

# 커뮤니티 컴퓨팅에서 사용자 요구 반영을 위한 통계적 패턴 인식 기법

## A Statistical Pattern Recognition Method for Providing User Demand in Community Computing

김성빈\* · 정혜동\*\* · 이형수\*\* · 김석윤\*\*\*

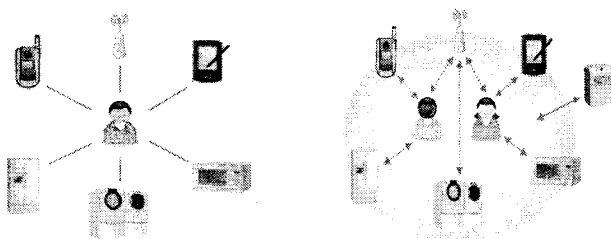
Sung-Bin Kim · Hye-Dong Jung · Hyung-Su Lee · Seok-Yoon Kim

**Abstract** - The conventional computing is a centralizing system, but it has been gradually going to develop ubiquitous computing which moves roles away from the main. The Community Computing, a new paradigm, is proposed to implement environment of ubiquitous computing. In this environment, it is important to accept the user demand. Hence in this paper recognizes pattern of user's activity statistically and proposes a method of pattern estimation in community computing. In addition, user's activity varies with time and the activity has the priority. We reflect these. Also, we improve accuracy of the method through Knowledge Base organization and the feedback system. We make program using Microsoft Visual C++ for evaluating performance of proposed method, then simulate it. We can confirm it from the experiment result that using proposal method is better in environment of community computing.

**Key Words** : Community Computing, pattern recognition, user demand, Ubiquitous Computing

### 1. 서론

유비쿼터스 컴퓨팅 환경은 기존의 중앙 관리 형식의 컴퓨팅이 아닌 복잡한 분산 환경 중 하나로 생각할 수 있다. 그림 1에 묘사된 것과 같이 커뮤니티 컴퓨팅은 이러한 유비쿼터스 컴퓨팅의 특징을 반영한 기법이다.



(a) 기존의 컴퓨팅

(b) 커뮤니티 컴퓨팅

그림 1. 커뮤니티 컴퓨팅의 개념[1]

커뮤니티 컴퓨팅 환경에 PDA, PMP 등 여러 개의 컴퓨팅 장치가 존재하고, 가정 내의 홈네트워크 환경이나 기타 다른 환경에서 이러한 장치들에 서비스를 제공하는 기능적 프로바이더(Provider) 역할을 하는 것을 Pway(Personal gateway)라 한다[3].

결국 이러한 것들은 사용자의 편의를 위해 제안되고 개발

저자 소개

\* 崇實大學 컴퓨터學科 碩士課程

\*\* 電子部品研究院 知能型 情報研究센터, KETI

\*\*\* 崇實大學 컴퓨터學科 正教授 · 工博

본 연구는 지식경제부 지역산업기술개발사업의 “PAMP용 개인화 미디어 게이트웨이 시스템 기술개발” 지원에 의한 것임.

된 것으로 본 논문은 이를 향상시키기 위하여 사용자의 요구를 신속하게 반영할 수 있는 기법을 제안한다. 제안하는 기법은 커뮤니티 컴퓨팅 환경에서 사용자의 행동을 통계화하여 일련의 연관성을 추출하고 이에 따라 보다 빠르게 사용자 요구를 반영하는 것이다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장은 커뮤니티 컴퓨팅에 대한 보다 자세한 언급을 하고, 3장에서는 실시간 요구 반영을 위한 패턴 인식 기법에 대해 설명한다. 4장에서는 실험을 통하여 기법을 적용하지 않았을 경우와 기법을 적용했을 경우의 성능 비교를 한다. 마지막으로 5장에서는 결론을 맺는다.

### 2. 커뮤니티 컴퓨팅(Community Computing)

개체들이 어떤 공통된 목표를 위하여 그룹을 결성하듯 유비쿼터스 컴퓨팅 환경에서 분산된 컴퓨팅 자원들이 어떤 특정한 서비스 목표를 이루기 위해 서로 협력하여 추구하는 과정을 커뮤니티를 구성한다고 한다. 커뮤니티 컴퓨팅이란 유비쿼터스 환경에 내재하는 여러 멤버들이 이러한 커뮤니티를 형성하고, 멤버간의 상호 협력을 통하여 유비쿼터스 서비스를 제공하는 것이다[2]. 또한 커뮤니티 컴퓨팅에서는 기존의 존재하는 컴퓨팅 장치(Computing Device)나 응용 서비스(Application Service) 등을 결합하기 위해 새롭게 설계, 개발하는 과정의 수행을 거치지 않고, 동적으로 기존의 서비스나 기기를 결합시킴으로써 발전된 서비스를 개발하는 게 가능하다[3]. 이러한 커뮤니티 컴퓨팅 환경에서는 다음과 같이 커뮤니티를 정적, 동적 그리고 자율적인 커뮤니티로 분류할 수 있고, 각각의 특징은 다음의 표 1과 같다[4].

표 1. 커뮤니티 컴퓨팅에서의 세 가지 커뮤니티

	커뮤니티 템플릿	역할 및 구성
정적 커뮤니티	고정	고정
동적 커뮤니티	고정	가변
자율적 커뮤니티	가변	가변

이러한 특징으로 인해 다른 두 가지 방식의 커뮤니티 보다 자율적인 커뮤니티의 구성이 커뮤니티 컴퓨팅에 있어서 궁극적인 목표가 될 수 있다.

### 3. 제안하는 기법

사용자가 서비스를 이용함에 있어 일련의 패턴을 가진다는 사실을 고려하여 신속하게 사용자의 요구를 반영하도록 한다. 예를 들어 홈 네트워크의 경우, 사용자가 집에 들어와서 할 수 있는 행동에는 전등을 켜거나 히터 혹은 에어컨을 동작시킬 수 있고, 또는 TV를 켤 수 있는 등 여러 가지 행동들이 있을 수 있다. 이러한 행동들을 묶어 연속된 행동들을 하나의 패턴으로 규정하여 knowledge base에 저장한다. Knowledge Base에 있는 패턴 테이블은 시간을 나타내는 *Time*, 패턴을 나타내는 *PatternID*, 그리고 그 패턴의 행동 순서를 나타내는 *Sequence*, 마지막으로 그 패턴의 사용된 횟수를 나타내는 *Count*라는 column들을 가지고 있다. 이러한 column에 필요한 데이터를 수집하기 위해 사용자의 행동을 통계적으로 분석한다. 그림 2는 이와 관련한 전체 모듈도이다.

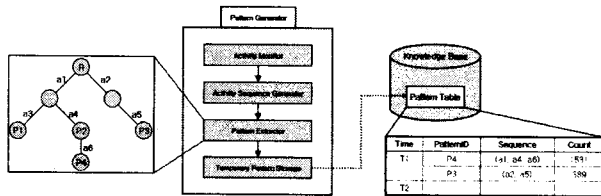


그림 2. 패턴 생성의 전체 모듈도

패턴 생성모듈에서는 먼저 사용자의 행동을 모니터링하고, 모니터링된 행동을 일련의 sequence로 만든다. 그리고 이런 행동의 순서를 패턴 추출트리에 대입하여 패턴이 존재하면 임시 패턴 저장소에 저장을 한다. 만약 패턴 추출트리에 이러한 sequence가 존재하지 않는다면 패턴 추출트리에 새로운 leaf를 생성하고 임시 패턴 저장소와 함께 업데이트한다. 일정한 양의 행동이 모니터링되면 임시 패턴 저장소에 있는 데이터를 knowledge base에 존재하는 패턴 테이블에 업데이트하도록 한다. 예를 들어 사용자의 행동의 순서(Sequence)가 a2 다음에 a5가 될 경우에는 P3가 되고  $S(a2, a5) = P3$ 로 표현한다. 그리고 행동이 시작된 시간과 끝나는 시간으로 계산된 Time값과 함께 패턴 테이블에 저장되게 된다.

Time은 사용자가 시간에 따라서 서비스를 이용함에 다르고, 이는 패턴 인식에 있어서 성능을 높일 수 있는 중요한 요소이다. 가령 예를 들자면, 홈 네트워크에서 사용자가 낮에 집에 들어오면 전등을 켜지 않을 수 있지만 밤에 들어오게 된다면 전등을 거의 켜게 된다. 그렇기 때문에 낮에 적용한 패턴을 밤에 적용하기에는 무리가 있다. 또한 시간에 따라 사용자의 행동이 주기성이 존재한다는 것을 감안하였을 때

이는 반드시 고려해야하는 요소이다. 그러므로 본 논문에서 제안하는 패턴 인식 기법에서는 이러한 사항을 고려하여 시간을 총 4개의 구간으로 나누어서 패턴을 다르게 저장함으로써 정확성을 향상시킨다. Count는 패턴 테이블이 depth만큼의 데이터를 채우기 위하여 높은 빈도를 가지지 않음에도 불구하고 저장하고 있는 패턴을 배제하기 위한 것으로 제안하는 기법에서 임계값(threshold) 이상의 값을 가질 경우에만 패턴 인식을 적용한다.

knowledge base의 패턴 테이블에 데이터가 저장되면 이를 기반으로 패턴 예측을 한다. 그림 3은 이러한 패턴 예측의 순서도를 나타낸다.

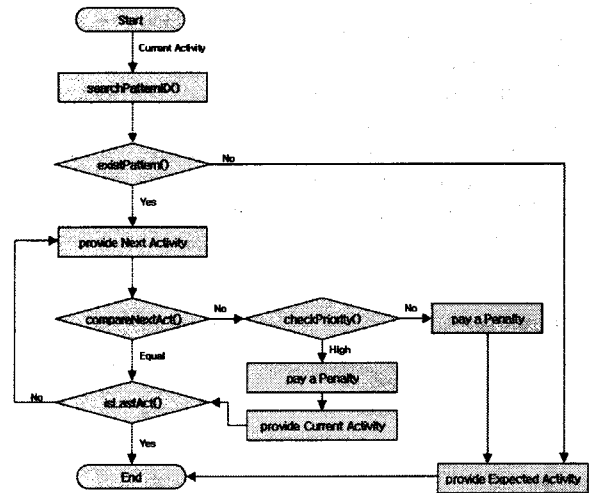


그림 3. 패턴 인식 적용 기법의 순서도

패턴이 인식하고 생성하여 패턴 테이블에 저장되게 되면 그림 3과 같은 흐름으로 패턴을 예측하여 사용자에게 서비스를 제공한다. 먼저 패턴 테이블에서 현재의 행동에 관한 패턴이 패턴 테이블에 존재하는지 검색해보고, 만약 그러한 패턴이 존재한다면 이에 따라 다음 서비스를 제공하는 것이다. 만약 중간에 패턴과 사용자의 행동이 일치하지 않는다면 패턴을 받고 사용자가 요구하는 현재의 서비스를 제공한다.

제안하는 기법은 사용자의 행동에서 정해진 우선순위(Priority)를 고려해 패턴을 유지한다. 예를 들어, 홈 네트워크에서 방에 들어갈 때 방의 전등을 켜고 음악이나 TV를 켤 수 있는데 만약 전화가 온다면 음악이나 TV를 켜기 전에 전화를 받을 수 있다. 전화를 받는 행동은 우선순위가 높은 행동이 갑작스럽게 생긴 것이 된다. 이처럼 우선순위가 높은 행동이 왔을 때 패턴 추론 실패로 보고 다음의 서비스들 모두 제공하지 않는 것이 아니라 우선순위가 높은 행동은 수행해주고 다음에는 기존에 예측한 패턴대로 수행해준다. 이렇게 함으로써 서비스 제공에 있어 보다 나은 성능을 발휘한다.

### 4. 실험 및 결과

본 논문에서 제안하는 기법을 실험하기 위해 Microsoft Visual C++로 프로그램을 구현하였다. 제안하는 기법의 성능도출을 위해서 패턴 인식이 적용하여 예상되는 패턴이 일치하면 서비스에 필요한 시간을 상대적으로 감소시킨다. 만약

패턴이 사용자의 행동과 일치하지 않는다면 패턴에 의한 서비스 제공을 중단하고 사용자가 원하는 서비스를 제공한다. 이미 예측하여 제공된 서비스에 대해서 제공하지 않아도 될 서비스를 제공하였으므로 이에 대한 패널티를 부여하여 시간을 상대적으로 크게 증가시킨다.

다음과 같이 서비스 제공에 필요한 시간( $T_{service}$ )을 준비 시간( $T_{ready}$ )과 실행시간( $T_{execution}$ )으로 구성하였을 때를 식 (1)이라고 한다

$$T_{service} = T_{ready} + T_{execution} \quad (1)$$

추론이 성공하였을 때는 준비시간이 필요하지 않기 때문에 식 (2)와 같이 된다.

$$T_{success} = T_{execution} \quad (2)$$

하지만 추론이 실패하였을 때는 이에 대한 패널티( $T_{penalty}$ )를 받기 때문에 식 (3)과 같이 표현할 수 있다.

$$T_{failure} = T_{penalty} + T_{service} \quad (3)$$

커뮤니티 컴퓨팅 환경에서의 행동을 최대한 반영하기 위해서 어떠한 행동과 패턴에 대해 다른 행동과 패턴에 비하여 빈도의 측면에서 보다 높은 가중치를 두어 랜덤하게 행동을 생성하였다. 또한 특정한 행동이 발생하는 상황을 고려하기 위하여 다른 행동들보다 높은 우선순위를 가지는 행동도 생성하였다. 제안하는 기법의 성능을 측정하기 위해 다음 표 2와 같이 항목을 가정하였다.

표 2. 성능평가를 위한 시간

생성한 행동 수	1000, 5000, 10000
$T_{ready}$	0.2
$T_{execution}$	0.8
$T_{penalty}$	각각 $T_{ready}$ 의 2, 4, 6, 8, 10배
패턴 테이블의 depth	2, 3, 4

이러한 항목으로 볼 때 모든 패턴 예측이 성공하더라도 실행시간의 평균은 0.8보다 작지 않게 된다. 각각의 행동 수에 대하여 실험을 10회씩 시행하였고, 각각의 평균 시간( $T_{unit}$ )을 다음의 식 (4)와 같이 계산하여 평균을 산출하였다.

$$T_{unit} = \frac{T_{no\ technique} - T_{ready} * N_{success} + T_{penalty} * N_{failure}}{N_{total\ Activity}} \quad (4)$$

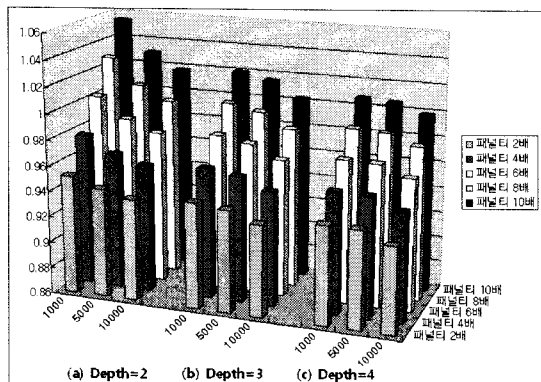


그림 4. 실험 결과 비교

그림 4에서 x축은 생성한 행동 수를 나타내며, y축은 패턴 테이블 변화에 따라 계산을 한 것이다. 그리고 z축은 1을 기준으로 한 성능 비교를 나타낸다.

보는 바와 같이 패턴 테이블에 예측할 수 있는 패턴의 수를 결정하는 depth가 증가할수록 성능이 좋아짐을 알 수 있다. 이는 사용자 행동의 수가 증가함에 따라 일정 임계값 이상의 패턴을 더 많이 가질 수 있어 패턴 예측의 정확도가 그만큼 증가하기 때문이다. 하지만 depth가 2에서 3으로 증가하였을 때만큼 depth가 3에서 4로 증가하였을 때 성능 향상이 좋아지지 않은 이유는 패턴의 추론이 성공했을 경우는 증가해도 패턴 테이블의 더 많은 패턴으로 인하여 추론이 실패할 경우는 크게 줄어들지 않았기 때문이다. 패널티가 10배인 경우를 제외하고는 대부분의 경우에서 아무런 기법이 적용되지 않은 경우보다 제안하는 기법을 적용하였을 경우가 더 좋은 성능 향상을 보였다. 이는 추론 실패에 따른 패널티가 극단적으로 높지 않는 이상 본 논문에서 제안하는 기법을 사용하였을 경우, 커뮤니티 컴퓨팅에서 사용자의 서비스 제공에 있어서 보다 나은 성능을 보인다는 것을 나타낸다.

## 5. 결론

본 논문에서는 커뮤니티 컴퓨팅 환경에서 사용자의 패턴을 인식하고 이를 통계적으로 접근하여 사용자에게 보다 빠르게 서비스를 제공할 수 있는 기법을 제안하였다. 커뮤니티 컴퓨팅에서 사용자는 개인의 선호도나 취향에 따라 행동들이 연관성이 있음을 이용하였고 시간에 따라 사용자 행동의 차이를 반영함으로써 보다 정확한 패턴 인식을 가능하게 하였다. 또한 예기치 못한 행동에 대한 처리를 위해 우선순위를 사용하여 이를 처리하였다. Microsoft Visual C++로 구현된 프로그램을 통하여 기존의 아무런 기법이 없을 때와 본 논문에서 제안하는 기법을 적용하였을 때를 비교하였을 때 사용자에게 서비스를 제공함에 있어 성능이 향상됨을 확인하였다. 비록 제안하는 기법이 행동 테이블이나 패턴 테이블을 생성하고 저장함에 있어서 추가적인 메모리를 요구하지만 그 크기가 매우 크지는 않기 때문에 이는 문제가 되지 않을 것으로 예상된다. 제안하는 기법을 적절히 적용한다면, 실시간으로 요구 반응을 필요로 하는 커뮤니티 컴퓨팅이나 혹은 유비쿼터스 환경에서 큰 도움이 될 거라 예상된다.

## 참 고 문 헌

- [1] 조위덕, Ubiquitous Computing Paradigm, 유비쿼터스 컴퓨팅사업단, 2005년 5월
- [2] 김동욱, 이정태, 류기열, “커뮤니티 컴퓨팅 어플리케이션 개발을 지원하기 위한 이클립스 기반의 통합개발환경,” 한국컴퓨터종합학술대회 논문집, Vol.33 No.1(c), 2006
- [3] 김기영, 박혜성, 노경우, 김석윤, “커뮤니티 컴퓨팅 환경에서의 멤버 생존시간 최적화 알고리즘 연구,” 대한전기학회 논문지 57권 7호, pp.1273-1278, Jul. 2008
- [4] 강경란, 김민구, 이정태, 정유나, 조위덕, 김현숙, “커뮤니티 컴퓨팅 : 협업 기반 환경 자동 적응의 컴퓨팅 모델,” 정보과학회 논문지 제24권 제12호, Dec. 2006