied by reusing the Document Data Server, Ingest, Client, and by using the bulletin board service. Also, the policy and procedures documents will be stored in ClearCase for Configuration Management of the documents.

The Policies and Procedures Management Service will provide the capability to prepare, store, maintain, and make available for distribution ECS policies and procedures as well as site specific policies and procedures. The Service will also post to the bulletin board service information on ECS status, events, and news.

5.10.2 Policies and Procedures Management Context

Figure 5.10-1 contains the context diagram for the Policies and Procedures Management. The searching, viewing, and retrieving of the documents is done via the Web interface provided by the Client Subsystem. The GUI provided to store the document is done via the Ingest Subsystem GUI. The actual storage of the documents are done via the Document Data Server.

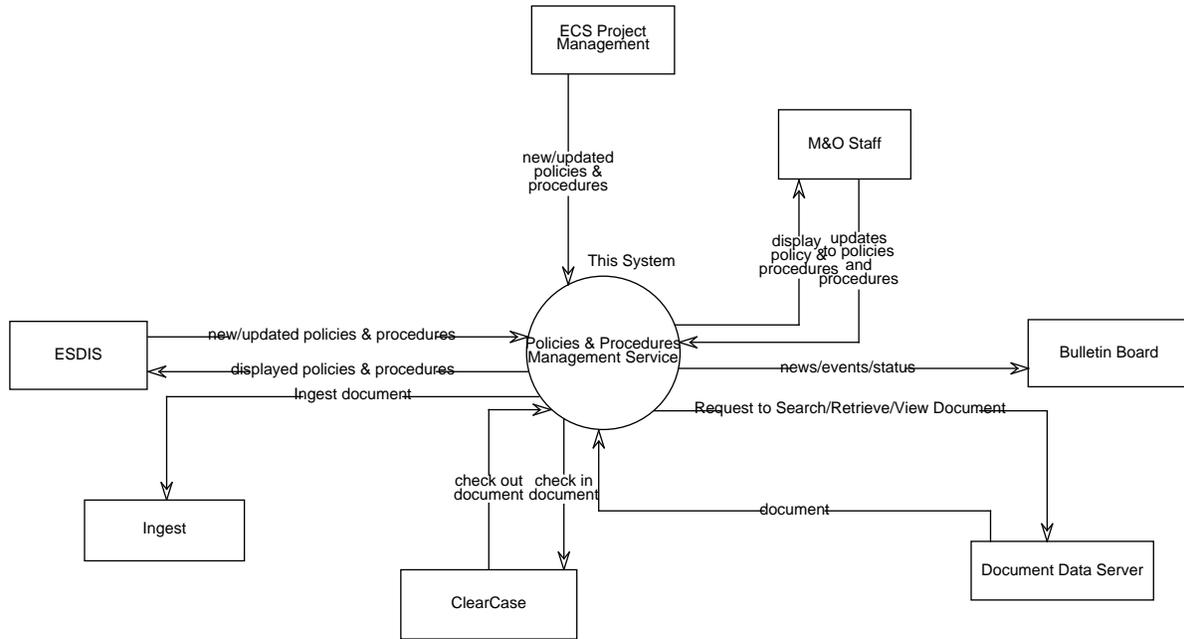


Figure 5.10-1. Policies and Procedures Management Context Diagram

5.10.3 Policies and Procedures Management Object Model

Figure 5.10-2, the Object Model for Policies and Procedures Management Service, depicts the major classes and their associations with one another. These are described below:

5.10.3.1 CsBBMailRelA Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

The MsBulletinBoardB class represents the posting to the bulletin board of the policies and procedures.

Attributes:

None

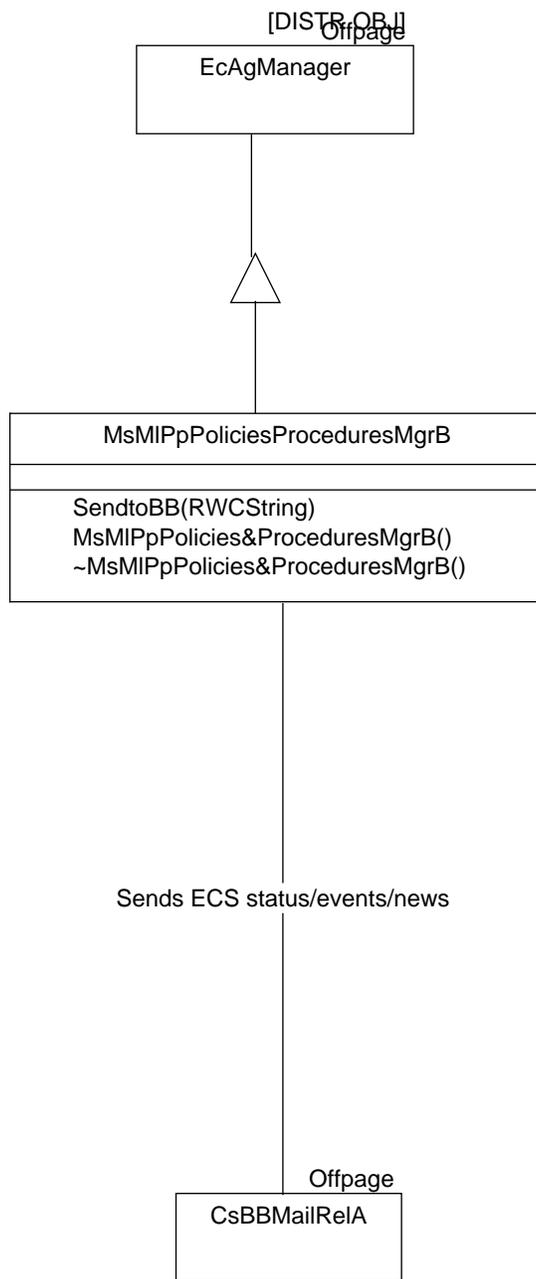


Figure 5.10-2. Policies and Procedures Management Object Model

Operations:

None

Associations:

The CsBBMailRelA class has associations with the following classes:
Class: MsMIPpPoliciesProceduresMgrB SendsECSstatus/events/news

5.10.3.2EcAgManager Class

Parent Class:Not Applicable

Public:No

Distributed Object:Yes

Purpose and Description:

The EcAgManager is an OODCE distributed object. The ECS applications use the server side of EcAgManager to communication with the subagent. The subagent uses the client side of EcAgManager to communicate with the ECS applications.

Attributes:

None

Operations:

None

Associations:

The EcAgManager class has associations with the following classes:
None

5.10.3.3MsMIPpPoliciesProceduresMgrB Class

Parent Class:EcAgManager

Public:No

Distributed Object:No

Purpose and Description:

This Policies and Procedures Manager class controls the policies and procedures for the ECS project.

Attributes:

All Attributes inherited from parent class

Operations:

MsMIPpPolicies&ProceduresMgrB - This operation is the constructor for the class.

Arguments:

SendtoBB - This operation sends the message to the Bulletin Board.

Arguments:RWCString

~MsMIPpPolicies&ProceduresMgrB - This operation is the destructor for the class.

Arguments:

Associations:

The MsMIPpPoliciesProceduresMgrB class has associations with the following classes:

Class: CsBBMailRelA SendsECSstatus/events/news

5.10.4 Policies and Procedures Management Dynamic Model

This scenario traces the actions associated with an authorized M&O Staff member performing an update to an existing policy. The scenario is depicted in Figure 5.10--3.

5.10.4.1 Beginning Assumptions

This scenario applies to authorized M&O Staff correcting or updating text (e.g., paragraph or other quantity) of either a ECS policy or a procedure.

5.10.4.2 Interfaces with Other Subsystems and Segments

Client

Ingest

Document Data Server

IDG

5.10.4.3 Stimulus

ESDIS Management, ECS Project Management, or any other relevant authority has determined a change to an existing policy is necessary. The change is communicated to authorized M&O Staff who will update the policy.

5.10.4.4 Participating Classes From the Object Model

MsMIPpPoliciesProceduresMgrB

MsBulletinBoardB

5.10.4.5 Beginning System, Segment and Subsystem State(s)

Normal Operations.

5.10.4.6 Ending State

Policy has been updated in the Document Data Server and a message is sent to the Bulletin Board indicating that a policy has been updated.

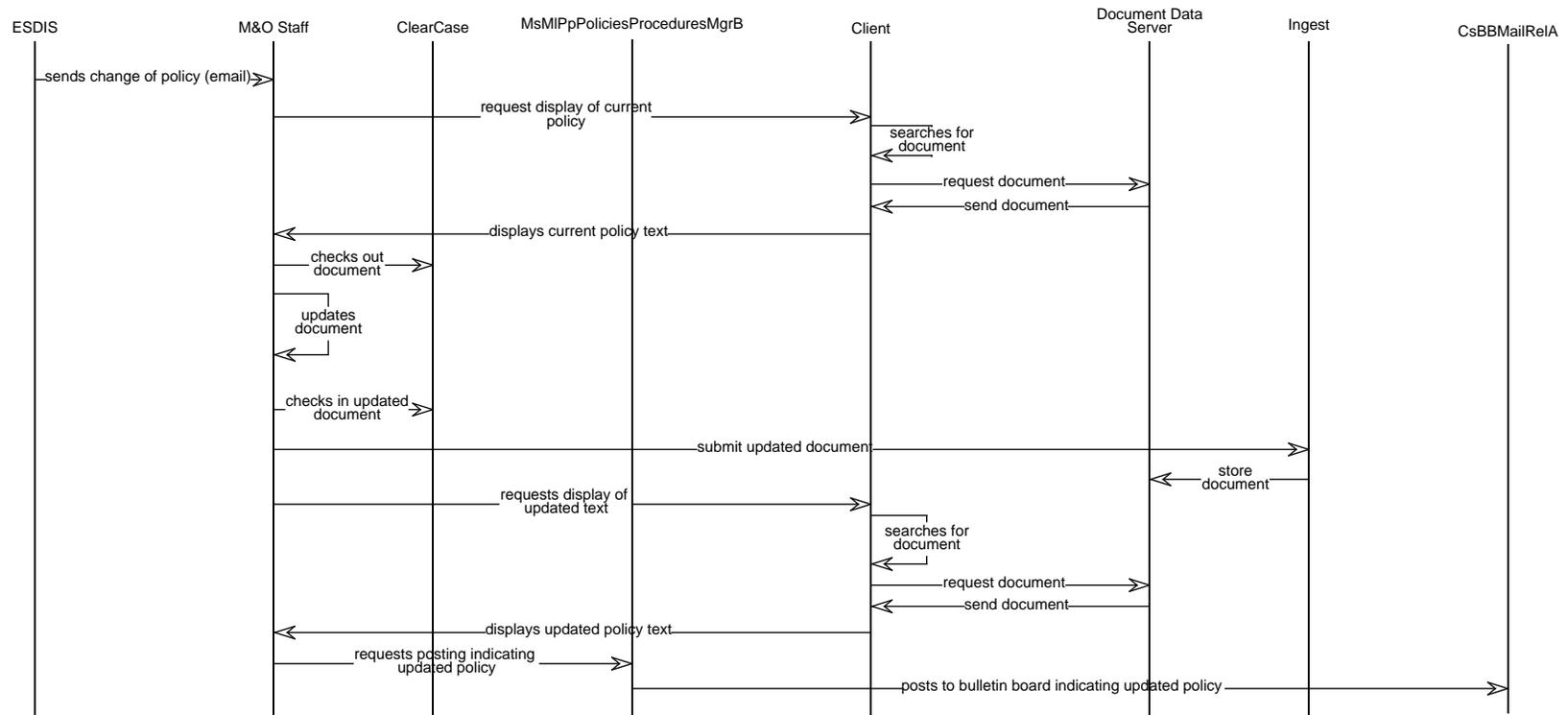


Figure 5.10-3. Policies and Procedures Scenario: Changing a Policy/Procedure Paragraph

5.10.4.7 Scenario Description

ESDIS sends change of policy via email to M&O Staff.

M&O Staff via the Client interface requests to see the current policy.

Client searches for the document.

Client requests document from Document Data Server.

Document Data Server returns requested document

Document is displayed via the Client interface to the M&O Staff.

M&O Staff checks out of ClearCase the document such that the modifications to documents are under CM control.

M&O Staff updates document.

M&O Staff checks in the document into ClearCase.

M&O Staff submits the updated document via the Ingest GUI.

Ingest stores the document on the Document Data Server.

M&O Staff via the Client interface requests to see the current policy.

Client searches for the document.

Client requests document from Document Data Server.

Document Data Server returns requested document

Document is displayed via the Client interface to the M&O Staff.

M&O Staff requests a posting be sent from MsMlPpPoliciesProceduresMgrB to the Bulletin Board indicating that a policy has been updated.

MsMlPpPoliciesProceduresMgrB posts indicated message to the Bulletin Board.

5.10.5 Policies and Procedures Management Structure

Table 5.10-1 lists the components of the Policies and Procedures Management Service.

Table 5.10-1. Policies and Procedures Manager Components

Object Class	Implementation
MsMIPpPoliciesProceduresMgrB	Custom

5.11 Inventory/Logistics/Maintenance (ILM) Management

5.11.1 ILM Management Overview

The Inventory/Logistics/Maintenance (ILM) Manager provides the capability for the M&O staff to manage inventory items, provide logistics management, and control maintenance. Several COTS packages are currently being evaluated under vendor proposals to satisfy the functional requirements of these three required services.

The ILM Manager can control, track, and maintain information on ECS inventory items. ECS inventory items may be ECS purchased equipment or GFE, and includes hardware, COTS software, documentation, training material, and consumable supplies. The ILM Manager's inventory activities will include maintaining an up-to-date list of all equipment items, controlled items, spare parts, and consumable supplies. Controlled items will be tracked throughout their service life until disposal. An inventory item can be made up of a combination of other inventory items (an assembly or sub-assembly). For example, an ECS workstation with a bar-code control number assigned may be comprised of a keyboard, mouse, monitor, and CPU. Each item in this example can be individually tracked within the ILM Manager, so that a monitor item can be part of a workstation and another identical monitor item can be tracked as a spare. A consumable item is one in which only the amount on-hand/ordered is maintained, and the item is removed from inventory control when distributed. For example, paper for a laser printer may be ordered by the box, and the box may have 12 reams in it. Once the box has been removed from the inventory (distributed to the computer floor for use), the number of reams left in the box is not tracked by the ILM Manager. A spare item is one which has at least one not in use and available for issue; this quantity can be queried from the ILM Manager.

The ILM Manager will track spare parts and components that are used to repair equipment end-items, consumable items (items that are used up or distributed during operations as a matter of course, such as paper and laser toner cartridges), and line replaceable units (LRU) which are returned to a third party vendor for restoration, overhaul, or other service. The ILM Manager provides the ability to do trend analysis of parts and LRU failure rates to spot trends that demand resolution. The ILM Manager will likewise track usage trends that enable the M&O Staff to set the most economical supply re-order point levels consistent with the requirements of operational readiness. Finally, the ILM Manager provides a variety of reports to enable the M&O staff to perform their varied responsibilities on a day-to-day basis.

An item tracked by the ILM Manager is composed of one or more of the following attributes, as shown on the object model: item identification, manufacturer, description, model number, part number, version, serial number, cost, date received, component of, status, condition, location, custodian, maintainer, ECS bar-code data, maintenance agreement, license agreement, warranty list, warranty expiration, etc. The ILM Manager can add items, update items, remove items, list

items according to operator entered criteria, generate reports on items based on operator entered criteria, or change the status of an item.

There will be an instance of the ILM Manager at both the SMC and at each of the sites. The ILM Manager at the SMC will provide M&O Staff a view of the system-wide inventory. The ILM Manager at each sites will provide M&O Staff a view of their respective sites inventory.

The ILM Manager's logistics activities involve the various actions to track all pertinent supplies (both consumable and non-consumable) needed for maintenance as well as routine replenishments of the system and equipment end-items (such as spare parts for maintenance, extra paper, and toner cartridges for printers).

The ILM Manager's maintenance function will provide the capability for the M&O staff to track preventive and corrective maintenance actions to completion - including the update of configuration management, logistics, and inventory records/baselines.

The ILM Manager receives from the Trouble Ticket Service initiation of corrective maintenance and preventive maintenance actions. Both types of maintenance activities can be tied to the scheduling function so that maintenance disruption to operations will be a minimum. The ILM Manager provides information on both corrective and preventive maintenance required; it also performs analysis on all corrective maintenance actions to identify site and system trends that can help to avoid future problems. The ILM Manager keeps information on inventory status changes to update the inventory as well as supply data changes (such as use of consumable supplies, line replaceable units, and spare parts) to update the respective logistics records.

5.11.2 ILM Management Context

Figure 5.11-1 contains the context diagram for ILM Management.

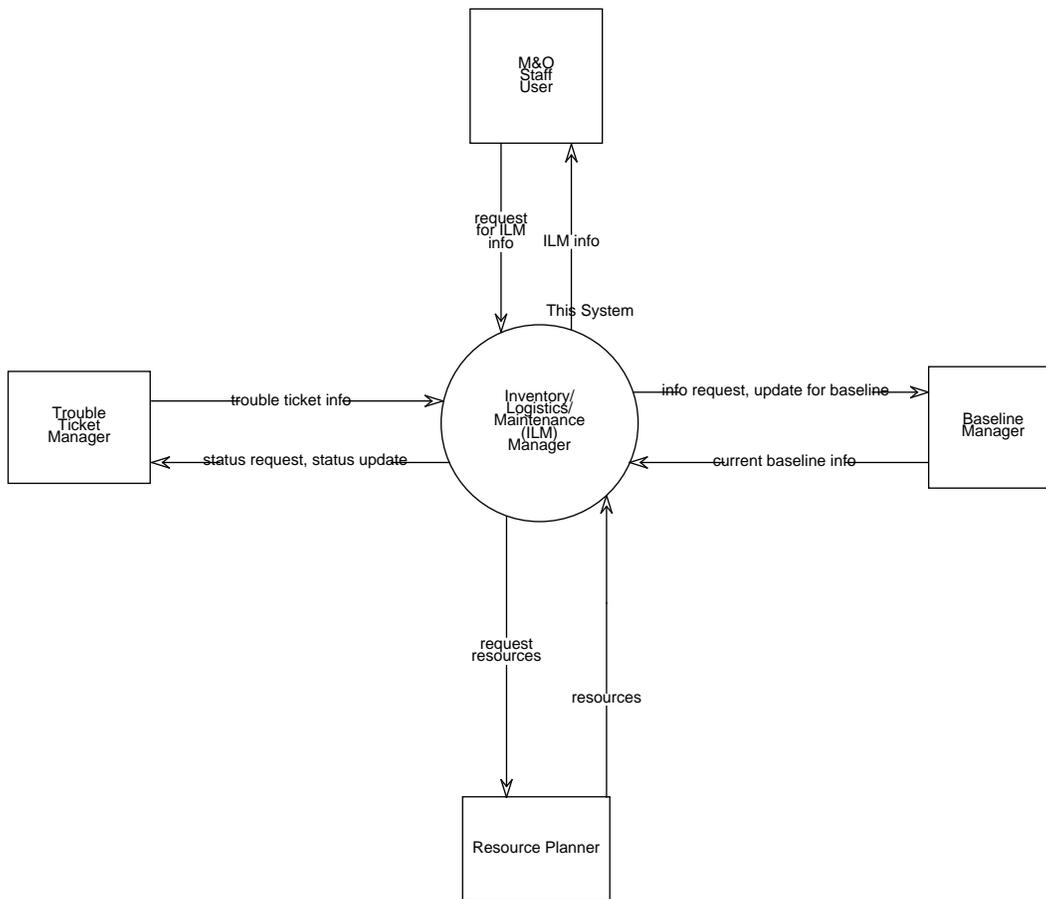


Figure 5.11-1. ILM Management Context Diagram

5.11.3 ILM Management Object Model

Figure 5.11-2, the Object Model for ILM Management, depicts the major classes and their associations with one another. These are described below:

5.11.3.1 EcAgCOTSManger Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

this abstract class embodies the characteristics and functionality of a manager object responsible for managing a single COTS process it encapsulates all MSS management application functions into a single class. The COTS proxy agent developer is responsible for inheriting from this class and specilizing it towards the COTS program to manage

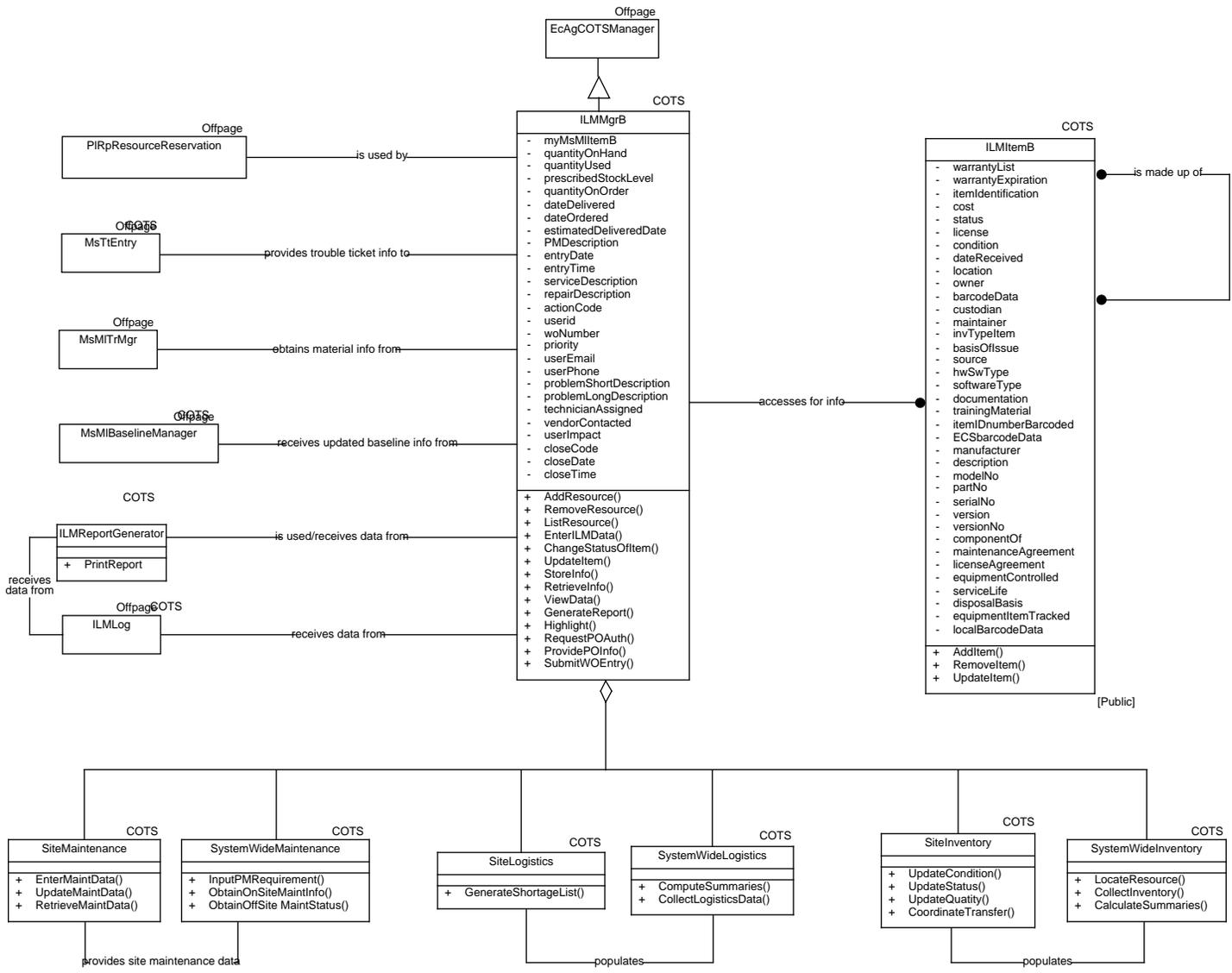


Figure 5.11-2. ILM Management Object Model

Attributes:

None

Operations:

None

Associations:

The EcAgCOTSManger class has associations with the following classes:

None

5.11.3.2ILMItemB Class

Parent Class:Not Applicable

Public:Yes

Distributed Object:No

Purpose and Description:

The ILM Item is the asset in the ECS inventory or held as an ECS supply (consumable or non-consumable) which is managed and/or maintained.

Attributes:

ECSbarcodeData - Gives the ECS bar-coded data.

Data Type:

Privilege:Private

Default Value:

barcodeData - Gives all the data that is stored on the ECS/local bar coding.

Data Type:

Privilege:Private

Default Value:

basisOfIssue - Indicates the unit(s) by/in which the item is counted (each, dozen, by the poihund, by the box, etc.).

Data Type:

Privilege:Private

Default Value:

componentOf - Indicates whether the item is a component or subassembly of a higher order assembly.

Data Type:

Privilege:Private

Default Value:

condition - UIndicates the current ocondition of the item.

Data Type:

Privilege:Private

Default Value:

cost - Indicates the original purchase cost of the item.

Data Type:

Privilege:Private

Default Value:

custodian - Indicates who is the assigned custodian for the asset/item.

Data Type:

Privilege:Private

Default Value:

dateReceived - Indicates the date that the item was received.

Data Type:

Privilege:Private

Default Value:

description - Gives the description of the item.

Data Type:

Privilege:Private

Default Value:

disposalBasis - Indicates if the item has any special disposal or turn-in instructions.

Data Type:

Privilege:Private

Default Value:

documentation - Indicates if the item is documentation and gives its designation.

Data Type:

Privilege:Private

Default Value:

equipmentControlled - Indicates whether the item is under any special controls (such as for high value or hazardous items).

Data Type:

Privilege:Private

Default Value:

equipmentItemTracked - Indicates whether the equipment item has any special system tracking such as maintenance up-time/down-time tracking (special interest).

Data Type:
Privilege:Private
Default Value:

hwSwType - Indicates whether the item is hardware or software.

Data Type:
Privilege:Private
Default Value:

invTypeItem - Indicates the type of inventory to which the item belongs (equipment item, line replaceable unit, consumable item, spare part).

Data Type:
Privilege:Private
Default Value:

itemIDnumberBarcoded - Indicates whether the item is bar-coded with an identification number.

Data Type:
Privilege:Private
Default Value:

itemIdentification - Unique alphanumeric code used to designate the item.

Data Type:
Privilege:Private
Default Value:

license - Indicates whether or not a software item is under license.

Data Type:
Privilege:Private
Default Value:

licenseAgreement - Indicates whether the item is under license from a vendor (especially for software).

Data Type:
Privilege:Private
Default Value:

localBarcodeData - Indicates any locally assigned bar-code data.

Data Type:
Privilege:Private
Default Value:

location - Indicates the location where the item is stored (designation locally specified).

Data Type:
Privilege:Private

Default Value:

maintainer - Indicates who is assigned to perform maintenance on the asset/item.

Data Type:

Privilege:Private

Default Value:

maintenanceAgreement - Indicates whether item is under a maintenance agreement with the vendor or a third-party vendor.

Data Type:

Privilege:Private

Default Value:

manufacturer - Indicates the name of the item manufacturer.

Data Type:

Privilege:Private

Default Value:

modelNo - Gives the model number of the item.

Data Type:

Privilege:Private

Default Value:

owner - Indicates who owns the item (Government/contractor; ECD project/other).

Data Type:

Privilege:Private

Default Value:

partNo - Gives the part number of the item.

Data Type:

Privilege:Private

Default Value:

serialNo - Gives the serial number of the item.

Data Type:

Privilege:Private

Default Value:

serviceLife - Indicates whether the item has a known or anticipated service life, such as battery shelf life.

Data Type:

Privilege:Private

Default Value:

softwareType - For software items, indicates what type of software (operating system,

application, utility, etc.).

Data Type:

Privilege:Private

Default Value:

source - Indicates where the item came from originally (such as the vendor's name).

Data Type:

Privilege:Private

Default Value:

status - Indicates the current status of the item.

Data Type:

Privilege:Private

Default Value:

trainingMaterial - Indicates whether the item is used for training purposes.

Data Type:

Privilege:Private

Default Value:

version - For software, whether the item is issued in versions or has a version number.

Data Type:

Privilege:Private

Default Value:

versionNo - For software, the version number assigned by the issuer.

Data Type:

Privilege:Private

Default Value:

warrantyExpiration - Indicates when the warranty on the item expires.

Data Type:

Privilege:Private

Default Value:

warrantyList - Indicates whether the item covered on a warranty list.

Data Type:

Privilege:Private

Default Value:

Operations:

AddItem - This operation adds an item to the inventory.

Arguments:

Return Type:Void
Privilege:Public

RemoveItem - This operation removes an item from the inventory.

Arguments:
Return Type:Void
Privilege:Public

UpdateItem - This operation updates the information about an item in the inventory.

Arguments:
Return Type:Void
Privilege:Public

Associations:

The ILMItemB class has associations with the following classes:

Class: ILMMgrB accessesforinfo

Class: ILMItemB ismadeupof

5.11.3.3ILMLog Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This ILMLog class is the COTS class that keeps an internal log of all actions that are performed in the ILM Manager. It can be recalled for later analysis.

Attributes:

None

Operations:

None

Associations:

The ILMLog class has associations with the following classes:

Class: ILMMgrB receivesdatafrom

Class: ILMReportGenerator receivesdatafrom

5.11.3.4ILMMgrB Class

Parent Class:EcAgCOTSManager

Public:No

Distributed Object:No

Purpose and Description:

This ILM Manager Class is responsible for managing attributes and operations that provide for inventory, logistics (supplies), and maintenance of ECS assets.

Attributes:

PMDescription - Description of preventive maintenance to be performed on ECS asset(s).

Data Type:

Privilege:Private

Default Value:

actionCode - Alphanumeric code used to designate particular types of actions taken (if used).

Data Type:

Privilege:Private

Default Value:

closeCode - Alphanumeric code used to designate the particular status at closure of the action(s).

Data Type:

Privilege:Private

Default Value:

closeDate - Date that the maintenance action was closed.

Data Type:

Privilege:Private

Default Value:

closeTime - Time that the maintenance action was closed.

Data Type:

Privilege:Private

Default Value:

dateDelivered - Date that item(s) (listed by a given name) in the (logistics) inventory were delivered.

Data Type:

Privilege:Private

Default Value:

dateOrdered - Date that the item(s) (listed by a given name) in the (logistics) inventory were ordered.

Data Type:

Privilege:Private

Default Value:

entryDate - Date that the particular ILM action was entered into the system.

Data Type:

Privilege:Private

Default Value:

entryTime - Time that the particular ILM action was entered into the system.

Data Type:

Privilege:Private

Default Value:

estimatedDeliveredDate - Date that item(s) (listed by a given name) in the (logistics) inventory are estimated for delivery.

Data Type:

Privilege:Private

Default Value:

myMsMIIItemB - This myMsMIIItemB attributes refers to the item in either the ECS asset inventory or the ECS logistics inventory which is managed by the ILM Manager.

Data Type:

Privilege:Private

Default Value:

prescribedStockLevel - Number of individual items (listed by a given name) that should be retained in stock; when the number drops belows this number, a re-order should be generated.

Data Type:

Privilege:Private

Default Value:

priority - Priority assigned to a given work order (if used).

Data Type:

Privilege:Private

Default Value:

problemLongDescription - Long description of the original trouble being tracked.

Data Type:

Privilege:Private

Default Value:

problemShortDescription - Short description of the original trouble being tracked.

Data Type:

Privilege:Private

Default Value:

quantityOnHand - Number of individual items (listed by a given name) that are present in the inventory and on-hand.

Data Type:

Privilege:Private

Default Value:

quantityOnOrder - Number of individual items (listed by a given name) in the (logistics) inventory that are on order.

Data Type:

Privilege:Private

Default Value:

quantityUsed - Number of individual items (listed by a given name) in the (logistics) inventory that have been used.

Data Type:

Privilege:Private

Default Value:

repairDescription - Description of the repair action required.

Data Type:

Privilege:Private

Default Value:

serviceDescription - Description of the particular maintenance service required.

Data Type:

Privilege:Private

Default Value:

technicianAssigned - Name of the technician who is assigned to evaluate/repair the problem.

Data Type:

Privilege:Private

Default Value:

userEmail - The E-mail address for reaching the user who had the original trouble.

Data Type:

Privilege:Private

Default Value:

userImpact - Short summary of the impact on the user's operation caused by the trouble.

Data Type:

Privilege:Private

Default Value:

userPhone - The phone number for reaching the user who had the original trouble.

Data Type:

Privilege:Private

Default Value:

userid - Identification code used to uniquely identify each user of the ILM Manager.

Data Type:

Privilege:Private

Default Value:

vendorContacted - Name of vendor contacted, if the maintenance is under contract.

Data Type:

Privilege:Private

Default Value:

woNumber - Unique number assigned to each work order for identification and tracking.

Data Type:

Privilege:Private

Default Value:

Operations:

AddResource - This operation adds an ECS asset/resource to the inventory.

Arguments:

Return Type:Void

Privilege:Public

ChangeStatusOfItem - This operation changes the status of an ECS asset/resource in the inventory.

Arguments:

Return Type:Void

Privilege:Public

EnterILMData

Arguments:

Return Type:Void

Privilege:Public

GenerateReport - This operation enables an authorized ECS user to generate a report about a needed subject.

Arguments:

Return Type:Void

Privilege:Public

Highlight - This operation enables a user to highlight a item entry.

Arguments:

Return Type:Void

Privilege:Public

ListResource - This operation lists an ECS asset/resource in the inventory.

Arguments:

Return Type:Void

Privilege:Public

ProvidePOInfo - This operation enables a user to provide information for generating a purchase order.

Arguments:

Return Type:Void

Privilege:Public

RemoveResource - This operation removes an ECS asset/resource from the inventory.

Arguments:

Return Type:Void

Privilege:Public

RequestPOAuth - This operation enables an authorized user to request a purchase order authorization number.

Arguments:

Return Type:Void

Privilege:Public

RetrieveInfo - This operation retrieves information about an ECS asset/item in the inventory.

Arguments:

Return Type:Void

Privilege:Public

StoreInfo - This operation stores information about an ECS asset/item in the inventory.

Arguments:

Return Type:Void

Privilege:Public

SubmitWOEntry - This operation enables a user to submit information for a work order entry.

Arguments:

Return Type:Void

Privilege:Public

UpdateItem - This operation updates the information about an ECS asset/item in the inventory.

Arguments:

Return Type:Void

Privilege:Public

ViewData - This operation enables an authorized user to view data about an ECS asset/item in the inventory.

Arguments:

Return Type:Void

Privilege:Public

Associations:

The ILMMgrB class has associations with the following classes:

Class: ILMItemB accessesforinfo

Class: ILMReportGenerator isused/receivesdatafrom

Class: PIRpResourceReservation isusedby

Class: MsMITrMgr obtainsmaterialinfofrom

Class: MsTtEntry providestroubleticketinfo

Class: ILMLog receivesdatafrom

Class: MsMIBaselineManager receivesupdatedbaselineinfofrom

5.11.3.5ILMReportGenerator Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

The ILMReportGenerator class is an internal COTS function for generating inventory, logistics, and/or maintenance reports.

Attributes:

None

Operations:

PrintReport - This PrintReport operation is the COTS actor for actually print out an inventory, logistics, or maintenance report.

Arguments:

Return Type:Void

Privilege:Public

Associations:

The ILMReportGenerator class has associations with the following classes:

Class: ILMMgrB isused/receivesdatafrom

Class: ILMLog receivesdatafrom

5.11.3.6MsMIBaselineManager Class

Parent Class:Not Applicable

Attributes:

None

Operations:

None

Associations:

The MsMIBaselineManager class has associations with the following classes:

Class: ILMMgrB receivesupdatedbaselineinfofrom

5.11.3.7MsMITrMgr Class

Parent Class:Not Applicable

Attributes:

None

Operations:

None

Associations:

The MsMITrMgr class has associations with the following classes:

Class: ILMMgrB obtainsmaterialinfofrom

5.11.3.8MsTtEntry Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

The MsTtEntry class models a request for action (Trouble Ticket) on a particular problem and the subsequent actions performed on it. This class contains the fields which are accessible to a user through the Web interface. This class will contain a trouble ticket object to be submitted to the Remedy Action Request System or a trouble ticket object retrieved from Remedy to be displayed to the user through the Web interface.

Attributes:

None

Operations:

None

Associations:

The MsTtEntry class has associations with the following classes:

Class: ILMMgrB providestroubleticketinfo

5.11.3.9PIRpResourceReservation Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

The MsMgSchedulingMgrB class represents the scheduling for the Software Distribution Managment Service.

Attributes:

None

Operations:

None

Associations:

The PIRpResourceReservation class has associations with the following classes:

Class: ILMMgrB isusedby

5.11.3.10 SiteInventory Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This SiteInventory class is the internal COTS means of tracking all inventory status and actions taken in the inventory area at the individual site level.

Attributes:

None

Operations:

CoordinateTransfer - This CoordinateTransfer operation indicates that various actions to coordinate transfer of inventory items can be taken at site level.

Arguments:

Return Type:Void

Privilege:Public

UpdateCondition - This UpdateCondition operation indicates that various conditions of inventory items can be updated reflecting changes from site operations.

Arguments:

Return Type:Void

Privilege:Public

UpdateQuantity - This UpdateQuantity operation indicates that the quantity of various inventory items can be updated reflecting changes from site operations.

Arguments:

Return Type:Void

Privilege:Public

UpdateStatus - This UpdateStatus operation indicates that the status of various inventory items can be updated reflecting changes from site operations.

Arguments:

Return Type:Void

Privilege:Public

Associations:

The SiteInventory class has associations with the following classes:

Class: SystemWideInventory populates

ILMMgrB (Aggregation)

5.11.3.11 SiteLogistics Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This SiteLogistics class is the internal COTS means of tracking all logistics status and actions taken in the logistics area at the individual site level.

Attributes:

None

Operations:

GenerateShortageList - This GenerateShortageList operation indicates that a list of logistics shortages can be generated for site operations.

Arguments:

Return Type:Void

Privilege:Public

Associations:

The SiteLogistics class has associations with the following classes:

Class: SystemWideLogistics populates

ILMMgrB (Aggregation)

5.11.3.12 SiteMaintenance Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This SiteMaintenance Class is the internal COTS means of tracking all maintenance actions taken at the individual site level.

Attributes:

None

Operations:

EnterMaintData - This EnterMaintData operation indicates that maintenance data is entered at site for site maintenance operations.

Arguments:

Return Type:Void

Privilege:Public

RetrieveMaintData - This RetrieveMaintData operation indicates that maintenance data is retrieved at site-level for site maintenance operations.

Arguments:

Return Type:Void

Privilege:Public

UpdateMaintData - This UpdateMaintData operation indicates that updated maintenance data is entered at the site for site maintenance operations.

Arguments:

Return Type:Void

Privilege:Public

Associations:

The SiteMaintenance class has associations with the following classes:

Class: SystemWideMaintenance providessitemaintenancedata

ILMMgrB (Aggregation)

5.11.3.13 SystemWideInventory Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This SiteWideInventory class is the internal COTS means of tracking all inventory actions taken throughout the entire ECS by one, several, or all ECS sites.

Attributes:

None

Operations:

CalculateSummaries - This CalculateSummaries operation indicates that summaries for inventory resources can be calculated for the entire ECS - for one, several, or all ECS sites.

Arguments:

Return Type:Void

Privilege:Public

CollectInventory - This CollectInventory operation indicates that inventory information can be collected for resources throughout the entire ECS - for one, several, or all ECS sites.

Arguments:

Return Type:Void

Privilege:Public

LocateResource - This LocateResources operation indicates that actions can be taken to locate any ECS inventory resource throughout the entire ECS for for one, several, or all ECS sites.

Arguments:

Return Type:Void

Privilege:Public

Associations:

The SystemWideInventory class has associations with the following classes:

Class: SiteInventory populates

ILMMgrB (Aggregation)

5.11.3.14SystemWideLogistics Class

Parent Class:Not Applicable

Public:No

Distributed Object:No

Purpose and Description:

This SystemWideLogistics class is the internal COTS means of tracking all logistics actions taken throughout the entire ECS by one, several, or all ECS sites.

Attributes:

None

Operations:

CollectLogisticsData - This CollectLogisticsData operation indicates that logistics data can be collected for analysis and action for the entire ECS - for one, several, or all ECS sites.

Arguments:

Return Type:Void

Privilege:Public

ComputeSummaries - This ComputeSummaries operation indicates that various logistics summaries can be computed for status on the ECS for one, several, or all ECS sites.

Arguments:
Return Type:Void
Privilege:Public

Associations:

The SystemWideLogistics class has associations with the following classes:

Class: SiteLogistics populates
ILMMgrB (Aggregation)

5.11.3.15SystemWideMaintenance Class

Parent Class:Not Applicable
Public:No
Distributed Object:No
Purpose and Description:

This SystemWideMaintenance class is the internal COTS means of tracking all maintenance actions taken throughout the entire ECS by one, several, or all ECS sites.

Attributes:

None

Operations:

InputPMRequirement - This InputPMRequirement operation indicates that PM requirements data are entered for action throughout the ECS by one, several, or all ECS sites.

Arguments:
Return Type:Void
Privilege:Public

ObtainOffSite MaintStatus - This ObtainOffSiteMaintStatus operation indicates that off-site maintenance status for one, several, or all ECS locations.

Arguments:
Return Type:Void
Privilege:Public

ObtainOnSiteMaintInfo - This ObtainOnSiteMaintInfo operation indicates that on-site maintenance data can be obtained for one, several, or all ECS sites.

Arguments:
Return Type:Void
Privilege:Public

Associations:

The SystemWideMaintenance class has associations with the following classes:

Class: SiteMaintenance providessitemaintenancedata

ILMMgrB (Aggregation)

5.11.4 ILM Management Dynamic Model 1: Updating the Status of an ECS Asset

This scenario traces the actions associated with updating the status of an ECS asset. The scenario is depicted in Figure 5.11-3.

5.11.4.1 Beginning Assumptions

M&O Staff has removed an asset (a CM item) and wishes to update the inventory records.

5.11.4.2 Interfaces with Other Subsystems and Segments

None

5.11.4.3 Stimulus

M&O Staff requests information on a particular item for modification.

5.11.4.4 Participating Classes From the Object Model

MsMIItemB

MsMIILMMgrB

MsMIBaselineManager

ILMLog

5.11.4.5 Beginning System, Segment and Subsystem State(s)

Normal Operations.

5.11.4.6 Ending State

The selected item reflects the status change in the inventory records.

5.11.3.7 Scenario Description

- MsMIILMMgrB requests the current inventory number from MsMIILMItemB.
- ILMLog records the request.
- MsMIILMItemB provides the item information.
- MsMIILMMgrB requests change of item information.
- MsMIILMMgrB sends information request to MsMIBaselineManager.
- ILMLog records the request.
- MsMIBaselineManager sends checked out copy to MsMIILMMgrB.
- MsMIILMMgrB modifies the information and submits to MsMIItemB.
- MsMIILMMgrB checks in changed copy to MsMIBaselineManager.

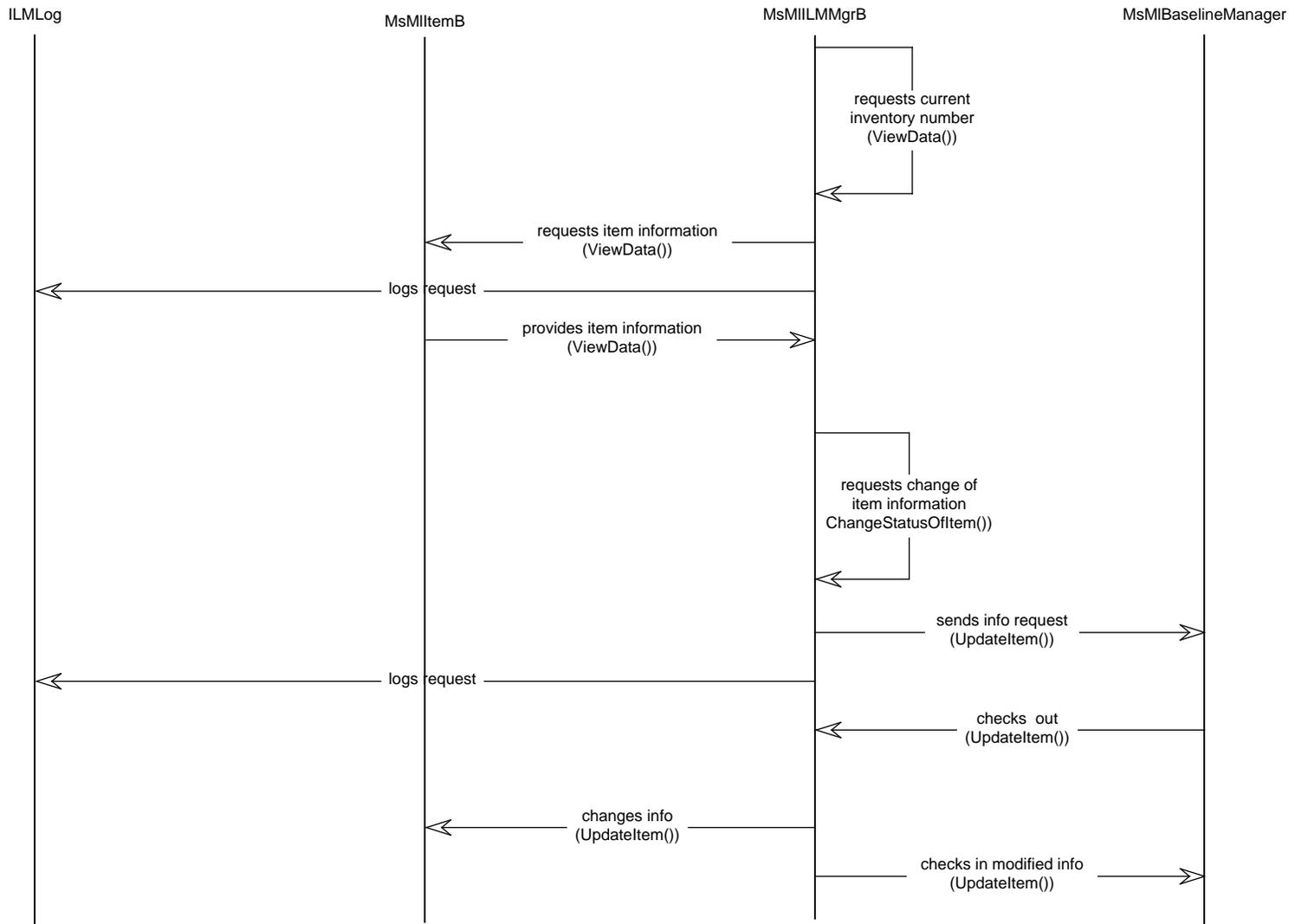


Figure 5.11-3. ILM (Inventory/Logistics) Scenario: Updating the Status of an ECS Asset

5.11.5 ILM Manager Dynamic Model 2: Perform Maintenance

This scenario traces the actions associated with an M&O Staff member performing a (corrective) maintenance action. The scenario is depicted in Figure 5.11-4.

5.11.5.1 Beginning Assumptions

A trouble ticket has been created for the (corrective) maintenance action.

5.11.5.2 Interfaces with Other Subsystems and Segments

None

5.11.5.3 Stimulus

The trouble ticket service notifies the ILM Manager of a (corrective) maintenance action to be performed.

5.11.5.4 Participating Classes From the Object Model

MsTtEntry

MsMILogB

MsMIILMMgrB

ILMReportGenerator

5.11.5.5 Beginning System, Segment and Subsystem State(s)

Normal Operations.

5.11.5.6 Ending State

The trouble ticket has been closed, and the spares inventory updated to reflect the removal of an item used to perform the (corrective) maintenance.

5.11.5.7 Scenario Description

- MsTtEntry provides a trouble ticket to MsMIILMMgrB (via E-mail).
- MsMIILMMgrB assigns the maintenance action.
- MsMIILMMgrB logs assigned action with MsMILogB.
- MsMIILMMgrB requests and receives inventory/logistics information to perform assigned action.
- MsTtManager requests status on the maintenance action from MsMIILMMgrB (via E-mail).
- MsMIILMMgrB queries about and gets maintenance status.
- MsMIMILMMgrB sends maintenance status (via E-mail) to MsTtEntry.
- MsMIILMMgrB updates inventory/logistics information.
- MsMIILMMgrB receives report of completed maintenance.
- MsMIILMMgrB sends status of completed maintenance to MsTtEntry (via E-mail).

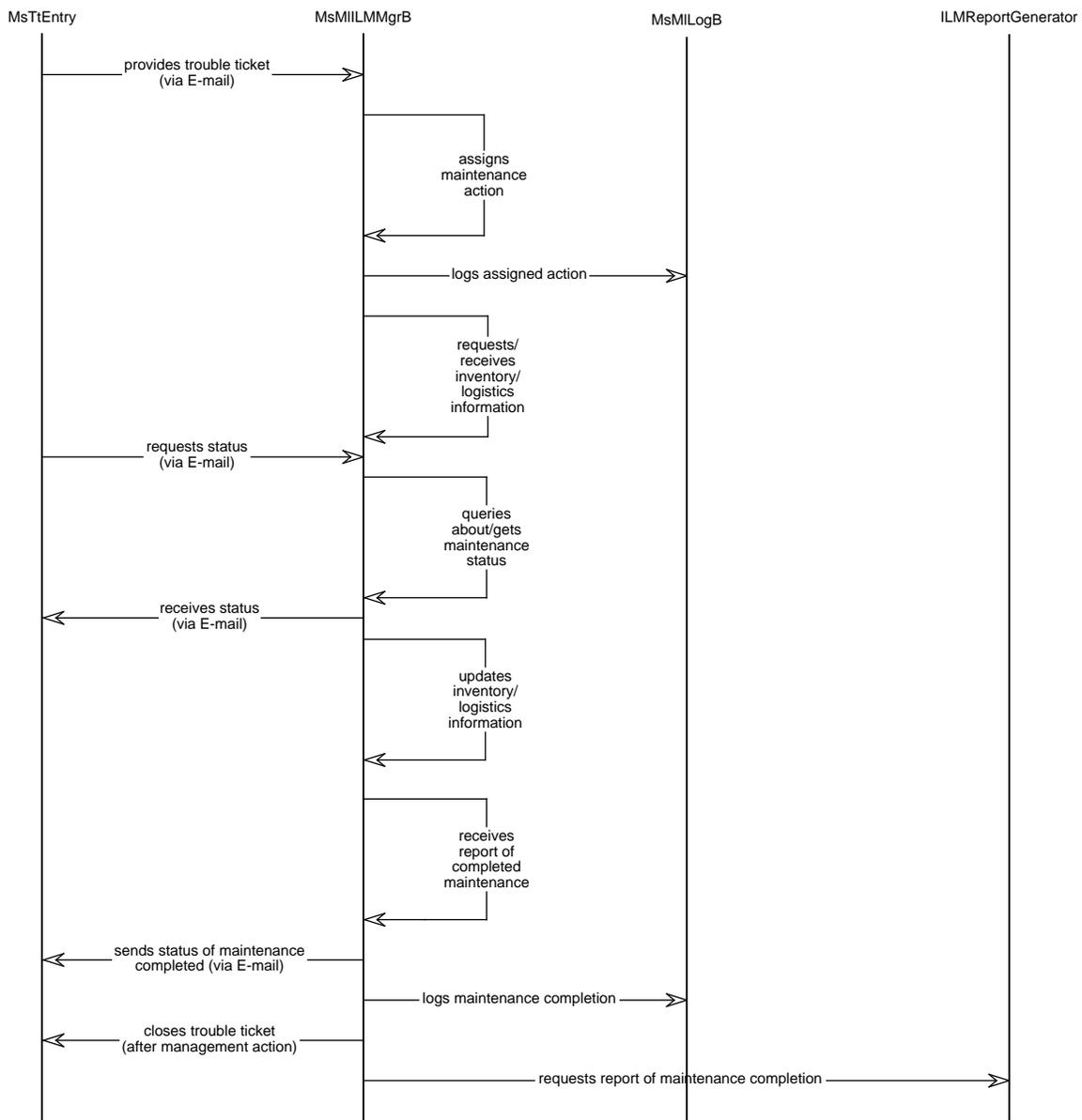


Figure 5.11-4. ILM Manager Scenario: Performing Maintenance

- MsMIILMMgrB logs maintenance completion with MsMIILMLogB.
- MsMIILMMgrB closes trouble ticket with MsTtEntry (after management action).
- MsMIILMMgrB requests report of completed maintenance from ILMReportGenerator.

5.11.6 Inventory Management Structure

Table 5.11.6-1 lists the components of the Inventory Management Service.

Table 5.11.6-1 Inventory Manager Components

Object Class	Implementation
MsMILMMgrB	COTS
SiteInventory	COTS
SystemWideInventory	COTS
SiteLogistics	COTS
SystemWideLogistics	COTS
SiteMaintenance	COTS
SystemWideMaintenance	COTS
ILMLog	COTS
ILMReportGenerator	COTS
MsMItemB	COTS

This page intentionally left blank.