

## 상호 협조하는

## 퍼지 정보 관리/검색 에이전트

김영일, 조남승, 김병학, 이광형  
 305-701, 대전광역시 유성구 구성동 373-1 한국과학기술원 전산학과  
 (cutty.nscho,bhkim,khlee@fuzzy.kaist.ac.kr)

## Abstract

많은 사람들이 다양한 검색 시스템을 이용하여 웹으로부터 정보를 얻고 있다. 그러나 웹의 크기가 점점 커지고 사용량 또한 증가함에 따라 이러한 시스템들은 몇 가지의 문제점들을 나타내고 있어서 때때로 원하는 수준의 정보를 원하는 시간안에 찾기는 매우 어렵다. 따라서 정보를 효율적으로 관리하는데 도움을 주며, 양적 증가 추세에 있는 웹 환경에 대처하는데 도움을 줄 수 있는 지능적이며 편리한 퍼지 에이전트의 필요성이 대두되고 있다.

본 논문에서는 사용자 정보 관리 에이전트와 이동 에이전트를 이용하여 웹을 검색하는 새로운 모델을 제안한다. 어떤 전문가는 자신의 전문 분야에 관한 양질의 정보를 가지고 있을 것이고, 우리가 그 정보를 공유하면 관련된 양질의 정보를 빠르게 얻을 수 있을 것이다. 이러한 구조를 구현하기 위해 사용자의 정보를 관리하는 에이전트와, 사용자가 직접 이동 에이전트를 보내고 이들이 상호 협조하여 전문가의 지식을 받아오는 방식을 이용하였다. 검색에는 이동 에이전트 기술을 이용하였고, 사용자 정보관리와 정보검색의 방법으로 퍼지 추론을 사용하였다. 또한 전문가에 대한 정보를 알려주는 호스트 네임 서버의 개념을 제안하였다.

## 1. 서론

최근 월드와이드웹(WWW)을 필두로 컴퓨터 네트워크가 발달하면서 사람들은 많은 양의 서비스와 정보들을 이용할 수 있게 되었다. 그러나 네트워크의 성장과 더불어 정보의 양 또한 많아져서 원하는 정보만을 효율적으로 찾기가 점점 더 어려워지고 있다. 현재 사용되는 정보 검색 형태는 서버 중심, 프록시 서버 중심, 그리고 사용자 중심의 시스템으로 분류할 수 있다[17]. 이러한 기존의 시스템들은 사용자가 원하는 정보를 찾는데 있어서 몇 가지 문제점들을 가지고 있으므로, 사용자가 좀 더 효율적으로 원하는 정보를 찾을 수 있

도록 하기 위해 새로운 구조의 시스템들이 연구되고 있다[17]. 본 논문에서는 사용자 정보 관리 에이전트를 사용하여 사용자의 정보를 관리하고, 이동 에이전트(Mobile Agent)를 이용하여 이 사용자 정보를 다른 사람들과 공유(Information Sharing)하는 형태의 검색 방법을 제안하려고 한다.

본 논문에서는 이동 에이전트 기술과 퍼지 이론(Fuzzy Theory)을 이용한 퍼지 관리/검색 시스템을 제안한다. 이동 에이전트는 네트워크를 스스로 돌아다니며 작업을 처리하는 프로그램의 일종이다[3][8][14]. 우리가 실생활에서 어떤 정보에 대해 알고자 할 때, 자신이 혼자 공부하고 찾아서 아는 것보다는 그 정보를 잘 아는 사람에게 직접 물어봄으로써 더 쉽고 빠르게 그 정보를 알 수 있다. 이 아이디어를 정보 관리/검색 방법에 적용하였다. 이동 에이전트를 이용하여 각 사용자의 에이전트는 상호 협조하여 전문가의 사용자 정보를 공유(Information Sharing)하는 구조를 사용한다. 이 시스템은 크게 나누어 사용자 정보 관리 에이전트(Personal Information Management Agent)와, 검색 에이전트(Search Agent) 두가지 에이전트로 구성된다.

관리/검색은 크게 두 단계로 나뉘어지는데, 첫째는 사용자 정보 관리 에이전트(Fuzzy Information Agent)가 각 사용자의 행동을 감시하여 각 사용자 정보를 축적하는 단계이다. 사용자의 행동과 사용자로부터 주어지는 사용자의 선호를 나타내는 키워드를 사용하여 사용자가 관심 있는 분야와 이에 관련된 정보들을 관리한다. 이때 퍼지 이론을 사용하여 사용자의 관심분야와의 관계에 따른 중요도를 계산하게 된다. 또 만일 캐쉬에 저장되어있는 내용 가운데 중요도가 높은 내용이 변경된 경우 캐쉬의 내용을 미리 갱신하여 후에 사용자가 이 정보를 다시 방문하는 때 좀더 빠른 처리를 위해 사용된다. 또한 사용자 정보 관리 에이전트는 사용자의 관심분야와 사용자 호스트의 주소를 호스트 네임 서버(Host Name Server)에 등록하는 일도 한다. 호스트 네임 서버는 '어떤 분야의 정보가 어떤 호스트에 많다'라는 정보를 관리하고 있다. 이 정보를

활용하여 검색 단계에서 검색 에이전트는 상호 협조적으로 동작한다.

둘째는 사용자가 검색하려는 키워드를 가진 검색 에이전트가 이동 에이전트(검색자 에이전트, Searcher Agent)를 생성해서 호스트 네임 서버로 전송하여 주소 정보 검색을 한다. 전송된 이동 에이전트는 호스트 네임서버가 가지고 있는 데이터베이스(Data-base)를 검색하여 그 키워드에 적합한 정보를 가지고 있는 호스트(Host)들의 주소를 결과로써 검색 에이전트에게 전송한다. 검색 에이전트는 결과로 받은 호스트들의 주소로 이동 에이전트들을 다시 전송한다. 전송된 이동 에이전트들은 검색 작업을 수행한다. 이동 에이전트는 그 호스트의 사용자 정보 관리 에이전트의 처리 결과인 사용자의 공유 정보를 여러가지의 기준을 가지고 평가하여 일치도가 높은 것을 결과로써 검색 에이전트에게 전송한다. 이때, 위와 같은 애매한 기준으로 좀 더 정확한 검색 결과를 얻기 위해서 퍼지 이론을 사용한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 기존의 검색 시스템에 대한 간단한 소개와 문제점에 대해 알아본다. 제 3 장에서는 본 논문에서 제안한 새로운 검색 시스템에 대한 모델을 소개한다. 제 4 장에서는 이 모델을 실제로 구현한 방법에 대해 설명하고 제 5 장에서 구현된 결과를 보인다. 마지막으로 제 6 장에서 결론을 맺는다.

## 2. 기존 시스템

여기서는 정보 검색을 위해 구현된 기존의 여러가지 시스템에 관해 알아본다. 기존의 시스템은 그 구조에 따라 크게 세 가지로 나누어 볼 수 있다.

### 2.1. 서버 중심의 시스템

서버 중심 시스템은 일반적인 예로써 검색 엔진을 들 수 있다[16]. 하나의 시스템이 모든 네트워크 상의 정보를 저장하고 있다가 사용자의 질의가 들어오면 그 질의에 맞는 정보를 검색하여 보여주는 방식이다. 이러한 시스템들은 실제로 가장 많이 사용되고 있다. 이러한 서버 중심의 검색 시스템의 문제점은 다음과 같다. 첫째 모든 작업이 서버에게 집중되어 서버의 부담이 너무 크다. 그러므로 네트워크의 크기가 커질 수록 점점 더 고성능의 서버가 필요하게 된다. 둘째, 네트워크의 트래픽을 줄일 방법이 없다. 서버에 연결하는 사용자들에 의한 연결과, 또한 웹 로봇이 네트워크를 통해 정보를 얻어오는데 생기는 연결 등에 의해 야기되는 네트워크 트래픽은 커다란 문제다. 세번째, 너무 많은 정보를 가지고 있으므로 검색 결과에 필요없는 정보가 너무

많아 정보의 질이 떨어지므로, 이중에서 원하는 정보를 찾기란 쉬운 일이 아니다.

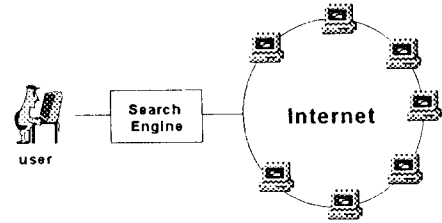


그림 2.1 서버 중심 시스템

### 2.2. 프록시 서버를 이용한 시스템

프록시 서버 중심의 시스템은 프록시 서버가 검색 엔진의 역할을 한다고 볼 수 있다. 도메인 안에서 밖으로, 또는 밖에서 안으로 들어오는 정보들은 모두 프록시 서버를 거쳐가므로, 프록시 서버에서 이러한 정보들을 필터링(Filtering)하여 저장해 둔다. 사용자가 프록시 서버에 질의를 하면, 프록시 서버는 저장된 정보들을 검색하여 결과를 사용자에게 보여주게 된다. 연구소처럼 같은 지역 내에 있는 사용자들은 대개 비슷한 분야의 일을 하고 또한 비슷한 분야에 관심이 있을 것이라는 아이디어에서 나온 시스템으로, OSF WAIBA (Wide Area Information Browsing Assistance)라는 시스템이 구현되어 있다[14].

이 부류의 시스템은 서버중심의 시스템보다는 서버의 부담이 많이 줄어들고 네트워크 트래픽도 감소한다. 그러나, 저장하고 있는 정보가 너무 편중되어 있기 때문에, 즉 사용자들이 이용하는 정보들이 너무 특정 분야에만 깊이 관련된 정보일 확률이 높기 때문에 넓은 분야의 정보를 얻을 수 없다는 단점이 있다.

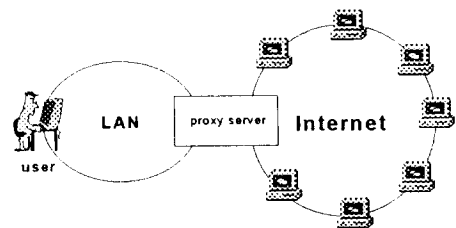


그림 2.2 프록시 서버 중심 시스템

### 2.3. 사용자 중심의 시스템

사용자 중심의 시스템은 사용자가 하나의 에이전트를 두어 자신의 기호(Preference)에 맞게 웹 브라우저를 관리하는 시스템으로써, Net Books[18]과 같은 시스템이 구현되어 있다. 에이전트는 자신의 사용자의 기호를 여러가지 방법으로 분석하여, 관련된 웹 사이트를 찾는 다던가 자주 방문하는 웹 사이트의 URL 을

북마크 파일에 포함시키는 등의 작업을 함으로써 사용자들이 좀 더 편리하고 빠르게 정보를 검색할 수 있게 한다.

사용자 중심의 검색 시스템은 진정한 의미의 검색 시스템으로 볼 수 없다. 이것은 단지 사용자에 속한 하나의 에이전트가 사용자의 기호에 맞게 검색 환경을 구성하여 사용자에게 좀 더 빠르고 편리한 웹 브라우징 환경을 제공하려는 것이 목적이다.

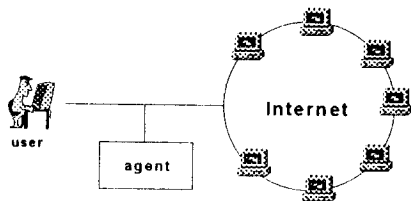


그림 2.3 사용자 중심 시스템

#### 2.4. 비교 분석 및 문제점

이제까지 소개한 여러가지의 정보 관리/검색 시스템들을 비교해 보면 표 2.1 과 같다. 제시된 4 가지의 평가 기준에 대한 설명은 다음과 같다.

- 서버의 부담 - 검색을 위한 서버에 가해지는 부담
- 네트워크 트래픽 - 시스템으로 인해 야기되는 트래픽
- 검색 정보의 효율성 - 검색 결과 중에서 유용한 정보가 차지하는 비율
- 검색 정보의 양 - 검색되어 나온 전체 정보의 양

	서버의 부담	네트워크 트래픽	검색 정보의 효율성	검색 정보의 양
Search Engines	×	×	×	●
OSF WAIBA	▲	●	▲	▲
Net Books	●	●	●	×

표 2.1 기존 시스템의 비교 분석

● : 좋음, ▲ : 보통, × : 나쁨

각각의 시스템들이 장단점을 가지고 있다. 서버중심 시스템은 가장 문제점이 많으나, 관리/검색 시스템의 가장 중요한 기능인 정보의 양이 월등히 많다. 프록시서버 중심의 시스템인 WAIBA 는 검색 엔진의 세 가지 문제점을 많이 보완했으나 원래의 구현 목적이 정보의 그룹화를 위한 것이므로, 정보의 양이 비교적 적고 편중되어서 한정된 분야의 정보에 대한 검색에만 좋은 효과를 나타낸다. 사용자 중심

의 시스템인 Net Books 는 검색 시스템이라기 보다는 사용자의 검색환경관리 시스템이므로 직접적인 비교는 불가능하지만, 사용자가 자주 이용하는 사이트에 대한 정보는 가장 효율적으로 관리할 수 있다. 그러나 정보의 양이 너무 적고, 실질적인 검색 기능은 제공하지 않는다는 치명적인 단점이 있다.

### 3. 제안된 모델

본 논문의 목표는 사용자의 정보를 축적/관리하고 이를 이동 에이전트를 사용하여 상호 협조하는 방식으로 전문가의 지식을 이용하는 검색 방법을 제안하여 기존 검색 방법의 문제점들을 해결하고자 하는 것이다.

#### 3.1. 새로운 시스템의 제안

##### 3.1.1. 전문가 지식을 공유하는 방법

우리가 실생활에서 어떤 정보에 대하여 알고자 할때, 그 정보를 얻는 방법은 두 가지가 있다. 하나는 자신이 직접 책을 찾아본다던가 하는 식으로 혼자 공부하여 알아내는 방법이고, 다른 하나는 그 정보에 대해 잘 아는 사람에게 물어보는 것이다. 일반적으로 두 번째 방법이 훨씬 쉽고 빠르게 원하는 정보를 얻을 수 있다. 이러한 아이디어를 웹 정보 검색에 적용하고자 한다.

만약 어떤 분야에 대한 전문가들이 많이 방문하는 웹 사이트를 알아낼 수 있다면, 내가 원하는 정보에 대한 전문가가 가진 정보를 공유하여 얻을 수 있다면 중요한 웹 사이트를 찾는 데 도움이 될 수 있다. 즉, '내가 알고자 하는 정보에 대한 전문가의 지식을 이용하자' 는 것이 아이디어이다.

예를 들어 어떤 사용자 A가 B라는 분야의 전문가이고 아주 관심이 많다고 해보자. 이젠 컴퓨터를 이용하는 사람이라면 누구나 웹을 필수적으로 이용한다고 생각할 수 있다. 이때 우리는 A가 웹 브라우저를 통해 방문하는 사이트들 중에 B에 관련된 중요한 정보가 아주 많을 것이라고 예상할 수 있다. 그러므로, 그 사용자가 그 동안 방문한 웹 사이트들에 대한 정보를 공유하면, B에 관련된 중요한 정보들을 가지고 있는 곳들을 알아낼 수 있다. 우리는 그러한 정보를 결과로 받아 직접 그 웹 사이트에 접속함으로써 B에 대한 지식을 얻을 수 있다.

그러나 이러한 검색 방법은 내가 원하는 정보에 대한 전문가는 누구인가에 대한 정보가 있어야 한다는 전제조건이 따른다. 그러나 이러한 정보를 인터넷 수준의 커다란 네트워크 상에서 개인이 알아내기란 사실상 불가능하므로, 그러한 정보를 제공해 줄수 있는, 정보 공유를 위한 네임 서버가 필요하게 된다.

그러므로 본 논문에서는 원하는 정보를 키워드로 주어 질의하면 그러한 키워드에 적합한 정보를 가지고 있는 사용자의 호스트 주소(Address)를 결과로 보내줄 수 있는 호스트 네임 서버(Host Name Server)를 제안한다. 호스트 네임 서버는 단지 어떤 호스트의 사용자가 어떠한 정보에 관심이 있는지에 관한 정보, 즉 호스트들의 주소와 키워드만을 저장하고 있으면 된다. 만약 누가 이 호스트 네임 서버에 키워드를 주고 질의를 하면, 단지 전송되어 온 키워드와 데이터베이스 내의 키워드들을 비교하여 적합한 정보를 가지고 있는 호스트의 주소를 결과로 전송하면 된다. 기존의 검색 엔진은 질의에 대한 결과로 웹 사이트에 대한 URL을 보내지만, 호스트 네임 서버는 그 분야에 대한 전문가들의 호스트에 대한 주소를 보낸다. 사용자는 이 호스트들에게 질의를 함으로서 주어진 키워드에 적합한 정보를 많이 가지고 있는 웹 사이트들의 URL을 받아 올 수 있다. 호스트 네임 서버는 네트워크 상에서 관련된 도메인에 따라 계층적 구조를 가질 수도 있다. 이 경우 각 호스트 네임 서버는 자신의 도메인 이외의 질의에 대해 상위/혹은 하위 도메인의 호스트 네임 서버로 질의를 전송한다.

### 3.1.2. 사용자 정보 관리 에이전트 (Fuzzy Information Agent)

사용자의 관심분야는 자신의 전문분야를 포함하여 몇 개의 키워드로 대표된다고 가정한다. 사용자는 사용자의 행동을 감시하는 자신의 사용자 정보 관리 에이전트를 가지고 있다. 사용자가 웹 브라우저와 같은 도구로 정보에 관한 일련의 행동을 하는 동안, 사용자 정보 관리 에이전트는 키워드와 감시된 사용자의 행동으로부터 정보자원(여기서는 웹 페이지)의 위치와 이에 관한 사용자 측면에서의 중요도를 추출한다. 이 중요도는 사용자의 행동과 키워드에 대한 다음의 몇가지 변수를 사용하여 퍼지 추론을 통해 구해진다. 이렇게 구해진 중요도와 정보 자원의 위치는 사용자의 관심분야에 관한 사용자의 지식을 대표하게 되고 후에 검색 에이전트가 보낸 이동 에이전트가 실제 검색을 위해 상호 협조 사용하는 공유 정보가 된다. 사용자 정보 관리 에이전트의 초기화나 이후 키워드가 바뀌게 되는 경우, 키워드와 사용자 호스트의 주소는 호스트 네임 서버에 등록이 되게 되어 다른 상요자의 이동 에이전트에 의해 참고된다. 마지막으로 사용자 정보 관리 에이전트는 갱신된 정보에 대한 빠른 접근을 위해 캐쉬된 자료를 미리 갱신해준다.(Cache Updater)

### 3.1.3. 검색 에이전트 (Search Agent)

앞에서 말한 전문가의 정보를 이용하는 구

조에서, 이동 에이전트가 전송된 호스트는 일반적인 호스트일뿐 검색 서버가 아니므로 검색 기능을 제공할 것이라고 기대할 수 없다. 그러므로 사용자 쪽에서 직접 검색을 수행할 수 있는 이동 에이전트를 전송하여야 한다. 이때 사용자가 이동 에이전트의 코드를 바꾸어 줌으로써 같은 키워드에 대해서도 사용자들마다 자신이 원하는 다른 결과를 얻을 수 있다. 또한, 이동 에이전트를 전송하고 결과를 받을 때만 호스트 사이에 연결(Connection)이 생성되고 그 사이에는 연결이 필요없으므로 네트워크 트래픽이 줄어들게 된다.

### 3.1.4. 새로운 검색 방법의 제안

제안된 검색 방법에 대한 모델은 그림 3.1과 같다.

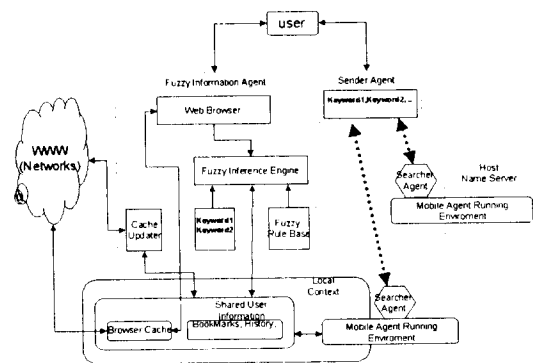


그림 3.1 제안된 모델

검색 방법은 사용자 정보관리 에이전트를 사용하여 사용자의 정보를 축적/관리하는 단계와 호스트 네임서버에 등록된 각 사용자의 정보를 사용하여 실제로 검색을 수행하는 두가지 단계로 이루어진다.

첫번째 단계에서는 사용자의 관심 분야 및 이에 관련된 페이지의 주소등을 중요도에 따라 관리하여 두번째 단계에서 이동 에이전트들이 상호 협조하여 검색에 활용할 수 있도록 공유된다.

두번째 단계에서는 호스트 네임 서버에 이동 에이전트를 전송하여 데이터베이스를 검색하게 함으로써 원하는 정보에 대한 전문가의 호스트 주소(대상 호스트 주소)를 얻어낸다. 그후 검색자 에이전트는 얻어진 대상 호스트들로 다시 이동 에이전트들을 전송시킨다. 전송된 이동 에이전트는 대상 호스트 사용자의 공유 정보를 검색한다.

사용자 정보관리 에이전트와 이동 에이전트가 정보를 평가하는 기준은 다음과 같은 것들이 있다.

- 어떤 웹 페이지를 몇 번 방문했는가
- 그 페이지의 크기는 얼마나 되는가
- 그 페이지가 주어진 키워드와 일치하는

정도는 얼마나 되는가

- 그 페이지가 북마크에 포함되어 있는가  
 이러한 여러가지 기준들을 가지고 검색 작업을 수행한다. 이때 이러한 애매한 기준들을 가지고 키워드와의 일치도를 평가하기 위해서 퍼지 이론을 적용한다. 수행을 끝낸 각각의 이동 에이전트는 가장 키워드와 관련이 깊은 10개의 웹 사이트의 URL을 수행 결과로써 전송한다.

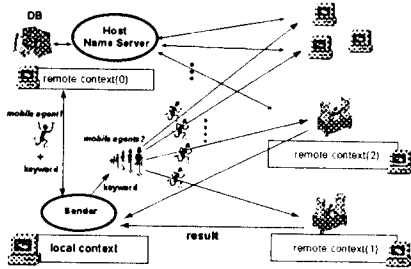


그림 3.2 제안된 검색 시스템의 실행 구조

### 3.2. 제안된 모델의 실행 구조

제안된 새로운 검색 시스템의 실행 구조는 그림 3.2 와 같다. 검색 시스템은 기본적으로 클라이언트/서버 모델을 기반으로 한다. 각각의 부분에 대한 자세한 설명은 다음과 같다.

#### 3.2.1. 사용자 정보 관리 에이전트

사용자 정보 관리 에이전트는 크게 세 가지 부분으로 이루어 진다.

- 웹 브라우저

구현을 위해서 웹 브라우저를 구현하였다. 이 브라우저의 캐쉬와 북 마크, history 는 사용자 정보 관리 에이전트와 캐쉬 갱신자에 의해 사용되어 사용자의 관심분야에 관한 공유 정보를 관리하는데 사용된다.

- 사용자 정보 관리 에이전트

사용자가 지정한 관심분야를 대표하는 키워드와 퍼지 규칙을 사용하여 퍼지 추론의 결과를 통해 북 마크와 캐쉬, history 의 각 중요도를 계산한다. 이 결과는 캐쉬 갱신자가 후에 사용한다. 사용자의 관심분야를 대표하는 키워드와 사용자 호스트의 주소를 호스트 네임 서버에 등록한다. 사용자 정보 관리 에이전트가 처리한 사용자의 관심분야에 관한 공유정보는 검색자 에이전트가 전송되어 검색자 에이전트가 찾고자 하는 키워드를 검색하는데 사용된다.

- 캐쉬 갱신자(Cache Updater)

사용자 정보 관리 에이전트가 처리한 중요도에 따라 웹 브라우저가 사용하는 캐쉬의 내용을 사용자가 시스템을 사용하지 않는 동안에 미리 검색하여 저장하여 사용자가 재접근하는 때에 빠르게 검색을 할 수 있도록 한다.

#### 3.2.2. 검색 에이전트(Search Agent)

검색 에이전트는 다음의 세 부분으로 구성 된다.

- 실행 환경 (Context)

Context 는 에이전트가 동작할수 있는 환경을 말한다. Context 는 실제적으로 에이전트 서버(Agent Server)로 구현되어 있다. 이러한 에이전트 서버는 지금 현재 에이전트가 구현된 언어에 따라 각각 다른 형태의 구조로 구현되어 있기 때문에, 서로 다른 에이전트 서버 사이에서는 에이전트의 이동이 불가능한 문제가 있다. 실행 환경은 사용자 정보 관리 에이전트와 공유되는 공유 정보 부분을 포함한다. 검색자 에이전트는 이 공유 정보 부분을 참고하여 원하는 자료를 검색한다.

- 전송자 에이전트 (Sender Agent)

전송자 에이전트는 local context 에 존재하는 에이전트로서, 사용자 인터페이스를 통해서 사용자로부터 키워드와 URL 을 입력받아 이동 에이전트를 전송하는 기능 이외에도 자신이 생성한 이동 에이전트들과 통신을 하고 또한 결과를 전달받아 다시 사용자 인터페이스를 통해 사용자에게 보여준다.

- 검색자 에이전트 (Searcher Agent)

검색자 에이전트는 이동 에이전트로서, 이 시스템에서는 두 가지의 검색자 에이전트를 사용한다. 하나는 호스트 네임 서버가 있는 호스트로 이동하여 데이터베이스를 검색한 후 결과를 보내주는 검색자 에이전트 1 이고, 다른 하나는 검색자 에이전트 1 이 보낸 결과에서 나온 호스트의 주소로 이동하여 실제 검색 작업을 수행하는 검색자 에이전트 2 이다. 이들은 전송자 에이전트에 의해 지정된 호스트로 전송되어 그 곳에서 검색 작업을 수행한다. 검색자 에이전트 2 는 검색 알고리즘으로 퍼지 이론을 적용하며, 퍼지 추론(Fuzzy Inference)의 결과로 나온 URL 을 메시지와 함께 전송자 에이전트에게 전송한다. 검색을 완료한 검색자 에이전트는 다시 전송자 에이전트가 있는 Place 로 되돌아가서 소멸된다. 좀 더 빠른 검색 결과를 얻기 위해, 여러개의 호스트로 검색자 에이전트 2 를 보낼때, 검색자 에이전트가 방문할 서버들의 주소를 가지고 차례로 방문하는 것이 아니라, 하나의 검색자 에이전트를 여러개 복제하여 방문할 서버들에게 하나씩 전송한다.

#### 3.2.3. 호스트 네임 서버

- 호스트 네임 서버 (Host Name Server)

사용자들은 호스트 네임 서버에 자신의 호스트를 등록시킴으로써 이동 에이전트가 자신의 호스트를 방문할 수 있게 할 수 있다. 호

스트 네임 서버는 사용자들로부터 등록을 받아 그 데이터들을 데이터베이스에 저장하는 기능만을 한다. 호스트 네임 서버에 질의를 하는 전송자 에이전트들은 이동 에이전트를 이 곳으로 보내어 데이터베이스를 직접 검색하게 함으로써 원하는 정보를 얻어낸다. 호스트 네임 서버와 직접 통신하지 않는 이유는 호스트 네임 서버의 부담을 최소화하여 단지 정보 수집만을 하게 하기 위해서이다.

#### 4. 설계 및 구현

##### 4.1. 설계

본 논문의 구현에 쓰인 기반 시스템으로는 Pentium PC 위에 Windows 95 운영체제를 채택했다. 사용자 정보 관리 에이전트를 포함하는 웹브라우저는 Internet Explorer의 ActiveX를 사용하여 개발하였고, 검색 에이전트의 개발틀로는 IBM사에서 개발된 Java Aglet 환경인 Aglets Workbench[19]를 사용하였다. 여기서 Aglet이란 Java로 구현된 이동 에이전트의 통칭이다. Aglets Workbench에서는 Java-AAPI (Aglet Application Programming Interface)을 제공하여 Aglet과 동작환경(Aglet Context) 사이의 상호작용을 가능하게 하고 수많은 라이브러리(Library)를 사용할 수 있게 함으로써 이동 에이전트를 이용하는 네트워크 기반의 어플리케이션 개발을 위한 개발 환경을 제공한다. 아직 최종 버전이 아니고 지금도 계속 새롭게 바뀌고 있다.

구현에서 이동 에이전트 시스템 부분은 Java언어를 사용하여 구현하였고, 호스트 네임서버는 C언어를 사용하여 구현하였다. 호스트 네임서버는 웹 서버에 CGI (Common Gateway Interface) 프로그램을 이용하여 구현하였으며, 이 등록 CGI를 통해 호스트 네임서버의 데이터베이스가 생성된다. 그림 4.1은 호스트 네임 서버가 관리하는 데이터베이스의 일례이다.

143.248.172.125 Mobile Agent Search Sports
143.248.182.102 Internet Fuzzy Genetic Algorithm
143.248.185.109 Search Engine Languages
143.248.172.120 Agent Platform Corba
143.248.187.100 Emotional Engineering Fuzzy
.....

그림 4.1 호스트 네임서버의 데이터베이스

##### 4.2. 적용된 퍼지 추론 방법

전체 시스템에는 두 부분에서 퍼지 추론이 상용된다. 두 부분에서 같은 퍼지 추론이 같은 방법으로 사용되었다.

먼저 사용자 정보관리 에이전트가 사용자가 지정한 키워드와 사용자의 행동에 기반해 공유정보와 캐쉬의 갱신을 위해 각 페이지의 중요도를 계산하는 부분이다. 그리고, 두번째

로 검색자 에이전트가 다른 호스트로 이동하여 검색 작업을 수행할 때 퍼지 추론 방법을 이용한다.

사용자 정보 관리 에이전트는 자신의 캐쉬와 북마크 그리고 history에 있는 웹 페이지와 사용자의 관심분야를 나타내는 키워드와의 일치도를 계산하여 캐쉬와 북마크 그리고 history에 있는 웹 페이지의 중요도를 계산한다. 그리고, 검색자 에이전트는 전송되어 대상 호스트의 사용자의 공유 정보(캐쉬, 북마크 그리고 history)를 읽는다. 이 파일에는 캐쉬 디렉토리 내에 있는 모든 파일들에 대한 전체적인 정보가 들어 있으므로, 이 정보들과 주어진 키워드와의 일치도를 계산하여 중요한 웹 페이지들을 찾아내게 된다. 여기서 이러한 일치도를 계산하는데 퍼지 추론을 사용한다.

이 두가지의 차이점은 사용된 키워드가 사용자가 지정한 것인가 다른 사용자가 보내온 검색자 에이전트가 가지고 있는 것인가 하는 것이다. 전자의 경우는 사용자가 자신의 관심분야를 대표하는 것으로 주어진 키워드이고 이를 바탕으로 각 페이지의 중요도를 계산하여 후에 캐쉬 갱신자(Cache Updater)가 미리 페이지를 갱신하려는 목적으로 사용한다. 후자의 경우는 검색자 에이전트가 가지는 질의자의 질의를 대표하는 키워드다. 이는 호스트 네임서버를 통해 현 응답자의 호스트에서 사용자의 공유정보와 일치도를 계산하여 검색자 에이전트가 대상 호스트에서 관련 정보를 검색하는데 사용된다.

##### 4.2.1. 입/출력 변수(Input/Output Variables)

퍼지 추론에 사용되는 입력 변수들은 다음의 네 가지로 구성된다.

- VC : Visiting Count
- CL : Content Variable
- KM : Keyword Match
- BM : BookMark

VC는 사용자가 그 웹 페이지를 방문한 횟수를 나타낸다. CL은 웹 페이지의 문서의 크기를 나타낸다. 어떤 웹 페이지의 크기가 크면 클 수록 가지고 있는 정보 또한 많을 것이라고 생각할 수 있기 때문이다. KM은 문서의 어느 부분에서 키워드와 일치하는 단어가 나타나는가에 대한 변수이다. BM은 어떤 웹 페이지의 URL이 웹 브라우저의 북마크 파일에 존재하는가에 대한 변수이다.

##### 4.2.2. 퍼지화 (Fuzzification)

위의 네 가지의 입력 변수들은 모두 0에서 1사이의 값으로 표준화(Normalization)된다. 각각의 변수에 대한 표준화 방법은 다음과 같다.

• VC(Visiting Count)

사용자가 가장 많이 방문한 웹 페이지의 방문 횟수를 MVC 라 하면, VC 를 MVC 로 나누어 표준화 한다.

$$VC = \frac{VC}{MVC}$$

• CL(Content Length)

일반적으로 웹 페이지의 크기는 어느정도 절대적인 숫자로 구분할 수 있다. 일반적으로 매우 큰 웹 페이지의 크기는 20000 byte 정도 이므로, 다음과 같이 구분하여 표준화한다.

- CL >= 20000 : VL
- 10000 <= CL < 20000 : L
- 4000 <= CL < 15000 : M
- 1000 <= CL < 4000 : S
- CL < 1000 : VS

• KM(Keyword Matching)

문서의 어느 부분에서 키워드와 일치하는 단어가 나타나는가로 구분할 수 있다. 다음의 Tag 안에 일치하는 단어가 있으면 지정된 값을 갖는다.

- <title></title> : VL
- <H1~6></H1~6> : L
- <EM></EM>, <B></B> : M
- text : S
- not exist : VS

• BM(BookMark)

이 변수는 단순히 그 웹 페이지가 북마크 파일에 존재하는가 아닌가에 달려있으므로, 0 또는 1 의 값을 가지게 된다.

- Exist : VL
- Not Exist : VS

이렇게 표준화된 입력 변수들은 다음의 5 가지의 언어항(Linguistic Term)에 의해 표현된다. 그림 4.2 는 퍼지 언어항을 나타낸다.

4.2.3. 퍼지 규칙 기반 (Fuzzy Rule Base)

4 개의 입력 변수에 대한 퍼지 규칙은 이론적으로  $5^4 = 625$  가지가 나올 수 있다. 그러나, 우리는 휴리스틱(Heuristic)적인 방법으로 규칙의 갯수를 줄일 수 있다. 여기서는 34 가지의 퍼지 규칙을 정의한다. 정의된 퍼지 규칙의 일부를 예로서 나타내면 그림 4.3 와 같다.

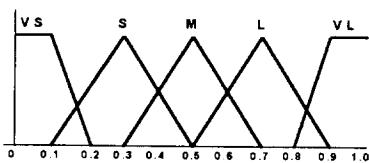


그림 4.2 퍼지 언어항

```

if KM=L and CL=VL then V=VL
if BM=VS and KM=VL then V=VL
if KM=S and VC=L and CL=L then V=L
if BM=VL and KM=M then V=VL
if KM=S and VC=S and CL=VS then V=VS
.....
    
```

그림 4.3 정의된 퍼지 규칙의 일부

4.2.4. 비퍼지화(Defuzzification)

퍼지 규칙을 적용한 후에 나온 퍼지값을 하나의 일반값으로 변환한다. 여기서는 무게 중심법(Center of Gravity)을 사용한다. 이 결과로 나온 값들의 순서로 정렬(Sorting)하여 중요한 웹 페이지들을 가려낸다.

5. 구현 결과

그림 5.1 은 구현된 웹 브라우저의 실행 화면이다.

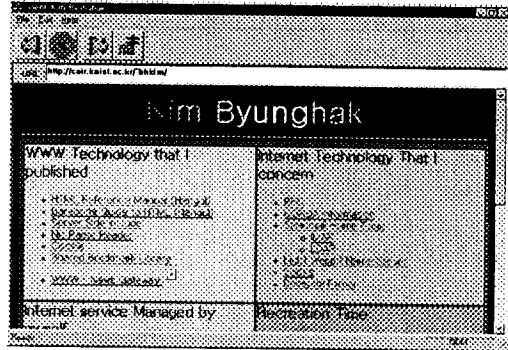


그림 5.1 웹 브라우저의 실행화면

그림 5.2 은 사용자가 웹 브라우저를 통해 호스트 네임 서버에 등록하는 화면이다. 실제로 구현된 시스템을 이용하여 검색을 수행하는 화면은 그림 5.3 에 나타나 있다. 그림 5.3 에서 왼쪽의 윈도우가 사용자 인터페이스 윈도우 화면이고, 오른쪽의 윈도우는 Aglet Server 로서 Aglet 의 실행 환경을 제공한다.

사용자는 A 칸에 가고자 하는 호스트 네임 서버의 주소를 입력하고, B 칸에는 검색하고자 하는 키워드를 입력한다. C 칸은 검색 후에 결과들을 보여주는 화면이고, D 칸은 이동 에이전트의 이동 상황을 사용자에게 보여주는 부분이다. 그림 5.4 는 C 칸에 들어가는 검색 결과의 예이다.

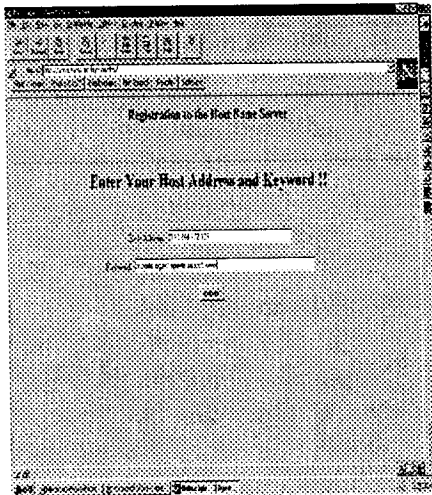


그림 5.2 호스트네임서버에의 등록화면

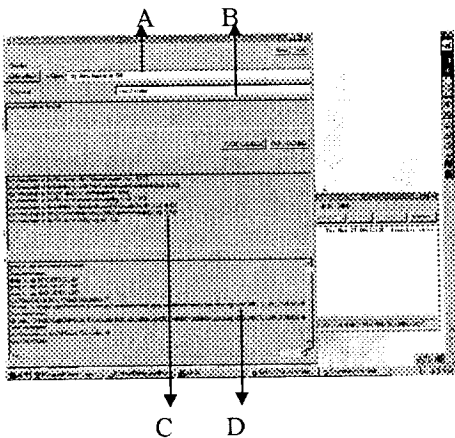


그림 5.3 검색 수행 화면

```
atp://emu.kaist.ac.kr::www.internic.net/tools/ 0.925
atp://emu.kaist.ac.kr::www.altavista.com/ 0.7
atp://emu.kaist.ac.kr::galaxy.einet.net/ 0.7
atp://emu.kaist.ac.kr::www.internic.net/ 0.3
atp://griffin.kaist.ac.kr::www.calcol.it/search/search_htm
0.925
atp://griffin.kaist.ac.kr::www.yahoo.com/computers_and_i
nternet/internet 0.925
```

그림 5.4 C 칸에 나타나는 검색 결과

```
going to: atp://143.248.172.125/
Searcher returned
atp://emu.kaist.ac.kr:434:Search Completed !!!
going to: atp://143.248.172.123/
going to: atp://143.248.172.127/
atp://griffin.kaist.ac.kr:434:Search Completed !!!
Searcher returned
atp://rogue.kaist.ac.kr:434:Search Completed !!!
Searcher returned
```

그림 5.5 D 칸, 에이전트의 이동상황 위의 예에서 순서대로 호스트 이름, 결과의 URL, 그리고 퍼지 주론에 의한 키워드와의 일치도를 나타낸다. 하나의 호스트에서의 결과는 일치도 순으로 나타내어진다. 그림 5.5

는 D 칸에 나타나는 내용의 예이다.

이동 에이전트가 호스트 네임서버 (atp://143.248.172.125/) 로 전송되어 작업을 처리하고 돌아온 후, 결과로 받은 두 곳의 호스트(atp://143.248.172.123/ , atp://143.248.172.127/) 로 이동 에이전트를 전송하고 돌아오는 과정이 차례로 나타나고 있다.

### 5.1. 검색 결과

실제로 키워드를 주고 검색 작업을 수행하여 보았다. 5 대의 Pentium PC 에 서버 환경을 설치하고 실험하였다. 호스트 네임 서버와 데이터베이스는 한 대의 PC 에 위치한다.

실제로 키워드를 주고 이동 에이전트를 전송했을 경우의 결과는 다음과 같다.

**Keyword : 'Search Engine' 의 실행 결과**

```
atp://emu.kaist.ac.kr::www.yahoo.com/ 1.0
atp://emu.kaist.ac.kr::www.internic.net/tools/web-search.html
0.925
atp://emu.kaist.ac.kr::www.altavista.digital.com/ 0.925
atp://emu.kaist.ac.kr::altavista.software.digital.com/search/index.htm
0.925
atp://emu.kaist.ac.kr::altavista.software.digital.com/search/product/
techview/ 0.925
atp://emu.kaist.ac.kr::www.tri.ibm.co.jp/aglets/ 0.925
atp://emu.kaist.ac.kr::www.internic.net/tools/ 0.86
atp://emu.kaist.ac.kr::www.altavista.com/ 0.7
atp://emu.kaist.ac.kr::galaxy.einet.net/ 0.7
atp://emu.kaist.ac.kr::www.internic.net/ 0.3
atp://griffin.kaist.ac.kr::www.calcol.it/search/search_htm 1.0
atp://griffin.kaist.ac.kr::www.yahoo.com/computers_and_internet/i
nternet 0.925
atp://griffin.kaist.ac.kr::www.altavista.yellowpages.com.au/
0.925
atp://griffin.kaist.ac.kr::www.esc20.tenet.edu/techserv/curr/search.
html 0.925
atp://griffin.kaist.ac.kr::calafia.com/webmasters/ 0.925
atp://griffin.kaist.ac.kr::www.monask.com/spidap.html 0.925
atp://griffin.kaist.ac.kr::www.digital-
cafe.com/~webmaster/set01.html 0.925
atp://griffin.kaist.ac.kr::neal.ctstateu.edu.2001/htdocs/websearch.htm
l 0.664
atp://griffin.kaist.ac.kr::www.yahoo.com/business_and_economy/c
ompanies/ 0.5
atp://griffin.kaist.ac.kr::www.ability.org.uk/ 0.3
atp://rogue.kaist.ac.kr::www.yahoo.com/ 1.0
atp://rogue.kaist.ac.kr::www.altavista.com/ 1.0
atp://rogue.kaist.ac.kr::monami.kaist.ac.kr/ 0.388785
atp://rogue.kaist.ac.kr::ai.kaist.ac.kr/overview.html 0.3
atp://rogue.kaist.ac.kr::www.cs.dartmouth.edu/~agent/ 0.3
```

검색 결과는 이동 에이전트가 전송된 호스트별로, 또한 검색 결과로 나온 일치도 순으로 정렬하여 나타낸다. 키워드를 'Search Engine'으로 준 경우 세 개의 호스트로 이동 에이전트를 보내어 검색을 수행하였다. 이 검색에 사용된 호스트 네임 서버의 데이터베이스는 그림 5.6 와 같다.



143.248.172.123	agent network search
143.248.172.124	fuzzy hardware
143.248.172.125	mobile agent search sports
143.248.172.127	internet agent search engine
143.248.172.129	game utility
143.248.187.106	user interface
143.248.187.105	electronic market encryption

그림 5.6 사용된 호스트 네임서버의 데이터

### 5.2. 결과 분석

앞의 5.1 절에서는 제안된 시스템을 이용하여 실제로 검색 작업을 수행하였다. 두 가지의 키워드를 주고 두 번의 검색을 수행한 결과를 나타내었다. 호스트 네임 서버에서 호스트의 주소를 찾아 세곳으로 이동 에이전트를 전송하여 결과를 얻어내었으며, 작업은 성공적으로 이루어졌다.

검색 결과를 현재 많이 사용되고 있는 기존의 검색 엔진의 결과와 비교하였다. 결과로 나온 관련된 정보에 대한 웹 페이지의 수는 표 5.1 과 같다.

앞에서의 검색 결과는 많게는 100 만개까지의 관련 웹 페이지가 결과로서 보여진다. 이 중에서 정말 중요한 웹 페이지를 찾는것은 매우 어렵다.

	Altavista	Yahoo	Meta-crawler	제안된 방법
웹 페이지 수	100 만개 이상	1156 개	57 개	25 개

표 5.1 키워드가 Search Engine 인 경우

기존의 시스템 중 Metacrawler 는 메타 검색 엔진으로써 여러 검색 엔진에 질의를 한 후, 가장 관련이 깊은 웹 사이트들을 몇 개씩 받아 전체적으로 종합하여 결과로 보여준다. 이 경우 결과는 57 개로, 비교적 좋은 정보를 얻을 수 있었지만 여러개의 검색 엔진이 검색을 해야하므로 네트워크 트래픽은 다른 검색 엔진보다 더 심하게 된다. 제안된 시스템에서는 전문가들이 많이 방문하는 웹 페이지들을 일차도 순으로 정렬하여 보여주므로 기존의 검색 시스템들 보다 효율적으로 중요한 정보들을 가진 웹 페이지를 찾아내어 접근할 수 있게 한다.

### 5.3. 평가

표 5.2 에서서 여러가지 기준으로 기존의 시스템들과 제안된 방법을 이용하여 구현된 시스템을 비교하였다. 제안된 방법에서는 이동 에이전트를 이용하여 전문가의 지식을 공유하는 방법을 사용하여 서버 중심 시스템을 탈피함으로써 서버의 부담을 제거하였고, 서버에 집중되는 네트워크 트래픽을 감소시켰다. 또

한, 전문가들이 이용하는 웹 페이지들 중에서 검색 결과를 받아오게 됨으로써 쓸모없는 정보들이 많이 줄어들었으므로, 결과 정보의 질이 향상되어 사용자들이 검색 결과에서 중요한 웹 페이지를 찾기가 쉬워졌다.

	서버의 부담	네트워크 트래픽	검색 정보의 효율성	검색 정보의 양
Search Engines	×	×	×	●
OSF WAIBA	▲	●	▲	▲
Net Books	●	●	●	×
제안된 방법	●	▲	●	▲

표 5.2 기존의 시스템과의 비교

또한, 이동 에이전트를 이용함으로써 사용자가 직접 검색 결과를 관리할 수 있게 되었다. 즉, 이동 에이전트 코드에서 퍼지 추론 부분의 파라미터를 바꾸어 줌으로써 자신이 원하는 결과를 받을 수 있게 되었고 결과적으로 자신이 원하는 검색 알고리즘을 수행하는 자신만의 이동 에이전트를 가질수 있게 되었다. 앞에서의 간단한 실험에서 정보의 양은 매우 적게 나왔으나, 제안된 시스템이 상용화되어 기존의 검색 엔진처럼 널리 알려지고, 그 결과로 호스트 네임 서버에 등록된 호스트가 많아질 수록 제안된 시스템을 이용하여 얻을 수 있는 정보의 범위는 점점 더 넓어질 것이다.

## 6. 결론 및 향후 과제

본 논문에서는 기존 검색 시스템의 문제점을 해결하기 위해 정보 공유 환경에서 이동 에이전트를 이용한 퍼지 정보검색 시스템을 설계하고 구현하였다.

제안된 시스템에서는 사용자 에이전트를 사용하여 사용자의 관심분야에 따라 사용자의 정보를 관리한다. 캐쉬 갱신자는 사용자가 시스템을 사용하지 않을 때, 관심분야에 관련된 중요도가 높은 페이지 가운데 갱신된 페이지를 미리 캐쉬로 읽어 놓아 사용자가 다시 그 페이지를 읽을 때 접근을 빠르게 한다. 사용자의 호스트 주소와 관심분야는 사용자 에이전트에서 호스트 네임 서버에 등록된다. 검색 에이전트는 이동 에이전트인 검색자 에이전트를 이용하여 다른 사용자의 에이전트들과 상호 협조하는 방법을 사용하여 검색을 수행한다. 먼저 키워드와 함께 이동 에이전트를 호스트 네임 서버로 전송하여 검색을 수행하게 함으로써, 키워드에 대한 전문가의 호스트 주소를 결과로 받아오게 한다. 다시 이 주소들

로 이동 에이전트들을 전송하여 전문가들이 많이 방문하는 웹 사이트의 URL을 검색 결과로써 받아온다. 이때 검색 알고리즘으로 퍼지 이론을 적용하였다.

제안된 시스템은 기존의 시스템이 가지고 있는 문제점들을 완화시켰다. 서버 중심 구조를 탈피함으로써 서버의 부담을 제거하였고, 서버에 집중되던 네트워크 트래픽이 줄어들었다. 또한 서로 협조하는 에이전트들을 사용한 정보 공유를 통해 전문가들이 많이 방문하는 웹 사이트들을 결과로 얻게 되므로 중요하지 않은 웹 사이트들이 많이 나오지 않아서 양질의 정보를 얻을 수 있게 되었고, 마지막으로 자신만의 검색 알고리즘을 수행해주는 이동 에이전트를 가질 수 있게 되어 같은 키워드에 대해서도 사용자마다 다른 결과를 얻을 수 있게 되었다.

구현 결과, 기존의 검색 엔진에 비해 결과로 얻은 데이터가 키워드와 훨씬 관련이 깊었고, 그 결과로 검색 결과에서 중요한 정보를 찾기가 쉽게 되었다. 향후 과제로 이러한 이동 에이전트 시스템의 표준화에 대한 연구가 필요하며, 호스트 네임 서버의 다양한 구현 방법에 대한 연구가 필요하다.

#### 참 고 문 헌

- [1] C. Harrison, D. Chess, A. Kershenbaum, "Mobile Agents : Are they a good Idea," IBM Research Report, RC 19887, 1995, <http://www.research.ibm.com/xv-d953-mobaq-ps>.
- [2] D. Chess, B. Grosz, C. Harrison, D. Levine and C. Parris, "Itinerant Agents for Mobile Computing," IEEE Personal Communications Magazine, pp.34-59, Vol.2, No.5, 1995.8.
- [3] T. Magedanz, K. Rothermel, S. Krause, "Intelligent Agents: An Emerging Technology for Next Generation Telecommunications?," INFOCOM '96, San Francisco, 1996.3.
- [4] C. Guilfoyle, E. Warner, "Intelligent Agents: the New Revolution in Software," Technical Report, OVUM Limited, 1994.
- [5] B. Hermans, "Intelligent Software Agent on the Internet," Tilburg University, Tilburg, the Netherlands, 1996.
- [6] J. E. White, "Mobile Agents White Paper," General Magic, 1996.
- [7] J.E. White, "Telescript Technology: The Foundation for the Electronic Marketplace," General Magic White Paper, <http://www.genmagic.com/Whitepapers>, 1994.
- [8] G. Lawton, "Agents to roam the Internet," Sunworld Online.
- [9] Danny B. Lange, M. Oshima, "Programming Mobile Agents in Java - With the Java Aglet API," Aglet Cookbook, <http://www.trl.ibm.co.jp/aglets/aglet-book/index.html>.
- [10] 이 광형, 오길록, "퍼지 이론 및 응용 I, II," 홍릉과학출판사, 서울, 1991.
- [11] E. Cox, "The Fuzzy System Handbook," AP Professional, 1994.
- [12] 정성중, 두길수, 노영만, 박치향, 임기욱, 채영도, "이동 에이전트 기술," 1996.
- [13] 김병학, 이광형, 조충호, "지능형 웹 브라우저 에이전트," HCI '96, 1996. 2.
- [14] Charles L. Brooks, "Wide Area Information Browsing Assistance : Final Technical Report," The Open Group Research Institute, <http://web1.osf.org/www/waiba/papers/y2report/y2report.html>, 1996.4.
- [15] 전응휘, "인터넷 정보검색," WWW-KR, 1996.11.
- [16] "Eureka! Internet Search Engine from Mentor Marketing Services," <http://www.best.clnetom/~mentorms/eureka.htm>, 1996.
- [17] 김병학, 이광형, 조충호, "지능형 웹 브라우저 에이전트," HCI '96, 1996. 2.
- [18] "Net Books," <http://www.frontier-software.com/>.
- [19] "IBM Aglet Workbench - Aglet Sample Program," IBM Corporation, <http://www.trl.ibm.co.jp/aglets/samples/index.html>, 1996.