

# 비서 에이전트 시스템의 설계 및 구현

곽재영\*, 류명춘\*\*, 박창현\*

\*영남대학교 컴퓨터공학과

\*\*경운대학교 컴퓨터전자정보공학부

## Design and Implementation of A Secretary Agent System

Jae-Young Kwak\*, Myung-Chun Ryoo\*\*, Chang-Hyun Park\*

\*Dept of Computer Engineering, Yeungnam University

\*\*Division of Computer Engineering, Kyungwoon University

### 요약

최근 인터넷 및 정보 통신 기술이 급속하게 발달함에 따라서 네트워크를 기반으로 한 응용 소프트웨어 및 네트워크 관련 기술들이 많이 연구, 개발되고 있다. 이와 관련하여 과거 인공지능 분야에서 많이 연구되어 왔던 에이전트 기술이 최근의 네트워크 기술을 기반으로 하여 다시 활발한 연구가 진행되고 있다. 이러한 에이전트 기술을 이용하여 본 논문에서는 일반 비서업무 중에서 상사의 스케줄 관리를 대행할 수 있도록 하는 비서 에이전트 시스템 모델을 설계하고 그 구성 기법을 기술한다. 본 논문에서 기술하는 비서 에이전트 시스템은 3-tier 구조하에 각 비서 에이전트의 스케줄 자료 처리 및 이를 이용한 판단 능력을 효과적으로 부여하기 위해서 데이터베이스 시스템과 지식기반 시스템이 유기적으로 결합되어 운용된다.

### 1. 서론

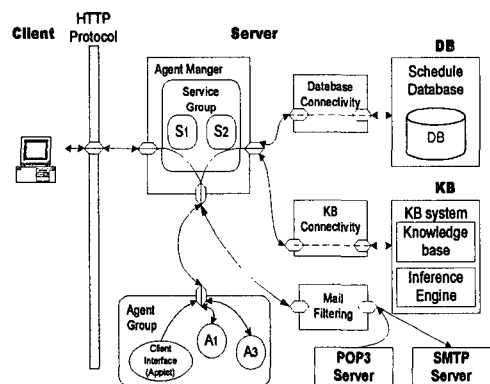
에이전트에 대한 정의는 여러 분야에서 다양하게 언급되고 있으나, 대체적으로 다른 대상을 위해 행동하는 객체로 정의하고 있다<sup>[1]</sup>. 컴퓨터 분야에서는 일반적인 에이전트의 정의로 자율성(autonomy), 반응성(reactivity), 사회성(social ability)를 가진 소프트웨어를 가리킨다<sup>[2]</sup>. 에이전트 시스템의 종류는 크게 지능형 에이전트와 이동형 에이전트로 구분된다. 지능형 에이전트는 특정한 작업을 수행하기 위해서 어떤 지식을 이용하여 자율적으로 동작하는 정적인 시스템을 의미한다. 반면, 이동형 에이전트는 특정 목표를 달성하기 위해서 에이전트가 직접 네트워크를 통해 여러 시스템을 이동하면서 작업을 수행할 수 있는 동적인 시스템을 지칭한다. 최근의 인터넷의 확산과 함께 에이전트 시스템에 대한 연구는 이동형 에이전트에 집중되고 있다<sup>[3]</sup>.

본 논문에서는 지능형 에이전트와 이동형 에이전트의 장점을 이용해 일상적인 비서 업무중에서 상사의 스케줄을 관리하는 비서 에이전트 시스템을 설계하고 이의 구현 기법을 기술한다. 본 논문의 비서 에이전트 시스템에서 서버 및 클라이언트의 각 에이전트는 자바 가상 기계 상에서 수행되며, 클라이언트와 서버간의 에이전트 전송은 HTTP 프로토콜을 이용해서 이루어지게 된다. 또한 사용자의 스케줄 관리는 HEXPERT라는 지식 기반 시스템을 이용해

서 이루어진다.

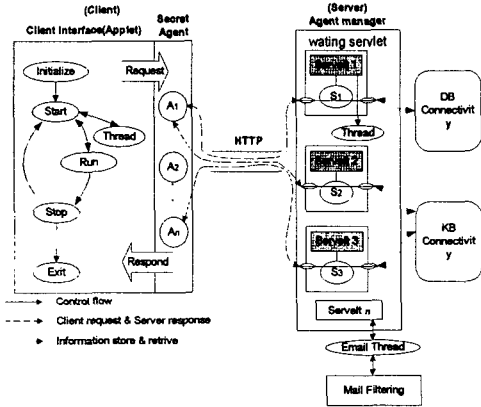
### 2. 비서 에이전트 환경 모델

본 논문의 비서 에이전트 시스템은 3-tier 기반의 클라이언트-서버 구조를 기반으로 네트워크에 연결된 사용자의 요구에 서버가 응답함으로써 각 비서 에이전트들이 동작되며, 에이전트 그룹, 에이전트 관리자, DB&KB Connectivity, 스케줄 데이터베이스, 지식기반 시스템, 메일 필터링으로 구성된다[그림 1].



[그림 1] Design Model of Secretary Agent System

[그림 2]는 클라이언트로 전송된 에이전트와 서버의 에이전트 관리자 사이의 통신 과정을 나타내고 있다. 서버의 에이전트 그룹에 있던 클라이언트 인터페이스와 비서 에이전트는 클라이언트로 전송된 후, 애플릿으로 구현된 클라이언트 인터페이스를 통해 사용자의 요구사항을 입력받고, 비서 에이전트는 그 요구사항을 가지고 서버로 이동한다. 서버의 에이전트 관리자는 서비스 그룹에 서블릿<sup>[4]</sup>을 대기시켜 두고 비서 에이전트가 사용자의 요구사항을 가지고 오면 해당 서블릿의 서비스 루틴을 실행하여 각 에이전트의 요구를 처리하여 결과를 전달해 줌으로써 해당 에이전트에 응답한다. 이 때 각 에이전트는 요구 코드, 전송객체, 그리고 동작 코드를 가지는 클래스로써, 이 클래스는 자바 가상 기계의 객체 전송 방식인 serializable code로 구현된다.



[그림 2] Communication between Agent and Agent Manager Agent Group

i) Agent Group

에이전트 그룹에는 클라이언트 위치에서 각 사용자가 스케줄 관리 또는 예약 작업을 할 수 있도록 하는 비서 에이전트와 애플릿으로 구현된 클라이언트 인터페이스가 존재한다. 클라이언트 인터페이스를 사용자의 요구사항을 얻어내고, 비서 에이전트는 사용자의 요구사항을 서버의 에이전트 관리자에게 전달해 처리를 요구하며, 그 처리 결과를 다시 클라이언트 인터페이스로 가져가 사용자에게 결과를 보여주는 역할을 한다.

ii) Agent Manager

에이전트 관리자는 클라이언트의 에이전트로부터 서비스 요구가 들어오기를 기다리면서 대기하다가 요구가 들어오면 해당 에이전트가 정상적으로 작동할 수 있도록 환경을 구성한다. 에이전트 관리자는 이러한 에이전트의 서비스 요구를 처리하기 위한 서비스 그룹을 가지는데, 이 서비스 그룹은 서비스 처리 메소드를 가진 서블릿들의 집합이다. 각 비서 에이전트의 요구를 처리하기 위해 초기 정보

처리, 사용자 정보 처리, 스케줄 정보 처리, 예약 정보 처리 등의 서비스 요구 처리를 위한 서블릿들이 있다. 에이전트 관리자는 서비스 그룹의 서블릿들을 초기화하여 항상 대기 상태로 있게 하는데, 에이전트로부터의 요구를 처리하고 응답을 해당 에이전트에 전달한다.

iii) DB & KB Connectivity

비서 에이전트 시스템은 데이터베이스의 저장 능력과 지식 기반 시스템의 추론 능력을 결합하여 보다 유용한 정보를 얻을 수 있다. 그러나 스케줄 자료를 이용하는 이 두 시스템의 표현 방식은 완전히 다르다. DB와 KB connectivity는 이처럼 서로 상이한 정보를 상호 변형하여 두 시스템 사이의 표현의 차이를 보완하는 역할을 한다.

DB connectivity는 데이터베이스 테이블의 내용을 에이전트 시스템의 스케줄 객체로 변환시키며, 스케줄 객체로 변환된 데이터베이스의 데이터들은 에이전트의 요구가 있을 때, KB connectivity에 의해 사실베이스 객체로 변환되어 지식기반 시스템에 필요한 사실베이스를 형성할 수 있다.

iv) Schedule Database

스케줄 데이터베이스는 각 에이전트가 스케줄 작업을 진행하면서 생성되는 스케줄 자료를 저장하고 이를 검색할 수 있는 수단을 제공한다. 본 논문의 비서 에이전트 환경을 위한 데이터베이스 시스템은 관계형 데이터베이스로 많이 이용되고 있는 MySQL<sup>[5]</sup>을 사용한다. 스케줄 데이터베이스는 스케줄 테이블, 사용자 테이블, 예약 테이블 등으로 구성된다.

스케줄 테이블은 각 사용자의 시간 관련 정보들로 이루어지고, 사용자 테이블은 사용자의 ID, 이름, 전자우편 주소, 암호 등의 각 사용자 개인 등록 정보를 저장하며, 예약 테이블은 각 교수에게 주어지는 모든 예약들을 임시로 저장하는 테이블로서, 예약자에 대한 정보 및 예약 내용, 예약 시간 등의 정보를 저장한다.

v) Knowledge-based System

지식 기반 시스템은 비서 에이전트 시스템에서 각 에이전트가 스케줄 시간에 대한 판단 정보를 추출할 수 있도록 추론 능력을 부여하기 위한 시스템으로, 스케줄 정보를 이용한 스케줄에 대한 시간적인 정보를 추출하는 것이 목적이다. 본 논문의 에이전트 시스템에서 지식 기반 시스템은 규칙 기반으로 운영되는 HEXPERT<sup>[6]</sup> 시스템을 이용한다. HEXPERT의 지식베이스는 factbase와 rulebase로 구성되는데, factbase는 대상, 상태 등의 성질 또는 속성 등을 객체 지향형 표현으로 나타내는 선언적 기술들의 집합이며, 규칙베이스는 대상, 상태 등에 대한 일반적인 법칙을 조건부-실행부의 형태로 나타내는 규칙들의 집합이다. HEXPERT의 추론엔진은 Rete matching algorithm을 객체 지향형 지식을 처리할 수 있도록 확장한 전방향 추론 기법, 결론 또는 목표를 사실로 가정하고 역방향으로 추론

하여 결론 또는 목표를 증명하는 역방향 추론 기법, 그리고 이 두 기법을 상황에 따라 적절히 혼합하여 사용하는 혼합형 추론 기법을 모두 가지는 통합형 추론 엔진이다.

vi) Mail Filtering

본 시스템의 메일 필터링은 비서 에이전트가 관리하는 일정을 카테고리 별로 메일 제목을 지정한 후 SMTP 서버로 메일을 전송하는 방식으로 이루어진다. 그리고, POP3로 사용자의 메일을 가져온 후 이러한 카테고리별 제목을 이용해 비서업무와 관련된 메일만을 클라이언트 인터페이스 상에서 보여지게 된다. 또한, 개별적인 메일 클라이언트를 통해 메일을 확인할 때에도 메일 클라이언트 자체적으로 가진 필터링 기능을 이용해 비서 에이전트 시스템 관련 메일을 분류하는데 도움을 준다.

3. 비서 에이전트의 추론 과정

이 장에서는 아래와 같은 예가 있을 때 비서 에이전트 시스템의 전체적인 동작과정을 기술한다.

[예 1]

· DB에 저장된 Schedule Data

(1) 1999. 1. 25. 13:30 - 14:30

(2) 1999. 1. 25. 15:30 - 16:00

· 목표

1999년 1월 25일 13:00부터 17:00사이에서 40분 정도 예약 가능한 시간을 찾아라.

사용자가 클라이언트에서 웹브라우저를 통해 서버로 접속하면 서버의 에이전트 그룹에 있던 클라이언트 인터페이스와 비서 에이전트가 클라이언트 쪽으로 이동한다. 사용자는 서버에서 이동해온 클라이언트 인터페이스를 통해 위의 예에 있는 목표를 입력한다. 비서 에이전트는 클라이언트 인터페이스를 통해 사용자가 입력한 요구사항을 받아서 서버로 이동한다. 서버의 에이전트 관리자는 클라이언트에서 비서 에이전트가 이동해 오면 서비스 그룹을 통해 클라이언트의 작업을 처리한다. 이 때 사용자의 요구 사항이 특정 시간대에 대한 판단이기 때문에 먼저 비서 에이전트를 DB connectivity를 통해 Database로 보낸다. 비서 에이전트 DB에서 SQL Query 형태로 관련 데이터를 얻어낸다. 비서 에이전트는 DB connectivity를 통해 SQL 형태의 데이터를 다시 스케줄 객체로 변환한 후 가지게 된다. 에이전트 관리자는 데이터에 대한 추론을 위해 데이터를 획득한 비서 에이전트를 KB connectivity를 통해 KB system으로 보내게 된다. KB system에서 추론한 결과를 획득한 에이전트는 해당 결과를 가지고 원래의 클라이언트로 되돌아가서 그 결과 데이터를 클라이언트 인터페이스에 넘겨준다. 클라이언트 인터페이스는 데이터를 받아서 웹 브라우저 상에 출력한다.

<pre>FACTCASS plan() {   ulD : string   Ts : long   Te : long   :   interval : int   state : string }</pre>	<pre>FACT p1(plan) {   ulD : "park"   Ts : 13:30   Te : 14:30   :   interval : 1:00   state : null }</pre>
<pre>FACT p2 (plan) {   ulD : "park"   Ts : 15:30   Te : 16:00   :   interval : 0:30   state : null }</pre>	<pre>FACT p1(plan) {   ulD : "park"   Ts : 13:30   Te : 17:30   :   interval : 0:40   state : "subgoal" }</pre>

위의 동작과정 중에서, 데이터의 추론을 위해 사용하는 KB system인 HEXPERT의 세부적인 동작 과정은 다음과 같다.

[예 1]의 스케줄 자료를 factbase로 표현하기 위해서는 먼저 fact object의 정의가 주어져야 한다. 스케줄을 위한 객체 정의는 'FACTCLASS'를 이용하여 필요한 애트리뷰트들을 선언하고, 실제의 스케줄 자료는 'FACT'로 생성된다. 위의 예를 사실베이스로 표현하면 다음과 같다.

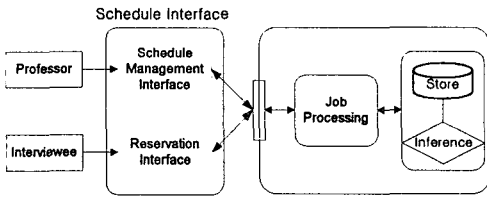
위의 규칙에서 'if' 부분의 세 번째, 네 번째 조건절은 두 사실간의 시간 전후 관계를 비교하는 부분인데, 이 시간 비교 방법에 따라서, 세 가지 다른 형태의 규칙들이 더 있다. 이 규칙들이 실행되고 나면 초기 부목표로 주어졌던 사실이 삭제되고 새로운 부목표들이 생성되게 되는데, 목표 사실을 생성하기 위한 목표 조건을 검사하는 규칙은 다음과 같이 표현될 수 있다.

```
Rule rule-final(RC1)
{
  for: (px instanceof plan)
      (py instanceof plan)
  if: (px.state == "subgoal")
      ((px.Te-px.Ts) > px.interval)
      - (py.state==null)
  do: (px.state="goal")
}
```

이제 이 규칙이 적용되어 생성되는 사실은 목표 조건을 만족하므로 이 사실에 주어진 시간은 예약 가능한 시간이 될 수 있다. 위의 예와 같이 단일한 시간대가 아닌 주기적(일별, 주별 등) 정보의 경우 HEXPERT는 보다 복잡한 룰을 가진다.

#### 4. 사용자별 관점에 따른 비서 시스템의 동작

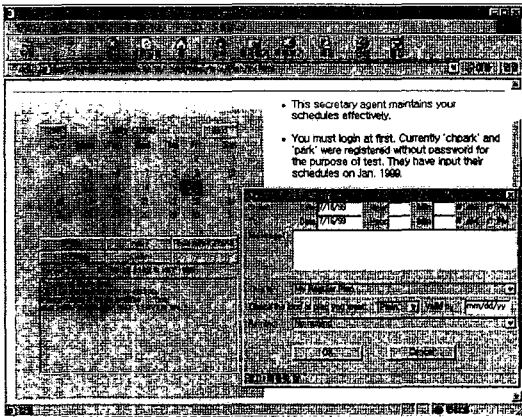
본 논문의 각 비서 에이전트에는 두 부류의 사용자가 있다. 그러나 어떤 사용자의 관점에서 보더라도 자신이 접근하고 있는 비서 에이전트는 저장 능력과 판단 능력을 지니고, 사용자로부터 특정의 작업을 받고 결과를 보여주는 동일 구조로 되어 있다고 볼 수 있다. 이와 같이 사용자의 관점에서 보는 비서 에이전트의 개념적 구조는 다음 [그림 3]과 같이 주어질 수 있다.



[그림 3] Conceptual Structure of Secretary Agent

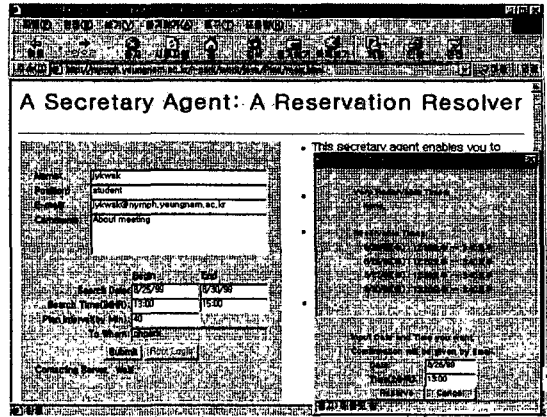
그러나, 비서 에이전트가 작업을 처리하는 과정과 내용은 사용자에 따라 차이가 있다.

교수의 경우, 먼저 스케줄에 대한 입력 처리와 검색 결과 출력 등의 스케줄 입출력 기능, 주기적인 스케줄에 대한 관리 기능, 그리고 e-mail을 통해 예약 상황에 대한 정보를 교수에게 알려주는 기능 등이 있다.[그림 4]



[그림 4] Display of Secretary Agent for Reservation

예약자에 대한 비서 에이전트의 주요 작업은 예약자로부터 예약 조건을 입력받은 후, 그 시간대의 스케줄 자료들을 데이터베이스에서 검색하여 지식 기반 시스템에게 예약 가능한 시간을 추출해 주기를 의뢰한다. 추출된 시간을 예약자에게 알려주고 예약을 접수한다. 예약이 접수되면 비서 에이전트는 이 예약을 교수에게 알려 교수가 확인할 수 있도록 한다.[그림 5]



[그림 5] Display of Secretary Agent for Schedule Management

#### 5. 결론 및 향후 연구 과제

본 논문에서는 일상적인 비서 업무중에서 교수의 스케줄 업무를 대행하는 스케줄링 비서 에이전트 시스템을 설계하고 이의 구현 기법을 기술하였다. 이 때, 사용자의 원시 데이터를 데이터베이스와 지식기반 시스템의 추론 능력을 결합하여 가공함으로써 사용자의 일정에 대한 효율적인 관리가 가능해졌다.

#### 참고문헌

- [1] P. Chan, R. Lee, "The Java Class Libraries", Second Edition, Vol. 2, Addison-Wesley, 1998.
- [2] Stan Franklin and Art Graesser, "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents", The 3rd International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.
- [3] J. E. White, "Telescript Technology: Mobile Agents", White Paper, General Magic, Inc., Sunnyvale, CA., 1996.
- [4] "The Java Tutorial: A Practical Guide for Programmers", Sun Microsystems, Inc., "http://java.sun.com/docs/tutorial/".
- [5] "MySQL", T.c.X, "http://www.mysql.com/".
- [6] Suk I. Yoo, Il K. Kim, Chang H. Park, Hea J. Chang, Tae G. Kim, Mee K. Min, "HEXPART: An Expert System Building Tool", Third IEEE International Conference on Tools for Artificial Intelligence, pp. 510-511, 1991.