|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | SDS #40 |
| Source:\* | Convida Wireless |
| Date:\* | 2019-05-20 |
| Contact:\* | Catalina Mladin, Convida, Mladin.Catalina@convidawireles.comDale Seed, Convida, Seed.Dale@convidawireles.comLu Liu, Convida, Liu.Lu@convidawireless.com |
| Reason for Change/s:\* | Provides potential solution affecting state and sequencing for action triggering. |
| CR against: Release\* | Release 4 |
| CR against: WI\* | [x]  Active <WI-0093> [ ]  MNT / < Work Item number(optional)>Is this a companion CR? Yes [ ]  No [ ] Companion CR number: (Note to Rapporteur - use latest agreed revision)Is this a mirror CR? Yes [ ]  No [ ] Mirror CR number: (Note to Rapporteur - use latest agreed revision)[ ]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TR-0060 V 0.0.1 |
| Clauses/Sub Clauses \* | Clause 7 |
| Type of change: \* | [ ]  Editorial change[ ]  Bug Fix or Correction[ ]  Change to existing feature or functionality[x]  New feature or functionalityOnly ONE of the above shall be ticked |
| Impacted other TS/TR(s) | <TS/TR number>, <Version Number>, and <Description on which aspect should be reflected in this TS/TR> |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES [x]  NO [ ] This CR may break backwards compatibility with the last approved version of the TS? YES [ ]  NO [x]  |
| Template Version: January 2017 (Do not modify) |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

GUIDELINES for Change Requests:

Provide an informative introduction containing the problem(s) being solved, and a summary list of proposals.

Each CR should contain changes related to only one particular issue/problem.

In case of a correction, and the change apply to previous releases, a separate “mirror CR” should be posted at the same time of this CR

Mirror CR: applies only when the text, including clause numbering are exactly the same.

Companion CR: applies when the change means the same but the baselines differ in some way (e.g. clause number).

Follow the principle of completeness, where all changes related to the issue or problem within a deliverable are simultaneously proposed to be made E.g. A change impacting 5 tables should not only include a proposal to change only 3 tables. Includes any changes to references, definitions, and acronyms in the same deliverable.

Follow the drafting rules.

All pictures must be editable.

Check spelling and grammar to the extent practicable.

Use Change bars for modifications.

The change should include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change. Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.

Multiple changes in a single CR shall be clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.

When subsequent changes are made to content of a CR, then the accepted version should not show changes over changes. The accepted version of the CR should only show changes relative to the baseline approved text.

## Introduction

The following introduces a solution addressing the issues discussed in contribution SDS-2019-xxxx-TR-0060 State and Sequencing in Action Triggering. The solution introduces two new resources <state> and <processManagement>, which may achieve the following functionalities.

* State-specific actions: An <action> is linked to the particular <state>(s) that it can be performed in. Only the <action>s linked by the currently active state will be monitored.
* Action branching: Each <state> may define a list of transitions, with each of them defining the corresponding transition condition and the next state to transition to. A state may transition to different states under different conditions, enabling action branching.
* Action and state sequencing: <processManagement> contains all the <state>s (as child resources) in the process. The <state>s in a progress will be activated one at a time through transitions, and the activated <state> will perform the linked <action>s, resulting in action and state sequencing. Besides, the sequencing could be initialized by meeting preconditions, and ended when reaching the final state or through exit conditions.
* Process level management: By controlling the status of the process, the process could be enabled/disabled/paused. The process may also be updated by modifying parameters associated with the <state>s, or by adding/removing <state> child resources.

### -----------------------Start of change 1-------------------------------------------

1.

# Candidate Solutions

## Solution 1: State and Sequencing for Action Triggering

##### New Resource Type: *<state>*

The *<state>* resource is proposed to store the information of a state and define the actions and transitions in this state.

**Table 7.x.x.x: Attributes of *<state>* resource**

| **Attributes of *<state>*** | **Multiplicity** | **RW/****RO/****WO** | **Description** |
| --- | --- | --- | --- |
| *currentStatus* | 1 | RO | The indicator of whether this state is currently active. This attribute may take values from “active” or “inactive”. |
| *stateActions* | 0..1 (L) | RW | The link(s) to the <action>(s) that may be performed in this state. The actions include both the ones that will be performed unconditionally when entering this state, and the ones that may be triggered by certain events/conditions within this state. |
| *stateTransitions* | 0..1 (L) | RW | The possible transition(s) that may happen from the current state. Each transition is defined as a tuple [transition criteria, next state]:Transition criteria: the event or condition that may trigger state transition;Next state: the *resourceID* of the next state to transition to.If this state is the last state in the process, this attribute will be empty.  |

After the IoT process enters a new state, the value of the *currentStatus* attribute will be changed to “active”. A timer will be started if the state has any duration constraint. Then, for all the <action>s that are linked through *stateActions* attribute, their *evalMode* attributes will be changed from “off” to “once/periodic/continuous”. For an unconditional action, it will be performed immediately; for a conditional action, the Hosting CSE will check or start to monitor the *evalCriteria* attribute of the <*action*> to determine whether the action will be performed or not. In addition, the transition criteria defined in *stateTransitions* will start to be monitored. If (one of) the event(s) defined in transition criteria happens or (one of) the defined condition(s) is met, the state transition will be triggered. At transition, the *currentStatus* will be changed to “inactive” and all the *evalMode* attributes of the corresponding actions will be changed back to “off”. If there is any ongoing action when the transition is triggered, the action will be completed or forced to end based off of local CSE policies. The Hosting CSE will then transition to the next state as indicated by the next state ID.

##### New Resource Type: <*processManagement>*

The *<processManagement>* resource is proposed to store the information of the entire process consisting of multiple states.

**Table 7.x.x.x: Child Resources of Proposed oneM2M <*processManagement*> Resource**

| **Child Resources of *<processManagement>*** | **Child Resource Type** | **Multiplicity** | **Description** |
| --- | --- | --- | --- |
| *[variable]* | *<state>* | 0..n | This resource describes the details of a particular state of an IoT process. |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 in [x]. |

**Table 7.x.x.x: Attributes of *<processManagement>* resource**

| **Attributes of *<processManagement>*** | **Multiplicity** | **RW/****RO/****WO** | **Description** |
| --- | --- | --- | --- |
| *processStatus* | 1 | RW | The status for the entire IoT process. The supported values for this attribute are:Disabled: The process is disabled. Enabled: The process is enabled, and the Hosting CSE will start to monitor the event/condition defined in *preconditions*, if the pre-condition is met, the *processStatus* will become “Active”.Active: The process is active, which means there is one state (other than *finalState*) currently active.Paused: The process is paused and will remain in the current state and will not transition to another state until “Active” again.Ended: The process has entered the final state or exited through *exitConditions*. |
| *currentState* | 1 | RW | The *resourceID* of the <*state*> resource that the process is currently in. If the *processStatus* is not “Active”, the value of this attribute will be empty. |
| *preconditions* | 0..1 (L) | RW | This attribute specifies any conditions that must be met for the process to begin. It can be used to trigger the start of the process. When the conditions are met, the *processStatus* will become “Active”, and the Hosting CSE will transition to the state defined in *initialState*. |
| *exitConditions* | 0..1 (L) | RW | This attribute specifies what events can end the IoT process. It allows for the premature and asynchronous exit of the process. When the exit conditions are met, the *processStatus* will become “Ended”, and all states in this process will be “inactive”. |
| *initialState* | 1 | RW | The *resourceID* of the first <*state*> resource of the process. In other words, this is the state transitioned to after “*preconditions*” are met. |
| *finalState* | 0..1 | RW | The *resourceID* of the last <*state*> resource in the process. If the process is a loop, this attribute is empty. |

-----------------------End of change 1 -------------------------------------------

CHECK LIST

* Does this Change Request include an informative introduction containing the problem(s) being solved, and a summary list of proposals.?
* Does this CR contain changes related to only one particular issue/problem?
* Have any mirror CRs been posted?
* Does this Change Request make **all** the changes necessary to address the issue or problem? E.g. A change impacting 5 tables should not include a proposal to change only 3 tables?Does this Change Request follow the drafting rules?
* Are all pictures editable?
* Have you checked the spelling and grammar?
* Have you used change bars for all modifications?
* Does the change include the current and surrounding clauses to clearly show where a change is located and to provide technical context of the proposed change? (Additions of complete clauses need not show surrounding clauses as long as the proposed clause number clearly shows where the new clause is proposed to be located.)
* Are multiple changes in this CR clearly separated by horizontal lines with embedded text such as, start of change 1, end of change 1, start of new clause, end of new clause.?