

# 유비쿼터스 컴퓨팅 시스템의 실용적 품질 평가 모델

## (UCQM: A Quality Model for Practical Evaluation of Ubiquitous Computing Systems)

오 상 현<sup>†</sup>   김수동<sup>\*\*</sup>   류성열<sup>\*\*</sup>  
(Sang Hun Oh)   (Soo Dong Kim)   (Sung Yul Rhew)

**요약** 유비쿼터스 컴퓨팅 시스템(Ubiquitous Computing System, UCS)은 언제 어디서나 장소에 구애를 받지 않고 사용자가 컴퓨터 네트워크를 이용하여 서비스를 얻을 수 있는 시스템이다. 유비쿼터스 컴퓨팅 시스템은 개인화된 시스템으로써, 다른 시스템과 상호운영이 되는 시스템이다. 이러한 유비쿼터스 컴퓨팅 시스템은 우리 생활에 생산과 소비, 정치, 경제, 사회, 문화, 전 분야에 걸쳐 많은 변화를 가져올 것이다. 따라서 본 논문에서는 이러한 유비쿼터스 컴퓨팅 시스템을 평가하기 위하여 ISO/IEC 9126을 기반으로 체계적인 품질모델을 제안한다. 그러므로, 유비쿼터스 컴퓨팅 시스템의 품질이 좋아야 사용자에게 제공하는 서비스가 좋기 때문이다. 또한 본 논문에서는 유비쿼터스 컴퓨팅 시스템의 중요한 특징들을 식별하고, 식별된 특징을 기반으로 하여 품질속성(Quality Attribute)을 도출한다. 이렇게 도출된 품질속성을 기반으로 하여 메트릭을 정의하고, 유비쿼터스 컴퓨팅 품질모델(Ubiquitous Computing Quality Model, UCQM)을 모델링 하여 유비쿼터스 컴퓨팅 시스템의 전반적인 환경과 중요한 특징을 평가한다.

**키워드** : 유비쿼터스 컴퓨팅, 품질모델, 품질속성, 메트릭, ISO/IEC 9126

**Abstract** Ubiquitous Computing System (UCS) is a system where the user can get information through computer network anytime and anywhere regardless of the places. Since UCS is a personalized system, it should interact with other systems. UCS will bring a remarkable change in production, consumption, politics, economy, community, culture, and other areas related to our daily life. That is, a high-quality UCS results in high-quality services provided to users. Hence, this paper proposes a systematic quality model based on ISO/IEC 9126 in order to evaluate the ubiquitous computing system, based on ISO/IEC 9126. And, we identify key characteristics of UCS and derive the set of quality attributes based on identified characteristics. We define metrics for each quality attribute and Ubiquitous Computing Quality Model (UCQM) so that we can evaluate overall environment and important characteristics of UCS.

**Key words** : Ubiquitous Computing, Quality Model, Quality Attribute, Metric, ISO/IEC 9126

### 1. 서론

유비쿼터스란, 사용자가 컴퓨터나 네트워크를 의식하지 않는 상태에서 장소에 구애 받지 않고, 자유롭게 네트워크에 접속 할 수 있는 환경을 말한다[1]. 유비쿼터

스의 기술은 변화하는 시대에 새로운 패러다임을 가져오게 될 것이다. 하지만, 유비쿼터스 컴퓨팅 시스템은 상황에 따라 제공되는 서비스가 변한다[2]. 따라서, 사용자가 누구이고 또 어디서 어떻게 사용하느냐에 따라 다른 서비스가 제공된다.

유비쿼터스 컴퓨팅 시스템은 기본적으로 무선 네트워크환경이 기본이 되어야 한다. RFID(Radio Frequency Identification)가 객체에 붙어 있는 태그가 리더의 발사 전파를 일시적으로 흡수를 해서, 태그의 고유 정보를 실어 반사하는 원리로 작동된다[3]. 따라서 사용자가 인식을 못하는 사이에도 유비쿼터스 컴퓨팅 시스템은 작동을 하고 있는 것이다. 그러므로, 모든 환경과 사물이 지

· 본 연구는 한국과학재단 특정기초연구(R01-2005-000-11215-0)지원으로 수행되었음

† 학생회원 : 송실대학교 컴퓨터학과  
shoh@otlab.ssu.ac.kr

\*\* 종신회원 : 송실대학교 컴퓨터학과 교수  
sdkim@ssu.ac.kr  
syrehew@ssu.ac.kr

논문접수 : 2006년 3월 27일  
심사완료 : 2007년 2월 6일

능화되어 서로간 연결, 대화가 가능한 것이다.

본 논문에서는 ISO/IEC9126을 기반으로 하여 유비쿼터스 컴퓨팅 시스템의 특징을 추출하여, 추출된 특징을 기반으로 품질평가를 위한 품질속성을 도출한다. 이러한 품질속성을 기반으로 하여 매트릭을 정의한다. ISO/IEC9126의 표준 구성은 주특징, 부특징 및 매트릭으로 구성되어 있다. 그리고 소프트웨어의 품질을 평가하기 위한 6개의 특징과 27개의 부특징과 내, 외부 및 사용자 매트릭을 정의하고 있다[4,5]. 그러나, 제품평가를 위한 표준인 ISO/IEC9126으로는 유비쿼터스 컴퓨팅 시스템을 평가하기에 미흡하다. 그 이유는, ISO/IEC9126은 완성된 단일 어플리케이션을 위한 표준이기 때문이다. 하지만 유비쿼터스 컴퓨팅 시스템은 단일 어플리케이션이 아닌 또 다른 시스템과 함께 상호 운영되는 시스템이기 때문이다. 따라서 본 논문에서는 표준 품질모델인 ISO/IEC9126을 기반으로 하여 유비쿼터스 컴퓨팅 시스템을 평가하기 위한 새로운 품질모델을 제안한다. 본 논문은 품질모델을 도출하기 위한 과정, 제안된 품질모델, 품질모델 내의 매트릭을 적용하기 위한 지침을 포함한다.

2장에서는 본 논문과 관련된 관련연구를, 3장에서는 기반연구, 그리고 4장에서는 유비쿼터스 컴퓨팅 시스템의 특징을 추출하고, 5장에서는 이렇게 추출된 특징을 기반으로 하여 품질특성을 유도하고, 6장에서는 이렇게 유도된 주특징으로부터 부특징을 도출한다. 7장에서는 각각의 품질속성에 대해 매트릭으로 정의를 하고, 이를 기반으로 사례연구를 작성하고 결론을 내린다.

## 2. 관련연구

### 2.1 Framework for middleware in Ubiquitous Computing Systems

본 연구에서는 유비쿼터스 컴퓨팅 시스템에 미들웨어가 필요한 이유에 대해서 몇 가지 제시하였다[6]. 기능성들이 상호작용하고 있는 독립형의 소프트웨어 구성요소에는 아키텍처 방법이 필요하다. 따라서 그 안에 설정되어 있는 미들웨어가 필요하다. 그리고 상호작용을 확립하고 유지하기 위해 기계장치들을 정의해야 하고, 인터페이스 규정을 제공하기 위하여 서비스를 제공해야 하는데, 이 서비스가 어떤 종류의 상호작용과 관련이 있는지를 미들웨어에서 정의해야 한다. 또한 서비스를 위해 결합방지 능력과 구성관리를 해야 하고, 사용자의 테스트를 위해 사용자 행동을 추론해야 하며, 상황정보를 상세히 알고 있어야 한다. 통합과 상호작용을 위하여 적당한 서비스를 선택해야 하고, 지원해야 한다. 또한 다른 서비스의 사이에서 교환되었던 정보를 전환해야 하는데 이때 미들웨어가 필요하다. 그리고 컴퓨팅의 동적인 성질 때문에 서비스는 재구성에 의해 이 변화들을

반영해야 하고, 미들웨어는 공급관리와 서비스의 보안, 데이터 프로세스를 제공해야 한다. 마지막으로 미들웨어의 서비스는 최종사용자 장치의 이종성을 극복하고 사용자 인터페이스를 통합한다.

### 2.2 System Software for Ubiquitous Computing

본 연구에서는 유비쿼터스 컴퓨팅 시스템에 두 가지의 중요한 특징을 제시하였다[7]. 첫 번째는 물리적인 통합이고, 다른 하나는 자연스러운 상호운영을 말한다. 이러한 두 가지의 특징이 유비쿼터스 컴퓨팅 시스템을 디자인 할 때, 그리고 미래의 방향에 어떠한 영향을 줄 것인가를 본 연구에서는 말하고 있다. 유비쿼터스 컴퓨팅 시스템 디자이너들은 다양한 물리적인 물체나 공간에서 장치들을 끼워 넣을 수 있도록 디자인을 한다. 모바일과 분산 컴퓨팅 연구는 이미 사용자의 요구사항을 만족시켰다. 그러나 질(Quality)적인 면에서 요구사항과 성취는 다른 의미를 가진다. 본 연구에서는 오늘날의 유비쿼터스 시스템과 소프트웨어 구조에 집중하고 논의한다. 또한 앞에서 언급한 두 가지의 중요한 특징을 본 연구에서 분석 하는데 이러한 유비쿼터스 컴퓨팅 시스템의 시각을 마크와이어 및 다른 연구자가 직접적으로 논의하고 설명을 했기 때문이다. 인간과 지역 그리고 문화적 차이의 의미는 유비쿼터스 컴퓨팅 시스템이 해결 해야 할 기본적인 환경(집, 학교, 공항라운지 등)이다.

### 2.3 A Tested for Evaluating Human Interaction with Ubiquitous Computing Environments

유비쿼터스 컴퓨팅 시스템 환경의 핵심은 사용자의 상황에 맞게 시스템의 기능을 변경시킬 수 있는 적응성이 있는 소프트웨어 시스템을 말한다[8]. 이 상황은 작업이 수행되는 물리적인 환경과 많은 밀접한 연관을 가지고 있다. 이러한 적응성이 있는 시스템의 효율성은 사람이 특정한 물리적인 환경과 사회적인 상황으로 표현되는 상황 내에서 제공되는 시스템의 기능(행위)을 인지하는데 좌우 된다. 그리고, 본 연구에서는 유비쿼터스 컴퓨팅 시뮬레이터인 TATUS를 소개한다. 이 TATUS는 가격과 위치추적 이슈를 극복할 수 있는 유비쿼터스 컴퓨팅 시뮬레이터이다. 이 시뮬레이터는 무선 통신을 위한 시험대와 함께 인터페이스를 가진다.

### 2.4 Measuring Software Product Quality: A Survey of ISO/IEC 9126

IT에 빠른 성장과 PC들의 확산은 매우 많은 소프트웨어 제품을 선보였다[9]. 이로 인해, 사용자 만족은 품질관리에 대한 중대한 결과를 가져왔다. 이러한 결과는 이익과 판매성장에 명백하게 영향을 주었다. 따라서 품질관리와 관련된 관계자들은 계속적으로 모니터를 하고 이를 기반으로 소프트웨어 제품의 품질을 높이는데 중점을 두었다. 본 연구에서는 ISO/IEC 9126의 분류가

정확하게 잘 되었는지, 소프트웨어 품질을 평가하는데 신뢰할 수 있는지를 연구하고, 각 부특징에 대한 규모나 중요성을 측정하였다. 이 연구에서는 측정 결과들을 설명하는데 연구의 가치를 두었다. 또한 이 결과들을 소프트웨어 제품을 평가하는데 사용하였으며, 어느 특정한 제품을 평가하는데 사용되는 것이 아니라, 보다 폭넓게 사용되고 소프트웨어 제품을 수정 할 수 있도록 하였다.

**2.5 Application of Hazard Analysis to Software Quality Modeling**

본 연구에서 말하는 품질이란 소프트웨어와 정보 시스템 발전의 기본적인 개념이라 정의하였다[10]. 또 다양한 품질모델들은 소프트웨어와 정보시스템들의 품질을 측정하고 예측한다. 그리고 본 연구에서는 조직적인 방법을 이용하여 정보시스템들을 위한 품질모델을 정의하였다. 도표의 표기법은 품질모델들을 표현하기 위해 사용된다. 위험분석과 안전에 관련된 기술들은 아키텍처상의 디자인들로부터 품질모델을 이끌어 냈다.

**3. 기반연구**

**3.1 Mark Weiser의 연구**

사람들은 지난 수십 년 동안 더 많은 기술과 메모리 그리고 색깔 장식등, 컴퓨터 환경에 많은 것을 원했고, 요구해 왔다. 하지만, 세계의 정보와 기술들은 항상 올바르게 진화가 되지는 않았다[1]. 사용자에 의해서 정보와 기술들은 그 한계에 부딪힌다. 컴퓨터는 네트워크, 상호작용, 프라이버시와 계산의 방법들을 포함하는 컴퓨터 과학의 모든 분야에 영향을 미친다. 이제 더 이상 컴퓨터는 가상 현실이 아니다.

유비쿼터스 컴퓨팅 시스템은 기존의 컴퓨터나 PDA(Personal Digital Assistant)와는 다르게 네트워크를 통해 다른 시스템의 장치와 서로 상호작용을 한다. 다시 말해서, 사용자 목소리에 응답하고 기존 조수역할을 했

던 컴퓨터와는 달리 사용자는 시스템의 존재여부를 잊고 생활을 하게 된다. 따라서, 무선 네트워크를 통해 하부 구조에 집중을 하게 된다. 이렇듯 수많은 컴퓨터를 대상으로 하기 때문에 무선 대역폭의 필요성은 막대하다.

미래의 유비쿼터스 컴퓨팅 시스템은 우리 생활 전반에 걸쳐 존재할 것이며, 사람들은 이러한 유비쿼터스 컴퓨팅 시스템 존재 여부에 대해서는 모르며 기존의 컴퓨터 세계와는 전혀 다른 세계에서 우리들은 살아갈 것이다.

**3.2 Friedemann Mattern의 연구**

고도의 기술적인 발전은 풍부하며 매우 작고, 매우 싼 마이크로 프로세서를 개발하였다[2]. 유비쿼터스 컴퓨팅 시스템은 감지 장치들을 갖추고 있으며, 무선 통신의 능력을 가진다. 또한 이 시스템은 우리 생활 전반에 걸쳐 영향을 미칠 것이다.

이러한 유비쿼터스 컴퓨팅 시스템의 기술적인 발전 뒤에는 마이크로 일렉트로닉스가 있었다. 특히 시스템과 스마트공학의 분야에서 최근의 업적은 대단하다. 또한 재료 과학의 분야에서의 발전은 미래의 컴퓨터의 형태를 완전히 바꿀 것이며, 매우 유연하고 얇고 가벼운 형태로 만들어질 것이다. 따라서 이전 펜과 종이에도 컴퓨터가 존재하게 되며, 이러한 유비쿼터스 컴퓨팅 시스템은 환경과 장소에 구애를 받지 않고 전세계의 강력한 새로운 산업으로 등장 할 것이다.

이렇게 우리 생활에 편리함을 가져다 줄 유비쿼터스 컴퓨팅 시스템의 기술은 개인이나 사회뿐만 아니라 우리의 생활 전반에 어떠한 영향을 미칠지는 아직은 미지수다. 곧, 우리의 생활 속에서 존재하게 될 유비쿼터스 컴퓨팅 시스템이 어떠한 형태로 자리를 잡을지는 시간만이 해결해 줄 것이다.

**3.3 ISO/IEC 9126 품질모델**

ISO/IEC 9126은 단일 소프트웨어를 평가하기 위한 표준 품질모델이다. 품질모델은 주특징, 부특징 및 메트



그림 1 ISO/IEC9126 소프트웨어 품질모델

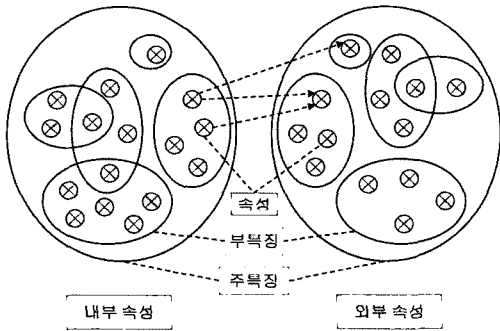


그림 2 내부속성과 외부속성간의 관계

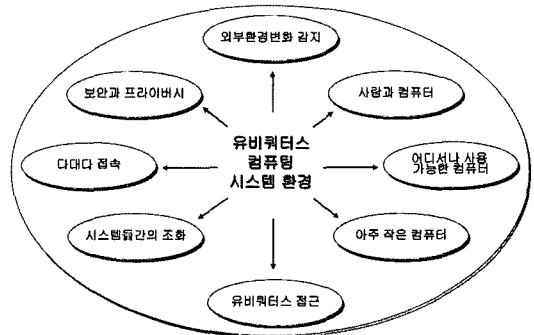


그림 3 유비쿼터스 컴퓨팅 시스템 환경내의 특징

릭으로 구성된다. 또한 ISO/IEC 9126은 소프트웨어의 품질을 평가하기 위한 6개의 주특징과 27개의 부특징 그리고 내, 외부 및 사용 중 메트릭을 정의한다[4,5].

하나의 주특징이 여러 개의 속성에 의해 영향을 받는 것과 마찬가지로 하나의 속성이 여러 개의 특징에 영향을 줄 수 있다.

**내부 메트릭(내부속성 측정 시 사용):** 설계나 코딩 도중에 실행할 수 없는 소프트웨어 제품에 적용할 수 있다. 내부 메트릭은 사용자, 평가자, 시험자 및 개발자가 소프트웨어 제품 품질을 평가할 수 있도록 도와주며, 그 소프트웨어 제품을 만들기 전에 미리 품질에 문제점들을 지적해 준다.

**외부 메트릭(외부속성 측정 시 사용):** 시스템 측정치에서 추출되는 소프트웨어 제품의 측정치를 위해 사용된다. 외부 메트릭은 사용자, 평가자, 시험자 및 개발자가 시험 수행이나 운영 중에 소프트웨어 제품 품질을 평가할 수 있도록 도와준다. 특정한 조직이나 기술적 환경에서 제품의 사용, 개발 및 관리와 관련된 기업 목적을 토대로 메트릭을 사용하여 평가 하는 것이 외부 메트릭이다.

**사용 품질 메트릭:** 제품이 사용될 경우에 정해진 목표를 달성하기 위하여 효율성, 생산성 및 만족도 측면에서 특정 사용자의 요구를 충족하는가를 측정한다. 사용 품질을 평가하는 것은 특정 사용자 작업 시나리오 내용에서 소프트웨어 품질을 확인하는 것이다. 소프트웨어 자체의 성질보다는 소프트웨어를 사용한 결과의 관점에서 측정한다.

#### 4. 유비쿼터스 컴퓨팅 시스템의 특징

본 장에서는 유비쿼터스 컴퓨팅 시스템의 중요한 특징을 식별하고, 이러한 특징을 기반으로 하여 유비쿼터스 컴퓨팅 시스템을 평가 할 수 있는 품질모델을 추출하기 위한 기반을 마련한다. 그림 3은 유비쿼터스 컴퓨팅 시스템 환경이 구축되기 위해서 필요한 특징과 기능

들을 보여주고 있다. 유비쿼터스 컴퓨팅 시스템의 중요한 특징은 다음과 같이 8가지이다.

##### 4.1 외부환경변화 감지

유비쿼터스 컴퓨팅 시스템의 환경은 개인화된 서비스를 지능적으로 제공하기 위해서 다양한 형태의 정보가 필수적이다[1,2]. 사용자의 상황(시간, 날씨, 온도, 명암, 장치, ID, 장소 등)에 따라 각 개인의 서비스가 변하고 달라야 한다. 이렇게 변화가 많은 환경에서 사용자의 요구사항과 사용자가 원하는 정보가 무엇인지를 잘 파악해야 하고 잘 인지 해야 한다. 이러한 여러 상황과 형태의 정보를 제공해주기 위해서 유비쿼터스 컴퓨팅 시스템은 사용자의 정보가 필수적이며, 이러한 정보를 얻기 위해서는 RFID와 같은 기술을 바탕으로 유비쿼터스 컴퓨팅 시스템의 환경을 구축해야 한다. 그러므로 외부환경변화에 따른 가변적인 환경이나 사용자가 원하는 정보를 제공해줄 수 없다면 유비쿼터스 컴퓨팅 시스템이라 할 수 없고, 외부환경변화에 대한 감지는 유비쿼터스 컴퓨팅 시스템의 가장 중요하면서도 가장 핵심이 되는 기술이다.

##### 4.2 사람과 컴퓨터

유비쿼터스 컴퓨팅 시스템과 사람의 조화는 다음 아닌 인간 친화적인 인터페이스를 통해 연결된다[1,2]. 인터페이스를 통하여 사용자는 시스템과 커뮤니케이션 할 수 있고, 시스템은 사용자의 요구사항을 만족시켜주게 되며, 시스템이 사람과 사물 그리고 환경과 서로 혼합이 되어 인간의 편리한 생활을 도울 수 있게 된다. 그 동안 한정된 공간에서만 존재했던 컴퓨터들은 인간생활과 연계되어 인간의 생활공간 범위를 확장시킨 것이라고 할 수 있다. 이제는 더 이상 사람과 컴퓨터는 동떨어져서 생각할 수 없다.

##### 4.3 어디서나 사용 가능한 컴퓨터

유비쿼터스 컴퓨팅 시스템은 가상공간이 아닌 현실세계 어디서나 컴퓨터의 사용이 가능해야 한다[1,2]. 사용자가 어디에 있어도 사용자의 위치추적, 건강상태, 이동

수단 등의 정보가 실시간으로 검색이 가능해야 하며, 또한 사용자가 필요한 정보를 얻고자 할 때는, 다른 시스템과의 연결을 통하여 필요한 정보를 획득한다. 그러므로, 이러한 환경이 구축되기 위해서는 유비쿼터스 컴퓨팅 시스템환경 내에서 존재할 수 있는 거짓정보나 전파 방해, 수신교란 등에 대해서 방해 받지 않아야 한다.

#### 4.4 아주 작은 컴퓨터(Invisibility)

기존의 방에만 있던 데스크탑 컴퓨터나 혹은 전력이 있어야 운영되는 노트북과는 달리 유비쿼터스 컴퓨팅 시스템은 낮은 전력에도 운영될 수 있는 시스템이며, 또한 눈에 보이지 않을 만큼 크기가 작다[1,2,11,12]. 크기가 매우 작기 때문에 사용자가 인식하지 못하는 사이에도 작동이 될 수 있다. 네트워크를 통해 막대한 정보를 기반으로 하여, 다른 어떠한 시스템보다 인간의 삶을 풍요롭게 한다.

#### 4.5 유비쿼터스 접근

유비쿼터스 컴퓨팅 시스템의 가장 큰 통신수단은 무선통신이고, 무선통신의 발달이 곧 유비쿼터스 컴퓨팅 시스템의 발달이다[3]. 그러므로 이러한 무선통신의 주요기능인 RFID 기술이 기반되어야 한다. 유비쿼터스 컴퓨팅 시스템은 다른 시스템에 대한 접근과 그 시스템과의 정보공유 등을 통하여 운영된다. 따라서 유비쿼터스 컴퓨팅 시스템의 접근성은 중요한 환경요인이 된다.

#### 4.6 시스템들간의 조화

유비쿼터스 컴퓨팅 시스템의 가장 큰 특징은 상호운영성이다. 유비쿼터스 컴퓨팅 시스템은 단독으로도 운영될 수 있지만, 다른 시스템과의 상호운영이 된다면, 더 많은 양의 정보를 공유할 수 있다[13,14]. 또한 유비쿼터스 컴퓨팅 시스템은 개인화된 시스템이기 때문에 사용자의 편리나 혹은 원하는 정보를 얻기 위해서는 시스템들간의 상호운영성은 필수적이라 할 수 있다.

#### 4.7 다대다 접속

기존의 현대의 컴퓨터에 여러 명의 사용자가 접속하던 네트워크 환경에서, 이제는 여러 대의 컴퓨터에 여러 명의 사용자가 접속할 수 있는 네트워크 환경에 이르르게 되었다[1,2]. 이러한 환경은 유비쿼터스 컴퓨팅 시스템 환경에 적합한 환경이다. 이러한 여러 명의 사용자가 접속할 수 있는 환경에서 정보 공유나 혹은 사용자가 원하는 정보를 언제 어디서나 얻을 수 있는 것처럼, 이렇듯 여러 명의 사용자가 접속할 수 있는 환경이 바로 유비쿼터스 컴퓨팅 시스템의 환경이다.

#### 4.8 보안과 프라이버시

유비쿼터스 컴퓨팅 시스템은 개인화된 시스템이다. 이렇게 개인화된 시스템은 사용자의 모든 정보가 내장되어, 그 어떤 시스템 보다 보안성이 더 강조되는 시스템이다[14,15]. 또한 눈에 보이지 않을 정도 크기의 시스템

이기 때문에 도난의 우려나 혹은 도난을 당하더라도 ID/PW의 입력뿐만 아니라, 사용자의 지문이나 안구 등의 확인을 통하여 시스템이 작동 되어야 한다. 유비쿼터스 컴퓨팅 시스템은 언제 어디서나 사용할 수 있는 네트워크 환경에서 존재하기 때문에 항상 시스템의 정보가 노출될 우려가 있고, 사용자의 정보나 특히 사용자의 프라이버시에 대한 보안성이 중요시 된다.

### 5. 유비쿼터스 컴퓨팅 시스템의 품질특성

본 장에서는 유비쿼터스 컴퓨팅 시스템의 중요한 특징을 기반으로 품질모형을 유도하기 위해 주특징과 부특징을 분석하고 정의한다. 따라서 본 장에서는 품질특성의 추출 과정을 알아보고, 품질 특성에 대해서 정의를 내린다.

각 주특징들과 품질특성들간에는 연관관계를 가진다. 그림 4에서 유비쿼터스 컴퓨팅 시스템을 평가하기 위한 품질특성과 유비쿼터스 컴퓨팅 시스템의 특징들 간의 맵핑관계를 표현하였다. 기능성, 사용성, 이식성은 ISO/IEC9126으로부터 도출 되었고, 이동성은 유비쿼터스 컴퓨팅 시스템을 평가하기 위하여 새롭게 정의되었다.

ISO/IEC9126과 유비쿼터스 컴퓨팅 시스템을 평가하기 위해 본 논문에서 제안된 품질모형과는 두 가지 다른 점이 있다. 첫 번째는 ISO/IEC9126은 단일 어플리케이션을 위한 것이고, 제안된 품질모형은 여러 유비쿼터스 컴퓨팅 시스템을 목표로 한 평가모형이다. 두 번째는 ISO/IEC9126은 평가 대상에 대해서 고정된 요구사항을 가지고 있지만, 제안된 품질모형 평가 대상에 대해서는 변할 수 있는 요구사항을 가진다. 그림 5는 ISO/IEC9126으로부터 제안된 품질모형의 품질특성을 재정의한 것이다. 본 논문에서는 ISO/IEC9126으로부터 추출된 기능성, 사용성, 이식성의 특성을 채택하고 새롭게 정의된 이동성의 특성을 정의 한다.

#### 5.1 기능성

기능성의 주특징은 유비쿼터스 컴퓨팅 시스템이 가지고 있어야 할 기능 중에 가장 기본이 되는 상호운영성과 보안성에 대해서 유비쿼터스 컴퓨팅 시스템이 갖추고 있는가를 측정하기 위한 특징이다. 유비쿼터스 컴퓨팅 시스템은 다른 시스템과 상호운영이 되어야 하기 때문에 측정이 필요하고, 또한 유비쿼터스 컴퓨팅 시스템은 개인화된 시스템이기 때문에 그 어떤 시스템보다 보안과 프라이버시가 보장되어야 한다. 따라서 기능성의 측정은 유비쿼터스 컴퓨팅 시스템의 기본적인 품질평가 항목이다.

#### 5.2 사용성

사용성의 주특징은 유비쿼터스 컴퓨팅 시스템을 실제 사용자가 이용하는데 필요한 이해성, 운용성, 선호도 등

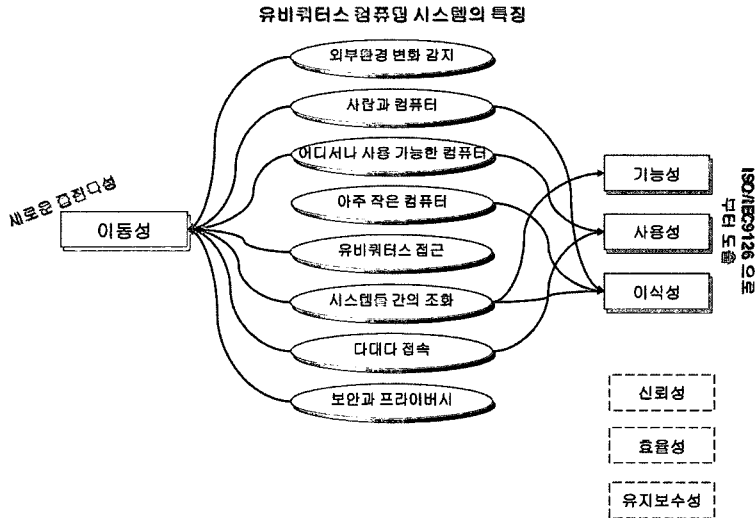


그림 4 유비쿼터스 컴퓨팅 시스템과 특징들간의 맵핑

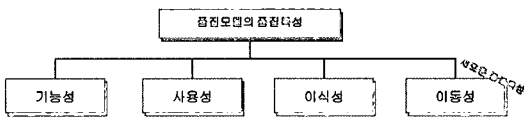


그림 5 품질모델의 품질특성

을 평가하게 된다. 사용자가 시스템으로부터 얻을 수 있는 정보나 데이터 혹은 시스템을 컨트롤하기 위해서는 사용성의 평가가 필요하며, 이 사용성을 통하여 시스템의 전반적인 구성과 이용할 수 있는 기능 등을 파악한다.

5.3 이식성

이식성의 특징은 현재 서비스 받고 있는 환경에서 다른 환경으로 환경이 전이 될 때, 유비쿼터스 컴퓨팅 시스템이 전이된 환경에서도 잘 적응을 할 수 있는지, 혹은 다른 독립적인 시스템과 함께 공존할 수 있는지를 평가한다. 특히 유비쿼터스 컴퓨팅 시스템은 이동성이 많은 시스템이기 때문에 한 곳에서만 머물러 있는 시스템과는 달리 사용자에 의해 환경이 수시로 변경될 수 있기 때문에 품질평가 항목으로 정의하였다.

5.4 이동성

이동성은 유비쿼터스 컴퓨팅 시스템을 품질평가하기 위한 중요한 특징이다. 또한 이동성은 유비쿼터스 컴퓨팅 시스템의 중요한 목표이자, 이동성의 범위나 혹은 특징에 따라 유비쿼터스 컴퓨팅 시스템의 성능이나 품질이 높게 평가 된다. 유비쿼터스 컴퓨팅 시스템에 대한 신뢰성이나 다른 시스템과의 연결성, 외부환경으로부터 인식할 수 있는 지각성, 마지막으로 모든 사용자들이 원하는 선호도가 높은 유비쿼터스 컴퓨팅 시스템이 되어야 한다. 이동성은 유비쿼터스 컴퓨팅 시스템을 보다 정

확하게 평가하기 위해서 ISO/IEC9126에 있는 품질속성 만으로는 품질평가가 미흡하기 때문에 이 항목을 새로이 추가하였으며, 새롭게 추가된 이동성은 기능성이나 사용성, 이식성에 대해서 상호종속적이지 않고 같은 수준의 품질 주특징이다.

5.5 신뢰성, 효율성, 유지보수성

신뢰성, 효율성, 유지보수성의 특징은 단일 시스템이 실행 될 때, 발생하는 여러 가지 측면을 측정하게 되는데, 유비쿼터스 컴퓨팅 시스템은 단일 시스템이긴 하지만, 다른 시스템과의 상호운영 되는 것이 더 중요하다. 예를 들면, 효율성은 얼마나 적합한 성능을 지원하는지를 측정하게 되고, 테스트 혹은 운영 되는 동안에 사용되는 자원과의 관계가 어떻게 되는지를 측정한다. 하지만 유비쿼터스 컴퓨팅 시스템은 단일 시스템으로 운영 되는 것이 아니라 다른 시스템과 함께 운영 되어야 하며 명세 된 성능보다 더 많은 성능과 기능을 지원할 수 있거나 반대로, 다른 시스템과의 운영에서 오히려 명세 된 성능이나 기능보다 떨어질 수 있기 때문이다. 따라서 단일 시스템을 평가하기에 신뢰성, 효율성은 유비쿼터스 컴퓨팅 시스템을 평가하기 위해서는 적합하지 않다.

6. 품질모델의 부특징 도출 및 정의

본 장에서는 제안된 품질모델의 부특징에 대해서 정의한다. ISO/IEC 9126의 부특징을 채택하여 유비쿼터스 컴퓨팅 시스템에 맞게 특화 한 후, 유비쿼터스 컴퓨팅 시스템의 특징을 측정하기 위해 새롭게 정의한다. 표 1은 ISO/IEC 9126 부특징 사이에서 관련성을 주의 깊게 관찰하고 기술하였다. 또한 표 1에서 5점의 척도를 사용한 매우높음(☆☆☆☆)은 실제 유비쿼터스 컴퓨팅 시스

표 1 ISO/IEC 9126의 부특징 특화

특징	부특징	관련성
기능성	적합성	☆☆☆
	정확성	☆☆☆
	상호운영성	☆☆☆☆☆
	보안성	☆☆☆☆☆
	준수성	☆☆
사용성	이해성	☆☆☆☆☆
	학습성	☆☆☆
	운용성	☆☆☆☆☆
	선호도	☆☆☆☆☆
	준수성	☆☆
이식성	적용성	☆☆☆☆☆
	설치성	☆☆☆
	공존성	☆☆☆☆
	대체성	☆☆☆
	준수성	☆☆

☆☆☆☆☆ 매우높음, ☆☆☆☆ 높음, ☆☆☆ 보통, ☆☆ 낮음, ☆ 매우낮음

템을 품질평가하는 가장 기본이 되고 핵심이 되는 평가 항목들이며, 높음(☆☆☆☆)은 유비쿼터스 컴퓨팅 시스템을 품질평가 하는데 기본적으로는 않지만 핵심이 되는 평가항목들이며, 보통(☆☆☆)은 본 논문에서 품질평가항목으로 추출되지는 않았지만 품질평가 항목으로 전혀 관련이 없는 항목은 아니며, 품질평가자에 의해 새롭게 정의 될 수 있다. 낮음(☆☆)은 유비쿼터스 컴퓨팅 시스템과 관련이 작은 품질항목이고, 매우낮음(☆)은 유비쿼터스 컴퓨팅 시스템과 관련이 없는 품질항목이다.

각 절의 부특징 설명과 정의는 유비쿼터스 컴퓨팅 시스템의 품질항목과 관련하여 이에 대해서 논리적으로 근거를 제시하기 위해 추가 되었다. 따라서, 정의에 대한 추가 설명이 없는 항목은 품질 평가 항목에서 제외 하였다.

6.1 기능성

본 절에서는 기능성의 부특징과 유비쿼터스 컴퓨팅 시스템의 관련성에 대해 논리적 근거를 제시한다. 기능성의 품질 부특징 중 상호운영성과 보안성은 유비쿼터스 컴퓨팅 시스템 품질평가와 관계가 있지만, 적합성, 정확성, 준수성과 같은 뚜렷하지 않은 관계도 가지고 있다. 그러므로 기능성의 특징을 측정하기 위해서 상호운영성과 보안성을 채택한다. 이러한 부특징의 도출과 정의는 다음과 같이 유비쿼터스 컴퓨팅 시스템의 특징을 기반으로 하여 특화 한다.

**적합성:** 적합성은 유비쿼터스 컴퓨팅 시스템과 관련성이 높지도 낮지도 않다. 적합성과 관련이 높지 않은 이유는, 개인화된 유비쿼터스 컴퓨팅 시스템의 적합성을 평가하기 위해서는 테스트되기 위한 기능들이 시스템 내부에 존재해야 하지만, 이미 이러한 테스트는 개발공

정 과정에서 이루어 졌다. 그리고 적합성과 관련성이 낮지 않은 이유는 비록 개발과정 과정에서 테스트가 통과되었다 할지라도, 시스템 외부의 인터페이스를 통하여 시스템의 간단한 적합성을 평가할 수 있기 때문에 품질평가 항목으로 도출 되지 않았다.

**정확성:** 정확성은 유비쿼터스 컴퓨팅 시스템과 관련성이 높지도 낮지도 않다. 정확성과 관련이 높지 않은 이유는 시스템에 문제가 발생되면 그 원인과 결과에 대해서 정확히 분석되어야 하지만, 유비쿼터스 컴퓨팅 시스템 자체 내에서 발생한 원인이 아니라 다른 시스템과의 운영 중에 다른 시스템에서 원인이 발생할 수 있기 때문에 정확성을 평가하기에는 어려움이 있다. 그리고, 정확성과 관련이 낮지 않은 이유는 다른 시스템과의 운영 중에도 유비쿼터스 컴퓨팅 시스템 자체 내에서 원인이 발생할 수 있기 때문이다.

**상호운영성:** 상호운영성은 유비쿼터스 컴퓨팅 시스템과 관련성이 매우 높다. 유비쿼터스 컴퓨팅 시스템 평가를 위한 품질속성으로 상호운영성을 도출한 이유는 개발된 유비쿼터스 컴퓨팅 시스템이 다른 시스템과 함께 상호운영이 되어야 하기 때문에 유비쿼터스 컴퓨팅 시스템에서 상호운영성의 평가는 중요한 품질평가 항목이다.

- 정의: 이 품질속성은 유비쿼터스 컴퓨팅 시스템 환경에서 실제 하드웨어와 소프트웨어가 서로 독립적이면서, 다른 시스템과의 상호운영이 이루어져야 한다. 그러므로, RFID와 같은 기술을 이용하여 다른 시스템과의 통신이나 정보교환에 대해서 정확하게 실행될 수 있어야 한다.

**보안성:** 보안성은 유비쿼터스 컴퓨팅 시스템과 관련성이 매우 높다. 유비쿼터스 컴퓨팅 시스템 평가를 위한 품질속성으로 보안성을 도출한 이유는 개인화된 유비쿼터스 컴퓨팅 시스템 DB에 사용자의 정보와 데이터가 저장 되어 보호가 가능한지를 평가해야 하기 때문에 보안성을 도출 하였다.

- 정의: 이 품질속성은 유비쿼터스 컴퓨팅 시스템 환경에서 다른 독립적인 시스템과의 운영 중에 사용자의 정보나 데이터가 외부로 유출된다면 사용자는 많은 피해를 볼 것이며 그러므로, 시스템에 대한 선호도나 혹은 신뢰성이 떨어질 수 있기 때문에 보안성에 대한 평가는 중요하다.

**준수성:** 유비쿼터스 컴퓨팅 시스템은 기능성의 준수성과 관련성이 낮다. 그 이유는 유비쿼터스 컴퓨팅 시스템에서 준수성에 관한 표준 규칙과 규정이 있을 경우, 그러한 사항들에 대해서 만족하면 되기 때문에 관련성이 낮다.

6.2 사용성

본 절에서는 사용성의 부특징과 유비쿼터스 컴퓨팅

시스템의 관련성에 대해 논리적 근거를 제시한다. 사용성의 품질 부특징 중 이해성과 운용성, 선호도는 유비쿼터스 컴퓨팅 시스템 품질평가와 관계가 있지만, 학습성, 준수성과 같은 뚜렷하지 않은 관계도 가지고 있다. 그러므로 사용성의 특징을 측정하기 위해서 이해성과 운용성, 선호도를 채택한다. 이러한 부특징의 도출과 정의는 다음과 같이 유비쿼터스 컴퓨팅 시스템의 특징을 기반으로 하여 특화 한다.

**이해성:** 이해성은 유비쿼터스 컴퓨팅 시스템과 관련성이 매우 높다. 유비쿼터스 컴퓨팅 시스템 평가를 위한 품질속성으로 이해성을 도출한 이유는 유비쿼터스 컴퓨팅 시스템에는 많은 기능들이 존재한다. 이러한 기능들에 대해서 사용자는 이해를 해야 하고 올바르게 사용해야 한다.

- **정의:** 사용자를 위해 많은 기능을 제공하는 유비쿼터스 컴퓨팅 시스템에 대해서 사용자가 그 기능들에 대해서 이해를 하지 못한다면 유비쿼터스 컴퓨팅 시스템을 사용하거나 관리하는데 많은 어려움이 따를 것이다. 따라서 유비쿼터스 컴퓨팅 시스템은 복잡한 기능이라 할 지라도, 사용자가 쉽게 이해 할 수 있도록 해야 하기 때문에 품질평가항목으로 도출하였다.

**학습성:** 학습성은 유비쿼터스 컴퓨팅 시스템과 관련성이 높지도 낮지도 않다. 학습성과 관련이 높지 않은 이유는 유비쿼터스 컴퓨팅 시스템을 사용하는 사용자는 시스템 내부가 어떻게 구성되어 있으며 어떻게 운영되는지를 학습할 필요는 없다. 그렇지만, 시스템의 OS나 플랫폼, 내부가 어떠한 하드웨어로 구성되어 있는지에 대한 정도만 사용자가 알고 있으면 되기 때문에 품질평가항목으로 도출되지 않았다.

**운용성:** 운용성은 유비쿼터스 컴퓨팅 시스템과 관련성이 매우 높다. 유비쿼터스 컴퓨팅 시스템 평가를 위한 품질속성으로 운용성을 도출한 이유는 실제 유비쿼터스 컴퓨팅 시스템을 사용자가 운용할 수 있어야 하고, 컨트롤 할 수 있어야 한다. 유비쿼터스 컴퓨팅 시스템은 사용자가 인식 못하는 사이에도 작동을 할 수 있는 시스템이기 때문에 사용자가 특정 기능에 대해서 컨트롤이 되어야 한다.

- **정의:** 유비쿼터스 컴퓨팅 시스템은 눈에 보이지도 않을 만큼의 아주 작은 시스템이다. 사용자가 인식을 못하는 상황에서도 유비쿼터스 컴퓨팅 시스템은 운용되고 있다. 따라서 사용자가 인식을 못하는 사이에 외부 시스템으로부터 사용자의 개인 정보나 데이터가 유출될 수 있기 때문에 중요한 몇 가지의 기능은 사용자가 컨트롤 할 수 있어야 한다.

**선호도:** 선호도는 유비쿼터스 컴퓨팅 시스템 평가를 위한 품질속성으로 선호도를 도출한 이유는 많은 기능

을 가지고 사용자의 편리성을 고려한 시스템이라 할지라도 사용자로부터 흥미를 끌지 못한다면 시스템의 품질은 낮다고 할 수 있다.

- **정의:** 선호도를 도출한 이유에서처럼 많은 기능이 사용자의 편리성을 최대한 보장한다 할지라도 사용자의 흥미를 얻지 못한다면 시스템의 품질은 낮게 평가될 것이다. 그러므로, 사용자가 원하는 요구사항을 잘 파악하여 사용자들로 하여금 선호도가 높게 개발되어야 할 것이다.

**준수성:** 유비쿼터스 컴퓨팅 시스템은 사용성의 준수성과 관련성이 낮다. 그 이유는 유비쿼터스 컴퓨팅 시스템에서 준수성에 관한 표준 규칙과 규정이 있을 경우, 그러한 사항들에 대해서 만족하면 되기 때문에 관련성이 낮다.

### 6.3 이식성

본 절에서는 이식성의 부특징과 유비쿼터스 컴퓨팅 시스템의 관련성에 대해 논리적 근거를 제시한다. 이식성의 품질 부특징 중 적응성과 공존성은 유비쿼터스 컴퓨팅 시스템 품질평가와 관계가 있지만, 설치성, 대체성, 준수성과 같은 뚜렷하지 않은 관계도 가지고 있다. 그러므로 이식성의 특징을 측정하기 위해서 적응성과 공존성을 채택한다. 이러한 부특징의 도출과 정의는 다음과 같이 유비쿼터스 컴퓨팅 시스템의 특징을 기반으로 하여 특화 한다.

**적응성:** 적응성은 유비쿼터스 컴퓨팅 시스템과 관련성이 매우 높다. 유비쿼터스 컴퓨팅 시스템 평가를 위한 품질속성으로 적응성을 도출한 이유는 유비쿼터스 컴퓨팅 시스템이 사용자에게 얼마나 친화적일 수 있는가를 평가한다. 유비쿼터스 컴퓨팅 시스템은 컴퓨터이면서도 컴퓨터가 아닌 시스템이기 때문에 인간 친화적인 인터페이스들을 평가해야 한다.

- **정의:** 유비쿼터스 컴퓨팅 시스템 평가를 위한 적응성을 정의한 이유는 유비쿼터스 컴퓨팅 시스템과 사람과의 관계에서의 친화적인 인터페이스가 반드시 필요하며, 이러한 인터페이스의 관계가 올바른지를 평가해야 한다.

**설치성:** 설치성은 유비쿼터스 컴퓨팅 시스템과 관련성이 높지도 낮지도 않다. 설치성과 관련이 높지 않은 이유는 유비쿼터스 컴퓨팅 시스템이 명세 되거나 혹은 지정된 환경에서 설치가 가능한지를 평가해야 하는데, 유비쿼터스 컴퓨팅 시스템은 유비쿼터스 컴퓨팅 시스템 환경에서만 존재 할 수 있다. 환경자체가 유비쿼터스 컴퓨팅 시스템 환경이 아니라면, 더 이상 유비쿼터스 컴퓨팅 시스템은 존재 할 수 없다. 그리고, 설치성과 관련이 낮지 않은 이유는 유비쿼터스 컴퓨팅 시스템 환경이라 할지라도 유비쿼터스 컴퓨팅 시스템은 설치가 가능한



소프트웨어가 아니라 하나의 독립적인 시스템이기 때문에 품질평가항목으로 도출되지 않았다.

**공존성:** 공존성은 유비쿼터스 컴퓨팅 시스템과 관련성이 매우 높다. 유비쿼터스 컴퓨팅 시스템 평가를 위한 품질속성으로 공존성을 도출한 이유는 유비쿼터스 컴퓨팅 시스템은 독립적인 시스템이기 하나, 다른 유비쿼터스 컴퓨팅 시스템과 함께 공존할 수 있어야 하고, 다른 독립적인 시스템과의 커뮤니케이션이 되어야 하기 때문에 공존성의 평가는 중요하다.

- **정의:** 이 품질속성은 유비쿼터스 컴퓨팅 시스템은 서로 다른 독립적인 시스템들과의 공존성이 있어야 한다. 서로 다른 OS(Operation System)환경이나 플랫폼이라 할지라도 유비쿼터스 컴퓨팅 시스템 환경 안에서는 시스템들 사이에서 서로 공존할 수 있어야 하기 때문에 품질평가항목으로 도출하였다[16].

**대체성:** 대체성은 유비쿼터스 컴퓨팅 시스템과 관련성이 높지도 낮지도 않다. 대체성과 관련이 높지 않은 이유는 유비쿼터스 컴퓨팅 시스템은 단일 시스템이기 때문에 다른 시스템과 대체 되는 것을 평가하기에는 어려움이 있다. 또한 대체성과 관련이 낮지 않은 이유는 유비쿼터스 컴퓨팅 시스템 내부의 OS나 혹은 플랫폼이 다른 소프트웨어와 변경이 될 수는 있지만, 본 논문에서 제시한 유비쿼터스 컴퓨팅 시스템 품질평가에서는 범위 밖의 내용이다.

**준수성:** 유비쿼터스 컴퓨팅 시스템은 이식성의 준수성과 관련성이 낮다. 그 이유는 유비쿼터스 컴퓨팅 시스템에서 준수성에 관한 표준 규칙과 규정이 있을 경우, 그러한 사항들에 대해서 만족하면 되기 때문에 관련성이 낮다.

6.4 이동성

본 절에서는 이동성의 부특징과 유비쿼터스 컴퓨팅 시스템과의 관련성에 대해 논리적 근거를 제시한다. ISO/IEC 9126기반의 부특징을 이용하여 유비쿼터스 컴퓨팅 시스템의 모든 특징을 평가하기에는 미흡한 점이 있다. 제안된 유비쿼터스 컴퓨팅 시스템의 특징을 평가하기 위한 추가적인 부특징을 제안하고 도출한 근거를 제시한다. 또한 유비쿼터스 컴퓨팅 시스템이 언제 어디서든 사용 가능해야 한다. 유비쿼터스 컴퓨팅 시스템 평가를 위한 이동성을 정의한 이유는 유비쿼터스 컴퓨팅 시스템의 중요한 환경적인 요소는 네트워크이다[17].

**신뢰성:** 신뢰성은 유비쿼터스 컴퓨팅 시스템이 가지고 있는 기능이나 혹은 인터페이스, 자료저장, 전력유지 등에 대해서 사용자에게 신뢰성이 보장되어야 한다. 또한 유비쿼터스 컴퓨팅 시스템은 다른 시스템과의 운영 중 시스템 자체에서 가지고 있는 기능들이 재 성능을 유지할 수 있어야 한다.

- **정의:** 유비쿼터스 컴퓨팅 시스템 평가를 위해 신뢰성을 정의한 이유는 유비쿼터스 컴퓨팅 시스템은 성숙성이 보장되어야 한다. 그래서, 시스템 자체에 실패나 오류가 발생 하였을 때, 회복성이 보장되어야 하고, 성능이 저조하거나 외부의 환경이 다른 이유로 동작이 안될 경우에 다시 회복할 수 있어야 한다. 따라서 유비쿼터스 컴퓨팅 시스템은 사용자로 하여금 신뢰성이 명확하게 보장되어야 한다.

**연결성:** 연결성은 유비쿼터스 컴퓨팅 시스템 환경에서 정보공유를 위해 서로 독립적인 유비쿼터스 컴퓨팅 시스템이라 할지라도, 유비쿼터스 컴퓨팅 시스템환경 내에서는 정보공유가 가능해야 한다. 그러므로, 정보공유가 가능 하려면 함께 운영되는 동안에 발생할 수 있는 실패에 대해서 얼마나 회피할 수 있는가를 평가한다.

- **정의:** 이 품질속성은 시스템들 간의 명세된 인터페이스가 규정에 의해 되어서는 안되기 때문이다. 따라서, 유비쿼터스 컴퓨팅 시스템의 연결관계에 대한 평가가 필요하기 때문에 품질평가항목으로 도출하였다.

**지각성:** 지각성은 유비쿼터스 컴퓨팅 시스템이 외부 환경에 대해서 얼마나 지능적으로 반응하는가에 대한 평가이다. 개인화된 시스템에 대해서 지능적으로 반응하기 위해서는 다양한 형태의 정보가 필요하다. 외부 환경에 대해서 반응하는 대표적인 기술로는 크게 RFID와 센서가 있다. RFID는 칩과 안테나 리더기로 구성된 무선주파수 시스템이고, 센서는 빛, 온도, 냄새 등을 통하여 간지한다.

- **정의:** 이 품질속성은 유비쿼터스 컴퓨팅 시스템이 외부 환경으로부터 지능적으로 반응할 수 있는 기술을 가지고 있어야 한다. 유비쿼터스 컴퓨팅 시스템은 사용자의 상황에 따라 각 개인의 서비스가 다르고 변할 수 있기 때문에 품질평가항목으로 도출하였다.

다음 그림 6은 유비쿼터스 컴퓨팅 시스템을 품질평가하기 위한 품질모델이다.

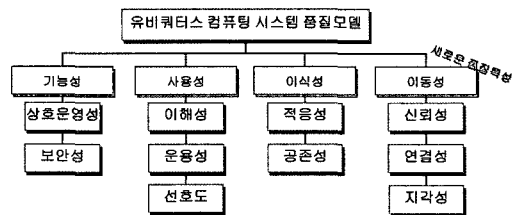


그림 6 유비쿼터스 컴퓨팅 시스템의 품질모델(UCQM)

7. 메트릭(Metric) 정의

이 장에서는 품질속성별 평가 메트릭을 제안한다. 각 메트릭에 대한 설명, 공식, 값의 범위, 그리고 관련된 해

석 등의 항목을 제시한다. 그림 7은 각 품질속성과 메트릭들 간의 맵핑을 보여준다. 또한 본 장에서 제시한 메트릭은 품질모델 및 메트릭의 표준인 ISO/IEC 9126의 메트릭 스케일을 준수하여 정의하였지만, 특정한 시스템이나 프로젝트의 특성을 고려하여 대표적 메트릭을 다소 수정하거나 최적화(Adaptation) 될 경우도 있다. 따라서 본 논문에서 제시한 메트릭은 대표적 메트릭의 의미를 가지지 않으며, 절대적 대표성을 가지지 않는다.

본 장에서 사용된 메트릭들은 상호운영성을 측정하기 위한 메트릭은 SI(System Interoperability)이고, 보안성을 측정하기 위한 메트릭은 PS(Privacy Security)이고, 이해성을 측정하기 위한 메트릭은 SU(System Understandability)이고, 운용성을 측정하기 위한 메트릭은 UO(User Operability)이고, 적응성을 측정하기 위한 메트릭은 HA(Human Adaptability)이고, 공존성을 측정하기 위한 메트릭은 TE(Together Existence)이고, 신뢰성을 측정하기 위한 메트릭은 PR(Performance Reliability)이고, 연결성을 측정하기 위한 메트릭은 PC(Public Connectability)이고, 지각성을 측정하기 위한 메트릭은 SE(Sensorability Environment)이고, 선호도를 측정하기 위한 메트릭은 PA(Product Attractiveness)이라 지정하였다. 메트릭의 이름은 각 품질속성의 특징을 잘 살려 명명하였다.

7.1 상호운영성

이 메트릭은 유비쿼터스 컴퓨팅 시스템이 다른 유비쿼터스 컴퓨팅 시스템과 상호운영성이 잘 이루어지는가를 측정하는 메트릭이다[18]. 이 메트릭은 다음과 같이 계산된다;

$$SI = 1 - ((\text{시스템간의 정보교환이 실패한 수}) / (\text{시스템간의 정보교환을 시도한 총 횟수}))$$

이 메트릭에서 정보교환의 실패는 사용자가 원하는 정보를 얻지 못했을 경우나, 다른 외부의 충돌로 인하여 생길 수 있는 실패를 말한다[11]. 상호운영성은 시스템들간의 정보교환이 얼마나 정확하게 이루어지는가를 평가한다. 상호운

영성을 잘 평가하기 위해서는 실패한 정보교환의 수를 파악하여 측정한다.

그러므로, SI의 범위는 0.1이고, SI가 높은 값을 가진다면 상호운영성은 높을 것이며, 정보교환의 실패한 수가 없다면 SI는 1의 값을 가진다. 따라서 SI의 값이 1에 가까울수록 유비쿼터스 컴퓨팅 시스템들 간의 상호운영성은 높다는 것을 의미한다.

7.2 보안성

이 메트릭은 개인화된 유비쿼터스 컴퓨팅 시스템에서 개인의 신상명세나 혹은 외부 시스템의 충격이나 오류로 인하여 정보가 유출될 수 있는가를 평가하는 메트릭이다. 이 메트릭은 다음과 같이 계산 된다;

$$PS = (\text{외부시스템과의 접속 후 발생하여 해결된 실패의 수}) / (\text{외부시스템과의 접속 후 발생한 전체 실패의 수})$$

이 메트릭에서의 실패는 정보유출, 시스템 오류 등을 말한다. 그러므로 유비쿼터스 컴퓨팅 시스템의 안전성을 잘 평가하기 위해서는 실패를 잘 파악하여 측정한다.

그러므로, PS의 범위는 0.1이고, PS가 낮은 값을 가진다면 보안성은 높게 평가 될 것 이며, 실패의 수가 0 이면 PS의 값은 0의 값을 가진다. 그러므로 PS의 값이 0에 가까울수록 유비쿼터스 컴퓨팅 시스템의 보안성은 높게 평가 될 것이다.

7.3 이해성

이 메트릭은 사용자가 유비쿼터스 컴퓨팅 시스템에 대해서 얼마나 용이하고, 능률적이며 정확하게 이해할 수 있는지를 측정한다[19]. 유비쿼터스 컴퓨팅 시스템 설명서에는 사용자 매뉴얼, 명세서 등의 문서들이 포함되어 있다. 이 메트릭은 다음과 같이 계산된다;

$$SU = (\text{이해할 수 있는 기능들의 수}) / (\text{전체 기능들의 수})$$

기능들은 유비쿼터스 컴퓨팅 시스템을 구성하는 대표적인 아이탬들이며, 이 기능들의 이해를 여러 가지 관점에 의해 표현된다. 예를 들어 보안에 관련된 기능의 경우에는 사용

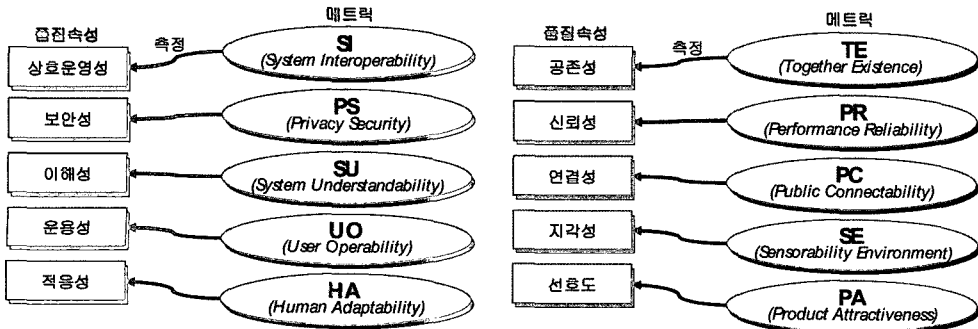


그림 7 품질속성과 메트릭들 간의 맵핑

자 개인의 정보가 들어있기 때문에 사용자가 알아야 하는 기능이고, 사용자는 이 기능을 이해해야 한다. 그러나, 이해할 수 있는 정도는 시스템의 특징을 고려하여 품질평가가 결정한다.

- 각 기능들은 쉽고, 기능들에 대해서 명확하고 정확하게 서술되어 있는가?
- 각 기능들의 구조는 쉽게 설명이 되어 있는가?
- 각 기능들 사이의 관계가 복잡한가?

그러므로 SU의 범위는 0..1이고, SU가 0이나 혹은 낮은 값을 가진다면 이해성에 대한 평가가 낮게 평가 될 것이며, 기능들에 대해서 사용자가 이해를 많이 하고 있다면 SU의 값은 1에 가까울 것이며 이해성에 대한 평가가 높게 평가 될 것이다. 따라서 SU의 값이 1에 가까우면 어려운 없이 유비쿼터스 컴퓨팅 시스템을 용이하게 이해할 수 있다.

#### 7.4 운용성

이 메트릭은 유비쿼터스 컴퓨팅 시스템이 사용자의 개인 정보나 데이터 유출 등의 우려로 사용자 스스로가 시스템을 컨트롤 할 수 있는가에 대해 측정을 한다. 이 메트릭은 다음과 같이 계산된다;

$$UO = 1 - \{(\text{사용자가 시스템의 데이터를 변화시킬 수 없는 수}) /$$

(사용자가 데이터를 변화시킬 수 있는 전체의 수)}

운용성은 사용자가 시스템에 대한 컨트롤 여부에 대한 측정이기 때문에 실제 사용자가 조작하여 시스템의 데이터를 컨트롤할 수 있는가에 대해 평가한다. 여기서 말하는 데이터란 사용자의 개인정보나 혹은 시스템들간의 메시지 정보 등을 말한다.

UO의 범위는 0..1이고, UO의 값이 1에 가까울수록 운용성의 품질은 높게 평가가 된다. 그러므로, UO의 값이 1에 가까울수록 사용자가 시스템을 어려움 없이 컨트롤 할 수 있다는 의미다.

#### 7.5 적응성

이 메트릭은 유비쿼터스 컴퓨팅 시스템이 사람에게 얼마나 친화적인지를 평가하는 메트릭이다. 또한 유비쿼터스 컴퓨팅 시스템이 제공하는 기능들을 바탕으로 사용자로부터 만족도를 평가한다. 이 메트릭은 다음과 같이 계산 된다;

$$HA = 1 - \{(\text{소프트웨어가 테스트 되는 동안 만족 치 못한 기능의 수}) /$$

(테스팅 되는 전체 기능의 수)}

이 메트릭에서 정의한 만족 치 못한 기능은 사용자의 요구 사항을 만족 치 못하는 기능이다. 또한, 소프트웨어는 유비쿼터스 컴퓨팅 시스템 소프트웨어를 말한다. 소프트웨어의 운영에 대한 기능에 대해서 평가를 한다. 그러나, 기능에 대한 만족할 수 있는 정도는 소프트웨어와 관련된 특징을 고려하여 품질평가가 결정한다.

그러므로 HA의 범위는 0..1이다. HA가 1에 가까울수록 유비쿼터스 컴퓨팅 시스템의 적응성은 높게 평가 된다. 또한 만족 치 못한 기능의 수가 0이라면 HA의 값은 1이 된다. 그러므로 유비쿼터스 컴퓨팅 시스템의 적응성 높다면, 사용자와 친화성이 높아져 시스템의 만족도 높게 평가 될 것이다.

#### 7.6 공존성

이 메트릭은 유비쿼터스 컴퓨팅 시스템이 다른 독립적인 시스템과 함께 공존할 수 있는가를 평가하는 메트릭이다. 이 메트릭은 다음과 같이 계산 된다;

$$E(\text{Existence}) = (\text{다른 소프트웨어와 함께 운영되는 동안 어떤 제약이나 예상치 못한 실패의 수})$$

$$TOI(\text{Together Operability Time}) = (\text{다른 소프트웨어와 함께 운영되는 시간})$$

$$TE = 1 - (E / TOT)$$

본 논문에서 공존성을 평가하기 위해 시스템의 특징에 따라 품질평가자가 다른 소프트웨어와 함께 운영되는 시간을 새로이 적용할 수 있다. 함께 공존하여 운영되는 시간 동안 일어날 수 있는 결과에 대해서 측정이 되기 때문에 유비쿼터스 컴퓨팅 시스템을 평가할 때 중요한 품질항목이다.

TE의 범위는 0..1이고, TE의 값이 1에 가까울수록 공존성은 높게 평가 될 것이다. 그러므로, TE의 값이 높을수록 시스템들 간의 공존성은 잘 이루어질 수 있다.

#### 7.7 신뢰성

이 메트릭은 유비쿼터스 컴퓨팅 시스템의 행위와 관련된 속성들을 측정한다. 즉, 시스템의 행위라 함은 유비쿼터스 컴퓨팅 시스템이 운영되는 동안에 시스템의 성능과 수준을 유지할 수 있는 능력을 평가한다. 신뢰성은 2개의 메트릭 *System Maturity (SM)*와 *System Recoverability (SR)*를 통해서 측정된다.

첫 번째, 성숙성은 *System Maturity (SM)*로 계산되며, 유비쿼터스 컴퓨팅 시스템의 실패를 발견하기 위한 테스트 방법과 이러한 실패 처리를 유비쿼터스 컴퓨팅 시스템 자체적으로 어떻게 처리 하는가를 알아야 할 것이다. 이 메트릭은 다음과 같이 계산된다;

$$SM = 1 - \{(\text{유비쿼터스 시스템에서 발견된 실패의 수}) /$$

(유비쿼터스 시스템의 테스트케이스의 수)}

유비쿼터스 컴퓨팅 시스템에서 발견될 수 있는 실패는 유비쿼터스 컴퓨팅 시스템을 테스트하기 위한 사례에서 도출된 실패들을 말한다.

SM의 값은 0..1이고, SM의 값이 1에 가까울수록 성숙성의 품질은 높게 평가되며 SM이 높을수록 유비쿼터스 컴퓨팅 시스템의 성숙도는 높다는 의미이다.

두 번째, 회복성은 *System Recoverability (SR)*로 계산되며, 유비쿼터스 컴퓨팅 시스템의 성능이 저조할 때, 적절한 레벨까지 재 생성할 수 있어야 하고, 유비쿼

터스 컴퓨팅 시스템이 외부의 환경이나 다른 이유로 운영이 안될 경우 이러한 환경에 영향을 받을 수 있는 개인화된 정보들이나 데이터를 복구 할 수 있어야 한다. 이 메트릭은 다음과 같이 계산된다;

$$T = \text{시스템이 다운(정지, 멈춤, Down)된 전체 시간}$$

$$SR = 1 - \{ (\text{시스템의 최악의 경우나 다운된 경우를 관찰하여 발견한 오류의 횟수}) / T \}$$

본 논문에서 회복성을 평가하기 위해 시스템이 다운되는 시간을 10초로 제약한다. 시스템의 특징에 따라 품질평가자는 시스템이 다운되는 시간을 새로이 적용할 수 있다. 유비쿼터스 컴퓨팅 시스템의 최악의 경우나 다운된 경우 시스템이 외부 환경에 의해서 오작동을 하거나 혹은 작동이 안될 경우를 말한다.

SR의 값은 0.1이고, SR의 값이 1에 가까울수록 회복성의 품질은 높게 평가되며 SR이 높은 값을 가질수록 유비쿼터스 컴퓨팅 시스템의 회복성이 높다는 의미이다.

마지막으로, 앞에서 측정된 SM과 SR의 메트릭들을 사용하여, *Performance Reliability (PR)*을 측정한다.

$$PR = W_{SM} \times SM + W_{SR} \times SR$$

$W_{SM}$ ,  $W_{SR}$ 은 각 메트릭을 위한 가중치이며, 각 가중치의 값은 유비쿼터스 컴퓨팅 시스템의 특징을 고려하여 품질평가자에 의해 결정되며 이 가중치들의 합은 1이다.

그러므로, PR의 범위는 0.1이다. 따라서, PR의 값이 1에 가까울수록 유비쿼터스 컴퓨팅 시스템에 대한 신뢰도는 높다.

### 7.8 연결성

유비쿼터스 컴퓨팅 시스템에 장착되어 있는 RFID칩을 이용하여 다른 시스템과 서로 상호운영 될 때, 명세된 인터페이스 규정에 의해서 상호작용이 되어야 한다. 이 메트릭은 다음과 같이 계산된다;

$$PC = (\text{시스템들간의 심각한 실패를 회피할 수 있는 횟수}) / (\text{시스템들이 운영되어 실행되는 테스트케이스의 수})$$

연결성은 시스템과 시스템들간의 운영되는 동안 발생될 수 있는 실패와 운영패턴들을 측정하게 되고, 이 메트릭에서 말하는 실패란 명세된 인터페이스가 규정에 위배 되었을 경우를 말하며, 이로 인해서 시스템들간의 불명확한 상호운영이 발생될 수 있다.

PC의 값은 0.1의 값을 가지며, PC가 1에 가까울수록 유비쿼터스 컴퓨팅 시스템의 연결성은 높게 평가 된다. 또한 명세된 인터페이스가 규정에 어긋날수록, PC는 0에 가까운 값을 가지게 될 것이다. 따라서 PC가 높은 값을 가진다면, 유비쿼터스 컴퓨팅 시스템의 연결성은 높다.

### 7.9 지각성

유비쿼터스 컴퓨팅 시스템은 사용자의 상황에 따라 각각의 개인 서비스가 변하고 달라진다. 여기서 사용자

의 상황이란 시간, 날씨, 온도, 명암, 장치, ID등 사용자에게 서비스를 지능적으로 다양하게 제공해야 한다 [20,21]. 이 메트릭은 다음과 같이 계산된다;

$$SE = \frac{\sum_{i=1}^n (\text{시스템이 제공할 수 있는 서비스의 수})}{\sum_{i=1}^n (\text{가변적인 환경에서 시스템이 제공할 수 있는 서비스의 수})}$$

가변적인 환경이란 어떤 상황에서 어떠한 사용자에게 어떠한 것이라도 제공할 수 있는 상황을 말한다. 각 개인의 환경이나 상황이 다르기 때문에 서비스는 지능적으로 제공되어야 한다.

SE의 범위는 0.1이고, SE의 값이 0에 가깝다면 환경이 변할 때 마다 제공할 수 있는 서비스는 한계가 있다. 그러므로, SE의 값이 1에 가깝다면 환경이 어떻게 변해도 어떠한 사용자가 사용을 해도 시스템의 지각성은 높다.

### 7.10 선호도

이 메트릭은 유비쿼터스 컴퓨팅 시스템이 사용자로부터 얼마나 많은 선호도를 가지고 있는가를 측정하는 메트릭이다. 다시 말해서, 사용자가 시스템을 커스터마이즈(Customize)할 수 없다면, 시스템의 품질은 낮게 평가 될 것이다. 이 메트릭은 다음과 같이 계산된다;

$$PA = (\text{사용자가 만족할 수 있는 커스터마이징 인터페이스의 수}) /$$

(사용자가 커스터마이징 할 수 있는 전체 인터페이스의 수)

선호도는 사용자가 시스템의 외형이나 사용자가 편리하게 사용할 수 있도록 커스터마이징 될 수 있는 인터페이스의 존재 여부를 알아본다. 사용자가 원하는 대로 커스터마이징 될 수 있다면 당연히 시스템에 대한 선호는 높ی 평가 될 것이다.

PA의 범위는 0.1이고, PA의 값이 0에 가깝다면 유비쿼터스 컴퓨팅 시스템에 대한 품질평가는 낮게 평가 될 것이다. 다시 말해, 사용자의 편리대로 커스터마이징 될 수 있는 인터페이스가 많이 존재한다면 PA의 값은 1에 가까운 것이다. 그러므로, PA의 값이 1이라면 유비쿼터스 컴퓨팅 시스템에 대한 선호도는 가장 높게 평가 될 수 있다.

## 8. 사례연구

본 장에서는 유비쿼터스 컴퓨팅 시스템을 품질평가하기 위하여 가상의 유비쿼터스 환경을 만들어, 5장에서 제시한 메트릭을 유비쿼터스 컴퓨팅 시스템에 적용한다. 다음 그림 8은 유비쿼터스 컴퓨팅 시스템을 평가하기 위한 유비쿼터스 환경(Ubiquitous Environment)이고, 유비쿼터스 라이프는 본 논문에서만 사용되었고 시나리오 적용을 위한 특정 환경이다.

### 8.1 유비쿼터스 컴퓨팅 시스템의 환경명세

유비쿼터스의 환경에서 존재하는 유비쿼터스 라이프

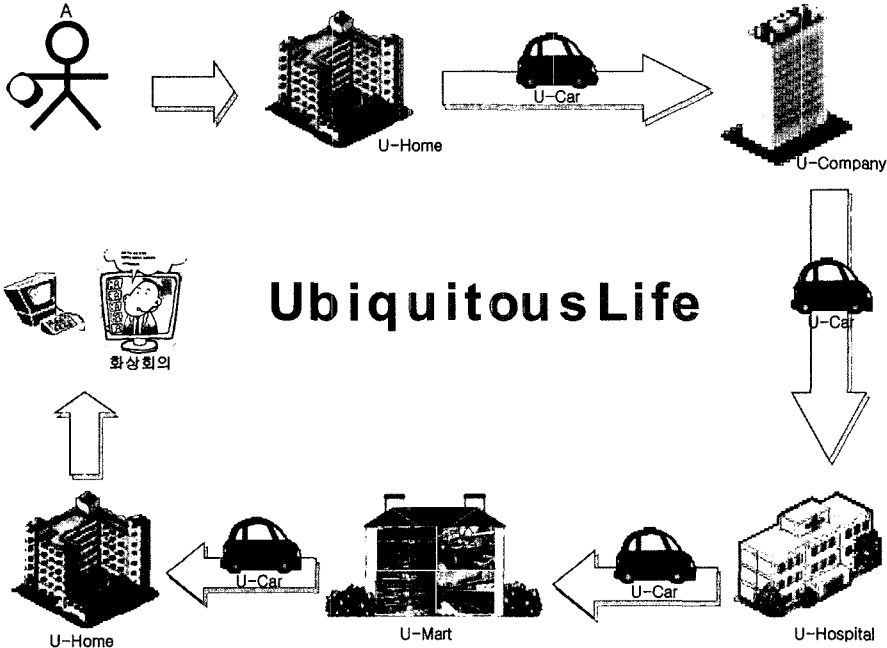


그림 8 유비쿼터스 환경 (Ubiquitous Environment)

는 A의 일상생활을 말한다. 유비쿼터스 환경에는 집, 회사, 자동차, 병원, 마트 등을 포함한다. 사례연구를 위한 시나리오는 다음과 같다.

사용자 A는 유비쿼터스 환경에서 존재하는 아파트에서 살고 있다. 처음 이 아파트로 이사온 날 유비컴(UbiCom)이라는 글씨가 새겨진 휴대폰 정도 크기의 시스템을 하나 받았다. 이 시스템에는 사용자의 생체신호를 감지하는 정교한 센서와 극소형 통신모듈(RFID)이 내장되어 있다. 출근 준비를 마친 A는 자동차에 오르지만 열쇠가 필요 없다. 유비컴에 달린 무선태그가 보낸 신호를 자동차가 수신, 스스로 출발신호를 갖춘다. 물론 신원이 확인되지 않으면, 시동이 걸리지 않는다. 또한 자동차에는 GPS시스템을 사용한 길안내와 도로정보는 기본이다. 회사에 도착한 A에게 유비컴은 차를 운전할 때 자동차 핸들을 통해 자신도 모르게 측정된 혈압 데이터가 5회 이상 정상치를 크게 넘었다고 메시지를 보낸다. 또한 이 메시지와 함께 주치의의 만날 것을 권장한다. 유비컴이 체크한 결과는 곧바로 주치의 단말기에 전달되고 주치의는 원격검진을 받아볼 것을 제안한다. 퇴근길에 A는 마트에 갔다. 마트의 전 품목에는 극소형 칩이 내장되어 있으며, 이 칩에 유비컴을 접촉 해본 A는 이 제품의 원산지가 어디고, 가격이 얼마고, 유통기간이 얼마인지를 유비컴을 통해서 알 수 있다. 그렇게 구입한 제품들에 대한 결제는 신용카드나 혹은 휴대폰 없이 유비컴 하나로 결제를 하였고, 이 대금들은 월말

정산을 한다. 이렇게 마트에 들러 집으로 가려는 A는 사이버 개인비서에게 전화한 통을 받는다. 전화 내용은 기상청 결과 현재기온이 섭씨 35도이니, 집안을 미리 시원하게 준비해도 좋겠냐는 전화였다. 집에 도착한 A는 집안에서 나오는 시원한 공기와 욕실에서는 시원한 물이 욕조를 채우고 있었다. 샤워를 마친 A는 곧 있을 세계 각 지점의 지점장들과 회의를 시작하기 위해 자리에 앉았다. 이전 세계각국을 회의를 위해서 방문 할 필요가 없어진 것이다. 회의를 마친 A는 피로함을 느꼈으며, 이를 감지한 사이버 개인비서는 평소 A가 좋아하는 음악을 조용히 켜고, 잠자리에 들기를 청하였다. 이렇게 잠든 A는 오늘보다 더 새롭고 더 나은 내일을 기대할 것이다.

본 시나리오에서 유비컴이 제공하는 총 기능의 수는 24개이고, 하루 동안에 발생한 시스템의 오류나 실패는 4회가 발생하였다.

**8.2 상호운영성의 계산 (SI)**

이 시나리오에서 A는 아침에 출근해서 저녁에 집으로 돌아오기까지 유비컴을 12회 사용을 하였고, 그 중에 미처 정보를 받지 못한 횟수는 2회가 있었다.

5.1에서 정의된 바와 같이, SI는  $1 - (2 / 12)$  이다. 그러므로, SI의 값은 0.83이 나오며, 이는 유비쿼터스 컴퓨팅 시스템들 간의 상호운영성은 83%가 이루어졌다는 의미다.

**8.3 보안성의 계산 (PS)**

이 시나리오에서 A는 유비컴을 사용하는데 자동차 시동을 걸 때와 GPS의 사용 그리고 마트에서 결제를 하는 동안에 시스템의 오류가 1건이 발생되었고, 외부로 사용자의 정보유출이 1건이 발생되었다. 그러나 2건 모두 유비쿼터스의 방화벽이나 혹은 다른 보안 시스템으로 인하여 다 차단되어 해결되었다.

5.2에서 정의된 바와 같이, PS는  $2 / 2$ 이 된다. 따라서, PS의 값은 1이 나오며, 이는 유비쿼터스 컴퓨팅 시스템들 간의 보안성은 100%가 이루어졌다는 의미이다.

**8.4 이해성의 계산 (SU)**

이 시나리오에서 A는 처음 유비컴이라는 시스템을 봤을 때, 이 시스템에 대해서 많이 알지 못했다. 따라서 사용자 매뉴얼을 보고 시스템의 기능을 익혔다. 그래서, 유비컴 전체 기능의 수는 24개인데, 그 중에 A가 이해한 기능의 수는 19개이다.

5.3에서 정의된 바와 같이, SU는  $21 / 24$ 가 되며, 이해할 수 있는 기능들을 판단하기 위한 기준은 5.3에 기술되어있는 기준을 참고하면 된다. 그래서, SU의 값은 0.875이 나오며 이는 사용자가 유비쿼터스 컴퓨팅 시스템을 87.5% 이해했다는 의미이다.

**8.5 운용성의 계산 (UO)**

이 시나리오에서 A는 운전하는 도중 자기의 건강상태가 체크 되는 줄 몰랐다. 혹시나 A는 자기의 건강상태가 잘못 측정이라도 될까 그 다음날 운전을 하면서 건강상태가 어떻게 체크되는지를 확인하였다. 확인한 결과 A가 유비컴을 관찰한 시간은 10초였고, 관찰되는 시간 동안 시스템의 에러가 2번 났으며, 이 에러는 모두 취소되거나 해결되었다.

5.4에서 정의된 바와 같이, UO는  $1 - (2 / 10)$ 이 된다. 그러므로, UO의 값은 0.8이 나오며, 1에 가까울수록 사용자가 유비쿼터스 컴퓨팅 시스템을 어려움 없이 컨트롤 할 수 있다. 따라서 유비쿼터스 컴퓨팅 시스템을 운용할 수 있는 운용도는 80%가 된다는 의미 이다.

**8.6 적응성의 계산 (HA)**

이 시나리오에서는 A에게 가상의 개인비서가 직접 대화를 하여 바깥날씨가 무척 더우니 집안이 온도와 욕조에 시원한 물을 받아 놓겠다고 A에게 전화를 한다. 유비컴 시스템이 제공하는 몇 가지의 기능들에 대해서 A가 만족을 해야 하기 때문에 테스트를 하였다. 유비컴 시스템 중 테스트 되는 기능은 24개이며, 이중에 만족치 못한 기능의 수는 3개로 나왔다.

5.5에서 정의된 바와 같이, HA는  $1 - (3 / 24)$ 이 되고, HA의 값은 0.875가 된다. 이는 유비쿼터스 컴퓨팅 시스템이 제공하는 기능들에 대한 적응도가 87.5%가 된다는 의미이다.

**8.7 공존성의 계산 (TE)**

이 시나리오에서는 상호운영성과 보안성에서 시스템을 평가한 것과 같이 자동차 시동을 걸 때와 GPS사용, 마트에서의 결제 등으로 인해 유비컴이 다른 외부 시스템과 함께 운영된 시간은 총 10초이며, 외부환경으로부터의 제약이나 예상치 못한 실패는 상호운영성에서 2회이다.

5.6에서 정의된 바와 같이 TE는  $1 - (2 / 10)$ 이 된다. 그러므로, TE의 값은 0.8이 나오며, 1에 가까울수록 유비쿼터스 컴퓨팅 시스템들 간의 공존도가 높다는 의미이다. 따라서 다른 시스템과의 공존성은 80%가 이루어졌다는 의미이다.

**8.8 신뢰성의 계산 (PR)**

이 시나리오를 기반으로 하여 유비컴의 신뢰성을 측정하기 위해서는 SM(성숙성)과 SR(회복성)의 메트릭을 적용한다. 먼저 하루 동안에 발생한 오류나 실패의 수가 4개이고, 유비컴을 테스트 하기 위한 테스트케이스의 수는 유비컴이 제공하는 총 기능의 수인 24개와 같다. 또한 좀더 정확한 신뢰성을 평가하기 위하여 시스템을 정지를 시켜 다시 회복 되는 시간 동안 발생할 수 있는 고장의 횟수를 알아보기 위하여 시스템을 10초간 정지시켰고, 정지된 시간 동안 외부 환경으로부터 발생한 오류의 횟수는 2회이다.

5.7에서 정의된 SM은 실패처리를 유비컴 자체적으로 어떻게 처리 되는가를 알아보기 위해 테스트 케이스와 발견된 실패의 수를 적용하여 평가한다. 따라서  $SM = 1 - (4 / 24)$ 가 되고, 값은 0.83이 나왔으며 1에 가까운 수치가 나왔기 때문에 성숙성의 품질은 높다.

5.7에서 정의된 SR은 유비컴이 다운된 경우나 외부 환경에 의해서 오작동 되는 경우를 평가한다. 따라서,  $SR = 1 - (2 / 10)$ 이 된다. SR의 값은 0.8이 나오며, 0에 가까운 수치가 나왔기 때문에 회복성의 품질은 높다.

5.7에서 정의된 바와 같이 PR은의 계산은 다음과 같다. WSM과 WSR에 각각 0.5씩 가중치를 주었다. 이 가중치는 본 논문에서 제시한 가중치이며, 시스템의 특징을 고려하여 품질 평가자가 다르게 가중치 값을 적용할 수 있다. 위에서 계산된 SM과 SR의 값들로부터,  $PR = (0.5 * 0.83) + (0.5 * 0.8)$ 이 된다. PR의 값은 0.815가 나왔고, 1에 가까운 수치가 나왔기 때문에 신뢰성의 품질평가는 높게 평가 되었다. 따라서, 유비쿼터스 컴퓨팅 시스템의 신뢰성은 81.5%가 된다는 의미이다.

**8.9 연결성의 계산 (PC)**

이 시나리오에서 A는 자동차 시동을 걸 때와, GPS 그리고 마트에서의 결제 시 다른 시스템들과의 연결 시 발생할 수 있는 실패를 고려하여 테스트를 하였다. 시스템들이 함께 운영되어 실행되는 테스트케이스의 수는 유비컴의 기능의 수와 같은 24개이고, 24개 중에 실패를

회피한 횟수는 22개이다. 2개의 기능은 실패를 회피하지 못하였다.

5.8에서 정의된 바와 같이 PC는 22 / 24가 되고, PC의 값은 0.916이 된다. 이는 유비컴이 다른 시스템들과의 연결성이 91.6%가 된다는 의미 이다.

**8.10 지각성의 계산 (SE)**

이 시나리오에서 A는 자동차를 타고 GPS의 도움을 받으며 회사에 출근을 했고 또, 마트에 들려 물건을 구입하였다. 이 외의 다른 환경에서 유비컴이 제공할 수 있는 서비스의 수는 8가지이다. 따라서 유비컴이 제공할 수 있는 서비스의 기능이 24가지이고, 다른 환경에서도 제공할 수 있는 서비스가 8가지이므로, 유비컴이 다른 가변적인 환경에서도 지능적으로 제공할 수 있는 총 서비스의 수는 32가지이다.

5.9에서 정의된 바와 같이 SE는 유비쿼터스 컴퓨팅 시스템이 사용자의 상황에 따라 개인 서비스가 다르고 변할 수 있기 때문에, 얼마나 사용자에게 지능적으로 다양하게 서비스를 제공할 수 있는가를 평가한다. 그러므로, SE는 24 / 32가 되고, SE의 값은 0.75이 된다. 이는 유비컴이 다른 어떠한 가변적인 환경에서도 지능적으로 사용자에게 제공해 줄 수 있는 서비스 확률은 75%가 된다는 의미이다.

**8.11 선호도의 계산 (PA)**

이 시나리오에서 A는 유비컴의 모양과 기능을 조금 변경하려고 한다. 사용자가 원하는 데로 변경할 수 있도록 유비컴이 제공하는 총 인터페이스의 수는 11가지 이고, 그 중에 A가 원하는데로 변경할 수 있는 인터페이스의 수는 총 9가지이다.

5.10에서 정의된 바와 같이 PA는 9 / 11이 되고, PA의 값은 0.818이 된다. 이 수치는 유비컴이 사용자가 원하는 데로 81.8% 커스터마이징 될 수 있다는 의미이다.

**8.12 유비쿼터스 컴퓨팅 시스템 품질평가**

QA<sub>i</sub>를 본 논문의 품질모델에 정의된 i번째 품질속성이라고 가정하자. i의 범위는 1부터 10이 될 것이다. 또한 품질속성과 관련된 중요도는 적용되는 시스템에 따라 달라질 수 있다. 따라서, 각 품질속성을 위해 가중치를 준다. QA<sub>i</sub>와 관련된 가중치를 Wi라고 하고 10개 품질속성의 가중치를 모두 더한 값은 1이 된다. 유비쿼터스 컴퓨팅 시스템 품질모델(UCQM)은 다음과 같이 계산될 수 있고, UCQM 값의 범위는 0부터 1사이의 값이 되며, 높은 값일수록 유비쿼터스 컴퓨팅 시스템의 품질이 좋다는 것을 의미한다.

$$UCQM = \sum_{i=1}^{10} W_i \cdot QA_i$$

다음 표 2는 본 논문에서 권장하는 각 품질속성 별 가중치로서 높음 / 낮음의 방법으로 표현하였다. 가중치

표 2 품질속성을 위한 가중치

품질속성	가중치(높음/낮음)
상호운용성(SI)	높음
보안성(PS)	높음
이해성(SU)	낮음
운용성(UO)	높음
적용성(HA)	높음
공존성(TE)	낮음
신뢰성(PR)	높음
연결성(PC)	높음
지각성(SE)	낮음
선호도(PA)	낮음

높음의 의미는 유비쿼터스 컴퓨팅 시스템을 평가하기 위한 가장 기본이 되는 항목을 의미한다. 즉, 일차평가 항목은 상호운용성, 보안성, 운용성, 적용성, 신뢰성, 연결성이다. 다음으로 이러한 일차평가 항목이 만족되었을 경우에만 이차평가가 이루어지게 된다. 이렇게 일차평가 항목 6개를 선택한 이유는 유비쿼터스 컴퓨팅 시스템의 환경을 기반으로 하여 품질항목을 선택하고, 매트릭을 적용해본 결과 유비쿼터스 컴퓨팅 시스템의 품질평가에 직접적인 영향을 주는 품질항목이다.

상호운용성, 보안성, 운용성, 적용성, 신뢰성, 연결성이 다른 품질속성들 보다 상대적으로 높은 가중치를 가진다. 왜냐하면, 유비쿼터스 컴퓨팅 시스템을 평가하기 위한 일차적인 요소가 되기 때문이다. 나머지 품질속성들은 이차적인 품질속성으로써 일차적인 요소들이 우선 만족되었을 경우에 남은 이해성, 공존성, 지각성, 선호도에 영향을 미친다. 그러나 이러한 가중치는 고정된 것이 아니라, 본 논문에서 제안한 UCQM을 통하여 유비쿼터스 컴퓨팅 시스템을 평가할 경우, 시스템의 특징을 고려하여 품질 평가자가 다르게 가중치 값을 적용할 수 있다.

따라서, 본 논문에서 제시한 사례연구를 UCQM에 적용하여 계산을 하면 다음과 같다. 품질속성의 가중치를 모두 더한 값은 1이 되기 때문에, 본 논문에서는 가중치를 다음과 같이 제시한다. 일차 평가대상인 상호운용성, 보안성, 운용성, 적용성, 신뢰성, 연결성 각각에 0.12의 가중치를 주었고, 이차 평가대상인 공존성, 지각성, 이해성, 선호도는 각각 0.07를 주었다. 이러한 가중치를 적용한 사례를 통하여 UCQM을 계산하면 다음의 표 3과 같다;

표 3에 의해 UCQM의 값은 0.85이며, 1에 가까울수록 품질이 높은 유비쿼터스 컴퓨팅 시스템이다. 따라서, 6장의 사례연구를 통하여 품질평가는 85%이 나왔다.

**9. 결론**

유비쿼터스 컴퓨팅 시스템은 앞으로 가까운 미래에

표 3 UCQM 계산 값

품질속성	메트릭값 × 기중치	합계
상호운용성(SI)	0.83 × 0.12	<b>0.099</b>
보안성(PS)	1 × 0.12	<b>0.12</b>
이해성(SU)	0.87 × 0.07	<b>0.06</b>
운용성(UO)	0.8 × 0.12	<b>0.096</b>
적응성(HA)	0.87 × 0.12	<b>0.104</b>
공존성(TE)	0.8 × 0.07	<b>0.056</b>
신뢰성(PR)	0.815 × 0.12	<b>0.097</b>
연결성(PC)	0.91 × 0.12	<b>0.109</b>
지각성(SE)	0.75 × 0.07	<b>0.052</b>
선호도(PA)	0.81 × 0.07	<b>0.056</b>
<b>총 합</b>		<b>0.85</b>

소비, 정치, 경제 사회, 문화, 전 분야에 많은 변화를 가져올 것이며, 이를 통해서 인간의 삶은 풍요로워 질 것이다. 또한 유비쿼터스 컴퓨팅 시스템은 개인화된 시스템으로써, 네트워크를 통하여 언제 어디서나 정보에 대한 서비스를 받을 수 있는 곳이다. 유비쿼터스 컴퓨팅 시스템은 언제 어디서 어떻게 사용하느냐에 따라 다른 서비스가 제공되며, 이제는 더 이상 가상현실이 아닌 우리의 삶 속에서 현실적으로 다가 올 것이다.

본 논문에서는 이러한 유비쿼터스 컴퓨팅 시스템에 대한 품질평가를 위해 유비쿼터스 컴퓨팅 시스템의 중요한 특징 8가지를 식별하였고, 식별된 특징을 기반으로 하여 10개의 품질속성을 도출하였다. 10가지의 품질속성 중 기능성, 사용성, 이식성은 ISO/IEC 9126으로부터 도출되었으며 이동성의 신뢰성, 연결성, 지각성은 새로이 제안되었다. 또한 이렇게 정의된 각 품질속성을 측정하기 위한 메트릭을 정의하였다. 또한, 사례연구를 통해 제안된 UCQM을 적용하여 유비쿼터스 컴퓨팅 시스템을 평가한 결과를 제시하였다. 제안된 UCQM을 사용하여 유비쿼터스 컴퓨팅 시스템을 더 효과적이고 보다 더 정확하게 평가할 수 있다.

**참 고 문 헌**

[1] Mark Weiser, "The Computer for the 21<sup>st</sup> Century," *Scientific American*, pp. 94-104, Vol. 265, No. 3, Sept. 1991.

[2] Friedemann Mattern, "The Vision and Technical Foundations of Ubiquitous Computing" *Novatica and Informatik/Informatique UPGRADE*, pp. 3-6, Vol. 2, No. 5, October 2001.

[3] Benjamin J. Alfonsi, "Privacy Debate Centers on Radio Frequency Identification," *IEEE SECURITY & PRIVACY*, p. 12, Vol. 2, March-April 2004.

[4] Software Engineering-Product Quality-Part 1: Quality Model. ISO/IEC 9126-1, June, 2001.

[5] Software Engineering-Product Quality-Part 2:

Internal Metrics. ISO/IEC TR 9126-2, July, 2003.

[6] Wuest B., et al., "Framework for middleware in Ubiquitous Computing Systems," *Proceedings of the 16<sup>th</sup> international Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 2262-2267, 2005.

[7] Kindberg., T., et al., "System Software for Ubiquitous Computing," *IEEE Pervasive Computing*, pp. 70-81, Vol. 1, January 2002.

[8] O'Neill., E., et al., "A Testbed for Evaluating Human interaction with Ubiquitous Computing Environments," *Proceedings of the First International Conference on Testbed and Research Infrastructures for the Development of NeTworks and Communities (TRIDENTCOM'05)*, pp. 60-69, 2005.

[9] Jung. H., et. al., "Measuring Software Product Quality: A Survey of ISO/IEC 9126," *IEEE SOFTWARE*, pp. 88-92, Vol. 21, No. 5, September/October 2004.

[10] Zhu H., et al., "Application of Hazard Analysis to Software Quality Modeling," *Proceedings of the 26<sup>th</sup> Annual International Computer Software and Applications Conference (COMPSAC'02)*, pp. 139-144, 2002.

[11] Eila Niemela, Juhani Latvakoski, "Survey of Requirements and Solutions for Ubiquitous Software," *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia MUM '04. (MUM 2004)*, pp. 71-78, Vol. 83, 2004.

[12] Jean Scholtz and Sunny Consolvo, "Toward a Framework for Evaluating Ubiquitous Computing Applications," *IEEE Pervasive Computing*, pp. 82-88, Vol. 3, No. 2, April-June 2004.

[13] Tim Kindberg and Armando Fox, "System Software for Ubiquitous Computing," *IEEE Pervasive Computing*, pp. 70-81, Vol. 1, January 2002.

[14] Mark Weiser, "Hot Topics: Ubiquitous Computing," *IEEE Computer*, pp. 71-72, Vol. 26, October 1993.

[15] Tatsuo Nakajima, Kaori Fujunami, Eiji Tokunaga, Hiroo Ishikawa, "Middleware Design Issues for Ubiquitous Computing," *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia MUM '04. (MUM 2004)*, pp. 55-62, Vol. 83, 2005.

[16] Simon Schubiger-Banz, Beat Hirsbrunner, "A Model for Software Configuration in Ubiquitous Computing Environments," *Pervasive 2002, LNCS 2414*, pp. 181-194, 2002.

[17] Roy Want, Trevor Pering, "System Challenges for Ubiquitous & Pervasive computes," *Proceedings of the 27th international conference on Software engineering. (IEEE ICSE 2005)*, pp. 9-14, 2005.

[18] Zahid Anwar, Jalal Al-Muhtadi, William Yurcik, Roy H. Campbell, "Plethora: A Framework for Converting Generic Applications to Run in a



- Ubiquitous Environment," *proceedings of Mobile and Ubiquitous Systems: Networking and Services, 2005. (ACM MobiQuitous 2005)*, pp. 192-201, 2005.
- [19] Guozhen Zhang, Man Lin, "A Framework of Social Interaction Support for Ubiquitous Learning," *proceedings of the 19<sup>th</sup> Advanced Information Networking and Applications, 2005. (IEEE AINA 2005)*, pp. 639-643, Vol.2, 2005.
- [20] Dulcinea Carvalho, Roy Campbell, Geneva Belford, and Bennis Mickunas, "Definition of a User Environment in a Ubiquitous System," *CoopIS/DOA/ODBASE 2003, LNCS 2888*, pp. 1151-1169, 2003.
- [21] Arturo Zambrano, Silvia Gordillo, and Ignacio Jaureguiberry, "Aspect-Based Adaptation for Ubiquitous Software," *Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954*, pp. 215-226, 2004.



오 상 현

2004년 건양대학교 정보전산학과 공학사  
2006년 숭실대학교 컴퓨터학과 공학석사  
2006년~현재 숭실대학교 컴퓨터학과 박사과정. 관심분야는 소프트웨어 품질, 테스트링, 유비쿼터스(Ubiquitous), 제품제열공학(PLE)



김 수 동

1984년 Northeast Missouri State University 전산학 학사. 1988년/1991년 The University of Iowa 전산학 석사/박사. 1991년~1993년 한국통신 연구개발단 선임연구원. 1994년~1995년 현대전자 소프트웨어연구소 책임연구원. 1995년 9월~현재 숭실대학교 컴퓨터학부 부교수. 관심분야는 객체지향 S/W공학, 컴포넌트 기반 개발 (CBD), 제품제열공학(PLE), 모델 기반 아키텍처(MDA)



류 성 열

1997년 아주대학교 컴퓨터학부 공학박사  
1997년 3월~1998년 3월 George Mason University 교환교수. 1981년 3월~현재 숭실대학교 컴퓨터학부 교수. 관심분야는 리엔지니어링, 소프트웨어 유지보수, 소프트웨어 재사용, 소프트웨어 재공학/역공학