

위치 인식을 이용한 음식점 추천 시스템의 설계 및 구현

윤혜진[†], 창병모^{**}

요 약

본 연구에서는 상황 적응 시스템을 이용하여 위치 인식 기반의 음식점 추천 서비스를 개발함으로써 이 시스템이 실제 상황 인식 응용 프로그램 개발에도 유용하게 사용될 수 있음을 보일 것이다. 이를 위해 상황 적응 시스템을 기반으로 하여 사용자의 위치, 선호도와 검색 히스토리 등의 정보를 이용하는 위치 인식 기반의 맞춤형 음식점 추천 응용 프로그램을 개발하였다. 상황 적응 시스템은 개발자가 작성한 정책 파일의 내용에 따라 변화된 상황에 맞도록 응용 프로그램을 자동적으로 적용시키고, 응용 프로그램은 위치 등과 같은 변화된 상황을 기반으로 음식점 추천 서비스를 제공한다.

Design and Implementation of Restaurant Recommendation System based on Location-Awareness

Hyejin Yoon[†], Byeong-Mo Chang^{**}

ABSTRACT

This research aims to show that the context adaptation system can be used to develop practical context-aware applications by developing a restaurant recommendation system based on location-awareness. In this research, we have designed and implemented a location-aware restaurant recommendation system which provides a customized restaurant recommendation service based on the user's current context. The context-adaptation engine adapts the application program according to the policy file as contexts are changed, and the application provides restaurant recommendation service based on the changed context like location.

Key words: Location-Awareness(위치-인식), Context Adaptation System(상황 적응 시스템), Smart Phone(스마트폰), Recommendation System(추천 시스템)

1. 서 론

상황 인식 컴퓨팅이란 사용자의 입력을 최소화하여 상황에 따라 언제 어디서나 적절한 서비스를 받을 수 있는 환경을 제공하는 것이다[1-3]. 최근 들어 상황 인식 컴퓨팅에 대한 요구가 높아지면서 다양한 연구 결과가 나오고 있으며 실제 서비스가 가능한 애플리케이션 개발이 진행되고 있다[1,4-6].

본 연구에서는 [7]에서 개발된 상황 적응 시스템을 이용하여 스마트폰 기반의 음식점 추천 애플리케이션을 개발함으로써 이 시스템이 실제 상황 인식 서비스 개발에도 유용하게 사용될 수 있음을 보일 것이다.

본 연구에서는 이를 위해 상황 적응 시스템을 이용하여 사용자의 현재 위치, 개인 선호도와 검색 히스토리 등의 변화하는 상황에 자동으로 적용하여 현재 상황에 맞는 음식점을 추천하는 스마트폰 기반의

※ 교신저자(Corresponding Author): 창병모, 주소: 서울시 용산구 청파동 숙명여대 컴퓨터과학부(140-742), 전화: 02)710-9378, FAX: 02)710-9296, E-mail: chang@sookmyung.ac.kr

접수일: 2010년 7월 30일, 수정일: 2010년 9월 29일

완료일: 2010년 11월 2일

[†] 정회원, NHN 서비스 연구원

(E-mail: pangru@naver.com)

^{**} 정회원, 숙명여대 컴퓨터과학부 교수

맞춤형 음식점 추천 애플리케이션을 개발하였다. 이를 위해 먼저 변화하는 상황에 적응하기 위한 상황 적응 정책을 정책 기술 언어[7]를 이용하여 작성하였다. 상황 적응 시스템은 이 상황 적응 정책에 따라 변화하는 상황에 자동으로 해당 애플리케이션을 적용시켜 주며 이 애플리케이션은 변화된 상황을 반영하여 주변 음식점들을 추천해준다.

본 연구는 위치와 같은 상황을 인식하는 애플리케이션을 상황 적응 시스템과 정책 기술 언어를 이용하여 보다 체계적이고 효과적으로 개발할 수 있다는 면에서 그 의의가 있다.

본 논문의 구성은 다음과 같다. 2장에서는 정책 기술 언어 및 상황 적응 시스템에 대해서 소개한다. 3장에서는 개발한 시스템의 설계 및 구현에 대해서 기술한다. 4장에서는 실행 과정에 대해서 기술한다. 5장에서 관련 연구들을 소개하고 6장에서 결론을 맺는다.

2. 상황 적응 시스템

본 연구에서는 유비쿼터스 환경을 위한 상황 적응 시스템[5]을 기반으로 하고 있으며 이 시스템에서는 정책 기술 언어를 이용하여 상황 적응 정책을 규칙 형태로 정의하며 상황 적응 엔진은 상황이 변함에 따라 적용 가능한 규칙을 적용하여 응용 프로그램을 변화된 상황에 적용시킨다. 정책 기술 언어 및 상황 적응 시스템에 대해서 보다 자세히 설명한다.

2.1 정책 기술 언어

변화하는 상황에 적응하는 규칙을 정책 기술 언어 [7]를 사용하여 기술한다. 먼저 상황을 표현하는 방법에 대해 살펴보고, 다음으로 상황 적응 정책을 어떻게 정의하는지 설명한다.

엔티티는 물리적·논리적 공간이나, 움직이거나 움직이지 않는 객체를 나타낸다. 예를 들어, “floor”, “sickroom”은 공간 엔티티이고 “Phone”은 움직이는 엔티티이다. 공간 엔티티는 그 위치가 고정되어 변하지 않는 반면, 움직이는 엔티티는 동적으로 그 위치나 상황이 변한다. 실세계의 엔티티들은 Entity 클래스의 인스턴스로서 그림 1과 같은 문법으로 표현된다.

Entity-Expression은 한 개의 엔티티나 엔티티의 집합을 표현한다. $id_1 : id_2$ 는 Phone:BobPhone와 같

$p \in \text{Entity-Expression} ::=$ $id_1 : id_2 \mid \$id \mid \$id_n \mid *$ $\mid p_1/p_2 \mid \dots/p$
$r \in \text{Relation-Expression} ::=$ $id_1 (p_1, id_2, p_2) \mid \sim r \mid r_1 r_2$
$n \in \text{Number}$

그림 1. 상황 표현을 위한 문법

이 id_1 이라는 이름을 갖는 클래스의 인스턴스 id_2 를 표현한다. \$id는 id라는 이름을 갖는 클래스의 일반적인 인스턴스 중의 하나를 나타내는 변수를 의미한다. \$Room은 Room이라는 클래스의 어떤 인스턴스를 담은 변수가 된다.

p_1/p_2 는 엔티티들간의 종속 관계를 표현한다. 예를 들어, 다음 표현식은 ubihosp라는 건물의 1층에 있는 어떤 방을 의미한다.

Building: UbiSoft/Floor : f11/\$Room

다음 표현식은 UbiSoft라는 건물에 있는 Phone의 인스턴스를 의미한다.

Building: UbiSoft/.../\$Phone

다음의 표현식은 UbiSoft 건물의 f11에 있는 모든 엔티티들을 의미한다.

Building: UbiSoft/Floor : f11/*

또한 ‘~’은 NOT을 의미하며, ‘^’은 AND 연산자로서 여러 개의 관계에 대해 동시에 참, 거짓을 판단하도록 유도한다.

상황 적응 정책은 변화하는 상황에 어떻게 적응해야 할 것인가에 대하여 정의한 것으로 문법은 다음과 같다.

Adaptation-Rule은 상황에 따라 어떻게 반응해야 할지를 기술하는 규칙이다. 이 규칙은 조건(r)이 참이 된다면 행동(a)을 실행하라는 것을 의미한다. 적용 규칙이 실행되기 위한 조건은 Relation-Expression을 이용하여 표현되고, 실행되는 행동은 매소드 호출 $p_1.id(p_2)$ 이나 새로운 관계 $id_1(p_1, id_2, p_2)$ 의 생성으로 표현된다. 예를 들어, ‘UbiSoft라는 빌딩의 사무실에 사람이 들어오면 관리자의 PC에 그 사

$d \in \text{Adaptation-Rule} ::= r \Rightarrow a \mid d_1 d_2$ $a \in \text{Action} ::= p_1.id(p_2) \mid id_1(p_1, id_2, p_2) \mid a_1; a_2$

그림 2. 상황 적응 정책의 문법

```
Location($Person, IsIn, Building:UbiSoft/.../$Room) ^
Ownership($Admin, Owns, $PC)
=> $PC.showInfo($Person)
```

그림 3. 상황 적응 규칙의 예

람의 정보를 보여준다.’ 라는 상황 적응 정책은 아래와 같이 작성될 수 있다.

2.2 상황 적응 엔진

[5]의 상황 적응 시스템은 상황 적응 정책이 기술 되어 있는 정책 파일을 읽고 분석하기 위한 정책 파일 리더기와 그 분석 내용에 따라 응용 프로그램을 상황에 적응시키기 위한 상황 적응 엔진으로 구성되며 시뮬레이션을 위한 시뮬레이터를 제공한다.

정책 파일 리더기는 상황 적응 정책을 클래스 형태로 분석하고 저장한다. 상황 적응 엔진은 정책 파일 리더기가 분석해 준 상황 적응 정책의 내용을 기반으로 프로그램을 변화하는 상황에 적응시킨다. 그림 4는 상황 적응 엔진이 실행되는 시스템의 구조를 보여준다.

상황 적응 엔진은 프로그래머가 작성한 정책 파일을 분석하고, 그 정보를 이용하여 해시 테이블을 구성한다. 그 후, 애플리케이션 프로그램에서의 상황 변화가 인지되면 해시 테이블에서 변화와 관련된 규칙을 검색하여 조건이 참 값을 갖는지 검사한다. 해당 조건의 일치 여부에 따라 해당하는 행동을 자동적으로 수행하게 된다.

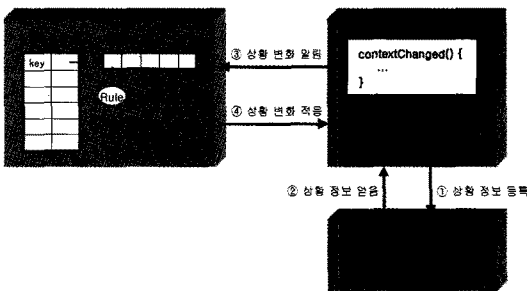


그림 4. 상황 적응 엔진과 응용 프로그램과의 연동

3. 시스템 설계 및 구현

본 연구에서는 정책 기술 언어를 이용하여 상황

적용 정책을 기술하여 사용자의 위치와 같은 상황 변화에 따라 시스템이 적응하여 변화된 상황에 맞는 음식점 추천 정보를 제공할 수 있도록 설계하였다.

3.1 시스템의 개요 및 구성

본 시스템의 주요 개요는 다음과 같다.

(1) 위치 기반 서비스

본 시스템은 GPS를 기반으로 현재 위치 정보를 이용하여 사용자에게 맞는 음식점 정보를 제공한다.

(2) 스마트폰 기반의 사용자 인터페이스

본 시스템은 스마트폰을 기반으로 하여 어디서든 이용할 수 있으며 편리한 사용자 인터페이스를 제공한다.

(3) 개인 맞춤형 추천 서비스

본 시스템은 사용자의 상황에 따라 각 개인에게 맞춤형 음식점들을 추천해준다.

(4) 정책파일을 이용한 편리한 개발 환경

본 연구에서는 상황 적응 시스템이 제공해주는 개발환경을 이용하며 개발자가 응용 프로그램과 정책 파일만 작성하면 상황 적응 시스템이 자동으로 응용 프로그램을 적응시켜준다.

본 시스템에서는 GPS 수신이 가능한 스마트폰을 기반으로 동작하며 그림 5와 같이 크게 네 부분으로 구성된다.

첫 번째 부분은 상황 적응 시스템과 애플리케이션 프로그램인 ubiArea이다. 상황 적응 시스템은 2.2에

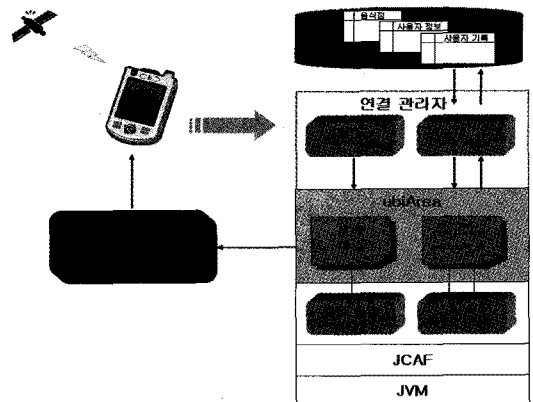


그림 5. 개인맞춤형 음식점 추천 시스템 구성

서 설명한 것처럼 현재 상황이 변화할 때마다 ubiArea를 미리 작성된 상황 적응 규칙에 따라 자동적으로 적절히 반응하도록 만들어 줌으로써 현재 위치와 같은 사용자의 변화하는 상황에 따라 적응시킨다. ubiArea는 음식점 추천 서비스에 등장하는 사용자의 스마트폰과 현재 위치 정보, 음식점들을 정의하고, 개체화시켜 상황 적응 시스템에 등록, 연동함으로써 상황 변화에 따라 상황 적응 시스템을 통해 적응될 수 있도록 한다.

두 번째 맞춤형 추천 웹 서비스 부분은 상황 적응 시스템에 의해 적용된 프로그램인 ubiArea로부터 받은 정보를 이용하여 사용자의 선호도와 검색 히스토리에 따라 우선순위를 정하여 사용자에게 더 필요한 정보가 먼저 보일 수 있도록 음식점을 추천해서 보여주는 부분이다. 이 시스템에서는 과거에 사용자가 선택했던 음식점들의 타입과 그 당시의 주변 상황, 선호도 등을 반영하여 현재 추천 내용에 반영될 수 있도록 한다.

세 번째 부분은 연결 관리자(connection manager)이다. 연결 관리자는 사용자의 스마트폰으로부터 현재 위치를 받아오는 GPS모듈과 사용자의 위치 정보에 따라 변화되는 음식점의 정보를 가져올 수 있도록 도와주는 DB모듈로 구성된다.

마지막 네 번째 부분은 사용자의 스마트폰에 설치되어 사용자의 현재 위치 정보를 연결 관리자에게 전달해주는 클라이언트 프로그램이다. 본 시스템에서는 삼성 T*옵니아 스마트폰을 기반으로 하여 이 부분을 구현하여 테스트하였다.

3.2 시스템의 진행 과정

사용자의 현재 상황을 기반으로 한 음식점 추천 서비스는 다음과 같은 과정으로 진행된다.

- (1) 사용자의 스마트폰에 설치된 클라이언트 프로그램 실행한다.
- (2) 맞춤형 추천 서비스를 위해 로그인한다.

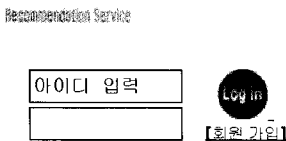


그림 6. 로그인

- (3) ubiArea는 연결 관리자의 GPS모듈을 통해 받은 사용자의 현재 위치로부터 Zone을 구성한다.

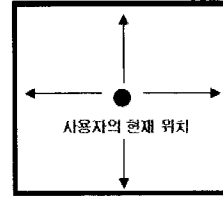


그림 7. 현재 Zone

- (4) 사용자의 주변 상황 변화에 따라 ubiArea는 상황 적응 엔진에게 상황 변화를 알리고, 적응을 요청한다.

- (5) 사용자의 변화된 위치에 따라 ubiArea를 통해 Zone을 재구성한다.

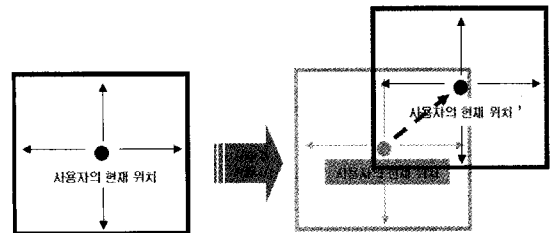


그림 8. 사용자 이동에 따른 Zone의 변화

- (6) ubiArea는 DB모듈을 통해 사용자의 현재 상황을 반영한 음식점 목록을 구성한다.

- (7) 맞춤형 추천 웹서비스는 ubiArea에 현재 음식점 목록을 요청한다.

- (8) ubiArea로부터 받은 음식점 목록에 사용자의 선호도와 히스토리를 반영하여 재구성한 후, 사용자의 웹 브라우저에 제공한다.

3.3 음식점 추천 서비스의 상황 적응 정책

그림 9는 본 서비스를 위한 정의로 ubiar라는 공간이 있고, 그 공간 안에 Zone이 존재한다는 의미이다.

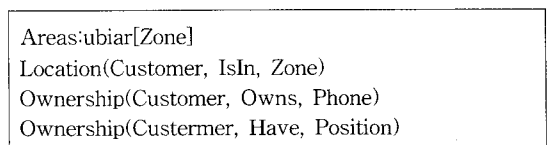


그림 9. 음식점 추천 서비스의 정의

두 번째 정의는 엔티티인 Customer가 Zone 안에 있다는 것을 의미하고, 세 번째 정의는 Customer가 스마트폰을 가지고 있고 Customer는 위치를 가질 수 있다는 상황을 의미한다.

그림 10은 본 서비스를 위한 상황 적응 정책의 일부를 기술한 것이다. 첫 번째 규칙은 사용자가 스마트폰을 가지고 있고, 현재 위치가 있으면 사용자는 Zone 안에 존재한다는 것을 의미한다. 이 규칙은 사용자가 가지고 있는 스마트폰에서 현재 위치 정보를 갖게 되면 사용자의 현재 위치 정보를 중심으로 Zone 안에 있다는 것을 의미한다. 두 번째 규칙은 사용자가 Zone안에 존재하면 사용자의 현재 위치 정보를 중심으로 일정 영역인 Zone안에 있는 음식점의 목록들을 보여준다는 의미이다.

```
Ownership($Customer, Owns, $Phone) ^
Ownership($Custermer, Have, $Position)
=> $Customer.setZone($Position)

Location($Customer, IsIn, Areas:ubiar/$Zone) ^
Status(Restaurants, IsOccurriedEmpty, Seats)
=> $Phone.getList($Phone)
```

그림 10. 음식점 추천 서비스를 위한 적응 정책

3.4 맞춤형 추천 과정

맞춤형 추천 서비스는 사용자가 등록한 선호도와 선택 히스토리를 이용하여 ubiArea에서 전달받은 사용자의 현재 Zone 내에 위치한 음식점 목록을 재구성하여 웹페이지를 통해 음식점 추천 서비스를 제공한다.

사용자는 음식점 추천 서비스를 제공받기 위해 회원 로그인 과정을 거쳐야 메인 화면의 현재 상황을 반영한 음식점 추천 목록들을 보여준다. 만약 사용자가 비회원이라면 초기(로그인) 화면의 [회원가입] 과정을 거쳐 서비스 받을 수 있다. 그림 11은 회원 가입 정보를 보여주고 있다. 그림 11에서 입력한 주변 반경 범위는 메인 화면에서 재지정할 수 있다.

그림 12의 알고리즘은 GPS로부터 자동으로 인식되는 현재 위치를 중심으로 하여 ubiArea로부터 전달받은 주변 음식점 목록으로부터 맞춤형 추천 목록을 구성한다. 이 알고리즘은 GPS를 이용한 사용자 현재 위치를 자동으로 파악하여 맞춤형 추천 목록을

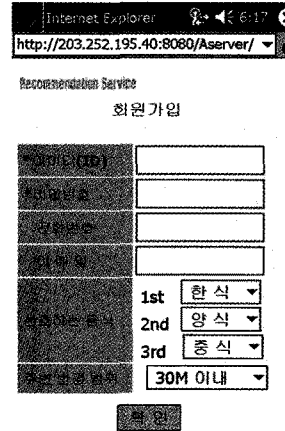


그림 11. 회원가입

STEP 1. 음식점의 평점
ubiArea로부터 전달받은 음식점들의 평점을 확인한다.
(1) 총평점이 1개 이하이면 해당 음식점에 priority 1을 감소한다.
(2) 총평점이 4개이면 해당 음식점에 priority 2를 증가한다.
(3) 그 외의 경우 해당 음식점에 priority 1을 증가한다.

STEP 2. 방문 기록 검색
ubiArea로부터 전달받은 음식점들 중 foodReview Table에 사용자가 등록한 감상평 목록을 확인한다.
(1) 총평점이 1개 이하이면 해당 음식점에 priority 1을 감소한다.
(2) 총평점이 4개이면 해당 음식점에 priority 2를 증가한다.
(3) 그 외의 경우 해당 음식점에 priority 1을 증가한다.

STEP 3. 사용자의 선호도 검색
(1) 사용자가 회원가입 시 등록한 음식점의 종류에 따른 우선순위에 따라 해당 음식점에 priority 3, priority 2, priority 1을 증가한다.
(2) 사용자의 현재 위치와의 음식점과의 거리를 고려하여 사용자가 등록한 주변 반경 범위 내에 존재하는 해당 음식점에 priority를 증가한다.
주변 반경 범위를 3 등급으로 분할하여 근거리 순에 따라 가장 가까운 범위는 priority 3을 증가시킨다. 그 다음 범위는 priority 2, 나머지 범위의 음식점들은 priority 1을 증가한다.

그림 12. 맞춤형 추천 목록 구성 알고리즘

구성한다는 면에서 Yahoo 거기 등과 같은 기존의 웹 기반 서비스와 차별화 된다. 본 알고리즘에서는 음식점에 달린 평점, 사용자가 등록한 감상평, 사용자가 현재 위치한 근거리 순서에 따라 우선순위를 부여한다. 각 단계를 거치며 검색된 음식점들의 최종

우선순위가 결정되고, 사용자에게 보여줄 추천 목록은 우선순위 값에 따라 구성된다.

본 시스템의 메인화면에서 사용자가 여러 음식점들 중 하나를 선택하면, 선택된 음식점의 상세정보를 보여준다. 그림 13은 특정 음식점의 평점, 장소, 메뉴, 지도 등을 보여주는 상세화면이다. 음식점의 상세 정보 페이지 우측 하단에 3개의 버튼이 있다. 버튼 홈 [13]은 음식점 상세 메인 화면으로 이동하며, 버튼 내비게이션 [14]은 사용자가 선택한 음식점의 위치를 보여준다. 버튼 노트 [15]는 사용자가 방문했던 음식점에 기록과 평점을 남길 수 있고, 이렇게 등록된 정보는 사용자의 맞춤형 추천 목록을 구성하는데 사용된다.

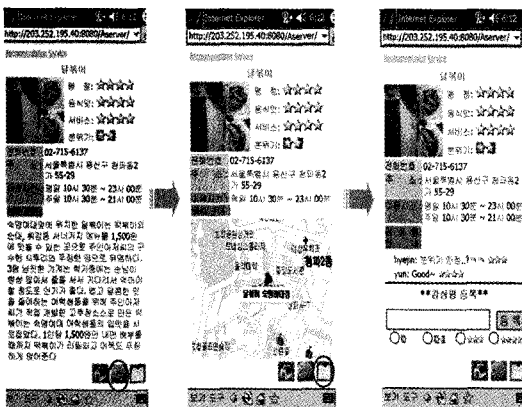


그림 13. 음식점 상세 화면 흐름도

4. 실행 과정

음식점 추천 서비스는 상황 적응 시스템에서 제공하는 시뮬레이터를 이용해 서비스를 실행시킨 후, 스마트폰에서 웹 브라우저를 이용해 서비스 이용이 가능하다.

그림 14는 유비쿼터스 프로그램인 ubiArea와 상황 적응 정책 파일을 등록한 화면이다.

시스템이 실행되면 사용자의 변화되는 현재 위치에 따라 현재 상황 정보를 그림 15의 시뮬레이터의 Current Context Status 화면에서 확인할 수 있다.

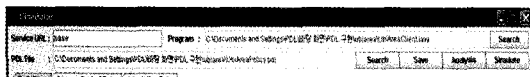


그림 14. 상황 적응을 위한 환경 설정

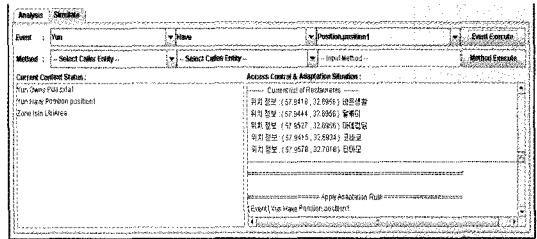


그림 15. 음식점 추천 서비스에 대한 현재 상황 및 실행된 상황 적응 정책

본 연구에서는 실험 대상 지역을 모 대학 근처로 지정하여 사용자의 현재 위치와 서로 다른 선호도에 따른 추천 정보를 확인해보았다.

4.1 사용자의 현재 위치 변화

이 실험은 사용자의 위치 변화에 따라 본 시스템이 변화하는 상황에 적응하여 어떠한 추천 목록을 보여주는지 알아본다. 그림 16. 사용자의 현재 위치 변화에 따라 Zone 안의 음식점 목록의 변화를 보여준다.

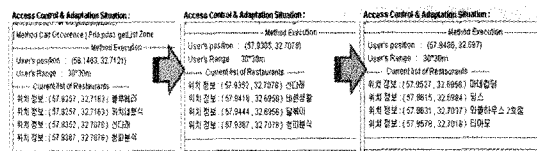


그림 16. 사용자의 Current Context Status

그림 17에서는 그림 16의 음식점 목록 구성에 사용자의 선호도가 반영되어 변화되는 추천 목록들을 확인할 수 있다.

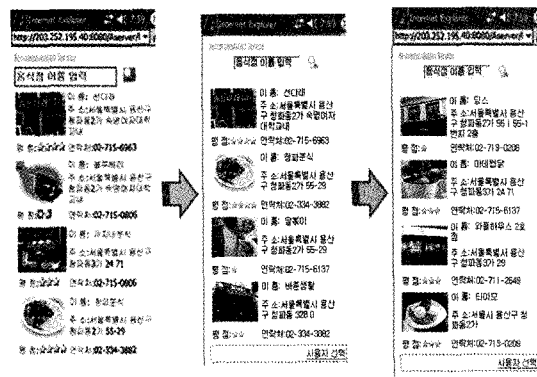


그림 17. 사용자의 현재 위치에 따른 추천

4.2 개인별 추천

본 실험은 같은 상황에서 서로 다른 기호에 따라 추천되는 개인 맞춤형 추천을 비교실험하기 위해 두 개의 아이디, 'yun'과 'hyejin'을 이용하여 비교하였다. 그림 18은 두 개의 아이디의 회원 가입 내용이다. yun과 hyejin이 총 순위별로 추천된 결과 화면은 그림 19와 그림 20과 같다.

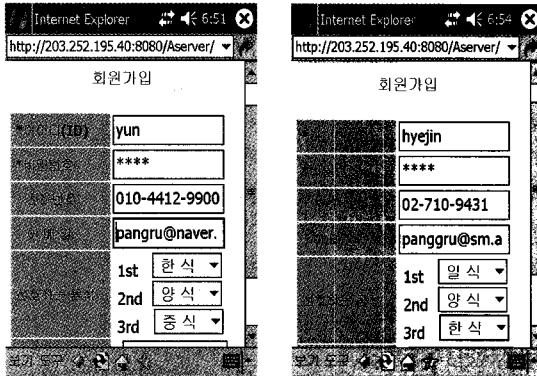


그림 18. 'yun'과 'hyejin'의 회원 가입 정보

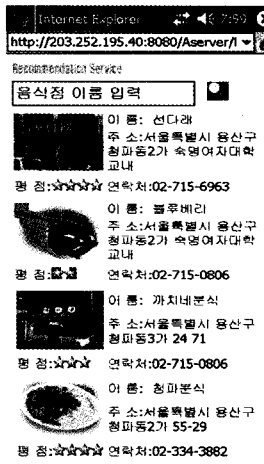
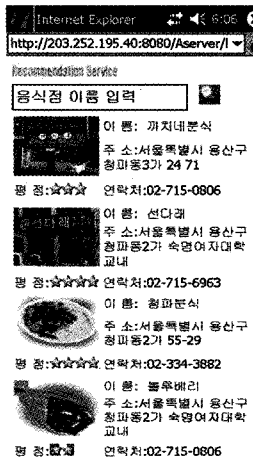


그림 19. yun의 추천목록

그림 20. hyejin의 추천목록

그림 21은 사용자의 현재 위치를 기반으로 추천받은 음식점 목록들 중 사용자의 선택에 따라 정렬하여 보여주는 화면이다.

5. 관련 연구

상황 인식 시스템 관해서 수행된 대표적인 연구

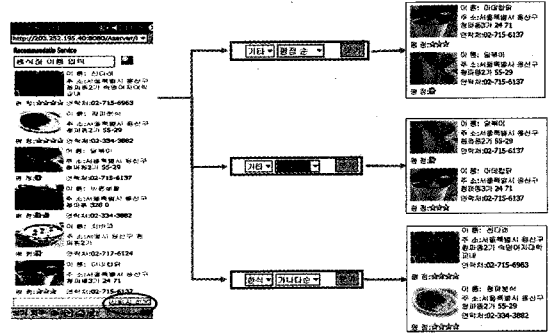


그림 21. 사용자의 선택 화면 흐름도

들을 정리하면 다음과 같다.

JCAF[8]는 상황 인식 프로그램 개발을 위한 하부 구조와 API를 제공하는 Java 기반 상황 인식 프로그래밍 프레임워크로 상황 인식을 효과적으로 지원할 수 있다. [2]에서 개발된 Gaia 상황인식 시스템은 상황에 따라 나타나야 하는 반응을 1차 논리를 기반으로 작성하고 처리한다. [8]에서 개발된 상황 인식 자원 관리 미들웨어 CARMEN은 메타데이터의 내용을 기반으로 하여 무선 인터넷 환경에서 변화된 환경을 재구성한다. [2,3]의 미들웨어를 이용한 애플리케이션 개발이 이루어지고 있으나 스마트폰 같은 최신 기술과 연동된 실용적인 애플리케이션은 아직 개발되지 못하고 있다.

최근 들어 [1,4,6]에서 위치 인식을 이용한 모바일 기반 관광 가이드 애플리케이션들이 개발되었는데 이 시스템들은 GPS를 이용한 위치 인식을 이용하고 있으며 상황 인식 시스템을 이용하기 보다는 하나의 독립적인 애플리케이션으로 구현되었다.

웹 혹은 스마트폰 기반의 대표적인 맛집 추천 서비스는 다음과 같다. “야후 거기”는 대표적인 웹 기반 지역 검색 및 맛집 추천 서비스이다[9]. 이 서비스는 음식 종류, 취향, 특징별로 맛집을 검색할 수 있다. 스마트폰을 위한 맛집 정보를 알려주는 앱으로는 ‘윙버스 맛집’이 있다[10]. 네이버에서 서비스되는 윙버스를 아이폰용으로 옮긴 것으로 음식 종류별, 특징별로 검색할 수 있다. 맛집에 대한 평점과 사용자의 댓글과 리뷰 등을 제공한다. 또한 “블루리본 서베이-서울의 레스토랑”이 아이폰용으로 개발되었다[11]. 서울 지역의 레스토랑을 현재 위치와 상세검색의 조건들을 조합해 보다 빨리, 보다 정확하게 원하는 결과를 찾을 수 있다.

본 연구의 의미 및 차별성은 다음과 같이 세 가지로 정리할 수 있다. 첫 번째로 위치와 같은 상황을 인식하는 애플리케이션을 상황 적응 시스템과 정책 기술 언어를 이용하여 보다 체계적이고 효과적으로 개발했다는 면에서 그 의의가 있다. 두 번째로 상황 적응 시스템과 같은 미들웨어를 이용한 애플리케이션이 최신 스마트폰 기술과 연동되어 개발되었다는 점에서 실용성 면에서 의의가 있다. 마지막으로 이 시스템은 GPS를 이용하여 사용자 현재 위치를 자동으로 파악하여 맞춤형 추천 목록을 구성한다는 면에서 웹 혹은 스마트폰 기반의 상용 맛집 추천 서비스와 차별화 될 수 있다.

6. 결 론

본 논문에서는 [7]의 상황 적응 시스템을 기반으로 변화하는 상황을 인식하는 스마트폰 기반 음식점 추천 애플리케이션을 개발함으로써 이 시스템이 상황 인식 애플리케이션 개발에 효과적으로 사용될 수 있음을 보였다.

향후 연구에서는 본 연구에서 이용한 상황 적응 시스템을 기반으로 하여 교통, 문화, 관광 등 보다 다양한 분야의 상황 인식 기반의 애플리케이션을 개발할 것이다.

참 고 문 헌

- [1] A. K. Dey, D. Salber, and G. D. Abowd, "A Conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Human-Computer Interaction*, Vol.16 No.2, 2001.
- [2] A. Ranganathan and Roy H. Campbell, "An infrastructure for context-awareness based on first order logic," *Personal and Ubiquitous Computing*, Vol.7 No.6, pp. 353-364, Dec. 2003.
- [3] P. Bellavista, A. Corradi, R. Montanari, and C. Stefanelli, "Context-Aware Middleware for Resource Management in the Wireless Internet," *IEEE Trans. on Soft. Eng.* Vol.29 No.12, pp. 1086-1099, Dec. 2003.
- [4] A. Pashan, R. Blattler, A. Heusser and P. Scheuermann, "CATIS: A Context-aware Tourist Information System," *Int. Workshop of Mobile Computing*, 2003.
- [5] M., Oh, J. Lee, B. Chang, J. Ahn, and K. Doh, "A Programming Environment for Ubiquitous Computing Environment," *ACM SIGPLAN Notices Volume 42(4)*, pp. 14-22, April 2007.
- [6] S. Wang, J. Min and B. Yi, "Location Based Services for Mobiles: Technologies and Standards," *IEEE International Conference on Communication (ICC) 2008*, Beijing, China.
- [7] J. Ahn, K. Doh and B. Chang, "A Policy Description Language for Context-based Access Control and Adaptation in Ubiquitous Environment," *TRUST'06*, Aug. 2006.
- [8] J. E. Bardram, "The Java Context Awareness Framework-A Service Infrastructure and Programming Framework for Context-Aware Applications," *Third International Conference, Pervasive 2005*, Munich, Germany, May, 2005.
- [9] 야후 거기, <http://kr.gugi.yahoo.com>
- [10] wingbus 맛집, <http://r.wingbus.com/seoul>
- [11] 블루리본 서베이 - 서울의 레스토랑 <http://ipodart.net/822>
- [12] H. Fagrell, K. Forsberg and J. Sanneblad, "FieldWise: A Mobile Knowledge Management Architecture," *ACM Conf. on CSCW*, 2000.
- [13] "Windows Mobile 6 Professional and Standard Software Development Kits," Microsoft, 2009.



윤 혜 진

2007년 숙명여자대학교 컴퓨터과
학과 졸업(학사)
2009년 숙명여자대학교 컴퓨터과
학과 졸업(석사)
2009~현재 NHN 서비스 연구원
관심분야: 프로그래밍 언어, 유비
쿼터스 소프트웨어



창 병 모

1988년 서울대학교 컴퓨터공학과
졸업(학사)
1990년 KAIST 전산학과 졸업
(석사)
1994년 KAIST 전산학과 졸업
(박사)

1995~현재 숙명여자대학교 컴퓨터과학부 교수
관심분야: 프로그래밍 언어, 프로그램 분석, 유비쿼터스
소프트웨어