

시간적 추론이 적용된 위치 기반 서비스 (Location based Service with Temporal Reasoning)

김 제 민 [†] 박 영 택 ^{**}
(Je-Min Kim) (Young-Tack Park)

요 약 차세대 정보통신 기술의 가장 중요한 패러다임으로 '유비쿼터스 컴퓨팅'이 새롭게 주목 받고 있다. 유비쿼터스 컴퓨팅 서비스 시스템을 개발하기 위한 중요한 초점 중의 하나는 이동 객체(사용자)의 시간과 이벤트의 관계를 파악하고 이동 객체의 위치 데이터로부터 시공간 이동 패턴을 추론하는 것이다. 본 논문에서는 유비쿼터스 환경 내에서 사용자에게 시간과 관련된 서비스를 적절히 제공하기 위해서 다음과 같은 연구를 한다. 첫째, 서비스 관점에서의 시간적 추론(Temporal Reasoning)이다. 둘째, 사용자의 취향을 기록하기 위한 시간적 추론(Temporal Reasoning)이다. 각 사용자들은 각자의 취향을 가지고 있으며 이는 시간과 밀접한 관계를 가지고 있다. 시간과 관련된 사용자의 취향을 기록된 각 사용자 프로파일을 기반으로 서비스 지원 시스템은 적절한 서비스를 제공할 수 있다.

키워드 : 시간적 추론, 유비쿼터스 환경, 위치 기반 서비스, 사용자 취향, 시간 규칙

Abstract 'Ubiquitous Computing' is the most important paradigm of the next generation Information-Communication technology. The one of important problems to develop ubiquitous computing service system get hold of relations between times of transfer objects and events of transfer objects. Another problem is what reason transfer-pattern through location data of transfer objects. In this paper, we propose an approach to offer temporal-relation service in ubiquitous computing environment. The first is temporal reasoning in service viewpoint. The second is temporal reasoning to record user's preference. Users have preferences that are closely connected with time. These preferences are recorded at user profile. Therefore, the user profile-based ubiquitous service system can offer suitable service to users.

Key words : Temporal Reasoning, Ubiquitous Environment, Location Based Service, User Preference, Time Axiom

1. 서 론

유비쿼터스 컴퓨팅이 지향하는 "Every Computing"은 사용자가 이동하여 장소가 변하거나, 휴대 및 사용하는 장비가 바뀌거나, 시간이 지나더라도, 지속적으로 연결되어 원하는 작업을 수행할 수 있게 해주는 연속적인 컴퓨팅 환경의 구현을 목표로 한다[1]. 유비쿼터스 환경을 구현하기 위해 여러 가지 유비쿼터스 컴퓨팅 서비스 시스템이 개발되고 있으며, 이 중 위치 기반 서비스(Location Based Service)는 개인의 위치를 파악하여 사용자에게 개인화되고 동적인 서비스를 제공한다. 위치

기반 서비스를 개발하는데 있어 대두되는 중요한 문제 중의 하나는 현재의 시간과 현재 사용자에게 서비스된 이벤트의 관계를 파악하고 사용자의 위치 이동 데이터로부터 시공간 이동 패턴을 알아내는 것이다. 이를 위해 서비스되는 이벤트들의 시간적 관계와 공간적 관계가 고려되어야 하며, 시간 논리(Temporal Logic)를 사용하여 시간과 이벤트 및 이벤트에 따라 변하는 사용자의 행위간의 관계를 추론할 필요가 있다[2].

위치 기반 서비스 시스템은 시간적 추론(Temporal Reasoning) 및 개인화 시스템과의 다음과 같은 관계를 보여 준다. 위치 기반 서비스 시스템이 사용자가 제공 받은 서비스와 시간 및 공간의 관계를 관계연산자로 변환하여 추론 엔진에 전송하면, 시간과 공간 및 이벤트간의 관계가 추론되어, 사용자 프로파일 학습에 기반이 되는 사용자 히스토리 파일을 생성한다. 이렇게 기록된 사용자 히스토리 파일을 기반으로 개인화 시스템은 각 사

· 본 논문은 숭실대학교의 지원을 받았습니다.

[†] 학생회원 : 숭실대학교 컴퓨터학부
kimjemins@hotmail.com

^{**} 종신회원 : 숭실대학교 컴퓨터학부 교수
park@computing.ssu.ac.kr

논문접수 : 2004년 8월 19일

심사완료 : 2006년 1월 26일

사용자의 시간에 따른 서비스 취향을 학습하여 사용자 프로파일을 생성한다.

본 논문은 유비쿼터스 환경 내에서 사용자에게 시간이 고려된 서비스를 적절히 제공하기 위해서 다음과 같은 연구를 한다. 첫째, 서비스 관점에서의 시간적 추론(Temporal Reasoning)이다. 각 사용자들은 각자의 취향을 가지고 있으며 이는 시간과 밀접한 관계를 가지고 있다. 시간과 관련된 사용자의 취향이 기록된 각 사용자 프로파일을 기반으로 유비쿼터스 컴퓨팅 서비스 시스템은 적절한 서비스를 제공할 수 있다. 둘째, 사용자의 취향을 기록하기 위한 시간적 추론(Temporal Reasoning)이다. 기록된 내용들은 사용자 프로파일(User Profile)을 생성하는데 도움을 준다.

본 논문의 구성은 다음과 같다. 2장에서는 위치 기반 서비스 시스템에 관해서 설명하고, 이와 관련된 기존 연구를 정리한다. 3장에서는 시간과 이벤트 및 공간에 따라 변하는 사용자의 행위간의 관계를 표현하기 위한 시간 논리(Temporal Logic)에 관하여 설명한다. 4장과 5장에서는 실제 유비쿼터스 환경에서 사용자가 받을 수 있는 가상의 시간 기반 서비스(Time based Service) 시나리오를 정의하고, 이러한 서비스가 가능하도록 사용자 프로파일을 생성하는데 기반이 되는 사용자 히스토리에 관해 설명한다. 6장에서는 사용자에게 적절한 시간 기반 서비스를 제공하기 위해 정의된 시간 규칙을 테스트하고, 7장에서는 본 논문이 제안한 “시간적 추론이 적용된 유비쿼터스 서비스”에 대한 결론과 향후 연구할 내용을 제시한다.

2. 위치 기반 서비스(Location Based Service)

본 장에서는 위치 기반 기술이 적용된 기존의 유비쿼터스 컴퓨팅 서비스에 관한 연구를 살펴보고 본 연구와의 차이점을 기술한다. 그리고 현재 위치 기반 서비스의 위치 추적 기술을 간단히 소개하고, 유비쿼터스 환경에서의 각 사물에 대한 기하학적 모델을 구축하여 동적 사물이 이동하는 단계마다 사물들 간의 위치 관계를 Qualitative 모델로 구축한 후 이를 기반으로 위치 기반 서비스 시행하는 시스템에 대해 설명한다.

2.1 관련 연구

위치 기반 기술이 적용된 대표적인 유비쿼터스 컴퓨팅 서비스에 관한 연구로 AT&T 연구소가 제시한 “Follow-Me”서비스를 들 수 있다[3]. “Follow-Me”서비스는 센서 역할을 하는 모바일 장치를 소지한 사용자가 일정한 장소에서 다른 특정한 곳으로 이동하였을 경우, 이동된 장소에 존재하는 여러 장치들 중에서 현재 사용자와 가장 가까운 장치를 통하여 이 전 장소에서 제공 받았던 서비스를 계속 제공 받는 것이다.

“Follow-Me”서비스 외에 위치 기반 기술을 적용한 유비쿼터스 서비스 기술로는 RFID 태그기반으로 원격지에 존재하는 상대방의 움직임을 인지하는 MIT 미디어 연구소의 헤비타트(Habitat) 프로젝트와 UWB 기반으로 회사나 병원 내에서 직원 또는 환자들의 위치를 실시간으로 인지하여 일정 조정이나 위험 감지와 같은 서비스를 제공하는 UbiSense사의 UbiSense Platform[4]이 있다.

“Follow-Me”서비스를 포함한 다른 유비쿼터스 컴퓨팅 서비스 시스템에서 위치 기반 기술은 하나의 시스템으로 동작하여 독자적인 서비스를 제공 한다기보다는 사용자와 사물의 위치를 감지해 줌으로써 유비쿼터스 컴퓨팅 시스템이 서비스를 받아야 하는 객체에게 서비스를 제공하는 객체를 적절하게 연결하도록 발판을 마련해준다. 그래서 이러한 관련 연구 프로젝트들은 연구 범위에서 시간적인 요소가 제외되어 있으며, 제공되는 서비스도 수동적이다. 사람마다 시간과 장소에 따라 원하는 서비스가 각각 다르기 때문에 능동적인 서비스를 제공하기 위해서는 그 사람의 위치뿐만 아니라 상대적인 시간 역시 중요하다. 본 논문에서는 유비쿼터스 컴퓨팅 환경에서 개인의 위치를 파악하여 사용자에게 보다 개인화되고 동적인 서비스를 능동적으로 제공 하기 위해 기존 위치 기반 기술에 시간적 추론을 적용하고자 한다.

2.2 위치 기반 서비스의 위치 추적 과 플랫폼

현재 위치 기반 서비스에서 위치 확인 방법으로는 독자적인 새로운 네트워크를 이용하는 방법 및 GPS를 이용하는 방법으로 진행되고 있다. 이러한 기술 개발 방법에 대해 위치확인을 수행하는 일반적인 방법은 네트워크 기반 방식, 단말기 기반 방식, 이들을 혼합하여 사용하는 혼합 방식으로 분류할 수 있다. 이러한 방식에 대해 위치 결정을 하는 방법으로는 기지국에서 단말기로부터 들어오는 신호의 도래각을 측정하여 단말기의 위치를 구하는 AOA(Angle of Arrival) 방법, 전파의 도달 시간을 이용하는 방법으로 전파 전달 시간을 측정하여 위치를 구하는 TOA(Time of Arrival)방법, 두 개의 기지국으로부터 전파 도달 시각의 상대적인 차를 이용하는TDOA(Time Difference of Arrival) 방법, 단말기에 GPS를 부착하여 위치를 결정하는 GPS 방식(WAG : Wireless Assisted GPS) 등이 있다.

위치 기반 서비스는 센서의 구성 방식에 따라서 서로 다른 방식으로 구성된다. 현재 유비쿼터스 환경에서의 센서 구성은 MIT의 Cricket[5]과 같은 분산 구조와 Cambridge의 BAT[3]와 같은 중앙 집중적인 구조로 구분된다. 중앙 집중적인 센서 구조에서는 유비쿼터스 환경 내에 있는 동적, 정적 객체에 대한 모든 위치 기반

상황 인지가 가능하다. 반면에 분산 구조를 가지는 경우에는 위치 추적 보다는 동적 객체의 개인적 자유(Privacy)를 존중하여 자신의 위치를 스스로 전송하도록 시스템이 구축되고 있다.

본 연구의 기반이 되는 위치 기반 서비스 시스템은 중앙 집중적인 센서 네트워크 방식으로부터 위치 추적(Location Tracking)이 이루어진 상태에서 유비쿼터스 사물들의 동적 상황에 대한 시멘틱 상황을 추론하는 위치 기반 서비스를 기반으로 하고 있다[6,7]. 기본적인 유비쿼터스 상황은 캠브리지의 BAT 시스템의 상황과 같지만, BAT기반의 시스템에서는 쿼드 트리 기반의 환경 모델과 공간 인덱싱 방식을 취하여 상황 인지를 수행한 반면에 본 논문에서 기반이 되는 시스템은 Qualitative 방식으로 상황을 모델하고 지식기반 방식을 적용하여 위치 기반 상황 인지를 수행한다. 시스템은 객체 구성관리, 정적 공간 모델, 위치 추적, 상황정보 해석, 상황정보 통합기능으로 구성된다.

먼저 객체 구성 관리는 집이나 사무실 같은 실제 공간에 배치되는 여러 가지 객체(TV, 냉장고, 오디오, 침대 등등)들을 정의하고, 관리한다. 이를 위해 각 객체의 식별정보와 타입정보 및 객체의 2차원적인 크기 (X, Y Size)를 정의한다. 여기에 부가적으로 상황정보 해석을 원활하게 하기 위해 각 객체의 영역(크기 및 타입)을 추가로 정의한다. 정의된 객체 정보들은 객체 속성 테이블에 저장되어 관리되며, 이 정보 들은 해당 객체를 홈맵에 아이콘으로 표현하거나 상황 정보를 해석하기 위한 Qualitative 모델을 구성하는 데 쓰인다.

정적 공간 모델은 그래픽 사용자 인터페이스(Graphic User Interface: GUI)를 통해서 집이나 사무실 같은 일정한 공간을 홈맵으로 표현하고, 정의된 홈맵에 여러 객체들을 초기 배치하는 기능이다. 이를 위해 하나의 정적인 공간(집, 사무실)을 표현하는 정보(그래픽 파일 또는 카드 파일)를 먼저 입력 받은 후 공간의 좌표 값을 설정한다. 여기에 홈맵에 배치될 객체의 식별정보와 타입정보 및 2차원 위치를 표현하는 X, Y좌표와 크기를 위치 센서와 객체-속성 테이블로부터 입력 받는다. 입력된 위치 정보와 크기 및 영역 정보를 바탕으로 각 객체는 홈맵으로 매핑되어 아이콘으로 표현된다.

위치 추적은 동적 객체의 2차원 위치를 표현하는 X, Y좌표를 위치 센서로부터 입력 받는다. 이 위치 정보는 정적 공간 모델링에 의해 작성된 홈맵으로 매핑되어 해당 객체의 아이콘으로 표현되며 객체의 이동은 GUI상에서 해당 객체의 아이콘의 이동으로 표현 된다. 이 외에도 위치 추적은 수동-동적 객체를 선택하여 시간 축을 기준으로 위치 정보를 보여주는 기능을 포함한다. 이 기능은 수동적으로 이동이 가능한 정적 객체의 위치 정

보를 입력 받아서 홈맵을 수정하는 기능을 제공한다.

상황 정보 해석은 동적 객체가 정적 공간 내에서 움직였을 경우, 공간 내에서의 동적 객체의 이동 상황을 인지하여 공간 모델로 표현하고, 이동 객체와 일반 객체의 위치 관계를 인지하여 Qualitative Model로 표현하는 기능이다. 따라서 동적 객체가 시간에 따라 위치되는 공간에 따라서 공간 모델정보가 구성되며, 이 공간 모델은 (동적 객체의 ID, 객체 영역, 시간)으로 구성된다. 여기에 더불어서 각 공간에 배치된 객체와 동적 객체의 관계(홍길동이 3시에 TV영역에 접근)를 인지하여 Qualitative Model을 구성한다. Qualitative Model의 연산자는 Positive-Overlap, Containment, Negative-Overlap으로 나누어지며, (객체 ID, 동적 객체의 ID, 시간)으로 구성된다.

마지막으로 상황 정보 통합은 일정한 시간 동안 발생하는 상황 정보 들을 에이전트가 이해할 수 있는 형태로 통합하는 기능이다. 입력 받은 Qualitative Model 자료를 바탕으로 에이전트가 이해할 수 있는 형태인 관계 모델(Relation Model)로 표현한다. 관계 모델은 (동적 객체의 행위 연산자[동적 객체와 객체의 관계를 표현하는 단어, 객체 ID, 동적 객체ID, 상황 시작 시간, 상황 종료 시간])의 형태로 표현된다.

3. 시간 논리

본 장에서는 사용자에게 적절한 위치 기반 서비스를 제공하기 위해서 시간에 따라 변하는 이벤트와 사용자의 행위간의 관계를 표현하는 시간 논리(Temporal Logic)에 관하여 설명한다. Allen, J.F은 행위와 시간간의 관계를 시간 논리(Temporal logic)로 표현함으로써 시간적 추론(Temporal Reasoning)이 가능하도록 하였다[8-10]. Allen, J.F가 제안한 시간 논리는 현재 시멘틱 웹 분야에 적용되어 웹 페이지를 구성하는 여러 메타데이터들 간의 시간 관계를 표현한다[11]. 다음 그림 1은 Allen이 정의한 시간 논리를 통하여 시간과 시간 간격간의 관계를 보여준다.

시간 논리에서 Instance는 하나의 시간 포인트를 나타내고, Interval은 시간 간격을 나타낸다. 시간 간격들의 관계를 표현하기 위해서는 Proper-Interval이라는 개념을 사용되는데, Proper-Interval은 시작점(Start-Point)과 끝점(End-Point)이 항상 존재하며, 이 두 점이 같지 않아야 성립된다. 시간 간격간의 관계는 크게 7가지로 표현된다. Before는 하나의 Proper-Interval의 끝점이 다른 Proper-Interval의 시작점보다 앞서 있을 때를, Interval-Equals는 하나의 Proper-Interval의 시작점과 끝점이 다른 Proper-Interval의 시작점과 끝점에 일치할 때를, Meet는 하나의 Proper-Interval의 끝점과

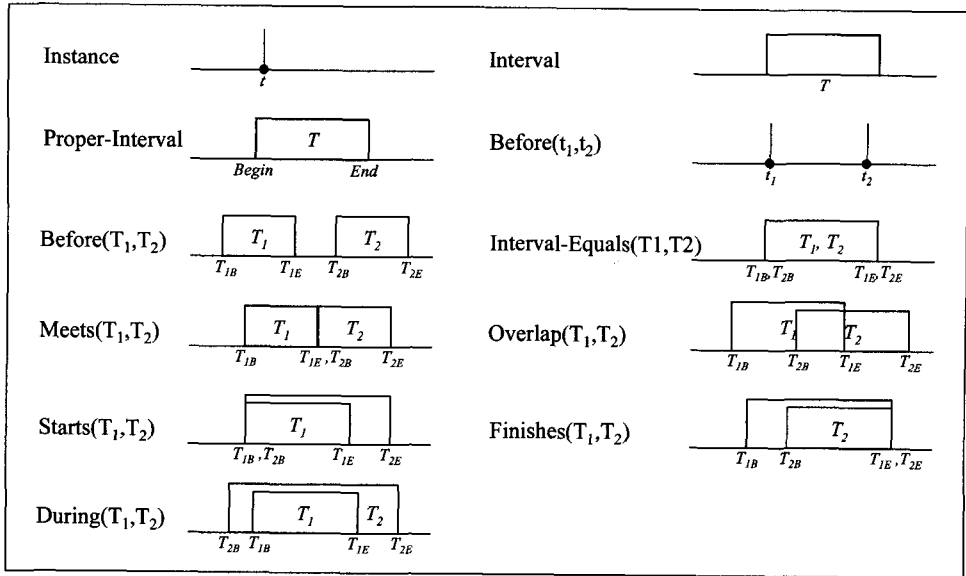


그림 1 시간과 시간 간격 간의 관계

다른 Proper-Interval의 시작점이 일치할 때를, Overlap은 하나의 Proper-Interval의 끝점이 다른 Proper-Interval의 시작점과 끝점의 사이에 존재하고 하나의 Proper-Interval의 시작점과 끝점 사이의 다른 Proper-Interval의 시작점이 존재할 때를, Start는 두 Proper-Interval의 시작점은 같고 끝점이 서로 다를 때를, Finish는 두 Proper-Interval의 시작점은 서로 다르고 끝점이 같은 때를, During은 하나의 Proper-Interval의 시작점과 끝점 사이에 다른 Proper-Interval이 존재할 때를 표현한다. 이러한 시간 간격들의 관계는 시간적 추론 시스템에 시간 규칙(Time-Axiom)으로 적용된다. 이러한 시간 규칙과 위치 기반 서비스 시스템에서 생성된 관계 연산자(Relation Operation)를 바탕으로 시간에 따라 변하는 이벤트와 사용자의 행위간의 관계를 추론한다. 관계 연산자는 $[t_1 \sim t_6](bed, park, on)$ 과 같이 표현되는데, 이는 사용자가 특정 이벤트를 벌인 시작 시간과 끝 시간을 의미한다. 따라서, 시간 별로 나열된 사용자와 객체의 관계를 분석하여, 시간 별로 서로 영키있는 사용자간 또는 사용자와 객체간의 관계를 추론할 수 있다.

3.1 시간적 추론 엔진

본 논문에서의 시간적 추론을 위한 기본적인 추론 전략은 단일화(Unification)을 기반으로 Rete 알고리즘을 사용하는 전방향 추론(Forward Chaining) 기법이다. 전방향 추론(Forward Chaining)은 $A \rightarrow B$ 즉 조건 A가 Working Memory에 있으면 B를 수행한다. 따라서 당연히 연역법(Deduction)에 속한다.

단일화는 두 개의 문장이 동일한 표현이 될 수 있도록 변수부호에 항을 대치하는 것을 말한다. 예를 들어서 $(\forall x) W(x)$ 문장에서 $W(A)$ 라는 문장을 생성하기 위해서는 변수 x 에 문제의 대상 영역에 있는 상수부호 A가 할당되어야 한다. 즉, $W(x)$ 라는 문장과 $W(A)$ 문장이 동일하게 되기 위해서는 x 가 A로 대치되어야 한다. 이러한 단일화 과정에서 여러 개의 항이 x 에 대치될 수도 있다. 변수를 대응하는 항으로 대체해서 얻어진 표현을 대치문 이라고 하는데, $W(A)$ 는 $W(x)$ 문장에서 얻어진 대치문 이 된다.

Rete 알고리즘(Forgy, 1982)은 전방향 추론 규칙 기반 시스템의 속도를 향상시키기 위한 만들어졌다. Rete 알고리즘은 순환이 없는 방향성 그래프(Acyclic Directed Graph)상에 규칙에 관한 정보를 저장함으로써 속도를 향상시키는 패턴 매치 알고리즘이다. 즉 그래프의 모든 노드 상에서 모든 규칙에 대한 사실(Facts)들을 매치시키는 것이 아니고, 변화된 사실들에 대해서만 매치시킨다. 따라서 각 노드에서 변화가 없었던 정적인 데이터는 무시되기 때문에 사실들의 매치 속도는 크게 향상된다. Rete 알고리즘이 적용된 시간적 추론 엔진은 시간 규칙들의 조건들을 하나씩 분리하여 루트 노드를 제외한 노드들에 표현한다. 즉, 루트 노드로부터 최하위 노드까지의 경로를 통해서 각 시간 규칙의 조건들이 표현되며 루트 노드로부터 현 위치의 노드까지 각 경로에 저장된 조건을 만족하는 사실(Fact)에 대한 정보를 각각의 노드에 저장한다. 이것은 추론엔진을 구동했을 때의 시간 중복성(Temporal Redundancy)을 줄여주는 장

점이 있다.

4. 서비스 시나리오

유비쿼터스 컴퓨팅을 논할 때 흔히 집, 자동차, 회사 등의 모든 기기에 내장된 컴퓨터를 네트워크화하여 사용자의 모든 작업 및 행동에 활용하는 것을 볼 수 있다. 또한 어디에서든지 필요한 정보를 얻을 수 있다는 점을 강조하고 개인을 위한 맞춤형 서비스에 초점이 맞춰져 있다는 것을 알 수 있다. 본 장에서는 실제 유비쿼터스 환경에서 사용자가 받을 수 있는 시간적 추론이 적용된 위치 기반 서비스에 대한 예를 간단한 시나리오를 통해 설명한다.

다음 시나리오는 위치기반 서비스가 적용된 홈 네트워크 환경에서 사용자 프로파일을 이용한 지능형 서비스의 흐름을 보여준다. Hong은 매일 일어나자마자 냉장고로 가서 우유를 마신 후 화장실로 가서 샤워를 한다. 샤워를 마친 후 TV 아침 뉴스를 시청하고, 오디오를 켜서 음악 방송을 듣는다. 음악 방송으로 어느 정도 심신이 맑아진 Hong은 컴퓨터를 부팅하여 하루 일과를 시작하고, 작업 중간에 탁자 의자에서 음료수를 마시면서 휴식을 취한다.

Hong의 이러한 반복된 일과는 저장되어 Hong의 취향을 담은 프로파일로 학습된다. 즉 시간 별로 이루어지는 여러 객체와 Hong의 인터랙션은 시간적 추론과 이벤트 추론을 통하여 통합 상황 정보(Relation Model)로 만들어져서 사용자 히스토리 파일에 저장되며, 학습 엔진은 Hong의 사용자 히스토리 파일을 바탕으로 Hong에게 서비스할 이벤트 패턴을 학습하여 Hong의 사용자

프로파일 생성한다.

생성된 사용자 프로파일을 바탕으로 서비스 에이전트는 위치기반 서비스와 연계하여 다음과 같은 서비스를 차례로 수행한다.

- [t6]에 거실의 등불을 켤다.
- [t8]에 화장실 샤워기의 물 온도를 따뜻하게 한다.
- [t9~t20]에 화장실의 등불을 켤다.
- [t22~t35]에 TV를 켜고, Hong이 좋아하는 프로그램이 시작되는 채널을 세팅
- [t38~t50]에 오디오를 켜고, Hong이 좋아하는 프로그램이 시작되는 주파수를 세팅
- [t51]에 컴퓨터를 미리 부팅
- [t130]에 컴퓨터를 대기 모드로 세팅

Hong은 자신에게 맞춰진 서비스를 제공 받음으로써 일반적인 위치기반 서비스보다 진화된 홈 네트워크 환경에서의 생활을 영위할 수 있다.

5. 사용자 히스토리

위의 서비스는 시나리오 사용자의 시간에 따른 이벤트 패턴을 사용자 프로파일 형태로 학습했다가 이를 이용하여 여러 가지 시간 관련 서비스를 사용자에게 제공하는 예를 보인다. 즉, 시간적 추론이 적용된 위치 기반 서비스를 실행하기 위해서는 사용자 프로파일의 필요하다. 사용자 프로파일을 학습하기 위해서는 가장 먼저 수행되어야 하는 것은 사용자가 취하는 행동들을 포괄적으로 탐지하는 것이다.

본 논문에서는 위치 기반 서비스를 이용하는 사용자의 모든 행위를 탐지하여 저장하는 방법으로 관계 연산

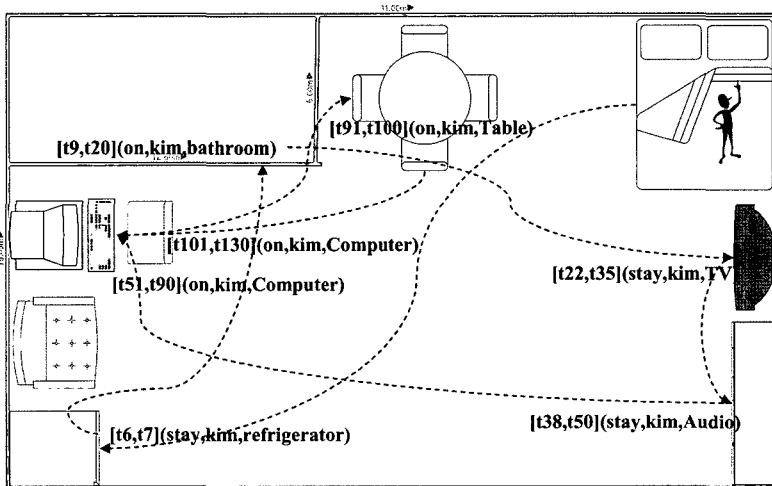


그림 2 사용자가 시간대 별로 받은 서비스의 예: LBS의 시공간 이벤트 데이터와 시간적 추론으로부터 생성된 사용자 프로파일을 사용하여 지정된 시간에 적절한 서비스를 자동으로 실행한다.

자(Relation Operation)와 의미해석 블록을 기반으로 하는 사용자 히스토리 파일(User History File)을 사용하였다. 사용자 히스토리 파일은 일반적으로 위치 기반 서비스 시스템을 통해서 사용자 에이전트가 위치하는 서버에 매우 작은 텍스트 파일로 저장되는 형태이다. 이렇게 저장된 사용자 히스토리 파일을 통해서 사용자가 접근했던 객체에 대한 행동이나 사용자가 제공받았던 서비스에 대한 정보들을 저장 할 수 있다. 또한 사용자 히스토리 파일은 사용자가 일정한 공간에서의 행위와 제공받은 서비스를 시간 별로 저장이 되기 때문에 누가, 언제, 어떤 일 또는 어떤 서비스를 제공 받았는지를 알 수 있다.

관계 연산자(Relation Operation)와 의미해석 블록을 통해서 얻어진 정보는 사용자 히스토리 파일이 쉽게 사용자에 대한 형식화된 기록정보를 구축하기 위해서 필요한 정보만을 골라서 저장한다. 이렇게 정보들을 정제함으로써 사용자 프로파일을 학습하는 엔진은 불필요한 학습을 피할 수 있다. 생성된 사용자 히스토리 파일은 사용자 별로 나눈 후 클러스터링과 학습 과정을 실행하여 각 사용자 별로 사용자 프로파일을 생성한다. 사용자 프로파일은 [Object_ID, Person_ID, startTime, endTime, Event, Relation, Object_Status]의 속성을 갖는 테이블 형태로 표현된다.

6. 실험

본 논문에서 제안하는 시간적 추론이 적용된 위치 기반 서비스는 유비쿼터스 환경을 지원하는 서비스 시스템중의 하나인 위치 기반 서비스 시스템에 적용되어 사용자에게 시간적 요소가 고려된 위치 기반 서비스를 제공할 수 있다. 이와 같은 기능은 시간이 흐름에 따라 수시로 변하는 사용자의 행위와 사용자가 제공받은 서비스의 패턴을 파악하고, 이러한 사실들을 정량화된 데이터로 정제하여 사용자 프로파일을 학습함으로써 가능하다. 사용자 프로파일은 시간적 요소를 바탕으로 둔 시간적 추론(Temporal Reasoning)을 실행하는 추론 엔진이 정확하게 구현되어야 하고 사용자, 시간, 서비스, 사용자 행위간의 관계가 정확히 파악된 사용자 히스토리 파일이 구축되어야 가능하다. 그래서 본 논문이 제안하는 시간적 추론이 적용된 위치 기반 서비스의 실행 가능성을 평가하기 위해 위치 기반 서비스에 연동하기 위해 구현된 시간적 추론 엔진(Temporal Reasoning Engine)과 이를 바탕으로 한 사용자 히스토리 파일 생성에 대해 실험해 보았다.

본 실험을 위해 구현된 위치 기반 서비스 시스템은 센서 네트워크 기반의 위치 기반 서비스를 보여주는 시뮬레이터로서 2004년 숭실대학교 인공지능 연구실에서

구축되었다. 본 시뮬레이터의 그래픽 사용자 인터페이스(Graphic User Interface: GUI)를 제외한 모든 모듈은 JAVA로 구현하였으며, 그래픽 사용자 인터페이스 부분은 JViews를 사용하여 구현하였다. 본 시뮬레이터는 센서 네트워크 시스템에서 전송한 사용자의 위치를 받아들여 사용자의 위치에 따라 제공하는 서비스를 그래픽 사용자 인터페이스 상으로 보여주고 관계 연산자(Relation Operator)를 생성한다. 또한 위치 기반 서비스로부터 관계 연산자를 받아들여 시간적 추론을 실행하는 엔진을 구축하기 위해 JESS(Java Expert System Shell) [12]를 사용하였다. JESS는 썬 마이크로 시스템에서 제작한 규칙-기반 전문가 시스템 저작 도구이며, 자바 환경에서 구동된다. 전문가 시스템 (Expert System)은 전문가가 가지고 있는 지식을 인위적으로 컴퓨터에게 부여하여 그 방면에 비전문가라 할지라도 그러한 전문가의 지식을 이용하여 상호 대화를 통하여 원하는 결과를 얻는 일종의 자문형 컴퓨터 시스템이다. JESS는 규칙을 바탕으로 한 추론 엔진을 제공하고, 추론 엔진에서 사용하는 사실(Fact)과 규칙(Rule)들을 묘사해 준다. 이러한 규칙 기반 프로그램은 수백 혹은 수천 개의 규칙을 가지고 있으며 규칙 베이스에 내장된 데이터들을 규칙에 계속적으로 적용한다.

본 실험은 3명의 사용자가 유비쿼터스 서비스가 적용된 지능형 홈 환경에서 겪는 20개의 상황에 대해 시뮬레이션을 실행한 후, 생성되는 관계 연산자들을 바탕으로 시간적 추론 엔진이 의미해석 블록과 사용자 히스토리 파일을 생성하는 과정을 보여준다. 표 1은 위치 기반 서비스 시스템에 서로 다른 시간 별로 인지된 상황들과 이러한 상황을 관계 연산자로 표현한 것이다.

이와 같은 관계 연산자들은 시간적 규칙이 적용된 시간적 추론 엔진에 적용하기 위해서, 추론을 실행하는데 필요한 시간적 사실들로 표현 되어야 한다. 이와 같은 시간적 사실들은 JESS로 정의된 시간적 규칙(Temporal Axiom)에 적용된다. 본 논문에서는 시간적 추론을 실행하기 위한 다음과 같은 기본 사실들을 설정하였다. 서비스나 사용자의 행위가 시작된 시간과 끝난 시간을 하나의 시간 객체(Time-Instance)로 표현하고, 서비스나 사용자의 행위가 실행된 시간을 하나의 시간 간격(Time-Interval)으로 표현했으며, 시간 간격과 시간 객체들의 관계를 Time-Between으로 표현한다. 또한 Begins와 Ends를 이용하여 각 시간 간격의 시작점과 끝점을 표현한다. 예를 들어 "park이 t1 에서부터 t6까지 침대에 누워 있었다."라는 상황(Context)은 관계 연산자 "stay(on,bed, park,t1,t6)"로 표현되며, 이러한 명령어는 "(instance (time-instance t1)), (instance (time-instance t6)), (interval (time-interval T1)), (time-

표 1 시간적 추론이 적용될 상황과 관계 연산자의 예

NO	상황	Relation Operation
1	park과 kim 두 사람이 같은 시간대에 침대에 누워있다.	[t1 ~t6](bed,park,on) [t1 ~t6](bed,kim,on)
2	park이 일정 시간대에 소파에 앉아 있다.	[t1 ~t6](bed,park,on)
3	park이 TV에서 나온 시간이 소파에 앉으로 들어간 시간보다 앞에 있다	[t8 ~t10](tv,park,stay) [t12 ~t20](sofa,park,on)
4	park이 화장실에 kim보다 먼저 가고, kim이 choi보다 화장실에 먼저 갔다.	[t22 ~t25](toilet,park,on) [t28 ~t30](toilet,kim,on) [t34 ~t50](toilet,choi,on)

between (time-interval T1) (begin-instance t1) (end-instance t6)), (begins (begin-instance t1) (time-interval T1)), (ends (end-instance t6) (time-interval T1)), (time-span(time-interval T1) (time-event Event1))” 라는 초기 사실로 변환되어 작업 메모리에 적재된다.

본 논문을 위해 구축된 JESS기반의 규칙 기반 시스템은 작업 메모리에 적재된 초기 사실들을 바탕으로 그림 3과 같이 규칙 베이스 내에 정의된 시간 규칙 (Time-Axiom)을 이용하여 시간적으로 연관된 사용자와 객체 및 각 사용자간의 관계를 추론한다.

추론된 사실 역시 작업 메모리에 적재되기 때문에 작업 메모리의 데이터가 방대해지게 된다. 따라서, 이러한 문제점을 보완하여 실제 사용되는 시간적 추론 결과물만을 추출하기 위해 의미 해석 블록을 생성한다. 이는 작업 메모리에 존재하는 모든 사실 중 시간적 관계를 나타내는 사실과 이러한 사실에 연관된 서비스(Event)를 연결하여 블록화 시킨 것이다. 이때 각 서비스는 (Event)는 어떤 사용자가 어떠한 객체와 어떠한 관계를 가지고 있는지를 리스트 형태로 저장한다. 이처럼 생성

된 의미 해석 블록은 위치 기반 서비스 시스템을 통하여 사용자에게 시간과 연관된 서비스를 제공받은 물론이고 사용자 프로파일을 구축하기 위해 사용되는 사용자 히스토리 파일을 만드는 중요한 자료로 쓰인다. 본 실험에서 관계 명령어를 시간적 규칙이 적용된 추론엔진에 적용한 결과 표 2와 같이 시간적 추론이 적용된 새로운 상황과 이러한 상황을 표현하는 의미 해석 블록을 얻을 수 있었다. 이와 같은 의미 해석 블록과 기존의 관계 명령어를 바탕으로 사용자 히스토리 파일이 최종적으로 추출되었다.

사용자 히스토리 파일은 시간과 장소에 따라 어떠한 서비스를 받았는가에 대해 기록되기 때문에, 사용자 히스토리 파일을 바탕으로 시간 또는 장소에 따른 서비스 클러스터링 과정과 각 사용자에 대한 서비스 패턴 학습 과정을 거쳐서 사용자 프로파일이 생성될 수 있으며, 이러한 사용자 프로파일을 바탕으로 유비쿼터스 컴퓨팅 서비스 시스템은 사용자에게 보다 적절한 서비스를 제공할 수 있을 것이다. 서비스 클러스터링 과정과 각 사용자에 대한 서비스 패턴 학습과정에 대한 설명은 본

```

(defrule instance **
  (instance (time-instance ?tinst))
  =>
  (assert (temporal-entity (time-temporal-entity ?tinst)))
)
(defrule interval **
  (interval (time-interval ?tinst))
  =>
  (assert (temporal-entity (time-temporal-entity ?tinst)))
)
(defrule begins **
  (begins (begin-instance ?inst) (time-interval ?intv))
  =>
  (assert (instance (time-instance ?inst)))
  (assert (temporal-entity (time-temporal-entity ?intv)))
)
(defrule ends **
  (ends (end-instance ?inst) (time-interval ?intv))
  =>
  (assert (instance (time-instance ?inst)))
  (assert (temporal-entity (time-temporal-entity ?intv)))
)
(defrule time-between **
  (begins (begin-instance ?inst1) (time-interval ?intv1))
  (ends (end-instance ?inst2) (time-interval ?intv2))
  (test (neq ?inst1 ?inst2))
  =>
  (assert (time-between (time-interval ?intv)
    (begin-instance ?inst1)
    (end-instance ?inst2)))
)
)

(defrule before00 **
  (ends (end-instance ?inst1) (time-interval ?intv1))
  (begins (begin-instance ?inst2) (time-interval ?intv2))
  (time-interval ?intv2)
  (before (t_bef ?inst1) (t_laft ?inst2))
  (test (neq ?inst1 ?inst2))
  =>
  (assert (before (t_bef ?intv1) (t_laft ?intv2)))
)
(defrule interval_before00 **
  (proper-interval (prop_interval ?intv1))
  (proper-interval (prop_interval ?intv2))
  (before (t_bef ?intv1) (t_laft ?intv2))
  (test (neq ?intv1 ?intv2))
  =>
  (assert (int-before (interval_bef ?intv1)
    (interval_laft ?intv2)))
)
(defrule interval_meet00 **
  (proper-interval (prop_interval ?intv1))
  (proper-interval (prop_interval ?intv2))
  (ends (end-instance ?inst1) (time-interval ?intv1))
  (begins (begin-instance ?inst) (time-interval ?intv2))
  (test (neq ?intv1 ?intv2))
  =>
  (assert (int-meet (interval_met1 ?intv1)
    (interval_met2 ?intv2)))
)
)
  
```

그림 3 JESS로 정의된 시간 규칙

표 2 시간적 추론 결과와 의미 해석 블록의 예

No	추론된 상황	의미해석 블록
1	park과 kim 두 사람이 침대에 올라간 시간이 같다.	intervalEquals((T1,Event1),(T2,Event2)) Event1(bed,park,on) Event2(bed,kim,on)
2	park이 침대에 올라간 시간은 t1이고 내려간 시간은 t6임을 알 수 있다.	timeBetween((Event1,t1,t2)) Event1(bed,park,on)
3	park은 tv를 켜 후 소파에 앉았다.	timeBefore((T1,Event1) (T2,Event2)) - Event1(tv,park,stay) Event2(sofa,park,on)
4	park은 choi보다 화장실에 먼저 간다.	timeBefore((T1,Event1) (T2,Event2)) Event1(toilet,park,on) Event2(toilet,choi,on)

논문의 범위를 넘기 때문에 생략하도록 한다.

7. 결론 및 향후 연구

본 논문은 유비쿼터스 컴퓨팅 환경에서 효과적인 위치 기반 서비스를 제공하도록 이동 객체(사용자)의 시간과 이벤트의 관계를 파악하고 위치 이동 데이터로부터 시간간 이동 패턴 분석에 대한 연구를 주 목적으로 하고 있다. 이는 사용자에게 시간적 요소가 고려되는 서비스를 제공할 수 있게 한다. 따라서 유비쿼터스 환경 내에서 사용자에게 시간이 고려된 서비스를 적절히 제공하기 위해서 다음과 같은 연구를 진행하였다. 첫째, 서비스 관점에서의 시간적 추론(Temporal Reasoning)이다. 둘째, 사용자의 취향을 기록하기 위한 시간적 추론(Temporal Reasoning)이다. 위 두 연구 사항의 공통점은 사용자 프로파일(User Profile)의 존재이다. 즉, 각 사용자 개개인에 맞는 시간적 요소가 적용된 유비쿼터스 서비스를 제공하기 위해서는 사용자의 취향 및 제공받은 서비스들이 시간 간격(Time Interval)별로 기록되어있는 사용자 프로파일이 필요하며, 이러한 사용자 프로파일을 구축하기 위해서는 시간적 추론이 필요하다. 이를 위해 본 논문에서는 시간적 추론 엔진을(Temporal Reasoning Engine) 구현하여 유비쿼터스 컴퓨팅 서비스 시스템중의 하나인 위치 기반 서비스 시스템에 적용하였다.

본 논문은 유비쿼터스 환경에서 사용자에게 적절한 서비스를 제공하기 위해 시간 추론(Temporal Reasoning)을 통해서 시간 별로 나타나는 사용자와 객체간의 인터랙션을 히스토리 파일로 저장한다. 이러한 히스토리 파일은 사용자의 버릇이나 취향을 담은 사용자 프로파일을 생성하는데 바탕이 된다. 일반적으로 위치 기반 서비스 시스템에서 제공하는 서비스는 사용자의 시간에 따라 변하는 버릇이나 취향을 고려하지 않으므로 불완전(Incomplete), 부정확(Incorrect)한 특성을 가지게 된다. 이러한 문제점을 해결하기 위해서 사용자의 습관과 취향이 반영된 사용자 프로파일을 시스템에 적용하여 보다 정확한 이벤트 서비스가 가능하도록 해야 한다.

정확한 사용자 프로파일을 구축하기 위해서 향후에는 시간과 공간을 고려한 위치기반 서비스 데이터들을 적절하게 클러스터링 하는 방법과 클러스터링 된 데이터들 사이에서 정확한 패턴을 학습하는 방법을 연구 할 것이다.

또한 위치 기반 서비스 시스템이 프로파일을 사용하여 사용자에게 서비스를 실행할 경우 사용자들은 개개인의 버릇이나 취향 등이 시시각각으로 변할 수 있기 때문에 자동으로 제공하는 서비스에 대한 취소 의사를 전달할 수 있어야 할 것이며, 사용자의 서비스 취소의사에 따라서 사용자 프로파일의 내용도 유동적으로 바뀌어야 한다. 이러한 점을 해결하기 위해서는 사용자의 몸짓이나 언어 인식하는 기술을 적용하는 문제에 대해서도 추후 연구될 것이다.

참고 문헌

- [1] Abowd G, Mynatt E, "Charting past, present, and future research in ubiquitous computing," ACM Transactions on Computer Human Interaction, Vol.7, No.1, pp.29-58, March 2000.
- [2] James F Allen, George Ferguson, "Actions and Events in Interval Temporal Logic," In Oliveira Stock (ed.), Spatial and Temporal Reasoning, Kluwer Academic Publishers, pp.205-245, 1994.
- [3] Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggles, Andy Ward, Andy Hopper, "Implementing a Sentient Computing System," IEEE Computer, pp.50-56, August 2001.
- [4] Andy Ward, Jay Cadman, "Deploying Commercial Location-Aware Systems," Presented at Ubicomp 2003, 2003.12.
- [5] Nissanka B. Priyantha, Anit Chakraborty, Hari Balakrishnan, "The Cricket Location-Support System," 6th ACM International Conference on Mobile Computing and Networking (ACM MOBICOM), Boston, MA, pp.32-43, August 2000.
- [6] Young-Tack Park, "Location-based Context-Aware System," 2004 ICAT Conference, pp.469-472, 2004.
- [7] Jun-Sang Park, Young-Tack Park, "Location-based Context Awareness in Ubiquitous Environ-

- ment," ICAT Workshop, pp.161-164, 2004.
- [8] Jerry Hobbs, James Pustejovsky, "Annotating and Reasoning about Time and Events," <http://www.cs.rochester.edu/~ferguson/daml/>, DAML-Time Homepage.
- [9] Frank Schilder, Christopher Habel, "From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages," in Proceedings of the ACL-2001 Workshop on Temporal and Spatial Information Processing, ACL-2001. Toulouse, France, pp.65-72, 2001.7.
- [10] James F Allen, "Planning as Temporal Reasoning," reprinted from the Proc. of the Second International Conference on Principles of Knowledge Representation and Reasoning, Cambridge, MA April, 1991.
- [11] Jerry R. Hobbs, "A DAML Ontology of Time," <http://www.cs.rochester.edu/~ferguson/daml/>, DAML-Time Homepage.
- [12] Ernest F.H, "Jess In Action", Manning Publications Co, 2003.



김 제 민

2001년 숭실대학교 컴퓨터학과 졸업(학사). 2004년 숭실대학교 대학원 컴퓨터학과 졸업(석사). 2004년~현재 숭실대학교 대학원 컴퓨터학과 박사과정. 관심분야는 인공지능, 게임 AI, 시멘틱 웹



박 영 택

1978년 서울대학교 전자공학과 졸업(학사). 1980년 KAIST 전산학 졸업(석사) 1992년 Univ. of Illinois at Urbana-Champaign 졸업(박사). 1981년~현재 숭실대학교 컴퓨터학과 교수. 관심분야는 인공지능, 에이전트, 전문가 시스템