|  |  |
| --- | --- |
| Input Contribution | |
| Meeting ID\* | MAS 37 |
| Title:\* | Procedure for optimizing semantic operations with reasoning support |
| Source:\* | Xu Li, Convida, [li.xu@convidawireless.com](mailto:li.xu@convidawireless.com)  Chonggang Wang, Convida, [wang.chonggang@convidawireless.com](mailto:wang.chonggang@convidawireless.com) |
| Date:\* | 2018-09-09 |
| Input related to\* | TR-0033-Study on Enhanced Semantic Enablement |
| Intended purpose of  document:\* | Decision  Discussion  Information  Other <specify> |
| Impacted other TS/TR(s) | n/a |
| Decision requested or recommendation:\* | The content is to be included in clause 8.7.6 of TR-0033 |
| 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.

# Introduction

One of the major features of the Semantic Reasoning Function (SRF) as described in TR-0033 V 4.2.0 clause 8.7 is to enhance/optimize existing semantic operations supported by oneM2M (such as semantic resource discovery or semantic query, etc.).

Accordingly, this contribution brings the next level of details regarding to this feature and proposes corresponding solution. In the meantime, the hospital facility surveillance use case described in clause 8.7.2 is also used in this contribution to illustrate how to use the proposed solution/procedure to optimize a semantic resource discovery operation with reasoning support.

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

### 8.7.6 Initiating a Semantic Operation with Reasoning Support

This clause introduces a solution for how the existing semantic operations (such as semantic query, semantic resource discovery, etc.) can benefit from semantic reasoning.

In general, it is assumed that in addition to SR, there is also a Semantic Engine or SE (e.g., a SPARQL engine) in the system. Accordingly, an oneM2M user can send a request to the SE in order to initiate a specific semantic operation, in which a SPARQL statement is often included to describe user’s needs. For the SE, when processing a specific received request (e.g., a semantic resource discovery request), it needs to collect the Involved Data Basis (IDB), which is a set of RDF triples that the SPARQL statement is to be executed on. However, the original IDB without semantic reasoning support may not be perfect or well match with the SPARQL statement (e.g., the RDF triples in IDB are described using a different ontology than the ontology adopted in the SPARQL query statement sent from the user). As a result, no result can be obtained when executing the SPARQL statement over the original IDB. When semantic reasoning is also supported in the system, one of potential enhancements is that the SE can further utilize the semantic reasoning capability provided by a SR (as a background support) in order to augment the original IDB with additional/implicit facts (i.e., the obtained reasoning result) and finally optimize the processing of the semantic operation at SE.

#### 8.7.6.1 Procedure for initiating a semantic operation with reasoning support

Figure 8.7.6.1-1 illustrates the proposed procedure for initiating a semantic operation with reasoning support and the key idea is that by utilizing the reasoning capability, the original IDB determined by the SE will be further augmented by integrating inferred facts (i.e., the reasoning result) with the existing facts in the IDB. Accordingly, the SPARQL statement will be executed on the “Augmented IDB”. Figure 8.7.6.1-1 illustrates the proposed procedure and the detailed descriptions are as follows:

Step 1: An oneM2M User-1 intends to initiate a semantic operation, which may refer to a semantic query operation, a resource discovery operation, etc. Accordingly, User-1 will compose a request message, in which a corresponding SPARQL statement is also included.

Step 2: User-1 sends the request to SE-1 in order to initiate the intended semantic operation.

Step 3: Based on the request from User-1, SE-1 starts the related processing. The first step is to determine the IDB for this request, on which the SPARQL statement is to be executed.

Below are the two examples of IDB for two different semantic operations (one is for semantic resource discovery operation and the other is for semantic query operation):

* In case of semantic resource discovery, User-1 may send a resource discovery request to a semantic-capable CSE (hosting a SE) and the “To” parameter indicates where the discovery should start, e.g., a specific resource <Resource-1>. In particular, for each of the normal resources under <Resource-1> (e.g., <Resource-2>), its <semanticDescriptor> child resource (if exists) will be evaluated in order to decide whether <Resource-2> should be included in the discovery result. In other words, when evaluating a specific resource (e.g. <Resource-2>), its IDB includes the RDF triples stored in its <semanticDescriptor> child resource. Then, the SPARQL statement will be executed over its <semanticDescriptor> child resource (if exists) in order to decide whether <Resource-2> should be included in the discovery result.
* In case of semantic query, User-1 may send a semantic query request to a semantic-capable CSE (hosting a SE) and the “To” parameter may refer to a specific resource <Resource-1>, which also defines the query scope of this request. It means that all the semantic-related resources (e.g., <semanticDescriptor> resources) under <Resource-1> are constituted as the IDB for this semantic query operation. Accordingly, the SPARQL statement will be executed over the aggregated RDF triples collected from those semantic-related resources.

The SE-1 also needs to decide whether semantic reasoning should be used for processing this request, which may have but not limited to the following potential ways:

* If no result can be produced by SE-1 based on the original IDB, SE-1 may decide to further leverage semantic reasoning.
* If User-1 is a preferred user (e.g. SE-1 needs to provide high-quality discovery result to User-1), SE-1 may decide to further leverage semantic reasoning.
* SE-1 can also be configured such that as long as it finds certain ontologies or the interested terms/concepts/properties adopted in the RDF triples included in the original IDB or in the SPARQL statement, SE-1 may decide to further leverage semantic reasoning.

As a result, if SE-1 decides to leverage semantic reasoning based on the above approaches, it will further contact SR-1 (Note that, the SE-1 and SR-1 may be hosted on the same CSE or can also be hosted by different CSEs).

Step 4: SE-1 sends a request to SR-1 in order to initiate a semantic reasoning operation. SE-1 may also indicate the Fact Set (FS) and Reasoning Rule Set (RS) to be used for reasoning. For example, the original IDB determined during Step 3 is the initial inputFS for the reasoning operation to be done by SR-1.

Step 5: In addition to the inputs provided by SE-1, optionally SR-1 may also decide whether additional FS and/or RS can be used. Then, SR-1 will collect all the needed FSs and RSs.

Step 6: SR-1 executes a semantic reasoning operation and yields the inferred facts (denoted as inferredFS-1).

Step 7: SR-1 sends back the inferredFS-1 to SE-1.

Step 8: SE-1 integrates the inferredFS-1 with the original IDB to generate an “Augmented IDB”, and executes the SPARQL statement over the augmented IDB to yield the corresponding result for the intended semantic operation required by User-1.

Step 9: When completing the processing, SE-1 sends back the processing result to User-1.



Figure 8.7.6.1-1: Procedure for Initiating a Semantic Operation with Reasoning Support

#### 8.7.6.2 Examples usage of procedure for initiating a semantic operation with reasoning support

In this clause, a real example shows how the procedure introduced in Figure 8.7.6.1-1 can be used. In particular, a hospital facility surveillance use case as illustrated in clause 8.7.2 is reused here.

In the hospital facility surveillance use case, due to the different usages of rooms, the hospital has defined several “Management Zones (MZ)” and each MZ likely comprises multiple rooms. For example, MZ-1 includes all the rooms that store blood testing samples.

Now, User-1 intends to retrieve real-time images from the rooms “belonging to a specific management zone (e.g., MZ-1)”. For this purpose, User-1 needs to first discover those related cameras using oneM2M semantic resource discovery mechanism. As a result, the following steps will be conducted (the steps shown below are as same as the steps shown in Figure 8.7.6.1-1):

Step 1: User-1 intends to initiate a semantic resource discovery operation. For example, User-1 is looking for cameras monitoring the rooms belonging to a specific management zone (e.g., MZ-1). The SPARQL query statement in this semantic resource discovery request can be written as follows:

SELECT ?device

WHERE {

?device is-a ex:Camera

?device monitors-room-in MZ-1

}

Step 2: User-1 sends a request to SE-1 in order to initiate an intended semantic resource discovery operation.

Step 3: Based on the request from User-1, SE-1 starts the related processing. The first step is to determine the IDB for this request, on which the SPARQL statement is to be executed.

In this example, it is assumed that now <Camera-11> is one of the candidate resources to be evaluated. When evaluating <Camera-11>, all the RDF triples stored in the <semanticDescriptor> child resource of <Camera-11> is the IDB (denoted as IDB-1). For example, IDB-1 may include the following two facts:

• Fact-1: Camera-11 is-a Camera

• Fact-2: Camera-11 is-located-in Room-232-of-Building-1

In the meantime, SE-1 is configured such that as long as it finds that building/room number and/or a specific predicate “is-located-in” appears in the RDF triples as included in the IDB, SE-1 may decide to further leverage semantic reasoning.

Step 4: SE-1 sends a request to SR-1 in order to initiate a semantic reasoning operation. SE-1 also indicates that IDB-1 is the input FS for the reasoning operation, which includes Fact-1 and Fact-2.

Step 5: In addition to the inputs provided by SE-1, SR-1 may also decide whether additional FS and/or RS can be used. In this example, SR-1 finds that there is a key word “is-located-in” in Fact-2 and a key word “is-managed-under” in the SPARQL statement, accordingly SR-1 may decide that MZ definition and room allocation knowledge may be beneficial for this semantic reasoning operation and should be utilized. In particular, this knowledge may include the following fact:

• Fact-3: Room-232-of-Building-1 is-managed-under MZ-1

For the same reason, SE-1 also decides the following reasoning result can be utilized:

• Rule-1: IF A is-located-in B && B is-managed-under C, THEN A monitors-room-in C

Step 6: SR-1 executes a semantic reasoning operation and yields the reasoning result. In particular, by using Fact-2, Fact-3 along with Rule-1, the following inferred fact can be obtained:

• Inferred Fact-1: Camera-11 monitors-room-in MZ-1

Step 7: SR-1 sends back the Inferred Fact-1 to SE-1.

Step 8: SE-1 integrates Inferred Fact-1 with the original IDB (i.e., Fact-1 and Fact-2) to generate an augmented IDB, and executes the SPARQL statement over the augmented IDB to yield the corresponding discovery result. In this example, there will be a match when executing the SPARQL statement over the augmented IDB (since now the Inferred Fact-1 can match the pattern “?device is-managed-under MZ-1” in the SPARQL statement and Fact-1 can match the pattern “?device is-a ex:Camera”) and therefore the URI of <Camera-11> will be included in the discovery result). After that, SE-1 completes the evaluation of <Camera-11> and will continue to evaluate the next candidate resource if exists.

Step 9: When completing the processing, SE-1 sends back the processing result to User-1. In this example, due to the utilization of semantic reasoning, the URI of <Camera-11> is included in the discovery result and sent back to User-1.

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