Device Management in SDT

Group Name: RDM
Related CR: RDM-2019-0127-Device_Management_in_SDT
Source:

- Orange: Cyrille Bareau, Leila Le Brun, Marianne Mohali, Przemyslaw Ratuszek - <firstname.name>@orange.com
- Deutsche Telekom: Andreas Kraft – Andreas.Kraft@t-systems.com
- Chordant: Bob Flynn – Bob.Flynn@chordant.io

Meeting Date: 2020-02-04
Agenda Item: TS-0023 related

Context

- We consider here the case of NoDN devices that are interworked by an IPE using the Smart Device Template (SDT) framework, i.e. mapped into oneM2M using <flexContainer> resources.
- TS-0023 models the functional aspect of devices but the remote management of the devices is not included.
- Currently, Device Management (DM) operations are modeled in oneM2M using <mgmtObj> resources.

Use Case 1 & Issue

Use-case: a NoDN device is not managed by a standardized DM protocol, but has some functions that pertain to DM (reboot, firmware update, configuration…).

An IPE is in charge of the interworking with this device.

Current workflow:

The IPE has to create and maintain:

- <flexContainer> SDT module classes for the functional behavior of the device,
- <mgmtObj> resources for the DM operations.

An AE that needs access to this device also has to handle both resource types.

Issues:

- More implementation work, for both IPE & client AE developers.
- Lack of simplicity.

Use Case 2 & Issue

Use-case: an application wants to read/write a specific parameter in a standardized Device Management Data Model.

Current workflow:

1. The parameter must be mapped as a custom attribute of a <mgmtObj> (this must be specified in TS-0001, TS-0004, TS-0006 for BBF or TS-0005 for OMA or…)
2. This custom attribute may have no *reverse mapping* for interworking/interoperability with other DM protocols.

Issues:

- Evolutions are made difficult due to the number of changes to do in specifications.
- Confusing for client AEs (the same parameter could be present or not on devices having the same SDT model).

Use Case 3 & Issue

Use-case: many NoDN devices handle information that pertain to classical DM features (battery status/level, memory availability, CPU usage…)

Current workflow: some of these features (battery) are modeled as SDT Module Classes (MC). Memory is not modeled as a MC but as a mgmtObj (MO). CPU is neither modeled as a MC nor as a MO.

Issues:

1. 'battery' MC: difficult to specify which SDT devices shall include a SDT battery or not.
2. 'memory' MO: see previous Use Case 1 (mixing MO *and* MC resources).
3. CPU: could be modeled as a MC, but issue 1 remains (see above). If modeled as a MO, see Use Cases 1 & 2.
4. It is not possible to design SDT MCs that are *transversal* to devices, unless explicitly adding them as optional MCs to *all* devices.

Use Case 4 & Issue

Use-case: a client AE has to deal with 2 types of devices e.g. lights. Some are represented as SDT Light devices, the others through a LwM2M IPE.

Current workflow:

1. SDT Light devices all share the same model whatever their underlying technology.
2. For the LwM2M light device, depending on the IPE's Interworking Function (as defined in TS-0014), the client AE may have to deal with either opaque containers, or semantically designed containers, or specialized MOs.

Issues:

- Lack of simplicity.
- Poor interoperability.

Proposed Solution (1/2)

Introduce SDT Module Classes for Device Management

- Link SDT devices to a [flexNode] <flexContainer>, which has roughly the same role as the <Node>, i.e. be the root for DM resources (MOs for the <Node> and MCs for the [flexNode]).
- Main MOs (battery, memory, reboot, firmware…) are 'translated' as MCs. Added MC for CPU usage.
- SDT Actions are used to model
  - CRUD operations on standardized Data Models (such as from BBF or OMA), so that it is possible to handle devices specific parameters if needed, without specifying them;
  - stateful or lengthy DM operations (reboot, firmware upgrade, etc.)
- All new SDT Module Classes & Actions are specified only in TS-0023.

Proposed Solution (2/2)

Expected Benefits

- Simplicity:
  - same kind of resources for Device Management / Service Management;
  - cleaner design for complex DM operations through SDT Actions (*method call*);
  - 'dashboard' interface with major DM parameters at the same level;
  - transversal, optional model for *all* SDT devices.
- Easier implementation, for both IPE & client AE developers.
- Easier specification work: evolutions only in TS-0023.
- Extended management of Device Management Data Models:
  - no need to explicitly specify parameters;
  - possible extensions to lower level IoT technology data models.
- Easier testing : see TDE-2020-0005R01- mgmtObjTPs (Convida)