

# A Semantic Service Discovery System for Smart-Cities

Chang Ho Yun<sup>†</sup> · Jong Won Park<sup>†</sup> · Hae Sun Jung<sup>\*\*</sup> · Yong Woo Lee<sup>\*\*\*</sup>

## ABSTRACT

In Smart-cities, various types of integrated services must be linked to provide services to applications. Therefore, flexibility must be ensured between services so that various services can be efficiently provided. In order to secure the flexibility among services, it is very important to have a function to dynamically discover and invoke a desired service by searching for a semantic service by reflecting a recognized context through real-time context-aware in smart-cities. To date, quite a number of semantic service discovery techniques have been developed. However, they have not been verified as suitable for use in the smart-city domain. In this study, we tried to verify the existing ones to use a suitable one. We tested most of existing semantic service discovery techniques, but we found that none of them is suitable to our research. Therefore, we developed our own semantic service discovery technique. This paper introduces our work and presents the performance evaluation results that demonstrate that our developed works well and show good performance. For the performance evaluation, the experimental system was actually constructed and the real performance was measured. In the experiment, we implemented the semantic service discovery scenario that dynamically searches and calls the services needed to provide fire accident management services in smart cities.

**Keywords :** Smart-City, Context-Aware Computing, Semantic Service Discovery, Semantic Matchmaking, Ontology

## 스마트시티를 위한 시맨틱 서비스 디스커버리 시스템

윤창호<sup>†</sup> · 박종원<sup>†</sup> · 정혜선<sup>\*\*</sup> · 이옹우<sup>\*\*\*</sup>

## 요약

스마트-시티에서는 다양한 종류의 정보들을 연계하여 서비스를 제공해야 한다. 따라서 서비스 간에는 유연성이 확보되어야 다양한 서비스를 효율적으로 제공할 수 있다. 서비스 간의 유연성이 확보되기 위해서, 스마트-시티에서는 실시간 상황인식을 통하여 인식된 상황을 반영하여, 시맨틱 서비스 검색을 함으로써, 동적으로 원하는 서비스를 발견하고 호출하는 기능이 매우 중요하다. 현재까지 꽤 많은 시맨틱 서비스 디스커버리 기법들이 개발되었다. 그런데, 이들은 스마트-시티라는 특정 도메인에 사용하기에 적절한지 검증이 되지 않았다. 본 연구에서는 기존의 것들을 일일이 다 검증하여 적합한 것을 사용하고자 하였다. 스마트-시티에서 우리와 같은 시도를 한 연구나 연구결과를 찾을 수 없어서, 일일이 검증을 하였으나, 우리의 스마트-시티 시스템을 위한 요구사항들을 만족시키기에는 매우 부족하거나 적절치 못했다. 그래서, 스마트-시티라는 우리의 사용영역에 맞는, 독자적인 시맨틱 서비스 디스커버리 기법들을 개발하지 않을 수 없었다. 본 논문은 이를 소개하고 성능평가 실험을 통하여 우리의 연구 개발시스템이 잘 작동하였으며, 우수한 성능을 보였음을 입증하는 결과와 이의 상세한 분석을 제시한다. 성능평가를 위하여, 실제로 실험시스템을 구축하고 성능을 측정하였다. 본 실험결과는 실제 스마트-시티를 구축하는데 매우 유용하게 쓰일 수 있다.

**키워드 :** 스마트-시티, 상황인식컴퓨팅, 시맨틱 서비스 디스커버리, 시맨틱 매치-메이킹, 온톨로지

\* Part of this study was supported by the Seoul Research and Business Development Program, Smart City Consortium (10561) and Seoul Grid Center. Part of this work was supported by the 2014 research fund of the University of Seoul. Part of this work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korea Ministry of Science, ICT & Future Planning (2015R1C1A1A02036461).

<sup>†</sup> 비회원: 서울시립대학교 전자전기컴퓨터공학과 박사과정

<sup>\*\*</sup> 정회원: 서울시립대학교 전자전기컴퓨터공학과 연구교수

<sup>\*\*\*</sup> 정회원: 서울시립대학교 전자전기컴퓨터공학과 교수

Manuscript Received: February 20, 2017

Accepted: March 7, 2017

\* Corresponding Author: Yong Woo Lee(ywlee@uos.ac.kr)

## 1. 서론

스마트-시티는 도시에 유비쿼터스 컴퓨팅[1], 사물인터넷 (Internet of Things: IoT)을 비롯한 최신 ICT 기술을 융합한 것으로, 지능적으로 도시를 통합 관리할 수 있게 하여, 도시민의 삶의 질을 향상시킨다는 목적을 달성하게 해 준다 [2]. 스마트-시티는 다양한 종류의 서비스들을 연계하는 통합서비스를 제공할 수 있다. 스마트-시티가 제공하는 다양한 서비스 중에서 주어진 상황에 최적의 서비스를 찾는 일

이 종종 요구된다. 이를 위하여, 서비스 디스커버리 기능이 필요하다. 스마트-시티에서 수집한 데이터를 시맨틱하게 분석하여 주어진 상황에 최적의 서비스를 찾는 시맨틱 서비스 디스커버리의 효용성이 점차 증대되어 가고 있다.

본 스마트-시티 사업단은 현재까지 연구된 시맨틱 서비스 디스커버리를 비교 연구하여, 스마트-시티라는 특정 도메인에서 잘 작동할 수 있는, 보다 향상된 시맨틱 서비스 디스커버리를 지원할 수 있는 핵심 알고리즘을 개발하였다. 이 알고리즘을 스마트-시티 사업단의 프로토타입 시스템에 구축하고 실크로드(Silkroad) 시스템이라고 명명하였다. 본 논문은 이 실크로드 시스템을 소개하고 이를 성능 평가한 결과를 제시한다. 이는, 이후에, 유사한 연구와 개발을 하고자 하는 연구자, 개발자들에게 도움이 되고자 함이다.

사용자의 상태를 정확히 분석하고 가장 적합한 서비스를 제공하기 위해서는 우리의 스마트-시티 시스템은 다음의 기능을 가져야 하는 것으로 설정되었다. 첫째, 시스템이 사용자의 상태를 동적으로 모니터링할 수 있어야 한다. 즉, 스마트-시티는 사용자가 필요로 하는 서비스를 능동적으로 제공하기 위해서, 상황 정보를 지능적으로 관리하는 기능이 필요하다. 둘째, 필요한 주위 환경 상황을 고려한 최적의 서비스를 찾아야 한다. 이를 통해 사용자에게 적절한 서비스를 자동적으로 제공할 수 있어야 한다. 마지막으로, 이 이와 같은 시맨틱 서비스 디스커버리를 통한 최적 서비스 기능은 응답시간이 빠를 것, 확장성이 좋을 것, 등등의 우수한 성능을 보유하고 있어야 한다는 것이 목표로 설정되었다.

성능평가를 위하여, 적절한 성능평가 지표를 선정해야 하는 것도 본 논문에서 소개된 연구의 목표의 하나이다. 정밀도(Precision), 재현율(Recall), F1-측정(F1-measure), 응답시간(Response Time)과 같은 성능 평가 지표들이 우리들이 개발한 시스템을 평가하는데 적절한 지표로 선정되었다. 본 논문에서는 이들 평가지표를 이용하여 성능평가를 하고 그 평가결과들을 설명한다. 또한, 본 연구에서는, OWL-S, WSMML 그리고 표준 SA-WSDL과 같은 시맨틱 서비스 포맷을 적합한 서비스 포맷으로 선정하여 사용하였다.

## 2. 관련 연구와 우리의 연구의 비교분석

현재까지 꽤 많은 시맨틱 서비스 디스커버리 기법들이 개발되었다. 그러나 이들은 스마트-시티라는 특정 도메인에 사용하는 데에 적절한지 검증이 되지 않았다. 본 연구에서는 기존의 것들을 일일이 다 검증하여 적합한 것을 사용하고자 하였다. 그러나, 결과적으로 실패하였다. 스마트-시티에서 우리와 같은 시도를 한 연구나 연구결과를 찾을 수도 없었다. 그래서, 본 사업단에서는 스마트-시티라는 우리의 사용영역에 맞는, 독자적인 시맨틱 서비스 디스커버리 기법들을 개발하지 않을 수 없었다. 이 장에서는, 관련연구들을 소개하고, 이런 측면에서 우리가 연구분석하고 사용해보고 경험했던 문제점들을 비판적인 분석 내용으로 함께 신는다.

그리디(Greedy) 알고리즘[3]은 시맨틱 서비스 디스커버리

분야에서 기존에 널리 알려진 알고리즘이다. 시맨틱 서비스 디스커버리에 등급을 부여하여, 등급에 따라 분류된 결과를 제공한다. 그러나 그리디 알고리즘은 서비스 결과에 대한 등급만을 제공하기 때문에 정확성에 있어서 일정한 한계점이 있다는 문제점과 정밀도가 좋지 못하다는 문제점을 보였다.

양분그래프매칭(Bipartite Graph Matching)을 이용한 시맨틱 매치-메이킹 알고리즘[4]은 입력과 출력뿐만 아니라 사전조건과 영향까지 고려하여 최적의 결과를 찾아낸다. 그러나 이 알고리즘 역시 문제점내지는 한계점을 보였다. 사용자의 선호도나 순위를 제공하지 않기 때문이다.

기술로직(Description Logic)을 사용한 알고리즘[5]은 선호도에 따라 순위를 정해주는 기능을 제공한다. 기술로직을 사용한 알고리즘은 사용자의 선호도와 연성 제약조건(Soft constraint)을 반영할 수 있기 때문에, 표현과 기술을 더욱 풍부하게 할 수 있다는 장점이 있다. 그러나, 정확도 문제 등에서 역시 만족스럽지 못하였다.

순위-인스턴스-검색(Ranked Instance Retrieval) 알고리즘[6]은 현재까지 널리 쓰이고 있다. 그러나 이 알고리즘은 입력력 변수만을 사용하여 매치-메이킹을 할 수 있기 때문에, 다양한 환경을 고려해야 하는 스마트-시티에서 사용자의 요청을 효율적으로 반영하기 어려운 단점을 보였다.

우리의 스마트-시티는 위의 여러 문제점 내지는 결점들을 가지지 않는 알고리즘을 필요로 하였다. 그래서, 우리의 독자적인 알고리즘을 개발하였으며, 이를 본 논문에서 소개하고자 한다. 우리가 개발한 알고리즘에서는, 사용자의 위치 정보에 따라 사전조건을 줄 수 있으며, 사용자의 선호도에 따른 순위 지정이 가능하다. 이것이 실크로드 알고리즘의 차별적인 특징점이다.

시맨틱 서비스 디스커버리 기법을 이용한 다양한 시스템들을 소개하는 연구결과들이 발표되고 있다. 그 중의 우리의 연구에 가까운 주요한 것들을 살펴보면 다음과 같다. 다스굽타 등(Dasgupta et al.)[7]은 효율적인 시맨틱 서비스 디스커버리를 위한 하이브리드 멀티에이전트 기반의 분산 플랫폼 스마트-스페이스(SMART-SPACE)를 제안한 바 있다. 이 논문은 스마트-디스커버리(Smart-Discoverer)라는 서비스 디스커버리 알고리즘을 제안하고 있다. 이 논문은, 스마트-디스커버리는 빠르고 확장성 있고 병렬로 동시에 서비스를 찾을 수 있도록 설계되고, IBM 클라우드 클러스터와 NetLogo 시뮬레이션 환경에서 JADE 3.7 에이전트 프레임워크상에서 구현되었다고 설명하고 있다. 이 논문은, 스마트-디스커버리의 정확성을 OWL-S MX와 비교한 성능평가 결과를 신고 있다. 그러나, 이 논문에는 온톨로지 기반의 다양한 의미론적 매치-메이킹에 대한 성능평가가 없다. 시맨틱 매치-메이킹에 대한 구체적인 알고리즘도 공개되지 않았다.

퀸안 등(Qian et al.)[8]은 태스크 기반의 서비스 활용을 지원하는 컴포넌트 기반의 로봇 시스템을 위한 프레임-워크를 제안하고 있다. 이 프레임워크는 자동화된 프로그래머이며, 로봇 서비스의 시맨틱 모델링, 서비스 디스커버리 그리고 동적인 서비스 선택을 지원하기 위한 시맨틱 유사성 매트릭

스 등의 기능을 포함하고 있다. 시맨틱 유사성 매트릭스는 개념적 유사성 측정을 이용하고 있으며, OWL 관계 정의와 매치-메이킹 알고리즘을 기반으로 작동한다고 소개되어 있다. 그러나 이 논문에 소개된 연구결과는 매치-메이킹 결과에 대한 순위 기능을 제공하지 않는다, 그래서, 로봇이 매치-메이킹의 결과를 이용하여 서비스를 선택할 때, 순위가 없으므로, 어느 것을 우선적으로 선택해야 하는지에 대한 가이드라인이 주는 또 다른 메커니즘을 사용해야 하는 단점이 있다.

카남(S. A. Khanam)[9]은 시맨틱 웹 서비스 디스커버리 스키마를 제안하였다. 시맨틱 웹 서비스 디스커버리 스키마는, 사용자의 질의와 웹 서비스 사이의 유사성을 결정하기 위하여, WSDL 명세와 온톨로지 그리고 향상된 헝가리언(Hungarian) 알고리즘을 이용하고 있다. 또한 정밀도와 재현을 그리고 추가적으로 F1-측정을 이용하여 서비스 디스커버리 성능을 향상하고자 하였다. 그러나, 서비스 온톨로지를 이용한 시맨틱 매치-메이킹 측면에서 볼 때, 고정된 변수만 고려가능하다는 단점이 있다. 즉, 다양한 요소에 대한 비교가 불가능하다는 단점이 있다.

위의 연구들과 비교할 때, 본 연구의 장점은 다음과 같다. 첫째, 변수 이외의 서비스형식(ServiceType) 그리고 위치(Location) 등 다양한 요소를 더 정확히 반영할 수 있는 방법을 통하여 더욱 정확한 시맨틱 매치-메이킹 결과를 제공하므로 훨씬 더 나은 결과를 가져올 수 있다. 둘째, 순위 개념을 제공하므로, 선택된 서비스들에 대한 추가 가이드-라인 없이 최선의 선택을 할 수 있다.

다음에서는, 스마트-시티 또는 이와 유사한 프로젝트에서, 우리 연구와 관련된 연구가 있는지 살펴보겠다. HP의 쿨타운(Cooltown) 프로젝트[10]는 가상 세계 상에서의 개체(웹상에 구현되어 있는 장소나 물건)를 실제 세계의 개체로 확장하는 것을 목표로 하고 있다. 하지만 쿨타운에서는 위치 기반으로 하는 서비스 디스커버리를 제공하지만 시맨틱하게 처리하지 않는다.

마이크로소프트사의 이지리빙(Easyliving)[11]은 지능적 환경(Intelligent Environment)을 기반으로 모바일 컴퓨팅 기술 지원하는 것에 목표를 두고 있다. 이지리빙은 디바이스 제어, 내부 로직과 유저 인터페이스의 분리를 위한 서비스 추상화를 제공하지만 시맨틱 서비스 디스커버리를 제공하지는 않는다.

일리노이대학에서 만든 가이아(Gaia)[12]는 물리적 공간과 소프트웨어 인프라가 융합되어 있는 액티브-공간 실현을 목표로 하고 있다. 가이아는 다양한 상황정보를 이용하여 추론하는 기능을 제공하며, 구조상에 있는 다양한 컴포넌트가 각자의 기능을 가지고 서로 상호작용을 통해 상황을 인식하는 서비스를 제공한다. 모바일 어플리케이션의 개발을 돕기 위한 프레임워크를 제공한다. 이 프레임워크의 역할은, 현재의 컨텍스트를 사용하여 자원인식과 컨텍스트인식을 할 수 있게 해주는 것이다. 하지만 시맨틱 디스커버리를 제공하지 않는다.

이와 같이 스마트-시티를 위한 서비스 디스커버리에 대한 연구는 현재 찾아 볼 수 없으므로 본 연구가 세계 최초라고 생각한다. 그동안, 단지, 시맨틱 서비스 디스커버리에 대한 많은 연구가 웹서비스 분야 위주로 이루어져 왔을 뿐이었다.

### 3. 스마트-시티 시스템

Fig. 1은 본 연구에서 개발한 시맨틱 서비스 디스커버리 메커니즘을 보여준다. 명시적인 유저의 개입 없이 현재의 상황인식을 기반으로 어플리케이션이 동작함을 보여주고 있다. 우리가 개발한 시스템은 온톨로지 기반의 지능적 추론 엔진을 사용하여 높은 수준으로 상황을 인식한다. 상황-변환기, 상황-분석기, 상황-관리가 중심이 되어서 이 기능을 수행한다. 인식된 상황 정보를 사용하여 시맨틱 서비스 디스커버리를 수행하여 최적의 서비스를 선택하고, 선택된 서비스를 실행한다. 서비스-발견기, 서비스-실행기가 중심이 되어서 이 기능을 수행한다.

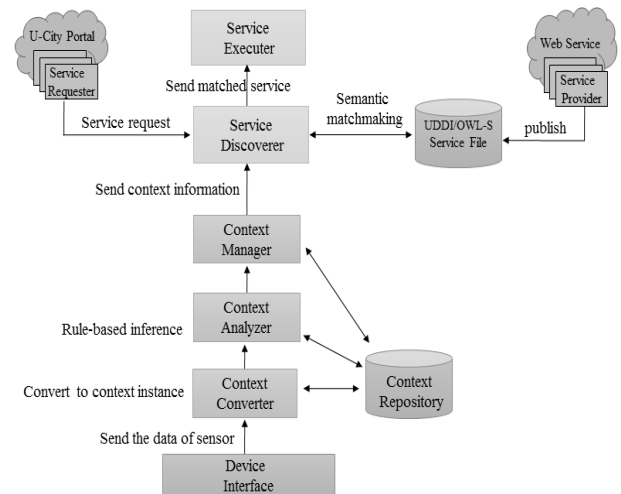


Fig. 1. The Operational Mechanism of Our Semantic Service Discovery

Fig. 1에서, 상황-변환기는 추론을 위하여 센서 데이터를 도메인 온톨로지 기반의 OWL 포맷의 컨텍스트 인스턴스로 변경하는 기능을 수행한다. 상황-분석기는 변경된 컨텍스트 인스턴스와 사용자가 정의한 규칙을 펠릿(Pellet) 추론기[13]에 입력하여 추론을 행하는 기능을 수행한다. 추론된 내용은 알맞은 서비스를 찾기 위한 상황 정보로 제공된다. 추론 시 컨텍스트 인스턴스로 변경된 데이터를 사용한다. 사용자 정의 규칙은 일차논리(First-order logic)형식으로 명시가 되어진다[14]. 이후에, 사용자가 정의한 규칙은 추론 엔진에 업로드 되어서, 컨텍스트 인스턴스와 함께 처리된다. 상황-관리는 모든 컨텍스트를 관리하는 기능을 한다.

서비스-발견기에서 상황정보를 요청할 때, 상황-관리는 질의문을 생성하고 실행하여 상황-저장소로부터 상황정보를

획득한다. 상황-저장소에는 센서 데이터, 컨텍스트 인스턴스, 그리고 추론된 컨텍스트가 저장된다. 서비스-발견기는 서비스를 찾고 서비스-실행기는 서비스를 추론된 컨텍스트 정보에 따라 서비스를 실행한다. 서비스-발견기에서는 서비스 요청자가 요청하는 서비스, 예를 들면 화재 서비스와 같은 요청에 응하여 상황-관리가 제공하는, 화재 위치 등등의 상황 인식된 컨텍스트를 조합하여 OWL-S 포맷의 서비스 명세를 만든다. 서비스-발견기의 WSDL2OWLS[15]는 서비스 제공자가 제공하는 명세를 수집하고 서비스-발견기에서 매치-메이킹을 할 수 있도록 OWL-S 포맷으로 변경한다. 이와 같이, 서비스 요청자의 요청에 의하여 서비스 명세가 서비스 온톨로지로 기술된다. 이제, 서비스-발견기의 시맨틱 매치-메이커는 서비스 도메인 온톨로지를 사용하여 시맨틱 매치-메이킹을 수행한다. 서비스-발견기는 실크로드를 사용하여 시맨틱 매치-메이킹을 수행한다. 개발된 실크로드 알고리즘은 그리디 알고리즘[3]의 모든 기능을 제공하며 시맨틱 매치-메이킹 된 결과는 순위를 매겨서 제공한다. 실크로드 알고리즘은, 입력력 매개변수, 서비스형식, 그리고 위치를 사용하여 서비스 요청자의 요청에 맞는 서비스를 찾는다. 본 연구에서는 OWL-S의 프로파일 속성을 확장한 확장 OWL-S를 개발하였다. 위의, 서비스형식과 위치를 기술하는데, 확장 OWL-S가 사용된다.

#### 4. 시맨틱 서비스 디스커버리

우리의 시스템은 제나(Jena)[16]를 추론기로 사용한다. 본 연구에서는 상황을 추론하기 위해 사용자가 정의한 일차논리를 사용하였다. 서비스형식은 미리 정의된 서비스형식으로서, 서비스-발견기가 서비스 매칭을 수행하기 이전에 서비스를 분류하기 위한 목적으로 사용된다. Fig. 2는 서비스형식을 위하여 본 연구에서 사용한 서비스 도메인 온톨로지 모델의 한 예이다. 여기서, 서비스형식은 화재(Fire), 환경(Environment), 교통(Traffic) 클래스로 분류되며 화재의 서비스 클래스는 소방서(Firestation), 병원(Hospital), 경찰(Police)로 구성된다.

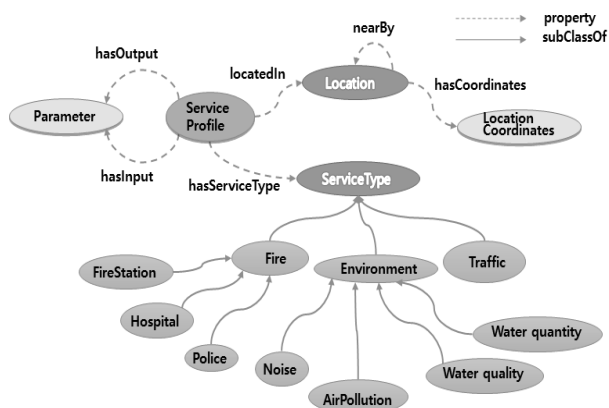


Fig. 2. Service Domain Ontology Model

우리는 다음의 다섯가지 요소를 반영하여 시맨틱 서비스 디스커버리 방법론을 개발하고 구현하였다. 1) 서비스를 어떤 방식으로 시맨틱하게 기술할 것인가? 2) 어떻게 서비스와 컨텍스트 정보를 기술할 것인가? 3) 어떻게 유저나 소프트웨어 에이전트가 서비스를 요청할 것인가? 4) 어떻게 시맨틱 서비스 디스커버리가 요청을 만족하는 서비스를 찾을 것인가? 5) 매칭 된 서비스의 등급[17]을 어떻게 부여할 것인가?

Fig. 3은 실크로드의 플로우 차트를 보여준다. 서비스 제공자는 서비스를 WSDL을 사용하여 기술되도록 만들어졌다. 서비스형식과 위치는 서비스 제공자에 의해 WSDL에 명시되도록 만들어졌다. WSDL에서는 구문(Syntactic)기반으로 기술되기 때문에 서비스를 시맨틱하게 찾을 수 없다. 그래서, 우리는, 시맨틱하게 서비스를 찾기 위하여 WSDL 포맷의 데이터를 OWL-S 포맷으로 변환하고, 변환된 데이터에, WSDL2OWLS 방법을 적용하여 시맨틱하게 원하는 서비스를 찾도록 우리의 시스템을 개발하였다. 사용자의 요청 역시 XML2OWLS 방법을 사용하여 OWL-S 포맷으로 변경된다.

시맨틱 매치-메이커는 서비스 요청에 의거하여 제공되는 서비스 중에서 최적의 서비스를 찾은 임무인 매치-메이킹 임무를 수행한다. 서비스형식을 매칭하기 위하여, 서비스형식 도메인 온톨로지가 방법론으로 사용되었다. 서비스 매치-메이커는 입력력 매개변수와 서비스형식이 지정하는 값을 사용하여 매치-메이킹 임무를 수행한다. 매치-메이킹 된 서비스들을 위치값을 기준으로 하여 순위를 부여하고 그 결과로서 순위가 매겨진 서비스 리스트를 반환한다. 서비스 매치-메이커는 서비스 요청에 필요한 위치값을 상황관리를 통하여 상황저장소로부터 얻는다.

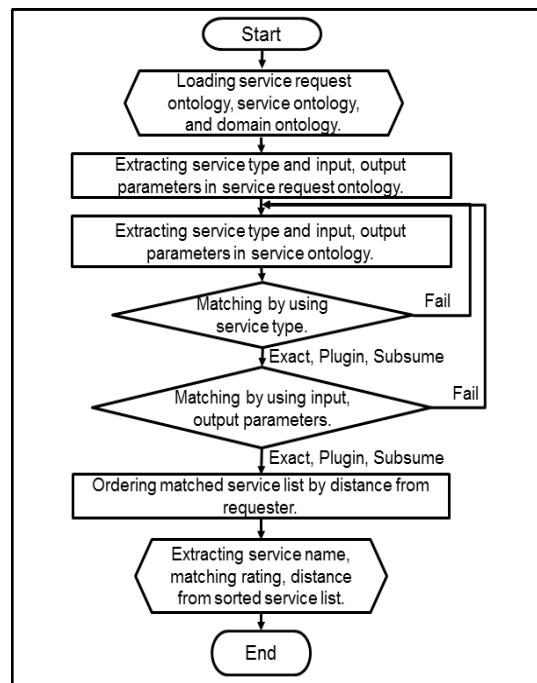


Fig. 3. Silkroad Flow Chart



## 5. 성능 평가

### 5.1 성능평가 지표

본 연구에서는 시맨틱 매치-메이킹 알고리즘을 평가하기 위하여 정성적 지표와 정량적인 지표를 개발하였다.

연구된 정성적 지표는 다음과 같다. 거짓양성과 거짓음성 (Regulated false positives and negatives), 히트 랭킹 (Ranking of hits), 연성제약조건(Soft constraint), 개념의 선호도(Preference of concepts), 사용자정의 매칭(User defined matching), 정확도 문제(Correctness issues), 계산복잡도 (Computational Complexity) 등등을 연구하였다.

본 연구에서는, 다음의 세 가지 정량적인 지표를 연구하여 사용하였다. 첫째, 정밀도 지표. 이 지표는 검색된 서비스 중에 관련된 서비스의 비율을 표시한다. 시맨틱 서비스 디스커버리 알고리즘이 정밀하지 못하면 거짓양성 값은 증가하고 정밀도 값은 감소한다. 둘째, 재현율 지표. 이 지표는 관련된 서비스 중에서 검색된 서비스의 비율을 표시한다. 시맨틱 서비스 디스커버리 알고리즘이 더 정밀해질수록, 거짓음성 값은 감소하고 재현율 값은 증가한다. 셋째, F<sub>1</sub>-측정 지표. 이 지표는 정밀도와 재현율 값의 조화 평균을 표시한다. 정밀도와 재현율은 일반적으로 상쇄효과(Trade-off) 관계를 가진다. 이 지표로서, 검색이 개선된 정도를 알 수 있다. Fig. 4는 정밀도, 재현율, 그리고, F<sub>1</sub>-측정을 구하기 위한 수식을 보여주고 있다.

$$Precision = \frac{| \{ Related Document \} \cap \{ Searched Document \} |}{| \{ Searched Document \} |}$$

$$Recall = \frac{| \{ Related Document \} \cap \{ Searched Document \} |}{| \{ Related Document \} |}$$

$$F1 Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Fig. 4. The Equation of Precision, Recall, and F1-measure

### 5.2 성능평가

본 논문에서는 정량적인 지표를 사용하여 수행한 성능평가 결과를 제시한다. 실험에 사용된 시나리오는 화재 발생 시 소방서와 관련된 서비스를 모두 찾아서 요청자에게 알려달라는 요청에 대하여 응답하는 것이다. 이 경우의 가정된 환경은 Fig. 5에 요약되어져 있다.

전체 300개의 서비스 제공자가 있다고 가정하였다. 이 중에서, 소방서 관련 서비스의 개수는 20개이다. 병원 관련 서비스 개수는 42개이다. 경찰서 관련 서비스 개수는 30개이다. 이 시나리오는 자바를 사용하여 프로그램화되었다. 이 시나리오를 우리가 개발한 실크로드 알고리즘을 사용한 시스템에 실행시키고 위에서 언급한 정량적인 지표를 사용하여 성능을 측정하였다.

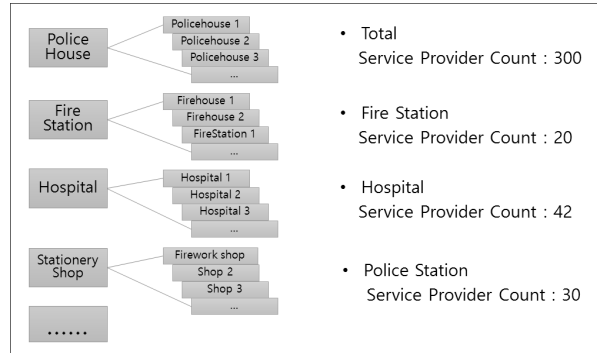


Fig. 5. Service Condition for Experiments

키워드기반 매치-메이킹 알고리즘(Keyword-based match-making algorithm)을 사용하는 시스템과 그리디 알고리즘 (Greedy algorithm)을 사용하는 시스템과 양분그래프매칭 (Bipartite graph matching) 알고리즘을 사용하는 시스템을 위와 같은 시스템 환경에서 각각 별도로 구축하였다. 그리고 위와 똑같은 시나리오를 사용하는 같은 실험을 각각의 시스템에서 반복하여 실행하였다. 이 네 종류의 실험에서 측정된 값들을 상호 비교하여 본 절에서 성능을 평가하였다. 성능평가 실험에 사용된 시스템은 Intel Core 2 Duo 6600 (2.4GHz Core 2 개로 구성됨) CPU와 4GB 메모리, 500GB 하드디스크로 구성되었다.

화재 발생 시 소방서와 관련된 서비스를 모두 찾아서 요청자에게 알려달라는 요청의 경우, 요청된 서비스의 입력 매개변수는 위치와 시간이고, 출력 매개변수는 응답행동이다. 서비스형식은 소방서로 이미 정의되어져 있다.

첫 번째로 키워드기반 알고리즘은 소방서와 글자가 일치하는 모든 서비스를 검색한다. 따라서 이 경우, 소방서로 기입된 모든 서비스가 찾아진다. 그리고, 소방서 서비스는 아니지만 키워드 등록 시 소방서라는 명칭과 연결되도록 관련 값들을 기입한 상점과 빌딩 그리고 장난감 회사의 서비스 3개가 추가로 찾아지는 원하지 않는 결과가 발생한다. 이 부분은 명백히 잘못된 선정된 결과로서, 이 알고리즘의 오류가 된다.

두 번째로 그리디 알고리즘을 사용하였을 경우, 요청과 등록된 서비스를 비교할 때에, 입출력 매개변수의 개념을 비교하기 때문에, 소방서 외에 같은 입출력 매개변수의 개념을 가지고 있는 것을 다 찾아 준다. 예를 들면, 본 실험에서는 병원의 앰블런스 서비스와 경찰차의 순찰 서비스까지도 찾아지게 되었다. 이것은 거짓양성에 해당한다. 화재사고가 발생한 위치에 근접한 곳에서 서비스를 제공한다고 등록한 것들이 있다면, 그리디 알고리즘은 등록된 모든 것들을 다 찾아주기 때문에, 소방서 서비스와 직접적인 상관은 없는데도, 이런 결과가 나온 것이다.

실크로드 알고리즘에서는, 위에서 설명한 바와 같이, 그리디 알고리즘이, 사용자가 원하는 바가 아닌 정보를 검색하여 제공하는 오류를 방지할 수 있도록 해결방법 제공한다. 실크로드는 모든 서비스를 지정할 때에 서비스형식을 추가

Table 1. The Quantitative Performance Evaluation with Requests Searching the Optimal Fire-Station

	Related Service Count	Keyword based Algorithm	Greedy Algorithm	Silkroad
Discovery Result	20	related:10 non-related:3	related:20 non-related:2	related:20 non-related:0
Ranking	-	X	X	Dongdaemoon Fire Station
Precision (%)	-	77% (10/13)	90% (20/22)	100% (20/20)
Recall (%)	-	50% (10/20)	100% (20/20)	100% (20/20)
F1-Measure	-	60.6	94.7	100

하게 함으로서, 이런 잘못된 정보들이 검색되면, 즉시 제거될 수 있게 하였다. 실크로드는, 이 방법을 사용하여, 거짓양성을 방지할 수 있게 하였다.

Table 1을 살펴보면, 실크로드가 그리디 알고리즘에 비하여 더 높은 정확도를 보이는 것을 알 수 있다. 이와 같은 이유 때문이다. Table 1은 화재발생장소 근처의 최적의 소방서를 찾는 검색요청에 대한 응답 실험을 통한 정량적 성능평가 결과를 요약한 것이다. 실크로드만 순위기능을 제공할 수 있다. 정확도에서는 실크로드가 최고의 성능을 나타냈고, 그다음으로 그리디 알고리즘이었다. 재현율에서는 그리디 알고리즘도 우수한 성능을 보였다.

Fig. 6은 정확도와 재현율의 상관관계를 실험한 결과를 보여준다. A선은 재현율이 일정한 경우에서의 정밀도를 나타내고 B선은 정밀도가 동일한 경우의 재현율 값을 나타낸다. 재현율 값이 일정한 경우에는 정밀도 값이 높을수록 좋다. 정밀도 값이 일정한 경우에는 재현율 값이 높을수록 좋

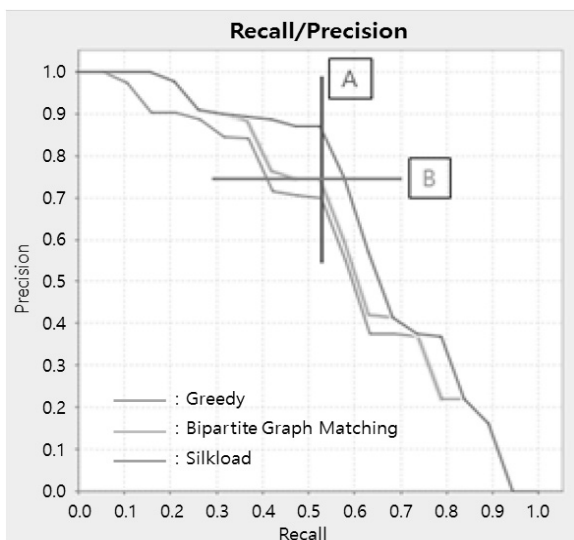


Fig. 6. The Correlation Between Precision and Recall

다. 실크로드 알고리즘이 그리디 알고리즘과 양분그래프매칭 알고리즘보다 재현율과 정밀도가 더 높게 나온다. 그 이유는 그리디 알고리즘과 양분그래프매칭 알고리즘은 다른 서비스형식으로 분류되어 있는 서비스까지 매칭 범주에 포함시키기 때문이다. 그래서, 이 성능지표에서, 성능이 그만큼 더 떨어진다. 실크로드 알고리즘은 사용자가 명시해놓은 서비스형식의 세밀한 분류를 통해 정밀도와 재현율의 성능을 향상시킬 수 있었음을 Fig. 6에서 확인할 수 있다.

Table 1과 Fig. 6의 성능평가 결과를 종합하여 보면, 매칭 등급만 가지고 가장 적합한 서비스를 찾는 데 한계를 가졌던 알고리즘들의 정확성 문제가 실크로드에서는 개선됨을 명백히 알 수 있다. 또한, 각 소방서, 병원, 경찰서를 찾을 때, 현재 사용자로부터 거리를 계산하여, 가장 근접한 순서대로 순위가 매겨져서, 추천되는 편리한 기능이 제대로 작동함을 알 수 있다.

## 6. 결 론

본 논문에서는, 본 연구진이 개발한 시스템을 적용한 스마트-시티에서, 상황에 최적인 서비스를 제공하기 위한 연구를 수행한 결과의 일부를 소개하였다. 우리가 개발한 시맨틱 서비스 디스커버리 방법론인 실크로드를 제시하고 그 작동 메커니즘을 설명하였다. 연구 개발된 시스템을 실제로 구축하여 성능평가 실험을 행하였다. 성능평가 실험에서는 기존의 존재하는 우수한 방법론들을 세 가지 선정하고, 이들을 모두 실험시스템에서 실측 성능평가를 하였다. 이들의 성능과 실크로드의 성능을 비교 분석한 결과를 본 논문에서 제시하였다. 성능평가를 위하여 우리의 성능평가 실험에 가장 유용하게 쓰일 수 있는 성능평가 지표를 연구 개발하였다. 정성적인 성능평가 지표와 정량적인 성능평가 지표를 개발하거나 선정하였는데, 본 실험에서는 정밀도와 재현율을 포함하는 정량적인 성능평가 지표 위주로 성능평가실험을 진행하였다.

시맨틱 매치-메이킹은 서비스 요청자와 제공자 사이에서, 가장 적절한 서비스를 매칭하는 역할을 한다. 본 논문에서 제안한 실크로드를 기반으로 하는 시맨틱 매치-메이킹 방법론은 기존 시맨틱 매치-메이킹이 가진 한계점들을 개선한 것이다. 사용자의 요청을 의미론적으로 분석할 수 있도록 상황 모델링과 추론을 통해 상황을 정확히 파악하여 이를 이용함으로써, 기존의 구문 기반의 매치-메이킹 보다 나은 시스템을 개발할 수 있었다.

본 연구의 결과는 스마트-시티에서 최적의 서비스를 찾아서 연결해주는 작업이 요구될 때, 시스템이 자동적이고 지능적으로 최적의 응답을 해 주는 일을 필요로 하는 경우에 매우 유용하게 이용될 수 있다. 아직까지, 본 연구를 제외하고 스마트-시티 분야에서 우리와 유사한 연구를 발견할 수 없었다. 본 연구는 앞으로 많은 점에서 발전을 필요로 한다. 따라서, 향후 추가 연구를 지속할 것이다.

References

[1] M. Weiser, "The Computer for the Twenty-First Century," *Scientific American*, Vol.265, No.3, pp.94-101, 1991.

[2] H. S. Jung, C. S. Jeong, Y. W. Lee, and P. D. Hong, "An Intelligent Ubiquitous Middleware for U-City: SmartUM," *Journal of Information Science and Engineering*, Vol.25, No.2, pp.375-388, 2009.

[3] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, "Semantic matching of web service capabilities," *Lecture Note in Computer Science (LNCS)*, Vol.2342, pp.333-347, 2002.

[4] U. Bellur and R. Kulkarni, "Improved Matchmaking Algorithm for Semantic Web Services based on Bipartite Graph Matching," *Proc. of IEEE International Conference on Web Services (ICWS)*, pp.86-93, 2007.

[5] N. Ryan, J. Pascoe, and D. Morse, "Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant," *Computer Applications in Archaeology, Proc. of the 25th Anniversary Conference*, pp.269-274, 1997.

[6] M. Beck and B. Freitag, "Semantic matchmaking using Ranked instance retrieval," *Proc. of the 1st International Workshop on Semantic Matchmaking and Resource Retrieval, Co-located with VLDB (SMR'06)*, pp.1-8, 2006.

[7] S. Dasgupta, A. Aroor, F. Shen, and Y. Lee, "SMARTSPACE: Multiagent Based Distributed Platform for Semantic Service Discovery," *IEEE Transactions on Systems*, Vol.44, Issue 7, pp.805-821, 2013.

[8] K. Qian, X. Ma, X. Dai, and F. Fang, "Knowledge-enabled decision making for robotic system utilizing ambient service components," *Journal of Ambient Intelligence and Smart Environments*, Vol.6, Issue 1, pp.5-19, 2014.

[9] S. A. Khanam and H. Y. Youn, "A Web Service Discovery Scheme Based on Structural and Semantic Similarity," *Journal of Information Science & Engineering*, Vol.32, Issue 1, pp.153-176, 2016.

[10] Location Awareness in HP's CoolTown [Internet], <http://www.w3.org/Mobile/posdep/HPw3cwapref.html>.

[11] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, "Easyliving: Technologies for intelligent environments," *Lecture Note in Computer Science (LNCS)*, Vol.1927, pp. 12-29, 2002.

[12] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, "Gaia: A middleware infrastructure to enable active spaces," *IEEE Pervasive Computing*, Vol.1, No.4, pp.74-83, 2002.

[13] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz "Pellet: A practical OWL-DL reasoner," *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 5, Issue 2, pp.51-53, 2007.

[14] A. Ranganathan and R. Campbell, "An infrastructure for Context-awareness based on First order logic," *Personal and Ubiquitous Computing*, Vol.7, No.6, pp.353-364, 2003.

[15] WDSL2OWLS [Internet], <http://www.mindswap.org/2004/owl-s/api/doc/javadoc/examples/WSDL2OWLS.html>.

[16] Jena API framework [Internet], <http://jena.sourceforge.net/inference/index.html>.

[17] A. R. El-Sayed and J. Blackm, "Semantic-based Context-aware service discovery in pervasive-computing environments," Master Thesis, Waterloo, Ontario, Canada, 2006.

윤 창 호



e-mail : touch011@uos.ac.kr

2008년 서울시립대학교  
전자전기컴퓨터공학부(공학사)

2010년 서울시립대학교  
전자전기컴퓨터공학과(공학석사)

2010년~현 재 서울시립대학교  
전자전기컴퓨터공학과 박사과정

관심분야 : 클라우드 컴퓨팅, 스마트시티, 시맨틱웹, ICT융합시스템, IoT, 인터넷

박 종 원



e-mail : comics77@uos.ac.kr

2009년 서울시립대학교  
전자전기컴퓨터공학부(공학사)

2011년 서울시립대학교  
전자전기컴퓨터공학과(공학석사)

2011년~현 재 서울시립대학교  
전자전기컴퓨터공학과 박사과정

관심분야 : 인터넷, 클라우드 컴퓨팅, 시스템 소프트웨어, 스마트시티, ICT융합시스템, IoT

정 해 선



e-mail : banyasun@uos.ac.kr

1992년 명지대학교 전자계산학과(공학사)

2001년 고려대학교 전자컴퓨터공학과  
(공학석사)

2011년 고려대학교 전자컴퓨터공학과  
(공학박사)

2015년~현 재 서울시립대학교 전자전기컴퓨터공학과 연구교수  
관심분야 : 인터넷, 클라우드 컴퓨팅, 시스템 소프트웨어, 스마트시티, ICT융합시스템, IoT



## 이 용 우

e-mail : ywlee@uos.ac.kr

1981년 서울대학교 전기&컴퓨터(공학사)

1990년 영국 Univ of Edinburg(공학석사)

1997년 영국 Univ of Edinburg(이학박사)

1999년~현재 서울시립대학교

전자전기컴퓨터공학과 교수

관심분야: 운영체제, 클라우드 컴퓨팅, 시스템 소프트웨어,  
스마트시티, ICT융합시스템, IoT, 스마트시스템,  
초고속컴퓨터통신