

상황인식 어플리케이션 개발의 컴포넌트 기반 접근법

문은영^o 윤회진 최병주
이화여자대학교 컴퓨터학과

siivern913@ewhain.net {hjyoon, bjchoi}@ewha.ac.kr

An Approach of Component Based Context Aware Application Development

Eunyoung Moon^o Hoijin Yoon, Byoungju Choi
Department of Computer Science and Engineering
Ewha Womans University, Seoul, Korea

요 약

유비쿼터스 컴퓨팅을 구현하는 어플리케이션의 대표적인 형태가 상황인식 어플리케이션이다. 기존의 상황인식 어플리케이션의 구현 형태는 IF-Then 구문을 사용하거나 규칙기반시스템을 사용하는 형태를 취하였다. 이는 상황정보가 어플리케이션 기능 구현 코드에 내재되어져 있으므로, 변화하는 상황정보에 적용하기 어려운 구조이다. 본 논문에서는 컴포넌트기반 소프트웨어 개발의 개념을 이용하여, 구현하고자 하는 상황인식 어플리케이션을 상황정보에 따른 소프트웨어 컴포넌트들의 조합으로 구현하는 연구를 제안한다. 이렇게 함으로써, 상황정보의 변화를 컴포넌트 단위로 적용할 수 있게 되며, 이를 통하여 상황정보 확장에 대처하는 확장성을 상황인식 어플리케이션에 부여할 수 있다. 본 논문에서는 대표적인 상황인식 어플리케이션 가운데 하나인 액티브 배지 시스템의 Call-forwarding 어플리케이션을 제안한 방법을 통해 구현하는 예제를 보이고, 이를 통해 얻는 특징인 확장성을 보인다.

1. 서 론

유비쿼터스 컴퓨팅의 대두로 다양한 소프트웨어에 대한 요구사항이 변하고 있다. 그 가운데 기술적 차원에서, 대규모의 유비쿼터스 소프트웨어가 동적으로 변화하는 환경에 적응하는 능력을 고려한 소프트웨어 설계와 구현에 대한 기술적인 문제들이 있다[1]. 즉, 유비쿼터스 컴퓨팅에서의 소프트웨어는 유기적으로 성장해야 한다[2]. 기존의 컴퓨팅 환경과 달리, 유비쿼터스 컴퓨팅이 적용되는 실세계는 동적으로 변화한다. 결국 유비쿼터스 컴퓨팅을 지원하는 소프트웨어는 동적으로 변화하는 상황에 자연스럽게 적응하면서 수행되어야 하며, 이 특성을 가능하게 하는 개발 기술이 요구되는 현실이다. 상황정보에 따라 소프트웨어의 행위를 변화시킬 능력이 요구되며, 이 능력을 상황인식(Context Awareness)라고 부른다. 유비쿼터스 컴퓨팅에서 상황정보는 더욱 자주 변화하게 되므로, 상황인식은 유비쿼터스 컴퓨팅의 중요한 조건이 된다.

따라서 유비쿼터스 소프트웨어에 대한 접근은 상황인식 소프트웨어로 시작할 수 있다. 그러나 기존의 상황인식 소프트웨어들은 처리해야 할 상황정보들이 상황인식 소프트웨어 내부에 녹아 들어있다. 보통 규칙기반시스템으로 상황인식소프트웨어를 작성하며, 규칙 내부에 상황정보가 포함되어져 있으며, 해당 행위들이 하나의 소프트웨어에 함께 표현되어져 있다.

본 논문에서는 이를 보완하기 위하여 컴포넌트 기반 접근방식을 소개한다. 액티브 배지 시스템[3]의 Call-Forwarding 어플리케이션을 모델로 한다. 2장에서는 본 논문에 대한 이해를 돕기 위한 관련 내용을 기술하고 3장에서는 상황정보 확장을 용이하게 하는 컴포넌트 기반 접근방안에 대하여 소개한다. 4장에서는 실제 구현 예제를 보이며 시스템의 확장성을 보장함을 보인다.

2. 관련 연구

2. 1 상황인식 컴퓨팅

유비쿼터스 컴퓨팅은 컴퓨팅, 통신, 접속 방식, 제공하는 콘텐츠 및 사람이 컴퓨터의 존재를 인지하지 않도록 조용히 처리하는 특성 (5C : Computing, Communication, Connectivity, Contents, Calm)을 이용하여 언제 어디서나 어떠한 형태의 네트워크에서도 모든 기기종 기기간의 연동을 통하여 다양한 서비스를 제공하는 것 (5 ANY : Anytime, Anywhere, Any Network, Any Device, Any Service)을 지향한다[4]. 이를 통하여 컴퓨터의 소지여부와 관계없이 컴퓨팅 서비스를 받을 수 있는 환경 제공이 가능하다. 이를 구현하기 위한 핵심요소로 상황(Context) 정보가 중요시 된다.

상황에 대한 정의는 다양하게 제시되어 왔으나, Schilit는 이에 대해 특히, 중요한 측면을 꼽아, "어디에 존재하고, 누구와 함께 있으며, 주변에 무슨 자원이 있

는지”로 정의하였다[4]. 이를 기반으로 하는 상황인식 컴퓨팅은 “사용자의 작업과 관련 있는 적절한 정보 또는 서비스를 사용자에게 제공하는 과정에서 ‘상황’을 사용하는 경우 이를 ‘상황인식 시스템’이라고 한다.”[5]로 정의된다. 이러한 상황의 종류는 다양하나, 일반적인 상황정보는 사용자 상황, 물리적 환경상황, 컴퓨팅 시스템 상황, 사용자-컴퓨터 상호작용, 기타 미 분류 상황으로 분류가 가능하다[6]. 따라서, 상황 정보에 대하여 컴퓨터의 접근성을 향상시킨다면, 결국 인간은 보다 유용한 컴퓨팅 서비스를 받게 될 것이다.

상황 정보 중에서도 가장 중요하게 인식되는 것은 위치정보이다. 이는 현재까지 연구된 유비쿼터스 응용시스템의 95%이상에서 현재 위치를 가장 비중 있게 사용하고 있는 점에서도 알 수 있다[7]. 상황인식 컴퓨팅에 대하여 진행된 연구로는 액티브 배지 시스템[3], e-class, DUMBO, Conference assistant등이 있다. 이 시스템들이 기반한 개념은 상황정보를 적절히 취합하여 저장하고 사용자가 이를 사용할 수 있도록 서비스를 하는 것이다.

본 논문에서는 진행된 연구들 중에서도 액티브 배지 시스템의 Call-Forwarding 어플리케이션을 모델로 하여, 확장성과 적응성을 보장하는 접근 방식을 설계, 분석하고자 한다.

2.2 액티브 배지 시스템

액티브 배지 시스템[3]은 Olivetti 연구소에서 진행한 연구이다. 이는 병원과 같은 곳에서 응급상황 발생 시 필요한 인력의 위치를 실시간으로 파악하여 그 장소로 호출하기 위한 필요에서 만들어진 시스템이다.

이 시스템에서는 필요한 멤버의 위치를 알기 위하여, ‘액티브 배지’를 이용한다. 이는 고유의 적외선 코드를 일정 시간단위로 방출하며, 주기적 신호는 호스트 빌딩 주위에 장착된 네트워크 센서에 의하여 탐지된다. 네트워크와 연결된 마스터 스테이션은 센서를 풀링하며, 데이터를 프로세싱하여 사용자가 사용할 수 있는 형태로 출력한다.

이 시스템에서 적외선 신호가 쓰이는 것은, 라디오 전파가 벽과 같은 파티션을 통과하는 것과 달리, 이는 벽을 통과하지 못하기 때문이다. 또한, ‘액티브 배지’는 멤버의 상의에 착용되며 허리 부위에 착용 시 책상에 앉을 경우 탐지가 쉽지 않으므로 가슴 부위에 착용된다. 이를 감지하는 센서는 벽의 윗부분 또는 사무실 천정에 장착되며 복도의 출입구와 그 외 공공장소에도 설치된

다.

사용자에게 보이는 형식은 3가지 요소를 포함한다. 첫 번째 항목은 멤버의 이름이고, 두 번째 항목은 멤버의 위치이며, 세 번째 항목은 멤버가 그 위치에 있을 확률이다. 확률이 100%로 표시되면 그 위치에 현재 있다는 것이고, 다른 곳으로 이동 중이거나 5분 동안 감지되지 않는 경우 수치는 100미만으로 떨어진다. 그리고 이 항목에는 최근 탐지된 시간과 그 위치를 보여준다. 만약 탐지된 시간이 24시간이 지났다면, 가장 최근 날짜를 보여주고, 일주일 이상 탐지되지 않은 경우는 ‘AWAY’로 표시한다. 다음 그림 1은 디스플레이 화면이며, 이 화면은 일정 시간 단위로 업데이트 된다.

ORL/STL Active Badge Project					
Name	Location	Prob	Name	Location	Prob
P Ainsworth	X343 Acce	100%	J Martin	X310 Mc Rm	100%
T Blackie	X222 Dvl Rm	85%	O Mason	X307 Lab	77%
M Chopping	X410 R302	TJJE	D Milway	X307 Drll	AWAY
D Clarke	X316 R321	10:30	B Miners	X202 Dvl Rm	10:40
V Falcao	X216 R435	AWAY	P Mittal	X213 Pw	11:20
D Garnett	X232 R310	100%	J Panser	X399 Lib	100%
J Gibson	X0 Res	AWAY	B Robertson	X307 Lab	100%
D Greaves	X304 F3	MON	C Turner	X307 Lab	MON
A Hopper	X434 AH	100%	R Want	X309 Meet. Rm	77%
A Jackson	X308 AJ	90%	M Wilkes	X300 Mw	100%
A Jones	X210 Coffee	100%	I Wilson	X307 Lab	100%
T King	X309 Meet. Rm	11:20	S Viray	X204 Svt	11:20
D Loupis	X304 R311	100%	K Zielinski	X402 Coffee	100%

12:00 1st January 1990

그림 1. 액티브 배지 시스템의 서버 화면

이 시스템에서 수행되는 명령에는 Find, Look, History 등이 있다. 각 명령어의 수행 기능은 다음과 같다.

Find(name) : 멤버의 현재 위치를 알려준다.

Look(location) : 특정 장소에 있는 멤버들을 탐지한다.

History(name) : 한 시간 동안 특정 멤버의 위치 이동을 리스트 형식으로 보여준다.

따라서, 사용자는 이 화면을 보고 필요에 따라 멤버의 위치를 실시간으로 확인하고 전화를 전달할 수 있다.

3. 컴포넌트 기반 상황인식 구현

Call-forwarding 어플리케이션은 사용자인 멤버가 적외선 방출 배지를 착용하고 이동하면, 신호 수신기가 적외선 신호(ID)를 읽어 사용자의 현재 위치를 상황정보로 하여 시스템에 업데이트 하며, 멤버에게 전화가 걸려온 경우, 시스템의 상황정보를 참조하여 멤버의 현재 위치와 가까운 곳으로 전화를 포워딩하는 응용 프로그램이다.

3.1 상황인식 어플리케이션 개발의 접근법 소개

우리는 위의 Call-forwarding 어플리케이션을 모델로 하여 기본 기능을 수행하는 프로토타입을 구현하며, 핵심 요소인 상황(Context)을 기준으로 기능을 분리하여 각 기능들을 모듈로 취급한다. 따라서, 구현할 응용은 인터페이스 모듈과 기능을 수행하는 모듈로 크게 구분할 수 있으며, 이는 상황의 조합인 조건(Condition)에 따라 좀 더 세분화된다.

3.2 상황인식 어플리케이션 개발의 분석

context type은 상황정보의 종류를 표현한다. 예를 들어 위치정보를 상황정보로 사용한다면, 상황정보 종류는 '위치'가 될 수 있다. 유비쿼터스에서의 상황정보의 종류는 행위(Activity), 신원(Identity), 위치(Location), 그리고 시간(Time)으로 나뉘어진다[8]. 본 논문에서 보일 어플리케이션에서 다루는 상황 타입은 멤버의 이름과 위치 정보이며, 다음과 같이 분류한다.

Context-type 1 : Name

Context-type 2 : Location

각 상황 타입은 구체화된 상황 정보로 사용되므로 다음과 같이 가정한다. 상황 타입1에 속하는 멤버의 이름으로 Tom, Jane, Kate, 상황 타입 2에 속하는 위치로는 Area A, Area B, Area C로 가정한다. 또한, 각 멤버는 세 위치 중에서 적어도 한 곳에 위치하며, 액티브 배지 시스템과 같이 센서는 두 명 이상이 같은 장소에 머물더라도 고유 ID 신호를 감지한다.

위의 요소들을 C1=(Name, Tom), C2=(Name, Jane), C3=(Name, Kate), C4=(Location, Area A), C5=(Location, Area B), C6=(Location, Area C)로 표기할 수 있다.

이 어플리케이션은 문제 상황인, 전화가 걸려온 경우 실행되므로 해당 멤버에게 전달해야 하는 전화를 어떻게 포워딩할 것인가가 분석의 대상이 되는데, 위와 같이 구체화된 상황 정보를 적절히 조합하여 기능을 수행할 수 있다. 먼저, Call-forwarding 어플리케이션에서 수행되어야 할 기능을 크게 두 가지로 나눌 수 있다. 첫 번째 기능은 "전화를 받는다."(F0)이고, 두 번째 기능은 "걸려온 전화를 해당 멤버에게 전달한다."이다. 두 번째 기능은 위의 C1~C6의 상황 정보의 조합인 조건(Condition)에 따라 실행된다. 각 기능은 곧 하나의 컴포넌트를 구성하여 컴포넌트 기반의 구조를 갖게 할 것이다.

3.3. 상황인식 어플리케이션 개발의 설계

앞의 분석을 토대로 우리는 상황 정보를 중심으로 기능을 세분화 할 수 있고, 세분화된 기능들은 어플리케이션을 구성하는 모듈이 된다. 이 모듈들로 구성된 하나의 응용 프로그램이 완성된다. 각 모듈은 조건(Condition)에 따라 상세화 되므로, 두 번째 기능을 다음과 같이 구체화하기로 한다.

전화를 멤버에게 전달하는 두 번째 기능을 상황 타입 2에 해당하는 '위치'에 따라 분류할 수 있다. 즉, 걸려온 전화를 Area A, Area B, Area C로 전달하는 3가지로 나눌 수 있다. 예를 들어, Tom에게 온 전화이며, Tom이 Area A에 위치한 경우 전화를 Area A로 전달하는 것이다. 이것은 다른 멤버인 Jane이나 Kate의 경우 역시 마찬가지로 적용된다.

따라서, F1은 "전화를 Area A로 전달한다."로, F2는 "전화를 Area B로 전달한다."로, F3은 "전화를 Area C로 전달한다."로 정의할 수 있다. F1, F2, F3은 어플리케이션의 각 독립적인 모듈에 해당되며 이를 component1, component2, component3으로 명명할 수 있다.

$$F1=(F(C1 \wedge C4) \vee (C2 \wedge C4) \vee (C3 \wedge C4))$$

$$F2=(F(C1 \wedge C5) \vee (C2 \wedge C5) \vee (C3 \wedge C5))$$

$$F3=(F(C1 \wedge C6) \vee (C2 \wedge C6) \vee (C3 \wedge C6))$$

그림 2는 컴포넌트 기반 어플리케이션의 구조도를 보여준다.

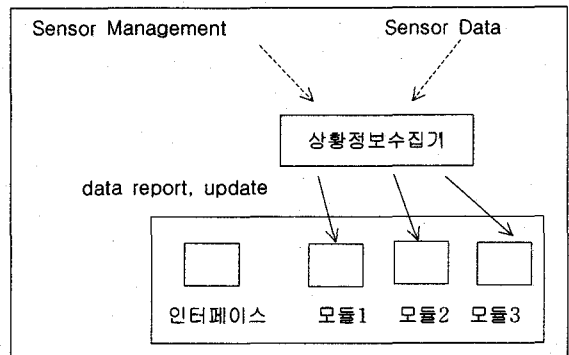


그림 2. 상황정보 기반 컴포넌트 어플리케이션 구조도

구조도에서 볼 수 있듯이, 어플리케이션은 사용자와 접하는 인터페이스와 F1 기능을 수행하는 모듈1, F2를 수행하는 모듈2, F3을 수행하는 모듈3으로 구성되고, 센싱된 데이터는 수집기로부터 적정 시간마다 보고되는 형태이다.

4. 예 제

4.1 Call-forwarding 어플리케이션

멤버에 속하는 Tom, Jane, Kate의 이동 경로는 8000ms마다 자동 업데이트 되어 서버에 들어와 화면에 출력된다. 전화가 걸려온 경우, 사용자는 해당 멤버의 이름을 선택한다. 그러면, 화면에 표시된 멤버가 위치한 장소의 전화가 울린다. 해당 멤버는 전화를 받는다. 따라서, Call forwarding의 기능이 수행되었다.

4.2 컴포넌트 기반 구현

실제로 구현할 어플리케이션의 인터페이스는 그림 3과 같다. 사용자는 상황정보타입 1에 해당하는 name을 보여주는 리스트에서 Tom, Jane, Kate 세 사람 중 선택할 수 있다. Jane에게 전화가 온 경우, 리스트에서 Jane을 마우스로 클릭한다. 리스트 밑의 Tom, Jane, Kate의 이동경로가 8000ms마다 자동 업데이트 되어 들어오며, 이는 상황정보타입 2에 해당하는 위치를 보인다. 만약 Jane이 Area B에 현재 있다면 $(C2 \wedge C5)$ 이므로 F2가 실행된다. 그리고, TextArea 영역에는 Jane이 있는 Area B의 전화로 포워딩된다는 메시지 - "Call-Forwarding to : Jane in Area B"를 띄운다.

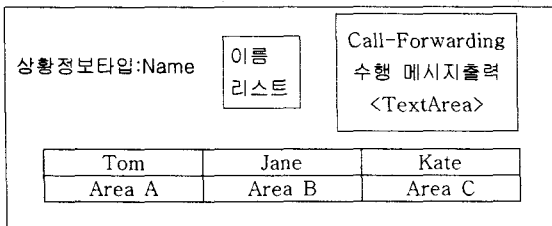


그림 3. 상황정보 기반 컴포넌트 어플리케이션의 구성화면

코드에서 사용자가 선택한 멤버의 위치 확인은 함수 Rule을 통해서 구현된다. 즉, 상황정보타입1에 속하는 C1, C2, C3중 어느 것이 선택되었는가를 확인한다. 이 상황정보가 확인되면 멤버의 위치를 확인한다. 이것은 Module1, Module2, Module3 함수를 통해 이루어진다. 각각은 현재의 위치에 따라서, Area A, Area B, Area C의 값을 반환한다. 이로써, 상황정보타입2인 위치정보 C4, C5, C6도 확인된다. 위에서 살펴본 예인, Jane이 Area B에 있는 경우 Rule함수에서 C2(Jane)가 확인되

고 Module2함수에서 $C5(Area B)$ 값을 가져온다. 그리고, 이에 해당하는 F2인 Area B로의 포워딩 메시지를 띄우는 함수는 이벤트 함수로 구현한다.

4.3 확장

이렇게 하여 모듈1,모듈2,모듈3에 해당하는 기능들을 컴포넌트 기반으로 구현하였다. 이 시점에서 사용자의 요구사항이 더 추가된다고 가정해 보자. 가령, Area D, Area E로의 이동이 가능해져서 이 위치에 멤버가 있을 경우 전화를 전달할 수 있어야 한다고 하면, Area D, Area E 역시 모듈로 취급하여 구현이 가능하다. 각 멤버가 이 장소로 이동하면, 그 장소로 전화를 전달하므로 상황정보-C7=(Location, Area D), C8=(Location, Area E)와 다음의 모듈이 추가된다.
 $F4=(F(C1 \wedge C7) \vee (C2 \wedge C7) \vee (C3 \wedge C7))$
 $F5=(F(C1 \wedge C8) \vee (C2 \wedge C8) \vee (C3 \wedge C8))$
 따라서, 이에 따라 구조도 역시 그림 4처럼 수정된다.

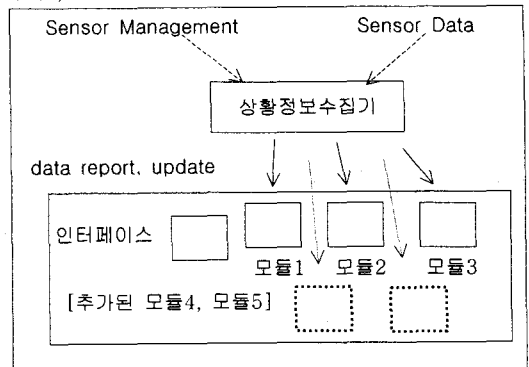


그림 4. 수정된 어플리케이션의 구조도

그림 4에서 점선으로 표현된 부분이 추가된 요구사항인 모듈4,모듈5이다. 또한, 위에서 어떤 모듈을 제거하는 것도 용이함을 알 수 있으며, 모듈의 재사용 역시 가능하므로 이 시스템은 더욱 효과적이다. 모듈 자체로 기능을 수행하므로 모듈 안에 속한 함수의 수정과 추가가 가능하다.

위의 예에서 Jane이 Area B에 있어 그 곳에 연결된 전화로 전달과 동시에 Area B의 라디오를 켜는 기능을 사용자로부터 추가로 의뢰받았다고 하자. 그러면, $(C2 \wedge C5)$ 를 수행하는 모듈2내에서 새로운 기능인 라디오 켜는 사항을 추가로 함수로 구현하여 새 버전인 F2' 모듈로 완성하여 사용자에게 신속하게 전달할 수 있다.

이처럼 컴포넌트 기반 접근 방법은 시스템을 자체적으로 변화시키지 않고도 구현부의 한 모듈을 다른 모듈

로도 대체가 가능하며 추가, 삭제에 있어서도 편리성을 제공한다. 또한, 각 모듈은 블랙 박스 유닛으로 취급되므로, 모듈을 교체, 추가 시 각 모듈에 대한 깊은 지식 없이도 확장이 가능하다. 그림 5는 실행화면이다.

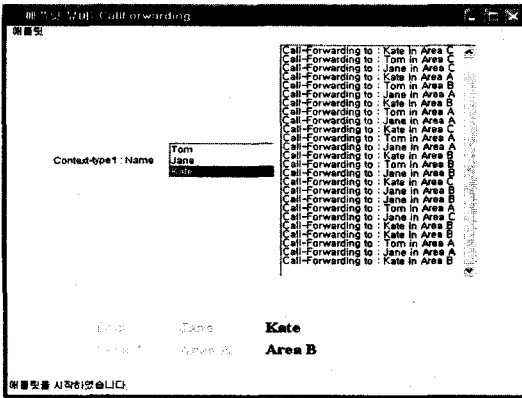


그림 5. 상황정보 기반 컴포넌트 어플리케이션의 실행화면

또한, Area D로의 이동이 가능해져서 C7=(Location, Area D)가 추가된 경우로의 확장을 구현하는 경우, Area D값을 반환하는 함수 모듈 4를 추가함으로써, 보장된다. 그림 6은 F4를 수행하는 모듈4를 추가한 어플리케이션의 실행화면이다.

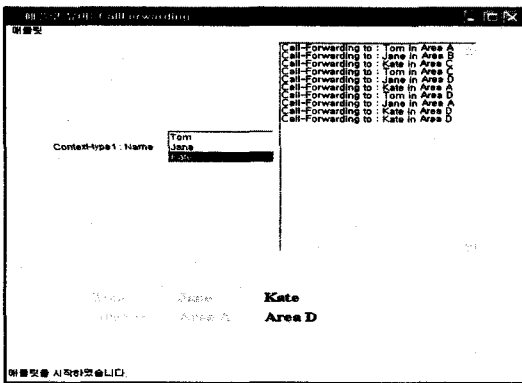


그림 6. 모듈4가 추가된 어플리케이션의 실행화면

이와 같은 특징은 규칙기반 시스템에서는 쉽게 얻을 수 없는 강력한 이점으로, 확장성과 적응성을 보장한다.

5. 결론

본 논문은 기존의 상황인식 어플리케이션 구현 모델이 갖는 상황정보의 코드내의 내재성을 극복하기 위하여, 컴포넌트 기반 소프트웨어 개발을 적용하였다. 분석과 설계를 통하여 어떻게 상황정보 기반 컴포넌트들을

정의하는지를 제안하였으며, 이 분석 및 설계 방안을 기존의 대표적인 상황인식 어플리케이션인 액티브 배지 시스템의 Call-forwarding 구현에 적용하는 예제를 보였다. 분석 및 설계의 결과 정의된 컴포넌트를 Java를 사용하여 구현하였고, 이후 상황정보가 확장되는 경우에 추가되는 상황정보를 인지하는 컴포넌트를 이전 어플리케이션에 덧붙임으로써 확장을 지원하는 예제 또한 보였다.

본 연구가 확장성 및 코드의 재사용성 측면에서 시작되었으나, 제안한 방안이 기타 다른 요소에서의 효율성에는 어떤 영향을 미치는지에 대한 분석이 요구된다. 이를 위한 실험을 향후 연구 과제로 고려하고 있으며, 나아가 제안한 분석 및 설계를 도와주는 도구 구현 또한 고려하고 있다.

6. 참고문헌

- [1] Kalle Lyytinen and Youngjin Yoo, "Issues and Challenges in Ubiquitous Computing," Communications of the ACM, Vo.45, No.12, pp.63-65, Dec. 2002
- [2] F. Brooks. Mythical Man-Month. Addison Wesley, 20th anniversary edition, 1995
- [3] Roy Want, Andy Hopper, Veronica Falcao and Jonathan Gibbons, "The Active Badge Location System", ACM Transactions on Information Systems 10(1):00.91-102, Jan. 1992
- [4] Shilit.B, Adams.N, Want.R, Context-Aware Computing Applications. Proceedings of the 1st International Workshop on Mobile Computing Systems and Applicatoins, pp.85-90,1994
- [5] Dey, A.K. Understanding and Using Context. Personal and Ubiquitous Computing Journal. Vol.5(1).pp4-7,2001
- [6] G.Chen nd D.Kotz, "A Survey of Context-Aware Mobile Computing Research", Dartmouth Computer Science Tech. Report TR2000-381,2000
- [7] 김재호, 신경철, 상황인식 서비스 기술 연구 동향, 주간기술동향 1178, 2004
- [8] Anind K. Dey and Gregory D. Abowd, "The Context Toolkit: Aiding the Development of Context-Aware Applications," in Proceedings of Workshop on Software Engineering for Wearable and Pervasive Computing, 2000.