

사물웹(WoT) 표준 동향



정창진 (주)게임빈 부사장

1. 머리말

인터넷을 기반으로 우리 주변의 모든 기기와 사물들을 연동하여 더 나은 삶을 가능하게 하는 사물인터넷 기술은 미래의 새로운 성장 동력으로 국내 기업뿐만 아니라 많은 글로벌 기업들을 중심으로 활발하게 개발되고 있다. 이에 따라 표준 활동 역시 빠르게 진행되고 있는데, oneM2M, OCF, AllSeen Alliance 등의 다양한 기구들을 통해 진행되어 오던 것이 최근 OCF를 중심으로 통합되었다. 2014년 7월 삼성전자와 인텔의 주도로 구성된 OIC(Open Internet Consortium)는 이후 2016년 1월에 UPnP(Universal Plug and Play)와 통합되어 OCF(Open Connectivity Foundation)로 명칭을 바꾸고 더 큰 조직으로 거듭나게 된다.

이와 별개로 쉐일과 엘지전자 주도의 AllSeen Alliance가 조직되었으나 이 또한 2016년 10월에 OCF와 통합되어 현재는 OCF가 사물인터넷 표준을 진행하는 가장 큰 단체로 자리매김하게 되었다. 다양한 기구들에 의해 분산되어 진행되던 사물인터넷 표준이 OCF를 중심으로 통합되는 상황이지만 OCF

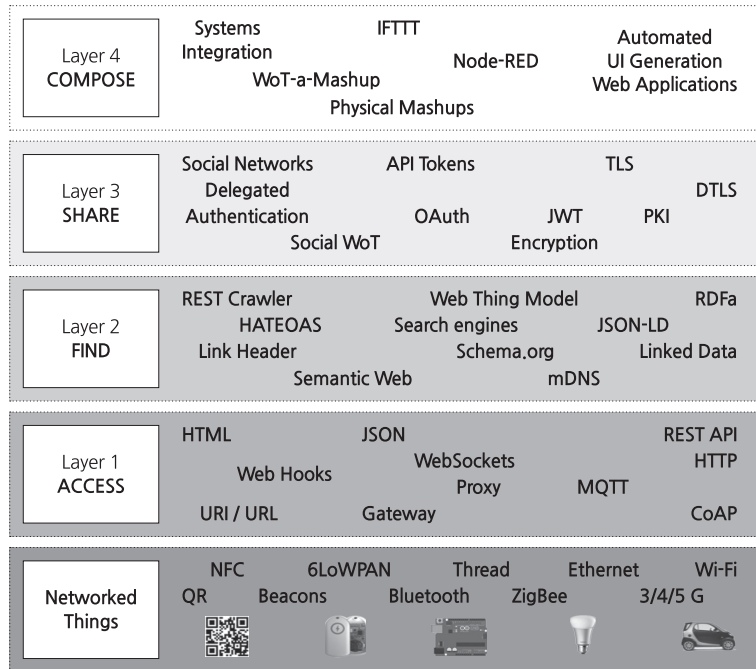
표준이 발표되기 이전에 이미 기존의 사물인터넷 플랫폼들이 비표준 기반으로 구축되고 서비스되기 시작했으며 이러한 비표준 기반의 서비스들은 사물인터넷 시장에 진입하고자 하는 중소기업들에게는 상당한 부담을 안겨줄 수밖에 없게 되었다.

간단하고 단순한 사물 인터넷 기기라고 하더라도 특정 플랫폼에 서비스되기 위해서는 해당 플랫폼을 개발한 개발업체의 사양을 따라야 하는데 각각의 플랫폼 업체들에 맞는 모든 사양을 사물기기에 탑재하는 것은 큰 부담이 아닐 수 없다. 이러한 파편화된 플랫폼 간의 연동 문제를 웹 기술을 통해 해결해 보고자 하는 움직임이 W3C를 중심으로 나타나기 시작했는데, 여기서 등장한 표준이 사물웹(Web of Things) 표준이다.

2. 사물웹(Web of Things)이란?

2.1 사물인터넷(Internet of Things)과 사물웹(Web of Things)의 차이

사물 인터넷이 NFC, Zigbee, Beacons, Bluetooth, TCP/IP 등의 다양한 네트워크 자원을 활용하여 통



※ 출처: www.webofthings.io

[그림 1] 사물웹 아키텍처

신하는 반면 사물웹은 HTTP, WebSocket 등의 웹 기반 통신을 중심으로 활용한다. 또 [그림 1]에서 보는 바와 같이 이러한 사물웹 아키텍처는 약 4개의 층위로 구성되어 있는데 먼저 사물기기에 접속하기 위한 Access 층위, 사물기기의 속성이나 기능이 어떤 것인지를 확인하기 위한 Find 층위, 사물 기기의 기능과 데이터를 연동하기 위한 SHARE 층위, 사물 기기의 데이터나 기능을 표시하고 UI를 제공하기 위한 Compose 층위가 그것이다.

이와 같이 사물인터넷 기술을 구현하기 위해 REST 등의 웹기술을 주요하게 활용하고 이를 통해 다른 서비스들과 연동할 수 있도록 구현하는 것을 ‘사물웹’ 기술이라고 통칭할 수 있다.

2.2 사물웹 표준 진행 경과

OCF가 사물인터넷의 주요한 표준 기구로 떠오르

기 전인 2015년 4월 W3C에서 먼저 사물웹 관심그룹이 구성되었다. 이들의 주요한 관심은 사물 인터넷 플랫폼들이 각 기업만의 비표준 기반으로 개발되어 파편화된 사물인터넷 서비스간의 연동 문제점을 해결하는 것이었다. 이후 W3C 내의 합의를 통해 2016년 12월에 사물웹 표준을 진행하기 위한 정식 워킹 그룹이 구성되었다.

2.1에서 사물웹은 웹 기술을 통해 사물간의 통신을 구현하는 것이라고 하였는데 여기서 한가지 주의 깊게 볼 것이 있다. 2016년 OCF에서도 웹 기반 통신 방식인 HTTP 및 REST 기술을 사물인터넷 연동을 위한 표준 규약으로 채택하였다는 것이다. OCF에서도 웹 기술을 활용하여 이미 사물 인터넷 표준을 구성하기 시작했는데 W3C 사물웹 표준이 과연 무슨 의미가 있겠는가 하는 질문이 당연히 따라 나올 수 있다.

W3C 사물웹 표준의 기본적인 방향은 '웹 기술을 이용한 사물웹 표준의 구성'이 아니라 처음부터 '사물들간의 연동'에 초점이 맞춰 있었다는 점이 중요하다.

예를 들어, OCF 표준이 존재한다고 하더라도 이미 기존에 개발되어 있던 비표준 기반의 사물인터넷 서비스들 간에 연동은 불가능하다. 이러한 연동 문제를 해결하기 위해 몇 가지 기존의 웹 표준 기술들을 활용할 것을 제안하는데 그 중에 가장 핵심이 되는 것이 JSON LD(Javascript Object Notation Linked Data)이다.

이기종 플랫폼 간의 연동을 위해 사물기기의 속성 및 기능에 대한 메타 데이터 정의를 위해 JSON LD 표준을 활용하도록 하고 이를 중심으로 아키텍처를 구성하는 것이 W3C 사물웹 표준의 핵심이라고 할 수 있다.

3. W3C의 사물웹 표준화 현황

3.1 JSON LD

JSON LD(JSON-based Serialization for Linked Data)는 2014년에 서비스 간의 데이터 연동을 하기 위해 제정된 표준이다. 팀버너스리에 의해 웹이 개발될 당시 웹의 정신은 공공성과 개방성이었으나 어느 순간 각각의 데이터들은 서비스 제공자들에 의해 통제되기 시작했다.

예를 들어, A라는 웹사이트는 사용자 로그인을 위한 아이디로 이메일을 사용하고 B라는 웹사이트는 특정 문자열을 사용한다. 따라서 A, B 두 서비스는 서로 정의하는 아이디의 의미가 달라 데이터를 연동할 수가 없다. 그리고 사용자가 생성한 데이터 역시 각각의 서비스 제공자의 자체 표준을 따라 보관되고 통신하기 때문에 다른 서비스에서는 해당 데이터를 가져올 수가 없다. 이러한 연동 문제를 해결하기 위

해 각각의 데이터가 어떤 의미를 지니고 어떤 속성을 갖는지 메타데이터를 정의하고 해당 메타데이터를 URI 기반으로 다른 서비스에서도 참조할 수 있도록 하여 해당 데이터의 의미가 공유되도록 한 것이 JSON LD이다.

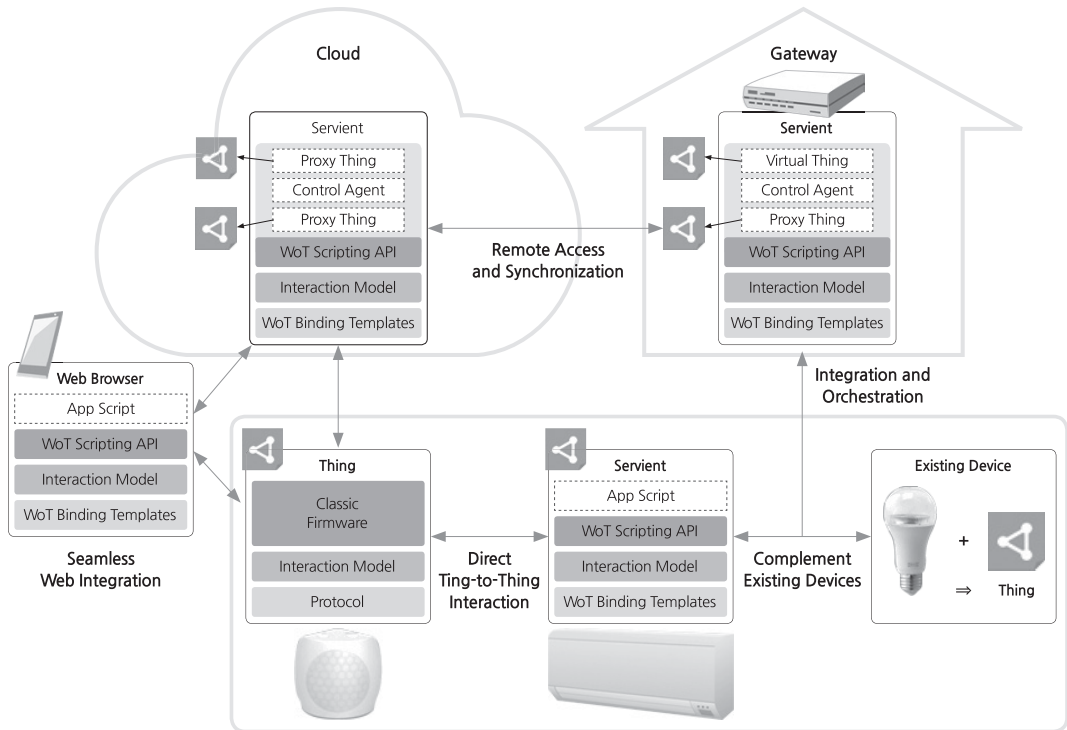
아래의 샘플 코드에서 볼 수 있듯이 기본적인 구조는 JSON 데이터 형식과 크게 다르지 않지만 JSON 내의 데이터가 무엇을 의미하는지에 대한 정의와 속성 등에 대한 메타 데이터가 어디에 정의되어 있는지 '@'이라는 기호를 통해 나타낸다.

<표 1> JSON LD 데이터 형식 예제

```
{
  "@context": {
    {
      "name": "http://schema.org/name",
      "image": {
        "@id": "http://schema.org/image",
        "@type": "@id"
      },
      "homepage": {
        "@id": "http://schema.org/url",
        "@type": "@id"
      }
    }
  }
}
```

서두에 언급하였다시피 W3C의 사물웹 표준은 어떤 특정한 사물인터넷 서비스 표준을 별도로 만드는 것이 아니라 기존의 사물웹 서비스들간의 데이터 연동에 가장 큰 초점이 맞춰져 있다. 이를 위해 기존에 표준으로 완료되어 있는 JSON LD를 기본 기술로 활용한다.

W3C 사물웹 표준이 JSON LD 표준을 기본표준 기술로 채택 하면서 JSON LD 내에 몇가지 보완해야



※ 출처: <https://www.w3.org/TR/wot-architecture/>

[그림 2] 사물웹 아키텍처

할 사항들이 드러났는데 현재 이를 위해 JSON LD 워킹그룹이 다시 구성되어 JSON LD 1.1 표준 작업이 활발하게 진행 중이다.

3.2 WoT Architecture

WoT 아키텍처는 사물웹 표준을 위한 기본적인 구조와 유스케이스들을 설명하고 사물 웹을 실제 구현하기 위한 요구사항, 사물인터넷 플랫폼 간의 연동을 위한 기본적인 개념 설명, 사물장치 및 소프트웨어에 대한 가이드 라인을 제시한다.

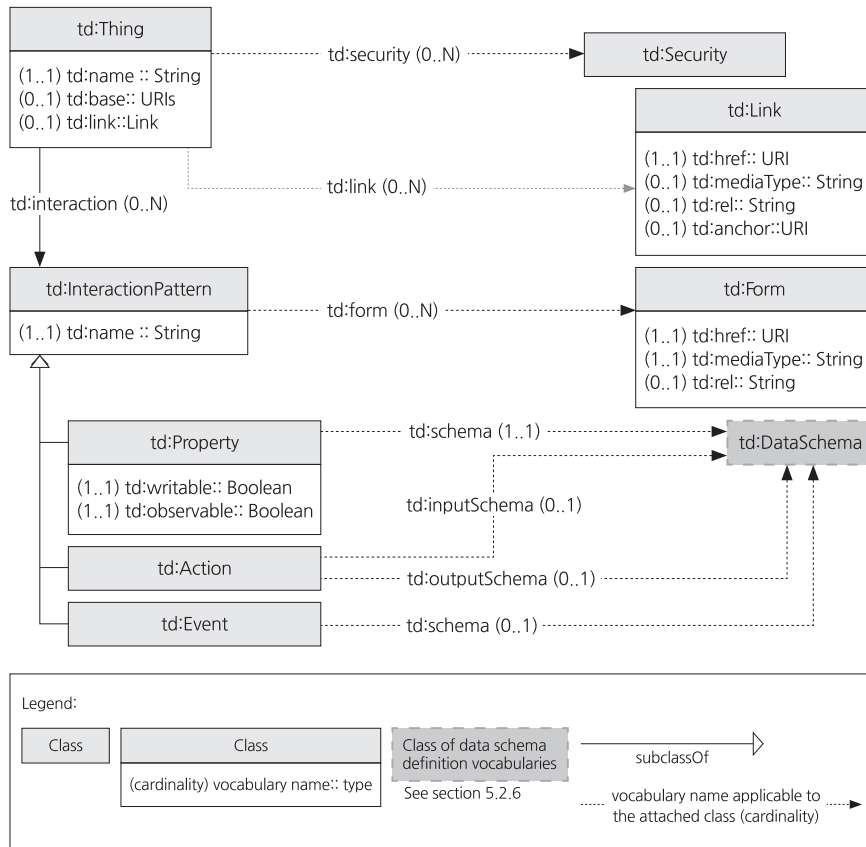
사물웹 표준을 구현하기 위해 필요한 사물의 메타데이터를 정의하기 위한 Thing Description 표준 및 사물의 속성과 기능을 발견하고 해당 사물에서 생성되는 데이터에 접근하고 송신하기 위한 Scripting APIs에 대하여 소개한다.

3.3 WoT Thing Description

WoT Thing Description 표준은 W3C WoT 표준의 가장 핵심이 되는 표준이다. WoT 표준 자체가 별도의 사물인터넷 표준을 지향하는 것이 아니고 플랫폼 간의 연동에 초점이 맞춰져 있으므로 WoT 표준은 사물간의 통신을 위한 별도의 규약을 만들지 않는다. 대신 사물이 어떤 속성을 지니고 있고 여기서 발생하는 데이터가 어떤 종류의 것인지 이 데이터를 서로 다른 플랫폼에서 이해할 수 있도록 해당 사물에 대한 메타 데이터를 정의한다. 메타 데이터는 JSON 기반으로 정의되고 해당 메타 데이터의 의미는 URI 기반으로 JSON LD 형태로 제공된다.

3.4 WoT Scripting APIs

WoT Scripting APIs 표준은 사물기기를 발견하고



※ 출처 <https://www.w3.org/TR/wot-thing-description/>

[그림 3] Thing Description Core Model

해당 기기를 플랫폼에 등록하고 사물기기에서 발생하는 이벤트나 데이터를 제어하기 위한 스크립트들을 정의한 표준이다. 스크립트는 발견(discover), 소비(consume), 등록(register) 등의 기초적인 사물기기 객체에 대한 메소드와 addProperty, addEvent, removeEvent 등의 사물기기를 제어하고 활용하기 위한 메소드들로 구성되어 있다.

또 사물기기에서 발생하는 데이터나 사물기기 자체의 속성을 구성하는 데이터 구조체와 다른 사물기기와의 연동을 위한 인터페이스 표준을 포함하고 있다. 그러나 이러한 스크립트들은 이미 기존 플랫폼이나 사물기기에서 사용하고 있는 자체 표준을 따

라 구현될 수 있다. 따라서 WoT 워킹 그룹에서는 Scripting API 표준은 추가적으로 필요에 의해 개발할 수는 있지만 필수적인 것은 아니라고 밝히고 있다.

3.5 W3C 사물웹 표준 개발 현황

2015년 WoT 관심그룹을 구성하고 WoT에 대한 의견을 정리한 이후 2016년 12월에 정식으로 워킹 그룹을 구성하여 현재 총 6개의 주요 산출물을 작성하였다. 이중에서 공식 표준으로 진행하고 있는 것은 WoT Architecture, WoT Thing Description, WoT Scripting API이다. 2018년 말까지 Candidate Recommendation을 정의하는 것을 목표로 하고

<표 2> W3C WoT 표준 문서 개발 현황

구분	문서	초안	CR 목표
Normative Deliverables	WoT Architecture	2017년 9월	2018년 말
	WoT Thing Description	2017년 9월	2018년 말
	WoT Scripting APIs	2017년 9월	2018년 말
Informative Deliverables	WoT Binding Templates	2018년 4월	2018년 말
	WoT Security Privacy Considerations	2017년 12월	2018년 말
	WoT Test Cases	2018년 3월	2018년 말

있지만 현재 JSON LD 1.1 표준과의 연동 문제, 사물 메타데이터 정의와 관련한 몇 가지 이슈가 있어 2019년 정도에 마무리 될 것으로 예측된다.

4. 사물웹 표준의 응용과 활용


4.1 소셜 서비스와 사물웹 표준의 연동

WoT 표준은 하나의 별개 사물인터넷 표준이 아니고 사물인터넷 플랫폼 간의 연동을 그 목적으로 하고 있으며 이를 위해 JSON LD 등의 기존 웹기술을 재활용한다고 밝힌바 있다. 그리고 JSON 기반의 REST 웹 표준을 활용할 경우 사물기기에서 발생하는 다양한 데이터들은 또 다른 웹 서비스들과 연동하여 다양한 형태의 추가적인 서비스가 가능하다. 예를 들면, 소셜웹 플랫폼과 연계하여 사물기기가 하나의 의인화 된 형태의 객체로 동작하도록 할 수도 있다. 홈 네트워크와 연동하여 집 주인이 나는 7시 경에 집에 들어갈 것이라고 홈 서버에 전달하면 홈 서버는 이에 맞춰 보일러 기기에 7시에 맞춰 보일러를 가동시키도록 명령하고 가습기도 이 시간에 맞춰 작동한다. 만약 가습기가 고장이 나거나 물을 더 채워야 하는 상황에 될 경우 가습기는 집 주인의 소셜 서비스에 물을 채워 달라는 포스팅을 전달 할 수도 있다.

사물기기와 인간 간에 딱딱한 기능적인 교류가 아니라 인간과 사물간의 경계가 허물어지는 다소 유연하고 재미있는 서비스가 될 수 있는 것이다. OMA에서는 이와 같은 기능을 위해 SNeW in WoT 라는 이름으로 표준이 마무리되기도 했다.

5. 맺음말

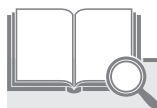
사물인터넷 표준이 등장하기 이전에 이미 해외 뿐 아니라 국내에서도 상당수의 사물인터넷 플랫폼이 상용서비스를 시작했다. 각각의 사물 인터넷 플랫폼은 각기 플랫폼 제공회사의 자체 표준을 활용하여 제작되었으므로 사물기기를 제조하는 제조사가 해당 플랫폼과 연동하기 위해서는 각 플랫폼 규격에 맞게 별도로 제조해야 한다. 하나의 플랫폼으로도 수익을 창출할 수 있는 대기업의 경우는 큰 문제가 없겠지만 단위 사물기기를 제조하는 중소기업의 경우 각각의 플랫폼에 맞게 일일이 연동규격을 개발하는 것은 쉬운 일이 아니다. 현재 OCF를 중심으로 국제 표준 움직임이 활발하게 이루어지고 있고 이는 매우 주목할 만한 일이지만 기존의 모든 사물 인터넷 플랫폼이거나 사물기기들이 OCF 표준에 맞게 자신들의 플랫폼을 다시 구현하는 것도 쉽지 않은 일이다. 이와 같은 상황에서 WoT 표준은 기존의 사

물인터넷 시장의 파편화를 극복하고 웹 기반으로 연
동 가능한 공개 사물 인터넷 기술을 위한 중요한 모
델을 제시할 수 있을 것으로 보인다. 

※ 본 연구는 정보통신방송표준개발 사업으로 수행됨[소셜웹 표준
및 소셜웹 응용 서비스 표준]

[참고문헌]

- [1] <https://www.w3.org/WoT/WG/>
- [2] <https://webofthings.org/standards/>
- [3] <https://webofthings.org/2017/04/08/what-is-the-web-of-things/>
- [4] <https://w3c.github.io/wot-architecture/>
- [5] <https://w3c.github.io/wot-scripting-api/>
- [6] <https://www.w3.org/TR/wot-thing-description/>
- [7] <https://www.w3.org/WoT/>
- [8] <https://www.w3.org/WoT/WG/>
- [9] <https://www.w3.org/2016/07/wot-ig-charter.html>



정보통신 용어 사전

<http://terms.tta.or.kr>



액티브엑스 ActiveX

마이크로소프트 윈도우(Microsoft Windows) 환경에서 응용 프로그램이나 웹 브라우저가 인터넷을 통해 추가 기능을 다운로드 및 실행할 수 있도록 지원하는 소프트웨어 프레임워크.

마이크로소프트사에서 1996년 응용 프로그램과 웹 페이지 개발에 재사용할 수 있는 소프트웨어 컴포넌트를 정의하기 위해 만들었다. 1996년 출시된 인터넷 익스플로러(Internet Explorer) 브라우저 3.0 버전부터 HTML 콘텐츠 내에 액티브엑스(ActiveX) 기능을 지원하였다.

액티브엑스는 일반적인 윈도우 응용 프로그램 형태로 개발하여 웹 브라우저 내에서 실행하는 방식으로 웹 브라우저의 기본 기능 외에 원하는 기능을 제한 없이 구현하여 웹 페이지에서 활용할 수 있다는 장점이 있다. 대표적인 예로 은행을 포함한 금융권 서비스나 결제 서비스 등에서 활용되고 있는 공인인증서이다.

액티브엑스는 인터넷 익스플로러 브라우저상에서 실행되지만 브라우저와는 별개의 독립적인 응용 프로그램이기 때문에 보안상의 잠재적 위험이 있어 사용 시 많은 주의를 필요로 한다. 또한 비표준 기술로 크롬(chrome), 사파리(Safari), 파이어폭스(Firefox) 등 타사 웹 브라우저에서 지원하지 않는다. 안드로이드, 아이폰(iOS) 등 모든 모바일 환경에서도 액티브엑스를 지원하지 않아 스마트폰에서 사용할 수 없다. 이러한 보안상 위험, 호환성 문제 등으로 액티브엑스 퇴출 논란이 야기되고 있다.

현재 윈도우 10(Windows 10) 환경에서 인터넷 익스플로러 11은 액티브엑스를 지원하지만 최신 마이크로소프트 에지(Microsoft Edge) 브라우저는 지원하지 않는다.