상황인지 기반 사용자 맞춤 자원 공유 시스템[1)]

A Situation-Aware based Personalized Resource Sharing System

이용대, 박원익, 김영국[2)], 강지훈
충남대학교 컴퓨터공학과

Yong-Dae Lee, Won-Ik Park,
Young-Kuk Kim and Ji-Hoon Kang
Dept. of Computer Engineering,
Chungnam National University

요약

유비쿼터스 시대에는 다양하고 수많은 컴퓨팅 자원들이 곳곳에 존재한다. 이런 유비쿼터스 환경에서 분산되어져 있는 자원들을 공유하는 연구가 최근 많이 진행되어져 왔다. 하지만 이런 연구들은 사용자의 상황과 취향을 고려하지 않은 문제점을 가지고 있다. 따라서 본 논문에서는 사용자의 상황을 고려한 사용자 맞춤 자원 공유 시스템을 제안한다. 제안한 시스템은 사용자의 스케줄 정보와 시간, 위치 정보와 같은 컨텍스트 정보를 이용하여 사용자의 상황을 인지하며 자원에 대한 사용자 선호도를 가중치 기반으로 적용하여 사용자 맞춤형 자원을 추천해준다. 본 논문에서는 회의 시나리오를 통해 제안한 상황인지 기반 사용자 맞춤 자원 공유 시스템을 검증한다.

Abstract

In ubiquitous environments, there are a large number of various mobile devices everywhere. Recently, many studies have been research on distributed resources sharing in this ubiquitous environments. However, the problems with this researches are no considering user's situation and preference. Therefore in this paper, we suggest the personalized resource sharing system considering user's situation and preference. Our system is aware of user's situation by using user's schedule information and a context information such as the time and location information and recommends personalized resources by using the simple weighted scheme. We prove our situation-aware based personalized resource sharing system by using a meeting scenario.

1. 서 론

유비쿼터스 시대에는 사용자들이 원하는 서비스를 시간과 장소, 네트워크나 컴퓨터의 종류에 구애 받지 않고 서비스를 받을 수 있어야 한다. 이러한 환경을 유비쿼터스 지능형 공간 즉, USS(Ubiquitous Smart Space)라 부른다[1,2]. 이런 USS 환경에는 다양하고 수많은 컴퓨팅 자원들이 존재한다.

최근 USS 환경에서 자원 공유에 대한 연구가 많이 진행되어져 왔다. 자원 공유 시스템은 사용자가 개인 모바일 단말기를 가지고 USS 환경에 들어가게 되면 USS 환경에 비치된 자원이나, 다른 사용자의 자원을 공유하여 서비스를 제공받는 시스템을 말한다[3,4]. 이런 시스템은 PDA나 스마트 폰과 같은 모바일 단말기의 제한적인 리소스 문제를 주변의 자원을 이용하여 해결하고자 나온 시스템이다.

하지만 기존의 자원 공유 시스템은 사용자의 상황과 취향을 고려하지 않고 수동적인 조작으로 서비스 되는 단점이 있다.

이러한 단점을 해결하기 위해 본 논문에서 제안한 시스템은 컨텍스트 정보와 사용자의 스케줄 정보를 이용하여 추론된 상황 정보와 자원에 대한 사용자 취향 정보를 고려하여 상황인지 기반의 사용자 맞춤 자원 공유 서비스를 제공한다.

우리는 상황 추론을 위해 규칙기반 추론 엔진인 JESS를 이용하였고, 사용자 맞춤 서비스를 위해 가중치 우선 순위(Weight-Priority Order, WPO) 알고리즘을 적용하였다.

본 논문의 구성은 다음과 같다. 2장에서 네트워크로 연결된 환경에서 장치 간 자원 공유 기술에 대해 알아보고 3장에서는 본 논문에서 제안한 상황 인지 기반 사용자 맞춤 자원 공유 시스템 설계와 구현에 대해 설명한다. 4장에서는 회의 시나리오를 통해 제안한 시스템에 대한 예를 제시한다. 마지막으로 5장에서는 결론 및 향후 과제를 제시한다.

2. 관련 연구

USS에 배치된 자원들을 공유하는 기술은 유비쿼터스 환경

1) 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스컴퓨팅및네트워크원천기반기술사업의 08B3-O1-30S 과제로 지원된 것임.

2) 교신저자

에서 필요한 기술 중에 하나이다. 자원 공유 기술은 디바이스들 간의 통신을 통해 사용자가 원하는 서비스를 제공하는 것이다. 디바이스 간 통신을 위해 제안된 표준 인터페이스로는 UPnP와 JINI가 있다. 이 두 기술은 홈 네트워크 환경에서 컴퓨터, 가전제품 등의 자원을 공유하고 제어하는 홈네트워크 미들웨어 기술이다.

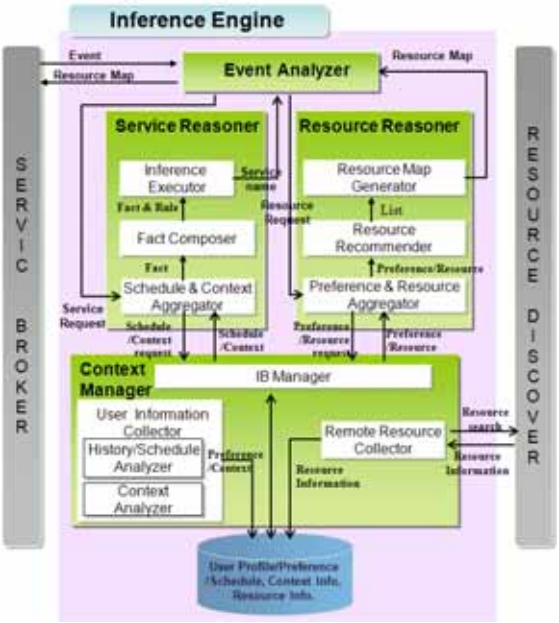
UPnP(Universal Plug and Play)는 여러 장소에 분산되어 있는 네트워크 장치와 서비스 간의 편리한 통신방법을 제공한다. PnP(Plug and Plag)를 지원하여 장비의 접속과 분리를 자동으로 인지하여, 보다 다양한 장비에 적용할 수 있게 확장한 기술이다. UPnP는 관리자가 없는 네트워크 환경에서 사용자의 작업 없이 표준화된 방법으로 쉽게 장비간의 연결이나 장비와 인터넷의 연결을 제공한다. UPnP는 장비간 1대 1통신(Peer-to-Peer)을 기반으로 하고 있으며 TCP/IP 프로토콜을 이용하여 그 구조를 정의하고 있다. 작은 리소스로도 이용이 가능하고, IP가 없는 가정장비에 대해서 SCP 프로토콜을 통해 브릿지로 연결 할 수 있다. 웹 기반 프로토콜이기 때문에 운영체제와 무관하게 사용할 수 있다[5].

JINI는 Sun Microsystems에서 개발한 기술로써 사용자들과 그 사용자들이 이용하려고 하는 장비와 같은 자원의 유기적인 결합을 지원하는 분산 시스템이다. JINI는 UPnP처럼 네트워크를 통해 사용자가 원하는 자원을 찾거나 사용자하고자 할 때, 관리자의 개입 없이 유연하게 동작하는 것을 목표로 하고 있다. 그러나 JINI는 자바 기반 미들웨어로써 자원은 Java machine이 있어야 하는 제약 조건이 따른다. JINI의 자원 공유 절차는 UPnP와 유사하지만 단일한 보안, 자원, 실행 환경을 가지고 전 네트워크가 동일기능에 동일 프로그램을 사용하는 마치 한 대의 컴퓨터 안에서 일어나는 일을 네트워크로 펼쳐놓은 듯한 모습을 하고 있는게 차이점이다[6].

본 논문에서 제안한 시스템은 UPnP, JINI와 같이 자원을 공유하는 관점에서는 유사하지만, 시간이나 위치 정보, 사용자의 스케줄 정보와 선호도 정보를 이용하여, 사용자의 상황을 추론하고 사용자의 상황에 맞는 디바이스를 공유해 준다는 점에서 다르다고 볼 수 있다.

3. 상황인지 기반 사용자 맞춤 자원 공유 시스템

그림 1은 상황기반 사용자 맞춤 서비스 제공 시스템의 전체 구성도 이다. 각 모듈은 다음과 같은 역할을 맡고 있다. Service Broker는 사용자 인터페이스를 제공하며, 사용자로부터 이벤트 명령을 받아 Inference Engine에게 보내고 그 결과를 처리하는 역할을 하고 있다. 여기서 이벤트는 사용자 원하는 서비스에 대한 이벤트도 있지만, 특정 시간이 되었거나 특정 장소에 위치하고 있을 때도 포함 된다. Inference Engine의 Event Analyzer는 Service Broker로부터 온 이벤트를 분석하여 상황 추론이 필요하면 Service Reasoner를 통해 사용자의 스케줄과 켄텍스트 정보를 이용하여, 적절한 서비스를 찾게 된다. 사용자로부터 서비스 요청이 직접 들어오면 Resource Reasoner를 통해 사용자의 선호도에 따른 자원을 추천한다. Inference Engine의 Context Manager는 크게 개인 프로파일 정보와 컨텍스트 정보, 자원 정보를 관리하며, Service Reasoner와 Resource Reasoner에서 필요로 하는 정보를 제공해 주는 일을 담당 한다. Resource Discovery는 주변의 공유 할 수 있는 자원들에 대해 수집하는 일을 한다. 본 논문에서는 Inference Engine 부분을 중심으로 설명하겠다.



▶▶ 그림 1. 상황인지 기반 사용자 맞춤 자원 공유 시스템 구성도

Inference Engine은 사용자로부터 온 이벤트에 대해 분석하는 Event Analyze과 서비스를 추론하는 Service Reasoner, 사용자에게 적절한 자원을 추천해 주는 Resource Reasoner 그리고, 사용자의 스케줄 정보, 선호도 정보, 상황정보 및 주변의 자원(디바이스 등)을 관리하는 Context Manager로 크게 4개의 모듈로 구성된다. 그리고 사용자의 정보와 주변의 자원을 저장하는 DB를 가지고 있다. 앞으로 각 모듈에 대해 알아보겠다.

3.1 Event Analyzer

사용자로부터 워드 프로세스 서비스나 프리젠테이션 서비스와 같이 직접 서비스에 대한 요청이 들어오거나 특정 장소에

위치하였을 때 또는 본 논문에서 제공하는 프레임워크를 실행했을 때와 같이 사용자가 직접 서비스 요청이 아닌 이벤트에 대해 분석하는 일을 담당한다.

3.2 Service Reasoner

Service Reasoner는 사용자의 위치와 시간, 스케줄 정보 등을 이용하여 사용자의 상황을 추론하는 역할을 한다. SChedule&Context Aggregator 클래스에서는 이벤트에 대해 관련된 정보를 Context Manager로부터 가져온다. Fact Composer는 규칙기반 추론기인 Jess의 입력 파일을 생성하는 곳으로 사용자의 스케줄 정보와 위치, 시간과 같은 컨텍스트 정보, 사용자가 정의한 Rule을 가지고 구성한다.

Inference Executor는 Fact Composer를 통해 나온 결과를 입력하여, Jess 추론을 하고, 그 결과를 Event Analyzer로 보낸다. 그림 3은 본 논문에서 서비스 추론에 이용한 Jess Rule을 보여주고 있다.

```
;;서비스 추론 - presentation service
(defrule PresentaionService
  (Environment_Context (currentDate ?cDate) (currentTime
?cTime)(currentLocation ?cLocation))
  (User_Schedule (scheduleDate ?cDate) (scheduleStartTime
?sTime) (scheduleEndTime ?eTime) (scheduleLocation
?cLocation) (scheduleContents meeting)(scheduleRole announcer))
  (test (<= ?cTime ?eTime))
  (test (>= ?cTime ?sTime))
  =>
 (assert (ServiceName "Presentation service")))
```

▶▶ 그림 3. Example Rule-based Inference

그림 3의 Rule은 발표 상황 추론을 나타낸 것으로 예를 들어 사용자 스케줄 정보에 2008년 4월 10일 오후 3시부터 5시까지 코엑스 인도양홀에서 회의가 있고, 이 사용자는 발표자의 역할을 맡고 있다.

```
<Schedule>
  <Date>
    <Year> 2008 </Year>
    <Month> 4 </Month>
    <Day> 10 </Day>
  </Date>
  <Time>
    <StartTime> 15:00 </StartTime>
    <EndTime> 17:00 </EndTime>
  </Time>
  <Location> 코엑스 인도양홀 </Location>
  <Contents> 회의 </Contents>
  <Role> 발표 </Role>
</Schedule>
```

▶▶ 그림 4. XML 스케줄 정보 예

그리고 Context 정보로부터 현재 사용자의 위치는 코엑스 인도양홀에 들어왔고, 현재 날짜는 4월 10일, 시간은 오후 3시 5분이면 Rule 정의에 의해 이 사용자에게 Presentation 서비스가 필요한 상황이란 것이 추론 된다. 여기서 스케줄 정보는 XML로 표현 되어있고, 파싱을 통해 필요한 데이터를 추출한다. 시간과 위치 정보는 USS 환경에 설치된 센서로부터 값을 얻어온다고 가정한다. 그림4는 사용자의 스케줄 정보의 예이다.

3.3 Resource Reasoner

직접 사용자로부터 또는 Service Reasoner를 통해 나온 서비스 요청에 대해 필요한 자원들을 추론하는 일을 한다. 먼저 Context Manager를 통해 서비스에 필요한 자원들의 정보와 사용자의 선호도(preference) 정보를 가져온다. Resource Recommender 클래스에서 그 정보들을 가지고 사용자에게 적합한 자원들의 리스트를 만든다. 최종 Resource Map을 만들어 Event Analyzer를 통해 Service Broker로 전달된다. 그 Resource Map 정보는 주변의 자원들과 공유하는데 사용된다.

여기서 사용한 자원 추천 알고리즘은 가중치 우선 순위 알고리즘(Weight-Priority Order)을 사용하였다. 이 알고리즘은 한 아이템의 여러 속성들에 대해 사용자가 평가한 선호도 점수를 가중치로 계산하여 합한 결과가 그 아이템의 평가 점수가 된다. 그림 5는 WPO 알고리즘이다.

$$r_i : Item의\ i속성에\ 대한\ 사용자\ 가중치\ 값 \quad (0 \le r_i \le 1)$$

$$AR_i \begin{cases} r_i & : 자원\ 속성\ 정보와\ 일치할\ 경우 \\ 0 & : 자원\ 속성\ 정보와\ 일치하지\ 않은\ 경우 \end{cases}$$

$$IR = \sum_{i=1}^{N} AR_i$$

▶▶ 그림 5. 가중치 우선 순위 알고리즘

Item에 대한 사용자 평가 정보와 주변 장치로부터 수집된 item에 대한 속성 정보가 표 1과 같다고 하자.

[표 1] 주변 자원 정보와 사용자 선호도 정보

Item 속성에 대한 사용자 평가

Item	속성1	속성2	속성3	속성4
사용자 평가(ri)	0.7	0.6	0.8	0.3

Item별 속성 일치 여부

Item1	○	○	×	×
Item2	×	○	○	×
Item3	×	×	○	○

본 논문에서 제시한 WPO 알고리즘을 이용하여 계산하면 item2, item1, item3이 사용자에게 적합한 순서로 나오게 된다.

3.4 Context Manager

Context Manager는 크게 세 종류의 정보를 관리한다. 첫 번째는 사용자 정보이고 두 번째는 주변의 자원 정보이다. 마지막은 시간과 위치 정보인 context 정보이다. 사용자 정보는 사용자 사용 이력(History)의 분석을 통한 선호도 정보와 사용자의 스케줄 정보로 구성된다.

주변 자원 정보는 Remote Resource Collector를 통해 주변 자원들의 정보를 실시간으로 업데이트 한다. 이 정보들은 DB에 저장되며, IB Manager를 통해 다른 모듈들의 정보 요청에 대해 해당 정보들을 제공하는 역할을 한다.

4. 실험 시나리오

회의 상황을 본 논문에서 제안한 시스템에 적용 하여 실험하였다. 여기서 몇 가지 가정을 하겠다. 사용자는 미리 자신의 모바일 단말기에 일정(스케줄) 정보를 세팅하고 있어야 한다. 또한 USS 환경의 고정된 디바이스 장치들은 자신의 자원들을 사용자에게 제공 할 수 있어야 한다.

사용자 A는 오후 3시부터 5시까지 회의 일정이 잡혀 있고, 발표를 맡고 있다. A는 자신의 모바일 단말기를 가지고 회의실에 입장한다. 오후 3시가 되면 모바일 단말기에 이미 실행되어져 있는 발표 화면이 자동으로 회의실 빔 프로젝터를 통해 나오고, 주변의 키보드나, 마우스 또는 프린젠터, 마이크, 스피커 등이 공유되어, A는 발표를 위한 준비 과정 없이 원활하게 발표를 할 수 있다. 이때 A의 선호 정보를 이용하여 평상시 A가 자주 사용했던 디바이스를 추천하게 된다. 다른 사용자 B는 A와 같이 같은 시간 같은 장소에서 회의 일정이 잡혀 있고, 회의 내용을 기록하는 서기의 일을 담당했다고 하자. B는 A와 역할이 다르기 때문에 B의 역할인 서기에 맞는 문서편집 서비스를 제공해 주어야 한다. 그래서 주변의 모니터, 키보드, 마우스 등의 디바이스를 제공해 준다. B 역시 사용 이력을 분석한 선호 정보를 이용하여 B 사용자가 좋아하는 디바이스가 추천되어진다. 시나리오와 관련된 Rule은 그림 3과 같고, 추천 알고리즘은 그림 5의 WPO를 이용하였다.

Inference Engine를 테스트한 결과 화면은 다음 그림 6과 같으며, 이 결과는 사용자의 선호 정보와 주변 디바이스의 종류에 의해 달라진다.

▶▶ 그림 6. Inference Engine 결과

5. 결론 및 향후 과제

유비쿼터스 시대에는 다양한 자원들이 집, 회사, 병원, 학교 등 곳곳에 존재하고, 사용자는 PDA나 스마트 폰과 같은 모바일 단말기를 이용하여 서비스를 받게 될 것이다. 본 논문은 이런 유비쿼터스 환경에서 주위에 있는 자원들을 효과적으로 공유하는 시스템을 제안 하였다. 특히 사용자의 스케줄 정보를 이용하여 사용자의 상황을 추론하고 그 상황에 적절한 서비스를 제공하였다. 이 때 사용자의 선호도 정보를 고려하여 사용자 맞춤 서비스를 제공하였다.

본 논문에서 제안한 시스템은 모바일 단말기의 부족한 자원 문제를 해결할 뿐만 아니라 사용자 맞춤 서비스를 제공함으로써 유비쿼터스 환경에서 자원 공유 시스템의 가이드 역할을 할 것으로 기대된다. 앞으로 우리는 여러 분야에 적용 할 수 있는 시스템으로 발전시킬 것이며, 사용자의 정보를 효과적으로 관리하고, 다양한 추천 알고리즘을 적용하여 사용자의 만족도를 높이겠다.

▌참 고 문 헌 ▌

[1] M. Weiser, "Some computer science issues in ubiquitous computing,"Communications of the ACM, Vol. 36, Issue 7, pp.75-84, July 2003.

[2] 조위덕, "뉴 트렌드의 리딩: 유비쿼터스 지능 공간," 한국지능정보시스템학회 2006 춘계학술대회논문집, pp.3-17, 2006.

[3] "About GAIA Project", http://gaia.cs.uiuc.edu/

[4] 최환수, 강선희, 이용대, 장서윤, 박원익, 박종현, 김영국, 강지훈, "유비쿼터스 환경에서의 상황 기반 디바이스 추천 시스템," 제28회 한국정보처리학회 추계학술발표논문집, pp.903-906, 2007.

[5] G. Bhatti, Z. Sahinoglu, K. A. Peker, J. Guo, and F. Matsubara, "A TV-Centric Home Network to Provide a Unified Access to UPnP and PLC Domains", Proceedings of the 2002 IEEE 4th International Workshop on Networked Applications, pp.234-242, Jan., 2002.

[6] Rahul Gupta, Sumeet, Talwar, Dharma P. Agrawal, "Jini Home Networking: A Step Toward Pervasive Computing", IEEE Computer, Vol. 35, No. B, pp.34-40, Aug, 2002
[7] JESS, the RuleEngine for the JavaTM Platform, http://herzberg.ca.sandia.gov/jess/
[8] Maarten Menken. "Jess Tutorial", December 24, 2002.
[9] UPnP, http://www.upnp.org
[10] Jini, http://www.jini.org
[11] XML Tutorial, http://www.w3schools.com/xml/