

홈 오토메이션을 위한 지능형 사용자 프로파일 관리 시스템

구현

조용곤^o, 이승훈, 신동규, 신동일
세종대학교 컴퓨터공학과

{ygon21c^o, shlee}@gce.sejong.ac.kr, {shindk, dshin}@sejong.ac.kr

Implementation of a User Profile Management System for Home Automation

Yonggon Cho^o, Seunghun Lee, Dongkyoo Shin, Dongil Shin
Sejong University Department of Computer Science

요약

최근 자동화 시스템의 연구가 활발히 진행되고 있다. 그중에서도 홈오토메이션의 가장 큰 특징은 전자·자동기기 등이 추가 되는 가사의 관리를 자동화 하는 것이다. 사용자의 현재 상황 및 선호도 등의 정보를 이용하여 서비스를 사용자에게 제공하는 개인화 서비스는 사용자 프로파일에 따라 차별화 되어서 제공되어진다. 하지만 현실상의 사용자 프로파일을 수집하기에는 어려움이 따르기 때문에 본 논문에서는 개인화 서비스를 효과적으로 지원할 수 있도록 가상의 사용자 프로파일을 다루는 사용자 프로파일 생성기 데이터베이스를 설계 및 사용자프로파일 생성기를 구현하였다.

1. 서론

홈 오토메이션이란 전자·자동기기 등이 추가 되는 가사의 관리를 자동화하는 시스템으로 가정·가사 생활의 자동화를 말한다[1][2]. 사용자의 편의를 위한 홈 오토메이션의 기능은 사용자의 홈서비스 사용 패턴이 사용자 프로파일로 정의되어 있어야 개인화된 서비스를 제공한다. 개인화 서비스란 사용자의 현재 상황, 행동, 성향, 선호도 등의 정보를 이용하여 사용자에게 적용화 된 서비스를 제공하는 것을 말한다[3][4][5]. 이러한 개인화 작업에서 사용자의 선호를 예측하기 위한 개별 고객의 계산모델 구축이 가장 중요한 이슈가 되고 있다[6]. 또한 사용자 프로파일에 따라 달라지는 개인화된 서비스의 성능을 높여야 한다. 그렇기 때문에 사용자 프로파일을 생성하는 것은 중요한 작업이라 할 수 있다. 하지만 현실에서는 홈오토메이션 실험을 위해 사용자 프로파일을 수집하기에는 조금의 어려움이 따른다. 본 논문에서는 상황인식 기술에 관한연구와 사용자 프로파일 관리기술을 설명하고, 현실에서는 얻어내기 어려운 사용자 프로파일을 가상으로 만들어서 사용자의 홈서비스 사용 패턴을 예측해서 서비스를 실험하기 위해 사용자 프로파일을 생성할 수 있는 생성기를 설계 및 구현한다.

2장에서는 사용자가 가장 원하는 정보 서비스를 개인화 프로파일을 처리하여 알아내는 상황인식 기술과 그에 관한 기술들에 대해 알아본다. 3장에서는 서비스 상황 데이터베이스설계 및 구현 방법에 대하여 설명한다. 마지

막으로 결론에서는 본 논문에서 제시한 사용자 프로파일 생성기를 통해 얻어낼 수 있는 정보를 바탕으로 서비스의 발전 방향을 제시한다.

2. 관련연구

본 논문에서 제시되는 홈 오토메이션 기술은 휴대형 휴먼인터페이스 터미널에서 취득된 멀티모달 생체 정보(ECG, 체온, 얼굴표정)등을 종합하여 사용자의 현재 감정 상태를 파악하고, 현재 사용자가 가장 원하는 정보 서비스를 개인화 프로파일을 처리하여 알아내는 기술이다. 사용자에게 가장 적절한 서비스는 사용자가 현재 사용하고 있거나 가까이 위치한 정보가전기기를 통해서 제공되며 이러한 여러 가지 정보 가전기기에 적합하도록 순간적으로 콘텐츠와 UI(User Interface)를 조절하는 기능을 수반한다.

2.1 상황인식 기술에 관한연구

현재까지 진행된 상황인식 기술은 첫째로, 커뮤니티 정보 관리를 지원하기 위한 개인정보 관리 기술, 비즈니스 정책과 목적을 지원하기 위한 기술과 같은 서비스 지원 측면의 기술이 있다. 둘째로, 홈 네트워크의 자율적 시스템 구성을 위한 미들웨어, 유비쿼터스 환경을 위한 응용프로그램 개발을 지원하는 미들웨어에 관련된 기술이 있다. 셋째로, 사용자에게 상황인식 서비스를 제공할 때 환경의 간섭을 최소화하는 것을 목표로 하는 기법,

본 연구는 서울시 산학연 협력사업(JP091000)의 지원에 의해 수행되었음.

넷째로, 새로운 기술이 등장했을 때 기존에 구현된 시스템에 쉽게 적용시킬 수 없다는 문제를 해결하기 위해 연구 중인 기법이 있다. 다음은 각각에 해당하는 대표적인 프로젝트와 기법들에 대해 알아보도록 하겠다.

(1) Community Computing

개인정보 관리를 확장하여 loosely coupled 되고, semi-trustful한 커뮤니티의 정보 관리를 시스템 차원의 지원을 위해 Microsoft가 제시한 솔루션 개념이다. 커뮤니티 구성, 멤버십 관리, 커뮤니티 발견과 가입, 정보 저장소, 정보 보호, 정보 구조화, 정보 생명주기 등을 중점 연구 중이다(아직 관련 Publication은 없음). 커뮤니티에서 집합적으로 구조화된 정보를 탐색하기 위해 사회 이론(Social Theory)을 적용하여 필요한 데이터 마이닝 및 정보 시각화 방법에 대한 연구이다. 대화 및 토론 환경에 초점을 맞춘 사회 가상공간(Social Cyberspace)에 대한 데이터 수집, 탐색 기법이다. 커뮤니티 내 상대방을 이해하고 협력적으로 상호작용할 수 있는 상호작용 도구, 인터페이스, 시각화 기법에 대한 디자인, 프로토타입 기법 연구이다. AURA(Advanced User Resource Annotation system), SLAM(Social Location Annotation Mobile), Netscan, SNARF(Social Network and Relationship Finder) 등의 세부 프로젝트 진행 중이다.

(2) Autonomic Computing

IBM을 중심으로 비즈니스 정책과 목적에 따라 변화에 동적으로 적응하기 위한 자가관리기술 이니셔티브이다. IBM은 2006년 오토노믹 컴퓨팅 아키텍처를 ITSM 솔루션(IT 관리와 연관된 프로세스를 정의하고 모델링하는 IT 관리솔루션에 적용. ITSM에서 오토노믹 컴퓨팅 아키텍처는 표준화된 관리 인터페이스와 데이터 포맷을 이용하여 IT 인프라를 위한 관리기능 제공한다.

2007.6월 개최된 ICAC 2007에서는 Service-Level Agreements, Policies, Modeling, Learning, Provisioning, Scheduling, Protection, Repair 등의 분야에 심화 연구이다.

(3) Amigo

네트워크로 연결된 홈을 위한 자율적인 서비스 지원을 위하여 Philips, Microsoft, France Telecom, Fraunhofer 등 15개 산학연 컨소시엄의 구성이다.

홈의 자율적 시스템 구성을 위하여 개방적이고 표준화된 미들웨어를 개발한다.

Home Care, Security, Home Information, Entertainment 등의 프로토타입 서비스를 개발한다.

Context Management Service를 통하여 컨텍스트들의 효율적인 관리에 대한 연구를 진행하나 상황에 대한 명확한 판단 기준 마련이 미흡하다.

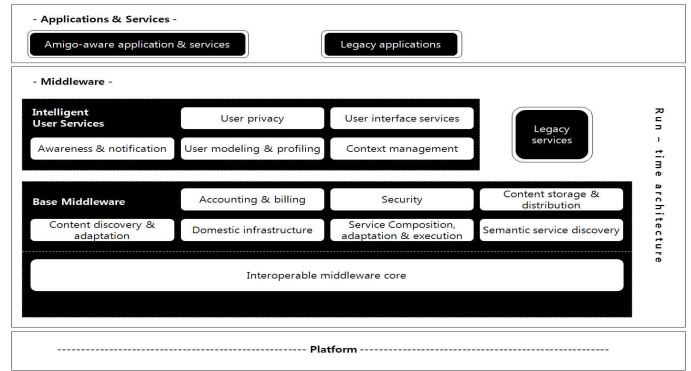


그림 1. Amigo 아키텍처

(4) Gaia

Gaia는 물리적 공간과 디지털 공간이 융합된 유비쿼터스 환경 (Active Space라 칭함) 을 위한 응용 프로그램을 개발, 실행하고 상황 변화에 따라 재구성할 수 있도록 돕는 소프트웨어 인프라스트럭처, 즉 미들웨어의 개발을 목표로 하고 있다. Gaia는 기본적인 컴포넌트 관리 및 서비스를 제공하는 Gaia 커널, 응용 프로그램 개발을 돕기 위한 응용 프로그램 프레임워크 (application framework) 로 구성되어 있다. Gaia의 전체 시스템 구조 및 내부 컴포넌트는 그림2와 같다.

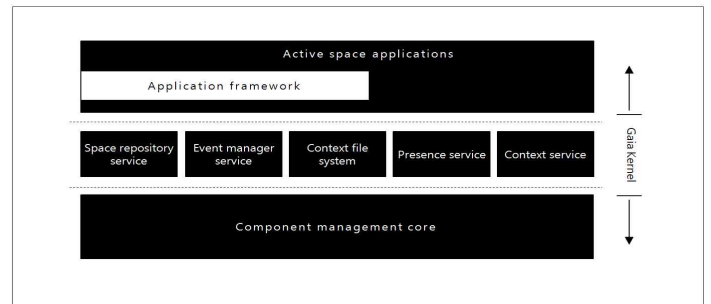


그림 2. Gaia 시스템 구조

(5) Aura

Aura는 사용자의 컨텍스트와 필요성에 적응하는 proactive하고 self-tuning하는 컴퓨팅 환경을 만들어 사용자에 대한 환경에 의한 간섭을 최소화하는 것을 목표로 하고 있다. Aura의 전체 시스템 구조 및 내부 컴포넌트는 그림3과 같다.

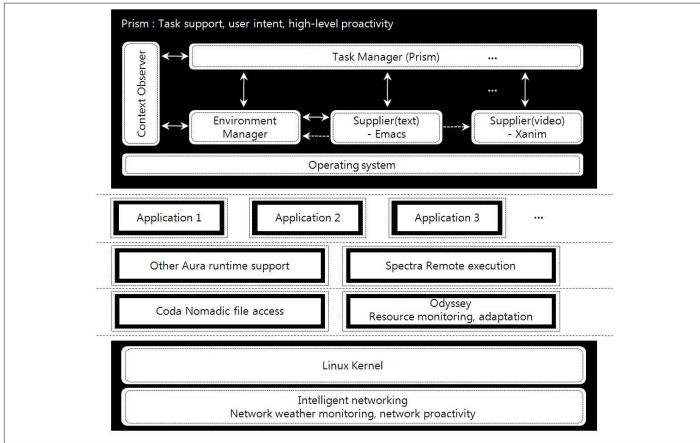


그림 3. Aura 시스템 구조

(6) Gator tech Smart Space

Gator tech에서는 기존의 유비쿼터스 시스템에서는 새로운 기술이 등장했을 때 지원하기 힘든 점을 해결하기 위해 진화 가능한 Programmable pervasive space를 만들고자 한다. 이 Smart Space는 Runtime 환경과 소프트웨어 라이브러리를 모두 가지고 있으며 Service discovery와 Gateway protocol이 시스템 요소들을 스스로 통합하여 관리한다. Smart Space의 시스템 구조는 다음과 같다.

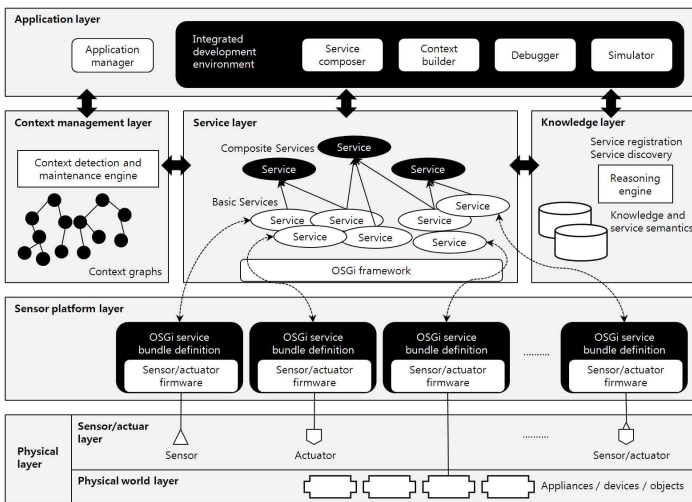


그림 4. Smart Space 미들웨어 구조

2.2 사용자 프로파일 관리기술

프로파일 관리 기술을 크게 3가지로 나누면 크게 정보 수집, 프로파일 활용, 그리고 프로파일 작성 및 저장 기

술로 나누어진다 이들 기술의 활용 방법 및 특징은 다음과 같다.

(1) 정보 수집 기술

정보 수집 기술은 개인화된 서비스를 제공하기 위해 필요한 정보를 수집하는 기술을 일컫는다. 이때 수집되는 정보로는 사용자 정보, 네트워크 정보 등이 있다. 일반적으로 사용되는 정보 수집 방법은 사용자의 서비스 이용도 통계 분석, 사용자의 직접 입력을 통한 방법 등이 있다.

(2) 프로파일의 활용

프로파일의 활용은 프로파일로 작성된 정보를 이용하여 서비스 적응화 또는 사용자 성향 분석을 위해 활용되는 것을 통칭한다. 현재 프로파일의 활용은 프로파일 정보의 단편성과 낮은 기술 수준의 프로파일 활용으로 인해 극히 한정되어 있다.

(3) 프로파일 작성 및 저장 기술

프로파일 작성 기술은 사용자의 기본정보, 사용자 선호도 등의 사용자와 관련된 정보를 일정한 방법으로 작성하는 기술을 말한다.

3 설계 및 구현

3.1 상황 데이터 베이스 설계

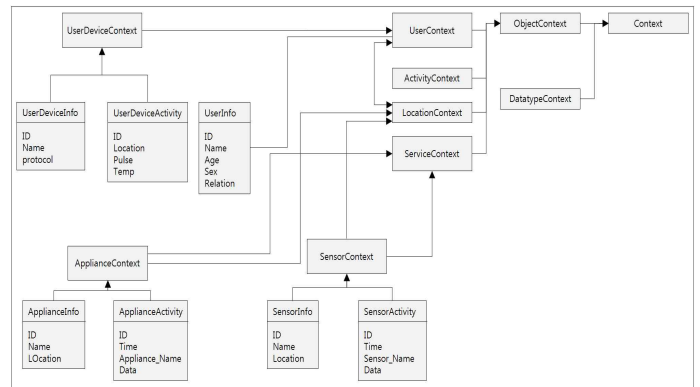


그림 5. 상황 데이터베이스 클래스 다이어그램

서비스 상황 데이터베이스는 여러 이질적인 서비스들을 포괄할 수 있게 서비스를 하나의 일반화된 형태로 모델링하여 개념화 시킨 구조체로써 하나의 서비스가 가지고 있는 공통적인 속성이나 특성을 개념화시켜 표현한다. 홈네트워크 환경에 존재하는 이질적인 서비스들을 일반화시켜 모델링한 서비스 상황 데이터베이스는 대체 서비스 검색을 지원하게 해주는 핵심적인 역할을 담당하게 된다. 이를 통하여 서비스 검색 시스템은 사용자의 요구사항에 완벽히 일치하는 서비스뿐 아니라 그와 유사한 서비스를 검색하여 주게 된다. 위 그림 5는 본 시스템에 사용되는 서비스 상황 데이터베이스를 클래스 다이

어그럼으로 나타낸 것이다. 우선적으로 홈 네트워크 환경에 사용되는 서비스들이 가지는 속성이나 특성을 파악하여 그에 따른 서비스의 공통적인 속성들이 서비스 상황 데이터베이스에 나타내도록 하여 일반화된 서비스 개념화 시켰다. 뿐만 아니라 각 서비스마다 가지는 고유한 속성이나 정보들이 동적으로 추가될 수 있게 하여 이질적이고 서로 다른 특성을 가진 서비스들이 표현되는데 문제가 없도록 하였다. 가령 하나의 서비스가 가지는 함수를 표현할 때에도 여러 개의 함수가 포함할 수 있도록 할 뿐 아니라 그 함수 안에서 여러 개의 효과가 나타날 수 있게 하여 유연한 서비스 모델링이 가능하도록 설계하였다. 이를 통하여 좀 더 확장성(Scalable)이고 어느 상황에서도 사용할 수 있는 유연한 서비스 검색 시스템을 구축하는데 도움이 되도록 하였다.

그림 5에서 나타난 바와 같이 본 서비스 상황 데이터베이스의 구조는 서비스(Resource) 클래스가 중심이 되어 그 안에 나타나는 속성들을 표현하기 위한 사용자 정의 클래스들이 표현되어 있고 이러한 사용자 정의 클래스들이 서비스 클래스와 관계를 이루는 것을 확인할 수 있다. 서비스 클래스 안에는 서비스 타입(service Type), 위치(Location), 메소드(method) 속성 이외에 다른 여러 속성이 정의된 것을 확인할 수 있다. 또한 서비스에 대한 정보를 더욱더 세분화시켜 표현할 수 있어 대체 서비스 검색뿐 아니라 더욱더 정교하고(fine-grained)자세한 검색을 본 시스템에서 지원할 수 있게 되었다. 특별히 서비스의 위치 속성을 통하여 한 서비스의 이동성을 고려하게 해주고 다중 지능공간에서의 서비스 검색을 가능하게 함으로써 홈네트워크 환경하에서 본 논문의 서비스 검색 시스템이 작동하는데 큰 어려움이 없도록 하였다. 본 논문에서 설계한 온톨로지에는 서비스를 정의하는 Resource 클래스가 중심이 되어 이 클래스 안에 있는 속성들을 표현하기 위해 정의된 여타 다른 클래스가 관계를 이루어 하나의 일반화된 서비스를 개념화시켜 준다.

3.2 사용자 프로파일 생성기 구현

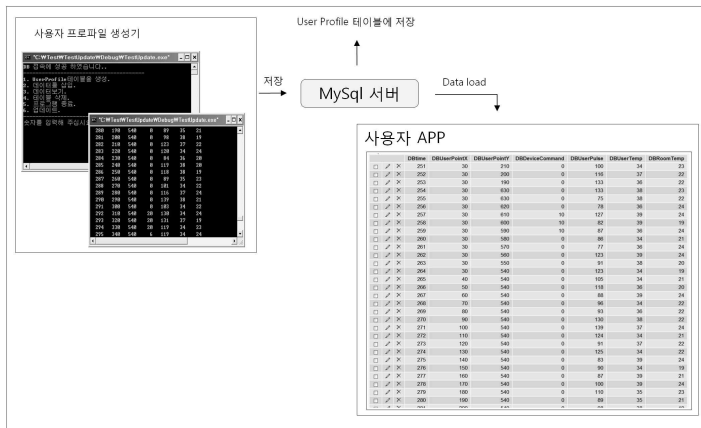


그림 6. 사용자 프로파일 생성기

그림 6은 사용자 프로파일생성기에서 생성된 데이터

가 어떤 방식으로 생성되고 데이터베이스 서버에 저장되며 저장된 데이터를 사용자가 확인 할 수 있는지를 보여 준다. 사용자 프로파일생성기에서 생성된 데이터는 기본적으로 User 프로파일 테이블에 저장된다. 이제 임의로 생성된 데이터는 어떤 것이 있고, 어떤 방법으로 생성되었는지를 살펴보겠다. 임의로 생성된 데이터는 임의의 컨텍스트 데이터라 할 수 있는데 그 속성으로는 시스템 시간, 사용자위치, 사용자 맥박, 사용자 체온, 실내온도, 사용자의 가전기기 컨트롤 명령이 오게 된다. 각 컨텍스트 데이터는 사용자의 가전기기 컨트롤 명령 속성을 제외하고 특정 범위의 데이터를 0.1~0.9의 9가지 값으로 정규화 한다. 맥박은 휴식기에 성인의 평균 맥박이 72~140인 것을 감안하여 총 9개의 값으로 정규화 된다. 체온은 보통 34~39도 임을 감안하여 9단계로 정규화 된다. 실내온도의 경우 보통사람이 쾌적하게 느끼는 19~22도를 감안하여 9단계로 정규화 된다. 정규화 되지 않는 값인 사용자의 가전기기 컨트롤 명령은 다음과 같이 구분하였다. 가스제어는 101, 현관문은 102, 화장실등은 103, 냉장고는 104, 환풍기는 105, TV는 106, 거실등1은 107, 방전원1은 108, 방등1은 109, 에어컨은 110, 거실등2는 111, 방전원2는 112, 방등2는 113, 커튼Close는 114, 커튼Stop은 115, 커튼Open은116으로 각 세부사항을 구분하였다.

3.3 사용자 프로파일 생성기 주요 소스코드

```

1 #define SOCKET int
2
3 #include <stdio.h>
4 #include <time.h>
5 #include <string.h>
6 #include <mysql.h>
7
8 ////////////////////////////////////////////////////
9 #pragma comment(lib, "libmysql.lib")
10 #pragma comment(lib, "ws2_32.lib")
11
12 #define DB_HOST "localhost"
13 #define DB_USER "shlee"
14 #define DB_PASS "1234"
15 #define DB_NAME "shlee"
16
17 #define SQL_CREATE_TABLE "CREATE TABLE User_Profile(DBtime INT NOT NULL AUTO_INCREMENT PRIMARY KEY, DBUserPointX INT NULL, DBUserPointY INT NULL, DBDeviceCommand INT NULL, DBUserPulse INT NULL, DBUserTemp INT NULL, DBRoomTemp INT NULL) TYPE = MYISAM ;"
18
19 #define SQL_INSERT_RECORD "INSERT INTO User_Profile(DBtime , DBUserPointX, DBUserPointY, DBDeviceCommand, DBUserPulse, DBUserTemp, DBRoomTemp ) VALUES(NULL, %d, %d, %d, %d, %d, %d);"
20
21 #define SQL_SELECT_RECORD "SELECT * FROM User_Profile"
22 #define SQL_DROP_TABLE "DROP TABLE User_Profile"
23 ////////////////////////////////////////////////////
24
25 void contextData();
26 void createTable();
27 void insertRecord(int insertCount);
28 int selectRecord();
29 void updateRecord();
30 void dropTable();
31 void programExit();
32 void menu();
33
34 MYSQL *connection=NULL, conn;
35 MYSQL_RES *sql_result;
36 MYSQL_ROW sql_row;
37 int query_stat;
38 char query[255];
39
40 int main(void)
41 {

```

그림 7. 사용자 프로파일 생성기 소스코드 -1

그림 7은 사용자 프로파일 생성기의 소스코드 일부로 MySQL용 ODBC 드라이버를 설치하지 않고 직접라이브러리를 이용하는 기법을 사용하고 있다. 위 소스코드처럼 구현함으로써 MySQL Database의 위치에 상관없이 MySQL서버의 접근 권한만 설정되어 있으면 사용자 프로파일 생성기 소스코드에서 생성된 사용자 프로파일은 MySQL서버 데이터베이스에 저장될 수 있다.

```

44 int main(void)
45 {
46     srand((unsigned)time(NULL));
47     mysql_init(&conn);
48
49     // DB 연결
50     connection = mysql_real_connect(&conn, DB_HOST,
51     DB_USER, DB_PASS, DB_NAME, 0, (char *)NULL, 0);
52     if(connection==NULL)
53     {
54         fprintf(stderr, "mysql connection error: %s", mysql_error(&conn));
55         return 1;
56     }
57     printf("DB 접속에 성공하였습니다.\n");
58
59     while(1){
60         menu();
61     }
62     return 0;
63 }
64
65 void menu() {
66     // 메뉴 출력
67     printf("1. insertRecord\n");
68     printf("2. selectRecord\n");
69     printf("3. updateRecord\n");
70     printf("4. createTable\n");
71     printf("5. dropTable\n");
72     printf("6. programExit\n");
73     printf("7. randomvalue\n");
74 }
75
76 void insertRecord(int insertCount){
77     // 레코드 삽입
78     for(i=1; i<=insertCount; i++){
79         contextData(); //난수 생성 (위치, 맥박, 체온, 실내)
80         sprintf(query, SQL_INSERT_RECORD, user_point_x, user_point_y, device_command, user_pulse, user_temp, room_temp);
81         query_stat = mysql_query(connection, query);
82         if (query_stat != 0)
83         {
84             fprintf(stderr, "mysql query error: %s", mysql_error(&conn));
85             return 1;
86         }
87         printf("레코드가 삽입 되었습니다.\n");
88     }
89 }
90
91 int selectRecord() {
92     // 레코드 선택
93     printf("레코드를 선택합니다.\n");
94 }
95
96 void updateRecord() {
97     // 레코드 업데이트
98     printf("레코드를 업데이트합니다.\n");
99 }
100
101 void dropTable() {
102     // 테이블 삭제
103     printf("테이블을 삭제합니다.\n");
104 }
105
106 void programExit() {
107     // 프로그램 종료
108     printf("프로그램을 종료합니다.\n");
109 }
110
111 void randomvalue() {
112     // 랜덤 값 생성
113     printf("랜덤 값을 생성합니다.\n");
114 }

```

그림 8. 사용자 프로파일 생성기 소스코드 -2

그림 8는 사용자 프로파일 생성기 소스코드의 일부로 insertRecord함수는 contextData함수에서 생성된 사용자 프로파일 데이터를 MySQL서버의 데이터베이스에 저장해주는 기능을 담당하고 있다.

3.4 생성된 프로파일과 사용자와의 수용

생성된 프로파일 속성에는 시스템시간, 사용자위치, 사용자 맥박, 사용자 체온, 실내온도, 사용자의 가전기기 컨트롤 명령이 오게 되는데 이러한 속성들을 20대 대학생 20명을 대상으로 프로파일 속성들을 실제와 비교하였다. 다음 표1과 표2는 그 결과이다.

표. 1 입구위치에서의 실제 데이터

실내온도	맥박	체온	가전기기
27	94~112	36~36.8	110
26	88~111	36.2~36.7	110
25	84~108	36~36.5	107
24	78~109	36.1~36.7	107
23	80~94	36.2~36.6	107

표. 2 입구위치에서의 생성된 프로파일

실내온도	맥박	체온	가전기기
27	88~132	34~37	110
26	85~127	34~36.8	110
25	79~128	35~36.7	110
24	78~129	35~37.7	107
23	76~124	36.2~36.6	107

이를 통해 실제 데이터와 생성된 프로파일과의 차이가 그리 크지 않다는 것을 알게 되었다.

4 결론 및 향후연구

지능형 홈 오토메이션 시스템에 대한 실험을 하기 위해서, 사용자 프로파일을 사용한 개인화 서비스 실험

이 필수적이다. 하지만 매년 실제 환경과 똑같은 상황에서 사용자 프로파일을 수집하여 실험하기에는 많은 어려움이 따른다. 그래서 본 논문은 이 문제점을 해결하기 위해 홈 오토메이션 시스템 사용자에게 개인화된 서비스를 제공하기 위한 사용자 프로파일의 관리 기법에 대해서 알아보고, 효율적인 홈 서비스를 위해 상황 데이터베이스의 설계 및 사용자프로파일 생성기를 구현 하였다.

본 논문에서 설계한 상황 데이터 베이스는 서비스 클래스가 중심이 되어 그 안에 나타나는 속성들을 표현하고 서비스 클래스와 관계를 이루고 있는 사용자 정의 클래스들이 있다. 그러한 클래스들은 서비스 타입, 위치, 매소드 속성 이외의 다른 여러 속성이 정의 되어 있어 서비스에 대한 정보를 더욱 세분화 시켜 표현할 수 있으며 대체 서비스 검색뿐 아니라 정교하고 자세한 검색을 지원할 수 있다. 구현된 사용자 프로파일 생성기는 그 속성으로 시스템 시간, 사용자위치, 사용자 맥박, 사용자 체온, 실내온도, 사용자의 가전기기 컨트롤 명령이 있으며 생성된 사용자 프로파일 데이터를 MySql서버에 자동으로 저장한다.

본 논문에서 구현된 사용자 프로파일 생성기는 단순 컨텍스트 데이터만을 생성하고 있다는 점이 단점이다. 향후 연구에서는 보다 많은 데이터를 다루거나 실제 데이터에 가까운 가상의 사용자 프로파일을 생성하도록 하여, 보다 정확한 데이터로써 홈네트워크 시스템 실험에 적용될 수 있도록 구현 되어야 할 것이다.

참고문헌

[1] Sherif, M. H, "Intelligent Homes: a new challenge in telecommunications standardization", Communication Magazine. IEEE, Vol. 40, Issue. 1, 2002, 8-8

[2] Das, S.K., Cook, D.J., "Guest Editorial - Smart Homes", Wireless Communications, IEEE, 9(6), 2002, page.62 - 62

[3] 박.재.영. 외. 4인., "국외개인화서비스기술동향", TTA 저A널/정보통신표준화소식, 2008.

[4]W. Y. Lum, F. C. M. Lau, "A Context-Aware Decision Engine for Content Adaptation," IEEE Pervasive computing, Vol. 1, pp. 41-49, Sep. 2002.

[5]Hsieh, R., Zhou, Z.-G., and Seneviratne, A. "S-MIP: A Seamless Handoff Architecture for Mobile IP," In Proc. of INFOCOM, San Francisco, USA, Vol. 3, pp. 1774-1784, Apr. 2003.

[6]Fan, W., M.D.Gordon, and P. Pathak, "Effective profiling of consumer information retrieval needs: a unified framework and empirical comparison,"Decision Suppor Systems, Vol.40, No.2(2005), pp.213-233.

[7]