FS – MAS Operate and Maintain Manager

Content

History

Version	Date	Adjustments
A	2006-10-16	First version (MHATU).
B	2006-12-05	Second version (MHATU)
C	2007-06-15	Minor changes. Removed check OMManager used to do to check process memory (ermkese).

1 Introduction

This document specifies the functionality of the Operate and Maintain Manager component (OMM) in a MAS. The design of OMM is generic even though parts are very specific to MAS.

2 Definitions

2.1 External Service Consumer

An entity in a system that utilize a service provided by the Application running on MAS.

2.2 External Service Provider

An entity in a system that the Application running on MAS utilizes.

2.3 Service

Service in this document refers to a service as registered in the Component Registry (MCR).

2.4 Service Consumer

An entity in MAS that consumes a Service.

2.5 Service Enabler

An entity in MAS that enables a Service to an External Service Consumer.

2.6 Service Provider

An entity in MAS that provides a Service.

3 Function Requirements (Commercial)

The following commercial requirements have been identified.

- The system shall run on Solaris 10.
- The system shall be easy to install and maintain
- The system shall be easy to trouble shoot

4 Function Specification (Design Related)

4.1 Introduction

The OMM handles the following basic functions:

- Installation/uninstallation/upgrade of software and patches
- Operation (start/restart, lock/unlock, configuration change etc.)
- Load regulation
- Provide supervision information on **provided** services through MIB. It is the responsibility of the Application to define what services it provides, i.e. the MAS don’t provide any services by itself.
- Provide supervision information on Service Enablers used for MAS provided services.
- Provide a monitoring function of Service Enabler statistics and connection information details
- Watchdog
- Backup/Restore

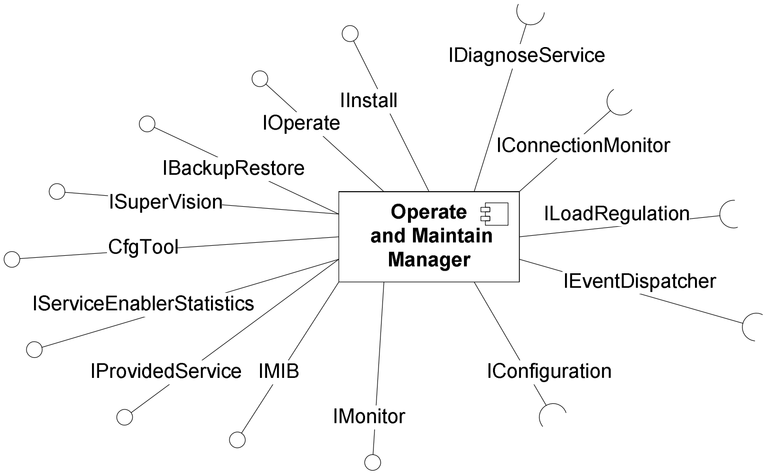

Figure 1 The Operate and Maintain component

The OMM exports the following interfaces. *IInstall* interface for installation, uninstallation and upgrade procedures, *IOperate* for operation of the provided functionality, *IBackupRestore* for backup and restore, *ISupervision*, *IProvidedService, CfgTool* and *IServiceEnablerStatistics* for providing supervision information, *IMIB* for providing supervision and other information to external systems and finally *IMonitor* for monitoring of active sessions.

The IOperate, IMIB interfaces executes separately from the MAS traffic handling process because the OMM must be able to report problems in the traffic part of MAS even if it has severe problem or even crashed.

4.2 Exported interfaces

4.2.1 CfgTool

This tool is used to manipulate the XML configuration files.

This is a manual interface i.e. the interface is used by a human being by using this tool to manipulate the configuration files.

Syntax:

```
CfgTool [-g param [-q][-v] <input file>]
        [-s param=value [-v] <input file> [output file]]
        [-m [-v] <old file> <new file> [output file]]
        [-l <input file>]
        [-h]
```

Options:

-g param - Print the value of one or more parameters.

-q "Quiet mode", only print the value.

-s param=value - Set one or more parameters to the specified value. The result is stored in <output file> if it has been specified. If not <input file> is overwritten.

-m Merge <old file> into <new file>. The result is store in <output file> If it has been specified. If not <input file> is overwritten.

-v Validate the XML file against its schema.

-l List all parameters in the configuation file.

-h Print this message.

The -g, -s, -m, and -l parameters are mutually exclusive and can't be used simultaneously.

4.2.2 IInstall

This interface is used by an operator when the MAS component shall be installed, uninstalled.

This is a manual interface i.e. the interface is used by a human being by sending commands to the OMM as specified in ref. [1] and ref. [4]

4.2.3 IOperate

This interface is used by an operator when the MAS component shall be operated for example started, stopped etc.

This is a manual interface i.e. the interface is used by a human being by sending commands to the OMM as specified in ref. [3] .

4.2.4 IBackupRestore

This interface is used by an operator when the MAS component shall be backup and restored.

This is a manual interface i.e. the interface is used by a human being by sending commands to the OMM as specified in ref. [2]

4.2.5 ISupervision

This interface is used to create entries in OMM to be used by a client to provide supervision information.

4.2.5.1 Create Provided Service Entry

IProvidedService createProvidedServiceEntry(Service s, Host h, port p, ServiceEnabler se);
This method returns a reference to a service entry for this Provided Service. If this is the first time since install time this request is done, OMM creates a new service entry otherwise it returns an existing one.
For each ServiceName there can be only one entry in O&M.

The entry will continuously be updated by the client to reflect the Provided Service actual status.

4.2.5.2 Get service enabler statistics

ServiceEnablerInfo getServiceEnablerStatistics(ServiceEnabler);

This method returns a reference to a statistics entry for this enabler.

The entry will continuously be updated by the client to reflect the Service Enabler statistics.

4.2.5.3 Get session info factory

SessionInfoFactory getSessionInfoFactory();

This method returns a reference to a factory.

The factory will be used when execution engine handles call sessions.
Execution engine will ask the factory for a session object that continuously will be updated.

4.2.5.4 Parameter type description

4.2.5.4.1 Service

Service contains the service as defined in MCR.

4.2.5.4.2 Host

Host contains the name of the host including the domain that provides the service.

4.2.5.4.3 Port

Port contains the port on the host that provides the service.

4.2.5.4.4 Service Enabler

Service Enabler contains the name of the service enabler.

4.2.5.4.5 Connection Monitor

A Connection Monitor is a reference to a part of MAS that provide Connection information.

4.2.6 IProvidedService

This interface is used to provide supervision information on provided services to OMM. The interface offers the following methods. Even if the Provided Service is unable to use (corrupted, invalid etc.), OMM expects the entry to be created anyway. Its state is then initially set to down. The purpose of this is that OMM needs to communicate this problem in the MIB.

4.2.6.1 Set Status

void setStatus(Status s);

Set’s the status of the provided service.

4.2.6.2 Set Application

void setApplication(ApplicationName an, ApplicationVersion av);

Set’s the Application information. This information will be used to provide the Application version in the MIB.

4.2.6.3 Set Current time

void setCurrentTime();

Set’s the current time. This is a kind of heartbeat that OMM can use to check that the Service Provider is alive. This time is updated each second.

4.2.6.4 Parameter type description

4.2.6.4.1 Status

Contain the status of the provided service. Can be one of:

Up	The service is fully operational
Down	The service is completely non-operational

4.2.7 ServiceEnablerInfo

This interface is used to provide statistical information on a Service Enabler and offers the following methods.

4.2.7.1 Set protocol

void setProtocol(Protocol p);

Set's the protocol used by this Service Enabler.

4.2.7.2 Set Max connections

void setMaxConnections(int mc);

Set's the max allowed number of connections (i.e. the value currently used by the Service Enabler). This value is updated whenever changed in the client (e.g. due to values provided in the load regulation interface).

Increment current connections

void incrementCurrentConnections(Type t, Direction d);

Increments the number of connections currently active in the Service Enabler by one for the specific type and direction.

4.2.7.3 Decrement current connections

void decrementCurrentConnections(Type t, Direction d);

Decrements the number of connections currently active in the Service Enabler by one for the specific type and direction.

4.2.7.4 Increment number of connections

void incrementNumberOfConnections(Type t, Result r, Direction d);

Increments number of started connections in the Service Enabler by one for the specific type, result and direction.

4.2.7.5 Parameter type descriptions

4.2.7.5.1 Protocol

Protocol contains the literal name of the protocol used e.g. "sip" or "xmp".

4.2.7.5.2 Type

Type contains the type of connection. Can be one of Voice, Video or Service Request.

4.2.7.5.3 Result

Variant contains the result of a connection. Can be one of:

Connected	The connection has reached connected state
Failed	The connection was rejected.
Abandoned	The connection was abandoned.
Far End Disconnected	The connections that were dropped by the far end
Near End Disconnected	The connections that were dropped by the near end
Dropped entities	Dropped entities for this connection. For example RTP packets.
Error	Connection was dropped due to failure.

4.2.7.5.4 Direction

Direction contains the direction of the connection. Can be one of Out or In.

4.2.8 IMIB

The IMIB interface allows an SNMP client to access the following Management Information Base (MIB) using the SNMP protocol as defined in ref. [7] .

The top OID for the MIB is:

- `enterprises.24261`

4.2.8.1.1 General Attributes

The following general information is supported.

Type	Description
Name	The name of the MAS component as registered in MCR i.e. <component_instancename>@<hostname> e.g. mas1@host1.com
Version	The version of the MAS component as registered in MCR.
Operational State	The operational state of MAS. One of Enabled or Disabled.
Administrative State	The Administrative state of MAS. One of Locked, Unlocked or Shutdown.
Install Date	The date the MAS was installed.
Current Up-Time	The time MAS has been up since last start of MAS
Accumulated Up-Time	The total up time of MAS since install date.
Reload configuration	A writeable attribute that instructs MAS to reload its configuration

Reload configuration time	The time in hundredths of seconds since the last successful reload configuration was performed
Reload configuration ticks	This variable shows the time in number of hundredths of a second since the last time the configuration was loaded

Table 1 General attributes

4.2.8.1.2 Provided Services

For each provided service the following information is supported as defined in ref. [6] (except the Application information):

Type	Description
Name	The name of the service as defined in MCR
Status	The status of the service as defined in section 4.2.6.4.1.
Host name	The host providing the service (i.e. the host MAS executes on)
Port	The port on the host providing the service
Zone	The logical zone of the Service Provider i.e. this MAS instance.
Application name	The name of the Application that provides the service. This is defined within each Application
Application version	The R-state of the Application that provides the service.

Table 2 Provided Services in the MIB

4.2.8.1.3 Service Enabler Statistics

For each Service Enabler the following information is supported.

Type	Description
Protocol	The protocol used by this Service Enabler i.e. a literal name with the name of the protocol e.g. “sip”, “xmp” etc.
MaxConnections	Max number allowed connections for this Service Enabler.
Connections	An entry for each type with information as defined in Table 4.

Table 3 Service Enabler Statistics in the MIB

For each connection type the following information is supported.

Type	Description
Connection type	The type of this connection, one of Voice, Video or Service Request

Current connections	Current active connections
Peak Connections	Peak connections since restart
Peak Time	The time when Peak Connections occurred. If no connections have been active since restart this time will indicate start time.
Total Connections	The number of connections since last restart
Accumulated Connections	The number of connections since install

Table 4 Connection details

4.2.9 IMonitor

This interface is used by an operator when the active connections or connection statistics are to be monitored.

The interface must be able to provide the information in a textual form i.e. possible to view over a telnet connection.

This is a manual interface i.e. the interface is used by a human being by sending commands to the OMM with the following syntax:

```
Rc.mas monitor
```

Monitor has two views, *connections* and *statistics*. When *connections* is used, O&M retrieves details on each connection as specified in section 4.3.3.

When *statistics* is used, O&M retrieves the following statistics for each Service Enabler and presented per connection type (as defined in section 4.2.7.5.2).

Type	Description
Current connections	Number of current connections that are at least in the initiated state and until they are terminated.
Current inbound connections	Out of Current connections the number of inbound connections.
Current outbound connections	Out of Current connections the number of outbound connections.
Total connections	Number of connections since the last restart (with the same definition as Current connections)
Total inbound connections	Out of Total connections the number of inbound connections.
Total outbound connections	Out of Total connections the number of outbound connections
Total connected connections	Out of Total connections the number of connections that have been answered since the last restart.

Total connected inbound connections	Out of Total answered connections the number of inbound connections
Total connected outbound connections	Out of Total answered connections the number of outbound connections
Total abandoned connections	Number of abandoned connections since restart. The definition of an abandoned connection is done by the Service Enabler.
Total abandoned inbound connections	Out of Total abandoned connections the number of inbound connections.
Total abandoned outbound connections	Out of Total abandoned connections the number of outbound connections.
Total failed connections	Numbers of connections since restart that where rejected by the near or far end for any reason.
Total failed inbound connections	Out of Total rejected connections the number of inbound connections.
Total failed outbound connections	Out of Total rejected connections the number of outbound connections.
Total disconnected connections	Number of connections since restart that were terminated by the near end.

Table 5 Connection statistics

4.3 Imported interfaces

4.3.1 IEventDispatcher

Used to handle events.

4.3.2 IConfiguration

Used to retrieve configuration as defined in ref.[9]

4.3.3 IConnectionMonitor

This interface is used by the OMM to retrieve connection related information. The interface implements the following methods.

4.3.3.1 Start monitor

void startMonitor();

StartMonitor instructs the client to start collecting information on the connection.

4.3.3.2 Stop monitor

void stopMonitor();

StopMonitor instructs the client to stop collecting information on the connection.

4.3.3.3 Get Connection Info

IConnectionInfo[] getConnectionInfo();

GetConnectionInfo retrieves the information about all current connection.

4.3.3.4 Parameter type description

4.3.3.4.1 IConnectionInfo

ConnectionInfo contains the following information:

Type	Description
Session Id	The identity of the session using this connection. This information can be used to see which connections are used for a particular session.
Service	The Service using this connection
Session initiator	The initiator of the session that created this connection. One of: ”sr:<xmpclientid>” for Service Requests ”sid:<session id>” when initiated by another session ”tel:” when initiated by a telephony session.
Connection type	One of Voice or Video
Connection State	The state of the connections as defined in ref. [8]
Direction	One of Inbound or Outbound
Outbound activity	One of Idle or Play.
Inbound activity	One of Idle or Record.
ANI	The telephone number of the calling party
DNIS	The telephone number of the called party
RDNIS	The telephone number of the redirected called party
Far end connection properties	A list of the far end IP address and ports (for the outbound traffic) used for this connection and protocol. E.g. “RTP:123.4.56.127:2080:2082, SIP:123.4.56.24:5060”

Table 6 Connection info

4.3.4 ServiceEnablerOperate

This interface is used by the OMM to operate and set the load threshold for Service Enablers. The interface implements the following methods:

4.3.4.1 Open

void open();

Unlock the Service Enabler and allows it to accept traffic according to the value earlier set by the setThreshold method.

4.3.4.2 Close

void close(forced);

When close(true) is called (Forced):

Lock the Service Enabler immediately. All ongoing sessions are interrupted and terminated (with respect to the semantics in the protocol used to access the service). All new service requests are rejected. When all service sessions has terminated the client calls the closed on the ServiceEnablerInfo interface.

When close(false) is called (Unforced):

Close the Service Enabler unforced, which means that all ongoing service sessions continue until normally ends, but all new service requests are rejected. When all service sessions has terminated the client calls the closed on the ServiceEnablerInfo interface.

4.3.4.3 Set Threshold

void setThreshold(Threshold th);

Update the threshold for the Service Enabler. If the client’s current sessions are greater than threshold, it will act to prohibit that new sessions are started until current sessions is equal or less than the threshold.

4.3.5 IDiagnoseService

This interface is used by OMM to check if a service offered by a Service Enabler is functional. The interface implements the following methods:

4.3.5.1 Service Request

Status Boolean serviceRequest(ServiceInstance si);

Check if the service enabled by the Service Enabler is functional. Return the service status.

Note: These Service Requests shall not in any way affect any counters or thresholds that are used for normal Service Requests.

4.3.5.2 Parameter type description

4.3.5.2.1 Status

Status can be one of

Up	The Service Enabler was able to fully perform the Service Request

Down	The Service Enabler completely failed to perform the Service Request

4.4 Function Specification

4.4.1 Installation

The installation of MAS is as specified in ref. [1] . The installation procedure register the MAS with the mas1@<domain> Service.

4.4.2 Operation

4.4.2.1 Managing Administrative state

The Administrative state only concerns the Provided Services.

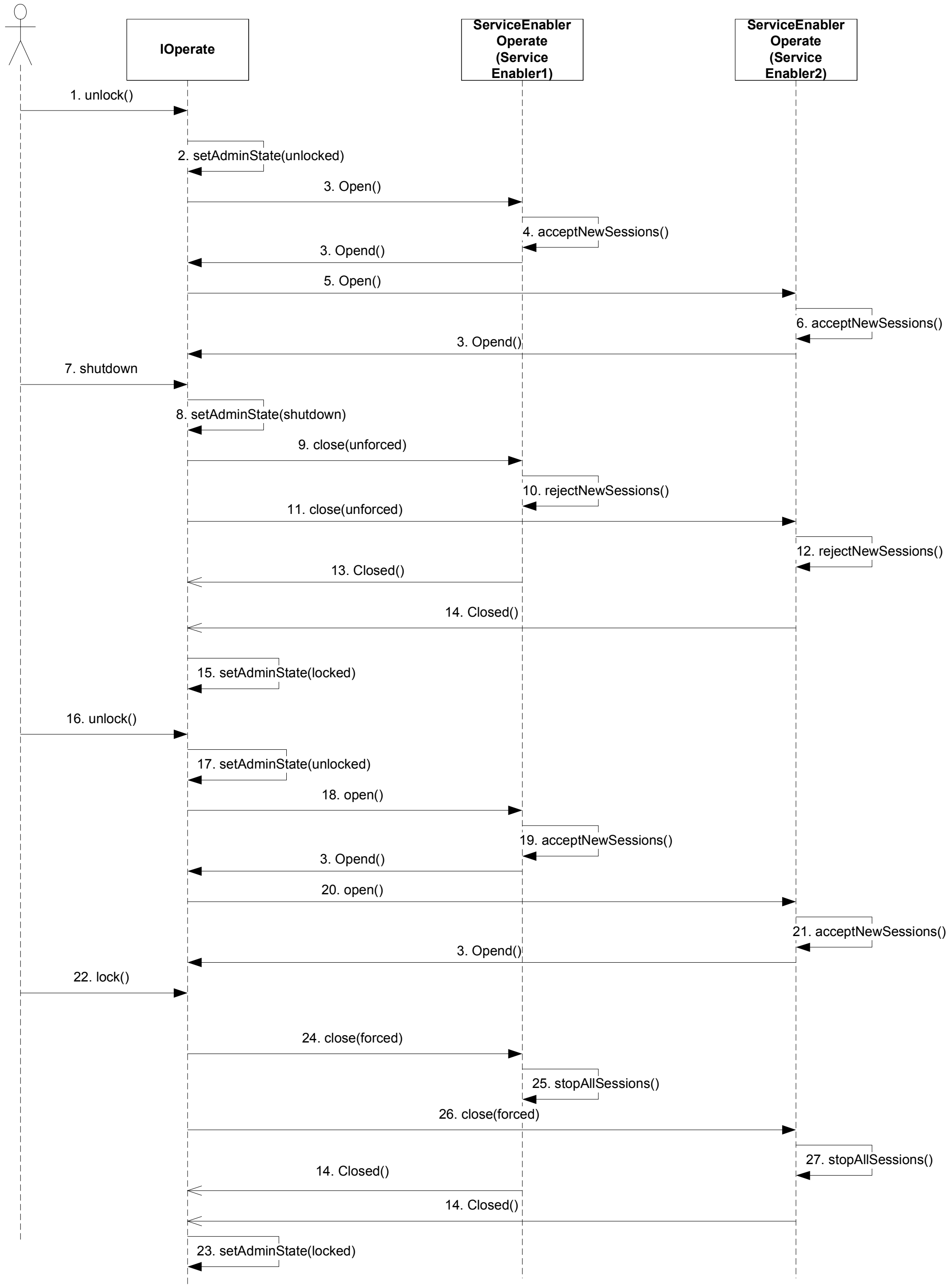

Figure 2 Operating Administrative state

1. The operator performs an unlock request on MAS.
2. OMM sets the Administrative State to unlocked.
3. OMM check the Provided Service entries and for each Service Enabler found issues an open() request.
4. The Service Enabler 1 starts accepting new session (if the current threshold value is > 0, see section 4.4.3 how threshold is set).
5. Similar …
6. … as for Service Enabler 1.
7. The operator performs a shutdown request on MAS
8. OMM sets the Administrative State to Shutdown.
9. OMM issues a close(unforced) request for each Service Enabler.
10. Service Enabler 1 stops accepting new sessions.
11. OMM issues a close(unforced) request for the next Service Enabler
12. Service Enabler 2 stops accepting new sessions.
13. When all sessions has finished it calls closed() on OMM.
14. Similar as for Service Enabler 1.
15. When all Service Enabler is closed, the OMM sets the Administrative state to Locked.
16. The operator performs an unlock request on MAS.
17. OMM sets the Administrative State to unlocked.
18. OMM issues an open() request for each Service Enabler.
19. Service Enabler 1 starts accepting new session.
20. Similar …
21. … as for Service Enabler 1.
22. The operator performs a lock request on MAS
23. OMM sets the Administrative state to locked.
24. OMM issues a close(forced) request for each Service Enabler.
25. Service Enabler 1 stop all current sessions
26. Similar …
27. … as for Service Enabler 1.

4.4.2.2 Operational state

The operational state of MAS is set by OMM to “disabled” if one of the following statements is true otherwise the operational state is “enabled”:

1. If one of the Provided Services is in state “down” either if itself reports “down” or if the service diagnose shows that the service is “down”.

2. No Service Provider has created a Provided Service Entry.

3. If the load criterion for one of the Provided Services results in a threshold value equals to zero.

4. The communication with the ILoadRegulation interface is broken (e.g. not responding etc.).

At any time when the operational state goes from enabled to disabled, an info log entry is created describing the cause why it happened (i.e. one of the reasons above). When the state then later on goes from disabled to enabled a info log tells that the MAS is now operational.

4.4.2.3 *Dynamic load of configuration*

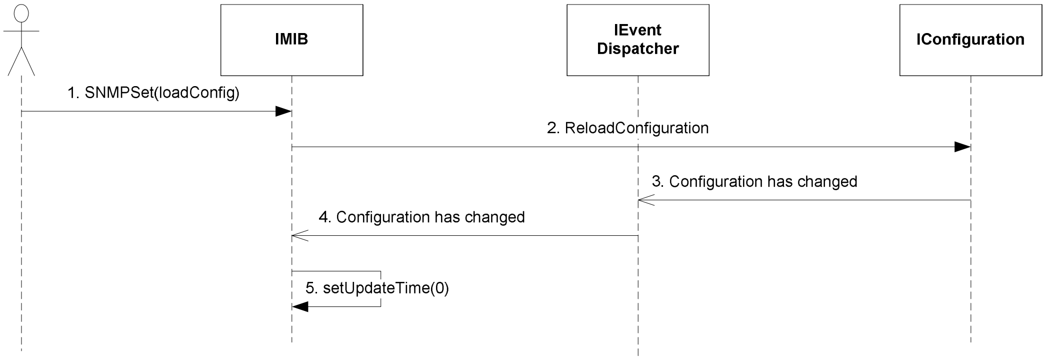

Figure 3 Reload configuration

1. The operator, using an SNMP request issues a load configuration request
2. OMM request a configuration load
3. IConfiguration sends the Configuration has changed event
4. OMM receives the Configuration has changed event.
5. OMM sets the update time to zero. This time then starts to increase each hundredths of a second.

4.4.3 Load regulation

OMM handles load regulation on Service Enablers. The Load regulation loads the threshold as soon as the service enabler is registered. The load regulation determines the status for each registered provided service and update the operational status for MAS

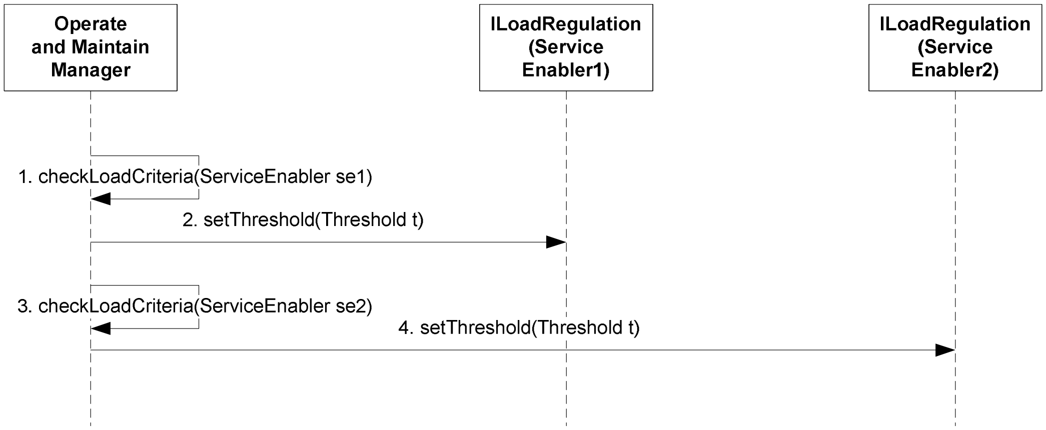
Operate and Maintain Manager
ILoadRegulation (Service Enabler1)
ILoadRegulation (Service Enabler2)
1. checkLoadCriteria(ServiceEnabler se1)
2. setThreshold(Threshold t)
3. checkLoadCriteria(ServiceEnabler se2)
4. setThreshold(Threshold t)

Figure 4 Load regulation

1. For each Service Enabler OMM checks load criteria. Load criterion is currently based on configuration only.
2. The OMM sets the threshold value Service Enabler 1
3. Similar …

4. … as for Service Enabler 1.

4.4.4 Supervision

Supervision conforms to the directions specified in [5] as defined in this section.

4.4.4.1 Provided Services

OMM maintains a list of Provided Services created by the Service Provider. The Service Provider updates the status.

For each Provided Service the OMM monitors the function of the service by performing tests at a regular interval. This test is also part of the installation procedure i.e. if the test fails the installer will be notified. How often this shall occur is configurable and defaults to once each 60 seconds.

An SNMP client can retrieve certain information from the Provided Services list based on the MIB definition (see section 4.2.7.).

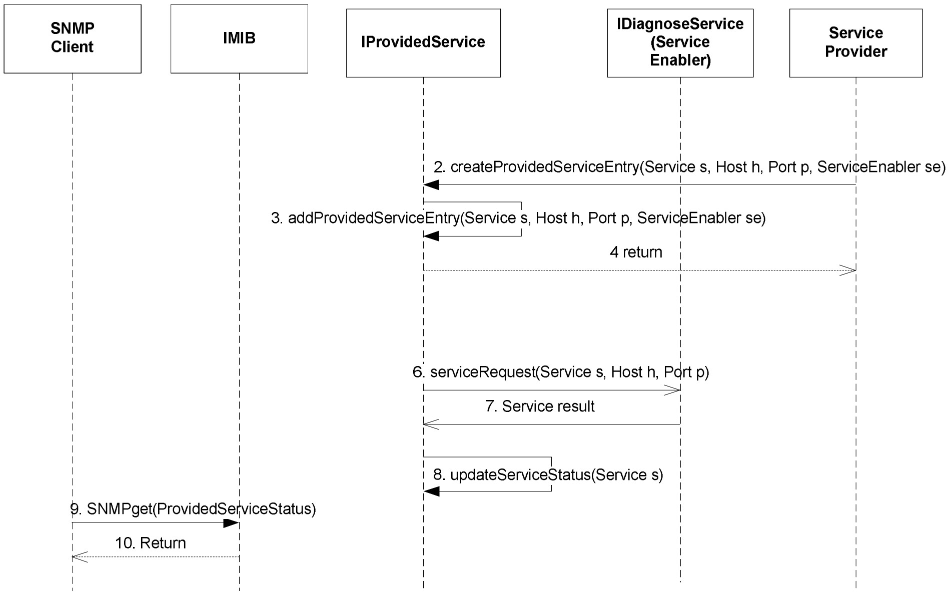
SNMP Client
IMIB
IProvidedService
IDiagnoseService (Service Enabler)
Service Provider
2. createProvidedServiceEntry(Service s, Host h, Port p, ServiceEnabler se)
3. addProvidedServiceEntry(Service s, Host h, Port p, ServiceEnabler se)
4 return
6. serviceRequest(Service s, Host h, Port p)
7. Service result
8. updateServiceStatus(Service s)
9. SNMPget(ProvidedServiceStatus)
10. Return

Figure 5 Provided Service supervision

1. The Service Provider initiates the service Enabler and receives a ServiceEnabler.
2. The Service Provider requests that a Provided Service entry is created in the OMM.
3. OMM creates the entry using that information (if it doesn’t exist otherwise the existing entry is used) and …
4. … returns a reference to the entry to the Service Provider
5. The Service Provider updates service status on the reference.

6. The OMM at a regular interval checks the Provided Service list and for each entry performs a Service Request to the Service Enabler passing the host and port in the request. This Service Request is a normal such. The request is asynchronous because some service may take some time to perform.
7. OMM receives the result after some time and …
8. … if the service request succeeds, the status of Provided Service is left unchanged otherwise the entry is updated, i.e. based on “worst case” scenario, only if both the Service Provider and OMM test shows that the service is up, the resulting state will be up.
9. The SNMP client retrieves the information from the Provided Service list as defined in the MIB and …
10. … returns.

4.4.4.2 Service Enabler

OMM maintains a list of Service Enablers. There is only one entry for each Service Enabler.

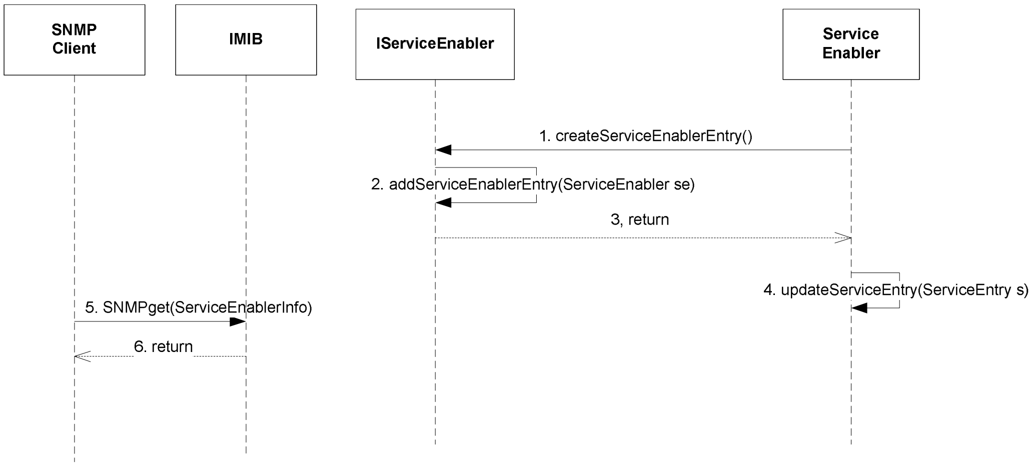
SNMP Client
IMIB
IServiceEnabler
Service Enabler
1. createServiceEnablerEntry()
2. addServiceEnablerEntry(ServiceEnabler se)
3, return
4. updateServiceEntry(ServiceEntry s)
5. SNMPget(ServiceEnablerInfo)
6. return

Figure 6 Service Enabler supervision

1. The Service Enabler requests that Service Enabler entry is created in the OMM.
2. OMM creates the entry (if it doesn’t exist otherwise the existing one is used) and …
3. … returns a reference to the entry to the Service Enabler
4. The Service Enabler continuously updates Service Enabler entry on the reference.
5. The SNMP client retrieves the information from the Service Enabler list as defined in the MIB.
6. And returns the information to the SNMP Client

4.4.5 Connection Monitoring

The OMM provides a monitoring tool that displays information on current connection.

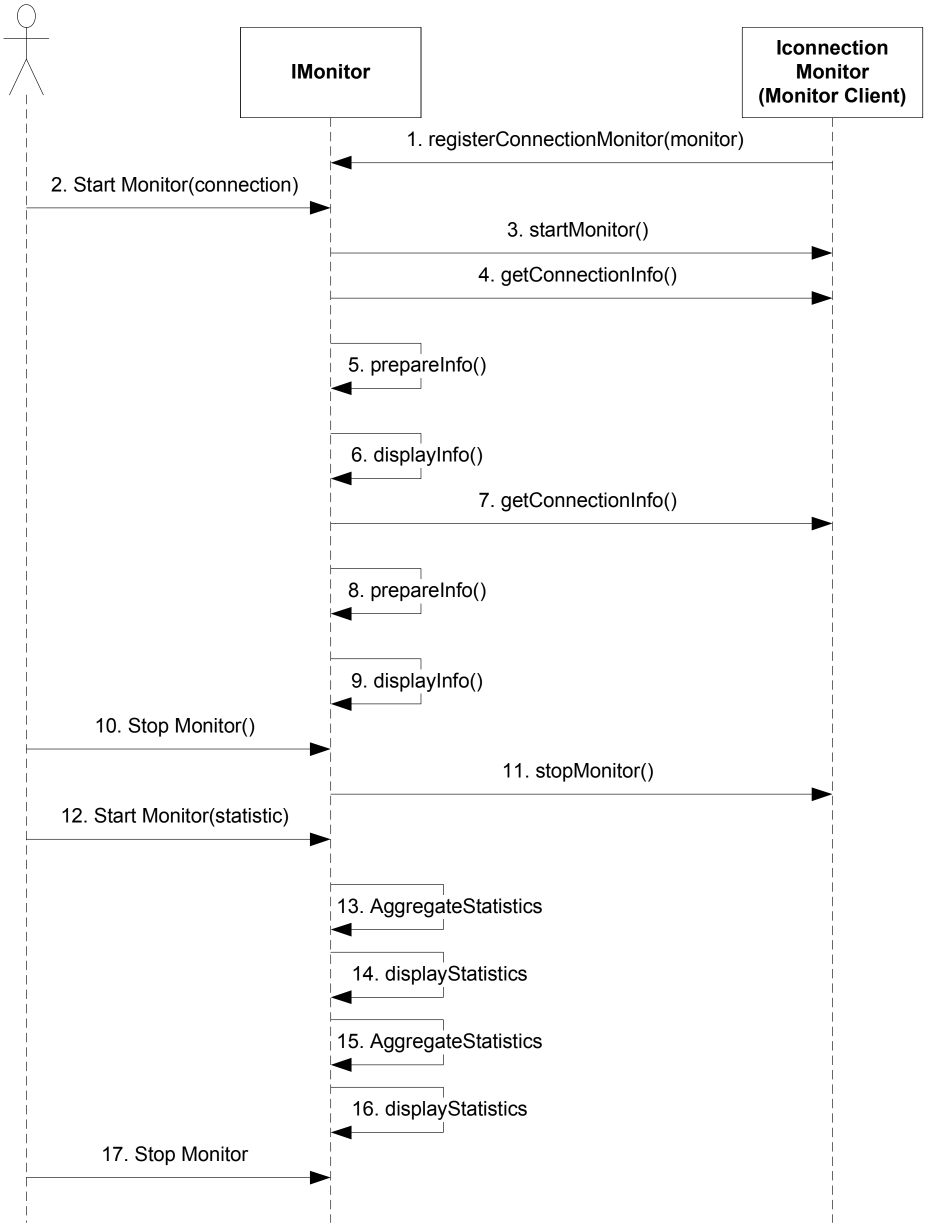

Figure 7 Connection monitoring

1. The Monitor Client at startup of MAS registers itself to OMM as a Connection Monitor
2. The operator starts the monitor to show connection details.
3. The OMM requests a monitor client to start a monitor.
4. The OMM requests connection information regularly …

5. … prepares the information and …
6. … displays the information to the operator.
7. … and…
8. … once …
9. … again
10. The operator stops the monitor and …
11. … OMM requests the Monitor client to be stop its monitor
12. The operator starts the monitor to show connection statistics.
13. OMM aggregates Service Enabler information
14. OMM displays the statistical information
15. … and …
16. … once again
17. The operator stops the monitor

4.4.6 Watchdog

The OMM supervises the processes involved in MAS and when a process unexpectantly for some reason terminates, OMM restarts this process. The watchdog starts its supervision as soon as the MAS component is started.

4.5 Events

4.5.1 Produced events

OMM produce no events.

4.5.2 Consumed events

The OMM consumes the following events

Service result	Contains the result of the service request, one of "up", "down" or "impaired".
Shutdown completed	The Service Enabler has completed the shutdown request and got into locked state.
Configuration Change	The operator has issued a reload of the configuration.

5 External Operation Conditions

5.1 Configuration

The following properties of OMM are possible to configure:

1. The interval in which the self monitoring occurs, defaults to 60 seconds.
2. The interval the Monitor poll is performed, defaults to 1 second.
3. The interval in which different counters are stored persistent.

6 Capabilities

This paragraph was intentionally left blank.

7 References

[1] UCD Install Component
12/155 53-1HDB 101 02 Uen

[2] UCD Backup and Restore Component
13/155 53-1HDB 101 02 Uen

[3] IWD Component Start & Restart
17/155 19-1HDB 101 02 Uen

[4] Patch handling
2/000 21-FCP 101 1154 Uen

[5] FS Supervision
15/155 17-1HDB 101 02 Uen

[6] IWD-Managed Object Interface
2/155 19 CRH 109 083 Uen

[7] RFC 3416 Operations for SNMP
http://www.ietf.org/rfc/rfc3416.txt?number=3416

[8] CCXML
www.w3c.org

[9] FS-Configuration
16/FS-MAS0001 Uen

[10] In Service Performance
3/1551-2/HDB 101 02 Uen

[11] Application Packaging Developers Guide
ftp://docs-pdf.sun.com/817-0406/817-0406.pdf

8 Terminology

MAS	Media Access Server
MIB	Management Information Base
OMM	Operate and Maintain Manager
SNMP	Simple Network Management Protocol