
EUCAS : 동적인 상황인식서비스 정의를 위한 사용자인터페이스 개발

EUCAS : Development of the User Interface for Dynamic Context-aware Service Definition

강기봉, Kibong Kang*, 박정규, Jeongkyu Park**, 이궁해, Keung Hae Lee***

요약 정보통신기술의 발달로 우리의 생활에는 많은 서비스들이 등장하였다. 이런 서비스들은 우리의 삶을 편리하고 안전하게 하지만 서비스의 증가와 비례하여 우리가 이를 인식하고 활용하는데 필요한 관심과 노력도 증가하게 되었다. 상황인식서비스는 이런 문제를 해결하기 위해 사용자 주변 상황을 스스로 인식하고 적시적소에 자동으로 제공되는 서비스이다. 기존 상황인식서비스에 관한 연구들을 살펴보면 상황인식서비스는 주로 개발자에 의해 정의된다. 개발자에 의한 상황인식서비스 정의는 사용자 주변 환경이나 개인의 생활패턴을 반영할 수 없다는 중요한 문제를 가진다. 본 논문에서는 서비스의 필요성을 느끼는 사용자 스스로가 상황인식서비스를 직접 정의할 수 있게 하는 사용자인터페이스인 EUCAS(by End-User, Context-Aware Service development)를 소개한다. EUCAS는 사용자가 상황인식서비스를 직접 정의할 수 있게 하기 위해 주변 상황정보와 실행 가능한 서비스들을 객체의 형태로 표현하고 이를 쉽게 조작할 수 있는 사용자인터페이스를 제공한다. EUCAS는 사용자가 자신이 필요한 상황인식서비스를 직접 정의하고 관리할 수 있게 함으로써 개인화된 상황인식서비스 환경구축을 가능하게 한다.

↓

Abstract

According to the development of information technology, there are many services in our life. These services make our life safe and convenient. However, the increment of the services also causes the increment of human concern and effort to control these services. The context-aware service is the service that provided their functionality at the right time and to the right place by analysis user's current situation. The most previous studies about context-aware service regard that context-aware services are defined by the developer who has expertise in information technology. The definition of the context-aware services by the developer makes difficult to reflect user's personal preferences and life pattern to the services. In this paper, we propose an user interface EUCAS(by End-User, Context-Aware Service development) that make the user can define and manage their own context-aware service according to their preferences. We expect EUCAS can be effective user interface technology for providing personalized context-aware service.

핵심어: Context-Awareness, Context-Aware Service, End-User Development, 상황인식, 상황인식서비스, 최종사용자 개발

*주저자 : 한국항공대학교 컴퓨터공학과 e-mail: kbkang@kau.ac.kr

**공동저자 : 한국항공대학교 컴퓨터공학과 e-mail: fcopark@kau.ac.kr

***교신저자 : 한국항공대학교 컴퓨터공학과 교수 e-mail: khlee@kau.ac.kr

1. 서론

정보통신기술의 발달로 우리의 생활에는 많은 서비스들이 등장하였다. 이런 서비스들은 우리의 삶을 편리하고 안전하게 한다. 그러나 서비스의 증가는 이를 인식하고 활용하는 우리의 관심과 노력의 증가도 초래하였다. 상황인식서비스는 서비스의 인식과 활용에 필요한 우리의 관심과 노력을 줄여 주는 서비스제공방법이다. 기존 상황인식서비스와 관련된 연구들에서 상황인식서비스는 개발자에 의해 정의되는 것으로 간주하고 있다[1-5]. 개발자에 의한 상황인식서비스 정의는 사용자의 생활환경과 개개인의 각기 다른 생활방식에 대한 정보 부족으로 인해 개인화된 상황인식서비스를 정의할 수 없다는 한계를 가진다[6,7]. 이에 반해 사용자가 상황인식서비스를 직접 정의할 수 있도록 하면 개인화된 상황인식서비스 정의가 가능해져 상황인식서비스의 유용성은 더욱 커지게 될 것이다[10-14].

상황인식서비스의 개발은 OWL-S[8]이나 ECA-DL[9]과 같은 서비스 명세 언어를 통해 이루어진다. 이러한 언어들은 개발자를 위해 만들어진 것으로 비전문개발자인 사용자가 이용하기 위해서는 올바른 코드를 입력할 수 있어야 하고 적합한 연산을 적용 시킬 수 있어야 한다. 그리고 작성된 코드에서 발생할 수 있는 오류를 발견해 방지할 수 있어야 한다[15]. 이는 비전문개발자인 사용자가 만족하기 어려운 요구이다. 따라서 사용자가 상황인식서비스를 직접 개발하기보다는 사용이 쉬운 인터페이스를 통해 정의할 수 있는 방법이 필요하다.

본 논문에서 제안하는 EUCAS는 상황인식서비스의 필요성을 인지하는 주체와 정의하는 주체를 통일한 사용자 인터페이스다. 정보통신과 관련된 전문지식이 없는 사용자도 EUCAS를 이용하면, 주변 상태정보를 파악하여 원하는 상황과 이에 따른 서비스의 실행을 정의할 수 있다.

2. 관련 연구

사용자가 상황인식서비스를 정의할 수 있게 하는 대표적인 연구들은 *Context-Studio*[10], *a CAPpella*[11], *Jigsaw Editor*[12]가 있다. *Context-Studio*는 모바일 장치를 대상으로 하는 기술로 특정 키나 키의 입력순서, 모바일 장치의 상태 등에 따라 원하는 모바일 장치의 기능을 사용자가 동적으로 할당할 수 있게 하는 기술이다. 예를 들어 휴대폰이 대기 모드이고 위치가 회의실인 경우 에티켓 모드로 전환하도록 서비스를 구성할 수 있다. 위치와 같은 주변 상황 정보는 모바일 장치가 센서를 부착하고 있는 것으로 가정하고 파악할 수 있도록 하고 있다. *Context-Studio*는 단일 장치(모바일 장치)에서 동작하는 상황인식서비스로 장치 간 협력을 정의할 수 없다는 한계를 가진다.

*a CAPpella*는 시연을 통하여 상황인식서비스를 정의할 수 있도록 하는 기술이다. 사용자가 시간 범위를 지정하면, 그 시간 동안 사용자의 행동 및 주변 장치의 상태와 센서들의 데이터를 기록하고 분석하여 상황인식서비스로 생성하여 준다. 예를 들어 사용자가 더울 때 거실 창문을 여는 서비스를 생성하고 싶다면 사용자는 *a CAPpella*에게 시연의 시작을 알리고 더위를 느낄 때 거실 창문을 열고 시연이 끝났음을 *a CAPpella*에게 알린다. *a CAPpella*는 기록된 데이터와 사용자의 행동을 분석하여 사용자가 행동을 취했던 상황이 되면 창문을 여는 행동을 하는 서비스를 생성한다. 사용자는 주변 상황과 자신이 어떠한 행동을 하는지 인지할 필요 없이 서비스를 생성하고 이용할 수 있다. 기록된 데이터에서 서비스 정의에 필요한 정보들을 추출하기 때문에 명확한 상황 정의를 위해서 여러 번의 시연을 거쳐 데이터를 누적해야 한다. *a CAPpella*는 사용자가 직접 서비스 정의에 참여할 수 있는 방법은 제공하지 않아 주변 환경이나 조건이 변화하면 시연을 반복해 줘야 하는 한계를 지닌다.

*Jigsaw Editor*는 조각 퍼즐을 조립하는 것처럼 상황인식 서비스를 구성할 수 있도록 하는 기술이다. 예를 들어 *Jigsaw Editor*상에 사람이 집안에 들어오는 그림의 조각과 거실등을 켜는 조각이 있다고 하자. 사용자는 이 두 조각을 순서대로 배치함으로써 사람이 집안에 들어왔을 때 불을 켜는 상황인식서비스를 생성할 수 있는 것이다. *Jigsaw Editor*의 조각은 정적인 상황과 행동으로 구성되며 사용자의 요구에 맞는 조각이 있다면 간단한 조합만으로 명확한 서비스를 생성할 수 있다. 정적인 조각의 구성은 다양한 상황인식서비스 구성을 위해 다양한 조각들이 추가되어야 한다. 조각들은 적절히 개발되어 제공받는 것으로 보고 있어 명확한 제공 기준이 없다는 한계를 지닌다.

상황인식서비스 정의에는 사용자의 주변 환경을 구성하는 다양한 장치와 센서들을 이용하여 상황을 표현할 수 있어야 하며 사용자가 직간접적으로 참여할 수 있는 방법이 제공되어야 한다. 사용자가 상황인식서비스를 정의하기 위해서는 이해와 사용이 쉬운 인터페이스로 상황인식서비스를 명세할 수 있어야 한다. 따라서 본 논문에서 제안하는 EUCAS는 사용자에게 의한 상황인식서비스 정의를 위해 *Jigsaw Editor*의 퍼즐조합 방식의 인터페이스를 활용하여 사용 난이도가 낮은 인터페이스를 구성하였다.

3. EUCAS

상황인식서비스는 상황(Context)과 그에 따른 행동(Action)을 미리 정의하고 해당 상황이 발생하였을 때 그 행동을 실행하는 서비스로 정의할 수 있다. 이는 주변의 상태와 실행 가능한 행동이 무엇인지 파악할 수 있어야만 상황인식서비스를 정의할 수 있음을 뜻한다. 본 논문에서는 상태

와 행동이 객체지향모델링(Object-oriented Modeling) 기술에 의해 표현된다고 가정한다. 즉, 상황인식서비스가 제공되는 공간인 서비스도메인 안에 모든 상태와 행동은 사용자가 쉽게 이해할 수 있는 장치, 사물 그리고 공간의 단위로 추상화되어 표현되고 각 객체들은 각각의 상태와 행동을 가진다. 예를 들어, 거실이라는 객체에는 온도, 습도, 조도라는 상태를 가지고 창문 Open/Close, 커튼 Open/Close, 라이트 On/Off 등의 행동을 가진 것으로 표현할 수 있다.

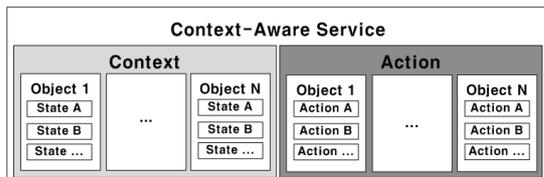
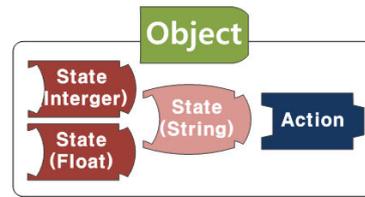


그림 1 상황인식서비스의 구성

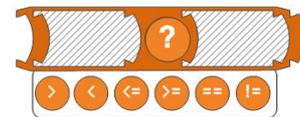
객체지향모델링을 통한 상황과 행동의 표현은 상황인식서비스를 그림 1과 같이 표현할 수 있게 한다. 상황인식서비스를 구성하는 상황과 행동에서 상황은 여러 객체들의 상태 조합으로 구성되고 행동은 제어 대상으로 하는 객체들의 행동 조합으로 구성된다. 객체의 상태들은 논리조건이나 부가적인 표현들이 더해져 상황으로 구성된다. 논리 조건이나 관계 연산이 이에 해당한다. 예를 들어 집안의 온도가 20도 이하이고 창문이 열려 있는 상태를 상황으로 구성한다면 집안 온도와 창문의 상태에 대한 관계 연산과 두 상태가 만족해야 한다는 논리 조건이 더해지는 것이다. 행동은 객체를 제어하고자 하는 상태들의 나열로 구성된다. 예를 들어, DVD 플레이어가 재생되는 상황이라면 행동은 스피커의 볼륨을 키우고 창문을 닫는다고 같이 주변 객체들의 상태 제어 방법이 나열된다.

EUCAS는 사용자가 직접 주변 환경을 구성하는 장치, 공간, 사물들을 객체로 추상화하여 상황인식서비스를 구성할 수 있게 한다. Jigsaw Editor의 경우, 사용자가 이해할 수 있는 수준이라는 기준으로 조각을 구성한다고 가정하고 있지만 EUCAS는 장치, 사물, 공간을 객체로 추상화하여 제공해 준다. EUCAS가 가지는 객체 단위의 퍼즐 조각 구성은 Jigsaw Editor의 퍼즐 조각 구성보다 명확한 제공 기준을 가지게 한다. EUCAS에서 객체는 그림 2.(a)와 같은 형태의 객체 조각으로 표현된다. 하나의 객체 조각은 어떠한 조건을 설정하느냐에 따라 상태 혹은 행동을 표현하게 된다. 예를 들어, TV 객체 조각의 전원 상태를 On으로 설정하면 TV 전원이 On이 된 상황을 표현하게 되는 것이다. 객체의 상태는 세 가지 유형을 가진다. 채널이나 음량과 같은 정수(Integer) 상태와 온도 같은 실수(Float) 상태 그리고 전원의 On/Off와 같은 문자열(String) 상태가 그것이다. 상황인식서비스의 상황은 상태를 표현하는 객체 조각들의 조합으로 구성할 수 있으며 행동은 객체의 행동을 표현하는 조각

들의 조합으로 구성할 수 있다.



(a) 객체 조각



(b) 상태 비교 조각



(c) 조건분기 조각

그림 2 EUCAS의 조각 유형

상황은 상태 간의 비교나 조건에 따른 분기가 있을 수 있으므로 EUCAS에서는 그림 2.(b)의 상태 비교 조각이 제공된다. Jigsaw Editor에서 집안 온도가 집밖 온도보다 높은 상황을 표현하는 조각 제공된다고 하자. 비교 대상이 거실과 안방으로 변경될 경우, 개발자가 그에 상응하는 표현 조각을 Jigsaw Editor에 추가하여 제공해 주지 않는 한 이를 표현할 수 없다. 조합 가능한 경우의 수만큼의 조각을 사용자에게 제공하지 않으려면 개발하려는 조각이 사용자에게 필요한 조각인지 파악하는 작업이 수반된다. EUCAS에서는 상태 비교 조각을 이용하여 다양한 조합을 구성할 수 있도록 하였다. 서로 비교하려는 객체 조각의 상태를 상태 비교 조각에 조합시켜 새로운 조각의 추가 없이도 다양한 상황을 표현할 수 있는 것이다. 예를 들어 거실 온도를 표현하는 객체 조각과 발코니의 온도를 표현하는 객체 조각을 상태 비교 조각에 끼워 넣으면 두 객체 조각이 표현하고 있는 온도를 비교하는 표현을 할 수 있게 된다. 그림 2.(c)의 조건분기 조각을 나타낸다. Jigsaw Editor는 조건 분기를 지원하지 않는다. EUCAS는 조건 분기 조각을 제공하여 분기 표현이 가능하다. 조건 분기 조각은 상황을 구성하는 객체의 상태 중 일부 조건의 만족 여부에 따라 분기하는 경우를 표현하기 위해 필요하다. 예를 들어 집안 공기가 탁할 때, 바깥 공기가 더 탁하다면 공기청정기를 작동시키고 집안 공기가 더 탁하다면 거실 창문을 개방하는 서비스가 있을 때, 공기가 탁하다는 동일한 상황에서 행동만 다르게 나타난다. 조건분기 조각을 이용하면 이러한 표현이 가능하다. 사용자는 EUCAS의 퍼즐 조합객체 조각, 상태 비교 조각, 조건 분기

조각들을 활용하여 자신만의 상황인식서비스를 정의할 수 있다. EUCAS는 사용자가 상황인식서비스로 구성된 조합을 왼쪽부터 순차적으로 해석하여 상황인식서비스를 명세 하는 Rule로 변환하게 된다. EUCAS를 통해 사용자는 코드를 직접 작성하는 어려움 없이 상황인식서비스를 정의할 수 있다.

4. 설계 및 구현

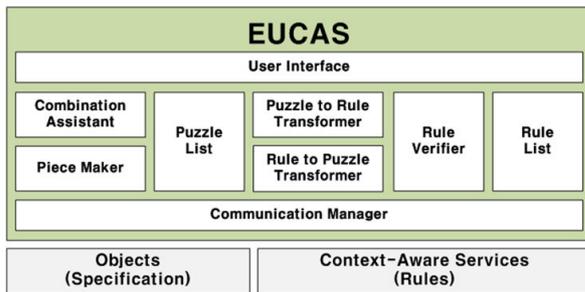


그림 3 EUCAS의 아키텍처

EUCAS는 Java를 이용하여 구현되었으며 그 구조는 그림 3과 같다. 객체에 대한 정보는 외부에서 제공받는다. EUCAS는 크게 입출력을 담당하는 Communication Manager와 내부 처리 모듈들(Piece Maker, Combination Assistant, Puzzle to Rule Transformer, Rule to Puzzle Transformer, Rule Verifier) 그리고 사용자인터페이스로 구성된다. EUCAS는 Communication Manager를 통해 객체들의 정보를 읽어 오고 Piece Maker를 통해 사용자가 사용할 수 있는 객체조각으로 변환한다. Combination Assistant는 정의과정에서 발생하는 조각 간의 연결 문제 해결과 위치 조정을 담당한다. 사용자가 정의하거나 재구성한 상황인식서비스는 Puzzle List로 저장되며 Rule로 변환되어 Rule Verifier의 검증을 거친 뒤 Rule List에 추가된다. Rule List의 변화 내역은 Communication manager를 통해 외부에 제공된다.

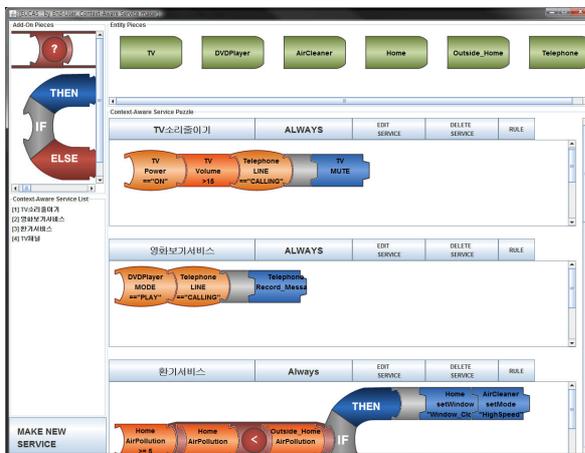


그림 4 EUCAS 프로토타입의 실행 화면

그림 4는 EUCAS의 프로토타입을 실행한 화면을 나타내

고 있다. 좌측 상단에는 상태 비교 조각과 조건 분기 조각이 위치하며 우측 상단에는 사용 가능한 객체 조각들이 위치한다. 사용자는 이 조각들을 선택하여 상황인식서비스 정의에 사용할 수 있다. 그림 5는 EUCAS로 정의한 상황인식서비스의 예로 집안의 공기오염도를 낮추기 위해 상황과 행동을 정의하고 있다. 상황은 집안의 공기오염도가 30이 넘었을 때 집안과 발코니의 공기 오염도를 비교 결과이고 행동은 비교 결과에 따라 집안이 더 높을 때는 창문을 열어 환기시키는 것과 아닐 때 공기청정기를 가동시키는 것이다. 구성된 서비스는 각 조각에 설정된 조건을 수정하거나 조각을 대체하여 재구성 할 수 있다.



그림 5 집안의 공기오염도를 제어하는 상황인식서비스의 예

5. 결론

지금까지 본 논문에는 사용자의 요구를 보다 명확하게 해결하기 위해 정보통신과 관련된 전문지식이 없는 사용자에게 서비스 정의 권한을 부여하는 EUCAS를 제안하였다. 정의가 어렵지 않도록 상황인식서비스의 주요 요소를 객체로 보고 퍼즐 조합 형태의 인터페이스를 통해 구성할 수 있도록 하였다. 이를 잘 활용한다면 개별 사용자들은 자신들에게 잘 맞춰진, 만족도 높은 상황인식서비스의 제공받을 수 있을 것으로 기대된다. 추후, 표현력 확장과 인터페이스 개선, 그리고 적용 범위 확대를 통해 더 다양한 분야에서의 활용할 계획이다.

참고문헌

- [1] T. Y. Sohn, A. K. Dey, iCAP: An Informal Tool for Interactive Prototyping Context-Aware Applications, ACM Conference on Human Factors in Computing Systems, pp974-975, 2003
- [2] Sudha, R. Rajagopalan, M.R. Selvanayaki, M. Selvi, S.T. , Ubiquitous Semantic Space: A context-aware and coordination middleware for Ubiquitous Computing, Coomunication Systems Software and Middleware, pp 1-7, 2007.
- [3] Tao Gu, H. K. Pung, Da Qing Zhang, A service-oriented middleware for building context-aware Service, Journal of Network and Computer Applications, Vol 28, Issue 1, pp 1- 18, 2005

- [4] L. B. Bonanni, E. Arroyo, CH Lee, T. Selker, Smart Sinks : Real-World Opportunities for Context-Aware Interaction, Conference on Human Factors in Computing Systems, pp 1232-1235, 2005
- [5] D. Hong, D. K. W Chiu, V. Y. Shen, V. Y. Shen, Requirements elicitation for the design of context-aware applications in a ubiquitous environment, ACM International Conference on Electronic commerce, Vol. 113, pp 590-596, 2005
- [6] T. Erickson, "Some problems with the notion of context-aware computing", Comm. ACM, Vol. 45, Issue 2, Ontology: different ways of representing the same concept, pp. 102-104, 2002
- [7] J. Humble, A. Crabtree, T. Hemmings, K.P. kesson, B. Koleva, T. Rodden, P. Hansson, "Playing with the Bits" User Configuration of Ubiquitous Domestic Environments, 5th International Conference on Ubiquitous Computing, pp.256-263, 2003
- [8] The OWL Services Coalition, OWL-S : Semantic Markup for Web Services, W3C Member Submission, 2004
- [9] Laura Maria Daniele, Towards a Rule-based Approach for Context-Aware Applications, Thesis for a Master of Science degree in Electronic Engineering, 2007
- [10] P. Korpipää, J. Hakkilä, J. Kela, S. Ronkainen, I. Kumpulainen, Utilising context ontology in mobile device application personalisation, 3rd ACM International Conference on Mobile and ubiquitous multimedia, Vol. 83, pp.133-140, 2004
- [11] A. K. Dey, R. Hamid, C. Beckmann, I. Li, D. Hsu, a CAPpella: programming by demonstration of context-aware applications, ACM, conference on Human factors in computing systems, pp.33-40, 2004
- [12] T. Rodden, A. Crabtree, T. Hemming, B. Koleva, J. Humble, K.P. kesson, P. Hansson, Configuring the Ubiquitous Home, 6th International Conference on the Design of Cooperative Systems, pp227-242, 2004
- [13] R. A. Farmer, B. Hughes, Towards a "personal cost" model for end-user development, ACM, Vol. 158, design centered HCI, pp. 75-82, 2006
- [14] F. Paternò. Towards a research agenda in end user development. ACM Conference on Human Factors in Computing (CHI'03), Workshop on End-User Development, pp. 69-73, 2003.
- [15] A. J. Ko, B. A. Myers, Human factors affecting dependability in end-user programming, ACM SIGSOFT Software Engineering Notes, Vol. 30, Issue 4, pp.1-4, 2005