

유비쿼터스 컴퓨팅 환경에서의 온톨로지를 이용한 동적인 서비스 구성 Dynamic Service Composition using Ontology for Ubiquitous Computing Environments

김지홍*, 하병현*, 김철영**, 이우기***, 강석호*

*서울대학교 산업공학과

**사이버메드 대표이사

***성결대학교 컴퓨터 공학부

Abstract

가까운 미래에는 보이지 않는 수많은 장치들과 소프트웨어들이 서로 연결되어 각각의 사용자들에게 편리한 서비스를 제공할 것이다. 유비쿼터스 컴퓨팅 환경의 목표는 상이하며 유동적인 사용자들의 요구를 달성하기 위해 많은 서비스들 중 적절한 것들을 선택하고, 그것을 알맞게 구성하여 제공하는 것이다. 동적 서비스 구성을 위하여 본 연구에서는 다음을 제시하고 있다. 첫째, 서비스 구성에 필요한 요소를 파악한 후, 각 요소의 역할, 관계, 계층구조 그리고 속성을 표현하는 온톨로지를 기술한다. 둘째, 서비스 구성을 위한 필요한 규칙들을 기술한다. 셋째, 위의 두 가지를 이용하여 사용자의 목적에 따르는 서비스를 구성하기 위한 체계적인 알고리즘을 제시한다. 마지막으로, 본 연구의 결과를 예제 시나리오에 적용하여 동적 서비스 구성 가능성을 제시하였다.

1. 서론

현재 IPv6와 RFID와 같은 새로운 IP 주소체계와 칩 기술, 수많은 네트워크 기술, 광대역 무선 인터넷 기술, 센서 기술 등이 빠르게 발전하고 있다. 따라서 “어떤 사람이라도, 언제 어디에서, 각종 단말기와 장치들을 통해서 광대역 네트워크에 접속하여 서비스를 받을 수 있는 공간 또는 환경”인 유비쿼터스 컴퓨팅[1]의 시대가 그 어느 때보다 가깝게 다가오고 있다.

이러한 유비쿼터스 컴퓨팅의 특징은 다음과 같다. 첫째 수많은 컴퓨터와 내장되어 있는 장치들이 네트워크에 접속되어 있고, 둘째로 사용자는 컴퓨터를 사용한다는 사실조차 인지하지 못하며, 셋째로 사용자의 요구나 상황에 따라 제공되는 서비스가 변한다는 것이다. [1] 우리가 본 연구에서 집중해서 살펴볼 특징은 세 번째의 것이다. 유비쿼터스 컴퓨팅 환경에서는 사용자를 위한 서비스들뿐만 아니라 사용자의 요구 또한 다양하고 폭넓은 형태를 취하게 된다. [2] 따라서 사용자의 요구가 발생했을 때, 하나의 단위 서비스로써 그 목적을 달성하는 것은 쉽지 않을 뿐만 아니라, 각각의 사용자 요구에 맞는 수많은 서비스를 모두 정의하여 사용하기도 역시 불가능하다. [3]

예를 들어 A라는 병에 걸린 사람의 상태를 체크

하여 그 사람이 물을 먹어야 한다는 목적이 도출되었을 때, 그것을 달성하기 위해서는 물을 제공할 수 있는 서비스를 탐색해야 하고, 사람의 위치를 파악하여 가까운 곳에 위치한 장치에서 물을 먹으라는 메시지를 전송해야 할 것이며, 사람이 이동했을 때 물을 제공해야 할 것이다.

위의 경우에서 미루어 볼 수 있듯이 일반적으로 하나의 목적을 달성하기 위해서는 여러 개의 각기 다른 단위 서비스들의 조합이 필요하다. 따라서 특정 목적을 해결하기 위해서는 여러 개의 단위 서비스를 통합, 구성하여 사용자에게 제공해야 한다. 그러나 모든 경우에 해당하는 서비스 구성을 미리 정의해서 사용하는 것은 불가능하다. 따라서 동적인 목표를 만족하기 위해 자동적인 단위 서비스의 통합과 구성이 필요하다.

이러한 자동적인 서비스 구성을 위해서는 사용자와 서비스들의 관계, 사용자의 요구와 목적의 관계, 각각의 서비스들이 할 수 있는 일, 그리고 서비스들의 상호작용 방식을 알아야 한다. 그리고 이들은 명료하게 정의된 형식과 의미를 제공하는 온톨로지 기술되어야 한다. 따라서, 본 연구에서는 서비스 구성에 필요한 단위 서비스들과 사용자들의 온톨로지를 표준에 맞추어 기술하고 필요한 지식들을 규칙으로써 정의하였다. 그리고 이렇게 정의된 온톨로지와 규칙들을 이용하여 사용자의 요구에 맞는 서비스 구성을 추론하는 알고리즘을 제시하였다. 또한 예제 시나리오를 통해 제시한 알고리즘의 가능성을 검증하였다.

본 논문은 다음과 같이 구성된다. 2장에서는 유비쿼터스 컴퓨팅과 온톨로지, 웹서비스 구성 그리고 유비쿼터스 환경의 서비스 구성에 대한 연구 현황을 알아보고 3장에서는 동적인 서비스 구성에 필요한 구성요소들을 정의, 기술하고 서비스 구성을 위한 알고리즘을 제시하였다. 4장에서는 앞서 언급한 시나리오에 3장에서 제시한 알고리즘을 적용하여 동적 서비스를 구성하는 예제를 제시하였다. 마지막으로 5장에서는 유비쿼터스 환경에서의 서비스 구성의 결과를 정리하고 결론과 추후과제를 제시한다.

2. 관련연구

2.1 유비쿼터스 컴퓨팅

유비쿼터스 컴퓨팅은 마크 와이저에 의해 처음 소개되었다. [1] 그는 "가장 심오한 기술은 사라지는 것"이라고 밝히고 있으며, 이것을 현재 우리가 사용하고 있는 문자에 비교하여 설명하고 있다. 그가 명확하게 문장으로 언급하지는 않았지만, "컴퓨터가 보이지 않게 내장되어 네트워크로 연결되어 있고 언제 어디서나 접속이 가능하여 서비스를 받을 수 있는 환경"이라고 정의 될 수 있는 유비쿼터스 컴퓨팅은 현재 매우 빠른 속도로 발전하여 IT산업의 키워드가 되었으며, 세계적인 연구가 진행되어지고 있다.

2.2 온톨로지 언어

온톨로지는 사용자와 서비스를 구체적으로 기술하고 그들의 관계를 정의할 수 있게 한다. 따라서 온톨로지는 장치들이 이해할 수 있는 의미를 제공하여 서비스를 구성하는데 핵심이 될 것이다. 온톨로지 표준으로는 W3C RDF Core Working Group에서 개발한 RDF [4]와 여기서 확장된 DAML+OIL [5], W3C Web Ontology Language Working Group에서 개발한 OWL [6]이 있으며, 또한 웹 서비스를 표현하기 위한 온톨로지 언어인 DAML-S [7]가 있다.

2.3 웹 서비스 구성

현재 웹 서비스 구성하는 방법은 크게 두 가지의 방식이 존재한다. 하나는 SOAP, WSDL 그리고 UDDI 등의 웹 서비스 표준을 이용하여 문법적인 구성에만 집중하는 것이고, 다른 한 가지는 시멘틱 웹 [8] 기술을 이용하여 의미론적인 접근을 통해 서로의 상호운영성을 달성하는 것이다.[3] 본 연구에서는 후자의 기법을 빌어 사용하고 있다. SWORD는 웹 서비스를 입력과 출력으로 모델링하고, 하나의 서비스의 출력을 다른 서비스의 입력으로 연결하는 방식을 통해 서비스 구성을 제시하는 프레임워크이다. [9] 또한 기존의 UDDI와 DAML-S를 이용하여 서비스를 정의하고 DAML-S Virtual Machine을 통하여 그들을 구성하는 방법이 있다. [3]

2.4 유비쿼터스 컴퓨팅 환경의 서비스 구성

현재 유비쿼터스 컴퓨팅 환경의 서비스 구성에 관한 연구 역시 활발하게 진행되고 있다. Takemoto et al.은 서비스를 SE (Service Element)와 ST (Service Template) 두 가지로 나눈 후, USON (Ubiquitous Service Oriented Network)의 구조를 바탕으로 적절한 ST를 검색하고, ST를 완성하는 SE를 탐색하여 서비스를 구성한다. [2] 그리고 IFNS (Interface-based Naming System)를 통해 서비스를 구성하는 연구 [10]와 컨텍스트 종류에 따라 서비스를 선택하는 연구 [11] 등이 있다.

3. 동적인 서비스 구성

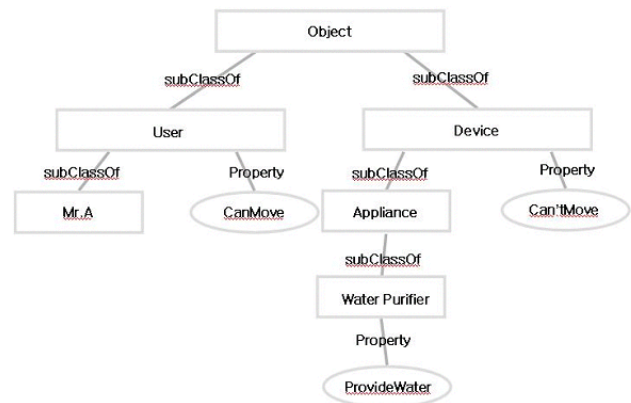
목적에 맞는 서비스를 검색하고 동적으로 구성하기 위해서는 두 가지 조건이 충족되어야 한다. 첫째, 유비쿼터스 컴퓨팅 환경을 이루고 있는 각각의 장치와 사용자들의 정보가 자세히 기술되어야 한다. 둘째, 기술된 정보들을 바탕으로 모든 객체가 서로의 역할과 관계를 정확히 파악하고 전달할 수 있어야 한다. 이 조건들을 충족시키기 위해 본 연구에서는 온톨로지를 도입하여 객체를 기술하였으

며, 또한 각 객체들의 자율적인 활동에 필요한 규칙을 기술하였다.

본 장에서는 유비쿼터스 컴퓨팅 환경에서 사용될 객체와 서비스를 온톨로지로 정의/기술하고, 추론을 위한 지식들을 규칙으로 기술함으로써 동적 서비스 구성의 방법론을 제시한다. 이어지는 3.1.절, 3.2.절, 3.3.절에서는 동적 서비스 구성의 필수 요소인 사용자, 장치, 서비스, 규칙을 정의하고, 3.4.절과 3.5.절에서는 동적 서비스 구성을 위한 시스템 구조와 구성 알고리즘을 제시한다.

3.1 객체

유비쿼터스 컴퓨팅 환경에서의 객체는 크게 사용자와 장치, 소프트웨어로 나누어질 수 있으며, 각 객체들에 대한 역할, 관계, 계층 구조, 성질, 속성 등의 정보가 온톨로지로 기술되어야 한다. [그림 1]은 서론에서 제시된 예제 시나리오에서 사용되는 온톨로지의 일부를 나타낸다.



[그림 1] 예제 시나리오를 위한 객체 온톨로지

3.2 서비스

유비쿼터스 컴퓨팅 환경에서 단위 서비스의 정의는 장치, 소프트웨어, 네트워크의 기능과 이들 사이의 정보 흐름을 포함해야 한다. [2]

이 조건을 만족하기 위해, 본 연구에서는 웹 서비스 표현 온톨로지인 DAML-S [7]의 Service Profile을 사용해 단위 서비스를 기술한다.

DAML-S의 Service Profile은 Input, Output, Precondition, Effect로 구성된다. [표 1]은 예제 시나리오에서 사용되는 서비스들의 일부를 나타낸다.

[표 1] 예제 시나리오를 위한 서비스 프로파일

Service	Water Provide Service	Alert Service
Precondition	Water(x)^Device(y)^ Exist(x, y), User(x)^Location(x,y) ^Device(z)^Location(z,k) ^Same(y,k)	N/A
Input	User(x)^Amount(y)	User(x)^Message(y)
Output	Amount(x)^Water(y,x)	User(x)^Message(y) ^Alert(x,y)
Effect	User(x) ^ Water(y) ^Provide(x,y)	User(x)^Action(y)^ Recognize(x, y)

3.3 규칙

동적 서비스 구성을 위해서는 객체와 서비스의

정의와 더불어 이들 간의 원칙과 서비스들의 추론 과정을 위한 지식들이 기술되어야 한다. 본 연구에서는 이러한 원칙과 지식들을 규칙으로써 표현하여 사용하였다.

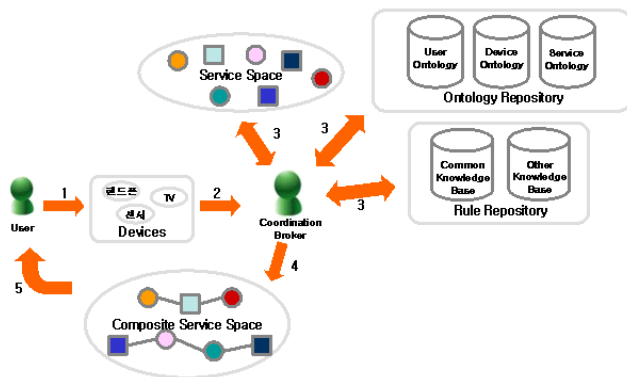
다음은 예제 시나리오에서 사용될 규칙 중 일부를 표현한 것이다.

- 사용자의 상태를 변화시키기 위해서는 변화될 상태를 사람에게 알려야 한다.
 $User(x) \wedge CurrentState(y) \wedge FutureState(z) \wedge ChangeState(y,z) \Rightarrow User(x) \wedge Alert(x,z)$

3.4 Coordination Agent

Coordination Agent는 사용자의 목적을 입력받아, 온톨로지 저장소와 규칙 저장소로부터 객체 정보와 규칙을 검색하여, 이를 이용해 목적에 적합한 서비스를 동적으로 구성한다.

[그림 2]는 온톨로지 저장소, 규칙 저장소, 사용자, Coordination Agent 등으로 구성된 동적 서비스 구성을 위한 시스템 구조를 나타낸다.



[그림 2] 동적 서비스 구성 시스템 구조

3.5 서비스 구성 구조와 알고리즘

Coordination Agent에 의해 실행되는 서비스 구성 알고리즘은 다음과 같다. 이 알고리즘은 이전 절에서 설명한 핵심 구성요소들 외에, Discovery Agent와 같은 부가적 요소들도 사용하고 있다.

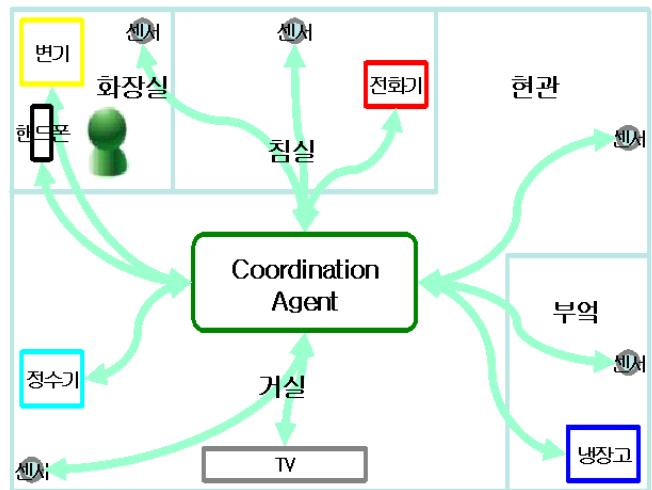
```

WHILE(TRUE)
  New_Goal = first(Goal_List)
  IF New_Goal is NOT NULL THEN
    Rules_By_Goal[] = Rule_Repository.getRules(New_Goal)
    Unsatisfied_Rules_By_Goal[] = compareToStatus(Rules_By_Goal[])
    IF Unsatisfied_Rules_By_Goal[] contains at least 1 element THEN
      Sub_Goals[] = Unsatisfied_Rules_By_Goal[]
      insertIntoFirst(Goal_List, Sub_Goals[])
    ELSE
      DeviceServiceMap[] =
        Discovery_Agent.search(Ontology_Repository, New_Goal)
      Rules_By_Device[] = Rule_Repository.getRules(DeviceServiceMap[])
      Unsatisfied_Rules_By_Device[] =
        compareToStatus(Rules_By_Device[])
      IF Unsatisfied_Rules_By_Device[] contains at least 1 element
        THEN
          Sub_Goals[] = Unsatisfied_Rules_By_Device[]
          insertIntoFirst(Goal_List, Sub_Goals[])
        ELSE
          IF DeviceServiceMap[] contains more than 1 element THEN
            New_Service = selectServiceCost(DeviceServiceMap[])
          END IF
          Rules_By_Precondition[] =
            Ontology_Repository.getPrecondition(New_Service)
          Unsatisfied_Rules_By_Precondition[] =
            compareToStatus(Rules_By_Precondition[])
          IF Unsatisfied_Rules_By_Precondition[] has at least 1 element
  
```

```

THEN
  Sub_Goals[] = Unsatisfied_Rules_By_Precondition[]
  insertIntoFirst(Goal_List, Sub_Goals[])
ELSE
  call Device.executeService(New_Service)
  deleteFirst(Goal_List)
END IF
END IF
END IF
END IF
WEND
  
```

4. 서비스 구성 예제



[그림 3] 예제 시나리오를 위한 환경 구성도

본 장에서는 서론에 언급한 예제 시나리오를 상세히 설명하고, 시나리오에서 필요한 사용자와 장치 온톨로지, 서비스 온톨로지 그리고 규칙들을 열거한 후, 3.5절에서 제시한 알고리즘을 적용한다.

예제 시나리오는 다음과 같다. 사용자 A씨는 당뇨병에 걸린 상태이다. 따라서 A씨는 당뇨병 치료를 위해서 헬스 케어 센터에 등록을 하였고, 헬스 케어 서비스를 받는 것에 동의하였다. A씨의 집은 유비쿼터스 컴퓨팅 환경을 위해 여러 가지의 장치들과 소프트웨어들을 같이 갖추고 있다. [그림 3]

서비스 등록 후 A씨는 평소와 같은 일과를 보내다 화장실에 들러 용변을 보게 되었다. 그 도중 화장실의 변기는 A씨의 상태를 Coordination Agent를 통해 헬스 케어 센터에 보내게 되고, 헬스 케어 센터는 추론을 통해 현재 A씨가 수분 부족 현상으로 물을 섭취해야 한다는 것을 Coordination Agent에게 전달하였다. Coordination Agent는 A씨에게 가까운 정수기로 가서 물을 먹을 것을 핸드폰을 통해 알리고 정수기에 물을 공급할 것을 명령한다. A씨는 정수기로 다가가 물을 마신다.

본 시나리오에서의 동적 서비스 구성을 위해 다음과 같은 사실들을 가정하였다. 첫째, [그림 3]에서 표현된 사용자와 여러 가지 장치들의 온톨로지가 정의되어 있다. 둘째, 목적을 이끌어내기 위한 규칙들이 정의되어 있다. 그 규칙들은 다음과 같다.

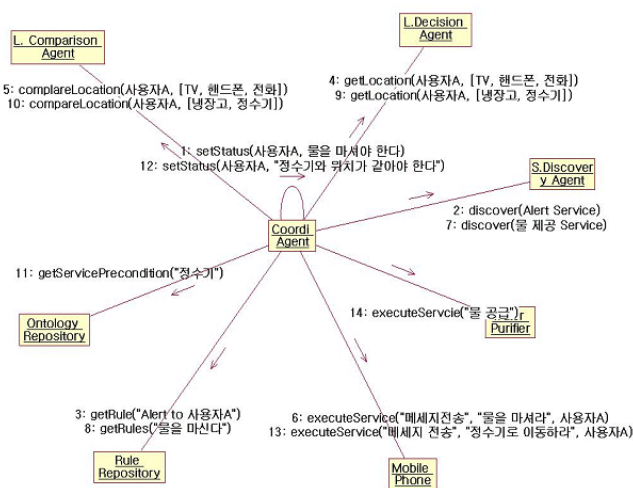
- 규칙 1 : 사람의 상태를 변화시키기 위해서는 변화될 상태를 사람에게 알린다.
 $User(x) \wedge CurrentState(y) \wedge FutureState(z) \wedge ChangeState(y,z) \Rightarrow User(x) \wedge Alert(x,z)$
 - 규칙 2 : 같은 서비스를 제공하는 장치가 여러 개일 때는 사용자에게 가장 가까운 거리에 있

는 것을 이용한다.

$$\text{Service}(x) \wedge \text{Device}(y,x) \wedge \text{Device}(z,x) \wedge \text{User}(k) \Rightarrow \text{User}(k) \wedge \text{UseDevice}(k,y) \wedge \text{Close}(k,y)$$

셋째, 추론에 필요한 단어들의 의미가 온톨로지와 규칙으로 정의되어 있어야 한다. 마지막으로 목적 달성을 위한 다음과 같은 서비스들이 정의되어 있어야 한다. 서비스를 가진 장치나 소프트웨어를 탐색할 수 있는 Discovery Service, 장치들과 사용자의 위치를 파악 하는 Location Decision Service, 장치와 사용자간의 거리를 판별하는 Location Comparison Service, 사용자에게 메시지를 전달하는 Alert Service, 그리고 사용자에게 물을 제공하는 Water Provide Service 등이 이에 해당한다.

[그림 4]는 3.5절에서 제시된 알고리즘을 위의 구성 요소들에 적용하여 동적으로 서비스가 구성되는 과정을 나타낸 Collaboration Diagram이다.



[그림 4] 서비스 구성 Collaboration Diagram

5. 결론 및 추후 과제

본 연구에서는 유비쿼터스 컴퓨팅 환경에서 사용자의 목적을 달성하기 위해서 동적 서비스 구성의 필요함을 설명하고, 이 동적 서비스 구성의 필요요소들을 정의, 구분한 후 서비스 구성 알고리즘을 제시함으로써 동적 서비스 구성의 체계적 방법론을 제시하였다. 덧붙여 이 알고리즘을 예제 시나리오에 적용하여 동적 서비스 구성의 가능성을 보였다.

추후 연구 과제는 다음과 같다. 첫째, 동적 서비스 구성을 위한 목적의 생성은 고려하지 않았으므로, 사용자 행동으로부터 목적을 생성하는 방법론에 대한 연구가 필요하다. 둘째, 요구된 목적에 대한 정보가 불충분한 경우 목적 달성을 위해 목적을 재구성하는 방법론에 대한 연구도 역시 필요하다.

6. Acknowledgement

본 연구는 과학기술부의 21C프론티어사업 [04A2-B1-31]의 지원으로 수행되었음.

7. 참고 문헌

[1] Mark Weiser, "The Computer for 21st Century", Scientific American, vol. 265, no 3, pp. 94-104, 1991.
 [2] Takemoto, M., Oh-ishi, T., Iwata, T.,

Yamato, Y., Tanaka, Y., Shinno, K., Tokumoto, S., and Shimamoto, N., "A Service-Composition and Service-Emergence Framework for Ubiquitous-Computing Environments", In Proceedings of the 2004 International Symposium on Applications and the Internet Workshops, pp. 287-292, 2004.

[3] K. Sycara, M. Paolucci, A. Ankolekar, and N. Srinivasan, "Automated discovery, interaction and composition of Semantic Web services", Web Semantics: Science, Services and Agents on the World Wide Web, vol. 1, no. 1, pp. 27-46, 2003.
 [4] F. Manola, and E. Miller, "RDF Primer", W3C Recommendation, 2004.
 [5] D. Connolly, F. van Harmelen, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein, "DAML+OIL Reference Description", W3C Note, 2001.
 [6] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein, "OWL Web Ontology Language Overview", W3C Recommendation, 2004.
 [7] A. Ankolekar, M. Burstein, J. R. Hobbs, O. Lassila, D. L. Martin, S. A. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, and H. Zeng, "DAML-S: Semantic Markup for Web Service, Proceeding of the International Semantic Web Workshop, 2001.
 [8] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web", Scientific American, vol. 284, no. 5, pp. 34-43, 2001.
 [9] S. R. Ponnekanti and A. Fox "SWORD : A Developer Toolkit for Web Service Composition", The Eleventh International World Wide Web Conference, 2002.
 [10] Minami, M. Morikawa, H. and Aoyama, T., "The Design of Naming-based Service Composition System for Ubiquitous computing Applications", In Proceedings of the 2004 International Symposium on Applications and the Internet Workshops, pp. 278-286, 2004.
 [11] Kawamichi, H. Sameshima, S. Kato, H and Kawano, K., "A Service Selection Method on context Types for a Ubiquitous Service System in a Public Space", In Proceedings of the 2004 International Symposium on Applications and the Internet Workshops, pp. 293-299, 2004.