Management Policy MO Technical Specification

Candidate Version 1.0 – 01 Apr 2014

Open Mobile Alliance

OMA-TS-MgmtPolicyMO-V1_0-20140401-C

Contents

Figures

Tables

No table of figures entries found.

1. Scope

This document defines the technical specification for Management Policy Management Object, which is based on the requirements and the use cases included in [MgmtPolicyMO-RD] and the architecture described in [MgmtPolicyMO-AD].

2. References

2.1 Normative References

[3GPP-TS_23.032] 3GPP TS 23.032: Technical Specification Group Services and System Aspects; Universal Geographical Area Description (GAD)

[DMDICT] "OMA Device Management Dictionary", Version 1.0, Open Mobile Alliance™, OMA-SUP-DM_Dictionary-V1_0, URL:http://www.openmobilealliance.org/

[DMPRO] "OMA Device Management Protocol, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_Protocol-V1_3. URL:http://www.openmobilealliance.org/

[DMTND] "OMA Device Management Tree and Description, Version 1.3". Open Mobile Alliance™. OMA-TS-DM_TND-V1_3. URL:http://www.openmobilealliance.org/

[ISO 8601] ISO 8601 Data elements and interchange formats – Information interchange – Representation of dates and times

[MgmtPolicyMO-AD] MgmtPolicyMO Architecture, Open Mobile Alliance™, OMA-AD-MgmtPolicyMO- V1_0, URL:http://www.openmobilealliance.org/

[MgmtPolicyMO-RD] MgmtPolicyMO Requirements, Open Mobile Alliance™, OMA-RD-MgmtPolicyMO- V1_0, URL:http://www.openmobilealliance.org/

[SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

2.2 Informative References

[DCMO-TS] "Device Capability Management Object Version 1.0", OMA-TS-DCMO-V1_0, URL:http://www.openmobilealliance.org/

[DiagMonFunc-TS] "Diagnostics and Monitoring Functions Specification Version 1.2", OMA-TS-DiagMon_Functions-V1_2, URL:http://www.openmobilealliance.org/

[DMStdObj-TS] "OMA Device Management Standardized Objects Version 1.3", OMA-TS-DM_StdObj-V1_3, URL:http://www.openmobilealliance.org/

[OMADICT] "Dictionary for OMA Specifications", Version 2.9, Open Mobile Alliance™, OMA-ORG-Dictionary-V2.9, URL:http://www.openmobilealliance.org/

[SCOMO-TS] "Software Component Management Object Version 1.0", OMA-TS-SCOMO-V1_0, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Action	When stated alone with no qualifiers (e.g. simple action, Negative Action), refers to a Compound Action triggered on the positive result transition of the Compound Condition, from false to true.
Compound Action	Combination of simple actions connected by logical operators.
Compound Condition	Combination of simple conditions connected by logical operators.
NegativeAction	Compound Action that may be triggered on the negatve result transition of the Compound Condition, from true to false.
Short-Circuit Evaluation	Semantics of some languages in which they will stop further computation of boolean equations as soon as the result is known. During the evaluation the second argument is only executed or evaluated if the first argument does not suffice to determine the value of the expression. In short-circuit evaluation when the first argument of the AND function evaluates to false, the overall value must be false; and when the first argument of the OR function evaluates to true, the overall value must be true.

Kindly consult [DMDICT] for all definitions used in this document.

3.3 Abbreviations

MgmtPolicyMO	Management Policy Management Object
OMA	Open Mobile Alliance

4. Introduction

The Management Policy Management Object, MgmtPolicyMO, aims to enable remote operations for Device(s) according to pre-defined Management Policies downloaded by the Management Authority. The MgmtPolicy MO specifies the capabilities of performing actions on Management Policies, such as installing, removing, updating, activating and de-activating.

A management tree object [DMTND] defined for MgmtPolicyMO will be used for setting up parameters and operational functionality necessary for managing a Management Policy object.

The objective of this document is to provide the technical specification for MgmtPolicyMO.

4.1 Version 1.0

MgmtPolicyMO V1.0 covers:

- Download, update, activation, deactivation and removal of management policy(ies) in the device
- Capability to configure specific management policies, composed of conditions and actions, based on MO objects
- Reporting the results of management policy actions to DMS

5. Management Policy Framework

5.1 Components of MgmtPolicyMO

5.1.1 Policy

A Policy is installed on the Device as an instance of the Management Policy Management Objects, and run by the MgmtPolicy Agent conforming to this specification. Each Policy SHALL be owned by a single Management Authority. It means that the Management Policy Management Object MUST have value assigned or inherited to the ACL property including only one server identifier that owns the Policy during normal operation.

The Policies SHALL be installed only through the direct management operations over the DM sessions [DMPRO]. That is, it is not possible for one Policy to install another Policy. The same is true for the update and removal of the Policy that already exists on the Device. However, one Policy MAY be activated or deactivated as a result of executing another Policy.

A Policy is defined by Management Objects consisting of conditions and actions, as described in the following sections.

Please refer to an illustrative example in Appendix C.

5.1.2 Condition

A simple condition is a logical expression defined in terms of Management Objects by the following components: a reference to the URI of an existing node in the Management Tree of the device, a comparison predicate and a predefined value object which the URI value will be compared against. Possible values used for comparison in a simple condition include date/time, location, device type and network type.

Multiple simple conditions can be combined into a Compound Condition, by using logical operators.

5.1.3 Action

A simple action is defined in terms of Management Objects by the following components: a reference to the URI of an existing node in the Management Tree of the device and the DM Command type.

The DM Command type represents what kind of OMA DM command will be executed on the node represented by that URI reference, either the Exec or Replace command.

Multiple simple actions can be combined into a Compound Action, by using logical operators.

The results of simple actions in a Compound Action SHALL be combined in order to return an aggregated action result back to the Management Authority.

An Action is a Compound Action triggered on the result transition of the Compound Condition, from ‘false’ to ‘true’.

A NegativeAction is a Compound Action which MAY be triggered on the result transition of the Compound Condition, from ‘true’ to ‘false’.

A common usecase is a given device capability, such as hardware feature (camera, speaker, memory card), I/O (usb, gps, peripherals) or connectivity (WiFi, Bluetooth, 3G, LTE) being enabled while a given condition is present (Action) or that capability being disabled when the same condition is absent (NegativeAction). Example of Compound Condition is a combination of time, location and/or network type.

6. Management Policy Management Objects (Normative)

The Management Objects associated with each Management Policy are assembled under the placeholder node x, dynamically or statically created, as displayed in figure 1.

Protocol Compatibility: This object is compatible with OMA Device Management protocol specifications, version 1.3 [DMPRO].

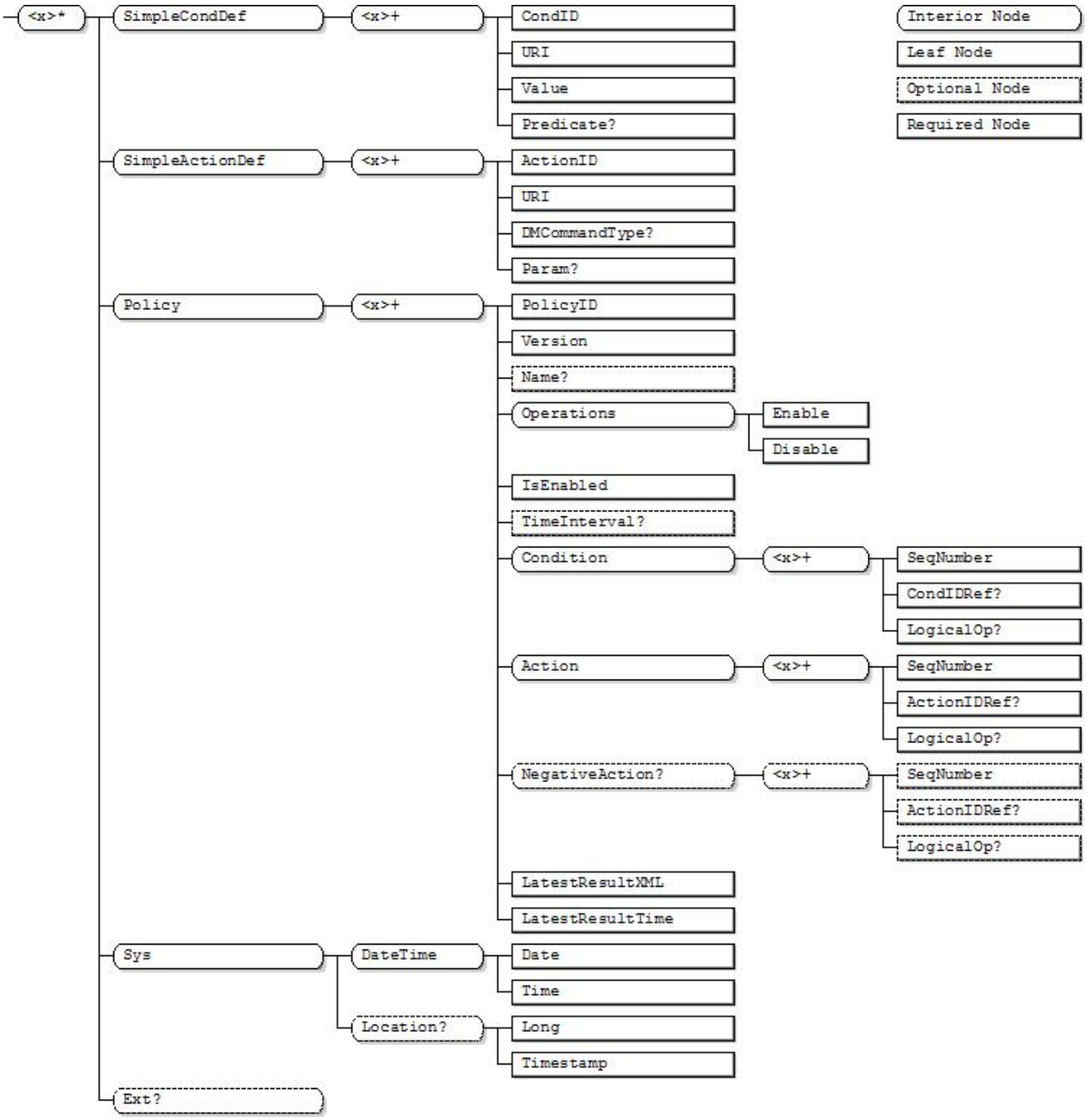

Figure 1: Mgmt Policy MO

<x>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This placeholder node is the root node for the Mgmt Policy MO. The parent node of this node defines the location of this MO in the Management Tree of the device. The type of this node MUST be the Management Policy Management Object ID "urn:oma:mo:oma-mgmtpolicymo:1.0".

<x>/SimpleCondDef

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the parent node for all simple condition definitions in the Mgmt Policy MO.

<x>/SimpleCondDef/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node is the parent node for one simple condition definition in the Mgmt Policy MO.

<x>/SimpleCondDef/<x>/CondID

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

The value of this leaf node is the identifier of the simple condition.

<x>/SimpleCondDef/<x>/URI

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

The value of this leaf node is the URI of the node in the Management Tree of the device, which will be used for evaluating the simple condition.

<x>/SimpleCondDef/<x>/Value

Status	Occurrence	Format	Min. Access Types
Required	One	bin	Get

The value of this leaf node is compared against the value of the sibling <x>/SimpleCondDef/<x>/URI node for evaluating the simple condition. If the data type associated with the node pointed to by the value of the sibling <x>/SimpleCondDef/<x>/URI node is not binary, the device MUST convert the value of this node to the proper data type (e.g. chr, int, bool etc.) prior to making the comparison.

<x>/ SimpleCondDef/<x>/Predicate

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

The value of this leaf node is the predicate that is used for evaluating the simple condition. The allowed values for this node are as per the following table:

0	Equals (==)
1	Not equals (!=)
2	Less than (<)
3	Less than or equal to (<=)
4	Greater than (>)
5	Greater than or equal to (>=)

If this node is absent, it is assumed that the predicate is *Equals* (i.e. value 0).

<x>/SimpleActionDef

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the parent node for all simple action definitions in the Mgmt Policy MO.

<x>/SimpleActionDef/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node is the parent node for one simple action definition in the Mgmt Policy MO.

<x>/SimpleActionDef/<x>/ActionID

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

The value of this leaf node is the identifier of the simple action.

<x>/SimpleActionDef/<x>/URI

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

The value of this leaf node is the URI of the node in the Management Tree of the device upon which the simple action is to be executed.

<x>/SimpleActionDef/<x>/DMCommandType

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

The value of this leaf node is the type of the OMA-DM command that needs to be executed on the node in the Management Tree whose URI is the value of the sibling <x>/SimpleActionDef/<x>/URI node. The allowed values for this node are as per the following table:

0	Exec
1	Replace

If this node is absent, it is assumed that the DM command type is Exec (i.e. value 0).

If the value of this node is 1 (i.e. Replace), its sibling <x>/SimpleActionDef/<x>/Param node MUST be present.

<x>/SimpleActionDef/<x>/Param

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bin	Get

The value of this leaf node value is the parameter for the DM Command in the case where the DM command type is Replace i.e. the value of the sibling <x>/SimpleActionDef/<x>/DMCommandType node is 1. If the value of the sibling <x>/SimpleActionDef/<x>/DMCommandType node is 0 (i.e. Exec), this node MUST NOT be present.

If the data type associated with the node pointed to by the value of the sibling <x>/SimpleActionDef/<x>/URI node is not binary, the device MUST convert the value of this node to the proper data type (e.g. chr, int, bool etc.) prior to replacing the existing value of the node.

<x>/Policy

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the parent node for all the Policies that constitute the device management policy.

<x>/Policy/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node is the parent node for one of the Policies that constitute the device management policy.

<x>/Policy/<x>/PolicyID

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the Policy ID of the Policy.

<x>/Policy/<x>/Version

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the Version of a Policy.

<x>/Policy/<x>/Name

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies the human readable name of the Policy.

<x>/Policy/<x>/Operations

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is a parent node for operation that can be executed on a Policy.

<x>/Policy/<x>/Operations/Enable

Status	Occurrence	Format	Min. Access Types
Required	One	null	Exec

This node is used with Exec command to enable the Policy. Once enabled, the IsEnabled node value transits to the Enabled.

<x>/Policy/<x>/Operations/Disable

Status	Occurrence	Format	Min. Access Types
Required	One	null	Exec

This node is used with Exec command to disable the Policy. Once disabled, the IsEnabled node value transits to the Disabled.

<x>/Policy/<x>/IsEnabled

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

This leaf node specifies if a Policy is enabled (´true') or disabled ('false'). If the Policy is disabled, no actions related to the Policy are performed.

<x>/Policy/<x>/TimeInterval

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

This leaf node is used to define a time interval by second. A Policy action is allowed to be executed only after the specified time interval counted from the previous action execution. If the node does not exist or does not contain the positive value, no time interval restricts for the MgmtPolicyMO Client to execute the Policy.

<x>/Policy/<x>/Condition

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the parent node for all the conditions that are included in the device management policy.

<x>/Policy/<x>/Condition/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node is the parent node for one of the Compound Conditions that constitute the device management policy.

<x>/Policy/<x>/Condition/<x>/SeqNumber

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

The value of this leaf is used for sequencing the sibling nodes, CondIDRef and/or LogicalOP, which will be stacked in the correct order to represent the Compound Condition expression, as in the Examples in Appendix C.

<x>/Policy/<x>/Condition/<x>/CondIDRef

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

The value of this leaf node points to the identifier of a simple condition, which is defined in the SimpleCondDef subtree of this MO. This node is mutually exclusive with its sibling LogicalOp node.

<x>/Policy/<x>/Condition/<x>/ LogicalOp

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

The value of this leaf node indicates the logical operator to use for constructing the Compound Condition from simple constituent conditions. The allowed values for this node are as per the following table:

Values	Meaning
0	NOT
1	OR
2	AND

This node is mutually exclusive with its sibling CondIDRef node.

<x>/Policy/<x>/Action

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the parent node for all the actions that are included in the device management policy. Action MUST be triggered on Compound Condition result transition from ‘false’ to ‘true’.

<x>/Policy/<x>/Action/<x>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Get

This interior node is the parent node for one of the Compound Actions that constitute the device management policy.

<x>/Policy/<x>/Action/<x>/SeqNumber

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

The value of this leaf is used for sequencing the sibling nodes, ActionIDRef and/or LogicalOP, which will be stacked in the correct order to represent the Compound Action expression, as in the Examples in Appendix C.

<x>/Policy/<x>/Action/<x>/ActionIDRef

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

The value of this leaf node points to the identifier of a simple action, which is defined in the SimpleActionDef subtree of this MO. This node is mutually exclusive with its sibling LogicalOp node.

<x>/Policy/<x>/Action/<x>/LogicalOp

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

The value of this leaf node indicates the logical operator to use for constructing the Compound Action from simple constituent actions. The allowed values for this node are as per the following table:

Values	Meaning
0	NOT
1	OR
2	AND
3	SOR
4	SAND

Logical Operators will operate on simple actions results: true if a simple action has been succeeded (success status code - e.g. OK) and false otherwise.

Please refer to Appendix C.2 - Action Operators, for information about how the operators above apply to simple action results, and how the short-circuit operators SAND and SOR may be used to control the execution of actions. This node is mutually exclusive with its sibling ActionIDRef node.

<x>/Policy/<x>/NegativeAction

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is the parent node for all the actions that are included in the device management policy.

NegativeAction MUST be triggered on Condition result transition from ‘true’ to ‘false’, if this node is present. If NegativeAction is absent, no actions will be triggered on Compound Condition result transitions from ‘true’ to ‘false’.

<x>/Policy/<x>/NegativeAction/<x>

Status	Occurrence	Format	Min. Access Types
Optional	OneOrMore	node	Get

This interior node is the parent node for one of the Compound Actions that constitute the device management policy.

<x>/Policy/<x>/NegativeAction/<x>/SeqNumber

Status	Occurrence	Format	Min. Access Types
Optional	One	int	Get

The value of this leaf is used for sequencing the sibling nodes, ActionIDRef and/or LogicalOP, which will be stacked in the correct order to represent the Compound Action expression, as in the Examples in Appendix C.

<x>/Policy/<x>/NegativeAction/<x>/ActionIDRef

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

The value of this leaf node points to the identifier of a simple action, which is defined in the SimpleActionDef subtree of this MO. This node is mutually exclusive with its sibling LogicalOp node.

<x>/Policy/<x>/NegativeAction/<x>/LogicalOp

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	int	Get

The value of this leaf node indicates the logical operator to use for constructing the Compound Action from simple constituent actions. The allowed values for this node are as per the following table:

Values	Meaning
0	NOT
1	OR
2	AND
3	SOR
4	SAND

Logical Operators will operate on simple actions results: true if a simple action has been succeeded (success status code - e.g. OK) and false otherwise.

Please refer to Appendix C2 - Action Operators, for information about how the operators above apply to simple action results, and how the short-circuit operators SAND and SOR may be used to control the execution of actions. This node is mutually exclusive with its sibling ActionIDRef node.

<x>/Policy/<x>/LatestResultXML

Status	Occurrence	Format	Min. Access Types
Required	One	xml	Get

The value of this leaf node is the Compound Action result code and the simple action result codes limited to simple actions which have been executed. The value is in XML format and conforms to the Generic Alert [DMPRO] mechanism, as specified in Appendix D.1.

<x>/Policy/<x>/LatestResultTime

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the date and time of the latest Policy Report Alert sent by the MgmtPolicyMO Client, associated with the value of the sibling <x>/Policy/<x>/Action/LatestResultTime. The representation of this node MUST follow the [YYYY]-[MM]-[DD] T[hh]:[mm]Z format, as defined by ISO 8601.

<x>/Policy/<x>/EnableSimpleActionReport

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	bool	Get

This leaf node is used to indicate whether the simple action result codes will be included in the Policy Report Alert or not. The simple action result codes will be included in the Policy Report alert when the value is "True." If this node is not present, the default value of "True" MUST apply.

<x>/Sys

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This interior node is the parent node for the device´s system properties in the Mgmt Policy MO. The child leaf nodes MAY be used or be combined with when defining Management Policy Conditions.

<x>/Sys/DateTime

Status	Occurrence	Format	Min. Access Types
Required	One	node	Get

This node contains the current device date and time. The value may be used to support scheduled execution of Management Policy Actions.

<x>/Sys/DateTime/Date

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

The value of this leaf node contains the current device system date value. The representation of this node MUST follow the compact date notation [YYYYMMDD] format, as defined by [ISO 8601].

<x>/Sys/DateTime/Time

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

The value of this leaf node contains the current device system date value. The representation of this node MUST follow the compact time notation [hhmm] format, as defined by [ISO 8601].

<x>/Sys/Location

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This node contains the data from the last known location.

<x>/Sys/DeviceLocation/Lat

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node contains the latitude as defined in [3GPP-TS_23.032], section 6.1 i.e. as an integer in the range of -8388607 to 8388607. If parent node Location is implemented, this leaf node and its siblings are required.

<x>/Sys/Location/Long

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node contains the longitude as defined in [3GPP-TS_23.032], section 6.1 i.e. as an integer in the range of -8388607 to 8388607. If parent node Location is implemented, this leaf node and its siblings are required.

<x>/Sys/Location/Timestamp

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node contains the timestamp that the latest location data is collected. The representation of this node MUST follow the [YYYY]-[MM]-[DD] T[hh]:[mm]Z format, as defined by [ISO 8601]. If parent node Location is implemented, this leaf node and its siblings are required.

<x>/Ext

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is for vendor-specific extensions to the Mgmt Policy MO.

7. Management Policy Alerts (Normative)

This section describes the alerts that are supported with the MgmtPolicyMO functionality.

7.1 Policy Report Alert

The Policy Report Alert is issued by the MgmtPolicyMO Client to report the results of the execution of the policy actions to the MgmtPolicyMO Server.

That alert MUST carry a XML body that conforms to the Generic Alert [DMPRO] mechanism, as specified in Appendix D.1.

The XML body MUST contain the Policy result code. In addition, it MUST contain the list with the individual simple action result codes, limited to the simple actions which have been executed, unless the **EnableSimpleActionReport** node is present and the value of this node is "False". Some simple actions which are connected by short-circuit operators SOR and SAND in a compound action MAY NOT be executed. Whereas all the simple actions connected by regular operators MUST be executed.

The alert message includes the following data:

- <Meta>/<Type> element: Contains the media type of the alert content. The value MUST be the alert type identifier 'urn:oma:at:oma-mgmtpolicymo:1.0:policyreport'.
- <Meta>/<Format> element: Contains the format of the alert content. The value MUST be 'xml'.
- <Meta>/<Mark> element: Contains the importance level of the alert message. This element is optional.
- <Source>/<LocURI> element: Contains the source address of the result of the Policy MO. The value MUST be the absolute URI [DMTND] of the <x>/Policy/<x>/LatestResultXML node which stores the Policy Report results currently being reported.
- <Item>/<Data> element: Contains the information of the Policy report in XML format as defined in Appendix D.1. The Policy result code and simple action result codes are reported respectively in the following XML fields:
    - <PolicyReport>/<ResultCode> field: Contains the Policy result code. Please refer to Section 7.2.1 for a list of Policy result codes.
    - <PolicyReport>/<SimpleActionResultList>/<SimpleActionResult>/<ResultCode> field: Contains the result code for each simple action, only for the simple actions which have been executed, representing the result of the execution of a DM command, locally invoked by the Device. In other words, this field is equal to the value a DM Client would return upon a regular remote invocation of a DM command by the Server.

The following is an example of the Policy Report Alert message, based on a successful execution of the Management Policy example in C.1:

```
<Alert>
  <CmdID>2</CmdID>
  <Data>1226</Data>       <!-- Generic Alert -->
  <Item>
    <Source><LocURI>./PolicyMO/Policy/1/LatestResultXML</LocURI></Source>
    <Meta>
      <Type xmlns='syncml:metinf'>
        urn:oma:at:oma-mgmtpolicymo:1.0:policyreport
      </Type>
      <Format xmlns='syncml:metinf'>xml</Format>
      <Mark xmlns='syncml:metinf'>informational</Mark> <!-- Optional -->
    </Meta>
    <Data>
      <![CDATA[
        <PolicyReport>
          <ResultCode>200</ResultCode> <!—- Composed action result code -->
          <SimpleActionResultList>
            <ResultCode ActionID="1">200</ResultCode>
            <ResultCode ActionID="2">200</ResultCode>
          </SimpleActionResultList>
        </PolicyReport>
       ]]>
     </Data>
   </Item>
</Alert>
```

7.2 Result Codes

Two kinds of result code are defined here - Policy result code and Simple action result code.

Policy result code MUST be sent in <PolicyReport>/<ResultCode> element of the Generic Alert [DMPRO] message as an integer value representing the result of the overall Policy execution. Simple action result code MUST be sent in <PolicyReport>/<SimpleActionResultList>/<SimpleActionResult>/<ResultCode> element as an integer value representing the result of the execution of a DM command, locally invoked by the Device.

7.2.1 Policy Result Codes

The Policy result code MUST be one of the values defined below:

Result Code	Result Message	Informative Description of Status Code Usage
1200	Successful	The policy is triggered and related action is executed successfully
1250-1299	Successful – Vendor Specified	Successful Operation with vendor specified Result Code.
1400	Error	MgmtPolicyMO client error
1401	Action error	The policy is triggered and related action is executed unsuccessfully
1402	Referred Simple Action does not exist	The simple action referred in composed action does not exist.
1403	Referred Simple Condition does not exist	The simple condition referred in composed condition does not exist.

1404	Positive Compound Action Action is not valid	The Positive Compound Action logical expression specified in Policy/Action is syntactically invalid.
1405	Negative Compound Action NegativeAction is not valid	The Negative Compound Action logical expression specified in Policy/NegativeAction is syntactically invalid
1406	Compound Condition is not valid	The Compound Condition logical expression specified in Policy/Condition is syntactically invalid.
1450-1499	Error - Vendor Specified	Error encountered for Operation with vendor specified Result Code

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-TS-MgmtPolicyMO-V1_0	2 May 2013	All	Baseline document
	26 Jun 2013	2.1, 4, 5, 6, App C	Incorporated CRs: OMA-DM-MgmtPolicyMO-2013-0017R02-CR_TS_Intro_Components OMA-DM-MgmtPolicyMO-2013-0018R04-CR_TS_Management_Objects Editorial changes
	04 Jul 2013	2.2, 5.1.3, 6, C.2	Incorporated CR: OMA-DM-MgmtPolicyMO-2013-0019R01-CR_TS_General_Editorial_Corrections Editorial changes
	24 Jul 2013	1, 2.1, 6, 7, D	Incorporated CRs: OMA-DM-MgmtPolicyMO-2013-0020-CR_TS_Scope OMA-DM-MgmtPolicyMO-2013-0021-CR_TS_Bug_Fix OMA-DM-MgmtPolicyMO-2013-0022R01-CR_TS_Alerts
	21 Aug 2013	6, 7.1, D.1	Incorporated CR: OMA-DM-MgmtPolicyMO-2013-0023R01-CR_TS_Alert_Results
	07 Nov 2013	3.2, 5.1.2, 5.1.3, 6, 7.1, 7.2, C.3	Incorporated CRs: OMA-DM-MgmtPolicyMO-2013-0026-CR_Result_code OMA-DM-MgmtPolicyMO-2013-0028R01-CR_ElseAction Editorial changes
	18 Dec 2013	2, 3.2, 6, 7.2.1, C.1, C.2.2, C.4	Incorporated CRs: OMA-DM-MgmtPolicyMO-2013-0029R01-CR_SystemProperties OMA-DM-MgmtPolicyMO-2013-0030R01-CR_Interoperability OMA-DM-MgmtPolicyMO-2013-0032R01-CR_MgmtPolicy_Examples OMA-DM-MgmtPolicyMO-2013-0033-CR_Editorial
	19 Feb 2014	2.2, 3.2, 4.1, 5.1.3, 6, 7.1, 7.2.1, B, C.1, C.3, C.4	Incorporated CR: OMA-DM-MgmtPolicyMO-2014-0004-CR_CONRR_B001_to_B007 Editorial changes
Candidate Version OMA-TS-MgmtPolicyMO-V1_0	01 Apr 2014	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2014-0064-INP_MgmtPolicyMO_V1_0_ERP_and_ETR_for_Candidate_approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for MgmtPolicyMO Tree Structure

Item	Function	Reference	Requirement
MgmtPolicyMO-T-001-M	Use of appropriate Management Object identifier	Section 6	MgmtPolicyMO-HLF-9 and MgmtPolicyMO-HLF-8
MgmtPolicyMO-T-002-M	Support for Required nodes under root node	Section 6	MgmtPolicyMO-HLF-9 and MgmtPolicyMO-HLF-8
MgmtPolicyMO-T-003-O	Support for Optional nodes	Section 6	MgmtPolicyMO-HLF-9 and MgmtPolicyMO-HLF-8
MgmtPolicyMO-T-004-M	Support for Required nodes under an Optional node if the Optional node is supported	Section 6	MgmtPolicyMO-HLF-9 and MgmtPolicyMO-HLF-8

B.2 SCR for MgmtPolicyMO Server

Item	Function	Reference	Requirement
MgmtPolicyMO-S-001-M	Support Management Policy Management Object	Section 6	MgmtPolicyMO-DMS-01 and MgmtPolicyMO-DMS-02

B.3 SCR for MgmtPolicyMO Client

Item	Function	Reference	Requirement
MgmtPolicyMO-C-001-M	Each Policy MO is owned by a single Management Authority	Section 5.1.1	MgmtPolicyMO-ATHOR-
MgmtPolicyMO-C-002-M	A Policy is installed only through the direct management operations over the DM sessions.	Section 5.1.1	MgmtPolicyMO-ATHOR-
MgmtPolicyMO-C-003-M	upport for Compound Condition, composed of multiple simple conditions.	Section 5.1.2	MgmtPolicyMO-MP-03
MgmtPolicyMO-C-004-M	Support for Compound Action, composed of multiple simple actions.	Section 5.1.3	MgmtPolicyMO-MP-04
MgmtPolicyMO-C-005-M	Support for Action and NegativeAction Compound Actions, triggered on Compound Conditions result transitions.	Section 5.1.3, Appendix C.3	MgmtPolicyMO-MP-05
MgmtPolicyMO-C-006-M	Support for Regular and Short-Circuit Action Operators.	Section 6, Appendices C.2.1, C.2.2	MgmtPolicyMO-MP-04

B.4 SCR for MgmtPolicyMO Policy Report Alert

Item	Function	Reference	Requirement
MgmtPolicyMO-P-001-M	Carry a XML body that conforms to the Generic Alert mechanism, as specified in Appendix D.1	Section 7.1	MgmtPolicyMO-DMS-01 and MgmtPolicyMO-DMS-02
MgmtPolicyMO-P-002-M	Contain the Policy result code	Section 7.1, Section 7.2.1 (Policy Result Code table)	MgmtPolicyMO-DMS-01 and MgmtPolicyMO-DMS-02
MgmtPolicyMO-P-003-M	Contain the alert type identifier: urn:oma:at:oma-mgmtpolicymo:1.0:policyreport	Section 7.1	MgmtPolicyMO-DMS-01 and MgmtPolicyMO-DMS-02
MgmtPolicyMO-P-004-M	The value of <Meta>/<Format> element is 'xml'	Section 7.1	MgmtPolicyMO-DMS-01 and MgmtPolicyMO-DMS-02
MgmtPolicyMO-P-005-M	The value of <Source>/<LocURI> element is the absolute URI [DMTND] of the <x>/Policy/<x>/LatestResultXML node which stores the Policy Report results currently being reported.	Section 7.1	MgmtPolicyMO-Device -01
MgmtPolicyMO-P-006-M	Support for Simple Action result codes, with the DM command result for each simple action locally invoked by the Device.	Section 7.1	MgmtPolicyMO-Device -01

Appendix C. Additional Information

C.1 Management Policy Examples

C.1.1 Example 1

The following is an illustrative example of a hypothetical Device Management Policy for camera and ring mode:

> *If the time is between 9:00 a.m. and 10:30 a.m. or between 3:00 p.m. and 5:00 p.m., then disable the camera and put the phone in vibrate mode, else enable the camera and restore phone ringtone to regular mode.*

In this hypothetical example, we have the following Compound Condition, Action and NegativeAction:

- time >= 0900 hours {simple condition}
- time <= 1030 hours {simple condition}
- time >= 1500 hours {simple condition}
- time <= 1700 hours {simple condition}
- (time >= 0930 hours AND time <= 1030 hours) OR (time >= 1500 hours AND time <= 1700 hours) {Compound Condition}
- disable camera {simple action}
- put phone in vibrate mode {simple action}
- disable the camera AND put phone in vibrate mode {Action - positive Compound Action}
- enable camera {simple action}
- put phone ringtone in regular mode {simple action}
- enable the camera AND put phone ringtone in regular mode {NegativeAction – negative Compound Action}

We will assume that to enable and disable the camera the device has to execute the Exec command on the following nodes in the Management Tree:

> /DCMO/Camera/Operations/Enable
>
> /DCMO/Camera/Operations/Disable

Furthermore, let us assume that to put the phone in vibrate and regular mode, the device replaces the value of the following node to 0 and 1, respectively:

> /DCMO/Ringer/state

Additionally, the device exposes its time value via the following MgmtPolicy MO node:

> /MPMO/ Sys/DateTime/Time

Figure 2 represents an instance of MgmtPolicyMO.

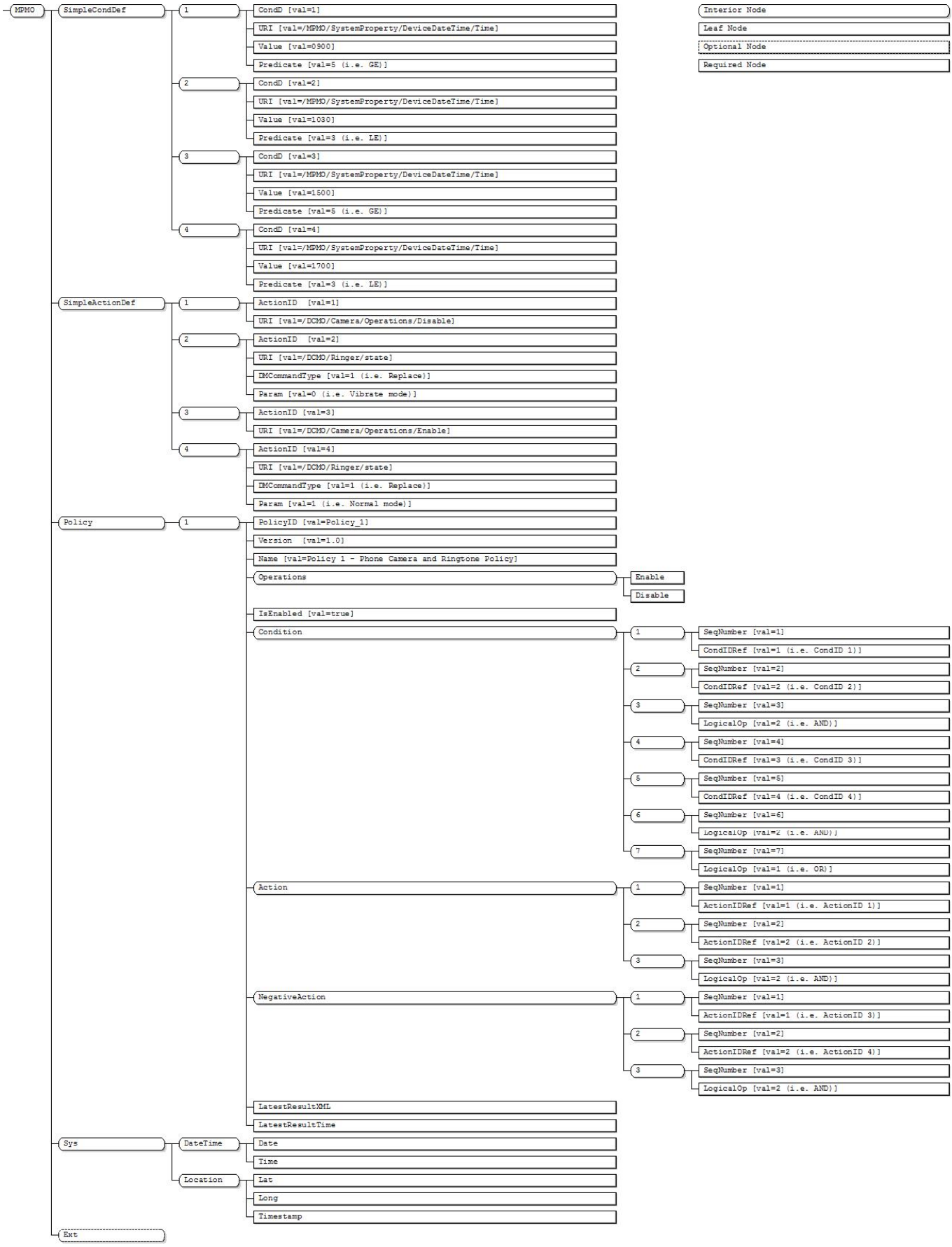

Figure 2: An instance of the Mgmt Policy MO: Camera and Ring Mode Policy

The leaf nodes under below the Policy/Condition node were sequenced in the following order:

CondID [val=1], CondID [val=2], LogicalOp [val=AND], CondID [val=3], CondID [val=4], LogicalOp [val=AND], LogicalOp [val=OR].

This happens to be the expression tree of the Compound Condition in Reverse Polish Notation as shown in **Figure 3**.

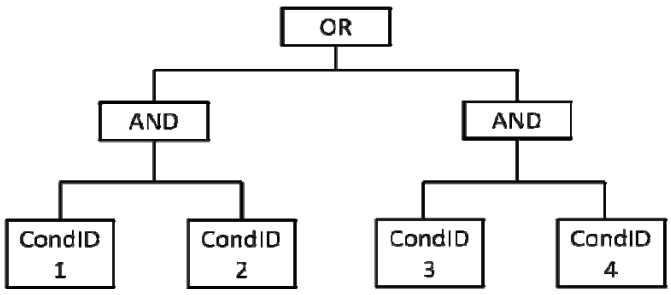

Figure 3: Expression Tree for Compound Condition

The specified Action will be triggered in the positive transition (from false to true) of the Compound Condition result.

The leaf nodes under the Policy/Action node were sequenced in the following order:

ActionID [val=1], ActionID [val=2], LogicalOp [val=AND].

This happens to be the expression tree of the Compound Action in Reverse Polish Notation as shown in **Figure 4**.

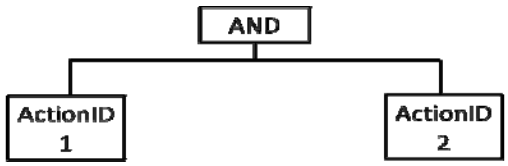

Figure 4: Expression Tree for Compound Action

In this example using the regular action operator AND, both simple actions will be executed and the Compound Action would result in success (i.e. OK) only if both simple actions resulted in success (i.e. OK) status codes.

The specified NegativeAction will be triggered in the negative transition (from true to false) of the Compound Condition result.

Similarly to the Action, the leaf nodes under the Policy/NegativeAction node were sequenced in the following order:

ActionID [val=3], ActionID [val=4], LogicalOp [val=AND].

C.1.2 Example 2

The following is an illustrative example of a hypothetical Device Management Policy for software update:

If Application XYZ version is lower than or equal V1.0, date is greater than or equal 01-Jan-2014 and battery level is greater than 20%, then perform a XYZ Software Update and restart the phone if update operation has been successful.

In this hypothetical example, we have the following Compound Condition and Compound Action:

- version <= 10 {simple condition}
- date >= 20140101 {simple condition}
- battery level > 20 percent {simple condition}
- (version <= 10 AND date >= 20140101 AND battery level > 20 percent) {Compound Condition}
- perform a Software Update to V20 {simple action}
- restart the phone {simple action}
- perform a Software Update to V20 SAND restart the phone {Action - positive Compound Action}

SAND Action operator means the second operand (restart the phone) will be evaluated and executed only if first operand (SW update) be successfully executed.

No Negative Action NegativeActions have been defined, so no actions will be triggered on the negative transition of the Compound Condition result (from true to false). Only the defined Positive Action will be triggered on the positive transition of the Compound Condition result (from false to true).

We will assume that, in order to perform a XYZ Software Update (download and install) and to restart the phone, the device has to execute the Exec command on the following nodes in the Management Tree:

/SCOMO/Download/XYZ/Operations/DonwloadInstall

/DCMO/Phone/Operations/Disable

Furthermore, let us assume that to check the installed XYZ version and battery level, the device performs a Get command on the following nodes, respectively:

/SCOMO/Inventory/Deployed/XYZ/Version

/DIAGMON/DiagmonData/1/BatteryLevel

Additionally, the device exposes its date value via the following MgmtPolicy MO node:

/MPMO/Sys/DateTime/Date

The leaf nodes under below the Policy/Condition node will be sequenced in the following order:

CondID [val=1], CondID [val=2], CondID [val=3], LogicalOp [val=AND]

The specified Action will be triggered in the positive transition (from false to true) of the Compound Condition result.

The leaf nodes under below the Policy/Action node will be sequenced in the following order:

CondID [val=1], CondID [val=2], LogicalOp [val=SAND]

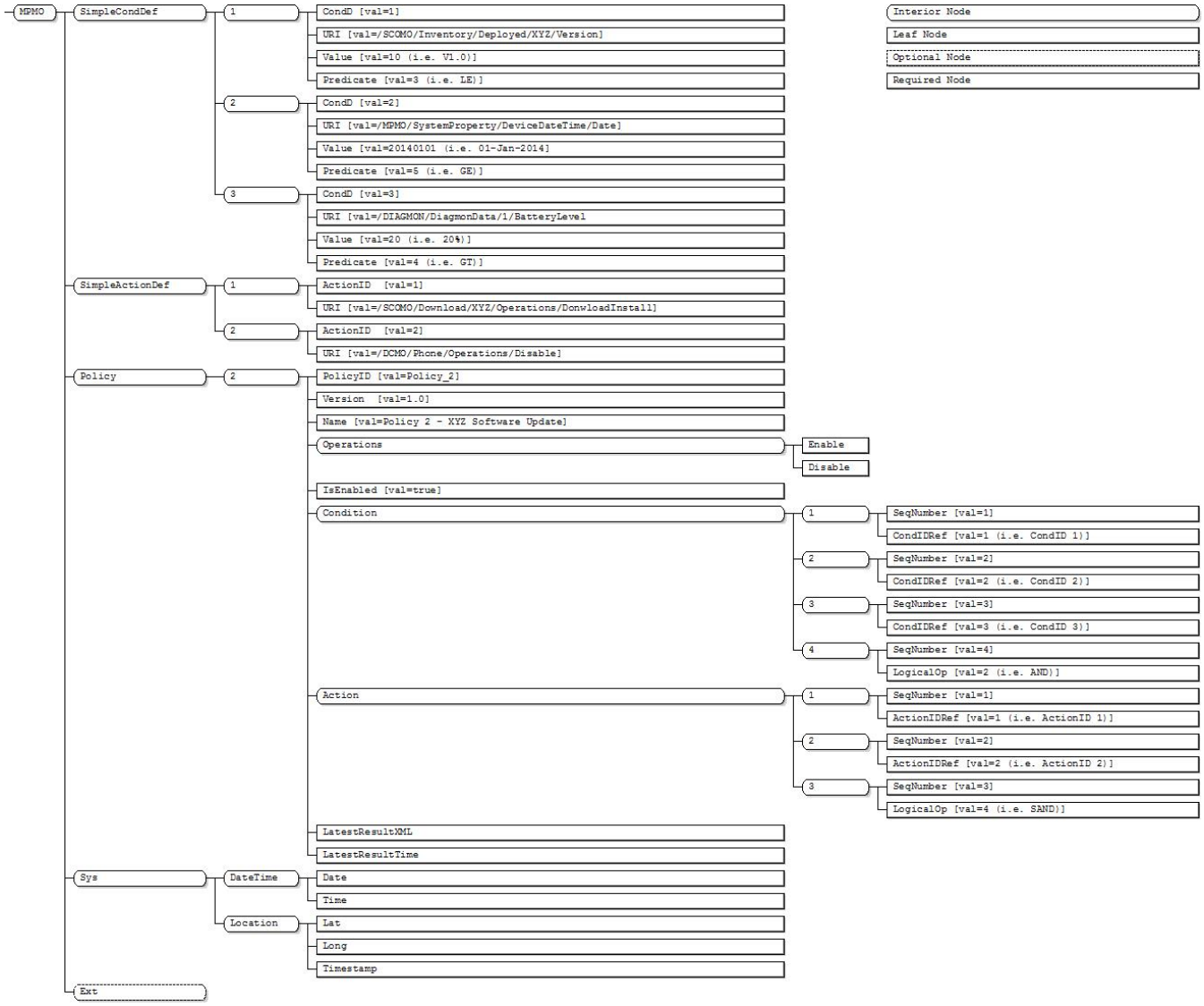

Figure 5: An instance of the Mgmt Policy MO: Software Update Policy

C.2 Action Operators

When using action operators, simple action results may be interpreted as following:

If a simple action results in success (OK), it can be logically interpreted as Boolean true result.

Otherwise, if it results in fail (NOK), it can be logically interpreted as Boolean false result.

The next sessions describes how regular and short-circuit operators may be applied to simple actions.

C.2.1 Regular Action Operators

Please consider the following expressions representing compound actions based on two simple actions Action_1, Action_2 connected by regular operators:

Action_1 **AND** Action_2:

- Expression result is success (OK) status code only if both simple actions result in success (OK) status codes.

Action_1 **OR** Action_2:

- Expression result is failed (NOK) status code only if both simple actions result in failed (NOK) status codes.

NOT Action_1:

- Expression result is success (OK) status code if simple Action_1 results in failed (NOK) status code.
- Expression result is failed (NOK) status code if simple Action_1 results in success (OK).status code

For regular operators, actions are also executed as in a list: in this example Action_1 is always executed regardless Action_2 result, and vice-versa.

C.2.2 Short-Circuit Action Operators

MgmtPolicyMO adapts Short-Circuit Evaluation in order to calculate a Compound Action result, as follows:

- The successful result of a simple action (OK) is interpreted as Boolean true, while the failed (NOK) result of a simple action is interpreted as Boolean false.
- In order to differentiate the Short-Circuit Evaluation from the regular evaluation of a Compound Action, MgmtPolicyMO uses the short-circuit operators SAND (short-circuit AND) and SOR (short-circuit OR).
- During the Short-Circuit Evaluation, as soon as the result of Compound Action result is determined, the remaining simple actions not evaluated yet, if any, MUST NOT be executed.

Please consider the following expressions representing Compound Actions, each one based on two simple actions Action_1 and Action_2 connected by short-circuit operators. The expression results are defined similarly to the regular operators, but the Short-Circuit operators can control how simple actions are going to be individually executed, as follows:

Action_1 **SAND** Action_2:

- Expression result is success (OK) status code only if both simple actions result in success (OK) status code – same as in regular AND.
- Simple Action_2 is only executed if simple Action_1 results in success (OK) status code. In other words, if simple Action_1 results in fail (NOK) status code, simple Action_2 is not executed (short-circuit)

Action_1 **SOR** Action_2:

- Expression result is failed (NOK) status code only if both simple actions result in fail (NOK) status code – same as in regular OR.
- Simple Action_2 is only executed if simple Action_1 results in fail (NOK) status code. In other words, if simple Action_1 results in success (OK) status code, simple Action_2 is not executed (short-circuit)

C.3 Trigger

An Action or a NegativeAction may be triggered according to the Composed Condition result transitions false-> true and true-> false respectively.

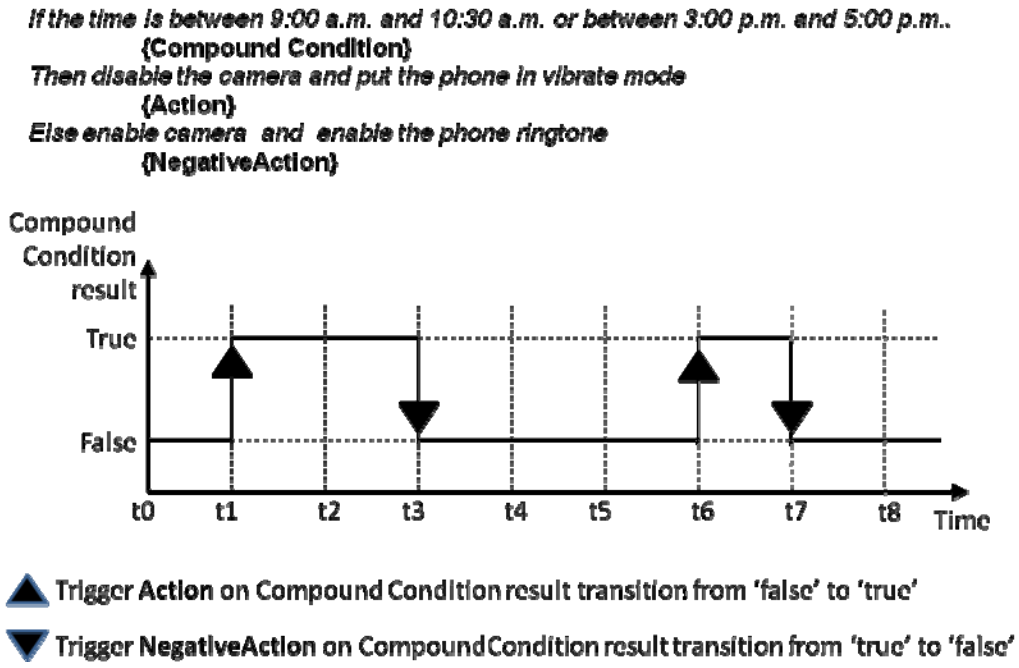

Figure 6: Trigger Action or NegativeAction based on Compound Condition result transition

C.4 Interoperability with OMA DM Enablers (Informative)

MgmtPolicyMO may interoperate by integrating Management Policy conditions and actions based on properties from different enablers.

The properties used in Management Policy conditions and actions may be based on:

- Management Policy´s Sys fields
- DM 1.3 properties, where DM 1.3 is prerequisite for MgmtPolicyMO (examples Device Type and Network Type)
- Other OMA DM Enabler´s properties

That is accomplished by using the URI fields in Simple Conditions and Simple Actions:

Simple Conditions: <x>/SimpleCondDef/<x>/URI refers to a readable node in the Management Tree of the device. This node must support at least Get Access and its value .will be used for evaluating the simple condition

Simple Actions: <x>/SimpleActionDef/<x>/URI refers to an executable or writable node in the Management Tree of the device, depending of the type of the DM command configured to be executed when an Action is triggered:

- If DM command to be executed is Exec, its value will contain a command or statement.
- If DM command to be executed is Replace, its value will be replaced by an additional Param passed together within the Simple Action.

The table below lists examples of Properties commonly supported by some OMA DM Enablers and how they may be referenced by the Management Policy MO´s simple conditions and simple actions:

Enabler	Property	Simple Condition URI	Simple Action URI
MgmtPolicyMO	Date, Time, Location	<x>/ Sys/DateTime <x>/ Sys/Location	
DM 1.3 StdObj **[DMStdObj-TS]**	Device Type, Network Type	DevDetail/DevType, DevInfo/CBT	
DCMO **[DCMO-TS]**	Hardware, I/O, Connectivity, Software Capabilities. E.g. Display, Camera, Speaker, USB, GPS, WLAN, WiFi, installed apps.	<x>/Property <x>/Attached <x>/Enabled	<x>/Operations/Enable, Disable
DiagMon Functions **[DiagMonFunc-TS]**	Battery Info, Location Memory, NFC, Sensor, GSM, 3GPP UMTS, 3GPP LTE QoS, SMS Usage, MMS Usage, PhoneBook	<x>/DiagMonData/<x>/BatteryStatus, Level <x>/DiagMonData/Lat, Long DiagMonData/RAMAvail, RAMTotal <x>/DiagMonData/NfcRFSignal, NfcSoftVer, NfcAntenna <x>/DiagMonData/<x>/SensorID, SensorStatus, SensorData <x>/DiagMonConfig/ConfigParms/QoSConfigParms <x>/DiagMonData/SMSSent, SMSReceived <x>/DiagMonData/MMSSent, MMSReceived <x>/DiagMonData/<x>/PBType, PBUsed, PBFree	<x>/Operations/Update <x>/Operations/Upload <x>/Operations/Start, Stop
SCOMO **[SCOMO-TS]**	Software ID, Name, Version, Status	Download/<X>/PkgID, Name, Status Inventory/Delivered/<X>/PkgID, Name, Status Inventory/Deployed/<X>/ID, PkgIDRef, Name, Version, Status	Download/<X>/Operations/Download, DownloadInstall Inventory/Delivered/<X>/Operations/Install, Remove Inventory/Deployed/<X>/Operations/Activate, Deactivate, Remove

Appendix D. XML Schema for MgmtPolicyMO Alerts (Normative)

D.1 Policy Report Alert

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="PolicyReport">
    <xs:complexType>
      <xs:element name="ResultCode" type="xs:unsignedInt"/>
      <xs:element name="SimpleActionResultList">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="SimpleActionResult" minOccurs=”0” maxOccurs="1"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="SimpleActionResult" minOccurs="0" maxOccurs="unbounded"/>
        <xs:complexType>
          <xs:element name="ResultCode" type="xs:unsignedInt">
            <xs:complexType>
              <xs:attribute name="ActionID" type="xs:unsignedInt" use="required"/>
            </xs:complexType>
          </xs:element>
        </xs:complexType>
      </xs:element>
    </xs:complexType>
  </xs:element>
</xs:schema>
```