



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

사물인터넷 탐색을 위한 시맨틱 정보
연동 연구

연세대학교 공학대학원

산업정보경영 전공

정 석 원

사물인터넷 탐색을 위한 시맨틱 정보 연동 연구

지도교수 김 우 주

이 논문을 석사학위 논문으로 제출함

2016년 6월 10일


연세대학교 공학대학원


산업정보경영 전공

정 석 원

정석원의 석사학위논문을 인준함

심사위원 김 우 주 

심사위원 모 정 훈 

심사위원 김 성 수 

연세대학교 공학대학원

2016년 6월 13일

감사의 글

불현듯 시간이 언제 이렇게 빨랐나 할 정도로, 짧지도 길지도 않았던 시간이 지나 졸업을 앞두고 있습니다. 이 2년 반 동안의 생활과 경험은 회사 생활에 지친 저에게 큰 활력소가 되었습니다. 앞으로 제가 사랑하는 일을 할 뒀에 큰 밑바탕이 될 것을 믿어 의심치 않는 소중한 대학원 생활과의 만남이었습니다.

2년 반 전에 새로운 포부를 안고서 학교에 입학하여 저에게 은사의 참 모습을 보여주시고, 아낌없는 격려와 지도를 해주신 김 우주 교수님께 진심으로 감사 드립니다. 또한, 저의 논문 심사를 맡아주신 모정훈 교수님과 조영상 교수님께 깊은 감사를 드립니다. 그리고 따뜻한 관심으로 돌보아 주신 정봉주 교수님께도 깊은 감사를 드립니다.

항상 긍정적으로 노력하며 학교 생활을 지냈지만, 아주 가끔은 힘든 현실과 부딪힐 때, 곁에서 공감아 가는 얘기들과 진심 어린 조언을 해 주던 우리 동기들에게도 늘 감사합니다. 다양한 연령대가 모인 우리 동기들과 뜻깊은 추억을 만들게 되어 고맙습니다. 특히 3학기 때에 이곳 저곳에서 뚜렷하게 방향을 잡지 못 하고 있을 때, 인간적으로, 사업적으로 돌파구를 만들어준 두 친구들에게도 고마움을 전합니다.

회사 생활하며 공부한다고 할 때, 누구랄 것도 없이 제게 축하한다는 말과 함께 대학원 생활 동안 많은 관심을 가져주고 실질적으로 많이 도와준 회사 동료들에게도 감사한 말을 전하고 싶습니다.

제가 2년반동안 좀 더 공부한다고 말씀 드렸을 때, 그 누구보다도 기뻐해주신 아버님 어머님께 깊은 감사 드립니다. 사람은 항상 배워야 한다는 그 말씀 잊지 않겠습니다. 우리 모든 형제 자매 분들도 감사합니다. 큰 동서, 큰 처형, 둘째 처형, 큰 처제, 작은 동서, 작은 처제에게도 감사 드립니다.

끝으로, 지금은 하늘나라에 계신 부모님께 감사하다는 마음을 꼭 전합니다. 또한, 어릴 때부터 남다른 우애로 함께 자란 누님, 형, 쌍둥이 동생 및 인생의 동반자인 자형, 형수님, 제수씨에게도 감사하다는 말을 하고 싶습니다.

이 글을 적고 있는 순간에도, 서재 바깥으로부터 웃음 소리가 환하게 들리는 사랑하는 아내와 덩달아 떠들썩하게 웃고 있는 우리 아이들. 그 동안 주말에도 같이 못 놀아 줘 미안한 마음이 큼니다. 하지만 앞으로는 아내와 우리 아이들과 더 많은, 즐거운 시간을 보내도록 노력하겠습니다. 실질적으로 대학원 과정을 해 보겠다고 했을 때, 가장 힘들었었을 텐데 늘 격려해주고 힘든 내색 한 번 없이 도와준 사랑하는 아내 임은혜에게 이 글을 드립니다. 그리고, 사랑하는 딸 하은이, 사랑하는 아들 하준이... 잘 자라줘서 너무 고맙고, 앞으로는 많은 시간을 보내자. 약속해.

대학원에 들어와 생활하고 성장하는 과정에 도움을 주신 모든 분들에게 깊은 감사 드립니다. 이 과정을 통해 습득한 모든 지식과 모든 경험들로 더욱 성장하고 성숙할 수 있었습니다. 미약하나마 이 모든 것을 주변과 함께 더불어 나누고, 살아가도록 하겠습니다.

모두가 소중합니다.

감사합니다.

목 차

그림목차	iii
표 목 차	iv
국문요약	v
제1장 서론	1
제 2 장 사물 인터넷 관련 연구	6
제 3 장 시맨틱 온톨로지 방법	16
3.1 온톨로지 기반 리소스 및 상황 모델링 기술	19
3.2 서비스 매시업 기술	20
3.3 사용자 친화적 서비스 정의 환경	22
3.4 오브젝트 데이터의 상호 운용성 증대를 위한 오브젝트 데이터 포맷	23
3.5 상황 인지를 위한 기반 온톨로지	25
3.6 서비스 시나리오 설정 예	28
3.7 온톨로지 기반의 센서 데이터 처리 단계	29
3.8 사용자 친화적 서비스 정의 환경	35
제 4 장 비교 분석	44
4.1 기본 컨텍스트 제공	44
4.2 컨텍스트 수정 및 개인화 제공	45

4.3 컨택스트 매시업 제공	46
4.4 환경 및 조건 재사용 가능	46
4.5 순차적 서비스 정의 기능 제공	47
4.6 프로그래밍 능력 불필요	47
제 5 장 결론 및 향후 연구	49
참고문헌	53
ABSTRACT	60

그림 목차

그림 1. 사물인터넷 시장 전망	2
그림 2. CoAP Proxy (Core WG, 2010)	11
그림 3. 예제 기반 온톨로지 모델 개념도	26
그림 4. 센서 데이터 처리를 위한 구현구조	30
그림 5. 온톨로지 기반의 센서 데이터 처리 단계	30
그림 6. 추상화된 오브젝트 데이터 (JSON)	31
그림 7. 추상화된 서비스 요구사항 (JSON)	32
그림 8. 기반 온톨로지	32
그림 9. Damp_jun 컨텍스트의 OWL 표현	33
그림 10. Damp 클래스의 추론 결과	34
그림 11. Jena rule을 이용한 서비스 정의	35
그림 12. 서비스 정의 과정	37
그림 13. 웹기반 서비스 정의 UI (장소 설정 단계)	38
그림 14. 장소 설정 단계가 적용된 기반 온톨로지 상태	39
그림 15. 웹기반 서비스 정의 UI (오브젝트 설정 단계)	40
그림 16. 웹기반 서비스 정의 UI (오브젝트 및 기능 선택 단계)	41
그림 17. 액추에이터 인스턴스의 정의	41
그림 18. 웹기반 서비스 정의 UI (상황 설정 단계)	43

표 목차

Table 1. 사물인터넷 관련 상설 기구들	3
Table 2. 상설 기관 현황 정리	3
Table 3. CoAP 구현 요약 정리	13
Table 4. 제안하는 오브젝트 데이터 포맷	25
Table 5. 사용자 친화적 서비스 정의 환경 비교	44

국문 요약

사물 인터넷 탐색을 위한 시맨틱 정보 연동 연구

연세대학교 공학대학원
산업정보경영전공
정 석 원

어느 덧 인터넷은 개인간 서로의 컴퓨터를 연결하고 각자 원하는 정보를 주고 받으면서 우리의 삶 속에 깊숙이 자리 매김을 하였다. 길지 않은 시간이 지나 인터넷을 통한 기기들간의 연결은 더욱 확대되어 지금은 많은 가전 기기, 웨어러블 및 심지어는 집 안방의 전구와 각종 센서에까지 인터넷에 연결되고 있다. 이렇게 연결되어 만들어지는 세상이 바로 사물인터넷(IoT, Internet of Things) 세상인 것이다. 상호 연결되는 기기의 종류와 성능이 다양해지면서 어떻게 수많은 기기들이 서로 호환 가능하고 상호 연동시킬 것인가 하는 고민이 사물인터넷 (IoT) 성공의 열쇠이자 시작이다. 현재 이를 위해 표준협의체 및 산업체 컨소시엄 등은 표준이 되고자 노력하고 있으며, 대표적으로는 ISO/IEC JTC 1/WG10, ITU-T SG 20, oneM2M, 스레드그룹(Thread Group), 올신얼라이언스(Allseen Alliance), OIC(Open Interconnect Consortium), IIC 등이 있다. 표준협의체 및 각 산업체 컨소시엄에서는 개별 컨소시엄만의 혹은 컨소시엄끼리 연합하여 프로토콜 등의 기술을 앞다투어 발표하고 있다.

사물 인터넷 환경은 모든 사물들이 IP 기반의 인터넷에 연결되어 필요한 서비스를 제공하거나 제공한다. 사물 인터넷 기술이 발전함에 따라 다양한 사물들의 수가 전 세계적으로 늘어나게 되었다. 그에 따라 관리해야 할 사물들의 양과 통신 데이터 양이 크게 늘어나게 되어 소모되는 전력량 및 여러 가지 또한 증가 추세에 있다. 따라서 사물들과 데이터들을 효과적으로 처리하기 위하여 많은 양의 사물들과 데이터를 관리하는 방법이 연구되고 있다. 그 중 대표적인 프로토콜이 IETF의 Core WG에서 발표한 웹 프로토콜과 유사하며 기존 HTTP 표준을 개량한 경량급의 HTTP 프로토콜인 CoAP(Constrained Application Protocol)이다. CoAP는 HTTP와 비슷하지만, 모든 면에서 같은 것은 아니다. 메시지 타입 및 크기가 달라 사물 인터넷 환경에 더욱 적합하다. 또한 CoAP은 멀티캐스트를 지원하며 다수의 사물들을 한 번에 관리 및 제어하는 것이 가능하다. 그러나 각 기구에서는 저마다의 표준을 사용하여, IoT 서비스 사업자나 IoT 기기 공급업체가 각종 분야별로 구축하여, IoT 서비스가 파편화 되어 가는 상황에서 각 IoT 서비스 내의 단말, 플랫폼간 상호운영성(Interoperability) 관련 기술 개발의 필요성이 더욱 증대되고 있다고 할 수 있다.

본 논문에서는 이런 다양하고 이 기종의 사물 인터넷 환경에서 사물인터넷 간의 식별(Discovery) 및 상호운영성(Interoperability)을 위한 용이성을 제공할 방법에 대해 연구하여 방안을 제시해 보고자 한다. 사물인터넷 서비스의 탐색과 상호운영성을 높이는 방안의 하나로서, 시맨틱 IoT 서비스 플랫폼 기술에 대해 살펴 본다. 시맨틱 IoT 서비스 플랫폼 기술이란, 일반 사용자가 시간과 공간에 제약받지 않고, 자기가 원하는 센싱 정보를 언제나 어디서나 손쉽게 이용할 수 있도록, 개인 맞춤형 실시간 센싱 정보 서비스를 제공하

는 의미정보 베이스의 IoT 서비스 플랫폼을 의미한다. 이 IoT 상호운영성 (Interoperability)의 활성화는 IoT 영역에서의 서비스 촉진과 시장 활성화에 결정적인 요소가 될 수 있을 것이다.

핵심되는 말: 사물 인터넷, CoAP, 시맨틱 조합, 식별, 상호운용성

제 1 장 서론

사물인터넷(Internet of Things : IoT)이란 인터넷을 기반으로 세상의 모든 사물을 연결하여 사람과 사물, 사물과 사물간의 정보를 상호 소통하는 지능형 기술 및 서비스를 말한다. 우리 사회가 산업혁명(오프라인), 정보화 혁명(온라인)을 거쳐 모든 것이 연결되는 초 연결혁명(온-오프라인 융합)으로 변화해 감에 따라 기기, 사물, 인간, 시간, 공간, 그리고 각종 데이터 등 모든 것이 인터넷으로 서로 연결되어, 정보가 만들어지고, 필요에 따라 수집되며, 상호 공유되고, 필요에 따라 활용이 되는 사물 인터넷이 등장하였다. 사물인터넷은 환경오염 최소화, 에너지 효율성 증대, 재해재난 예방 등의 공공분야, 공장자동화, 제품 생산·유통 이력 관리 등의 산업분야 및 건강, 운동 관리, 생활 편의 등의 개인분야에서 전통적인 산업과 결합하여 생산성과 효율성을 극대화하고 기존 제품에 새로운 부가가치를 창출하고 있다.

사물인터넷은 아직 도입기 또는 성장단계의 초기에 위치한 것으로 평가되며, 사물인터넷 관련 시장은 기기 단말기 등의 하드웨어뿐만 아니라, 사물인터넷 관련 고객 혹은 산업 서비스 분야가 높은 성장세를 보일 것으로 예상된다. 선진국과 글로벌 기업들은 이러한 사물인터넷 시장을 선도하기 위해 핵심·원천기술 개발 및 서비스 활성화에 적극 노력하고 있다. 중국, 미국, 유럽 등 해외 주요국에서는 사물인터넷을 국가경쟁력의 핵심 기반으로 인식하여 정부 차원의 계획을 수립·추진하고 있으며, 구글, 애플, 오라클 등 글로벌 IT 기업들은 각 사의 핵심 역량을 바탕으로 사물인터넷 생태계를 리딩하기 위해 다양한 산업분야에서 상호 경쟁과 협력을 모색 중이다.

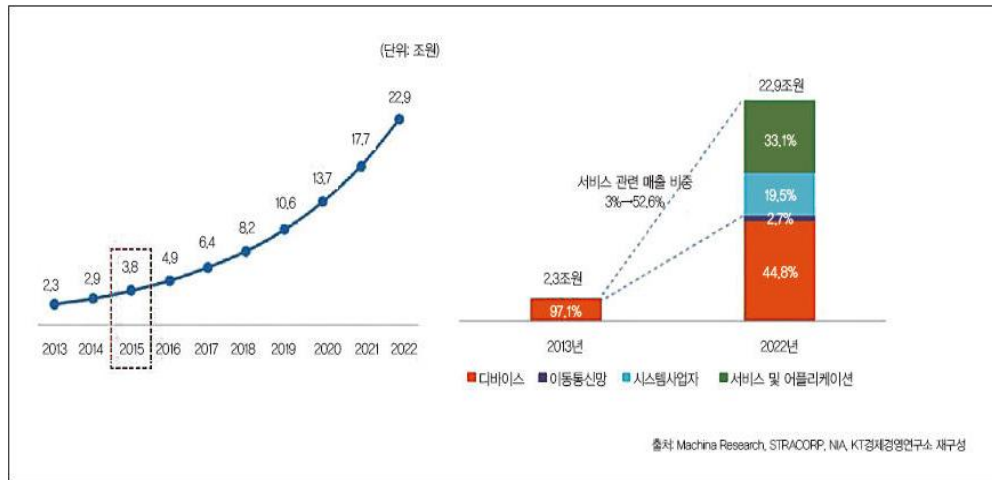


그림 1. 사물인터넷 시장 전망

(Machina Research, STRAOORP, NIA, KT 경제경영연구소 재구성)

최근 사물인터넷의 실용화를 위한 연구와 시범사업 및 산업현장 적용이 활발하게 이루어지고 있으며, ITU-T 및 ISO/IEC 등 국제표준화 기구와 한국정보통신기술협회(TTA)를 중심으로 사물인터넷 관련 국내외 표준화에 대한 논의가 진행되고 있다. 또한, 글로벌 선도 기업들도 AllSeen Alliance, Open automotive Alliance, Open Interconnect Consortium, Thread Group 등 별도의 표준화 협의체를 구성하여 회원사간 공동 연구를 통해 사물인터넷 표준을 선점하기 위해 노력하고 있다.

Table 1. 사물인터넷 관련 상설 기구들

표준기구	추진목표	주요 참여기관
Allseen Alliance	IoT의 커넥티비티 표준화로 기기간 및 플랫폼간 연결 호환성 확보	LG전자, 하이얼, 쉘컴, 마이크로소프트, 파나소닉, AT&T 등
OIC	IoT의 커넥티비티 표준화로 기기간 및 플랫폼간 연결 호환성 확보	삼성전자, 델, 인텔 등
Thread	Zigbee 기술 단점을 보완한 보안 및 저전력 기술의 표준화	삼성전자, ARM, 프리스케일, SiliconLabs 등
HomeKit	애플 생태계의 폐쇄적 IoT 기술 표준화	필립스, 하니웰, 애플, 하이얼, 텍사스인스트루먼트
IIC (Industrial Internet Consortium)	산업용 IoT 활성화를 위한 표준 개발	Inter, IBM, AT&T, 마이크로소프트, 시스코 등
스마트융합가전포럼	IoT 기반의 국내 스마트홈 가전 기술 표준화	삼성전자, LG전자, 코웨이, 모뉴엘, 경동원 등

Table 2. 상설 기관 현황 정리

항목	oneM2M	AllSeen Alliance	Thread Group	OIC
주요 설립 목적	M2M/IoT 서비스 위한 Horizontal 표준 개발	Qualcomm이 주도하는 IoT 분야 Ecosystem 확산 (현재는 비영리 컨소시엄으로 전환)	Google/Nest 주도로 스마트 홈 활성화 추진	Intel 주도로 OS, 서비스 공급자에 무관하게 상호 운용 및 여러 Vertical Market에 활용될 수 있는 단일 솔루션 확보
설립 시기	'12.7	'13.12	'14.7	'14.7
회원사	국가표준단체(ITA, ETSI, TTA, ATIS 등) 및 이통사, 벤더 등 226개사	Qualcomm, Haier, LG, Sharp, Panasonic, Microsoft 등 51개사	Nest Labs, 삼성전자, ARM, Silicon Labs 등 7개사	Atmel, Broadcom, Dell, Intel, 삼성전자, Wind River 6개사
특징	- 공통 플랫폼 사용으로 융합 서비스 가능 - Security 및 Device 관리 기능도 포함 (OMA DM 등 기존 표준을 연동해 재사용)	기 확보된 Qualcomm의 AllJoyn 솔루션 기반으로 스마트 홈 서비스를 중심으로 IoT 표준 확보 추진 중	- 무선 기술은 ZigBee를 Main으로 채택 - Nest사가 저전력 Mesh NW의 효율적 구성을 위해 최적화한 프로토콜	'14년 내 표준 정립과 스마트 홈/오피스 환경 구축 위한 솔루션 공개 목표

본 논문에서는 이런 다양하고 이 기종의 사물 인터넷 환경에서 사물인터넷 간의 식별(Discovery) 및 상호운용성(Interoperability)을 위한 용이성을 제공할 방법에 대해 연구하여 방안을 제시해 보고자 한다. IoT 상호운용성

(Interoperability)은 IoT 영역에서의 서비스 촉진과 시장 활성화에 결정적인 요소가 될 수 있을 것이다. IoT 무선 디바이스와 스마트폰 응용 앱들이 점진적으로 출시되고 있으나 대부분이 개별적이고 상호 연동 되지 않는 독자적인 플랫폼을 사용하고 있으며, 산업분야별, IoT 기기 단말 공급업체별로 파편화되어 상호연동이 되지 않고 있는 상황이다. 일반적으로 사물인터넷 서비스를 사용하고자 하는 고객은 단일 제조사와 단일 서비스업자만을 원하지 않으며, 제품의 비용과, 고객 서비스, 제품 기능에 따라서 제품을 선정하기를 원할 것이다. 따라서 다양한 사물인터넷 제품들간 또는 서비스 플랫폼간의 상호운용성(Interoperability)이 없는 경우에 사물인터넷 서비스 확산에 저해 요인이 될 수 있을 것이며, 연동을 위한 서비스 개발 기간 및 비용이 추가로 들어가는 요인이 될 것이다. 과거 통신사의 응용분야 별로 개별적 또는 독자적으로 제공되던 사물 인터넷 플랫폼이 표준화라는 단계를 통해 플랫폼의 공용화를 진행하고 있으나, 현재 사업자 중심의 B2B 및 B2C 서비스 제공에 적합하게 설계가 되었기 때문에 서비스 확산에 한계가 있을 것이다. 또한 이러한 사물인터넷 플랫폼간 또는 사물인터넷 단말간 상호운용성을 향상하는 기술을 적용한 서비스를 개발한다면 사물인터넷 서비스 발굴과 확산을 유도할 수 있으므로 관련 기술 개발에 대한 필요성이 점차 증대되고 있다. 사물인터넷 기기간, 사물인터넷 플랫폼간의 상호운용성은 사용자가 다중의 단말과 다중의 서비스를 활용하여 스마트한 환경을 만들 수 있도록 해 줄 것이다. 더 나아가 상호운용성은 거대 기업의 시장 지배력을 낮추며 다수의 사물인터넷 제품 공급업체와 서비스 사업자가 시장 진입할 수 있는 기회를 높이는 효과가 있을 것이다. 사물인터넷 상호운용성을 높이기 위한 방안으로는 제품과 서비스 사업자간의 파트너십, 개방형 표준 플랫폼 활용, 플랫폼간의 연동 기능 개발등의 다수의 방법이 있을 것이다.

본 논문에서는 사물인터넷 서비스의 탐색과 상호운영성을 높이는 방안의 하나로서, 시맨틱 IoT 서비스 플랫폼 기술에 대해 살펴 본다. 시맨틱 IoT 서비스 플랫폼 기술이란, 어떤 사용자들이 사용함에 있어서, 특정 시공간에 제한되지 않고, 언제 어디서나 본인이 원하는 센서된 정보를 쉽게 이용할 수 있도록 사용자에게 대한 맞춤형 실시간 센서 정보 서비스를 제공하는 의미정보 기반의 사물인터넷 서비스 플랫폼을 의미한다.

본 논문의 구성은 다음과 같다. 우선 2 장에서는 사물인터넷 현황 및 표준화 기술 동향 등과 관련된 선행 연구 및 프로토콜 CoAP에 대해 살펴 볼 것이다. 3 장에서는 본 논문에서 제안하는 온톨로지 시맨틱 기반의 전반적인 구조 및 그에 따른 세부사항 및 온톨로지 기반 리소스 및 상황 모델링 기술, 서비스 매시업 기술, 사용자 친화적 서비스 정의 기술에 관한 관련 연구를 분석한다. 4 장에서는 기존의 연구와 제안하는 방식의 비교 분석을 수행하고 5 장에서 결론을 맺는다.

제 2 장 사물 인터넷 관련 연구

네트워크 관련 기술의 표준화와 다양한 저 전력 무선 단말의 출현으로 사물인터넷 서비스와 시장 확산의 기반 환경이 구축되고 있으며, 새로운 개념의 다양한 사물인터넷 단말이 출시되고 있다. 사물인터넷 플랫폼 측면에서 서비스와 콘텐츠간 상호운용성을 제공할 주도적인 표준 플랫폼이 부재인 상황에서 사물인터넷 플랫폼간의 연동 기술에 대해 표준화가 활발하게 이루어지고 있다. 또한 사물인터넷 단말 측면에서 IoTivity, AllJoyn 과 같은 개방형 플랫폼을 제품에 탑재하여 다양한 사물인터넷 단말 또는 다른 제조업체의 제품간 연동을 통해 상호운용성을 제공하고 있다. 사물인터넷 서비스를 위해서는 사물인터넷 단말로부터 수집된 정보를 연동해서 관리 컨트롤 할 수 있는 효과적인 서비스를 구축하는 것이 필요한 상황이며, 더 나아가, 사물인터넷 디바이스가 단순히 원격 관리 제어 기능을 넘어서 데이터 기반의 상황 인지 기능과 멀티미디어 기능을 통한 인터랙티브 기능에 대한 요구가 커지고 있는 상황이다.

사물인터넷이 산업계의 화두로 떠오르면서, 스마트홈, 헬스 서비스를 중심으로 한 다양한 서비스가 등장하고 있다. 스마트폰을 이용하여 원격으로 스위치나 전구 등의 가전 기기를 제어하거나, 스마트워치나 웨어러블 디바이스를 통해 건강 관리 정보를 제공받음으로써, 사용자가 사물인터넷 서비스를 쉽게 접할 수 있는 계기가 되었다. 기존의 자동차, 공장, 사무실 등의 산업 부문별로 독자적인 사물인터넷 환경을 구축하고 있는 중이다. 하지만 아직 이러한 사물인터넷 서비스를 통해 실제 사물인터넷 시장 확산에 성공적

인 사례를 보여주지 못하고 있다. 이것은 사물인터넷 기능이 활용이 파편화된 솔루션으로 특정 사업 모델에 의존적이기 때문이다. 이러한 수직적 모델의 사물인터넷 서비스를 통해서 특수한 서비스를 단기적으로 제공하는 것이 가능해졌다. 수직적 모델은 사물인터넷 기기가 증가함에 따라 사용자들의 단말 기기 운용에 대한 부담을 증가시키며 또한 상호 연동을 통한 융·복합 서비스를 제공하지 못하여 추가적인 서비스 및 부가가치 창출에 한계를 가져 오고 있을 것이다. 사물인터넷 표준 플랫폼의 목적은 개방화되고 표준화된 방식으로 다양한 사물인터넷 기기를 용이하게 연동하게 하며 단말 기기 간 연동을 통해서 사용자에게 새로운 혁신적인 융·복합 서비스를 제공하는 것이다. 사용자 관점에서 이중의 기기를 설치할지라도 결합된 응용 시스템으로 통합된 운용 환경을 제공한다. 공공 측면에서 전력, 원격 검침, 의료, 가전 등 다른 산업 분야간의 융·복합 서비스를 제공하며 시스템 관리 운용을 용이하고 저비용으로 관리할 수 있을 것이다. 사물인터넷 무선센서 단말은 저전력 저 사양의 프로세서를 이용한 단말을 사용하고 있다. 사물인터넷 단말은 2020 년에 약 500 억개 정도가 될 것으로 예상하고 있으며 이중 대부분은 무선센서 단말일 것이다. 사물인터넷 서비스 망 구조 측면에서 보면 센서 단말은 빅 데이터를 처리하는 대규모 서버와 직접 연결하지 않으며, 수집된 데이터를 IoT Gateway 를 통해서 서버로 연결하는 구조를 가지게 될 것이다. 사물인터넷 생태계는 Application, Common Service, Connectivity, Device Management 계층으로 구성 되어 진다.

국가별로 수행되던 사물인터넷 표준화 활동을 통합하여 단일화된 글로벌 사물인터넷 서비스 플랫폼 표준 규격을 만들기 위해서 TTA 를 비롯한 세계 7 개 주요 표준 제정 기관(Standard Developing Organization)이 참여하여

oneM2M 파트너십 프로젝트가 2012 년 7 월에 출범하였으며, 2015 년 1 월에 릴리즈 1 표준 규격을 완료하였다. 릴리즈 1 표준 규격을 기반으로 릴리즈 2 에 추가적인 요구 사항을 반영할 예정이다. 하지만, 국제 규격인 oneM2M 상용화를 위한 규격이 미흡하고 상용화까지는 시간이 더 필요한 상황이다.

산업체 주도의 오픈 컨소시엄은 AllSeen, OIC(Open Interconnect Consortium), IIC(Industrial Internet Consortium)등이 있다. AllJoyn 은 Qualcomm 주도하에 Allseen Alliance 에서 표준화한 오픈소스 기반 IoT 플랫폼이다. Allseen Alliance 에는 Qualcomm, LG 전자, 시스코, 하이얼,MS, 모다정보통신 등 세계 100 여개 기업이 참여하고 있다. AllSeen Alliance 에 대항하는 OIC 는 Intel, 삼성 등이 주도하고 있으며, IoTivity 라는 프레임워크를 공개하였다. IoTivity 는 IoT application 을 개발하는 과정에 필요한 API 를 제공하며 AllSeen 과 동일하게 다양한 운영체제를 지원한다. API 는 RESTful 방식으로 지원한다. IoTivity 0.9 가 2014 년 12 월 배포되었으며 Release 1.0 배포 준비 중에 있다. oneM2M 에서는 IoT 시장의 활성화를 위해서 기존의 사물인터넷 관련 기술인 AllJoyn, OIC(Open Interconnect Consortium) 등과 상호연동을 추진하고 있는 상황이다. oneM2M-AllJoyn interworking 를 위해서 oneM2M 에서 2014 년 11 월 기술총회에서 Working Item 18 를 구성하였으며 oneM2M 의 resource 와 AllJoyn 의 interface 를 연동하기 위해 기능적 구조, resource mapping 정의, 절차적 명세화를 포함한 표준 규격을 제시하는 것을 목적으로 하고 있다. AllJoyn 은 method, signal, property 와 같은 interface 로 기능을 정의하고 있으며, oneM2M 은 resource 와 function 으로 기능을 표현하고 있다. Interworking 를 위해서는 oneM2M resource 와 AllJoyn interface 간의 의미적인 호환성이 필요하다. Interworking 관련 표준화는 oneM2M 18 회 기술총회까지 완료하는 것을 목표로 진행하고 있다.

위와 같은 표준화 기관에서 만든 CoAP 은 한정적인 자원을 갖고 있는 네트워크(Constrained network)에서 사용하는 전송 프로토콜이다. 6LoWPAN(IPv6 over Low power WPAN)과 같은 무선 네트워크 환경에서 사물들간 통신을 하기 위해 IPv6 패킷을 작게 분할하여 전송한다. 패킷 전송이 줄어들기 때문에 메시지 부하를 줄이게 되어 혼잡(Congestion)을 줄인다. CoAP 의 주요 목적은 저전력, 건물 자동화(Building Automation) 그리고 M2M 응용을 하기 위함이다. 그 외에 CoAP 의 특징은 다음과 같다. CoAP 의 메시지(message)는 UDP 를 이용하여 종단점(endpoint)간에 교환을 한다. CoAP 은 사물에 따라 세부적인 옵션을 지정할 수 있으며 메시지를 전송할 때 있어서 옵션 항목에 따라서 전송하게 된다. RESTful 서비스를 충족하여 기본적으로 GET, POST, PUT, DELETE 등의 HTTP 메소드를 사용한다.

CoAP 은 사물인터넷 환경에서 다수의 디바이스들과 통신할 수 있는 멀티캐스트 통신(multicast communication)을 지원한다. 멀티캐스트는 일대다수(one-to-many) 또는 다수대다수(many-to-many)로 통신을 할 수 있다. HTTP 같은 경우에는 멀티캐스트가 지원되지 않기 때문에 디바이스들과 통신하기 위해서는 단일통신인 유니캐스트(unicast) 통신으로 여러 요청 메시지를 전송하여 통신하게 된다. 이때 여러 요청 메시지와 응답 메시지를 받기 때문에 네트워크 혼잡이 생기게 된다.

반면에 CoAP 에서의 멀티캐스트 통신은 하나의 요청 메시지만 보내게 되고 Acknowledgement(ACK)가 없는 non-confirmable(NON) 메시지를 보내서 HTTP 환경에서 생기는 네트워크 혼잡을 줄일 수 있다. 다만 NON 메시지는 ACK 를 받지 않기 때문에 ACK 를 보내는 confirmable(CON) 메시

지 에 비해 신뢰성이 떨어진다.

사용자가 리소스 정보 및 옵션을 사용자 임의대로 넣을 경우에는 해당 시스템에서 용어들을 다른 형태로 이해할 수가 있다. 예를 들어, 온도 센서 또는 그룹의 정보에 temp 로 넣을 경우에 temp 의 의미를 temperature 가 아닌 temporal feature 로 이해할 가능성이 있다. 용어 이해 문제를 해결하고 각 도메인들의 정보를 공통의 용어로 정의하기 위한 연구도 진행되었다. W3C Semantic Sensor Network Incubator Group 에서는 Semantic Sensor Network(SSN) 온톨로지를 연구 및 개발하여 각종 사물들의 개념을 정의하여 용어 정리를 하였다.

SSN 온톨로지 외에 Stephan 등이 사물 인터넷 환경을 위한 도메인 모델을 이용하여 주요 개념들과 관계를 정의하여 사물 인터넷 환경을 위한 공통의 용어 및 사물의 분류 체계를 제안하였다. 하지만 도메인들이 이 공통 모델을 사용하지 않고 자체 도메인을 사용하게 된다면 이종의 도메인들의 혼용을 위한 방법이 필요하다. 각 도메인들의 사물 정보를 의미적으로 이해하는 방법으로는 온톨로지에 기반한 시맨틱 검색이 있다. Paolucci 등은 DAML-S (<http://www.daml.org/>) 에 기반하여 웹 서비스 프로파일들의 시맨틱 매칭 방법을 제안하였다. 매칭 엔진을 개발하여 사용자가 요청하는 서비스를 키워드 매칭을 통해 정교하게 탐색한다. Ruta 등에서는 CoAP 환경에서 시맨틱 정보를 이용한 사물 탐색을 수행한다. 자체 온톨로지를 이용하여 사물들을 등록하고 시맨틱 정보로 구성되어 있는 사용자 질의에 따라 탐색을 수행하게 된다. 탐색을 할 때는 시맨틱 매칭을 이용하여 질의에 가장 부합하는 센서를 탐색하여 순위를 정한 다음 가장 높은 순위의 센서를 선택하게 된다.

CoAP 의 네트워크 환경은 사용자들이 주로 쓰는 HTTP 네트워크 환경과 차이점이 있다. 기본적으로 CoAP 은 이더넷(Ethernet) 네트워크가 아닌 6LoWPAN 환경으로 TCP/IP 계층 구조가 다르며 전송하는 메시지 포맷도 다르다. 기본적인 구조가 다르기 때문에 HTTP 네트워크에서 CoAP 네트워크와 통신을 할 경우에는 다음 그림. 2 처럼 프록시 게이트웨이를 거쳐야 한다.

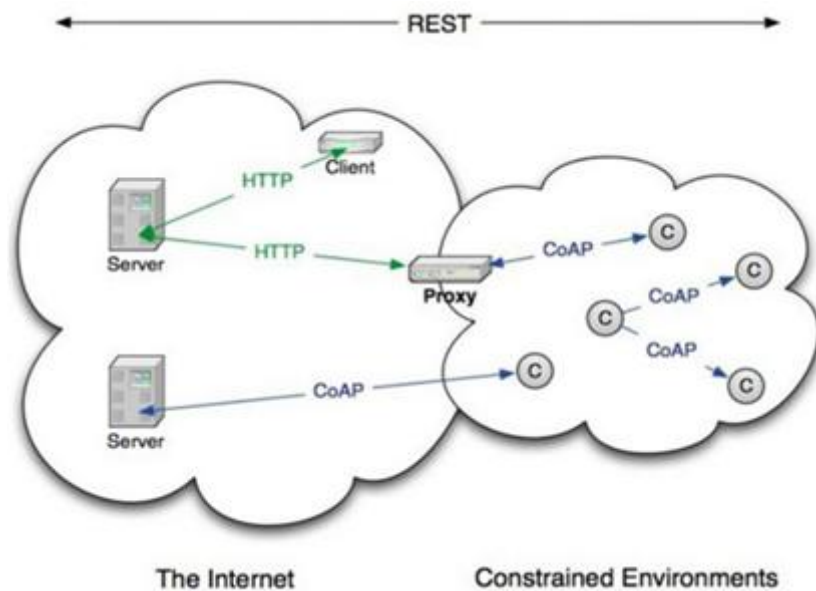


그림 2. CoAP Proxy (Core WG, 2010)

프록시 게이트웨이는 클라이언트로부터 요청된 URL 을 다른 네트워크 서버에 전달하게 되고 그에 대한 응답을 되돌려 받는다. 프록시가 필요한 상황은 여러 가지가 있다. 첫 번째 예로 스마트 폰에서 가정에 배치되어 있는 CoAP 네트워크 기반의 센서를 확인할 때, 스마트 폰의 HTTP 요청을 CoAP 으로 변환시켜주는 프록시가 필요하게 된다. 두 번째는 웹 브라우저

와 같은 애플리케이션이 CoAP 은 지원하지 않고 HTTP 만 지원할 경우에 프록시가 필요하게 된다.

Mäenpää 등에서는 무선 센서 네트워크 환경에서 CoAP 노드를 이용하여 2G 망, 3G 망과 같은 다른 네트워크와 통신할 수 있는 환경을 제안하였다. CoAP 망과 다른 네트워크 망들과 네트워킹을 하기 위하여 게이트웨이와 프록시 노드를 네트워크에 배치하였다. 게이트웨이와 프록시 노드를 다른 네트워크와 연결을 하면서 동시에 REsource Location And Discovery (RELOAD)라는 오버레이 네트워크(Overlay Network)를 이용하여 근처의 센서들을 등록하고 탐색한다고 한다. 하지만 통신방법은 P2P(peer to peer) 방식으로 다른 센서 노드에 대한 정보를 미리 알아야 하며 네트워크 자체의 부하를 해결해야 하는 문제점이 있다.

Dijk 등에서는 HTTP 의 URL 을 CoAP URL 로 사상(mapping)하는 방법을 제안하였고 프록시 기능만이 아닌 사물의 식별 및 탐색 기능과 함께 웹 API 를 포함하고 있는 프레임워크들을 개발하였다. CoAP 과 HTTP 클라이언트/서버 간의 정확한 상호교환을 하기 위하여 프록시를 실행하는 방법을 보여준다. 요청을 전달할 때 forward proxy 를 통하여 HTTP 요청을 CoAP 요청으로 변환을 하게 되고, reverse proxy 를 통하여 CoAP 응답을 HTTP 응답으로 변환하게 된다. 그 외에도 CoAP 서버 및 클라이언트 개발 이 많이 진행되었으며, 다음 Table 3. 은 CoAP 관련 개발을 정리한 표이다.

Table 3. CoAP 구현 요약 정리 [98]

이름	개발 언어	CoAP 구현 정도	구현 특징
libcoap	C	Client & Server	Observe, Blockwise Transfers
iCoAP	Objective-C	Client	Observe, Blockwise Transfers
nCoAP	Java	Client & Server	Observe
jCoAP	Java	Client & Server	Observe, Blockwise Transfers
CoAPython	Python	Client & Server & Proxy	Observe, Multicast Server discovery, CoRE Link Format parsing
txThings	Python	Client & Server	Observe, Blockwise Transfers
TinyOS CoapBlip	nesC/C	Client & Server	Observe, Blockwise Transfers
Eribium for Contiki	C	Client & Server	Observe, Blockwise Transfers
Californium	Java	Client & Server	Observe, Blockwise Transfers, DTLS
Copper	JavaScript	Client	Observe, Blockwise Transfers
Microcoap	C	Client & Server	Core, Observe, Block, RD
ETRI CoAP	C	Client & Server	Core, Observe, Block,

하지만, 프록시의 방법은 멀티캐스팅 방식으로 요청을 보낼 경우에는 문제가 없지만, 응답을 받는 것에 있어서 문제점이 존재한다. 요청을 보낼 때는 HTTP 클라이언트에서 유니캐스트 방식으로 단 하나의 요청으로 각 CoAP 서버로 멀티캐스트 요청을 보내게 된다. 이때, HTTP 클라이언트에서는 유니캐스트 요청을 보냈기 때문에 한 개의 응답 받는 것을 예상한다. 하지만, 각 CoAP 서버에서는 멀티캐스트 요청으로 보내졌기 때문에 응답을 여러 개 받게 된다. 응답을 여러 개 받게 됨으로 네트워크 혼잡이 일어나며 네트워크 부하가 생기게 되며 시간이 걸리게 된다. 애플리케이션 계층에서 응답들을 합쳐서 처리하는 방법이 가능하나 이를 해결하기 위한 방법은 아직 미비한 실정이며 CoAP 명세 에서 다루지 않고 있다.

Bovet 등은 IP 기반의 사물들이 존재하는 스마트 빌딩에서 건물의 자동화를 위한 시스템을 제안하였다. DNS 서버를 이용하여 각 구역의 건물과 건물 안의 층 그리고 방의 주소 체계를 자동으로 지정하였다. 네임 스페이스(name space)를 자동으로 지정하면서 각 구역의 건물들을 건물 별로 그룹으로 구성하였다. 멀티캐스트 방식을 사용하게 되는데 각 그룹의 업데이트 또는 탐색을 한다. 이 방식으로 인해 통신 값을 줄이게 되어 에너지 효율성을 높였다. 하지만 구체적인 그룹 조합에 대한 설명이 부족하고 도메인 모델과 온톨로지를 사용하지 않으므로 이 논문 또한 각각의 네임 스페이스에 대한 의미를 알 수가 없다.

Bovet 등은 기존의 자동화 시스템이 갖고 있는 서브 시스템 간의 이질적인 문제를 해결하기 위하여 웹 기반의 프로토콜을 제안하였다. 각기 다른 네트워크들과 인터페이스들을 통합하기 위하여 멀티 프로토콜 아키텍처를 제안하였다. 이 아키텍처는 CoAP, oBIX 등과 같은 여러 종류의 프로토콜

이 사용 가능하며, 리소스의 식별, 연결(linking), 속성 기술, REST 서비스를 지원한다. 또한 리소스들을 기술하는데 있어서 의미를 부여하기 위하여 온톨로지를 사용하였다. SSN 온톨로지를 사용하여 메타 데이터를 기술하였으며, RDF 를 이용하여 각 리소스들을 SPO (subject, predicate, object) 구조로 기술하였다. 하지만 온톨로지로 기술한 센서 정보를 구체적으로 활용하는 방법이 부족하며, 건물 자동화 시스템에서의 활용 방법도 제시되어 있지 않다.

서로 다른 웹 서버간에 상호 통신을 주고 받기 위해서는 서로 서버 간의 목적지 주소를 알고 있어야 한다. Copper(Cu)에서는 CoAP 리소스들과 상호 교환할 수 있는 웹 브라우저 를 개발하였다. 사물 인터넷을 위한 웹 브라우저에서는 사용자들이 사물과 상호 정보를 교환을 하기 위한 서비스를 제공한다. CoAP 리소스 디렉토리인 `/.well-known/core` 에 존재하는 리소스들을 브라우저 창에서 확인할 수 있으며 REST 기능들을 통하여 리소스들의 조회, 등록, 수정 및 삭제가 가능하다. 하지만 다른 도메인의 CoAP 서버 탐색이 불가능하므로 다른 CoAP 서버의 위치 정보에 대해서는 알 수가 없다. 또한 CoAP 리소스들만 탐색을 하고 CoAP 리소스 정보를 제공하기 때문에 HTTP 서버 또는 클라이언트와의 상호교환이 불가능하다. 이를 해결하기 위하여 HTTP 클라이언트에서도 CoAP 리소스들의 조회가 가능한 시스템을 제안한다.

제 3 장 시맨틱 온톨로지 방법

사물 인터넷(Internet of Things) 기술이 발전함에 따라 서로 다른 특성을 갖는 수 많은 사물이 인터넷을 통해 연결 가능하게 되었다. 하지만, 더 많은 사물들이 서로 연결되면 될수록 사물들간의 상호 운용성 문제점은 더욱 대두되고 있다. 이러한 상호 운용성 이슈들을 해결하기 위해 HTTP, JSON 과 같은 웹(Web) 기술들을 적용하는 많은 연구가 진행되고 있다. 최근의 사물 웹(Web of Things) 환경에서 우리는 다양한 사물 인터넷 플랫폼을 접하고 있다.

사물 인터넷 플랫폼은 사용자들에게 웹 상에서 보다 쉽게 수많은 양의 사물과 데이터를 관리하고, 그것들을 조합하여 새로운 서비스를 만들어 내는 기능을 제공하고 있다.

특히, 서로 다른 특성을 갖고 있는 수많은 사물들이 서로 다른 플랫폼에 배치되는 실 세계 상황에서는, 다양한 사물들과 플랫폼간의 상호 운용성 문제는 반드시 해결해야 할 중요한 문제가 되고 있다.

3 장에서는 이러한 상호 운용성 이슈를 해결하기 위해, 다양한 플랫폼에 적용될 수 있는 시맨틱 서비스 명세 온톨로지 (Semantic Service Description Ontology)를 제안한다. 해당 온톨로지는 이종의 사물들에 대한 속성과 특성들을 기술하는 것뿐만 아니라, 각각의 IoT 플랫폼이 정의하고 있는 정보들을 포괄하고 있다.

예를 들어 사용자가 지금 현재 어떤 상황에 처해 있는지를 알고자 한다면 센서에서 제공하는 정보들을 활용하여 알아내고, 추정 혹은 상태를 정확히

알 수 있을 것이다. 그러나 각 센서 장비에서 올라오는 데이터들은 각각의 장비마다 구조, 문법, 시스템, 그리고 각각이 제공하는 의미 자체가 서로 다를 것이다. 그러면 하나의 관점에서 다른 성질의 서비스들을 대상으로 사용하기에 무리가 있을 것이다. 이를 처리하기 위해서 센서에서 제공해 주는 데이터들은 컨텍스트 형태로 정보가 제공되어야 한다. 그러나 센서로부터의 알고자 하는 상황은, 그 상황을 판단하기 위해 필요한 센서들의 종류와 그에 대한 기준, 프로그래밍 능력 등이 필요하여 기존의 일반 사용자들은 손쉽게 사용하기가 어려웠다. 따라서 여기에서는 앞서 설명한 사물인터넷의 경향과 추세 등을 통해 인지되는 상황 데이터를 통해 그 상황을 파악하고 그에 적합하게 사물인터넷 서비스를 제공할 수 있게, 사용자에게 적합한 온톨로지 기반 서비스를 제공하고자 한다. 그 구성은 장비에서 제공하는 센싱 데이터를 컨텍스트 정보로 바꾸는 온톨로지 기반 시멘틱 데이터 처리 메커니즘과 사용자들이 편리하게 이용 가능하도록 하는 부분으로 구성된다.

유비쿼터스, 편재형 컴퓨팅 (pervasive computing) 등의 상황인지 환경을 실현하고 사용자 주변 환경의 정황을 인지하기 위해서는 다양한 센서가 광범위하게 배치되어야만 한다. 일반적으로 센서 네트워크는 이와 같은 환경 감시에 효과적인 해결책으로 사용되고 있다. 하지만 기존의 센서 네트워크는 서비스 공급자, 센서와 장비 제조사 및 프로토콜에 의존적인 구조를 가지고 있어 재정적, 기술적인 문제가 존재한다. 이러한 이질성은 크게 시스템, 문법, 구조, 의미적 측면의 4 가지로 구분하고 있으며 이기종 환경에서 센서데이터의 재사용이 어렵다. 이를 해결하기 위해 웹 프로토콜을 이용하는 패러다임이 대두되고 있다. IoT (Internet of Things, 사물인터넷), WoT (Web of Things)와 같은 새로운 패러다임은 인터넷 프로토콜 기반의 웹 기술을 이용하여 각각의 센서나 장치의 제어, 관리, 모니터링을 제공한다. 또

한 웹 프로토콜 표준인 HTTP (Hyper TextTransfer Protocol) 를 이용하여 센서 및 장치를 웹 환경의 자원으로 인지하고 처리할 수 있어 센서 및 장비 제조사, 서비스 제공자, 이기종의 네트워크 구조 등에 독립적인 제어가 가능하다. 이와 같은 패러다임을 통해 시스템, 문법, 구조적 측면의 이질성을 제거할 수 있으며 이는 비용적인 측면에서도 큰 장점이라고 할 수 있다. 또한 사용자의 다양한 욕구를 충족시키기 위한 맞춤형 서비스 혹은 자신의 상황 및 주변 정황 정보를 인지하는 상황인지 기반의 서비스에 관한 요구가 증가하고 있다. 이를 위해 온톨로지 기술을 이용하여 수집된 센서 정보의 의미적 이질성을 제거하고 상황을 인지하는 SSW (Semantic Sensor Web) 기술에 관한 연구가 진행되고 있다.

또한 스마트 기기의 보급이 증가하면서 기존의 획일화 된 휴대전화 환경에서 자신만의 환경을 구성하여 사용하는 스마트 환경으로 발전 하였다. 이는 기존의 휴대전화 뿐 만 아니라 스마트 TV 를 필두로 가정, 빌딩, 사무실 등의 여러 환경의 다양한 장치로 확산되고 있는 실정이다. 따라서 상황 인지형 서비스의 경우에도 일반 사용자가 자신만의 맞춤형 서비스를 직접 생성하고자 하는 요구사항이 증가할 것이다. 하지만 기존의 온톨로지를 이용한 상황인지 서비스 제공 기술은 대체적으로 특정 활용 영역의 어휘를 상세하게 작성하고 상황의 정의를 위해 다양한 정보를 입력하고 서비스 조건을 정의한 후 센서에서 생성된 데이터를 도입하여 추론을 하는 방식을 따르고 있다. 따라서 이를 기반으로 하는 서비스 매시업 및 컴포지션 기술은 전문적인 지식이 없는 일반 사용자가 서비스 요구사항, 조건, 필요한 정보 등을 정의하고 이를 생성 및 구현하기에 매우 어려운 실정이다. 왜냐하면 사용자가 서비스를 정의하기 위해서는 서비스를 제공하는 플랫폼의 전체적인 구조를 이해할 필요가 있으며 서비스의 조건이 되는 컨텍스트를 정의하기 위해

필요로 하는 관련 센서들의 종류와 각 측정값의 조건 및 기준들을 알아야 하기 때문이다.

결과적으로 쉽게 서비스를 정의할 수 있는 환경이 제공된다면 사용자가 원하는 환경에서 원하는 서비스를 쉽게 제공 받을 수 있으며 플랫폼 입장에서 보다 정확한 서비스 정의 정보를 통해 사용자의 의도를 정확히 파악할 수 있을 것으로 생각된다.

따라서 본 논문에서는 사물인터넷 패러다임으로 연결된 센서나 장비를 이용하여 상황을 인지하고 이러한 정보를 기반으로 상황을 인지하고 각 사용자마다 각 받고자, 알고자 하는 정보를 제공하는 것이 상황인지 IoT 서비스라고 정의하고 이러한 정보 제공을 위해 사용자에게 적합한 온톨로지 기반으로 제안한다. 정보가 제공되는 센서 데이터 처리 메커니즘은 오브젝트 및 서비스 추상화 (Object & Service abstraction), 시멘틱 어노테이션 (Semanticannotation), 추론 (Reasoning), 서비스 실행 (Service execution) 의 4 가지 단계를 통해 센서 데이터를 상황 정보로 변환한다. 또한 제안하는 사용자 친화적 서비스 정의 환경은 장소 설정 (Place setting), 오브젝트 설정 (Object setting), 오브젝트 및 기능 선택 (Object & function selection), 상황 설정 (Condition setting) 의 4 단계로 구성되어 있으며 웹 기반의 서비스 정의 UI 를 제공하여 프로그래밍 능력이 없는 일반적인 사용자가 서비스를 쉽게 정의 하도록 한다.

3.1 온톨로지 기반 리소스 및 상황 모델링 기술

상황인지 서비스를 제공할 수 있는 스마트 환경을 실현하기 위해서는 모

든 정보를 기계가 인식 가능한 형태로 제공해야 한다. 이를 위해서는 센서에서 생성된 원시 데이터를 컨텍스트 정보로 변환하는 메커니즘과 변환된 정보를 웹상의 리소스로 표현하는 표현기술이 요구된다. 이러한 컨텍스트 정보를 표현하는 표현 기술로는 표현력의 정도에 따라 대표적으로 OWL(Web Ontology Language), RDF (Resource Description Framework), RDFs (RDF Schema)를 사용한다. 상황인지 서비스 제공을 위해서는 이와 같이 표현된 정보를 이용하여 상황 모델링을 제공해야 한다. 상황은 어떠한 개체의 상태를 특징화 시킬 수 있도록 사용되는 정보이다. 이러한 상황을 인간의 사고를 통해 인식하는 것과 같이 상황 정보에 대한 높은 수준의 추정적 개념을 제공하기 위해 상황 모델링이 요구된다. 상황인지 서비스를 위한 온톨로지 기반의 상황 모델링 기법을 적용한 연구들이 진행되고 있다. 앞에서는 사용자 지역의 기상 및 보건 정보를 입력하여 u 헬스 서비스에 필요한 상황 정보를 온톨로지 기반으로 도출하고 그에 맞는 건강 지수, 운동 및 식단 추천 서비스를 제공한다. 이는 온톨로지를 건강정보와 같은 특정 영역의 상황을 상세하게 정의하여 서비스를 제공하므로 이를 위해서는 전문적 지식 및 기술적 수준을 요구하며 범용적 서비스를 제공하기 위한 상황 정의로 확장하기에는 구조적 한계가 존재한다.

3.2 서비스 매시업 기술

다양한 서비스의 제공을 위해 리소스 및 서비스 정의 기술에 의해 표현된 정보들을 이용하여 서비스를 매시업 하고 제공하는 연구가 활발히 진행되고 있다. E.Avilés-López 은 웹 기술기반의 IoT 서비스를 위한 사용자 상호작용 모델을 지원하는 프레임워크를 제안하였다. 이 프레임 워크는 기능적 측면

에서 각 서비스명세를 기술하는 방법, 사용자 환경에서 사용할 수 있는 서비스를 검색하는 방법, 관심 있는 이벤트가 발생할 때 동료와 통신 하는 방법, 서비스에서 데이터와 기능을 결합하는 방법을 제시하고 있다. 또한 Application Manifests 라 정의한 컴포지션 언어를 제안한다. 이는 컴포지션에 참여하는 서비스, 서비스 간의 흐름, 매시업된 결과물, 실행일정, 인증자격증명 및 기타 메타데이터와 같은 응용프로그램 제어 데이터를 포함하고 있다. S.Han 는 서비스라고 정의한 표준화된 형태의 독립적인 컴포넌트로 BAS (Building Automation System)의 기능을 분리하여 이를 SOA (Service Oriented Architecture)기반의 분산형 서비스로 제공하는 방법을 제안한다. 또한 미리 정의된 규칙을 기반으로 6 단계 컴포지션 처리과정을 통해 해당 장치 및 서비스를 매시업, 모델링하고 처리하여 사용자에게 적합한 서비스를 제공한다. 또한 서비스 캐시를 이용하여 큰 규모의 환경에서 효과적임을 보였으나 액츄에이터의 기능 한 가지를 서비스로 정의하고 규칙에 맞게 여러 서비스를 실행하는 구조로 되어 있어 복잡한 상황 및 다양한 서비스를 매시업할 경우 서비스 레시피를 정의하는데 복잡도가 상승할 것으로 예상된다. 또한 온톨로지를 이용하여 상황을 인지하였으나 대부분 컴포지션 계획에 따라 진행되는 구조로써 보다 적극적으로 상황정보를 이용할 수 있는 구조가 요구된다. C. H. Quyet 은 다양한 장치, 사양, 프로토콜을 통합하고 효과적으로 이용하기 위한 클라우드 기반의 스마트 홈 구조를 제안하고 있으며 DPWS (Device Profile for Web Services) 기반의 홈 네트워크 환경에 온톨로지 기술을 적용하여 스마트 홈 서비스를 제공한다. 사용자가 수동적으로 장치를 제어하는 방법, 미리 정의된 규칙을 기반으로 자동적으로 제어하는 방법, 사용자의 행동의 패턴을 조사하여 규칙에 추가하는 방법으로 구성되어 있다. 이 연구의 가장 큰 특징은 사용자의 패턴을 분석하여 서비스 규

칙에 적용시키는 것이다. 이는 사용자의 행위 패턴을 조사하여 특정 행위 자체를 규칙에 저장하게 된다. 하지만 사용자의 모든 행위를 기록, 분석하는 것은 많은 처리 비용을 야기할 것으로 예상된다.

3.3 사용자 친화적 서비스 정의 환경

사용자 주변의 상황을 인지하고 그 상황에 맞는 서비스를 동적으로 제공하는 것은 매우 복잡하며 고려해야 할 사항들이 많다. 따라서 사용자에게 원하는 서비스를 손쉽게 정의할 수 있는 환경을 제공하려는 연구가 활발히 진행되고 있다. UbiSOA 은 스마트 환경과 사용자간의 상호작용을 할 수 있도록 만든 매시업 에디터로써 사용자는 사용가능한 서비스 혹은 이전에 정의된 서비스를 통해 새로운 서비스를 정의할 수 있다. UbiSOA 에디터는 웹 기반 환경에서 끌어 놓기 (Drag&Drop) 방식의 GUI (Graphic User Interface)를 제공하여 서비스 정의를 지원한다. 하지만 사용자가 서비스를 정의하기 위해서는 서비스 실행 엔진 메커니즘의 동작 방법 및 조건을 나타내는 레시피 (recipes)를 정의해야 한다. 앞에서 제안하는 레시피는 PHP (Personal Hypertext Preprocessor)의 확장된 버전으로 이를 작성하기 위해서는 사용자가 웹 프로그래밍 언어 능력을 가지고 있고 시스템의 명확한 이해가 필요하므로 보다 직관적이고 쉬운 컴포지션 에디터가 요구된다. ClickScript 는 서비스 개발자에게 자원 및 연산자를 블록화 하여 쉽게 매시업 기능을 수행할 수 있게 도와주는 비주얼 프로그래밍 툴이다. 자바스크립트로 구현되어 거의 모든 웹 브라우저에 동작한다. ClickScript 를 이용하여 Bluetooth, Zigbee 와 같은 하위 수준 프로토콜을 직접적으로 제어 할 수는 없지만 HTTP 를 이용하여 각 자원에 접근할 수 있다. 각 자원은 하나의 블록으로

표현되며 연산자들과 더불어 새로운 서비스를 만들 수 있다. 하지만 이 에디터는 각 자원을 웹 기반으로 연결하여 보여주며 조건을 정의하는 등의 방식으로 수행되기 때문에 기본적인 시스템과 for, while, if, than 등의 제어문의 이해 및 서비스 흐름의 설계 능력이 필요하다. 따라서 일반적인 사용자가 각각의 장치의 특성을 이해하고 특정 조건을 정의하기 위한 장치를 구성하는데 어려움이 따를 것으로 생각된다. JIGSAW는 태블릿 기반 에디터로 여러 상황을 그림으로 표현한 퍼즐 조각들을 사용자가 결합하여 원하는 서비스를 정의할 수 있는 환경을 제공한다. 퍼즐로 여러 가지 기능적 모듈을 표현하였지만 사용자의 수준 및 특성에 따라 서로 다른 이해를 도출할 가능성이 있어 직관적이지만 오류를 범할 확률이 높다. 또한 관계연산자와 산술연산자를 지원하지 않아 산술이 필요한 서비스는 정의할 수 없다.

3.4 오브젝트 데이터의 상호 운용성 증대를 위한 오브젝트 데이터 포맷

상황인지 서비스의 제공을 위해서는 센서 데이터의 재사용 및 상호 운용성의 제공이 필요하다. 이를 위해서는 서비스 제공 요구사항을 만족하는 공통된 데이터포맷이 요구된다. 또한 센서는 대체적으로 전원 및 처리능력이 제한적이며 무선 통신 환경에서 데이터를 전송하므로 매우 제약적인 문제점을 가지고 있다. 따라서 적용되는 프로토콜 역시 단순하고 가벼운 특징을 가져야 하며 전송되는 데이터 또한 적은 양을 갖는 것이 유리하다. XML, EXI (Efficient XML Interchange), BXML (Binary XML), Fast Infoset 등 데이터를 표현하는 인코딩 방법은 다양하다. 단순히 같은 정보를 적은 비트로 표현하는 것도 중요하지만 데이터를 습득했을 때 값을 전달하는 파싱의 복

잡도 또한 중요하다.

이는 플랫폼 혹은 데이터 처리 단계에서 비용을 낮춰주며 또한 잘못된 값의 전달을 줄여줄 수 있다. 따라서 본 논문에서는 Table 4.와 같은 오브젝트 데이터 포맷을 제안하였으며 이는 JSON 형태로 작성되어 플랫폼에 전달된다. 제안하는 오브젝트 데이터 포맷은 제어하는 오브젝트의 특성에 따라서 크게 센서와 액추에이터로 구분한다. 일반적으로 센서는 주변의 현상을 전기적인 수치로 반환한다.

센서 데이터 포맷은 센서를 구분, 접근할 수 있도록 하는 URI (Uniform Resource Identifier)와 센서의 종류를 나타내는 Type, 측정된 값의 단위를 나타내는 Unit, 실제 측정된 측정값을 나타내는 Value, 측정 간격을 나타내는 Accuracy, 위치를 나타내는 Location, 소유 및 접근 권한을 나타내는 Ownership 을 포함하고 있다. 플랫폼은 전달받은 오브젝트 데이터를 온톨로지에 매핑하게 되며 추후에 이것이 상황정보로 변환된다. 액추에이터 데이터 포맷은 센서와 마찬가지로 URI 와 액추에이터의 이름, 현재 상태를 나타내는 Status, 제공해 줄 수 있는 기능 목록인 Function list, 위치를 나타내는 Location, 소유 및 제어 권한을 나타내는 Ownership 을 포함한다. Table 4.의 예를 보면 센서의 종류는 온도, 값은 섭씨 28 도, Jun 이 소유하고 있으며 침실에 위치하고 있다는 정보를 포함하고 있다. 이와 같은 정보를 습득하면 우리는 침실은 섭씨 28 도 라는 사실을 알 수 있게 된다. 또한 액추에이터의 경우에는 Lamp 이면서 현재 상태는 On, 즉 켜져 있는 것을 알 수 있으며 제공되는 기능 목록을 통해 Turn on, Turn off 기능을 이용할 수 있으며 Location 정보를 통해 부엌에 위치하고 있다는 것을 알 수 있다. 결과적으로 우리는 이 액추에이터의 URI 를 통해 부엌의 전등을 제어할 수 있다. 하지만 이와 같은 정보만으로는 상황인지 서비스 제공이 용이하지 않다. 예를

들어 Table 4.의 센서로 부터 받은 정보는 섭씨 28 도의 온도정보인 것은 판단할 수 있지만 이는 말 그대로 현재 센서가 위치하는 방의 온도가 될 수 있고 특정 사용자의 체온이 될 수도 있다. 이와 같은 이유 때문에 정확한 의미상의 구별이 필요하다. 본 논문에서는 이를 위해 센서의 원시 데이터를 컨텍스트 정보로 변환하는 시멘틱 어노테이션을 제공한다.

Table 4. 제안하는 오브젝트 데이터 포맷¹⁾

Sensor		Actuator	
URI	../sensor/ tempsensor1	URI	../actuator/ lamp2
Type	Temperature	Name	Lamp
Unit	centigrade	Status	ON
Value	28	Function list	Turn on, Turn off
Accuracy	5	Location	Kitchen
Location	Bedroom	Ownership	Jun
Ownership	Jun		

3.5 상황 인지를 위한 기반 온톨로지

온톨로지는 특정 영역을 표현하는 데이터 모델로 개념간의 관계를 기술하는 정형화된 어휘의 집합이다. 이는 시멘틱의 기반 기술로써 본 논문에서는 온톨로지를 이용하여 센서 데이터를 상황정보로 변환한다. 그림 3 온톨로지는 장소, 오브젝트, 컨텍스트, 서비스 온톨로지의 4 계층으로 구성되며 이와 같은 온톨로지의 계층적 개념을 통해 수정, 확장 및 기존 온톨로지의 재사용이 용이하다. 또한 서비스 정의에 필요한 컨텍스트를 온톨로지로 정의하여 제공함으로써 사용자는 자신이 원하는 상황을 정의하기 위해 요구되는

1) 최환석, 이준영, 양나리, 이우섭 (2014). 상황인지 IoT 서비스 제공을 위한 온톨로지 기반 사용자 친화적 서비스 환경. 한국콘텐츠학회논문지, 14(7), 29-44.”

센서의 목록이나 기준 값을 고려하지 않아도 보다 쉽게 조건을 정의할 수 있으며 컨텍스트의 계층적 관리와 수정 및 개인화를 제공한다. 추후 장소 온톨로지를 확장하여 새로운 환경을 지원할 수 있으며 오브젝트가 추가적으로 필요할 경우 오브젝트 온톨로지에 오브젝트 정보를 추가하고 기존에 제안한 온톨로지 계층 간의 프로퍼티를 재사용 하여 여러 상황 및 환경을 지원할 수 있다.

각 계층별 설명은 다음과 같다.

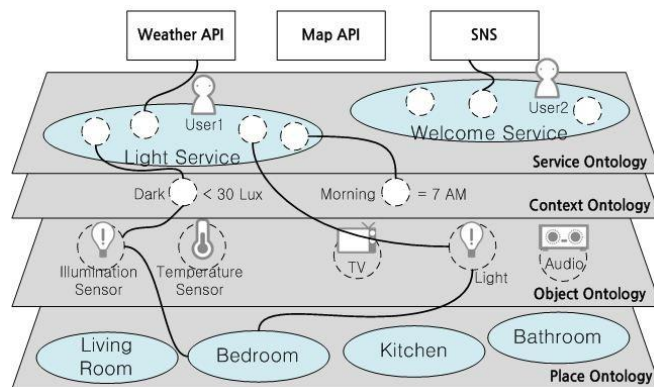


그림 3. 예제 기반 온톨로지 모델 개념도

3.5.1 장소 온톨로지

장소 온톨로지는 서비스 도메인을 구성하는 온톨로지이다. 서비스 도메인은 부엌, 거실, 침실, 서재 등 다양한 종류를 포함할 수 있으며 각 장소는 각각의 특징을 가지고 있다. 예를 들면 침실의 경우 일반적으로 침대가 배치될 수 있으며 부엌의 경우 주방 가전 및 가스 시설을 포함하고 있다. 만약 서비스 도메인이 회사, 공장, 농장 등으로 다양하게 확장될 경우 요구되는 온톨로지를 확장하여 정의함으로써 다양한 서비스 도메인을 지원할 수 있다.

3.5.2 오브젝트 온톨로지

오브젝트 온톨로지는 서비스 도메인에 위치하는 다양한 센서 및 액추에이터를 구성한다. 오브젝트는 각각의 특징을 가지고 있다. 센서나 액추에이터의 목적에 따라 오브젝트를 온톨로지 형태로 정의하며 센서의 경우 `hasValue` 프로퍼티를 통해 습득한 값과 연결되게 된다. 액추에이터의 경우에는 `hasStatus` 프로퍼티를 통해 상태 값을 가지게 된다. 또한 위에서 설명한 장소 온톨로지와의 연결을 통해 현재 센서의 위치를 정의한다. 이와 같이 상관관계에 있는 온톨로지 사이의 연결을 통해 오브젝트에서 습득한 정보를 매핑한다.

3.5.3 컨텍스트 온톨로지

컨텍스트 온톨로지는 서비스 조건이 되는 컨텍스트를 정의한다. 사용자는 특정 상황 (예: 어둡다, 덥다, 밝다, 습하다 등)으로 서비스 제공 조건을 정의하게 되는 데 본 논문에서는 이러한 특정 상황을 컨텍스트라 정의한다. 이는 특정 상황을 판단할 수 있는 센서와의 연결을 통해 정의되며 상황에 따라 조건 값을 가지게 된다. 예를 들어 조도 센서에서 습득된 값이 30 lux 보다 적으면 “어둡다” 라는 컨텍스트로 정의할 수 있다. 사용자는 컨텍스트 온톨로지를 통해 자신이 원하는 서비스 제공 조건을 판단하기 위한 사항 (센서의 종류, 컨텍스트 조건 값 등)에 관한 지식이 없어도 기반 온톨로지에 미리 정의된 기본 컨텍스트를 통해서 보다 쉽게 서비스 조건을 정의할 수 있다. 또한 컨텍스트의 계층적 관리를 통해 다수의 컨텍스트를 이용하여 새로운 컨텍스트를 생성하는 컨텍스트 매시업을 용이하게 한다.

3.5.4 서비스 온톨로지

가장 상위에 위치하는 서비스 온톨로지는 서비스 조건에 해당하는 컨텍스트와 서비스 목적 달성을 위한 액추에이터의 조합으로 이루어진다. 사용자가 선택 혹은 생성한 컨텍스트와 액추에이터의 종류 및 기능, 최종 상태 등의 정보를 통해 서비스를 정의한다. 이는 SWRL(Semantic Web Rule Language)과 Jena rule 에 의해 표현된다.

3.6 서비스 시나리오 설정 예

센서로부터 수신되는 데이터 처리 단계의 설명에 앞서 보다 명확한 처리 단계를 보여주기 위해 정형화된 하나의 서비스 시나리오를 정의한다. 본 시나리오는 사용자 단일 환경의 1인 가구로 가정하며, 사용자의 위치는 움직임 센서에 의해 구별한다고 가정을 한다. 각 순서 별 시나리오는 다음과 같을 수 있을 것이다.

- ① 침실에 설치된 조도 센서에 의해 침실의 조도가 측정되고 있다. 사용자가 아침에 일어났을 때 침실이 어둡다면 램프를 켜고 커튼을 연다.
- ② 기상을 한 사용자는 부엌으로 이동한다. 아침시간에 부엌으로 사용자가 이동한 것을 감지하게 되면 커피포트가 동작한다.
- ③ 사용자가 거실로 이동하면 TV에서는 주로 보던 채널이 방송된다.
- ④ 사용자는 출근 준비를 위해 욕실로 이동한다. 샤워를 시작하면 욕실안의 습도가 증가한다. 습도센서는 욕실의 습도를 측정하고 일정 습도 이상이 된 것을 감지하면 환풍기를 동작시킨다.

위의 시나리오는 서비스 조건이 되는 컨텍스트와 서비스 행위가 되는 액츄에이터의 기능과 같은 구성요소를 포함하고 있다. 시나리오 4 번의 경우 환풍기가 켜지는 서비스의 조건은 사용자의 위치와 습도에 관한 컨텍스트이다. 또한 1 번의 경우 아침이라는 상황과 어둡다는 상황이 합쳐져 새로운 컨텍스트가 서비스 조건이 된 경우이다. 이와 같은 다양한 조건의 시나리오를 서비스로 제공하고자 할 경우 다양한 컨텍스트를 만족시키는 새로운 컨텍스트가 필요하며 이를 위해 컨텍스트 매시업이 요구된다.

3.7 온톨로지 기반의 센서 데이터 처리 단계

앞서 설명한 바와 같이 센서 데이터는 시스템, 문법, 구조, 의미적 측면의 이질성을 가지고 있다. 본 논문에서는 이를 효과적으로 제거하기 위해 온톨로지 기반의 센서 데이터 처리 메커니즘을 제안한다. 그림 4는 제안하는 메커니즘의 구현구조이다. 센서 데이터의 효율적인 처리를 위해서 Java EE 기반의 실험 환경을 구축하였다. Apache Tomcat 6.0 은 RESTful 웹 서비스의 제공 및 관리를 위해서 사용하였고 시멘틱 데이터의 처리를 위해 자바 기반의 Jena framework 2.6.4 를 이용하였다. 또한 시멘틱 데이터의 효율적인 질의를 위해 SPARQL API 를 이용하였다. 여러 상황정보가 입력된 온톨로지에서 상황의 판단 및 추론을 위해 Pellet 추론기를 이용하였다. SWRL 과 Jena rule 을 이용하여 데이터 처리 및 서비스 제공을 위한 서비스 규칙을 정의하였다. 또한 오브젝트 데이터의 손쉬운 습득 및 Jena 프레임워크와의 파싱을 위해 Jackson API 2.2.3 을 이용하여 JSON 데이터와 자바 오브젝트간의 변환을 처리하였다. 그리고 온톨로지의 저장 및 관리를 위해 MySQL5.1 과 JDBC 5.1.25 를 이용하였다.

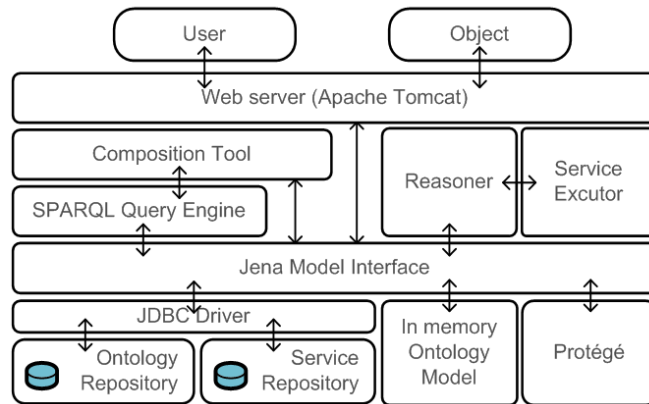


그림 4. 센서 데이터 처리를 위한 구현구조

그림 4는 오브젝트 및 서비스 추상화, 시멘틱 어노테이션, 추론, 서비스 실행의 4 단계로 이루어진 제안하는 메커니즘의 처리 단계를 보여준다. 다양한 프로토콜 및 메시지 포맷을 가지고 있는 오브젝트는 추상화되며 사용자의 서비스 정의 또한 추상화되어 플랫폼에 전달된다. 이렇게 전달된 오브젝트나 서비스의 정보는 기반 온톨로지에 매핑되는데 이 과정을 시멘틱 어노테이션이라고 한다. 이렇게 매핑된 온톨로지에서 상황을 인지하기 위해 정의된 각종 컨텍스트는 추상화 과정을 통해 분류(classification)되며 최종적으로 Jena rule로 정의된 서비스 조건 및 결과 정보에 의해 각종 액추에이터 혹은 사용자 장치에 상태정보를 전달하고 제어한다.

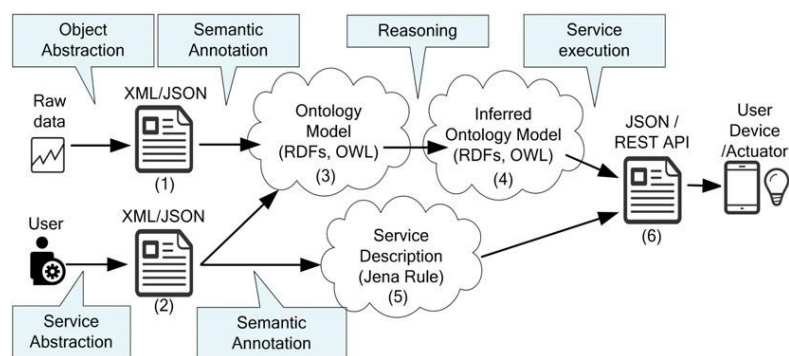


그림 5. 온톨로지 기반의 센서 데이터 처리 단계

3.7.1 오브젝트 및 서비스 추상화(Object & Service abstraction)

그림 5 는 센서 데이터의 시스템, 문법, 구조적 측면의 이질성을 제거하기 위해 추상화를 거친 오브젝트 데이터이다. 제한적인 센서 환경에서의 효율적인 전송을 위해 파싱이 쉽고 데이터양이 적은 JSON 포맷을 사용하였다. 이는 타겟 시나리오 4 를 기반으로 한 오브젝트데이터로 오브젝트의 URI, 유형은 습도, 측정된 값은 70%임을 알아 낼 수 있으며 위치정보를 통해 욕실에 설치된 것을 알 수 있다. 결과적으로 욕실의 습도는 70%임을 파악할 수 있다.

```
{
  "URI" : "HumiditySensor 1", "Type" : "HumiditySensor",
  "Unit" : "Percent", "Value" : 70, "Accuracy" : 5,
  "Location" : "Bathroom_Jun", "Ownership" : "Jun"
}
```

그림 6. 추상화된 오브젝트 데이터 (JSON)

그림 6 은 서비스 요구사항을 정의한 내용이다. 서비스는 서비스 추상화를 통해 그림 6 과 같이 기술된다. 이는 컨텍스트와 서비스 규칙을 정의하는데 사용된다. 이는 서비스와 컨텍스트의 이름, 하위 컨텍스트의 목록 및 기준 값, 액추에이터의 이름 및 기능 목록을 포함하고 있다. 컨텍스트를 계층적으로 구성하는 이유는 사용자가 원하는 서비스 조건이 다수의 컨텍스트를 만족하는 조건일 경우 이를 정의하기 위한 것이다. 결과적으로 그림 6 의 메시지를 통해서 사용자가 정의한 서비스의 요구사항을 파악할 수 있으며 온톨로지에 컨텍스트를 생성할 수 있다. 내용을 보면 서비스 조건인 컨텍스트는 습도가 50 이상일 경우 습하다 (Damp)로 정의하고 있으며 이를 만족할 경우 Ventilator1 을 동작시킨다.

있으므로 온톨로지의 센서 클래스의 하위 클래스인 HumiditySensor 클래스에 인스턴스를 생성한다. 이는 미리 정의된 기반 온톨로지의 특성에 따라 Sensor 클래스는 수집한 값을 갖기 때문에 Value 클래스의 하위 클래스인 ObservationValue 클래스와 hasValue 프로퍼티를 이용하여 연결된다. 또한 미리 정의된 컨텍스트는 여러 가지 현상과 감각을 분류하여 계층적 클래스로 구성되어 있다. 앞의 데이터의 경우 습도센서에서 수집된 것 이므로 TemperatureSense 클래스의 하위 클래스인 Damp 클래스와 연결되어 있다. 또한 Object 클래스는 Place 클래스와 연결되어 위치를 알 수 있다. 결과적으로 센서에서 수집된 정보를 온톨로지에 적용하여 “욕실은 습하다.”와 같은 상황 정보로 처리 한다.

그림 8 은 특정 컨텍스트(Damp_Jun)의 OWL 표현이며 기반 온톨로지의 일부로 컨텍스트의 정의를 보인다. 이는 예로서 컨텍스트 클래스의 인스턴스인 Damp_Jun 은 습도 센서 인스턴스인 Humidity Sensor1 과 Humidity Sensor ObservationFrom 프로퍼티를 이용하여 연결된다. 또한 hasHumidity-SensorObservationValue 프로퍼티를 이용하여 습도 값 정수형 70 을 가진다. 이 상수 값은 ObservationValue 클래스의 인스턴스로 존재한다.

```
<!-- http://www.semanticweb.org/dia/ontologies/2013/8/shbo#Damp_Jun -->
<owl: NamedIndividual rdf:about="" & shbo: Damp_Jun>
  <rdf: type rdf:resource="" & shbo: TemperatureSense/>
  <hasHumiditySensorObservationValue rdf:datatype="" & xsd: integer> 70
  </hasHumiditySensorObservationValue >
  <shbo: HumiditySensorObservationFrom rdf: resource="" & shbo:HumiditySensor 1"/>
</owl: NamedIndividual >
```

그림 9. Damp_jun 컨텍스트의 OWL 표현

3.7.3 추론(Reasoning)

서비스 조건인 컨텍스트를 정의하고 센서로부터 전달받은 정보를 온톨로

지에 매핑하면 컨텍스트 판단 조건의 만족 유무를 판단해야 한다. 이를 위해
 서 본 논문에서는 Pellet 추론기를 사용한다. 추론기는 조건에 맞는 클래스나
 인스턴스를 분류 하는 역할을 한다. 이를 통해 센서 값을 가지고 있는 컨텍스트
 인스턴스가 특정 컨텍스트의 조건에 맞는 경우 추론기가 이를 분류한다.

그림 9는 추론 결과를 보여주는 것으로 Protege 내부에 클래스의 정의 상
 태를 보여준다. 그림 9에서는 Damp 컨텍스트를 클래스로 정의하였으며 이
 때 50 이상의 값을 갖는 인스턴스가 이 클래스에 속하게 된다. 이는
 hasHumiditySensorObservat-ionValue 프로퍼티를 이용하여 값을 가지며 결
 과적으로 Damp_Jun 인스턴스는 70 이라는 값을 가지므로 그림 9의 하단부
 와 같이 Damp 클래스로 구분 된 것을 확인할 수 있다.



그림 10. Damp 클래스의 추론 결과

3.7.4 서비스 실행(Service execution)

기반 온톨로지를 이용하여 서비스 조건의 만족을 판단하면 서비스를 실행
 해야 한다. 이를 위해 본 논문에서는 Jena 프레임워크를 기반으로 서비스
 조건의 판단 및 실행을 한다. 그림 10은 Jena rule을 이용한 서비스 정의를
 보여준다.

```
[bath_Jun:
  (shbo:Damp_Jun rdf:type shbo:Damp),
  (shbo:Ventilator 1 shbo:hasStatus "OFF"^^xsd:string)
  -> (shbo:Ventilator 1 shbo:hasStatus "ON"^^xsd:string)
]
```

그림 11. Jena rule 을 이용한 서비스 정의

Damp_Jun 인스턴스가 Damp 클래스의 구성원 (type)이 된다면 서비스 실행 조건을 만족한 것이다. 이때 Bath_Jun 이라고 정의한 서비스가 실행된다. 따라서 Ventilator1 은 OFF 의 상태에서 ON 의 상태로 변경된다. 이와 같이 서비스 조건에 의해 액츄에이터 상태가 변경되려면 이러한 상태정보를 액츄에이터에 전달 해야 한다. 만약 전달해야 할 정보의 내용이 다수의 항목을 포함하고 있다면 오브젝트 데이터 포맷과 같이 JSON 형태로 전달된다. 하지만 단순히 상태정보를 변경하는 것이라면 간단하게 REST operation 을 이용할 수 있다. 예를 들어 Ventilator1 의 상태정보를 on 으로 변경하고자 한다면 해당 오브젝트의 URI(예: <http://hostaddr/put/actuator/ventilator1/status/on>)를 이용하여 변경한다. 이는 오브젝트 내부의 REST operation 처리 모듈에 따라 다르게 표현될 수 있다.

3.8 사용자 친화적 서비스 정의 환경

모든 서비스는 특정 조건에서 제공된다. 예를 들면 극장에서 영화표를 구매하고 상영관에 들어가면 시작시간에 맞추어 영화가 방영된다. 또는 비밀번호 잠금장치에 올바른 비밀번호를 입력한다면 문이 열린다. 이와 같이 어떤 형태의 서비스라 할지라도 서비스를 제공하는 조건 및 제공되는 행위, 동작, 콘텐츠 등의 서비스 행동은 정해져 있다. 상황인지 서비스의 경우에도

마찬가지이다. 다만 서비스 조건의 만족 여부를 플랫폼의 기반 온톨로지를 통해 판단하고 해당 서비스 행동을 제공하도록 지시할 뿐이다. 앞 장의 관련연구에서 서술한 바와 같이 사용자에게 서비스 정의 환경을 제공하는 연구는 계속되고 있으나 기존의 서비스 제공 환경은 서비스 정의에 필요한 각종 상황의 기준 값, 온톨로지의 추론 결과를 알기 위한 SPARQL, 서비스 정의를 위한 스크립트 기반의 각종 프로그래밍 언어능력을 요구하는 등 일반적인 사용자에게는 수행하기 어려운 실정이다. 따라서 본 논문에서는 플랫폼의 구조, 컨텍스트의 정의를 위해 필요한 센서의 목록이나 조건, 서비스를 정의하기 위한 프로그래밍 능력 등이 없는 일반적인 사용자가 쉽게 자신이 원하는 서비스를 정의할 수 있는 환경을 제안한다. 이 환경에서 사용자는 단지 서비스 제공 조건 및 서비스 제공결과인 액츄에이터의 기능 두 가지 요소만을 선택하여 서비스를 정의할 수 있다. 사용자는 서비스를 생각 할 때 내가 어떤 동작, 정보, 행위를 제공받을 수 있는지 고민하고 또한 그 결과가 어떤 상황에 제공되는 것이 좋을지 판단한다. 이와 같이 서비스 정의 환경은 서비스의 결과를 정의 하는 것으로 시작된다. 또한 서비스 상황은 여러 가지 컨텍스트로 정의되며 사용자는 기 정의된 컨텍스트를 이용하여 자신만의 서비스 조건을 정의할 수 있다.

3.8.1 제안하는 사용자 친화적 서비스 정의 과정

본 논문에서 제안하는 사용자 중심 서비스 정의 과정을 나타내는 것으로 단계마다 필요한 입출력을 정의하고 있다. 이는 장소 설정, 오브젝트 설정, 오브젝트 및 기능 선택, 상황 설정의 4 단계로 구성되어 있다. 서비스 정의 과정은 크게 서비스 환경을 정의하는 앞 두 단계와 서비스 자체를 정의하는

뒤의 두 단계로 구분 될 수 있다. 플랫폼 관점에서는 사용자에게 상황인지 서비스를 제공하고자 할 때 고려해야 할 요소들이 매우 많다. 이는 환경적으로 인지해야 할 공간, 오브젝트의 종류 및 배치, 서비스 조건 등 매우 다양하다. 이를 사용자가 정확하게 정의한다면 플랫폼 입장에서 처리(추론)해야 할 정보들이 확연하게 감소되며 정확한 서비스 제공이 용이해 진다. 추후 사용자의 행동을 반영하는 적응형 서비스로 발전시켜 나갈 수 있는 장점도 가지고 있다. 관련연구에서 살펴본 모든 사용자 중심 서비스 정의 환경은 한 화면에 전체적인 서비스를 정의하는 플로우차트 개념으로 전체적인 흐름을 파악하기엔 좋지만 사용자가 전체적인 서비스 흐름을 이해하고 정의해야 하며 단계를 나누지 않고 모든 요소들을 나열하여 실수가 생길 수 있다.

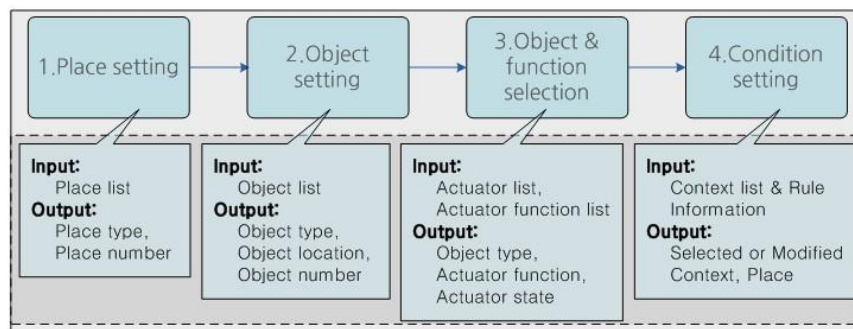


그림 12. 서비스 정의 과정

제안하는 정의 과정은 그림 12의 서비스 정의 과정을 단계별로 진행하는 방식으로 서비스 정의 마법사 방식을 적용하여 사용자가 정의해야 하는 사항들을 순차적으로 도출하여 어떤 사항을 선택해야 하는지 도움을 준다. 따라서 사용자의 오류를 줄일 수 있으며 전체적인 서비스 흐름의 이해를 요구하지 않고 단순히 가능한 요소들을 선택하는 방식으로 서비스를 정의할 수

있다. 이를 위해 본 연구에서는 그림 13의 웹기반 서비스 정의 UI를 제공한다. 이는 HTML5, CSS3, 자바스크립트등을 이용하여 구현하였으며 데이터 전달을 위해 HTTP와 JSON을 이용한다.

3.8.1.1 장소 설정 단계(Place Setting)

첫 단계는 그림 13과 같이 서비스 제공 환경을 정의하는 장소 설정 단계이다. 같은 가정환경이라도 방의 개수 및 종류의 차이가 존재하며 추후 회사, 공장 등 다양한 환경으로 확장될 수 있기 때문이다. 본 논문은 가정환경을 타겟으로 기반 온톨로지의 Place 클래스에 일반적인 가정에 존재하는 장소의 특징 및 종류가 정의되어 있다. 장소 설정을 위해 기반 온톨로지에 정의된 Place 클래스의 하위 클래스 목록 (장소의 종류)을 UI에 전달한다. 사용자는 그림 13과 같이 전달받은 목록 중 자신의 서비스 환경에 맞는 종류를 선택 (Check)하고 만약 한 종류의 장소가 여럿 존재할 경우 장소의 개수를 입력한다. 사용자의 입력이 끝나면 그림 13의 하단부와 같이 사용자가 선택한 장소가 생성된다. 이와 같은 단계를 통해 사용자의 서비스 환경이 기반 온톨로지에 적용된다.



The image shows a web-based service definition UI for 'Jun's House'. At the top, there are four tabs: PLACE, OBJECT, ACTUATOR, and CONTEXT, with 'PLACE' selected. Below the tabs, the title 'Jun's House' is displayed. The main area contains a list of rooms with checkboxes and input fields for counts:

Room Type	Count
<input checked="" type="checkbox"/> Kitchen	1
<input checked="" type="checkbox"/> Living Room	1
<input checked="" type="checkbox"/> Bed Room	1
<input type="checkbox"/> Study Room	
<input checked="" type="checkbox"/> Bathroom	1
<input type="checkbox"/> Veranda	

Below this list, there are four large blue rectangular boxes representing the selected rooms: Bed Room, Living Room, Kitchen, and Bathroom.

그림 13. 웹기반 서비스 정의 UI (장소 설정 단계)

그림 14는 그림 13과 같이 장소를 설정했을 때 그 결과가 적용된 기반 온톨로지의 상태를 보여준다. 사용자가 선택한 결과에 따라 장소 온톨로지의 클래스 하위에 인스턴스 형태로 생성된다. 그림 14를 보면 Kitchen_Jun 인스턴스가 Kitchen 클래스의 소속으로 생성된 것을 확인할 수 있다.

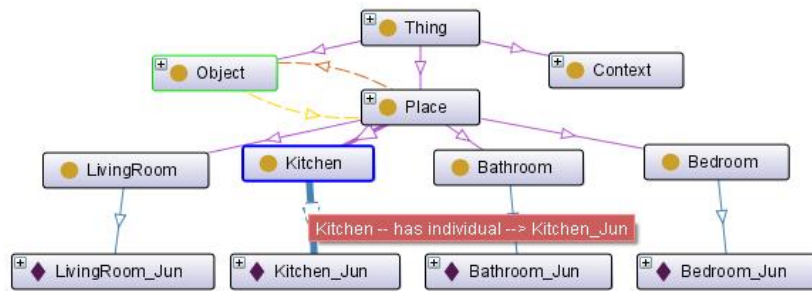


그림 14. 장소 설정 단계가 적용된 기반 온톨로지 상태

3.8.1.2 오브젝트 설정 단계(Object Setting)

장소 설정 단계가 끝나면 서비스 제공에 필요한 오브젝트를 등록하는 오브젝트 설정 단계가 진행된다. 사용자에게 실제적인 서비스를 제공하는 것은 어떤 행위 및 동작을 할 수 있는 액추에이터라고 볼 수 있다. 따라서 사용자는 원하는 액추에이터의 기능을 선택하여 서비스를 정의한다. 플랫폼은 범위 안에 있는 오브젝트의 추상화된 데이터를 제공 받아 기반 온톨로지의 Object 클래스에 등록한다. 이렇게 등록된 정보 중 사용자가 권한이 있는 액추에이터의 목록을 UI에 제공한다. 그림 15는 오브젝트 설정단계의 서비스 정의를 위한 UI를 보여준다. 사용 가능한 목록은 상단부에 노출하였으며 사용자는 끌어 놓기 형식으로 원하는 액추에이터를 특정 장소에 배치한다. 배치된 액추에이터는 앞과 같이 종류에 따라 기반 온톨로지에 인스턴스 형태로 등록된다. 이와 같이 오브젝트 설정 단계를 통해 배치된 액추에이터는

장소 온톨로지와 연결하여 위치 정보를 갖는다.

기존의 연구들은 실제 오브젝트의 배치를 위해 추가적인 메타데이터 입력 또는 스크립트 기반의 프로그래밍 능력을 요구한다. 또한 동일한 환경을 재사용하기 위해서 반복하여 환경 정의를 해야 하는 단점을 가지고 있다. 그러나, 제안하는 환경은 프로그래밍 능력이 없는 사용자가 손쉽게 환경을 정의할 수 있고 환경을 재사용하여 추가적인 설정을 줄여줄 수 있다. 또한 사용자의 서비스 환경이 변경되거나 추가적인 장소를 설정하고자 할 경우 장소 및 오브젝트 설정 단계를 다시 수행하여 새로운 서비스 환경을 추가할 수 있으며 이렇게 설정된 정보는 온톨로지에 사용자 별로 관리된다.

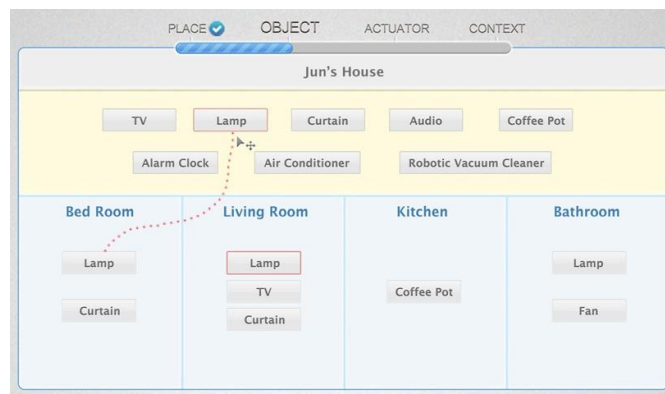


그림 15. 웹기반 서비스 정의 UI (오브젝트 설정 단계)

3.8.1.3 오브젝트 기능 선택 단계(Object & Function Selection)

이 단계에서는 서비스 기능 및 상태를 정의한다. 그림 16는 오브젝트 및 기능 선택 단계의 서비스 정의를 위한 UI이다. 우선적으로 다양한 서비스를 효과적으로 관리하기 위해 서비스 이름을 입력한다. 서비스 정의 환경은 그림 16의 상단부와 같이 장소 및 오브젝트 설정단계를 통해 온톨로지에 정의

된 서비스 환경 정보를 이용하여 사용가능한 액츄에이터 목록을 보여준다. 사용자는 제어하길 원하는 액츄에이터를 선택한다. 액츄에이터가 선택되면 해당 기능 목록이 출력된다. 사용자는 이 목록 중 자신이 원하는 기능 혹은 상태를 선택하여 서비스 행위를 정의한다. 예를 들어 “전등이 켜지고 커튼이 열림” 과 같이 사용자가 다수의 액츄에이터 기능을 동시에 제공받고 싶다면 사용자는 가용 액츄에이터 목록에서 다수의 액츄에이터와 해당 기능을 선택한다.

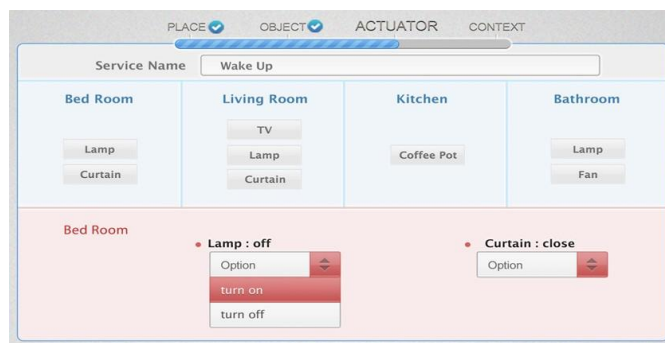


그림 16. 웹기반 서비스 정의 UI (오브젝트 및 기능 선택 단계)

그림 17 은 Protege 를 이용하여 특정 액츄에이터 인스턴스의 정의를 표현한 것이다. 앞서 설명한 바와 같이 기반 온톨로지에는 해당 액츄에이터가 지원하는 기능 목록 및 위치, 현재 상태 정보가 포함되어 있다. 이러한 정보를 바탕으로 사용자는 액츄에이터가 지원하는 목록 및 상태를 알 수 있으며 이를 변경하여 액츄에이터를 제어하게 된다.

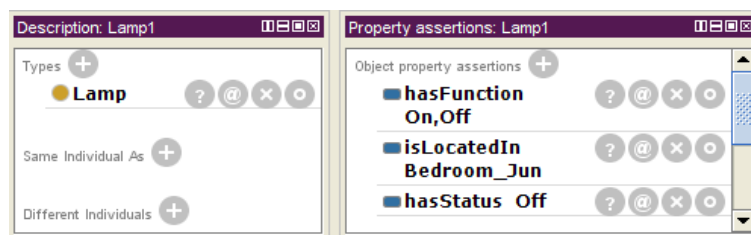


그림 17. 액츄에이터 인스턴스의 정의

3.8.1.4 상황 설정 단계(Condition Setting)

서비스 제공 조건은 플랫폼에서 기 정의된 다양한 컨텍스트를 통해 정의된다. 이는 컨텍스트의 특성에 따라 분류되어 있으며 다양한 상황 (예: 아침 + 어둡다, 욕실에 사용자가 있을 때 + 습하다)을 정의 할 수 있도록 컨텍스트 메시업을 제공한다. 하지만 컨텍스트의 특성에 따라 동시에 선택된다면 조건을 만족시킬 수 없는 경우가 존재한다. 이를 위해서 상충되는 조건 (예: 춥다, 덥다)을 감각이나 현상의 종류에 따라 분류하여 제공한다. 같은 분류에 존재하는 컨텍스트는 다중 선택 될 수 없으며 다른 분류 간에는 동시에 선택할 수 있게 하였다. 또한 컨텍스트는 기준을 가지고 있는데 이는 사용자마다 그 기준이 다를 수 있다. 예를 들어 한 사용자는 26 도의 온도가 덥다고 느끼는 반면 다른 사용자는 28 도의 온도가 되어야 덥다고 느낄 수 있기 때문이다. 이러한 상황을 만족시키기 위해 각 컨텍스트의 기준을 변경할 수 있게 하였고 설정된 조건은 앞과 같은 서비스 요구사항으로 플랫폼에 전달된다. 그림 18은 상황 설정 단계의 서비스 정의를 위한 UI 이다. 좌측에는 사용 가능한 컨텍스트가 분류되어 제공되며 우측에 선택된 컨텍스트의 기준값이 표현된다. 이와 같이 선택 혹은 수정된 컨텍스트는 컨텍스트 메시지를 통해 새로운 컨텍스트로 생성된다. 컨텍스트는 계층적 구조를 가짐으로써 기존에 정의된 기본 컨텍스트 및 기준값을 포함한다. 이와 같은 계층적 관리를 통해 기본 컨텍스트를 기반으로 사용자가 원하는 다양한 컨텍스트를 생성 / 관리 할 수 있다.



그림 18. 웹기반 서비스 정의 UI (상황 설정 단계)

제 4 장 비교 분석

본 장에서는 제안하는 사용자 친화적 서비스 정의 환경과 관련 연구들을 비교 분석한다.

Table 5는 사용자 친화적 서비스 정의 환경을 비교 분석한 결과로써 사용하고 있는 비교 항목은 관련연구의 특징 및 장단점 분석을 통해 도출되었다. 본 논문에서는 사물인터넷의 각종 장비들을 쉽게 탐색하고 찾고 각 사용자가 처한 상황에 맞추어, 온톨로지 기반의 시맨틱 제공을 위해 3가지 사항 (기본 온톨로지, 센서 데이터 처리 메커니즘, UI)을 제안하였다. 이를 위한 비교 항목은 총 6가지로 구성되어 있으며 기반 온톨로지의 적용을 통해 앞의 1-4 번 항목을 제공하고 센서 데이터 처리 메커니즘을 통해 4, 6 번 항목을 제공한다. 또한 사용자 친화적 UI를 통해 1-6 번 항목의 환경을 제공한다.

Table 5. 사용자 친화적 서비스 정의 환경 비교

비교 항목	UbiSOA [10]	ClickScript [14]	JIGSAW [15]	제안 방식
1. 기본 컨텍스트 제공	x	x	o	o
2. 컨텍스트 수정 및 개인화 제공	x	x	x	o
3. 컨텍스트 매시업 제공	o	x	x	o
4. 환경 및 조건 재 사용 가능	o	x	o	o
5. 순차적 서비스 정의 기능 제공	x	x	x	o
6. 프로그래밍 능력 불필요	x	x	o	o

4.1 기본 컨텍스트 제공

사용자가 처한 환경은 매우 다양하며 이를 판단하기 위한 기준도 매우 다양하다. 하지만 일반적인 사용자는 자신이 원하는 상황 정보의 판단을 위해

어떤 정보가 필요한지, 어떤 기준으로 판단할 수 있는지 구상하기란 어렵다고 할 수 있다. 단순히 '덥다'와 같은 상황의 경우에는 온도 센서에서 생성된 데이터를 이용할 수 있다고 생각할 수 있지만 다양한 주변 요인에 따라서 판단이 어려울 수 있다. 예를 들어 섭씨 15도의 온도의 경우 계절, 습도 등 다양한 요인에 따라 사람이 느끼는 기준은 변할 수 있기 때문이다. UbiSOA, Click-Script 등의 연구에서는 사용자가 컨텍스트를 정의하기 위해 사용 가능한 센서를 선택하고 기준 값을 입력하고 연산식을 표현한다. JIGSAW는 퍼즐 조각 형태로 미리 정의된 기능을 사용할 수 있지만 관계연산 및 산술 연산자를 지원하지 않아 세부적인 서비스 정의가 어렵다. 그러나 제안하는 방식은 미리 정의된 기본 컨텍스트를 제공함으로써 사용자는 자신이 원하는 상황을 선택하고 그 상황을 정의하기 위한 추가적인 지식을 요구하지 않는다.

4.2 컨텍스트 수정 및 개인화 제공

사람마다 특정 현상에 대해 느끼는 기준이 다르므로 일반적인 기준을 사용자에게 맞추어 수정 혹은 개인화 할 필요가 있다. 예를 들어 한 사용자는 40lux의 조도에서 어둡다고 느끼는 반면 다른 사용자는 어둡지 않다고 느낄 수 있기 때문이다. 따라서 손쉽게 기준에 정의한 컨텍스트를 수정하고 사용자마다 다른 기준의 컨텍스트를 사용할 수 있어야 한다. UbiSOA, Click-Script, JIGSAW 모두 사용자 별로 컨텍스트를 관리하지 않으며 수정을 위해서는 새로운 컨텍스트를 생성해야 한다. 또한 이는 미리 정의된 컨텍스트의 수정을 위해서 온톨로지에 직접 프로그래밍 하여 추가해야 한다. 그러나 제안하는 방식은 온톨로지에 인스턴스 형태로 사용자별 컨텍스트를

생성, 관리하고 클래스의 조건을 통해 전체적인 컨텍스트 유형을 정의, 관리한다. 이를 통해 사용자는 기 정의된 컨텍스트를 수정, 개인화 하여 사용할 수 있다.

4.3 컨텍스트 매시업 제공

서비스 조건은 매우 다양하기 때문에 이것을 표현하기 위해서는 기존에 존재하는 컨텍스트의 재사용 및 매시업이 필요하다. UbiSOA에서는 부분적으로 기존에 정의한 컨텍스트간의 연결을 통해 새로운 컨텍스트의 생성이 가능하다. 하지만 Click-Script 와 JIGSAW 는 컨텍스트 간의 연결을 통해 새로운 컨텍스트를 생성하는 것이 아니라 전체적인 정의를 통해 컨텍스트를 생성한다. 그러나 제안하는 방식은 컨텍스트간의 계층적인 구조를 통해 하위 컨텍스트를 가질 수 있다. 따라서 사용자는 기본 컨텍스트를 이용하거나 수정, 및 개인화된 컨텍스트를 재사용 하여 새로운 컨텍스트를 정의할 수 있다.

4.4 환경 및 조건 재사용 가능

비슷한 상황 및 환경을 재사용 하여 새로운 서비스를 정의 할 필요가 있다. 예를 들어 기존의 서비스가 A 라는 상황에 B 라는 동작을 제공했을 때 같은 상황에 다른 장소의 C 라는 동작을 제공받길 원하는 경우이다. UbiSOA 와 JIGSAW 의 경우 기존에 정의한 컨텍스트를 모듈형태로 재사용할 수 있다. 하지만 Click-Script 의 경우 환경 / 서비스 구분 없이 한 화면에서 모두 정의해야 하므로 매번 환경 및 서비스를 정의해야 한다. 또한 환경

(예: 거실, 부엌, 침실 등)이 수정되거나 사용자 별로 환경이 변화할 수 있는데 제안하는 방식에서는 온톨로지의 Place 와 Object 클래스로 환경 설정 정보를 관리하여 상황의 재사용뿐만 아니라 환경의 변화에 적응할 수 있다.

4.5 순차적 서비스 정의 기능 제공

상황인지 서비스를 정의하기 위해서는 다양한 환경, 관련 센서, 현상에 관한 기준 값, 서비스 행위 및 절차 등 수많은 사항을 고려해야 한다. 일반적인 사용자는 어떤 항목을 우선적으로 고려해야 할지 생각하기 어려우며 이는 복잡한 서비스의 경우 더 심각하다고 할 수 있다. 또한 각 플랫폼의 처리 과정에 따라 고려해야 할 사항 및 순서는 달라질 수 있으며 이에 따라 제공되는 서비스는 사용자의 의도 및 요구사항을 만족시키지 못하는 경우가 발생할 수 있다. 이에 대해 관련연구 모두 서비스의 흐름에 관계없이 한 화면에서 서비스를 정의하는 방식만을 제공한다. 그러나 본 논문에서는 서비스 정의 마법사 방식을 제안하여 사용자가 결정해야 할 사항을 순차적으로 제시하여 사용자의 혼란 및 오류를 줄여준다. 또한 앞의 상단처럼 현재 처리과정을 확인할 수 있게 하여 사용자가 직관적으로 서비스 정의방식을 이해할 수 있도록 한다.

4.6 프로그래밍 능력 불필요

일반 사용자는 PHP, 자바스크립트 같은 프로그래밍 능력이 없고 if, than, else 와 같은 조건문, for 또는 while 과 같은 제어문의 이해와 응용능력이 부족하다. UbiSOA 는 단위 모듈마다 PHP 기반의 프로그래밍이 필요하며 모듈

간의 연결을 통해 서비스를 정의한다. Click-Script 의 경우 블록간의 연결 및 연산자를 사용하여 서비스를 정의하며 새로운 블록을 정의하기 위해 자바스크립트 프로그래밍이 요구된다. 이러한 방법은 플로우차트의 작성이 익숙하지 않은 사용자들에게 어려울 수 있으며 연산자의 연결 시 시스템의 이해가 요구된다. JIGSAW 의 경우 단위 모듈을 퍼즐조각의 그림으로 표현하여 프로그래밍 능력이 요구되지는 않지만 표현의 다양성이 부족하며 퍼즐 조각의 의미를 다르게 이해 할 가능성이 있고 정보의 전달 및 조건의 설정을 구분하기 힘들다. 그러나 제안하는 방식은 기본 컨텍스트를 선택하고 필요에 의해 기준 값을 변경하고 원하는 동작을 선택하는 방식으로 일반적인 사용자의 프로그래밍 능력이 요구되지 않으며 순차적으로 서비스를 정의할 수 있다.

제 5 장 결론 및 향후 연구

다양한 사물들이 연결되어 있는 사물 인터넷 환경에서 각각의 사물들과 통신을 하면서 에너지 소모율을 줄이고자 하는 연구가 제안되었다. 그 중에서도 자원이 한정적인 환경에서 통신하기 위한 프로토콜로 CoAP가 HTTP 프로토콜 대체로 표준화되었다. CoAP의 특징으로는 한정적인 자원 환경에서 작은 크기의 메시지로 통신을 하여 에너지 소모를 낮춘다. 또한 멀티캐스트 통신을 제 공함으로써 다수의 사물들과 동시에 통신할 수 있는 특징이 있다. 하지만 멀티캐스트 통신을 할 때, 서로 다른 네트워크 환경간 통신시 다수의 응답 메시지에 대한 병목 현상이 생기기 마련이다. CoAP에서는 이 문제에 대해 다루지 않고 있으며 트래픽 오버헤드가 발생하게 된다. 또한 멀티캐스트 통신을 하기에 앞서 그룹을 미리 생성해야 하는데 그룹의 생성은 항상 사용자가 직접 그룹에 등록해야 한다.

이러한 문제점들을 해결하기 위하여 사물인터넷 장비 탐색이나 각종 사용자가 처한 상황인지 서비스를 위해서, CoAP 프록시 게이트웨이와 시맨틱 사물 조합 시스템을 제안하였다. 서로 다른 네트워크 환경인 HTTP와 CoAP 환경 간에 메시지를 주고 받으며 멀티캐스트 통신의 병목 현상을 처리하기 위한 프록시 게이트웨이를 제시하였다. 또한 시맨틱 질의를 기반으로 사용자의 질의를 분석하고 사용자의 요구를 정확히 반영하는 그룹을 자동 조합하는 방법을 제시하였다. 그 결과 높은 정확도를 유지하며 사용자의 요구를 정확히 반영하는 그룹을 자동으로 조합하게 되었으며 기존에 수동으로 그룹을 만든 방법보다 편리성과 시간을 적게 들일 수가 있었다. 자동으로 조합

한 그룹을 이용하여 멀티캐스트 통신을 하게 되는데, CoAP 프록시 게이트웨이에서 다수의 응답 메시지들을 한 개의 메시지로 처리하게 된다. 그런 요인으로 트래픽 데이터의 감소와 응답 시간을 감소하게 되어 병목 현상을 피할 수가 있다.

그러나 제안하는 CoAP 프록시 게이트웨이와 전체적인 시스템에서 다양한 종류의 사물들을 처리하기 위하여 더 많은 요소들을 고려해야 한다. 예를 들어 조명을 작동할 때 있어서 어떤 조명은 on/off로 작동을 하지만 어떤 조명은 디밍 방식으로 수치에 의하여 작동을 할 수가 있다. 이와 같이 두 가지의 작동 값을 갖는 사물이 한 그룹에 있을 경우를 처리하기 위하여 동시 처리 방안이 필요하다. 또한 다양한 운영 및 전개를 할 수 있는 온톨로지 설계를 통해 사용자에게 다양한 기능을 제공할 필요가 있다. 제대로 구축된 큰 환경을 구성하여 다양한 응용의 실험 결과를 보여줄 필요가 있다.

사용자가 지금 현재 어떤 상황에 처해 있는지를 알고자 한다면 센서에서 제공하는 정보들을 활용하여 알아내고, 추정 혹은 상태를 정확히 알 수 있을 것이다. 그러나 각 센서 장비에서 올라오는 데이터들은 각각의 장비마다 구조, 문법, 시스템, 그리고 각각이 제공하는 의미 자체가 서로 다를 것이다. 또한 사용자의 다양한 욕구를 만족시키기 위해 사용자가 원하는 서비스를 정의하는 환경에 관한 요구가 증대되고 있다. 하지만 기존 연구의 경우 서비스를 정의하기 위해 서비스의 전체적인 흐름 및 서비스 조건을 정의하기 위해 필요한 센서 및 기준 값, 이를 표현하기 위한 프로그래밍 능력이 요구되는 등 일반적인 사용자에게는 다소 어렵고 복잡한 환경을 제공하였다. 따라

서 여기에서는 앞서 설명한 사물인터넷의 경향과 추세 등을 통해 인지되는 상황 데이터를 통해 그 상황을 파악하고 그에 적합하게 사물인터넷 서비스를 제공할 수 있게, 사용자에게 적합한 온톨로지 기반 서비스를 제공하고자 한다. 그 구성은 장비에서 제공하는 센싱 데이터를 컨텍스트 정보로 바꾸는 온톨로지 기반 시멘틱 데이터 처리 메커니즘과 사용자들이 편리하게 이용 가능하도록 하는 부분으로 구성된다.

온톨로지 기반 시멘틱 센서 데이터 처리 메커니즘은 오브젝트 및 서비스 추상화를 통해 문법 및 구조적 측면의 이질성을 제거하였으며 시멘틱 어노테이션을 통하여 추상화 된 오브젝트 데이터를 기반 온톨로지에 매핑하여 상황 정보로 전환하였다. 또한 상황의 인지를 위해 기반 온톨로지에 정의된 컨텍스트 조건을 통해 현 상황을 판단하는 추론을 수행하고 서비스 실행단계를 통해 서비스 조건을 만족할 시 해당되는 액추에이터를 제어한다. 각 처리 단계에 따라 Java 기반의 Jena 프레임워크를 통해 처리된 온톨로지의 상태와 표현 언어를 제시하여 데이터 처리 단계를 설명하였다.

또한 사용자 친화적 서비스 정의환경은 서비스 환경 정의를 위한 장소 설정단계 및 오브젝트 설정단계, 서비스 정의를 위한 오브젝트 및 기능 선택 단계 및 상황설정 단계로 이루어지며 웹 기반 UI를 통해 일반 사용자의 서비스 정의를 용이하게 하였다. 본 논문에서 제안한 메커니즘 및 서비스 정의 환경을 통해 사용자는 보다 손쉬운 서비스 정의가 가능하며 플랫폼 관점에서는 보다 정확하게 사용자의 요구를 파악하고 처리하여 다양한 환경의 상황인지 서비스 제공이 용이할 것으로 생각된다. 추후에는 Zigbee, Bluetooth, Wifi 등의 다양한 인터페이스를 통해 전달되는 오브젝트의 데이터

를 실제적으로 추상화 하는 방법과 다양한 환경의 빠른 검색 및 서비스 확장을 위해 서로 다른 환경 온톨로지의 적용 기술에 관한 연구가 필요할 것으로 생각된다. 또한 다양한 실질적인 방식들을 고민하고 제안해 보고자 한다.

그리고, 보안 적인 측면에서 하나 덧붙이면 사물인터넷은 다양한 기술의 복합적 특성 외에, 사물인터넷 서비스가 기존의 서비스와 다른 특성을 가진다. 즉, 기존의 응용 서비스는 단일 사업자가 주도하는 vertical application market 이었지만, 사물인터넷 서비스는 다양한 주체가 공존하는 horizontal market 특성을 가진다. 이 때문에 여러 주체 간 프라이버시 보호 책임과 권한 선정의 어려움이 생긴다. 즉, 사물인터넷 환경은 센서 디바이스 공급자와 통신/네트워크 공급자, 서비스 개발자, API 개발자, 플랫폼 공급자, 데이터 소유자 등 다양한 주체가 존재하기 때문에 보안/프라이버시 보장이 어렵게 되는 것이다. 물론 프라이버시를 보장하는 기술은 프라이버시 보호를 위한 법이나 체계에서 프라이버시 보호를 위한 범위 및 대상을 명확히 정의해야 하는데, 현재 개인정보보호법(한국)과 Consumer Protection/Data Protection Act (미국)에서는 사물인터넷 환경에 적용 가능한 구체적인 프라이버시 보호 체계가 정의되지 않았다는 문제가 존재한다.

참고문헌

<국내 >

- 김용진, 김경수, “소셜 M2M 서비스 개발” 한국통신학회지 제 32 권 제 2 호, 2015.1
- 남궁정일, 김용진, “ 소셜 M2M 서비스를 위한 지능형 M2M 플랫폼”, 한국통신학회지 제 30 권 제 8 호, 2013.7
- 고영준, 김용진, “IoT 상호운용성 및 서비스 개발” ETRI 제 32 권 제 2 호, 2015.1
- 김호원, “사물인터넷 환경에서의 보안/프라이버시 이슈” Special Theme_ ICT 융복합 시대의 사이버 보안
- 류중경, 김종훈, 김재권, 이정현, 정경용, “상황인식 기반의 유헬스 환경정보 서비스”, 한국콘텐츠 학회논문지, 제 11 권, 제 7 호, pp.21-29, 2011.
- 류중경, 정경용, 김종훈, 임기욱, 이정현, “서비스 온톨로지 기반의 상황인식 모델링을 이용한 추천”, 한국콘텐츠학회논문지, 제 11 권, 제 2 호, pp.22-30, 2011.
- 안병익, “IoT 의 수평적 모델과 초연결망 시대”, Fortune korea, 2014. 11.
- 이영설, 양건모, 조성배, “상황인지 스마트 TV 서비스를 위한 온톨로지 기반 스마트 TV 환경 모델링”, 한국정보과학회학술발표회, pp. 195-197, 2012.
- 조건희, “사물 인터넷 환경에서 CoAP 을 이용한 시맨틱 그룹 통신” 연세대학교, 2015.8
- 최환석, 이준영, 양나리, 이우섭, “상황인지 IoT 서비스 제공을 위한 온톨로지 기반 사용자 친화적 서비스 환경” 국립한밭대학교, 2014.7

ETRI 김선태, 정종수, 송준근, 김해용, "IoT 단말 플랫폼 동향 및 생태계 구축",
2014

<국외 >

D. Miorandi, S. Sicari, F. De Pellegrini and I. Chlamtac, "Internet of Things: Vision, Applications and Research Challenges," *Ad Hoc Networks*, Vol. 10, No. 7, pp. 1497-1516, 2008.

BACnet, <http://www.bacnet.org/>.

Knx association, <http://www.knx.org/>.

EnOcean, <http://www.enOcean.com/en/home/>.

L. Pérez-Lombard, J. Ortiz and C. Pout, "A review on buildings energy consumption information," *Energy and buildings*, Vol. 40, Issue. 3, pp. 394-398, 2008.

Z. Shelby, K. Hartke, C. Bormann and B. Frank, "The Constrained Application Protocol (CoAP)," *Internet Engineering Task Force*, <https://datatracker.ietf.org/doc/rfc7252/>.

A. Rahman and E. Dijk, "Group Communication for the Constrained Application Protocol (CoAP)," *Internet Engineering Task Force*, <https://datatracker.ietf.org/doc/rfc7390/>.

Z. Shelby, S. Krco and C. Bormann, "CoRE Resource Directory," <http://tools.ietf.org/html/draft-shelby-core-resource-directory-04/>.

I. Ishaq, J. Hoebeke, F. Van den Abeele, I. Moerman and P. Demeester, "Group communication in constrained environments using CoAP-based entities," In *Proc. the 2013 IEEE Int'l Conf. Distributed Computing in Sensor Systems*, pp. 345-350, 2013.

- I. Ishaq, J. Hoebeke, F. Van den Abeele, J. Rossey, I. Moerman and P. Demeester, "Flexible Unicast-Based Group Communication for CoAP Enabled Devices," *Sensors*, Vol. 6, pp. 9833-9877, 2014.
- M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox and K. Taylor, "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 17, pp. 25-32, 2012.
- H. Stephan, S. Alexandru, B. Marin and C. Francois, "A Domain Model for the Internet of Things," In *Proc. the 2013 IEEE Int'l Conf. on Internet of Things (iThings)*, pp. 411-417, 2013.
- M. Paolucci, T. Kawamura, T. R. Payne and K. Sycara, "Semantic matching of web services capabilities," *The Semantic Web—ISWC 2002*. Springer Berlin Heidelberg, pp. 333-347, 2002.
- M. Ruta, F. Scioscia, G. Loseto, F. Gramegna, A. Pinto, S. Ieva and D. Sciascio, "A Logic-based CoAP Extension for Resource Discovery in Semantic Sensor Networks," In *the 11th Int'l Semantic Web Conference*, pp. 17-32, 2012.
- J. Mäenpää, J. J. Bolonio and S. Loreto, "Using RELOAD and CoAP for wide area sensor and actuator networking," *EURASIP Journal on Wireless Communications and Networking*, Vol. 1, pp. 1-22, 2012.
- E. Dijk, A. Rahman, T. Fossati, S. Loreto and A. Castellani, "Guidelines for HTTP-CoAP Mapping Implementations," <https://datatracker.ietf.org/doc/draft-ietf-core-http-mapping/>.
- "Californium (Cf) CoAP framework in Java," ETH Zurich, <http://people.inf.ethz.ch/mkovatsc/californium.php/>.
- jCoAP, <http://code.google.com/p/jcoap/>.
- WebThings, <https://github.com/koanlogic/webthings/>.

Nodes, <http://code.google.com/p/nodes/>.

"Squid 3.1.9 with transparent HTTP-CoAP mapping module,"

<http://telecom.dei.unipd.it/iot/>.

G. Bovet and J. Hennebert, "A Distributed Web-based Naming System for Smart Buildings," In Proc. of IEEE Int'l Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1-6, 2014.

G. Bovet, A. Ridi and J. Hennebert, "Toward Web Enhanced Building Automation Systems," In Big Data and Internet of Things: A Roadmap for Smart Environments, Springer Int'l Publishing, pp. 259-283, 2014.

M. Kovatsch, "CoAP for the web of things: from tiny resourceconstrained devices to the web browser," In Proc. of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication. ACM Press, pp. 1495-1504, 2013.

S. Chun, J. Jung, X. Jin, G. Cho, J. Shin and K.-H. Lee, "Short paper: Semantic URI-based event-driven physical mashup," In Proc. of the IEEE World Forum on Internet of Things (WF-IoT), pp. 195-196, 2014.

Charith Perera, Chi Harold Liu, Srimal Jayawardena, MinChen, "Context-aware Computing in the Internet of Things: A Survey on Internet of Things From Industrial Market Perspective" in IEEE Access, JAN. 2015

AllSeen Alliance, "Intoduction to the AllJoyn™ Framework", 10 Dec 2013, available at <https://allseenalliance.org/developer-resources/alljoyn/docsdownloads>.

T. Koshimizu, D. Steahle, "An Effective and Secure Scheme for Accommodation of IEEE802.15.5e/g FAN into 3G/LTE network and its simulation evaluation", IEICE, 2013

S. Husain, A. Kunz, J. Song, and T. Koshimizu, "Recent Rends in Standards related to the Internet of Things and Machine-to-machine

- communications" in JICCE, Dec. 2014
- Blase Ur, Elyse McManus, Melwyn Pak YongHo, Michael L. Littman, "Practical Trigger-action programming in the Smart Home", CHI 2014, Apr.
- M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox and K. Taylor, "The SSN ontology of the W3C semantic sensor network incubator group," Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 17, pp. 25-32, 2012.
- H. Stephan, S. Alexandru, B. Marin and C. Francois, "A Domain Model for the Internet of Things," In Proc. the 2013 IEEE Int'l Conf. on Internet of Things (iThings), pp. 411-417, 2013.
- M. Paolucci, T. Kawamura, T. R. Payne and K. Sycara, "Semantic matching of web services capabilities," The Semantic Web—ISWC 2002. Springer Berlin Heidelberg, pp. 333-347, 2002.
- M. Ruta, F. Scioscia, G. Loseto, F. Gramegna, A. Pinto, S. Ieva and D. Sciascio, "A Logic-based CoAP Extension for Resource Discovery in Semantic Sensor Networks," In the 11th Int'l Semantic Web Conference, pp. 17-32, 2012.
- J. Mäenpää, J. J. Bolonio and S. Loreto, "Using RELOAD and CoAP for wide area sensor and actuator networking," EURASIP Journal on Wireless Communications and Networking, Vol. 1, pp. 1-22, 2012.
- E. Dijk, A. Rahman, T. Fossati, S. Loreto and A. Castellani, "Guidelines for HTTP-CoAP Mapping Implementations,"
<https://datatracker.ietf.org/doc/draft-ietf-core-http-mapping/>.
- "Californium (Cf) CoAP framework in Java," ETH Zurich,
<http://people.inf.ethz.ch/mkovatsc/californium.php/>.
- jCoAP, <http://code.google.com/p/jcoap/>.
- WebThings, <https://github.com/koanlogic/webthings/>.

Nodes, <http://code.google.com/p/nodes/>.

“Squid 3.1.9 with transparent HTTP-CoAP mapping module,”
<http://telecom.dei.unipd.it/iot/>.

V. Huang and M. K. Javed, “Semantic sensor information description and processing,” *Sensor Technologies and Applications (SENSORCOMM)*, pp.456-461, 2008.

A. P. Sheth, “Changing focus on interoperability in information systems: from system, syntax, structure to semantics,” *Interoperating Geographic Information Systems*, pp.1-30, 1999.

B. Christophe, “Semantic profiles to model the web of things,” *IEEE SKG*, pp.51-58, 2011.

B. Ostermaier, F. Schlup, and K. Romer, “WebPlug: A Framework for the Web of Things,” *Pervasive Computing and Communications Workshops*, pp.690-695, 2010.

D. Guinard and V. Trifa, “Towards the web of things: web mashups for embedded devices,” *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009)*, 2009.

C. Iván, F. J. Martínez, M. S. Familiar, and L. López, “Knowledge-Aware and Service-Oriented Middleware for deploying pervasive services,” *Journal of Network and Computer Applications*, pp.562-576, 2012.

D. Guinard, “A web of things application architecture – integrating the real-world into the web,” *PhD thesis No.19891, ETH Zurich*, 2011.

E. Avilés-López and J. A. García-Macías, “Mashing up the Internet of Things: a framework for smart environments,” *EURASIP Journal on Wireless Communications and Networking*, 2012.

S. Han, G. Lee, and N. Crespi, “Semantic context-aware service composition for building automation system,” *IEEE Transactions on Industrial*

Informatics, Vol.10, No.1, pp.752-761, 2013.

C. H. Quyet, "A hybrid semantic energy saving service for cloud-based smart homes," M.S thesis, School of engineering and technology thailand, unpublished, 2013.

<http://clickscript.ch>

T. Rodden, A. Crabtree, T. Hemmings, B. Koleva, J. Humble, K. Akesson, and P. Hansson, "Configuring the Ubiquitous Home," The Sixth International Conference on Designing Cooperative Systems, pp.227-241, 2004.

ABSTRACT

A Study on the Semantic Information Interoperability for Search for Internet of Things

SeokWon Jung

Major in Industrial Information Management

The Graduate School of Engineering

Yonsei University

Supervised by Prof. WooJu Kim

Quite unnoticeably, Internet has been positioned deep in our life, while helping individuals to be interconnected via computer with each other to share the information. Over not so long time, the devices have been interconnected increasingly with each other via Internet, and now, various home appliances, wearables and even the home electric bulbs and diverse sensors are interconnected with Internet. The world created thus is the world of IoT (Internet of Things). As the types and performances of the devices interconnected with each other become more and more diverse, the key to the success of the world of IoT seems to be how to make the numerous devices compatible and interlocked.

Currently, various standards consultative bodies and industrial consortiums endeavor to be positioned as a standard: ISO/IEC, JTC 1/WG10, ITU-T SG 20, oneM2M, Thread Group, Allseen Alliance, OIC(Open Interconnect Consortium), IIC, etc. They compete fiercely with each other or cooperate with each other to establish the technological standards including protocols.

The environment for the IoT provides the services required to make all things interconnected with a IP-based Internet. As the IoT technology develops, the number of the things interconnected increases throughout the world. Accordingly, the quantity of the things and communication data that should be managed increases greatly, and thus, the consumption of power and others increases. Hence, many researchers attempt to find the effective ways to manage the things and data to process them effectively. The most outstanding protocol among them is similar to the web protocol published by Core WG of IETF. It is CoAP(Constrained Application Protocol), a HTTP protocol improved from the existing HTTP standards. Although CoAP is similar to HTTP, it is different from HTTP in terms of message type and size, and therefore, it is more suitable to the IoT environment. In addition, CoAP supports the multi-cast, while managing and controlling multiple things all at once. However, each organization uses their own standards, while IoT service companies or IoT device suppliers establish their own services or devices. The results are fragmented IoT services, which should make it necessary

to develop the technologies relevant to the interoperability among IoT service terminals and platforms.

The purpose of this study is to research into the ways to discovery and interoperability among IoTs in such diverse and heterogenous IoT environments and thereby, suggest some solutions. In particular, this study reviews the semantic IoT service platform technology as a way to improve the search and interoperability among IoT services. The semantic IoT service platform is an IoT service platform based on the information to allow the ordinary users to use the sensing information regardless of time and place. In short, it allows for a personally customized real-time sensing information service. Whether this IoT interoperability can be activated or not will be a decisive factor affecting the promotion of the service and activation of the market

Key words: IoT, CoAP, semantic combination, discovery, interoperability