분산된 IoT 플랫폼 환경에서 시맨틱 데이터 검색을 수행하는 방법

정승명*, 김성윤, Sunil Kumar
한국전자기술연구원
sm.jeong@keti.re.kr*, seongyun.kim@keti.re.kr, sunil75umar@keti.re.kr

A Method on Semantic Data Discovery for Distributed IoT Platforms

SeungMyeong Jeong*, Seongyun Kim, Sunil Kumar
Korea Electronics Technology Institute*

요 약

본 논문은 IoT 플랫폼 표준인 oneM2M 표준 기술을 확장하여 분산된 IoT 플랫폼 환경에서 불특정 플랫폼을 대상으로 시맨틱 데이터를 검색하는 방법을 제안한다. 이는 종래 표준 기술에서 검색 대상 플랫폼을 특정해야만 검색을 수행할 수 있는 한계점을 보완하기 위함이며 이를 위한 플랫폼 간 시맨틱 데이터 요약 테이블 관리 방법 및 해당 테이블을 이용한 시맨틱 쿼리의 다중 플랫폼 연계 수행 방법을 제안한다. 제안하는 시맨틱 데이터 검색 기술을 활용하면 시스템 내 분산되어 있는 데이터에 대한 효율적인 검색을 수행하여 서비스에 활용 가능한 다양한 데이터를 어플리케이션 간 직접적인 이해관계 없이도 검색할 수 있다.

Ⅰ. 서 론

본 논문에서는 oneM2M 표준에서 제공하는 시맨틱 데이터 검색 기능을 확장하여 쿼리 요청자에게 알려지지 않은 복수의 oneM2M 플랫폼에서 시맨틱 쿼리를 수행할 수 있는 방안을 제시한다. 시맨틱 검색은 일반 검색에 비해 데이터 상호호환성을 보장할 수 있는 온톨로지를 활용한 정확한 의미 기반 데이터 검색을 가능하게 하는 장점이 있다[1][2].

종래 oneM2M 표준의 시맨틱 검색은 <semanticDescriptor> 리소스에 포함된 RDF (Resource Description Framework) Triple 데이터를 대상으로 SPARQL (SPARQL Protocol and RDF Query Language) 쿼리를 수행하여 이와 연관된 oneM2M 리소스 식별자를 반환한다[3]. 이 때 검색 대상은 대상 리소스 하위에 있는 <semanticDescriptor> 리소스들이거나 사전에 <group> 리소스로 설정한 분산되어 있는 <semanticDescriptor> 리소스들이다[4]. 이는 곧 특정 oneM2M 플랫폼(CSE: Common Services Entity)이나 해당 <semanticDescriptor> 리소스의 위치를 알고 있어야함을 의미한다. 일반적으로 비즈니스 관계가 있는 이해관계자 간 해당 정보를 사전에 공유하여 이러한 시맨틱 검색을 구현하는 것이 가능하지만 그렇지 않은 경우, 특히 해당 CSE 식별자를 알지 못하는 경우 시맨틱 검색 요청이 불가하다.

제안하는 향상된 시맨틱 검색 기법은 oneM2M 어플리케이션이 원하는 데이터를 잠정적으로 가지고 있는 CSE를 인지하지 못한 상태에서도 해당 CSE들에 시맨틱 쿼리를 전달하여 검색을 수행하고 취합된 결과를 수신할 수 있는 방안을 제안한다.

Ⅱ. 본론

어플리케이션이 <semanticDescriptor> 리소스의 위치를 복수의 CSE로 구성된 oneM2M 시스템 내에서 특정하지 못한 상태에서 SPARQL 쿼리를 효과적으로 수행하기 위해서는 해당 쿼리의 매칭 결과를 가질 것으로 판단하는 CSE를 선정하여 해당 CSE에서 쿼리를 수행할 수 있어야 한다. 즉, 어플리케이션은 시맨틱 검색 요청을 특정 CSE에 요청하고 해당 CSE는 검색 결과를 가지고 있을 CSE를 자신이 선정하여 시스템 내에서 검색을 수행하는 것을 의미한다. 이를 위해서 먼저 각 CSE가 가진 시맨틱 데이터 요약 정보를 CSE 간에 교환하고 각자의 시맨틱 데이터 요약 테이블을 관리하는 방안을 제시한다.

예를 들어 CSE1은 자신이 가진 <semanticDescriptor> 리소스에 대한 요약 정보를 CSE2에 제공하거나 역으로 CSE2는 CSE1이 가진 시맨틱 데이터 요약 정보를 구독하여 신규 정보를 통지받을 수 있다. 이러한 두 가지 방법은 CSE1이 CSE2의 가칭 <semanticSummaryTable>과 같은 가상 리소스에 갱신 요청을 전송하여 CSE2가 가진 요약 정보 테이블에서 CSE1의 시맨틱 데이터 요약 정보를 갱신하거나 CSE2가 CSE1의 <semanticSummaryTable> 리소스에 구독을 요청하여 CSE1의 신규 요약 정보를 통지 메시지로 수신하는 인터페이스로 정의할 수 있다. 구독/통지의 경우에는 해당하는 이벤트 유형이 <subscription> 리소스에 정의되어 있지 않으므로 신규 시맨틱 요약 정보 발생을 notificationEventType 속성에 추가하는 것을 가정한다.[5]

시맨틱 데이터 요약은 <semanticDescriptor> 리소스에 저장된 RDF Triple 인스턴스의 타입 (RDF Class 및 RDF Property [6]) 정보로서 URI (Uniform Resource Identifier) 형식을 가진다. 즉, Triple 데이터의 볼륨에 상관 없이 타입 정보를 나타냄으로써 해당 테이블은 특정 CSE가 특정 온톨로지 타입의 Triple 인스턴스를 가지고 있을 것이라는 힌트를 제공한다. 어플리케이션으로부터 불특정 CSE를 대상으로 시맨틱 검색을 요청받은 CSE는 자신의 시맨틱 요약 정보 테이블에 기록된 다른 CSE가 가지고 있는 RDF 타입 정보와 어플리케이션이 요청한 SPARQL 쿼리를 비교하여 매칭 결과를 가지고 있는 CSE를 시맨틱 쿼리를 수행할 CSE로 선정할 수 있다.

어플리케이션은 종래 시맨틱 검색과는 달리 특정되지 않은 CSE에서 시맨틱이 수행될 수 있도록 가칭 <advancedSemanticDiscovery> 가상 리소스를 타겟하여 SPARQL 쿼리를 포함한 시맨틱 검색을 요청할 수 있다.

해당 가상 리소스를 가지고 있는 CSE는 요청을 수신하고 SPARQL 쿼리를 수행할 CSE를 선정하기에 앞서 쿼리 분할을 수행할 수도 있다. 이는

본 논문에서 제안하는 시맨틱 검색은 다수의 CSE를 대상으로 할 수 있으므로 어플리케이션 입장에서는 복잡한 쿼리를 한 번에 요청하여 각 CSE가 각자 일부 쿼리에 해당하는 결과를 반환하고 취합된 결과를 한 번에 수신할 수 있는 방법을 고안하였다.

쿼리 분할은 본 논문에서는 결과를 단순 취합할 수 있는 UNION 조건절에 대해 수행하는 것을 가정한다. 예를 들어 A UNION B 조건이 SPARQL 쿼리에 포함된 경우 A 및 B의 수행 결과에 해당하는 oneM2M 리소스 식별자를 하나의 목록으로 병합하여 결과를 어플리케이션에 반환할 수 있다.

시맨틱 검색 요청을 수신한 CSE는 시맨틱 요약 정보 테이블에서 SPARQL 쿼리와 매칭되는 대상 CSE를 선정하기 위해 3가지 다른 매칭 방법을 선택할 수 있으며 이는 검색을 요청하는 어플리케이션이 지시할 수 있도록 인터페이스를 정의할 수 있다.

첫째, 온톨로지 URI 수준으로 시맨틱 검색을 수행할 CSE를 선정할 수 있다. 이는 가능한 많은 CSE에서 SPARQL 쿼리를 수행하여 가능한 많은 결과를 수신할 수 있는 장점이 있으나 그만큼 결과 수신에 시간이 소요될 수 있다. 예를 들어 CSE1이 요청 메시지를 받았을 때 자신이 가진 요약 정보 테이블에 CSE2이 "http://www.citydatahub.kr/ontology/core-v1#SmartDevice"라는 URI 정보를 가지고 있을 때 온톨로지를 가리키는 "http://www.citydatahub.kr/ontology/core-v1" 만으로 요청자의 SPARQL 쿼리와 매칭을 수행한다.

둘째, 타입 URI 값으로 CSE를 선택할 수 있다. 이는 예를 들어 상기 예에서 타입값인 "http://www.citydatahub.kr/ontology/core-v1#SmartDevice" 그대로 SPARQL과 매칭하여 시맨틱 검색을 요청할지 CSE1이 판단한다. 이 방법은 첫 번째와 비교하여 온톨로지 수준의 매칭보다는 온톨로지의 Class나 Property가 더 정교한 수준의 정보이므로 상대적으로 쿼리 결과를 반환할 수 있는 CSE를 좀 더 높은 확률로 선정할 수 있다는 장점이 있다.

그림1은 어플리케이션이 요청한 쿼리를 UNION 조건으로 분할하여 분할된 쿼리 별로 이에 매칭되는 CSE를 선정하는 과정을 예시로 나타낸다. 분할된 쿼리는 CSE1의 시맨틱 요약 정보 테이블에서 CSE 별로 가지는 URI 정보와 매칭을 수행한다.

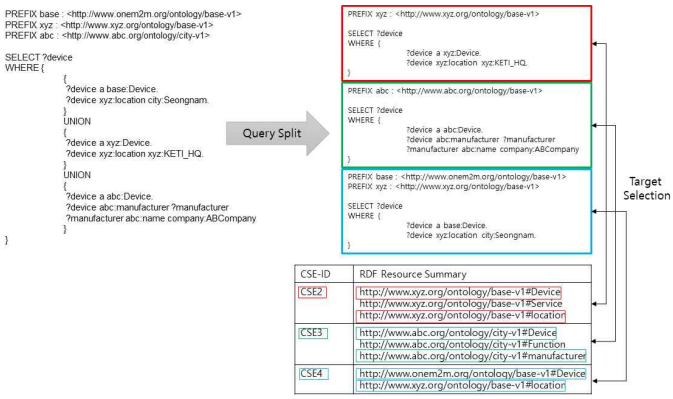

그림 1 시맨틱 쿼리 분할 및 쿼리 수행 CSE 선정 과정 예시

마지막으로 시맨틱 추론에 의해 시맨틱 검색을 수행할 CSE를 선택할 수 있다. 예를 들어 어플리케이션이 "http://www.citydatahub.kr/ontology/core-v1#Device" 타입을 SPARQL 쿼리에 포함하고 포함하고 시맨틱 추론을 활용한 시맨틱 검색을 요청하면 이를 수신한 CSE1은 CSE2가 가진 시맨틱 데이터에 사용된 온톨로지를 획득하여 추론을 통해 SmartDevice가 Device의 하위 클래스임을 확인하고 CSE2에 해당 시맨틱 쿼리를 전달하기로 결정할 수 있다. 이러한 매칭 방법은 어플리케이션이 특정 CSE뿐만 아니라 다른 CSE 가진 시맨틱 데이터의 온톨로지에 대한 정보가 충분하지 않을 때 시맨틱 검색을 수행하는 CSE를 선정할 때 유용하게 활용될 수 있다.

이렇게 선정된 CSE에 어플리케이션의 시맨틱 검색 요청을 전달한다. 앞서 쿼리 분할이 수행된 경우 분할된 쿼리로 대상 CSE를 선정하고 해당 CSE에는 선정과정에 사용한 분할된 쿼리를 전달한다. 이후 CSE1은 수신된 개별 쿼리 응답을 취합하여 어플리케이션에 전달한다.

Ⅲ. 결론

본 논문에서는 oneM2M 시스템에서 어플리케이션이 검색하고자 하는 시맨틱 데이터의 위치를 특정하지 못하는 상황에서도 여러 CSE에 분산되어 있는 시맨틱 데이터 검색을 수행할 수 있는 방안을 제시하였다. 이를 위해 각 CSE가 가지고 있는 시맨틱 데이터의 요약 정보를 서로 간에 공유하고 어플리케이션이 불특정 CSE를 대상으로한 시맨틱 검색 요청 시, 요약 정보를 참고하여 시맨틱 쿼리 결과를 반환할 수 있는 CSE를 선정하여 시스템 내에서 분산된 시맨틱 검색을 수행한다.

제안하는 방법은 쿼리를 분할하는 조건이 현재 UNION에 한정되어 있고 시맨틱 검색이 아닌 시맨틱 쿼리[4]인 경우 분할 쿼리의 결과를 항상 단순 취합할 수 없는 유즈케이스가 존재하여 향후 추가적인 연구가 필요한 것으로 보인다.

어플리케이션이 사전에 인지하고 있는 플랫폼(CSE)뿐만 아니라 불특정 플랫폼의 데이터를 온톨로지를 활용한 정확한 의미 기반의 검색을 수행할 수 있는 본 제안은 서비스 제공에 필요한 데이터 확보를 용이하게 할 수 있을 것으로 기대된다.

ACKNOWLEDGMENT

This work is supported by the Korea Agency for Infrastructure Technology Advancement(KAIA) grant funded by the Ministry of Land, Infrastructure and Transport(Grant 21DEAP-B158906-02).

참 고 문 헌

[1] Alaya, Mahdi Ben, et al. "Toward semantic interoperability in oneM2M architecture." IEEE Communications Magazine 53.12 (2015): 35-41.

[2] Kovacs, Erno, et al. "Standards-based worldwide semantic interoperability for IoT." IEEE Communications Magazine 54.12 (2016): 40-46.

[3] Gilani, Komal, et al. "Semantic enablement in IoT service layers–Standard progress and challenges." IEEE Internet Computing 22.4 (2018): 56-63.

[4] oneM2M TS-0034, "Semantics Support v4.2.0," 2020, (https://www.onem2m.org/technical/published-specifications/release-4).

[5] oneM2M TS-0001, "Functional Architecture v4.10.1," 2020, (https://www.onem2m.org/technical/published-specifications/release-4).

[6] W3C, "RDF Schema 1.1", February 2014, (https://www.w3.org/TR/rdf-schema).