
Large-scale 맥락 인식 시스템의 평가 방법에 대한 연구¹

Study on Evaluating a Large Scale Context-Aware System

오유수, Yoosoo Oh, 우운택, Woontack Woo
광주과학기술원 U-VR 연구실

요약 맥락 정보와 맥락 인식 시스템에 대한 연구는 지난 10 여 년 동안 유비쿼터스 컴퓨팅 분야에서 중요한 이슈로 다루어졌다. 대부분의 맥락 인식 시스템은 위치 정보와 같이 단일 형태의 맥락 정보를 위해서 설계되었거나 연구실 수준의 크기로 제한되었다. 그러나 많은 종류의 센서와 actuator 를 포함하고 다수의 관리 도메인으로 확장 가능한 스케일이 큰 시스템에 대한 개발 및 평가는 여전히 미흡한 수준이다. 특히, 맥락 퓨전과 추론 구조를 가지는 Large-scale 의 맥락 인식 시스템에 대한 평가 방법이 필요한 실정이다. 본 논문에서는 휴리스틱 평가를 이용한 Large-scale 맥락 인식 시스템의 평가 방법에 대하여 제안한다. 그리고 우리는 동적인 맥락 인식 시스템을 지원하고 맥락 퓨전 및 추론을 위한 메커니즘을 포함하는 기본 구조에 대해서 자세히 설명한다. 맥락 인식 시스템 평가를 위해서 제안된 접근법은 사용자 인터페이스 도메인에서 잘 알려진 전문가에 의한 평가 방법으로 Large-scale 맥락 인식 시스템에 적합하도록 특별히 선택된 heuristics 집합을 이용하는 휴리스틱 평가(Heuristic Evaluation)이다.

핵심어: Context Fusion, Context Reasoning, Heuristic Evaluation, Large-scale Context-aware System

1. 서론

유비쿼터스 컴퓨팅 분야에서 맥락 인식 시스템은 지금까지 중요한 연구 주제로 다루어지고 있다 [1]. 많은 관련 시스템들이 개발되었으나 연구실보다 큰 규모의 맥락 인식 시스템에 대한 개발 및 평가는 여전히 미흡한 수준이다. 특히, 시물레이션, 설치, 디버깅, 동적인 변화 적응, 새로운 어플리케이션의 추가 등과 같은 맥락 인식 시스템의 라이프 사이클에 대한 총체적인 지원과 같은 이슈는 상대적으로 적은 관심을 받아왔다. 이와 같은 내용들은 일반적인 해결책을 제시하기 위해서 구조적인 수준에서 설계되고 평가되어야 한다.

Large-scale의 맥락 인식 시스템을 위해서 다양한 소스 및 도메인으로부터 얻어진 맥락 정보들을 통합하는 과정은 시스템 설계 및 개발에서 중요하다. 맥락 퓨전 및 추론은 고수준의 맥락 인식 과정이며 Large-scale의 맥락 인식 시스템에서 다양한 맥락 정보들을 효율적으로 처리할 수 있는 방법이다. 따라서, Large-scale의 맥락 인

식 시스템에 대한 평가는 맥락 퓨전 및 추론 구조를 이해하는 것으로부터 시작된다.

Large-scale의 맥락 인식 시스템 평가는 시스템의 설계부터 개발까지 어느 과정에서도 가능해야 한다. Large-scale의 맥락 인식 시스템의 완성은 많은 시간과 노력이 요구되므로 완성된 시스템에 대한 평가를 이용하여 시스템을 디버깅하는 것보다 시스템 시물레이션에 대한 평가를 디버깅하는 것이 효율적이다.

Large-scale의 맥락 인식 시스템의 설계 및 개발 시에 제시된 접근법이 적절한 것인지를 보여주는 절차는 중요하다. 많은 관련 연구에서는 맥락 인식 구조에 대한 접근법의 적합성을 평가한다. 그러나 Large-scale의 맥락 인식 구조를 위한 평가 방법으로는 적절하지 못하다. 또 다른 접근법은 관련 연구와의 비교를 매트릭스 형태로 표현하여 평가하는 것이다. 그러나 이 방법은 평가자의 관점에 따라서 비교 결과가 영향을 받을 수 있다. 따라서 본 논문에서는 Large-scale의 맥락 인식 시스템 평가를 사용자 인터페이스 도메인에서 잘 알려진 휴리스틱 평가(Heuristic Evaluation)를 이용한다 [2].

그러므로, 본 논문은 휴리스틱 평가를 이용한 Large-scale 맥락 인식 시스템의 평가 방법에 대하여 기술한다.

¹ 본 연구는 정보통신부의 선도기반기술개발 사업의 지원으로 수행되었음.

본 논문은 Large-scale 맥락 인식 시스템을 위한 새로운 구조를 소개하고 Large-scale 맥락 인식 시스템에 적합한 heuristics 집합을 선별하고 전문가 평가에 의해서 발견된 결과를 논의한다.

본 논문은 다음과 같이 구성된다. 2장에서는 Large-scale 맥락 인식 시스템의 아키텍처 및 프로토타입 구현에 대하여 설명한다. 3장에서는 Large-scale 맥락 인식 시스템을 평가하기 위한 접근법인 휴리스틱 평가에 대하여 자세히 기술한다. 그리고 4장에서는 휴리스틱 평가 결과에 대하여 논의한다. 마지막으로 5장에서는 결론 및 추후연구에 대하여 기술한다.

2. Large-scale 맥락 인식 시스템

Large-scale 맥락 인식 시스템을 설계하고 개발하기 위해서는 다음과 같은 요소들을 고려해야 한다.

- 많은 센서들과 actuator들을 포함하는 대형 시스템을 위한 맥락 인식 기법
- 센서나 actuator의 추가, 교체, 고장과 같은 동적 변화에 대한 맥락 인식 시스템의 수용력
- 설치, 시뮬레이션, 디버깅 등과 같은 시스템 라이프 사이클 지원
- 센서와 서비스 간의 상호 독립성, 맥락 퓨전과 추론 기법

2.1 Large-scale 맥락 인식 시스템 프로토타입

본 논문에서는 Large-scale 맥락 인식 시스템 개발을 위한 프로토타입을 그림 1과 같이 구현하였다. 구현된 시스템 프로토타입은 물리적인 센서 및 액추에이터와 시뮬레이션 센서 및 액추에이터를 병행하여 사용하는 빌딩 응용 시스템이다. 구현된 물리적인 센서로는 착용형 activity 센서, 사용자의 상태 모니터링 센서, 그리고 사용자 정보와 상호작용 기능을 가진 모바일 센서이다. Large-scale 맥락 인식 시스템 프로토타입은 이와 같은 센서들과 빌딩 시뮬레이터, 그리고 맥락 인식 아키텍처로 이루어진다.

착용형 activity 센서는 기존 연구[2]와 같이 작은 프로세싱 파워를 가지는 센서 네트워크 플랫폼인 파티클 컴퓨터를 기반으로 구성된다. 착용형 activity 센서는 사용자의 activity와 자세를 알아내기 위하여 3축 가속도 데이터를 생성하고 사운드, 주변온도, 주변밝기 정보와 같은 센싱 데이터도 생성해낸다. 상태 모니터링 센서는 사용자의 작업 상황을 알기 위한 센서로서 PC의 로그인 상태나 Skype 메신저[4]의 사용자 상태를 모니터링 한다. 상태 모니터링 센서는 사용자의 프로파일 정보와 PC의 고유 아이디와 설치 장소와 같은 메타 데이터를 이용하여 사용

자의 상태를 맥락 정보로 생성해낸다. 사용자 프로파일 및 제어를 위한 모바일 센서는 Nokia6600 모바일 폰 [5] 기반으로 구현되었으며, 사용자의 개인 정보(프로파일, 스케줄)를 생성 및 관리하고 환경의 장치(전등, 히터, 도어락)를 제어할 수 있다. 모바일 센서는 개인 정보와 제어 정보를 사용자 맥락 정보로 생성해낸다.

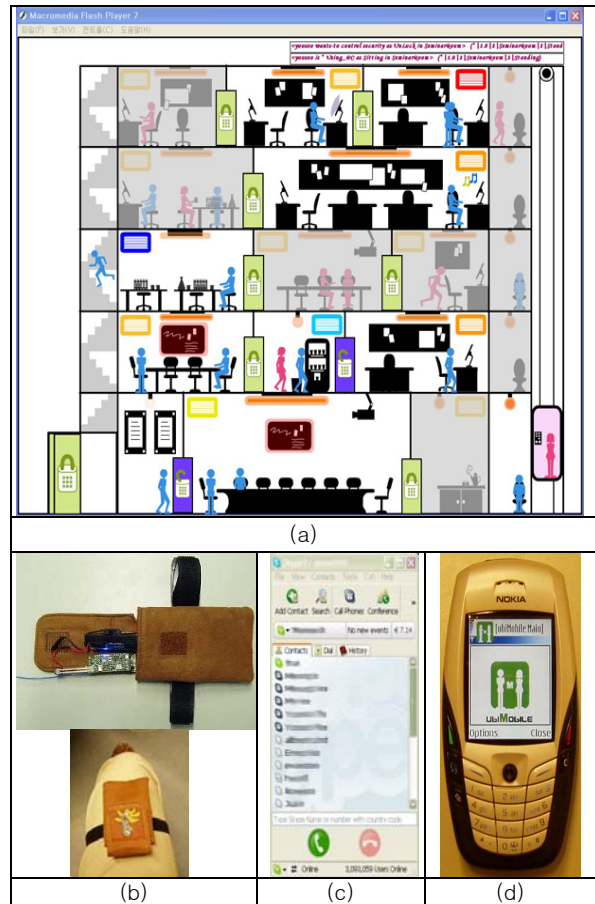


그림 1 프로토타입 시스템. (a) Flash로 개발된 맥락 인식 빌딩 시뮬레이터(ubiBuilding) (b) 착용형 activity 센서 (c) Skype를 이용한 상태 감지 센서 (d) 모바일 폰으로 개발된 프로파일 및 제어 센서

ubiBuilding 시뮬레이터는 그림 1 (a)에서 보는 것과 같이 Macromedia Flash로 구현된 Large-scale 맥락 인식 빌딩 시뮬레이션 시스템이다. ubiBuilding 시뮬레이터는 5층의 19개 방과 41개의 액추에이터(9개의 도어락, 19개의 전등, 13개의 히터)로 구성된다. 또한, 100개의 사용자 심볼을 동시에 처리하며, 4가지의 사용자 행동(앉기, 서기, 걷기, 뛰기)을 나타낼 수 있다.

시스템 시뮬레이션은 시스템 개발 이전에 빠른 prototyping과 평가를 가능하게 한다. 특히, 맥락 인식 구조와 같은 소프트웨어가 미리 개발되고 새로운 환경을 위한 맥락 인식 시스템 개발이 시작될 경우에 시스템 시뮬레이션은 유용하다. 그림 1 (a)와 같이 구현된 ubiBuilding 시뮬레이터는 제안된 맥락 인식 아키텍처(소프트웨어)에 대한 평가와 실제 빌딩 환경을 위한 Large-

scale 맥락 인식 시스템의 빠른 구축을 가능하게 한다.

2.2 Large-scale 맥락 인식 시스템 아키텍처

Large-scale 맥락 인식 시스템의 핵심 구조는 그림 2와 같은 맥락 퓨전과 추론 메카니즘이다. 그림 2의 퓨전 관련 모듈은 사용자 중심의 정형화된 맥락 모델인 5W1H(육하원칙) 맥락 기법[3]에 의하여 기술되며, 추론 모듈은 JESS (Java Expert System Shell)에 기반하여 동작된다.

그림 2의 맥락 통합기는 다양한 종류의 센서로부터 다수의 맥락 정보 입력을 동시에 통합하여 각 거주자의 activity에 대한 의도를 추론한다. 맥락 정보 퓨전 및 추론은 맥락 인식 시스템의 새로운 아키텍처인 맥락 통합기에 의해서 제공되는 핵심 기술 요소이다. 그림 2는 맥락 통합기의 아키텍처를 보여준다. 맥락 통합기는 Context Object Analyzer, Context Repository, Preliminary Context Fusion module, Context Inference Engine, 그리고 Integrated Context Generator로 구성된다.

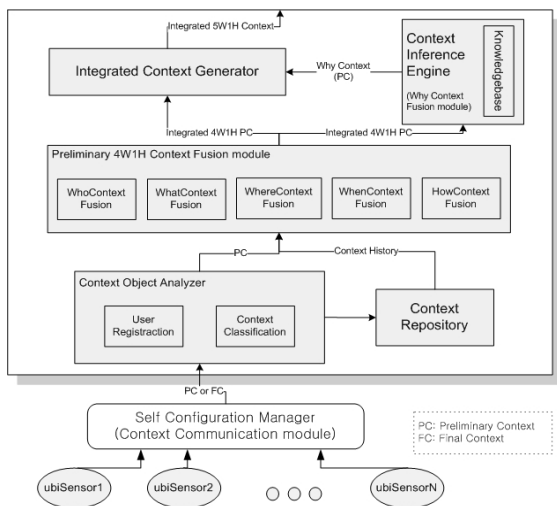


그림 2 Large-scale 맥락 인식 시스템을 위한 맥락 퓨전 및 추론 아키텍처

Context Object Analyzer는 활성영역에 위치한 다양한 종류의 센서들로부터 맥락 정보를 수집하고 ‘누가(Who)’의 부분 맥락 중의 사용자의 신원 정보에 따라서 수집된 맥락 정보들을 분류한다. Context Repository는 통합된 맥락 정보들의 히스토리를 저장하고 관리한다. Preliminary Context Fusion module은 입력된 초별 맥락 정보들을 각 특성에 따라서 4W1H(Who, What, Where, When, and How)의 통합 맥락으로 구성한다. 통합 맥락은 적절한 융합 방법을 적용하여 생성된다. 초별 맥락 (Preliminary Context)은 특징 정보나 센서 상세를 나타

낸다. 통합 맥락(Integrated Context)은 맥락 통합기에 의해서 생성된 사용자 의도 및 감정 정보를 포함하여 완전한 5W1H 맥락을 형성한다. Context Inference Engine은 Preliminary Context Fusion module의 결과를 이용하여 ‘왜(Why)’의 맥락을 추론해낸다. 이 추론 엔진은 시스템의 행동을 기술하는 facts와 rules로 구성되는 지식 베이스를 포함한다. Integrated Context Generator는 이미 구성된 4W1H의 맥락 정보와 추론된 ‘왜(Why)’의 맥락을 통합하여 5W1H의 통합 맥락을 생성한다. 이 5W1H의 통합 맥락은 거주자의 신원, 위치, 행동, 패턴, 의도 등과 같은 정보를 포함한다.

맥락 통합기는 맥락 통합 과정에서 특히 맥락의 재사용을 위한 메카니즘으로 구성되었다. 맥락의 재사용은 추론 과정에서 이전 과정의 추론된 결과를 다시 사용하는 연속 과정의 결과물로서 시스템의 효율성을 높일 수 있다. 맥락의 재사용의 과정이 진행될수록 높은 수준의 맥락이 추론될 수 있다. 또한, 맥락의 재사용은 Context Repository에서 관리되는 맥락 히스토리에서 저장된 맥락 정보를 사용함으로써도 이루어진다.

맥락 통합기는 사용자의 행동이나 제스처를 추론할 수 있다. 이때, 이전의 맥락 정보는 중요한 단서가 된다. 그래서 맥락 히스토리는 사용자의 행동을 평가하기 위해서 사용된다. 맥락 통합기는 사용자의 위치 좌표와 방위 정보를 이용하여 사용자의 실내 위치나 자세를 추론한다. 또한, 맥락 통합기는 사용자의 다음 행동을 예측하기 위해서 추론된 사용자자의 위치나 자세 등을 이용한다.

맥락 통합기에 기반한 맥락 인식 시스템의 일반화된 행동과 기능은 규칙(rule)에 의하여 생성될 수 있다. 개발자나 관리자에 의한 시스템의 변화 및 재구성은 규칙(rule)을 추가, 삭제 및 수정함으로써 가능하다. 이와 같은 규칙(rule)을 간략화하기 위해서 일반적인 텍스트 에디터를 이용하여 쉽게 재구성할 수 있다. 그러므로 맥락 통합기에서 소스 코드를 바꾸지 않고도 전체 시스템 행동에 대한 변화가 가능하다. 맥락 통합기의 Context Inference Engine은 JESS (Java Expert System Shell)를 이용하여 구현되었다. 구현된 규칙(rule)은 LISP 언어의 구문에 기반한다. 다른 선언적 프로그래밍 언어와 비교하여 이와 같은 구현은 규칙(rule)의 순서에 상관없이 재구성을 쉽게 한다.

3. Large-scale 맥락 인식 시스템의 평가

미들웨어나 아키텍처를 평가하는 것은 매우 어려운 일이다 [6]. 특히, 선택된 전문가가 관련 연구 분야의 전문 연구 지식을 갖지 않았거나, 전문가들 간의 동기가 이루어지지 않는다면 시스템이 적절히 평가되지 않는다. 따라서, 본 논문에서는 사용자 인터페이스 평가 도메인에서 잘 사용되는 채택된 전문가들에 의한 평가 방식을 도입한다 [7], [8], [9], [10]. 먼저, Large-scale 맥락 인식 아

키택처의 평가를 위한 기본 원리인 heuristics를 정하고, Large-scale 맥락 인식 시스템 프로토타입과 같은 구체적인 예를 제시한다.

시스템 평가를 위한 heuristics 집합은 표1의 11개 목록이다.

표 1. Large-scale 맥락 인식 아키텍처 평가를 위한 11개의 heuristics

Heuristics		내용
H1	Separation of Concerns	센서, 프로세싱, 서비스, 어플리케이션과 같은 요소들의 분리 구조. (각 요소의 변화는 다른 요소에 독립적인가)
H2	Flexibility and Openness	시스템의 공개성. (다른 요소의 통합이 가능한가)
H3	Scalability	시스템의 복잡도는 규칙(rules)의 수와 복잡도의 영향을 받으나 센서, 액추에이터, 사용자의 수에는 영향을 받지 않음. (새로운 요소의 추가가 전체 성능 저하에 영향을 미치는가)
H4	Support for Reuse	맥락 정보의 재사용 또는 맥락 히스토리의 활용. (사용된 맥락 정보가 다시 활용되는가)
H5	Support during development, debugging and deployment	개발, 디버깅 및 프로토타이핑의 지속적인 지원. (디버깅 및 오류 추적이 가능한가)
H6	Explanations and Accountability	시스템 동작 및 원인 규명의 명확성. (시스템에 의한 동작을 이해할 수 있는가/시스템의 동작이 맥락 정보의 흐름을 보여주는가)
H7	Security and Privacy	보안 및 사생활 보호. (인증 수단이 제공되는가/아키텍처에서 지원되는가)
H8	Reliability	시스템 동작의 신뢰성 및 사용자 의도 반영. (같은 상황이 시스템 행동의 같은 결과를 내는가)
H9	Match between context-aware system and the real world	맥락 인식 시스템과 실생활과의 부합도. (실제 외부 변화가 맥락 인식 시스템에 즉각적으로 반영되는가)
H10	Manual override	사용자 제어(맥락 정보에 의한 시스템의 결정을 사용자가 바꿀 수 있는 수단이 있는가)
H11	Reconfiguration and Management	쉬운 재구성. (런타임시에 시스템 행동을 채택하고 변화시키기 쉬운가)

Large-scale 맥락 인식 시스템의 평가는 시스템에 적합한 heuristics를 이용하여 선별된 전문가에 의한 휴리스틱

평가로 진행된다. Heuristics를 이용하는 전문가 평가의 기본적인 개념은 적은 수(4~6명)의 전문가들이 주어진 heuristics 집합에 기반하여 시스템을 평가하는 것이다. 전문가들은 맥락 인식 시스템 도메인에 대한 지식 또는 경험을 가진 사람으로 제한된다. 또한 평가를 수행하는 전문가들은 제시된 시스템에 대한 깊은 지식을 지니고 있지 않아야 한다. 전문가들이 heuristics 집합 중에서 발견한 문제점은 1점(사소한 문제)~5점(가장 심각한 문제)으로 심각도(severity)가 표시된다.

휴리스틱 평가를 위해서 맥락 인식 시스템을 설계하거나 개발한 경험이 있는 5명의 전문가들(컴퓨터과학 분야의 박사과정 4명과 박사 1명)을 선정하였다. 선택된 전문가들은 제시된 Large-scale 맥락 인식 시스템 프로토타입 개발에 참여하지 않았다. 평가를 위해 걸린 시간은 각 전문가 당 평균 1.2시간이 소모되었다. 평가 결과, 전문가들은 심각도 (severity)를 판단하여 문제가 되는 heuristics를 지적하고 그 이유에 대한 자세한 설명을 기술하였다. 또한, 전문가들은 평가 과정이나 평가한 프로토타입 시스템에 대한 코멘트를 주었다.

사용자 인터페이스의 기능들은 가시적이지만 시스템 구조는 비가시적이며 명확한 관찰이 어렵기 때문에, 사용자 인터페이스 평가와 시스템 구조 평가는 큰 차이가 있다. 따라서 사용자 인터페이스에서 이용된 휴리스틱 평가를 제안된 평가 방법으로 이용하기 위해서, 본 논문에서는 heuristics 집합과 함께 시스템에 대한 문서와 구현된 어플리케이션을 제공하는 방법을 제안한다.

평가 결과, 모든 전문가들이 시스템 아키텍처와 관련된 문항에 대해서 지적하고 높은 심각도(severity)를 표시했다. 특히, H-7 (Security and Privacy)는 5명의 전문가 중 2명은 심각한 문제, 나머지 2명은 사소한 문제로 지적하여 제시된 heuristics 중 가장 높은 심각도(severity)를 나타내었다. 보안 및 사생활 보호 문제는 중요한 이슈이지만 제시된 아키텍처에서는 잘 해결하지 못하였다고 평가되었다. 처음 아키텍처 설계 시 보안 및 사생활 보호 이슈는 부가요소로 간주되었으나, 평가 결과에 의하여 다음 설계 시에는 필수요소로 고려되어야 한다.

두 번째로 지적된 문제점은 H-5 (Support during development, debugging and deployment)의 디버깅 문제이다. 이 평가는 예상치 못한 결과였다. 제시된 아키텍처 설계 시 중요 기여 중의 하나가 라이프 사이클(개발, 디버깅 및 프로토타이핑)을 지원하는 것이었기 때문이다. 제시된 평가 자료에는 디버깅을 위한 이벤트가 생성되는 이유와 액션이 이루어지는 과정을 그림으로 자세히 묘사하여 시스템이 충분히 디버깅을 지원한다고 믿었으나, 전문가들은 이것이 아직은 초기단계로 판단하였고 히스토리 추적 및 관리와 같은 기능이 추가되어야 한다고 기술하였다.

세 번째는 3명의 전문가들이 사소한 문제점으로 지적한 H-3 (Scalability)이다. 이와 같은 결과는 전문가들이

맥락 정보 처리 (규칙[rules] 엔진)에 대한 깊은 지식이 부족하여, 다수의 센서들과 액츄에이터들이 일으키는 효과를 정확히 관측하지 못함으로 야기되었다. 따라서, 향후 시스템의 확장성을 명확하게 보여주기 위하여 맥락 통합기의 내부 과정을 묘사하고 분석한 결과를 보여주는 시뮬레이션이 개발되어야 한다.

마지막으로 H-11 (Reconfiguration and Management) 이 2명의 전문가들에 의하여 룰이 생성되고 수정되는 과정이 텍스트 기반으로 이루어져서 어렵다고 지적되었다. 전문가들의 평가 결과는 재구성 및 관리에 대한 내용 보다는 인터페이스에 대한 외부 기능에 대한 비평이었다. 이와 같은 인터페이스의 역할도 주된 이슈에 영향을 미칠 수 있음을 알게 되었다. 개발 시에는 텍스트 기반 룰 추가 및 수정 작업이 다양한 에디터에서 활용 가능하므로 쉬운 방법이라고 생각되었으나, 향후 GUI 기반으로 발전시켜 좀더 사용자 친화적인 인터페이스를 구축해야겠다.

시스템 평가 결과, 일반적인 코멘트로는 아키텍처와 관련된 이슈와 평가 방법 자체에 대한 내용이었고, 향후 아키텍처 개발에 도움이 되는 유익한 내용이었다. 전문가들은 제안된 Large-scale 맥락 인식 시스템의 구조가 security와 privacy 문제에 대한 해결 방안을 제시하기를 기대하였다. 또한, Large-scale 맥락 인식 시스템 설계 시에 맥락 히스토리를 관리하는 정책의 필요성에 대한 코멘트가 있었다.

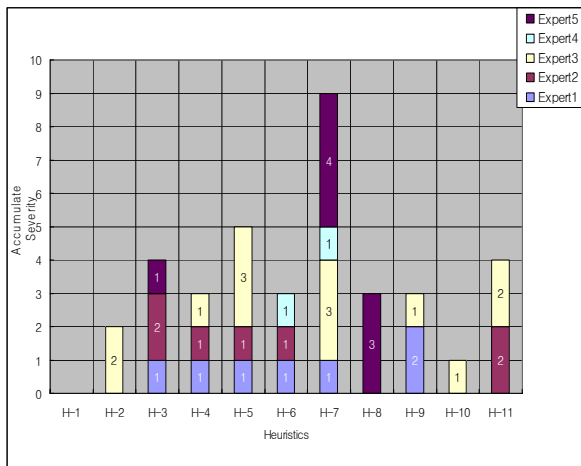


그림 3. 휴리스틱 평가 결과 분석. (그래프는 누적 심각도 [severity]를 나타냄)

4. 평가 분석

맥락 인식 아키텍처나 미들웨어에 대한 전문가 평가는 객관적인 관점에 의한 피드백을 얻을 수 있는 효과적인 방법이다. 같은 평가 방법이라도 사용자 인터페이스가 시각적이며 외형적인 것과는 달리, 아키텍처나 미들웨어는 비시각적이며 외부에서 쉽게 관측되지 않는 어려움이 있다. 그러나 본 논문에서는 시스템 데모, 관련 설명 문서와

동작되는 시스템 프로토타입이 전문가 평가를 수행하기 위해 준비되어 있다면 아키텍처나 미들웨어의 평가가 가능하다는 것을 보여주었다.

휴리스틱 평가를 위한 전문가 선택은 어려운 작업이었다. Large-scale 맥락 인식 시스템 분야에서 주어진 heuristics에 깊은 지식을 갖는 것은 평가를 위한 필요조건은 아니었다. 본 평가에서는 맥락 인식에 대한 기본적인 지식과 소프트웨어 아키텍처에 대한 전문적인 지식을 갖추면 전문가 평가자로서 충분한 자격 요건이 되었다. 또한, 선택된 전문가의 다양성은 평가로부터 훌륭한 피드백을 얻는 데 가치가 있었다.

평가 결과 분석에서 심각도(severity)가 전문가마다 다르게 나타났다. 이것은 관심사와 시스템에 대한 이해도가 서로 다른 전문가들의 평가 결과를 평가 진행 책임자 (coordinator)가 전문가들과 의논하는 시간을 갖는 것이 중요하다는 것을 보여준다. 또한, 사용자 인터페이스 도메인과는 달리, 비시각적인 이슈들에 대한 모호성이 이와 같은 결과를 얻게 된 것을 알 수 있었다.

대체로 휴리스틱 평가 방법은 객관적인 평가를 보장하지는 못하지만 개발자에게 세부적인 피드백을 줄 수 있는 정성적인 평가의 관점을 제시한다. 평가 사례 연구에 기반하여 본 논문에서는 개발 중인 복잡한 시스템의 평가를 상대적으로 적은 노력으로 독립적인 관점의 평가 결과를 얻을 수 있는 접근법으로 휴리스틱 평가를 채택하였다.

5. 결론

본 논문에서 우리는 Large-scale 맥락 인식 시스템을 위한 heuristics 집합을 정의하고 전문가에 의한 평가 방법인 Heuristic Evaluation을 이용한 접근법에 대하여 제안하였다. 맥락 인식 시스템, 시스템 아키텍처 및 미들웨어에 대한 평가 방법 조사 결과, 본 논문에서는 개발 중인 맥락 인식 시스템의 평가를 위하여 휴리스틱 평가를 적용하였다. 또한, 제시된 Large-scale 맥락 인식 시스템 평가를 위하여 11개의 heuristics를 정하고 수정된 휴리스틱 평가를 이용하였다. 평가 결과, 전문가들에 의한 가치있는 피드백을 얻었고 향후 개발 시에 적용할 수 있는 기틀을 마련하게 되었다.

추후 연구로는 평가 결과로 나타난 이슈들을 고려하여 더욱 강건한 맥락 인식 시스템을 개발할 것이다. 또한, 실제 환경에 적용 가능한 Large-scale 맥락 인식 시스템의 개발을 지속할 것이다. 맥락 인식 아키텍처와 관련된 전문가 평가에 의하여 툴킷(toolkit)의 평가와 같은 분야에도 제안된 휴리스틱 평가를 적용할 계획이다.

참고문헌

- [1] B. Schilit, N. Adams, and R. Want, "Context - Aware Computing Applications," Proc. Workshop Mobile Computer Systems and Applications, IEEE CS Press, Los Alamitos, Calif., 1994, pp. 85-90.
- [2] J. Nielsen, Enhancing the explanatory power of usability heuristics, In Proceedings of ACM CHI'94 Conference, Boston, MA, April 24-28, 1994, pp152-158.
- [3] Y.Oh, C.Shin, S.Jang and W.Woo, "ubi-UCAM 2.0: A Unified Context-aware Application Model for Ubiquitous Computing Environments", UbiCNS05, 2005.
- [4] www.skype.com
- [5] <http://www.nokia.com/A4145110>
- [6] R. Sharp, K. Rehman, "The 2005 UbiApp Workshop: What Makes Good Application-Led Research?," *IEEE Pervasive Computing*, vol. 04, no. 3, pp. 80-82, Jul-Sept, 2005.
- [7] J. Nielsen, Enhancing the explanatory power of usability heuristics, In Proceedings of ACM CHI'94 Conference, Boston, MA, April 24-28, 1994, pp152-158.
- [8] J. Nielsen. Heuristic Evaluation.
<http://www.useit.com/papers/heuristic/>. 2006
- [9] J. Nielsen. Ten Usability Heuristics.
http://www.useit.com/papers/heuristic/heuristic_list.html. 2006
- [10] Jennifer Mankoff, Anind Dey, Gary Hsieh, Julie Kientz, Scott Lederer, and Morgan Ames. Heuristic evaluation of ambient displays. In Proc. of ACM Conf. on Human Factors and Computing Systems (CHI 2003)