

# 유비쿼터스 서비스 개발을 위한 컴포넌트 기반의 서비스 개발 프레임워크

## A component based framework for service development of ubiquitous healthcare

양원섭\*, 이건명\*, 김원재\*\*, 윤석중\*\*

\*충북대학교 전자계산학과

School of Electrical and Computer Engineering, Chungbuk National University

\*\*충북대학교 의과대학 비뇨기학 교실

Research Institute for Computer and Information Communication, Chungbuk National University

E-mail: soryu@ailab.cbnu.ac.kr

### 요 약

유비쿼터스 헬스케어는 단일화된 서비스가 아니라 다양한 기술들이 복합적으로 결합되어 운용되는 서비스이다. 따라서 서비스의 형태가 고정적이지 않고 매우 다양하게 나타난다. 하지만 실제로 차이가 발생하는 부분은 서비스의 구현에 관한 세부적 내용에서 나타나고, 서비스 운용을 위한 기본 구성요소에 있어서는 큰 차이가 없이 유사한 형태를 가진다. 그 결과 유비쿼터스 헬스케어 서비스 개발 과정에서는 실제 서비스의 구현 외의 통신과 데이터베이스의 이용, 메시지 전달과 같은 중복되는 항목에 대한 고려가 매번 이루어져야 한다. 이것은 개발 과정에 있어 불필요한 비용의 증가를 불러온다. 본 논문에서는 이와 같은 불필요한 비용을 감소시키며 서비스의 개발과 운용이 가능한 유비쿼터스 헬스케어 서비스의 제공을 위한 아키텍처와 서비스 개발을 위한 프레임워크를 제안한다. 제안하는 서비스 제공 아키텍처는 크게 사용자 단말, 유비쿼터스 헬스케어 서비스 센터, 외부 기관으로 구성된다. 서비스 개발 프레임워크는 서버와 클라이언트 프레임워크로 구분된다. 서비스 개발 프레임워크는 서비스를 제공하는 서버에서 필요한 유비쿼터스 헬스케어 서비스의 공통 구성요소를 가진다. 서비스의 개발을 위해 우선 프로세스에 대한 정의를 수행하고, 정의된 내용에 따라 필요한 코드 템플릿을 결합하여 서비스의 초기 형태를 만들어낸다. 여기에 각 서비스가 필요로 하는 세부 사항을 작성하는 것으로 서비스의 개발을 수행하게 된다. 제안된 서비스 제공 아키텍처와 서비스 개발 프레임워크를 실제 적용해보기 위해 전립선비대증 환자 진료를 위한 시스템을 설계하고 구현하였다.

**Key Words** : ubiquitous healthcare, u-healthcare service, service development framework

### 1. 서 론

유비쿼터스 기술이 계속해서 발전되고, 서비스 수요자들의 새로운 요구사항 등이 증대됨에 따라 유비쿼터스 컴퓨팅 기술과 기존의 서비스 기술이 결합하여 다양한 영역에서 새로운 기술이 등장하고 있다. 이러한 새로운 기술들로는 군사 시스템과 유비쿼터스 컴퓨팅 기술이 결합한 u-Military, 일반 가정애 적용된 u-Home 과 도시 전역에까지 적용 범위를 확장하는 u-City, 상점에 적용되는 u-Market, 의료 시스템에 적용된 u-Hospital 등의 다양한 시도가 이루어지고 있다.

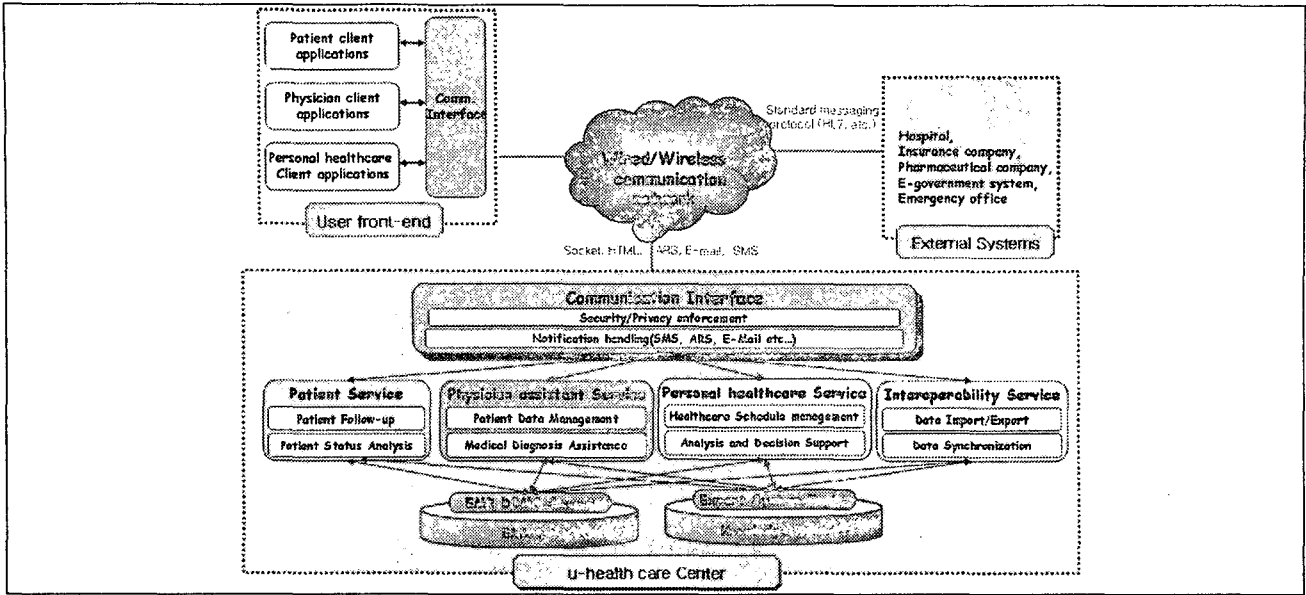
유비쿼터스 헬스케어(u-Healthcare : Ubiquitous Healthcare)도 이러한 새로운 기술영역의 하나로, 기존의 보건의료 서비스와 유비쿼터스 컴퓨팅 기술을 결합한 형태이다. 유비쿼터스 헬스케어는 시간과 장소의 제약 없이 사용가능한 네트워크화 되거나 직접 측정 및 진단 결과의 확인이 가능한 기기, 센서, 단말기 등을 활

용하여 질병에 대한 예방, 진단, 치료와 개인의 건강관리, 환자과 노약자에 대한 보호 및 관리 서비스, 재활복지 등의 서비스를 제공하는 것이다.[1]

유비쿼터스 헬스케어 서비스를 수행하기 위해서는 단일 기술이 아닌 네트워크, 센서 기술, 자료 처리 등의 다양한 분야의 기술들이 이용되어진다. 따라서 서비스의 형태가 매우 다양하고, 그 형태가 정형화되기 힘들다. 하지만 서비스의 실제 내부 구조로는 각 유비쿼터스 헬스케어 서비스들이 수행하는 세부적인 내용에 대해서만 차이를 가질 뿐 통신, 자료 처리 등의 기반이 되는 환경에 대한 공통요소들을 가지게 된다. 따라서 유비쿼터스 헬스케어 서비스들의 공통된 구성요소들에 대한 기본적인 개발·관리 환경을 제공한다면 서비스 개발 과정에서 개발자는 실제 서비스의 핵심 부분에만 집중하면 되므로 중복되는 내용들에 대한 설계와 구현 과정을 피할 수 있게 된다.

본 논문에서는 유비쿼터스 헬스케어 서비스에 대한 기본 구성요소를 정의하고, 각각의 요소를 만족할 수 있는 시스템 구성을 위한 유비쿼터스 헬스케어 서비스 시스템 아키텍처를 설계한다. 구성된 시스템 아키텍처에 맞추어 서비스 개발에 필요한 기능들을 선정하고 각

1) 이 연구는 지역산업융합기술개발 사업(보건의료 III 3-41)의 지원을 받아 수행된 것임.



[그림 1] 유비쿼터스 헬스케어 서비스 시스템 아키텍처

각의 기능들에 대한 코드 템플릿과 이를 결합하기 위한 프로세스의 정의 및 코드 생성을 수행하는 프레임워크를 설계한다. 설계된 프레임워크를 통해 전립선환자관리를 위한 시나리오를 통해 실제 서비스 개발에 적용하고 그 결과를 확인한다.

## 2. 유비쿼터스 헬스케어 서비스 시스템 아키텍처

서비스 아키텍처는 서비스 제공을 위해 구성된 시스템의 추상적인 구조도를 보여주게 된다. 서비스 아키텍처는 실제 서비스의 구현을 위한 물리적 설계, 서비스 모듈 설계 등 전체적인 서비스 시스템 구현에 이용하게 된다.

유비쿼터스 헬스케어 서비스에 대한 요구사항과 그에 따른 구성을 통해 설계된 서비스 아키텍처는 [그림 1]과 같다.

### 2.1 User front-end

사용자측에서는 환자용 클라이언트, 의사용 클라이언트, 개인 건강관리 클라이언트 응용 모듈을 통해서, 사용자로부터 또는 사용자에게 대한 정보를 수집하여 서버측의 해당 서비스에 정보를 전달하고, 결과 또는 서비스로부터의 메시지 또는 관리 명령을 전달받게 된다.

### 2.2 External Systems

단순한 질환 관리가 아닌 통합적인 보건의료 서비스를 제공하기 위해서는 서비스 수혜자로부터 제공받는 자료 외에도 외부 시스템들이 보유한 자료들을 필요로 한다. 이에 따라 외부의 헬스케어와 관련된 기관들(병원, 보험회사, 제약회사, 전자정부 시스템, 응급센터 등)

과의 신속하고 원활한 정보 교환 및 공유를 위해, 의료 정보교환 표준인 HL7[2]을 비롯한, PACS, OCS, EMR 시스템과의 연동을 위한 서비스를 제공한다. 이 기관들은 각각이 독립된 시스템이므로 서비스 시스템 아키텍처에서는 black box와 같이 추상적 형태로 취급한다.

### 2.3 u-Healthcare Center

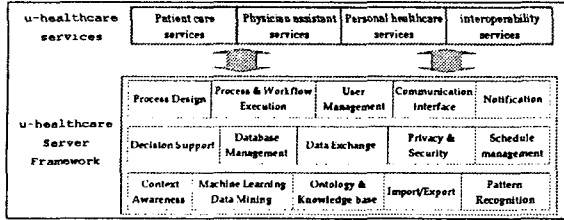
실제 유비쿼터스 헬스케어 서비스의 중심이 되는 서버로서 제공되는 서비스는 크게 환자에 대한 서비스(patient service), 의사에 대한 서비스(physician assistant service), 개인 건강관리 서비스(personal healthcare service), 외부시스템과의 정보공유 및 교환에 관련된 서비스(interoperability)로 구분된다.

## 3. 유비쿼터스 헬스케어 서비스 개발 프레임워크 설계

유비쿼터스 헬스케어 서비스를 수행하기 위해서는 단일 기능의 제공이 아닌 각각의 영역에 특화된 다수의 기능을 통합하여 제공하는 것이 필요하다. 예를 들어 혈압 관리를 수행하는 서비스를 제공하는 경우 이용자에 대한 기본 정보와 임상 정보 등에 대해 관리하는 기능, 이용자의 현재 상태를 파악하는 기능, 위험 상황을 판단하는 기능, 서비스 이용 중 각종 이벤트를 알려주는 기능을 기본으로 하여 여러 기능이 함께 작동해야 정상적인 서비스의 운용이 가능하게 된다. 이러한 기능들은 그 종류가 매우 세세해질 수 있으므로 각 기능의 공통 요소들을 기본으로 하여 프레임워크를 구성하게 된다. 이러한 요소들로 이용자의 관리, 상태 판단, 상황 인식, 지식 정보, 데이터 처리, 의사 결정, 통신, 보안, 일정 관리 등으로 구분이 가능하다.

### 3.1 서버측 서비스 개발 프레임워크

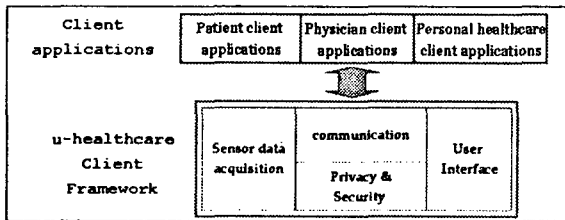
서버 측에 환자 서비스, 의사 서비스, 개인건강관리 서비스, 상호작업 서비스의 개발에 일반적으로 필요한 컴포넌트로서 다음의 [그림 2]와 같이 선정하고, 각각에 대한 요구사항을 정의한다.[3]



[그림 2] 서버측 헬스케어 서비스 개발 프레임워크

### 3.2 클라이언트측 개발 프레임워크

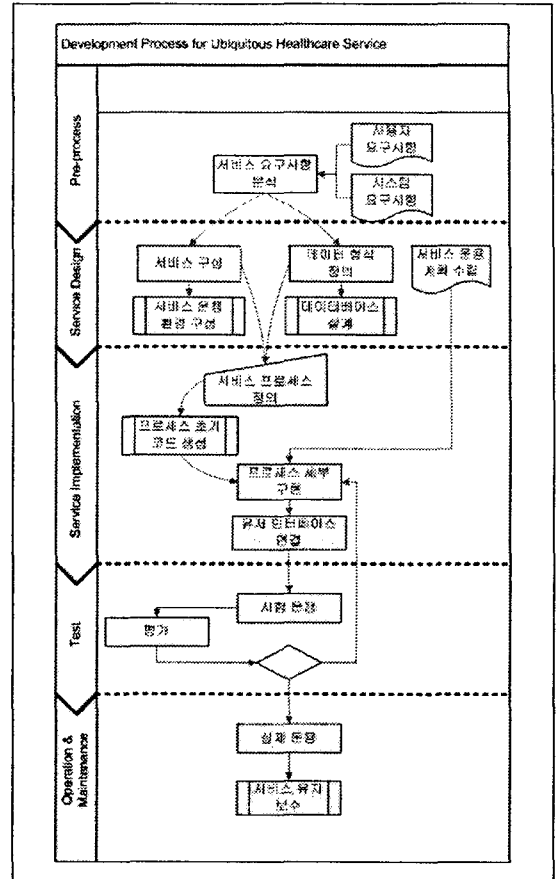
클라이언트측의 응용 모듈에 대한 소프트웨어 개발 프레임워크는 다음의 [그림 3]과 같은 구성을 가진다.[3]



[그림 3] 클라이언트측 개발 프레임워크

### 3.3 서비스 개발 프로세스

제안한 유비쿼터스 헬스케어 서비스 개발 프레임워크를 이용하여 실제 서비스를 개발하는 과정은 [그림 4]와 같다. 개발 단계는 크게 Pre-process, Service Design, Service Implementation, Test, Operation & Maintenance의 5단계로 구분된다.

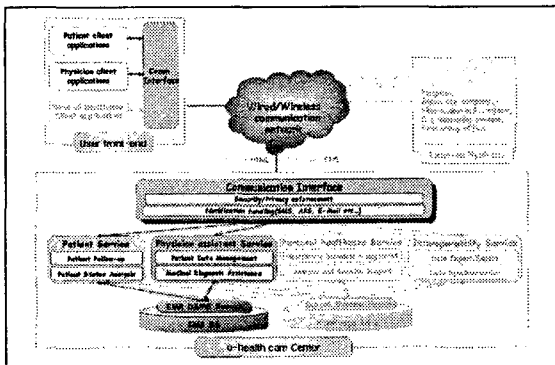


[그림 4] 서비스 개발 과정

- Pre-process : 서비스 개발의 첫 단계로 개발하려는 서비스에 대한 요구사항을 수집하고, 이를 분석하게 된다. 서비스 요구사항을 파악하기 위하여 사용자들의 요구사항과 시스템의 요구사항을 수집하고 이를 반영한다. 분석된 서비스 요구사항을 바탕으로 서비스를 디자인하게 된다.
- Service Design : 이전 단계에서 분석된 서비스 요구사항에 따라 서비스 개발을 위한 기본적인 설계 작업을 수행하게 된다. 서비스 구성 작업에서는 요구사항에서 필요로 하는 기능들을 제공하기 위해 필요한 모듈을 선택하고, 그 모듈들을 조합하여 시스템 아키텍처의 구조를 결정한다. 서비스의 구성이 완료되면 그 구조를 바탕으로 하여 실제 서비스가 동작할 하드웨어적인 설계를 수행할 수 있다. 데이터 형식 정의 작업은 서비스 요구사항에 맞추어 시스템에서 이용할 데이터의 항목을 결정하고, 각각의 단위와 범위 등을 정의하게 된다. 이 내용을 바탕으로 하여 시스템에서 이용할 데이터베이스의 기본 설계를 수행하게 된다. 데이터베이스의 설계에는 정의된 데이터 형식 외에도 시스템 운용을 위해 필요한 항목들이 추가되게 된다. 서비스 디자인에서는 서비스 운용을 위한 서비스 운용 계획을 수립하는 작업도 수행하게 된다. 서비스의 실제 운용을 위해서는 개인정보정책과 같은 문서적 장치도 필요하게 된다. 이를 시스템에 반영하기 위해서는 구현 단계에서부터 충분히 고려되어야 하므로 서비스 디자인 단계에서 운용 계획을 수립하는 것이 필요하다.

- **Service Implementation** : 이 단계에서는 실제 서비스 수행을 위한 구현을 수행하게 된다. 서비스 디자인 단계에서 수행된 서비스 구성과 데이터 형식의 정의를 통해 서비스를 수행할 각각의 프로세스들에 대한 정의(PDL)[4],[5]를 작성하게 된다. 작성된 프로세스 정의에 따라 서비스 개발 프레임워크는 프로세스에서 요구되는 기능들에 대한 컴포넌트들을 결합하여 초기 형태의 서비스 프로세스 코드를 생성하여 준다. 생성된 코드를 이용하여 개발자는 각 프로세스의 세부적인 기능 구현을 위한 코드를 작성하고, 유저 인터페이스와 연결하여 서비스 개발 단계를 완료하게 된다.
- **Test** : 서비스 구현이 완료되면 다음 단계로 서비스의 시험 운용을 수행하게 된다. 시험 운용을 거쳐 서비스의 요구사항에 대한 만족도와 시험 운용기간의 문제점, 개선점 등에 대한 평가를 수행하게 된다. 평가의 결과에 따라 수정 또는 개선이 필요한 경우 프로세스의 세부 구현 과정부터 평가까지를 반복하여 수행하게 된다. 평가의 결과가 충분히 만족스럽다면 다음 단계로 진행하게 된다.
- **Operation & Maintenance** : 이 단계는 개발된 서비스를 실제 운용하며 필요한 유지 보수를 수행하는 단계이다. 개발된 서비스를 준비된 서비스 시스템 환경에 설치하여 서비스를 제공하게 된다. 실제 서비스를 운용하며 필요에 따라 적절히 서비스에 대한 유지 보수를 수행한다.

#### 4. 전립선비대증 환자관리 시스템 구현

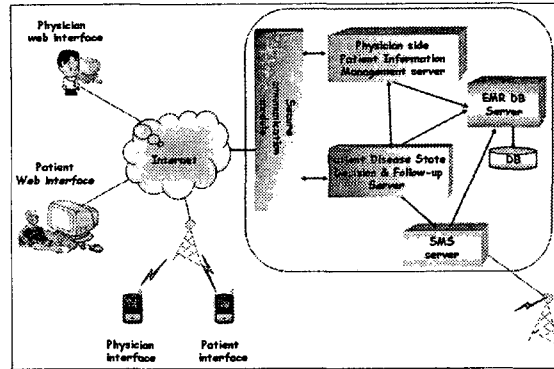


[그림 5] 구현에 사용된 아키텍처 구성요소

본 논문에서 제안한 서비스 아키텍처를 이용하여 구현된 전립선비대증 환자관리 시스템의 구성에 사용되는 시스템 아키텍처의 요소는 [그림 5]와 같다. 전립선비대증 환자관리 시스템은 환자와 전문의 이용자가 시스템을 이용하게 되고, 외부기관과의 자료 교환을 필요치 않는 범위에서 작동하게 된다.

전립선비대증 환자관리 시스템은 크게 나누어 유비쿼터스 헬스케어 센터의 역할을 수행하는 전립선비대증 환자관리 서버와 user front-end인 이용자 단말로 구성된다. 구현된 전립선비대증 환자관리 시스템의 구성은 [그림 6]과 같다.

서비스 구성에 따라 전문의와 환자 서비스 제공을 위



[그림 6] 전립선비대증 환자관리 시스템 구성도  
한 서비스 프로세스에 대한 정의를 수행하게 된다. 다음의 [표 1]은 전문의 서비스 프로세스 정의의 일부이다.

[표 1] 전문의 서비스 프로세스 정의

```

Process PhysicianService
Component::
    dc1:DataBaseConnector;
    s1:SMSNotify;
    ac1:AccountManager;
    dm1:DecisionMaker;

Event::
    addPhysician;
    addPatient;
    modifyUser;
    deleteUser;
    insertMedicalData;
    doDiagnosis;
    sendMessageToPatient;

Initial-Event-List::
    dc1.connectDataBase(#DB_Addr#, #DB_ID#,
#DB_Pass#, #DB_Name#);

Behavior::
    OnEvent(addPhysician) {
        ac1.addUser(dc1, #UserData#, "Phy");
    }
    .
    .
    .
    
```

각 서비스 프로세스의 정의가 완성되면 이를 이용하여 서비스의 초기 코드를 생성하게 된다. [그림 7]은 생성된 전문의 서비스 초기 코드를 보여준다.

생성된 서비스별 초기 코드는 기능 구현을 위해 필요한 초기화와 인스턴스의 생성, 데이터베이스의 이용 등의 기본적인 내용만 구현된 상태이다. 이를 바탕으로 실제 서비스 제공을 위한 구현을 수행하게 된다.

```

294
295 // This service is to be support phys. user.
296 class PhysicianService
297 {
298     private s: I;
299     private d: I;
300     private m: I;
301     private n: I;
302
303     class PhysicianService()
304     {
305         //Generated Code
306         s := new AccountManager();
307         d := new DataBaseConnector();
308         m := new DecisionMaker();
309         n := new SMSNotifier();
310
311         s := connectDataBase(~/DB_Addr~/,~/DB_ID~/,~/DB_Pass~/,~/DB_Name~/);
312
313         //Extra Code Begin
314     }
315
316
317
318     function addPhysician(~/Parameter~/)
319     {
320         //Generated Code
321         s := s.addUser(s,~/UserData~/,~/P~/);
322
323         //Extra Code Begin
324     }
325
326     function addPatient(~/Parameter~/)
327     {
328         //Generated Code
329         s := s.addUser(s,~/UserData~/,~/P~/);
330         m := m.addMD(s,~/MedicalData~/,~/UserData~/);
331
332         //Extra Code Begin
333     }
334
335
336     function modifyUser(~/Parameter~/)
337     {
338         //Generated Code
339         s := s.updateUser(s,~/UserData~/,~/UserData~/);
340
341         //Extra Code Begin
342     }
343
344

```

[그림 7] 전문의 서비스 프로세스 초기 코드 생성 예

### 5. 결론 및 향후 연구

유비쿼터스 헬스케어 서비스는 단일 기술로 구성되지 않고 다양한 기술들이 통합되어 작동하기 때문에 그 형태가 정형적이지 않고 매우 다양하게 나타난다. 하지만 실제 서비스를 수행하기 위한 부분에서만 큰 차이를 보일뿐 통신, 데이터베이스연결, 보안 제공 등의 서비스 기본 요소에서는 세부적인 구현의 내용만 다를 뿐 유사한 형태를 보인다. 따라서 서비스 개발 과정에서 이러한 공통되는 기본 요소들을 제공받을 수 있다면 서비스 개발자는 실제 서비스 수행에 관한 로직에만 집중할 수 있게 되어 결과적으로는 개발 비용의 절감을 가져오게 된다.

본 논문에서는 새롭게 등장한 서비스의 한가지인 유비쿼터스 헬스케어 서비스 개발을 위한 컴포넌트 기반의 프레임워크에 대한 내용을 소개하였다. 서비스 제공을 위한 시스템 아키텍처로는 수행하는 기능적 관점에서 이용자 측 단말, 서비스 제공 서버, 외부 기관으로 분류하였고 각 분류별로 수행할 기능들을 정의하였다. 서비스 개발 프레임워크는 서버 측의 서비스 개발과 클라이언트 측의 어플리케이션 개발을 각각 분류하여 유비쿼터스 헬스케어 서비스가 필요로 하는 공통되는 기본 요소들에 대한 설계를 제공하였다. 서비스 개발 프레임워크는 PDL을 이용하여 서비스 수행을 위한 프로세스를 정의하고, 이를 바탕으로 하여 미리 준비된 코드 템플릿을 조합하는 방식으로 서비스 수행에 필요한 기본 컴포넌트를 형성한다. 여기에 개발자는 서비스 수행을 위한 세부적인 코드를 작성함으로써 서비스 컴포넌트의 개발을 수행할 수 있게 된다.

향후 과제로는 유비쿼터스 헬스케어 서비스의 규모가 거대해지면 단일 서버가 아닌 분산된 서버를 통해 서비스가 이루어져야 한다. 따라서 시스템 아키텍처가 분산 환경에서 효과적인 서비스 제공을 위한 고려가 이루어져야 한다. 서비스 개발 프레임워크는 개발 과정에 더 유연하게 적용이 가능하도록 하기 위해 더 다양한 언어별 코드 템플릿을 갖추고, 이를 관리하기 위한 환

경이 필요하다. 또한 프로세스 정의와 서비스 세부사항 구현, 인터페이스 설계 등을 위한 GUI 환경의 편집기와 같은 개발자 환경에 대한 연구가 필요하다.

### 참 고 문 헌

- [1] 한국전산원 정보화기획단 u-전략팀, "유비쿼터스 사회, 새로운 희망과 도전", 한국전산원, pp225-240, 2006
- [2] Health Level Seven, <http://www.hl7.org>
- [3] 양원섭, 이진명, 김원재, 윤석중, "u-Hospital을 위한 u-Healthcare System Framework 설계", 한국퍼지 및 지능시스템학회 16권 1호, 2006년도 춘계학술대회 학술발표논문집, pp297-301, 2006.
- [4] Inoue, K., Ogihara, T., Kikuno, T. and Torii, K., "A Formal Adaptation Method for Process Descriptions", Proc. of 11th ICSE, Pittsburgh, PA, pp.145-153, May, 1989.
- [5] Inoue, K., Ogihara, T., Iida, H., Nitta, M., and Torii, K., "Functional Language for Enacting Software Processes", Proc. IEEE COMPSAC-91, Tokyo, Japan, pp.219-224, September 1991.