

컨텍스트 인지 시스템의 아키텍처 패턴

최종명

국립목포대학교 컴퓨터공학전공

e-mail: jmchoi@mokpo.ac.kr

Architecture Patterns of Context-Aware Systems

Jongmyung Choi

Dept. of Computer Eng., Mokpo National University

요 약

컨텍스트 인지 시스템에 대한 관심이 높아지면서, 시스템을 효과적으로 개발할 필요성도 역시 높아지고 있다. 본 논문에서는 컨텍스트 인지 서비스의 특성을 품질 속성에 추가함으로써 서비스 형태에 따라 컨텍스트 인지 시스템의 소프트웨어 아키텍처를 결정하는 방법을 소개한다.

1. 서론

유비쿼터스 컴퓨팅에서 주된 특징 중의 하나는 컨텍스트를 인식해서 사용자에게 가장 적합한 서비스를 제공할 수 있다는 점이다. 컨텍스트 인지 시스템은 환경 혹은 자신의 상태를 의미하는 컨텍스트를 인식하고, 이러한 정보를 바탕으로 사용자에게 적합한 서비스를 제공하는 시스템을 의미한다[1]. 이에 따라 컨텍스트 인지에 관련된 다양한 연구들이 수행되고 있다. 예를 들어, 컨텍스트를 이용한 시스템, 컨텍스트를 모델링하기 위한 연구, 컨텍스트 시스템의 소프트웨어 아키텍처 패턴 등의 연구들이 진행되고 있다.

컨텍스트 인지 시스템은 기존 시스템에 비해 복잡하며, 컨텍스트 인지 서비스 형태에 따라 시스템의 특성이 결정된다. 현재까지 진행된 연구들은 컨텍스트 인지 시스템을 한 개의 소프트웨어 아키텍처로 설명하려고 노력하였다. 그러나 서비스의 형태가 다른 경우에 한 가지 형태로 시스템 아키텍처를 설명하는 것이 어렵다.

본 논문에서는 컨텍스트 인지 서비스를 유형별로 분류하고, 서비스 유형에 따라 시스템에 적합한 아키텍처를 소개한다. 컨텍스트 인지 서비스 유형은 기능적인 특성을 가지고 있을 뿐만 아니라 품질 속성(quality attribute)의 특성을 가지고 있기 때문에 소프트웨어 아키텍처에 지대한 영향을 미친다.

컨텍스트는 시스템의 사용 시나리오 및 구현에서 매우 중요한 의미를 갖는다. 즉, 컨텍스트에 대한 정의와 이에 따른 사용 시나리오가 사소하게 변경되는 경우에 시스템 기능 및 구조에 많은 영향을 미칠 수 있다. 또한 외부 환경 정보를 인식하기 위한 다양한 입력 혹은 센서 장비들은 컨텍스트 정의에 따라 영향을 받게 된다. 컨텍스트 인지 시스템은 확장성이 좋아야 한다. 컨텍스트 인지 시스템은 새로운 장비의 도입이 쉬워야 하고, 새로운 기능을 추가할 수 있어야 하기 때문에 확장성이 매우 중요하다.

본 논문은 5개 장으로 구성되어 있다. 2장에서는 관련연구를 소개하고, 3장에서는 컨텍스트 인지 시스템의 특징을, 4장에서는 서비스 유형별 시스템 아키텍처를 소개한다. 끝으로 5장에서는 결론과 향후 연구 계획을 밝힌다.

2. 관련연구

현재 컨텍스트 인지 시스템의 소프트웨어 아키텍처 및 디자인 패턴에 관련된 연구들이 꾸준히 진행되고 있다. 그러나 현재까지 진행된 대부분의 연구들은 컨텍스트 인지 시스템의 다양한 유형을 다루지 못하고, 한 가지 형태의 서비스에만 집중하는 문제점이 있다.

현재 컨텍스트 인지 시스템을 위한 대표적인 소프트웨어 아키텍처 컨텍스트 인지 시스템을 위한 소

소프트웨어 아키텍처로는 ECA (Event-Condition-Action) 패턴[2], 멀티 에이전트 아키텍처, 블랙보드 패턴 등이 있다. ECA 패턴은 이벤트를 중심으로 처리되는 경우에 매우 효과적으로 처리할 수 있다는 장점을 가지고 있다. 멀티 에이전트 패턴은 시스템의 확장성이 뛰어난 장점을 가지고 있다. 블랙보드 패턴은 컨텍스트를 추론하는 기능이 매우 뛰어난 특징을 가지고 있다.

3. 컨텍스트 인지 시스템의 특징

컨텍스트란 외부 혹은 내부의 상황 정보(사용자 위치, 시간, 외부 온도 등)를 의미하며, 컨텍스트 인지 시스템(context-aware system)[1]은 이러한 정보를 인지하고, 이에 따라 사용자에게 적절한 서비스를 제공한다. 따라서 컨텍스트 인지 시스템은 컨텍스트에 따라 사용자에게 가장 적합한 서비스를 제공하는 것을 목적으로 한다.

컨텍스트 인지 서비스는 컨텍스트의 변화에 의해 자동적으로 실행되거나 혹은 컨텍스트의 변화에 따라 서비스 내용이 변경될 수 있는 서비스를 의미한다. 컨텍스트 인지 서비스는 서비스와 컨텍스트의 관계에 따라서 다음과 같이 5가지 유형으로 분류할 수 있다.

- 컨텍스트 리코딩(context recording) : 현재 컨텍스트에 대한 정보를 저장한다. 예를 들어, Forget-me-not 시스템[7]은 컨텍스트를 이용해서 사용자의 기억을 보조해주는 시스템으로 컨텍스트 정보로 사용자 주위 환경을 기록할 수 있다.
- 트리거(trigger) : 컨텍스트가 지정된 조건을 만족시키는 경우에 서비스가 자동적으로 실행된다. 예를 들어, 사용자가 집에 들어오면, 자동적으로 현관의 불이 켜지는 것은 간단한 형태의 트리거 서비스이다.
- 브랜치(branch) : 현재 컨텍스트에 따라 서비스 내용과 형태가 결정된다. 예를 들어, Bravo[8]가 제시한 시스템은 강의실에서 사용자가 칠판에 접근하는 경우에 사용자의 역할(강사 혹은 학생)에 따라 프레젠테이션에 보이는 내용이 달라진다. 즉, 강사인 경우에는 강의 내용을 보여주고, 학생인 경우에는 학생의 숙제를 화면에 보여준다.
- 자원검색(resource scanning) : 사용가능한 자원들을 찾고, 이를 활용할 수 있는 서비스가 제공된다. 예를 들어 냉장고가 음식 재료들을 인식해서 사용자에게 적합한 음식의 요리법을 소개해주는 것도

있다[6].

- 팔로우미(follow-me) : 팔로우미 서비스[9]는 사용자의 이동에 따라 서비스가 이동할 수 있다. 따라서 팔로우미 서비스는 이동성과 자원 검색이라는 2가지 기능을 동시에 지원하는 서비스이다.

4. 서비스 유형별 시스템 아키텍처

4.1 컨텍스트 리코딩

일반적으로 컨텍스트 리코딩은 내/외부 센서를 이용해서 현재 상황을 기록하고, 향후에 기록된 컨텍스트를 바탕으로 서비스를 제공하는 것을 기본적인 목표로 한다. 따라서 시스템에서 주로 관심을 갖는 컨텍스트 정보로는 사용자 주변의 정보이다. 사용자 주변 정보를 컨텍스트 정보로 사용하기 위해서는 사용자와 함께 이동할 수 있는 핸드폰, PDA 등의 모바일 단말기를 필요로 한다. 모바일 단말기에서 파악한 상황 정보는 서버로 전송되어서 저장되어서 관리된다. 이렇게 저장된 컨텍스트 정보는 향후에 비즈니스 로직과 결합되어서 사용된다.

컨텍스트 인식을 위해서 여러 개의 센서를 사용하는 경우에 센서를 통한 이벤트들은 개별적인 아닌 연속적인 스트림 형태로 이벤트들이 발생하게 되고, 이에 따라 시스템은 스트림 이벤트들을 처리할 수 있는 구조를 가져야 한다. 그림 1은 컨텍스트 리코딩 서비스를 위한 아키텍처이다.

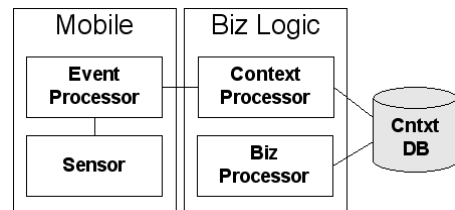


그림 1. Pipe 패턴

4.2 트리거

트리거는 특정 컨텍스트로 들어가는 경우에 자동적으로 어떤 서비스가 수행되는 것이다. 예를 들어, 사용자가 집에 들어오면, 자동적으로 현관의 불이 켜지는 것은 간단한 형태의 트리거 서비스이다. 트리거 서비스에서 가장 중요한 점은 이벤트의 발생이다. 컨텍스트 변화라는 이벤트가 발생하는 경우에 서비스가 수행되기 때문에 트리거 서비스는 이벤트 드리븐의 ECA 아키텍처가 적합하다. 그림 2는 트리거 서비스를 위한 ECA 아키텍처이다.

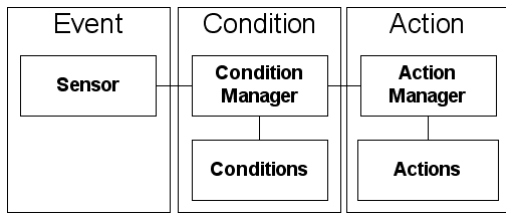


그림 2. ECA 패턴

트리거 서비스에서 비즈니스 로직은 모듈화되어 있어야 하며, 컨텍스트는 비즈니스 로직을 호출하는 역할을 한다. 따라서 비즈니스 로직과 컨텍스트 로직은 느슨한 형태로 연결되어 있다.

4.3 브랜치

브랜치는 컨텍스트에 따라서 비즈니스 서비스 내용이 달라지는 것이다. 서비스 내용이 달라지는 것은 여러 레벨이 존재할 수 있다. 어떤 경우에는 알고리즘이 달라질 수도 있고, 또 다른 경우에는 사용자에게 전달되는 데이터 형태가 달라질 수 있다. 따라서 서비스 내용이 달라지는 것을 적절하게 분류할 필요가 있다.

- 프레젠테이션 레벨 : 컨텍스트에 따라 사용자에게 전달되는 데이터 포맷이 달라진다. 예를 들어, 사용자의 디바이스 환경에 따라 A 사용자에게는 동영상 정보를 제공하고, B 사용자는 디바이스 특성상 동영상을 볼 수 없다면, 텍스트 형태로 데이터를 전달하는 경우이다. 프레젠테이션 레벨은 디바이스 환경에 따라 미리 내용이 만들어져 있어서 컨텍스트에 의해서 결정만 되는 정적인 형태와 실행 시에 자동적으로 만들어야 하는 동적인 형태가 있다.

브랜치 서비스가 프레젠테이션 레벨에서 이루어지는 경우에는 MVC 패턴의 일종으로 볼 수 있다. 즉, 서비스의 내용(Model)은 동일하지만, 서비스의 뷰(View)는 변경이 되며, 뷰를 결정하는 것이 컨텍스트(Controller)의 역할이다. 그림 3은 MVC 아키텍처를 보여준다.

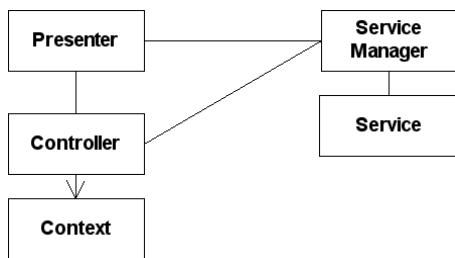


그림 3. MVC 패턴

• 로직 레벨

- 서비스 레벨 : 서비스 레벨은 컨텍스트에 따라 제공하는 서비스들이 미리 구축되어 있고, 컨텍스트 정보가 매개 변수 형태로 전달되는 경우이다.
- 알고리즘 레벨 : 알고리즘 레벨은 컨텍스트에 따라 사용되는 알고리즘이 바뀔 때 발생한다. 이때에도 컨텍스트별로 사용하는 알고리즘은 미리 만들어져 있고, 컨텍스트는 매개 변수의 역할만 하게 된다.
- 필터 레벨 : 필터 레벨은 서비스의 시작과 끝 혹은 중간에서 컨텍스트에 따라 수행하는 내용이 달라지는 경우이다. 필터 레벨은 비즈니스 로직과 컨텍스트가 밀접하게 연결되어 있다.

브랜치 서비스가 로직 레벨에서 이루어지는 경우에는 ECA 패턴에서 이벤트가 빠진 것으로 파악할 수 있다. 또한 액션 부분은 필터 패턴으로 알고리즘에 따라 동적으로 변환이 가능해야 한다. 이러한 패턴을 CA(Condition-Action) 패턴이라고 하며, 그림 4는 CA 패턴을 보여준다.



그림 4. CA 패턴

4.4 자원검색

자원 검색은 주위의 자원을 검색해서 이를 컨텍스트로 활용하는 서비스이다. 예를 들어, 휴대폰이 주위에 프린터가 존재하는 경우에 휴대폰에 저장된 사진을 그 프린터로 출력할 수 있는 서비스를 제공하는 것은 대표적인 자원 검색 서비스이다. 이 서비스는 컨텍스트를 이용해서 새로운 비즈니스 서비스를 생성한 것으로 볼 수 있다. 왜냐하면, 휴대폰 자체는 프린트에 관련된 비즈니스 기능이 없는데, 주위에 있는 프린터를 이용해서 새로운 비즈니스 기능을 추가한 것이다. 또 다른 예는 냉장고가 음식 재료를 인식해서 사용자에게 적합한 음식의 요리법을 소개해주는 것도 있다[10]. 이 서비스는 컨텍스트가 비즈니스 서비스에 대해서 매개 변수의 역할을 수행한다고 할 수 있다. 그림 5는 자원검색에 적합한 ESA 아키텍처 모습을 보여준다.

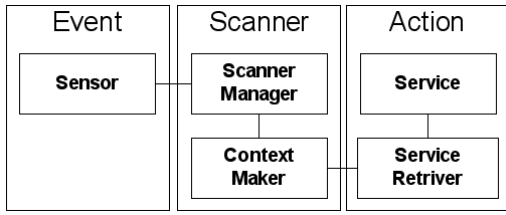


그림 5. ESA 패턴

4.5 팔로우미

팔로우미 서비스는 사용자의 이동에 따라 서비스가 이동하는 경우이다. 팔로우미 서비스는 이동성과 자원 검색이라는 2가지 기능을 동시에 지원하는 서비스이다. 즉, 사용자 이동에 따라 현재 사용자의 작업 내용을 기억 및 이동해야 하며, 새로운 위치에서 자원 검색을 통해서 작업 내용을 지속할 수 있도록 해야 한다. 이러한 경우에도 컨텍스트가 새로운 비즈니스 로직을 생성하였다고 볼 수 있다. 그림 6은 팔로우미 패턴에 적합한 CSA 아키텍처를 보여준다.

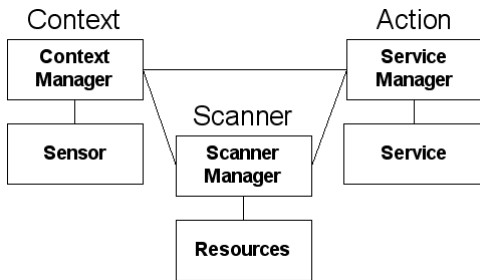


그림 6. CSA 패턴

5. 결론

유비쿼터스 컴퓨팅에서 사용자 측면에서 가장 중요한 특징 중의 하나는 컨텍스트 인지 서비스이다. 컨텍스트 인지 서비스는 사용자가 복잡한 시스템의 내부 혹은 사용법을 모르면서도 쉽게 원하는 서비스를 받을 수 있도록 한다. 이러한 특징 때문에 컨텍스트 인지 시스템을 위한 많은 연구와 시도가 있었다. 그러나 현재까지 개발된 시스템들은 프로토타입 수준에 머무르고 있다.

컨텍스트 인지 시스템은 다양한 입력 시스템으로 구성되어 있기 때문에 복잡하고, 지능적인 서비스를 제공해야 하기 때문에 시스템 아키텍처를 올바르게 파악하는 것이 필요하다. 따라서 본 논문에서는 컨텍스트 인지 시스템의 서비스 형태를 중심으로 시스템의 아키텍처를 결정하는 방법을 소개하였다. 서비스 형태에 따른 소프트웨어 아키텍처를 결정하는 방법은 시스템 개발에서 많은 시행착오를 감소시켜주

고, 효과적으로 시스템을 개발할 수 있도록 도움을 줄 것이다.

참고문헌

- [1] Anind K. Dey, Gregory D. Abowd, "Towards a Better Understanding of Context and Context-Awareness," *Workshop on The What, Who, Where, When, and How of Context-Awareness in CHI'00*, 2000.
- [2] Wolfgang Beer, et. al. "Modeling Context-Aware Behavior by Interpreted ECA Rules", *LNCS(Mobile and Ubiquitous Computing)*, pp. 1064-1973, 2004.
- [3] Gustavo Rossi, Silvia Gordillo, and Fernando Lyardet, "Design Patterns for Context-Aware Adaptation"
- [4] James A. Landay and Gaetano Borriello, "Design Patterns for Ubiquitous Computing", *IEEE Computer*, pp. 93-95, Aug., 2003.
- [5] Bill Schilit, *A context-aware system architecture for mobile distributed computing*, Ph.D. Dissertation, Columbia University, 1995.
- [6] P.J. Brown, "The Stick-E Document: A Framework for Creating Context-Aware Applications," *EP'96*, pp. 259-272, 1996.
- [7] Nicky Kern, et. al., "Context Annotation for a Live Life Recording", *Pervasive 2004 Workshop on Memory and Sharing of Experiences*, 2004.
- [8] Bravo J., Hervas R, Chavira G, and Nava S., "Modeling Contexts by RFID-Sensor Fusion", *Proc. of Pervasive Computing and Communications Workshops*, pp. 30-34, 2006.
- [9] Bill N. Schilit, Norman Adams, and Roy Want, "Context-Aware Computing Applications", *Proc. of the Workshop on Mobile Computing Systems and Applications*, pp. 85 - 90, 1994.
- [10] M. Levinson, "All-in-One Appliance - THE REFRIGERATOR," *CIO Magazine*, Feb., 2003.
- [11] Chun-Dong Wang and Xiu-Feng Wang, "Multi-agent Based Architecture of Context-aware Systems", *Int'l Conf. on Multimedia and Ubiquitous Eng.*, 2007.