

Rescorla-Wagner 모형을 활용한 다중 에이전트 웹서비스 기반 욕구인지 상기 서비스 구축 및 성능분석*

권오병
경희대학교 국제경영학부
(obkwon@khu.ac.kr)

최근호
경희대학교 국제경영학부
(kino4u@khu.ac.kr)

최성철
한동대학교 경영경제학부
(blissray@handong.edu)

개인화된 상기 시스템은 사용자의 현재 상황 정보를 토대로 현재 욕구를 동적이며 선응적으로 파악하여야 한다. 하지만 기존의 욕구 인식 방법론 및 상기시스템 아키텍처들은 이러한 요구 사항을 잘 반영하지 못해왔다. 따라서 본 논문은 에이전트, 시맨틱 웹, 그리고 RFID기반의 상황인지를 활용한 선응적인 욕구 인지 메커니즘을 유력한 유비쿼터스 서비스 지원환경의 하나인 개인화된 상기 시스템에 적용하는 것을 목적으로 한다. 이를 위하여 주된 욕구 인지 이론으로 Rescorla - Wagner 모형을 채택하였다. 또한 본 논문에서 제안하는 방법론의 실현 가능성을 보이기 위해 NAMA (Need Aware Multi-Agent)-RFID라고 하는 프로토타입 시스템을 개발하였다. NAMA는 사용자의 욕구를 인지하기 위해 상황 정보 및 사용자 프로파일과 선호도, 가용 서비스 관련 정보 등을 고려할 수 있으며, 웹 서비스의 형태로 구현된 서비스 집합들을 사용자에게 연결시켜준다. 더욱이 범위성 측면에서의 시스템 성능을 보이기 위해 시뮬레이션을 수행하였으며, 그 결과를 보였다.

논문접수일 : 2005년 11월

게재확정일 : 2005년 11월

교신저자 : 권오병

1. Introduction

최근 유비쿼터스 서비스에 대한 제안과 사용화 시도가 활발히 진행되고 있다. 유비쿼터스 서비스란 유비쿼터스 컴퓨팅의 요소 기술들을 활용하여 언제 어디서나 어떤 기기나 네트워크 하에서도 사용자가 원하는 어떤 정보 서비스 및 지원 서비스라도 제공하는 것이다. 최근에는 유비쿼터스 컴퓨팅의 요소기술들로 센서 및 센서네트워크를 포함한 상황인지 컴퓨팅 기술, 발전된 사용자 인터페이스를 위한 증강현실 기술과 웨어러블 컴퓨팅 기술,

개인화를 가능하게 하는 에이전트 기술과 서비스의 자동화를 위한 시맨틱 웹 기술 및 웹서비스 기술이 주로 언급되고 있다.

유비쿼터스 서비스는 기존 서비스에 비교할 때 선응성과 개인화, 그리고 실시간 서비스로 특징될 수 있는데, 이러한 특징을 잘 나타내는 것 중의 하나로서 상기 시스템(reminder system)이 있다. 상기 시스템이란 특정 사용자가 해야 할 활동을 미리 알려주는 특수한 형태의 개인 정보시스템이다. 특히 상기 시스템은 의료 및 진료 시스템 분야에서 EMR 등 기존의 의료 정보 시스템을 고도화하

* 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크원천기반 기술개발사업의 지원에 의한 것임.

여 자동화된 건강 검진 등 진료의 질(quality of care)를 향상시키기 위한 방법으로 많이 제안되고 있으며, 자동차 활용에 있어서의 안전 운행을 위한 상기 서비스에도 활용되고 있다 [8], [23]. 또한 최근에는 전문가시스템 분야의 일종으로 이동 중 쇼핑이나 광고를 가능하게 하는 지원 서비스로도 제시되고 있다 [11].

이러한 상기 시스템이 성공적으로 활용되기 위해서 Abowd는 다음과 같은 능력이 있어야 한다고 보았다 [1]:

- 간단한 시간 혹은 위치에 관한 것 뿐 아니라 여타의 풍부한 컨텍스트의 사용과 그러한 컨텍스트의 선응적인 제공 능력
- 사용자 혹은 제삼자가 상기할 내용들을 제출할 수 있도록 하는 능력
- 다양한 여러 입력 장치들을 활용하여 상기할 내용들을 창출할 수 있는 능력
- 다양한 장치로부터 상기할 내용들을 사용자의 상태에 맞게 받을 수 있는 능력
- 신호와 묘사 내용을 모두 받아 활용할 수 있는 능력
- 사용자에게 사용 가능한 모든 상기 시스템의 목록을 보여줄 수 있는 능력

현존 시스템은 이러한 기능 모두를 다 만족시켜 주고 있지는 못하는데, 그 주된 이유는 어떤 상황에서 상기 시스템이 가동되어야 할 지에 대해 도움을 주는 컨텍스트에 대한 풍부한 수집 및 활용의 미비에 있다. 또한 사용자가 현재 느끼는 필요가 무엇인지를 추정할 수 있는 능력의 부족이다. 사용자는 자신의 욕구를 사전에 기록해 둘 수는 있지만, 어떤 특정한 상황 하에서 부각되는 욕구가 무엇인지를 인지하기란 쉬운 일이 아니다. 이미 상황인지 기술을 활용한 프로토타입 시스템 내지 사용화된 서비스들이 많이 제안되고 있지만, 아직 사용자의 현재 욕구를 자동으로 인지할 수 있는 방

법론이나 서비스는 존재하지 않는다.

따라서 본 논문의 목적은 특정한 상황 하에서 발견되는 사용자의 욕구가 무엇인지를 자동으로 추정 및 인지할 수 있는 방법론을 개발하는 것이다. 이를 위해 기존의 상황인지 기술 외에 추가적으로 사용자 욕구 인지를 위하여 욕구 인지 추정 메커니즘을 제안하고, 이를 자율적으로 수행할 수 있는 다중 에이전트 기반 시스템을 고려하였다. 또한 에이전트 간의 자동화되고 지능화된 협동을 위해 시맨틱 웹과 웹서비스 기술을 응용하여, 궁극적으로 욕구 인지 상기 시스템(need-aware reminder system)을 개발하고자 하였다. 이를 위해 본 논문의 방법론의 실현가능성을 보이기 위해 NAMA-RFID라고 하는 프로토타입 시스템을 개발하였다.

그런데 다중 에이전트, 시맨틱 웹, 웹서비스, 상황인지 기술 등 유비쿼터스 컴퓨팅 기술의 주요 요소기술들을 문 활용하여 유비쿼터스 서비스를 구현함에 있어 시스템의 성능, 특히 규모성(scalability)의 문제를 검토하지 않을 수 없었다. 그러나 기존의 연구들은 시스템의 활용 가능성에 대해 집중되어 있으며, 규모성과 같은 성능 분석에 대해서는 많이 진전되지 않았다. 따라서 본 논문의 또 하나의 목적은 실제적으로 구현된 유비쿼터스 서비스에 대해서 본격적인 성능 분석을 실시해 보는 것이다. 성능 분석을 통해 유비쿼터스 컴퓨팅 요소 기술을 복합적으로 활용함에 있어서 실제적으로 발생할 수 있는 문제가 무엇인지 인식하는데 유용한 결과를 도출하고자 했다.

본 논문의 구성은 다음과 같다. 욕구 추정을 위해 도입한 연상이론 및 욕구 인식에 대한 기존 연구를 2장에서 언급하였다. 3장에서는 욕구 인지 능력을 가진 유비쿼터스 서비스로서의 상기 시스템인 NAMA-RFID에 대한 개념적인 틀을 포함한 전반적 설명을 소개하였으며, 4장은 서비스의 가

능성을 보이기 위한 구현 방법을 제안하였다. 5장에서는 이러한 서비스가 실제적으로 실현 가능할지를 점검하기 위해 중요한 몇 가지 측면에 대한 실증적인 테스트를 수행하였다. 그리고 결론 및 추후연구방안을 6장에서 밝혔다.

2. 욕구인식(Need Identification)

상거래 서비스에서 욕구 인지에 관해 처음으로 제시된 모델은 소비자 구매 행동 모델(Consumer buying behavior)이며 욕구 인식(Need Identification), 제품 소개, 판매자 소개, 협상, 구매 및 배송, 그리고 평가의 여섯 가지 단계로 이루어져 있다 [20]. 이 중에서 욕구 인식이란 소비자가 어떤 상품에 대한 필요성을 느끼는지를 획득하는 과정이다. 이 단계는 앵겔-블랙웰 모형(Engel-Blackwell model)에서는 문제 인식(Problem Recognition) 단계라고도 불리어진다[5]. 그런데 에이전트 기술 등 정보기술에 의해 지원하려고 하는 시도는 제품 소개, 판매자 소개와 협상에 집중되어 있는 반면, 욕구 인식은 그 중요성에도 불구하고 아직 정보기술에 의하여 지원되지 못하는 것으로 나타났다[16]. 이러한 어려움의 핵심은 욕구 인식에 관련된 사용자의 상황을 동적으로 모을 수 없으며 단지 사전에 입력된 사용자 관련 정보를 가지고 욕구를 파악하는데 있다. 즉, 특정 상황에서 발견되는 욕구가 무엇인지에 대한 인식이 수동적으로 이루어졌다는 것이다. 그렇지 않으면 에이전트는 사용자의 상품 구매가 반복 된다는 것을 예상할 수 있어야 했으며 이전에 많은 데이터를 가지고 있는 것으로 가정해야 한다. 선행 연구에서 이러한 정도의 수준을 가진 욕구 인식을 고려한 에이전트 시스템은 여러 가지가 제안되어 왔으며 그 중 가장 대표

적인 것은 Eyes와 Firefly이다 [12], [14], [16], [17].

첫째는 Amazon.com에서 제공되는 Eyes이다. Eyes는 사용자가 이전에 샀던 책들의 정보를 바탕으로 사용자에게 새로운 책에 대한 추천을 해준다. 예를 들면 사용자가 이전에 특정 저자의 책을 구매 했으면 그 저자의 새로운 책이 나왔을 때 추천해 준다거나 사용자가 이전에 샀던 카테고리에서 새 책이 나오면 그 책 역시 추천해 주는 과정을 거치게 되고 사용자 간의 관계를 공통점등을 분석하여 역시 사용자의 욕구를 추론한다. 이러한 기능을 가진 에이전트들은 FastParts.com의 AutoWatch와 Exicite.com의 Classifieds2000 이라는 프로그램의 Cool Notify등이 있다.

둘째로 Firefly는 Collaborative filtering 이라고 불리는 자동화된 추천 메커니즘을 가지고 있다. 이 시스템은 먼저 고객의 상품 등급을 다른 고객의 것과 비교 한 후 고객에게 가장 가까운 다른 사용자의 욕구를 파악하여 고객에게 추천해 주는 형태를 가진다. 순수한 욕구 인식의 기능이라기 보다 오히려 단순 상품 추천의 성격이 더 강하다.

그런데 그 동안 소비자 구매 행동 모델의 첫 단계인 욕구인식에서의 에이전트 사용은 사용자 정보 수집과 동적인 정보 수집의 한계라는 측면에서 많은 어려움을 가지고 있다. 더욱이 지금까지 제안되어 온 욕구인식 방법은 상황 인지를 고려하지 않았다는 한계점을 가지고 있다.

3. Rescorla-Wagner 모형과 욕구 인지(Need Awareness)

본 연구에서 제안하는 상황인지 기반의 자동화된 욕구 인식은 기존의 수동적이거나 정적인 요구 인식과 구별하기 위하여 욕구 인지(need awareness)

라고 하는 용어를 활용하고자 한다. 욕구 인지 메커니즘은 기본적으로 연상이론에 바탕을 두고 고안되었는데, 이중 가장 대표적인 모형인 Rescorla - Wagner 모형을 채택하였다. Rescorla - Wagner 모형은 지난 수십 년간 인간 행동에 대한 유추를 위해 사용되어 온 인지심리학 분야의 주요 이론의 하나이다. Rescorla - Wagner 모형에 의하면 인과성 발견에 대한 사회학습적 접근방식은 연상 원리와 상황 원리의 작동을 가정한다 [15]. 즉, 연상 강도 (V)는 연상원리에 의한 조건 자극과 상황 원리에 의한 무조건적 자극에 의하여 결정된다고 본다. 또한 연상 강도의 변화량(ΔV)은 조건 자극의 강도(α), 무조건 자극의 강도(β), 그리고 무조건 자극의 연상강도(λ)와 전체적인 연상 강도(V_T)와의 차이의 함수로 나타낸다 [3].

$$\Delta V = \alpha\beta(\lambda - V_T) \quad (1)$$

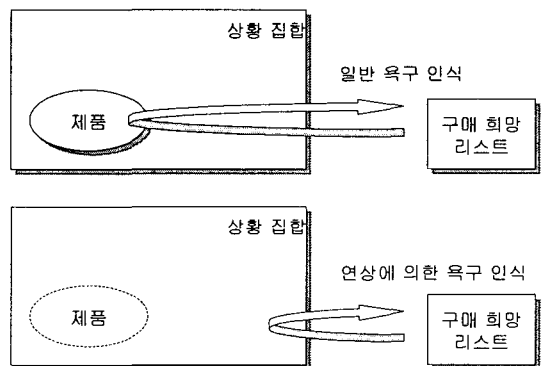
물론 특정 시점에 자극이 없었다면 연상 강도의 변화량은 0이며, 자극은 있었으나 아무런 반응이 없다면 연상 강도의 변화량은,

$$\Delta V = \alpha\beta(0 - V_T) \quad (2)$$

으로 표현된다.

이러한 Rescorla - Wagner 모형은 욕구 인식에 있어서 중요한 시사점을 제공한다. 첫째, 구매자들이 구매에 대한 욕구를 발현할 때 제공되는 제품 정보에 대한 자극뿐 아니라 무조건적으로 제공되는 상황의 자극에 의해서도 영향을 받을 수 있다는 점이다. 둘째, 이 두 종류의 자극은 자체적인 강도를 가지며, 이 또한 연상강도에 영향을 준다는 점이다. 결국 정확한 욕구의 인지는 그러한 욕구를

가지게끔 하는 직접적인 대상물뿐 아니라 그 대상물에 노출되는 시점에 같이 제공되는 상황에 대한 정보도 필요하다는 것이다. 셋째, 연상은 무조건적 자극에 의해서도 이루어질 수 있다는 점이다. 따라서 구매를 위한 직접적인 자극이 없거나 관련도가 낮은 일반적인 웹페이지를 보더라도 무조건적인 자극이 되어 구매욕구를 연상시킬 수도 있다는 점이다. [그림 1]에 의하면 사용자의 구매희망 리스트(to purchase list)가 사용자의 기억이나 혹은 보조기억장치에 있는 경우, 평소에는 제품이라고 하는 자극을 보고 욕구인식을 하게 된다. 이때 욕구인식을 할 당시의 상황 집합도 같이 자극으로 사용자에게 입력될 것이다. 이러한 일반 욕구 인식이 반복되면 추후에는 구매희망리스트에 포함되어있는 제품에 대한 자극을 직접적으로 받지 못하더라도 동일 혹은 유사한 상황집합으로부터의 자극에 노출될 때 사용자는 연상에 의한 욕구인식을 하게 된다는 것이다. 만약에 구매희망 리스트를 사전에 입력 받을 수 있고, 제품에 대한 자극을 받아 일반 욕구를 인식하게 되는 시점에서 상황 집합 내 정보가 사용자의 수동적인 입력 없이 자동으로 인지될 수 있다면, 이러한 연상 이론의 가정 위에서 제품에 대한 직접적인 자극이 없는 단계에라도



[그림 1] 연상의 의한 욕구인식과 욕구인지

상황 집합 내 정보를 자동으로 인지하는 것만으로 사용자의 욕구 인식을 추정할 수 있을 것이다. 그렇다면 이는 욕구인지 프로세스가 될 수 있다.

상황집합을 표현하기 위해 다음과 같은 상황 모형을 사용하였다.

$$C_{ij} = \langle T_{ij}, I_{ij}, L_{ij}, E_{ij} \rangle \quad (3)$$

단, 여기서 i 는 사용자 ID이며, j 는 sensor에 의하여 인식되는 event 일련번호이다. 예를 들어, 10번 ID를 가진 사용자가 20번째 event가 2004-09-17-10:00am에 발생하고, 25번, 30번 사용자와 같이 있으며, WeanHall에 있고, 현재 163.180.99.54라고 하는 IP주소를 가진 PDA를 사용하고 있다면,

$$C_{10,20} = \langle 2004-9-17-10:00am, \{25,30\}, "WeanHall", 163.180.99.54 \rangle \quad (4)$$

가 된다.

본 논문의 욕구 인지는 바로 이러한 연상이론에 바탕을 두고 설계 되었다 [4], [19], [22]. 연상이론에 근거하여 다음과 같은 원칙을 가지고 욕구인지를 지원하는 에이전트를 설계하였다. 첫째, 사전에 선언되는 사용자의 선호도와 구매 희망 리스트에 대한 정보는 연상 강도의 결정에 영향을 준다. 둘째, 구매 관련 자극이 주어지는 시점의 사용자의 컨텍스트 정보는 무조건적 자극으로서 연상에 영향을 준다. 본 논문에서는 사용자가 하고 있는 행동, 위치, 스케줄, 바라보고 있는 사이트의 내용 등을 컨텍스트로 사용하려고 한다. 셋째, 구매 관련 자극이 직접적으로 주어지지 않더라도 간접적인 관련성이 있는 내용만으로도 조건 자극의 강도가 주어지기 때문에 연상이 가능하다.

따라서 욕구 인지 에이전트는 프로파일과 선호

도, 구매 희망 리스트와 같은 사용자 정보를 조건 자극으로 컨텍스트 정보를 무조건적 자극으로 습득하여 사용자의 연상 작용을 예측하는 의사결정을 자율적으로 수행하고 그 결과 사용자에게 적시에 적합한 에이전트 기반 웹서비스를 찾아서 연결하는 일을 수행하게 된다.

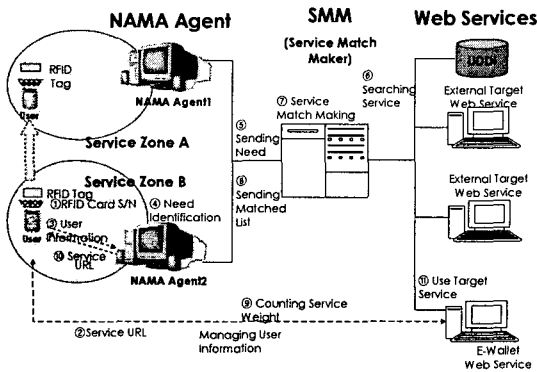
욕구 인식을 자동화하기 위해 특정의 사용자의 상황인지가 가능한 사용자 에이전트를 고려하였다. 이러한 사용자 에이전트는 다음과 같은 능력이 존재할 것을 가정하였다:

- 상황인지 능력: 사용자의 내재적인 욕구를 자동으로 인지하기 위해서는 사용자의 현재 상황, 특히 사용자가 현재 보고 있는 URL 주소와 그 내용을 사용자 요청 없이 인식할 수 있어야 한다.
- 상호작용성: 공유된 사용자의 선호도를 파악하기 위하여 사용자 정보 자원을 탐색하고 획득할 수 있어야 한다.
- 욕구 인식 능력: 파악된 욕구와 현재의 상황을 결합하여 어떠한 자동 서비스를 할 것인지를 스스로 결정하고 사용자 요청 없이도 관련 웹서비스를 호출할 수 있어야 한다.

4. NAMA-RFID: 다중 에이전트 웹서비스 기반 욕구인지 상기 서비스

[그림 2]에서 보듯이 NAMA-RFID 아키텍처는 NAMA 에이전트, Service Match Maker, Web Services 의 총 3가지 부분으로 나누어진다. 이 세 가지 부분의 프로그램들이 서로 협동하여 전체적인 욕구 인지 과정을 통한 서비스 제공을 가능하게 해준다. 각각의 서비스는 스스로 작동하며 서로 협동하고 이전에 자료를 학습 함으로써 최종적인

결과를 산출할 수 있게 도와준다. 각 부분의 대한 구체적인 설명은 아래와 같다.



[그림 2] NAMA-RFID 아키텍처

4.1 NAMA 에이전트

전체 NAMA의 과정 중 가장 핵심이 되고 사용자와 직접적인 상호 교류가 있는 부분이 NAMA 에이전트이다. NAMA 에이전트는 크게 세가지 중요한 역할을 한다. 첫째 NAMA 에이전트는 현재 지역에 대한 상황 관리를 한다. NAMA 에이전트는 TILE(Time, Identity, Location, Entity)라고 하는 네 가지 상황들로 이루어진 상황집합을 활용하여 현재 지역의 어떠한 상황 정보들이 있는지를 관리한다. 이러한 정보들은 추후에 사용자들에게 가장 적합한 정보를 찾는 기본 자료가 되며 상황에 변화에 자동으로 대처 할 수 있는 기능을 제공한다. 두 번째로 NAMA 에이전트는 현재 인식되는 사용자에 대한 정보 요청하며 사용자에 대한 정보를 공유한다. 한 서비스 지역 내에 여러 사용자들이 이동하며 그러한 사용자가 이동하고 있다는 정보를 NAMA 에이전트는 받아서 관리해야 한다. 또한 한 사용자가 다른 지역으로 이동할 때 역시 NAMA 에이전트에 의해 정보를 주고 받는

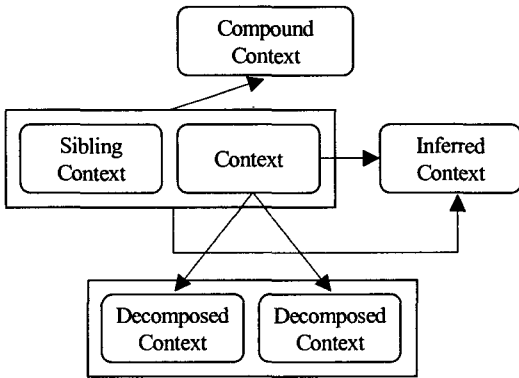
다. 마지막으로 NAMA 에이전트는 이전까지의 학습을 통한 자료를 바탕으로 사용자의 욕구를 파악 사용자에게 적합한 서비스를 제공 하여야 한다. 서비스 지역 내에는 매우 많은 사람들이 움직일 수 있기 때문에 많은 정보를 쌓을 수 있다. 각 사용자의 성향과 그 사용자가 어떤 선택을 하는가에 따라 앞으로 올 사용자에게 제공되는 서비스가 달라진다. 그리고 이러한 정보의 축적을 통한 학습과 그 정보를 바탕으로 한 사례기반추론으로 사용자에게 최적의 서비스를 제공할 수 있다. 이때 사례는 다음 <표 1>과 같이 구성된다.

<표 1> 사례의 구성

문제 (자극)	구매 희망 리스트	문자열의 집합 (제품 리스트)	
	무조건적 자극 (상황 집합)	T	문자열
		I	문자열의 집합
		L	문자열의 집합 (사용자 현재 위치 x, y, z 좌표)
		E	문자열의 집합
조건적 자극 (제품)	문자열 (제품명)		
결과 (반응)	문자열 (행동의 결과)		

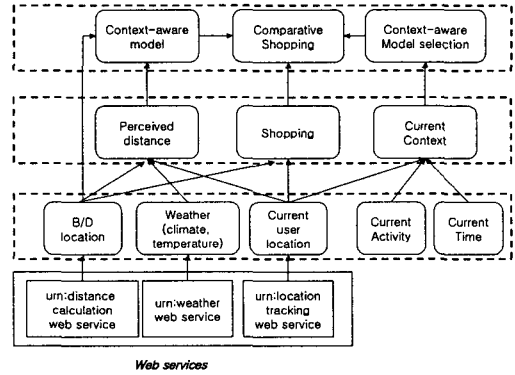
그런데 외부에서 인지하는 상황집합과 사례 베이스에 존재하는 상황집합이 일치하지 않을 수 있다. 즉, 현재 인지 가능한 상황 중에는 사례 베이스에 등록되지 않은 것도 있고, 반대로 사례 베이스 내에는 존재하지만 현재 시스템에서는 인지되지 않는 상황도 존재할 수 있는 것이다. 더욱이 연상 메커니즘을 고려할 때, 현재 인지 가능한 상황은 무조건적 자극에 등록된 상황과 그렇지 않은 상황의 연합으로 이루어진 것도 있다. 이러한 경우 현재 인지한 상황을 분해하여 무조건적 자극에 등록

된 유의한 상황을 인식해야 할 것이다. 예를 들어서 어떤 사람이 현재 쇼핑을 하고 있다고 하는 상황은 그 사람이 현재 서 있는 장소의 이름이라고 하는 상황과 사용자의 현재 위치라고 하는 상황에 의하여 이루어져 있다고 보자. 이때 쇼핑이라고 하는 행위 상황은 현재 장소와 사용자의 현재 위치라고 하는 두 상황을 통해 추론된 상황(inferred context)이다. 한편, 사용자가 현재 보고 있는 특정 URL 웹사이트내의 이미지와 문서는 해당 URL의 분해된 상황(decomposed context)이다. 역으로 URL은 그 사용자가 보고 있는 해당 이미지와 문서의 복합 상황(compound context)라고 볼 수 있다. 그리고 추론 혹은 복합 상황을 이루는 데 같이 활용된 상황을 형제 상황(sibling context)라고 칭한다. 이러한 상황과의 관계성을 보인 것이 [그림 3]이다.



[그림 3] 상황 정보의 관계성

이렇게 NAMA 시스템과 관련된 문제 영역에서 등장하는 상황들의 집합은 하나의 상황 모형을 형성하게 되는데 그 예는 다음 [그림 4]와 같다. 그리고 이러한 상황들의 일부는 웹서비스에 의하여 획득될 수도 있다.



[그림 4] 상황 모형 예

이때 어떤 특정 상황 모형을 전제로 할 때 그러한 상황집합을 통하여 욕구를 인지하기 위한 사례 기반추론 알고리즘은 다음과 같다.

단계1: 상황집합 $C_{ij} = \langle T_{ij}, I_{ij}, L_{ij}, E_{ij} \rangle$ 을 인지한다.

단계2: 각 상황에 대하여 inferred context, compound context, decomposed context를 획득한다. 획득된 상황들을 포함한 확장된 상황집합(augmented context set) $C^A_{ij} = \langle T^A_{ij}, I^A_{ij}, L^A_{ij}, E^A_{ij} \rangle$ 을 구한다.

단계3: 확장된 상황집합과 사례 베이스의 무조건적 자극 상황집합, $C^C_{ij} = \langle T^C_{ij}, I^C_{ij}, L^C_{ij}, E^C_{ij} \rangle$ 사이의 교집합인 $C^I_{ij} = \langle T^I_{ij}, I^I_{ij}, L^I_{ij}, E^I_{ij} \rangle$ 을 구한다.

단계4: $C^I_{ij} = \langle T^I_{ij}, I^I_{ij}, L^I_{ij}, E^I_{ij} \rangle$ 을 당시의 구매희망 리스트와 함께 문제집합으로 인식하고 다음의 함수식을 활용하여 nearest-neighbor retrieval을 수행하여, 가장 유사성이 높은 사례를 선택한다.

$$Similarity(C^I, S) = \sum_{j=1}^n f(C^I_j, S_j) \times w_j \quad (5)$$

단, S는 사례베이스 안에 존재하는 사례의 집합이며, w는 사례 내의 각 feature들의 가중치이다.

4.2 Service Matchmaker

웹서비스 기술은 사용자의 컨텍스트에 따라서 웹서비스의 행동이 동적으로 적응해 나가고 또한 복수의 웹서비스와 복잡한 상호작용을 통한 복합된 서비스 수행의 실현을 지향하고 있다 [9]. 이를 위해 웹서비스 발견과 결합 방법론이 성공적인 차세대 웹서비스의 구현에 중요한 요소로 인식되고 있다 [2]. 그리고 이러한 웹서비스 환경 구축에 서비스 중개자는 매우 중요하다. 서비스 중개자는 중개를 할 수 있는 웹서비스의 선택을 위해 정확한 쿼리(exact match)와 제약중심의 쿼리(constraint-based query)의 두 가지 방법이 있다. 정확한 쿼리가 시간이 많이 소요되는 단점이 있는 반면 제약중심의 쿼리는 먼저 제약조건에 의하여 탐색 공간을 신속하게 줄인 다음 최선의 웹서비스를 찾기 때문에 탐색 공간이 방대하거나 공간의 확장이 크게 이루어지거나 예측 불가능하여 범위가 중요하게 대두될 때 사용되기 위해 고안되었다[7].

현재의 SMM(Service MatchMaker)는 위와 같은 점을 고려하여 설계 되었다. SMM은 세 가지 단계를 거쳐 여러 에이전트들과 통신한다. SMM은 먼저 NAMA 에이전트로부터 사용자의 욕구를 제공 받는다. SMM이 가장 먼저 해야 하는 것이 사용자의 욕구를 바탕으로 적절한 서비스를 찾는 것이기 때문에 NAMA 에이전트의 요청에 의해서 SMM은 서비스가 시작되어야 한다. 다음으로 요청을 받은 SMM은 UDDI 검색을 통해 현재 사용자의 욕구를 충족시켜줄 수 있는 서비스를 제공할

수 있는 데스크 웹서비스를 검색하여야 한다. 에이전트는 이때 다음과 같은 제약식을 바탕으로 제약중심의 쿼리를 하게 된다.

```
Location = {Service ZoneA}
OpenTime >= 09:00
CloseTime <= 20:00
Service Quality >= {Moderate}
```

이런 제약 중심 쿼리를 통해 사용 가능한 서비스를 결정함으로써 NAMA 에이전트가 해야 할 사례 기반 추론의 효율성을 증대 시켜 준다. 마지막으로 SMM은 NAMA 에이전트에게 사용 가능한 서비스 목록을 보냄으로써 자신의 역할을 마치게 된다.

4.3 RFID 기반의 상황인지

NAMA-RFID에서 상황인지는 RFID에 의한 식별을 기반으로 하고 있다. RFID (radio frequency identification)는 소형 반도체 칩을 이용해 사물의 정보를 처리하는 기술을 말한다. 판독·해독기능이 있는 판독기와 고유 정보를 내장한 RFID 태그, 운영 소프트웨어, 네트워크 등으로 구성된 RFID 시스템은 사물에 부착된 얇은 평면 형태의 태그를 식별함으로써 정보를 처리한다. 본 연구에서 RFID를 채택한 이유는 현재 유비쿼터스 기반에서 가장 실현 가능한 접근으로 RFID를 꼽기 때문이다. 또한 정보축적과 발신 기능을 가진 작은 칩을 통해 고주파 신호를 받아 내장된 정보를 전송할 수 있는 기능을 가지고 있으며, 좁쌀 보다 작아 사람의 옷이나 사물, 공간 등 어디에나 부착이 가능하기 때문이다. 더욱이 또한 무선으로 신호를 주고받기 때문에 거리에 제한이 없이 자유롭게 데이터를 스캐닝 할 수 있으며 자동으로 고주파 신호를

인식하여 컨테이너 박스나 팔레트 단위 등에 저장된 수십 개의 제품 정보들을 인식할 수 있다. 따라서 구매 전 단계로서의 욕구인지를 위한 사용자 환경에서 쉽게 구현 가능한 상황인지 방법으로 판단되었다.

4.4 웹서비스

4.4.1 UDDI

NAMA 시스템의 UDDI는 기본적으로 Target Web Service의 정보들의 시맨틱웹 URI를 모아두고 있다. UDDI가 태스크 웹서비스들의 정보를 직접 가지고 있지 않은 것은 그 정보의 관리가 기본적으로 해당 웹 서비스들의 에이전트에 의해서 관리 되어야만 보안성 문제를 해결 할 수 있기 때문이다. 이러한 정보들은 이전에 exXML이나 사용자가 직접 쿼리 하여 얻어내는 방법 보다 에이전트의 자동성과 협력성을 좀 더 향상 시킬 수 있다 [21].

4.4.2 태스크 웹서비스

태스크 웹서비스는 실제로 사용자들에게 제공되는 서비스이다. 태스크 웹서비스는 최종적인 서비스가 제공될 수 있도록 내부적으로 에이전트가 움직이며 사용자의 요청에 반응 한다. 각 서비스는 현재의 서비스의 정보를 가지고 있으며 이를 지속적으로 UDDI에 보내어 자신의 서비스가 존재한다는 것을 SMM에 알릴 수 있다. 또한 현재의 서비스에 대한 사용자의 피드백을 받아 사용자와 상호 교류 함으로써 현재의 서비스를 좀더 나은 방향으로 설계 할 수 있다. 태스크 웹서비스의 정보 자체는 시맨틱 웹으로 관리 되며 에이전트가 이 정보를 관리한다.

4.4.3 E-Wallet 웹서비스

NAMA-RFID 시스템의 개인정보는 관리는 개인화된 사용자 정보 저장 시스템인 E-Wallet 웹서비스를 적용하려고 한다. E-Wallet 웹서비스는 에이전트와 같은 외부 시스템으로부터의 사용자 정보 요청에 대해 선택적으로 허용하며 선택적으로 정보를 제공하는 웹서비스 기반의 에이전트 시스템이다[18]. NAMA-RFID 시스템의 E-Wallet 웹서비스는 시맨틱 웹서비스로써 공개 가능한 개인 정보에 한해 시맨틱웹상에 존재하고, 정보의 관리 및 추론은 에이전트 방식에 의해서 처리 된다. E-Wallet 웹서비스는 4가지 범주로 나누어진다 [6]. 첫째는 정적인 지식(Static Knowledge)로서 가장 전형적인 사용자 프로파일 정보로서 사용자의 이름, E-mail, 집 주소, 현재의 직업 등이 저장 된다. 이러한 정보는 사용자의 욕구를 추론하는 사례 기반 추론의 가장 기본적인 정보를 제공하게 된다. 둘째는 동적인 지식(Dynamic knowledge)로서 사용자의 현재 상황에 따라 바뀌는 정보로서 현재 사용자가 있는 위치, 현재 사용자가 있는 지역의 시간 등이 이것에 포함된다. 욕구를 파악하는 상황에 관한 중요한 정보를 제공한다. 다음으로 서비스 호출 규칙(Service Invocation Rules)이 있다. 이는 사용자의 정보를 바탕으로 에이전트가 현재 사용자가 필요한 서비스를 추론 할 수 있게 해주는 정보이다. 사용자의 선호도 정보나 현재 사용자의 스케줄 정보 등을 포함한다. 이러한 정보들과 사용자의 정적, 동적인 지식을 바탕으로 NAMA 에이전트가 사용자의 욕구를 인지하고 태스크 웹서비스를 호출할 수 있다. 마지막으로 사생활 정보 보안 선호도(Personal Preference for Privacy)로서 사용자가 정보를 공개할 것인지에 대한 것을 저장하고 있는 정보이다. 사용자가 가지고 있는 여

러 정보들이 사용자가 원치 않을 경우나 특정한 에이전트에만 제공하기 원할 경우 이러한 정보들이 보안 선호도가 되어 E-wallet 웹서비스에 저장되어 있다. NAMA 에이전트는 사용자의 여러 정보를 각 사용자의 보안 선호도에 맞춰 다른 에이전트에 제공하게 된다.

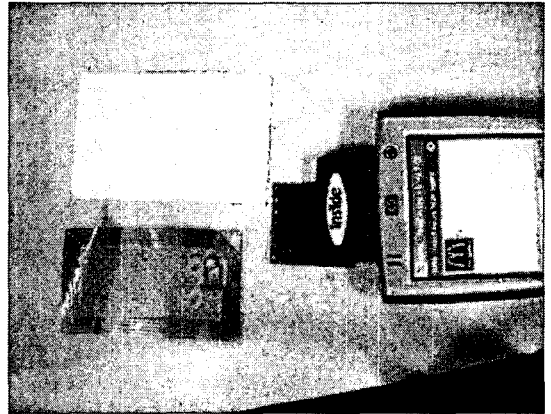
5. 구현

본 논문의 아이디어의 가능성을 보이기 위해 NAMA라고 하는 다중 에이전트 시스템을 개발하였다. NAMA-RFID는 RFID와 시맨틱 웹 기반의 위치 추적 기법을 사용하였다. 온톨로지 생성과 관리를 위해서는 Protégé-2000를 사용했고 JENA 2.0을 사용하여 파싱하였다. PDA내에서 공개하기 어려운 개인 정보를 데이터베이스로 관리하기 위해 Pointbase를 사용하였다. 자바 가상 머신으로 자바 표준 가상 머신인 SUN 사의 JAVA SE 1.4.x를 사용하였고 Pocket PC용 가상머신으로 Jeobe 사의 EVM을 사용하였다. 플랫폼 환경으로서는 Windows XP Professional과 리눅스 기반의 Redhat 9.0을 사용하였다. PDA클라이언트 부분의 효율적인 개발을 위해 Sun의 CDC와 Personal Profile을 준수하는 IBM의 J9를 사용한다. WSDD 5.1 이상의 버전에서 Compile 되었으며 J9의 링크 파일인 jxe와 범용 jar 파일로 실행 파일을 생성한다. 또한 PDA에서 직접적으로 웹서비스를 부르기 위해 PDA에 kSOAP을 장착하여 사용한다. [그림 5]는 실험에 사용된 RFID와 PDA장비이다. 현재 구현된 NAMA-RFID시스템은 다음과 같이 설치되었다.

- NAMA 에이전트: ubiq1.khu.ac.kr
- 타겟 웹서비스: ubiq2.khu.ac.kr
- UDDI: ubiq3.khu.ac.kr

Step1. RFID 인식

사용자가 특정 지역에 가서 해당 지역의 RFID 태그를 읽게 되면, PDA의 사용자에이전트(UA) 중 RFID Reader Controller가 작동하여 RFID 태그의 CARD S/N을 가져오게 된다. RFID 정보에 의하여 위치와 대상에 대한 상황인지를 하게 된다. 이다. 상황은 이전에 기술 된 것처럼 바 대로 TILE의 네 가지 정보를 적고 있고 다음과 같이 정된다.



[그림 5] 실험에 사용된 장비

Step2. 서비스 URI 인식

RFID의 Card S/N을 받은 PDA에 내장되어 있는 UA는 다음으로 현재의 장소를 찾기 위해 사용자 정보 저장소를 관리하는 E-Wallet 웹서비스와 통신하여 현재의 지역의 Service URI와 현재의 위치를 받아오게 된다. 이때 통신은 SOAP 프로토콜을 이용하게 되며 E-Wallet 웹서비스는 요청하는 상대방에 대한 인증 절차를 거쳐서 응답 여부를 결정하게 된다. 또한 사생활 침해의 우려를 고려하기 위해 E-Wallet 웹서비스는 공개 가능한 개인 정보를 관리하는 것을 기본으로 하되, 사용자의 선호도에 따라서 정보의 공개 정도를 결정할 수 있

으며, 이에 대한 작업은 사용자가 직접 PDA와 같은 사용자 개인 단말기에 내장된 UA를 통해서 할 수 있도록 허용했다.

Step3. NAMA 에이전트 접속 및 사용자 정보 전달

E-Wallet 웹서비스로부터 NAMA 에이전트의 Service URI를 받아오면 PDA의 UA는 해당 NAMA 에이전트와 통신하여 자신의 정보와 자신의 정보를 가지고 있는 E-Wallet Web Service에 접근할 수 있는 권한을 준다. 이 때 전달되는 값은 사용자의 ID와 비밀번호이며, 공개 가능 자료에 한해서 PDA내에 있는 개인 정보들 역시 필요한 경우 전송하게 된다. 마지막으로 PDA안에 있는 사용자의 스케줄 프로그램에 기록되어 있는 구매 희망목록에 관한 정보들이 전송되게 된다.

Step4. 욕구 인지

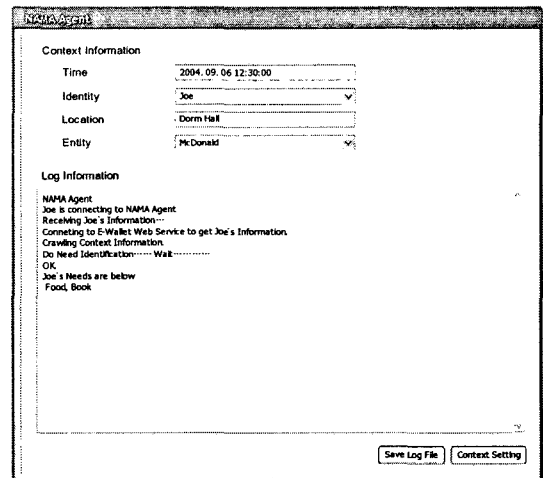
NAMA 에이전트가 사용자의 욕구를 인지하는 과정은 다음과 같다

- 사용자 정보 및 선호도 정보 획득
- 상황인지
- 상황기반 욕구 인지

NAMA 에이전트는 먼저 E-Wallet 웹서비스에 접근하여 사용자의 정보를 가져온다. 모든 정보를 가져온 다음 Bean형태의 사용자 정보 객체를 생성한다. 다음으로 상황 정보를 가져온다. 역시 모든 정보를 가져온 다음 Bean 형태의 상황 객체를 생성한다.

마지막으로 상황을 바탕으로 사용자의 욕구를 추출한다. 사용자가 현재 원하는 정보를 상황과 사용자의 정보를 바탕으로 추출해 낸다. 예를 들면 현재 사용자의 스케줄에 'Shopping Birthday Present'가 있다고 가정하고 현재의 시간이 점심

시간이라면 NAMA 에이전트는 사용자가 현재 음식이나 도서 구매나 음반 구매와 관련된 서비스가 필요할 것이라고 생각할 것이다. [그림 6]은 욕구 인식을 하는 NAMA인터페이스의 예이다.



[그림 6] 욕구 인지

욕구를 인지하는 과정 중 핵심은 사례기반추론을 통한 방법이다. 이 전에 사용자가 현재의 NAMA에서 사용한 사례베이스를 활용한 Fuzzy Linear 검색 이나 Fuzzy Logarithm 을 사용하여 가장 유사한 사례를 발견하여 사용자에게 추천하게 된다. 이때 조건 부분은 정적인 정보와 상황 정보이며, 결과 부분은 사용한 서비스이다.

Step5. 욕구 전달

NAMA 에이전트가 욕구 인지 작업을 끝낸 후 추출된 욕구를 SMM에게 전송하게 된다. SMM도 역시 웹서비스이므로 SOAP프로토콜을 활용한다.

Step6. 서비스 탐색

NAMA 에이전트로부터 욕구 정보를 받은

SMM은 UDDI의 질의를 통하여 서비스 URI 리스트를 작성하게 된다.

Step7. Service Match Maker(SMM)의 서비스 연결

모든 서비스 리스트를 받은 SMM은 개별적으로 서비스들과 통신하며 상황 기반의 현재의 사용 가능한 서비스를 찾아낸다. 만약 발견된 서비스가 현재의 Entity 상에 없거나 시간이 서비스 가능 시간을 초과하였다는 해당 서비스는 리스트에서 빠지게 된다. 결과적으로 현재 상황에 기반하여 추론된 욕구를 충족시킬 수 있는 태스크 웹서비스를 다시 NAMA 에이전트에게 전송하여 준다.

Step8. 서비스 선택

받아온 서비스URI를 바탕으로 각각의 서비스를 호출하여 해당 서비스의 정보를 가져와 사용자의 선호도와 비교하여 가장 높은 서비스를 사용자에게 리스트로 제공하여 준다.

이때 사용자의 선호도는 각 서비스 별로 서비스 온톨로지가 존재하여 해당 서비스에 대한 비교 항

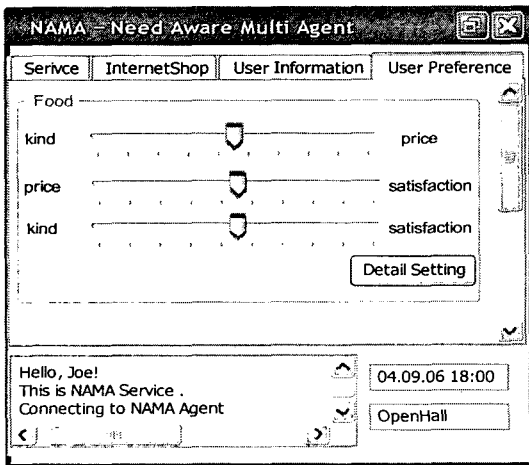
목과 내용들을 불러오게 된다. 사용자 에이전트는 이러한 정보를 온톨로지로 부터 가져온다. 온톨로지는 특정 URI에 저장되어 있으며 PDA의 사용자 에이전트에 의해 정보가 전송된다. 사용자의 선호도를 확보하는 방법은 사용자가 직접 입력하는 방식과 학습 메커니즘을 활용하여 추정하는 두 가지 방법을 모두 사용할 수 있다. 먼저 사용자가 입력을 할 경우 탭을 바꿀 때 정보가 바뀌게 된다. [그림 7]은 사용자가 정보 변경을 선택할 수 있도록 나타나는 화면이다.

사용자가 직접 입력하지 않고 여러 서비스를 사용함을 통해 사용자 에이전트가 스스로 추론을 하는 방법이다. 추론은 다음과 같은 순서로 일어난다.

- 초기값 설정
- 사용자의 서비스 사용
- 계속된 서비스 사용을 통한 기존 값 수정
- 위 두 번째와 세 번째 단계의 반복

먼저 초기값을 설정해 주게 되는데 초기값 설정에 관한 내용은 MyMessage 시스템을 참고하여 설계 하였다[10].

제공된 웹 서비스 중 최적의 서비스를 찾기 위해서도 사례기반추론을 사용한다. 욕구를 정하는 방식과 같은 방식으로 현재의 상황 내에서 사용 가능한 서비스 목록을 이전에 사용자들이 어떠한 서비스를 사용하였는지에 대한 사례베이스를 참고하여 순위를 정하게 된다. 정해진 순위는 사용자의 선호도에 의한 참조가 이뤄져서 최종적인 서비스를 선택할 수 있게 된다. 화면에 동적으로 나타난 서비스를 실제로 사용하기 위해 해당 서비스를 선택하였을 경우 사용자의 정보가 해당 웹 서비스에 넘어가 그 사용자에게 맞는 광고 팝업이 뜨게 한다. 물론 이 팝업은 사용자의 욕구에 대한 정보가 해당 웹 서비스에 넘어 갔을 때 웹 서비스에 동적으로 생성하여 주어 사용자가 현재 원하는 제품



[그림 7] 사용자 선호도 입력

을 추천하는 형태이다. 이때 사용자 정보는 다음 중 하나의 형태로 전달된다.

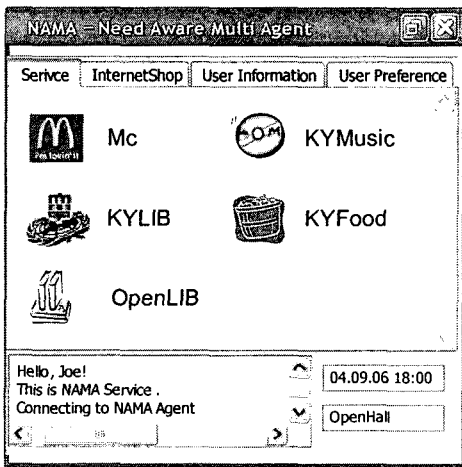
대안 1. 사용자 정보에 대한 메타데이터와 실제 값이 존재하는 URI.

대안 2. 사용자 클라이언트 장비 내의 데이터베이스에 저장되어 있는 사용자 정보의 일부.

대안 3. 사적인 정보는 사용자 클라이언트 장비내의 데이터베이스에 저장되어 있는 사용자 정보의 일부로부터, 그리고 공적인 정보는 해당 정보에 대한 메타데이터와 실제 값이 존재하는 URI.

Step9. 선택된 서비스 보여주기

NAMA 에이전트가 모든 과정을 끝낸 후 최종적으로 남은 서비스URI를 사용자에게 보내준다. 그림 8(a)은 선택 메커니즘이 구동되지 않을 경우의 서비스를 보여주는 예이며, 그림 8(b)의 경우는 NAMA에 의해 선별된 서비스가 사용자의 PDA에 나타난 화면의 예이다.



(a) 특정 상황에서서 가능한 전체 서비스 예



(b) 선별된 서비스 예
[그림 8] 서비스 최종 선택 화면

Step10. 선정된 서비스 활용

사용자가 해당 서비스를 선택하게 되면 사용자 에이전트는 해당 서비스의 URI로 접근하게 되어 해당 서비스를 호출할 수 있게 된다. 해당 웹 서비스는 사용자의 정보를 받아서 사용자에게 맞는 웹 페이지를 보여주거나, 모든 사용자에게 동일한 광고를 제공할 수도 있고, 필요한 경우 결재도 가능하게 한다. [그림 8]은 실제 사용 모습이다. 각 매장에 RFID태그가 부착되어 있다고 가정하며, 사용자는 RFID 리더기가 부착된 PDA를 사용하고 가정하였다. 이때 다음과 같은 시나리오가 가능하다. 아래의 시나리오는 순차적으로 발생하는 것이다.

“사용자는 사전에 주말 점심식사로 값이 저렴한 인스턴트 식품으로 식사를 하겠다고 하는 개인의 선호도를 자신의 PDA로 입력해 두었다. 주말이 되어 사용자는 기숙사를 나와 시내를 거닐고 있었다. 그러나 사용자는 이전에 자신의 식사에 관련하여 입력한 사실을 모르고 있다. 사용자가 무심코

한 인스턴트 음식점을 지나치려고 한다. 이때 사용자의 PDA에 부착된 RFID 리더기는 그 음식점에 부착된 RFID 태그를 통하여 현재 자신의 사용자가 어느 위치에서 어떤 곳을 지나치고 있는지를 인지하게 된다. 이 사용자를 돕고 있는 NAMA 시스템이 곧 작동하여 혹시 사용자가 이전에 입력해 둔 요구사항에 관련이 있는지의 여부를 E-Wallet 서비스의 협조를 받아 점검한다. 물론 그날이 주말이라는 것과 현재 날씨에 대한 상황은 각각 PDA의 기본 기능과 날씨에 대한 웹서비스의 도움을 받아 확인해 둔 상태이다. 결국 NAMA는 사용자가 지금 이 근처에서 인스턴트 식품으로 점심을 하려는 욕구가 있음을 파악하게 된다. 그리고는 신속하게 현재의 위치에서 가까운 반경 이내에 현재 그 사용자의 앞에 있는 음식점 외에 다른 경쟁할 만한 음식점이 있는지를 찾는다. 이를 위해 현 사용자의 위치와 그의 현재의 식사 관련 선호도에 대한 정보를 SMM에게 알리고 그로 하여금 서비스 온톨로지를 보유하고 있는 음식점 웹서비스들에게 협상에 참여토록 한다. 여러 음식점들의 제안을 청취한 NAMA는 SMM을 통하여 선별된 음식점에 대한 정보를 사용자의 PDA에 보여준다. 그런데 그 선별된 음식점은 마침 그 사용자의 바로 앞에 있는 것이었다. 사용자는 자신의 PDA화면에 보이는 결과로 인해 무심코 지나칠 뻔 했던 음식점을 알아차리게 되고 자신이 이날 점심으로 이 음식점에서 판매하는 음식과 같은 것으로 점심을 하려던 사실을 상기하게 된다. 혹 몰라 이 음식점에 대한 더욱 상세한 정보를 알고 싶어서 그 음식점의 로고 모습이 담긴 아이콘을 클릭하니 마침 모바일 쿠폰이 담겨 있었다. 그래서 그 쿠폰을 가지고 음식점에 들어가 만족하게 식사를 하게 되었다.”



[그림 9] 실제 사용 모습

6. 실험

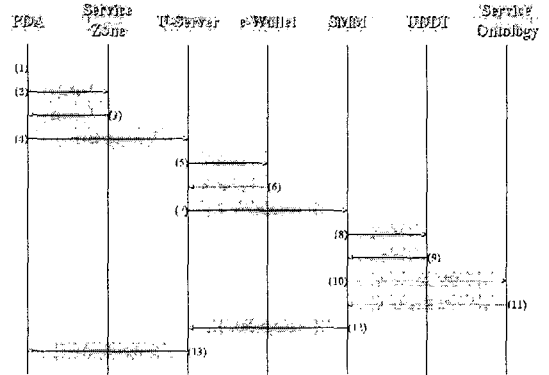
6.1 실험설계

본 논문에서 제안한 시스템이 작동가능한지를 점검하기 위해 시뮬레이션 방법으로 실험을 수행하였다. 본 시뮬레이션을 위해 사용된 시스템의 사양은 다음 <표 2>와 같다.

<표 2> 활용 시스템 사양

서버	Petium4 2.8
	1기가 RAM
	Windows XP SP1
	Tomcat Server: jakarta-tomcat-4.1.30
	Axis 1.2
	Access DB
	java version 1.4.2_04-b05
SMM (Service Match Maker)	Pentium4 1.6Ghz
	256 RAM
	MS Windows Server 2003
	Tomcat Server: jakarta-tomcat-4.1.30
	Axis 1.2
	java version 1.4.x

Service Zone Match Maker	Pentium4 2.0Ghz
	512 RAM
	Windows XP SP1
	Tomcat Server: jakarta-tomcat-4.1.30
	Axis 1.2
	Access DB
	java version 1.4.2_04-b05
E-wallet 웹서비스	Celeron CPU 1100Mhz
	448 RAM
	Windows XP SP1
	Tomcat Server: jakarta-tomcat-4.1.30
	Axis 1.2
	java version 1.4.2_04-b05
통신환경	100M 급 교육전산망



[그림 10] 시스템 수행 순서도

<표 4> 예제 시스템 수행 순서

다음 <표 3>은 예제 시스템에 사용된 구성요소의 용어이다.

<표 3> 예제 시스템에 사용된 구성요소 소개

PDA	개인이 지니고 있는 휴대 단말기를 대표
Service Zone	서비스가 진행되고 있는 센싱 공간
Web Service (U-Server)	각각의 웹서비스로 자료를 요청하고 수집하는 시스템
Web Service (e-Wallet)	개인 정보 자산 관리 시스템
Web Service (SMM)	서비스 중개자 에이전트 기반 웹서비스
Web Service (UDDI)	서비스가 가능한 업체의 목록을 제공하는 서비스
Service Ontology	서비스에 대한 정보 및 자료를 포함하고 있는 웹 사이트

이를 위한 예제 시스템의 수행 순서는 다음 [그림 10] 및 <표 4>와 같다.

플로우 넘버	설 명
(1)	개별 PDA 존재
(2)	PDA가 특정 서비스 지역에 진입
(3)	서비스 지역에서 PDA를 감지함
(4)	PDA가 U-서버에 접속
(5)	U-서버는 e-Wallet에 개인 프로필을 요청
(6)	e-Wallet, U-서버에 개인 프로필을 회신
(7)	U-서버는 SMM에 적합한 웹서비스 명단을 요청
(8)	SMM, U-서버의 요청에 의해 UDDI에 적합한 웹서비스 명단을 요청
(9)	UDDI, SMM에 적합한 웹서비스 명단을 회신
(10)	SMM, U-서버의 요청에 의해 서비스 온톨로지에 관련 정보를 요청
(11)	서비스 온톨로지, SMM에 관련 정보를 회신
(12)	SMM, U-서버에 선택된 웹서비스 명단과 관련 정보를 회신
(13)	U-서버, PDA에 선택된 웹서비스 명단과 관련정보를 회신

시뮬레이션에 수행된 파라미터는 대기행렬 이론에 따라 다음과 같이 결정하였다. 대기이론은 도착자(고객)와 서비스 설비와의 관계를 확률이론을 적용해서 대기행렬의 모델을 작성하고, 고객의 도착 상황에 잘 적응할 수 있는 능력 내지 규모를 결정하려는 것이다. 기본구조를 보면, 어떤 서비스를 규칙이나 순서에 따라 제공하는 서비스 시설과 서비스를 받으러 오는 시스템의 투입물로서의 고객과, 이들에게 제공된 산출물인 서비스 등으로 구분된다.

이 시뮬레이션은 현재 사용하고 있는 서버 시스템을 과연 U-서버로 사용할 수 있을지를 확인해 보고자 하는 것을 목적으로 했다. U-서버는 공간 중심의 서비스이며 개인화된 서비스 이기 때문에 서비스를 받으려고 하는 사용자의 수에 따라 성능이 매우 민감하게 반응할 것으로 예상된다. 그래서 기존의 시스템을 가지고 동시에 몇 명까지 서비스할 수 있는지 알아보려 했다.

첫째, 사용자의 고객 도착 간격(interarrival time)은 10초 당 1명씩 도착하는 것에서 시작해서, 초당 10명씩 도착하는 것까지 범위를 잡았다. 둘째로, 대기행렬의 크기(Queue Size)는 10개~100개로 한정해서 실험했다. 셋째, 웹서비스의 숫자는 20개까지로 한정했다. 왜냐하면 실험적인 차원에서 마이크로소프트사의 UDDI에 등록할 수 있는 최대 숫자가 20이기 때문이다. 이에 대한 측정 단위는 다음 <표 5>와 같다.

<표 5> 시뮬레이션 파라미터

파라미터	측정 단위
고객의 도착시간 간격 (Interarrival Time)	10, 8, 6, 4, 2, 1, 0.8, 0.6, 0.4, 0.2, 0.1 (초)
Queue 길이 (Size of Queue)	10, 20, 30, 40, 50, 60, 70, 80, 90, 100 (명)
웹 서비스의 숫자	10, 20 (개)

6.2 실험결과

이처럼 수행된 시뮬레이션 결과는 <표 6>과 같았다.

그런데 <표 6>에서 나타난 평균서비스 시간은 서비스가 실패한 경우를 제외한 수치이기 때문에 실패한 서비스를 서비스 시간이 가장 많이 소요된 것, 즉 일종의 penalty값으로 고려해야 더 정확한 의미를 찾을 수 있을 것이다. 그래서 시뮬레이션을 수행하면서 실패한 시도에 대해서는 패널티를 주기 위해 10분 동안 서비스를 수행한 것으로 가정하고, 원본 데이터에 10분씩의 가중치를 부여하는 방식으로 원본 데이터를 수정했다. 즉,

$$\begin{aligned} & \text{수정된 평균서비스 시간} \\ & = \text{평균서비스 시간} * (1 - \text{실패율}) \\ & \quad + 10\text{분} * \text{실패율} \end{aligned} \tag{6}$$

와 같이 구했다.

원본 데이터에 있는 서비스 실패율(Error Rate)은 데이터에 반영했으므로 생략하기로 하고 다음 데이터에는 평균서비스 시간만 제시하도록 한다 (<표 7> 참조).

<표 6> 시뮬레이션 수행 결과

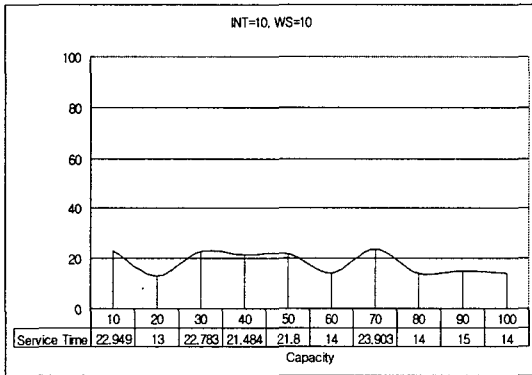
		10	20	30	40	50	60	70	80	90	100
0.1	INT	23	47	72	91	141	152	179	59	61	75
	STD	0.00	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.01	0.01
	ER	1.65	4.69	8.63	22.06	26.09	38.83	34.73	94.59	97.02	99.02
0.2	INT	27	62	98	137	157	220	232	99	109	121
	STD	0.00	0.01	0.03	0.02	0.04	0.04	0.08	0.08	0.06	0.02
	ER	1.89	5.03	17.01	4.84	15.81	21.43	80.14	97.24	86.65	87.83
0.4	INT	28	61	106	135	210	216	241	58	103	85
	STD	0.00	0.01	0.02	0.02	0.06	0.04	0.05	0.07	0.10	0.11
	ER	1.46	8.16	8.05	29.41	36.81	15.38	25.12	98.78	91.37	93.28
0.6	INT	13	58	81	155	181	154	13	14	14	65
	STD	0.00	0.01	0.01	0.01	0.02	0.00	0.00	0.00	0.00	0.06
	ER	2.37	2.94	8.52	5.52	8.67	91.12	22.58	97.38	97.78	98.25
0.8	INT	12	12	12	12	12	12	12	12	12	151
	STD	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10
	ER	3.63	4.32	5.88	6.52	3.39	4.19	14.18	78.60	96.75	90.25
1	INT	19	84	140	148	178	241	88	146	22	75
	STD	0.00	0.01	0.02	0.02	0.02	0.02	0.04	0.02	0.00	0.01
	ER	3.95	1.40	2.26	7.78	2.19	14.97	97.00	85.00	95.55	96.33
2	INT	21	50	180	293	314	358	291	176	235	171
	STD	0.00	0.00	0.02	0.07	0.00	0.08	0.05	0.10	0.06	0.05
	ER	1.72	2.30	9.62	11.11	4.13	30.08	23.16	63.21	49.34	55.00
4	INT	14	14	14	14	13	13	14	14	14	14
	STD	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ER	0.68	0.66	0.00	1.34	0.00	1.35	0.68	1.33	0.66	1.34
6	INT	13	14	15	1:10	15	15	13	14	14	14
	STD	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00
	ER	1.01	2.00	0.00	1.00	0.00	0.00	1.02	1.01	0.00	1.01
8	INT	8	12	12	12	12	12	12	12	12	13
	STD	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ER	1.37	0.00	1.33	0.00	1.35	0.00	1.33	1.33	1.35	0.00
10	INT	13	13	13	12	12	14	14	14	15	14
	STD	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ER	1.69	0.00	1.67	1.61	1.67	0.00	1.69	0.00	0.00	0.00

단, INT = 평균 서비스 시간; STD = 표준편차; ER = 서비스 실패율

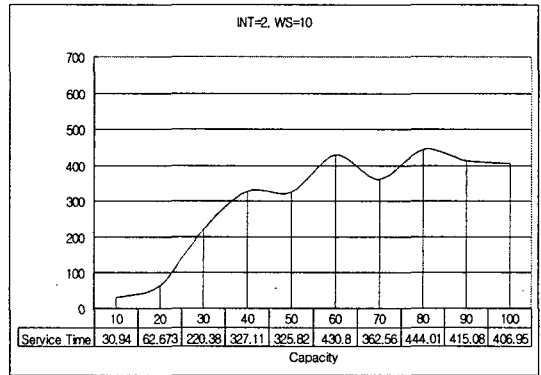
<표 7> 수정 데이터

		웹서비스개수									
		10	20	30	40	50	60	70	80	90	100
수정된 평균 서비스 시간	0.1	32.54	72.92	117.55	203.28	260.74	325.95	325.23	570.71	583.96	594.85
	0.2	37.83	89.06	183.39	210.20	285.68	299.96	526.90	586.17	534.44	541.72
	0.4	36.37	105.00	145.75	271.76	353.56	275.08	331.19	593.37	557.11	565.39
	0.6	26.91	73.94	125.22	179.57	217.33	262.82	145.55	584.63	586.96	590.64
	0.8	33.33	37.43	46.59	50.35	31.93	36.65	95.37	474.18	580.88	556.24
	1	41.95	91.22	150.38	183.19	187.22	294.73	584.64	531.90	574.28	580.75
	2	30.94	62.67	220.38	327.11	325.82	430.80	362.56	444.01	415.08	406.95
	4	17.96	17.86	14.00	21.87	13.00	20.93	17.96	21.81	17.88	21.87
	6	18.93	25.72	15.00	6.05	15.00	15.00	18.99	19.92	14.00	19.92
	8	16.11	12.00	19.84	12.00	19.95	12.00	19.84	19.84	19.95	13.00
10	22.95	13.00	22.78	21.48	21.80	14.00	23.90	14.00	15.00	14.00	

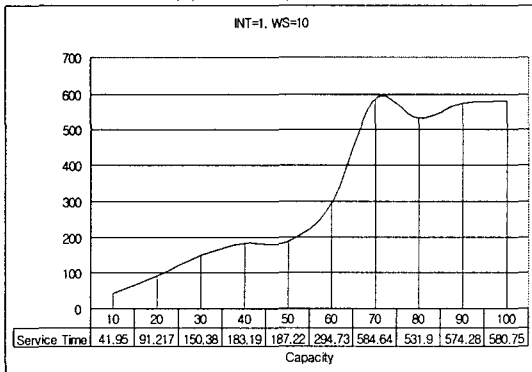
<표 7> 가운데 대표적인 몇 가지만 보이면 다음 [그림 11]과 같다.



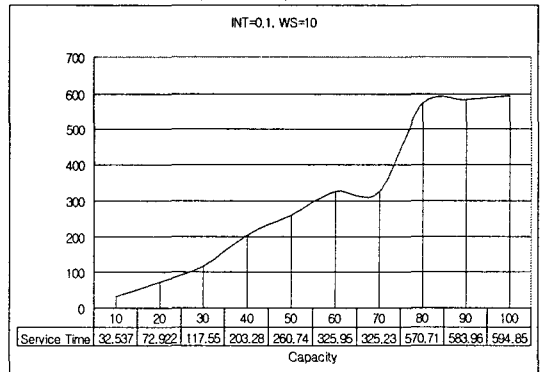
(a) INT=10, WS=10



(b) INT=2, WS=10



(c) INT=1, WS=10



(d) INT=0.1, WS=10

[그림 11] 평균 이벤트 발생 빈도와 웹서비스 수에 의한 성능 측정 결과

결과적으로 RFID값을 읽고 처리하는 과정 상에서 한계 값을 주더라도 속도의 문제는 특별히 노출되지 않았음을 볼 수 있고, 서비스 실패율이나 수정된 서비스 시간에서 나타나는 문제점들은 웹서비스 자체의 문제인 것으로 보인다. 즉, 다음과 같은 두 가지가 주요원인이었다. 첫째는 Tomcat 또는 AXIS, 미들웨어들의 한계이다. 시뮬레이션 자료 중 INT=0.1초 queue size=100, WS= 10을 보게 되면 NAMA에 연결할 때 많은 에러가 난다. 그에 비해서 같은 0.1초지만 queue size=70 WS=10을 보게 되면 에러율이 현저하게 줄어 든다. 이 말은 Tomcat이나 AXIS가 접속자가 많고 적음에 문제 보다, 한꺼번에 동시접속자가 많을 경우 생기는 문제점이 더 크다는 것을 알 수 있다. 0.1초로 수행할 경우 거의 동시에 서비스 요청들이 들어오는데, 이 한계가 70일 경우는 견딜 수 있으나 70이 넘어가서 80이나 그 이상 일 경우 에러가 발생한다. 둘째로는 UDDI의 안정성 문제이다. SMMException을 보면 간간히 UDDI를 접속 중에 에러가 발생하는 것을 알 수 있다. 이는 현재의 서버로 서비스를 제공 할 경우 UDDI에 접속하는 횟수가 빈번해 질 것이고, 이를 UDDI가 좀 더 안정적으로 서비스를 받도록 해야 하는 문제점을 노출하였다. 결론적으로 본 연구에서 개발한 NAMA-RFID 구조는 웹서비스의 근본적인 문제가 해결된다면 실제적으로도 적용 가능하다고 볼 수 있다.

7. 결론

본 논문은 상황인지 및 에이전트 기반 웹서비스가 욕구인지를 통한 개인화된 서비스에 어떻게 기여할 수 있는지를 보여주는 데 중점을 두

었으며, 이를 위해 상기 시스템을 대상으로 시스템 특성과 구현을 하였다. 특히 욕구인지를 위해 Rescorla-Wagner 모형에 의한 연상이론을 적용하였다. 논문 아이디어의 실현가능성을 보이기 위해 NAMA-RFID라고 하는 에이전트 기반의 웹서비스를 제안하였으며, PDA와 같은 최종 사용자의 모바일 장비를 통하여 서비스하도록 구현하였다. 웹서비스간 정보 교환의 자동화를 위해 온톨로지를 활용하였으며, 개인의 정보 자산은 공개 가능한 경우 e-Wallet이라고 하는 웹서비스에서 관리하도록 하였다. NAMA-RFID와 같은 접근방식은 고객관계관리나 기타 표준화된 개인화 서비스에 있어서 온톨로지 활용과 상황인지 시스템을 활용할 수 있다는 시사점을 제공한다고 판단된다.

본 연구의 또 하나의 공헌은 유비쿼터스 컴퓨팅 요소기술로 거론되고 있는 웹서비스, 에이전트, RFID기반 센싱, 그리고 시맨틱 웹 기술을 모두 동원하여 실제로 구현된 시스템으로 성능 분석을 실시하였다는 것이다. 성능 분석에서 살펴본 바와 같이 현존 웹서비스 미들웨어와 UDDI가 웹서비스 및 시맨틱 웹 기반의 유비쿼터스 서비스를 구축함에 있어서 규모성의 문제에 노출되어 있음을 알 수 있었다. 서비스 요구의 간격이 매우 빈번한 서비스의 경우에는 웹서비스 외의 다른 다중 에이전트 협동 메커니즘을 고려해야 할 것이다.

추후 연구방향으로는 온톨로지를 활용한 AHP 모형 자동생성 및 수정 방법 및 사례기반추론 방법을 활용한 욕구 인식과 협상 알고리즘을 응용하여 더 지능화된 유비쿼터스 상기 서비스를 구현하고 검증하는 것이다.

References

- [1] Abowd, G.D. (1999). "Software Engineering Issues for Ubiquitous Computing," *Proceedings of the 21st International Conference on Software Engineering (ICSE '99), Los Angeles, CA*, pp. 75-84.
- [2] Avila-Rosas, A., Moreau, L., Dialani, V., Miles, S., and Liu, X. (2002). "Agents for the Grid: a Comparison with Web Services (part II: Service Discovery)," *Proceedings of the 1st International Workshop on Challenges in Open Agent Systems, Bolona*.
- [3] Dank, D. (2003). "Equilibria of the Rescorla - Wagner model," *Journal of Mathematical Psychology*, Vol. 46, No. 2, pp. 109-121.
- [4] Dickinson, A. and Shanks, D.R. (1985). "Animal Conditioning and Human Causality Judgment," In L.G. Nilsson & T. Archer (Eds.) *Perspectives on Learning and Memory*, Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
- [5] Engel, J. and Blackwell, R. (1982). *Consumer Behavior, 4th ed.* CBS College Publishing.
- [6] Gandon, F. and Sadeh, N. (2004). "Semantic Web Technologies to Reconcile Privacy and Context Awareness," *Web Semantics Journal*, Vol. 1, No. 3, pp. 30-37.
- [7] Han, J. (1994). "Constraint-based query evaluation in deductive databases," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 1, pp. 96 - 107.
- [8] Kralj B., Iverson D., Hotz K., and Ashbury FD (2003). "The Impact of Computerized Clinical Reminders on Physician Prescribing Behavior: Evidence from Community Oncology Practice," *American Journal of Medical Quality*, Vol. 18, No. 5, pp. 197-203.
- [9] Kuno, H. and Sahai, A. (2002). "My Agent Wants to Talk to Your Service: Personalizing Web Services through Agents," *Proceedings of the 1st International Workshop on Challenges in Open Agent Systems, Bolona*.
- [10] Kwon, O.B. and Kim, M.Y. (2004). "MyMessage: Case-Based Reasoning and Multicriteria Decision Making Techniques for Intelligent Context-Aware Message Filtering," *Expert Systems With Applications*, Vol. 27, pp. 467-480.
- [11] Kwon, O.B., Choi, S.C., and Park, G. (2005). "NAMA: A Context-Aware Multi-Agent Based Web Service Approach to Proactive Need Identification for Personalized Reminder Systems," *Expert Systems With Applications*, Vol. 29, No. 1, pp. 17-32.
- [12] Maes, P., Robert, H., Guttman, R., and Moukas, A. (1999). "Agents that Buy and Sell: Transforming Commerce as we Know It," *Communications of the ACM*, Vol. 37, No. 7, pp.31-40.
- [13] Moukas, A., Guttman, R., and Maes, P. (1988). "Agent-mediated Electronic Commerce: An MIT Media Laboratory Perspective," *Proceedings of the International Conference on Electronic Commerce, Seoul, Korea*, pp. 84-86.
- [14] Pivk, A. and Gams, M. (2000). "E-commerce Intelligent Agents," *Proceedings of ICTEC'00*, pp.418-429.
- [15] Rescorla, R.A. and Wagner, A.R. (1972). *A Theory of Pavlovian conditioning: Variations in the Effectiveness of Reinforcement and Nonreinforcement*. A. H., & Prokasy, W. F. (Eds.), *Classical*

- conditioning II: Current Research and Theory*. New York: Appleton-Century-Crofts.
- [16] Robert, H., Guttman, R., Moukas, A., and Maes, P. (1998). "Agents as Mediators in Electronic Commerce," *Electronic Markets*, Vol. 8, No. 1, pp. 22-27.
- [17] Rockinger, R. and Baumeister, H. (2000). "BABSy: Basic Agent Framework Billing System," *Proceedings MAMA 2000, Wollongong, Australia*.
- [18] Sadeh, N., Chan, T., Van, L., Kwon, O.B., and Takizawa, K. (2003). "Creating an Open Agent Environment for Context-aware M-Commerce," in "Agentcities: Challenges in Open Agent Environments," Ed. by Burg, Dale, Finin, Nakashima, Padgham, Sierra, and Willmott, LNAI, Springer Verlag, pp.152-158.
- [19] Shanks, D.R., Holyoak, K.J., and Medin, D.L. (1996). *The Psychology of Learning and Motivation, Vol. 34*. San Diego, CA: Academic Press.
- [20] Terpsidis, I., Moukas, A., Pergiodakis, B., Doukidis, B., and Maes, P. (1997). *The Potential of Electronic Commerce in Re-Engineering Consumer-Retail Relationships through Intelligent Agents.* J.-Y. Roger, B. Stanford-Smith, and P. Kidd. (eds.) *Advances in Information Technologies: The Business Challenge*, IOS Press.
- [21] Trastour, D., Bartolini, C., and Gonzalez-Castillo, J. (2001). "A Semantic Web Approach to Service Description for Matchmaking of Services," *Proceedings of the International Semantic Web Working Symposium (SWWS), Stanford, CA, USA*.
- [22] Wasserman, E.A. (1996). "Attribution of Causality to Common and Distinctive Elements of Compound Stimuli," *Psychological Science*, Vol. 1, No. 3, pp.298-302.
- [23] Williams, A.F., Wells, J.K., and Farmer, C.M. (2002). "Effectiveness of Ford's Belt Reminder System in Increasing Seat Belt Use," *Injury Prevention*, Vol. 8, No. 4, pp. 293-296.

Abstract

Applying Rescorla-Wagner Model to Multi-Agent Web Service and Performance Evaluation for Need Awaiting Reminder Service

Kwon, Oh Byung* · Choi, Keon Ho* · Choi, Sung Chul**

Personalized reminder systems have to identify the user's current needs dynamically and proactively based on the user's current context. However, need identification methodologies and their feasible architectures for personalized reminder systems have been so far rare. Hence, this paper aims to propose a proactive need awaiting mechanism by applying agent, semantic web technologies and RFID-based context subsystem for a personalized reminder system, which is one of the supporting systems for a robust ubiquitous service support environment. RescorlaWagner model is adopted as an underlying need awaiting theory. We have created a prototype system, called NAMA(Need Aware Multi-Agent)-RFID, to demonstrate the feasibility of the methodology and of the mobile settings framework that we propose in this paper. NAMA considers the context, user profile with preferences, and information about currently available services, to discover the user's current needs and then link the user to a set of services, which are implemented as web services. Moreover, to test if the proposed system works in terms of scalability, a simulation was performed and the results are described.

Key words : RFID, Semantic web, Agent technology, Context-awareness, Reminder system, Ubiquitous service

* College of Management and International Relation, Kyunghee Univ.

** College of Management and Economics, Handong Global Univ.

