

# 웹 에이전트

한국통신 신봉기\*·김영환\*\*

● 목 차 ●	
<ul style="list-style-type: none"> <li>1. 서 론</li> <li>2. 연구 동향</li> <li>3. 웹 에이전트 기능                             <ul style="list-style-type: none"> <li>3.1 대화와 협력</li> <li>3.2 이동</li> <li>3.3 추론과 판단</li> <li>3.4 학습</li> <li>3.5 이동 컴퓨팅</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>4. 에이전트 서비스                             <ul style="list-style-type: none"> <li>4.1 정보 검색</li> <li>4.2 정보 검색 응용</li> <li>4.3 지능형 정보 분석</li> <li>4.4 전자상거래</li> <li>4.5 인터페이스 에이전트</li> </ul> </li> <li>5. 에이전트 플랫폼</li> <li>6. 결 론</li> </ul>

## 1. 서 론

인터넷과 월드와이드 웹의 선풍은 그칠 줄 모르고 그 열기를 더해가고 있다. 이것은 사람들이 컴퓨터 모니터 앞에 앉아 있는 시간이 그만큼 늘어났다는 이야기도 된다. 웹에서 원하는 정보를 얻는 방법은 분산되어 있는 문서 사이의 연결고리인 하이퍼링크를 일일이 쫓아가면서 모든 문서를 읽어 보고 취사선택하는 것이다. 이 작업은 비록 마우스 단추를 누르는 손가락과 문서를 읽을 눈만 있으면 되지만 장시간의 끈기와 고통을 수반하는 육체적 노동이나 다를 바가 없다. 이는 마치 산업 혁명으로 기계를 사용함에 따라 육체적 노동의 강도는 줄었지만 24시간 가동되는 기계를 끊임 없이 보살피 주어야만 하는 것처럼 노동의 모양이 달라졌다는 것을 의미한다.

웹에서는 분산 처리가 매우 단순한 프로토콜 HTTP(HyperText Transfer Protocol)[1]와 RPC(Remote Procedure Call) 동작 원리에 따라 이루어진다. 그 방식은 전통적인 고객-서버 모형에 따른다. 사용자 측의 브라우저에서

시행 명령이 있어야 정보 검색 등 일련의 동작이 촉발되며, 그 작업이 끝나면 다시 새로운 명령을 기다리는 수동적인 체제를 갖고 있다. 요컨대 웹도 이와 같이 고객, 서버 각각의 역할과 둘 사이의 관계는 개발 완료 시점에서 고정된 구조를 갖고 있다.

전통적인 고객-서버 모형에서 고객과 서버 간 대화가 많이 필요한 경우에는 상대적으로 느린 통신이 처리 시간의 대부분을 차지하게 된다. 이때 누가 중간에 개입하여 사용자가 하는 지루한 일-기다림-을 대신해줄 수 있다면 대단히 편리한 대리자, 즉 '에이전트'일 것이다. 이와는 달리 만약 원격 호스트의 서버 쪽으로 고객의 에이전트를 보내서 필요한 대화를 하고 정보를 가져오도록 한다면 때에 따라서 훨씬 큰 효과를 볼 수 있을 것이다. 그 고객은 그 에이전트가 돌아올 때까지 기다리지 않고 단말기의 전원을 끄고 쉬거나 다른 작업에 신경을 돌릴 수 있기 때문이다.

에이전트 모형은 전통적인 고객-서버 모형의 확장형으로서 객체 지향 모형과 마찬가지로 복잡한 시스템 개발에 적합한 새로운 설계 모형의 하나이다. 에이전트 모형은 고객-서버 모형을 일반화한 형태의 분산 처리 모형 또는 분

\*정회원  
\*\*중신회원

산 컴퓨팅 모형이라고 할 수 있다. 분산 객체 모형은 전형적인 분산 처리 모형의 하나라는 점에서 에이전트 모형과 구별된다. 구조적으로 볼때 에이전트는 객체 형태로 실현할 수 있으며, 각 에이전트의 능동성을 제한한다면 기존의 분산처리 형태와 기능면에서 본질적인 차이를 발견하기 어렵다. 그러나 각 에이전트는 고객의 역할뿐만 아니라 서버의 역할도 할 수 있다. 에이전트는 고객의 하나로서 서버에 정보나 서비스를 요구할 수 있고, 다른 에이전트와 접촉해서 거꾸로 정보를 제공해주는 서버의 역할도 할 수 있다. 이러한 에이전트 모형의 장점 중의 하나는 고정적인 웹 브라우저와 달리 필요에 따라 적시에 사용자 요구에 맞는 에이전트를 개발할 수 있고, 따라서 때문에 동적으로 변하는 망 환경에 쉽게 적응하고 망 진화를 능동적으로 유도해 나갈 수 있다는 것이다.

소프트웨어 에이전트의 직접적인 탄생지는 분산 인공지능 분야라고 할 수 있다. 에이전트는 최근 인공 지능, 가상 현실, 인공 생명, 통신 등 여러 분야에서 하나의 새로운 시스템 설계 모델로 자리를 잡아가고 있다. 인터넷, 웹과 같은 분산 환경은 지식과 정보의 거대한 원천이라고 할 수 있다. 하지만 현재의 웹은 본질적으로 고객의 명령에 따라 지정된 정보만 전달해주는 수동적인 성격을 갖고 있다. 이러한 웹 공간에 제각기 다른 목적을 갖고 스스로 동작하는 에이전트를 불러 넣어 보다 능동적인 에이전트 공간으로 만들어가고자 한다.

## 2. 연구 동향

HTTP(HyperText Transfer Protocol)[1]는 웹 공간의 중심핵이라고 할 수 있는 통신 프로토콜로써, 현재 표준화가 진행 중이다. 웹 환경 자체는 확장 여지가 매우 크며, 압축화, 전자 결제 등 현재 개발 중인 각종 기술은 웹의 응용 영역을 늘릴 뿐만 아니라 에이전트가 등장 무대를 마련해주고 있다. 비록 앞질의 기능을 한가지라도 제대로 갖추지는 않더라도 웹 공간에는 웹 브라우저, 검색 엔진의 로봇 등 이미 수많은 넓은 의미의 에이전트가 활동하고 있다. 그리고 현재 웹에서 접할 수 있는 본격

적인 에이전트도 상당히 있는데 앞으로 기능별, 서비스 별로 언급하기로 한다.

최근 웹 에이전트의 붐과 지능형 에이전트의 요구와 더불어 상호 연동 가능한 여러 에이전트 시스템간의 공통된 기술의 사양을 정하고 국제적 협력을 도모하려는 움직임이 나타났다. 그 결과 구성된 것이 세계 유수의 업체와 기관이 모여서 만든 비영리 기구 Foundation for Intelligent Physical Agent(FIPA)이다. FIPA의 주목표는 에이전트 기술의 개발과 관련 기술 사양을 제정하는 것이다. 그리고 단기적으로는 시장성이 있는 지능형 에이전트(Intelligent Physical Agent)를 목표로 하여 요소 기술 사이의 인터페이스와 프로토콜을 정의하고 형식적 표준으로 추진하고자 한다.

IBM이 제안한 Agent Transfer Protocol(ATP)은 분산 에이전트 기반 정보 시스템을 위한 응용 계층 수준의 표준 프로토콜이다. ATP는 인터넷에서 URL(Uniform Resource Locator)[2]로 에이전트 자원의 위치를 표시하여, 에이전트가 하드웨어를 가리지 않고 컴퓨터간에 이동할 수 있도록 해주는 통일된 프로토콜을 말한다. 이동형 에이전트는 서로 다른 기계에서 다른 언어로 쓰일 수 있지만 ATP를 따르면 보편적인 한가지 방법으로 다른 호스트로 이동할 수 있다.

앞으로 에이전트가 쉽게 확산되지 못하게 된다면 유력한 이유 중의 하나는 바로 컴퓨터 시스템의 보안 때문일 것이다. 따라서 에이전트의 인증, 권한 제한 등을 포함한 보안 문제는 매우 중요한 위치를 차지하고 있다[4].

Telescript는 에이전트 언어로서 상당한 수준의 보안 기능을 내재하고 있다고 알려져 있다. 그리고 Java도 보안의 허점이 발생할 소지를 근본적으로 제거하여 신뢰 받고 있지만 앞으로 기술적으로 개선될 가능성 역시 크게 제한되어 있다.

한편 Tcl을 보안에 초점을 맞춰 확장한 Safe-Tcl[5]은 현재 Tcl 7.5에 수용되었다. 반면 또 하나의 Tcl 변종으로서 Telescript와 유사한 Agent-Tcl은 에이전트의 이동 방법에만 치중하여 보안 기능이 극히 제한적이기 때문에 아직 개선의 여지를 남겨 두었다.

### 3. 웹 에이전트 기능

웹에서 에이전트는 웹 공간 안에서 활동하는 능동 개체로 정의한다. 이러한 에이전트를 웹 에이전트 또는 그냥 에이전트라고 부르기로 하자. 에이전트는 사용자의 개성을 반영하는 대리자로서 사용자당 하나 이상, 웹 전체 공간에는 수없이 많은 사람들의 수없이 많은 에이전트가 활동하게 된다. 그리고 웹은 정보가 물리적으로 분산되어 있고 끊임 없이 계속 변하는 동적인 것이 특징이다. 이절에서는 웹과 웹 사용자의 요구를 반영하는 여러 에이전트의 주요 기능은 어떤 것들이 있는지 살펴보기로 하자.

#### 3.1 대화와 협력

에이전트는 다른 에이전트에게 도움을 청하거나-예컨대 도움말을 구하거나 쓸만한 웹사이트 추천을 의뢰한다-또는 몇몇 에이전트와 한데 어울려 어떤 문제를 풀기 위해 협력할 수 있다. 이를 위해서는 다자간에 대화가 필요하며, 대화가 이루어지려면 상호 정보 교환 방식을 맞추어야 한다.

KQML은 정보의 형식을 규정하기 위한 언어인 동시에 에이전트 사이에 정보나 지식을 공유할 수 있도록 해주는 메시지 처리 프로토콜이다. Speech act 이론에 근거한 KQML의 Performative는 에이전트의 기본 동작으로서, 이들이 한데 모여 에이전트간 대화 모델-예를 들어 contract net-을 구축할 수 있는 계층을 이룬다. 그리고 KQML에는 통신 facilitator라는 대화 조정 에이전트가 따로 있어서 지식을 공유할 수 있는 구조를 마련해 준다.

KIF는 서로 다른 언어, 서로 다른 프로그램 사이에 정보의 교환을 위한(사람이 아니라) 컴퓨터 위주의 언어이다. KIF 표현은(Postscript 처럼 인터프리터나 언어 외적인 요소와 무관하며) 그 의미는 KIF 자체 만으로도 파악할 수 있는 선언 및 일차 술어 논리 형식으로 표현된다.

인공 지능은 '존제하는 것은 모두 표현가능하다'는 가정에서 출발한다. Ontology는 그러한 대상 세계를 추상화한 모습을 규정하는 것 또는 개념화 방법을 말한다. 간단히 말해서

Ontology는 에이전트 사이에 질문이나 메시지를 교환하는 형식적 용어를 정의한 것이다.

#### 3.2 이동

ATP는 에이전트가 이동할 수 있는 방법을 제공해주는 프로토콜로써 에이전트 자체의 표현과는 구별된다. 에이전트 공간을 돌아다니면서 임무를 수행하는 에이전트는 여러 기종의 호스트를 만나게 된다. 따라서 에이전트 코드, 즉 절차에 관한 지식은 특정 컴퓨터에서 컴파일된 이진 코드로 표현할 수는 없고, 어디서나 통용될 수 있는 코드로 표현해 주어야 한다. 이러한 이유로 일반적으로 스크립트 언어가 사용된다. 스크립트 언어는 해석기(interpreter)에 의하여 해석되고 '대리' 수행된다. 이와 같이 해석되는 언어로는 Tcl[3, 5]과 Perl[6] 등을 들 수 있다.

한편 Java[6]와 같이 해석이 가능한 중간 단계 표현(Java의 경우 byte code라고 한다) 언어로 컴파일하는 형태의 스크립트 언어도 있다. 그러나 지금의 Java는 능동 문서의 하나로서 제한된 조건에서 에이전트의 일부 기능만 제공하는데 그치고 있다. 따라서 세계적으로 Java를 이동 에이전트의 기능을 첨가하여 확장하려는 시도가 다수 진행중이다. 그중 대표적인 것이 IBM에서 개발된 aglet이다. Java의 applet에 상응하는 개념의 aglet은 인터넷 상의 한 호스트에서 다른 호스트로 이동할 수 있는 Java 객체이다. 나아가 IBM은 aglet와 환경에 간단한 인터페이스를 제공해주는 Java Aglet API를 표준으로 제안해놓고 있다[10].

버클리 대학의 W. Li가 개발한 Java-To-Go는 아직 실험적인 것이지만 이동 에이전트와 에이전트 응용 시스템을 개발 및 시험할 수 있는 환경을 제공한다. 표준 Java와 달리 Java-To-Go 에이전트는 하나 또는 그 이상의 원격 호스트에서 이동 컴퓨팅이 가능하다[11].

자기의 의지를 갖고 있는 자동 에이전트는 언제 이동할 지를 스스로 판단하고, 이동시에는 현 수행 상태를 그대로 보관하여 목적지에서 소생한다. 또한 다른 에이전트를 찾아 대화 및 협상을 하여 서로 도움을 주기도 하고 또 필요한 정보를 얻거나 찾아내기도 한다. 그리

고 현재의 위치를 주인에게 알리고 중간 결과를 보고할 수도 있어야 할 것이다.

### 3.3 추론과 판단

에이전트는 환경을 인식하고 적절히 판단하여 스스로 어떤 행동을 취하는 능동성을 갖는다. 특히 분산 환경의 여러 에이전트는 서로 협력하여 문제를 해결하기도 한다. 이를 위해서 개개의 에이전트는 지식 기반과 추론 능력, 대화 능력 등이 필요하다. 이러한 관점에서 에이전트는 하나의 전문가 시스템으로 구성된다. 에이전트의 지능적 기능은 주로 분산 인공지능 분야에서 널리 연구되어 왔다. SOAR와 CLIPS는 에이전트의 지능 모델로써 자주 이용되는 소프트웨어이다. 80년대 초반에 개발되어 계속 쓰이고 있는 SOAR는 지능형 에이전트를 위한 보편적 인지 모델로 개발되었다. 참고로 '지능'이 쓸모있는 에이전트를 만들어 주는 것이 아니라는 점이다. 단지 현재로서는 지능의 존재 여부가 다른 에이전트와 구별 지어 줄 뿐이다. 그러나 향후 고도의 정보 분석 능력이 필요한 응용의 경우에는 지능이 상당한 역할을 할 것으로 기대된다.

### 3.4 학습

특정 개인의 일을 대신한다는 에이전트의 본질은 에이전트가 그 개인의 특징이나 성향을 익히 알고 있어야 함을 뜻한다. 에이전트에게 사용자의 특징과 성향은 최초 한번에 모든 것을 가르쳐 줄 수도 있을 것이다. 하지만 일회 완전학습의 어려움은 차치하더라도 시시각각 변하는 환경에서 보다 쓸만한 에이전트가 되기 위해서는 사용자의 취향을 스스로 익혀서 적응해 나가는 학습 기능이 필요하다. 이런 기능의 모델의 약점은 충분한 시간이 흐르기 전까지는 아무짝에도 쓸모 없으며, 현재의 성능이 과거의 비슷한 경험에 크게 좌우된다는 것이다. 만약 다른 에이전트와 대화 협력 기능이 있다면 이런 약점을 상당히 보완할 수 있을 것이다.

단독 지능형 에이전트의 기본적인 모습은 일종의 분류(또는 판단) 시스템이다. 단독 지능형 에이전트 시스템의 경우 학습은 전통적 패턴 인식 시스템의 학습 이론을 이용하면 될 것

이다. 그러나 다중 에이전트 환경에서의 학습 모델은 아직도 연구 중이다. 다중 에이전트 환경의 에이전트 학습에 관한 연구 사례는 일일이 열거할 수 없을 정도로 많다[12]. 이점은 문제의 어려움과 흥미를 대변한다고 할 수 있을 것이다. 앞으로 많은 연구가 필요한 문제이다.

### 3.5 이동 컴퓨팅

일반적으로 휴대형 단말기는 컴퓨팅 능력이 부족하며 전원이 오래 지속되지 못한다. 그러나 언제 어디서나 접속이 가능하다는 점을 충분히 활용하면 이동 중이라도 간단한 계산은 단말기에서 하고 중요한 계산이나 큰 계산은 호스트 컴퓨터에서 수행하는 방법을 선택할 수 있다. 이동 에이전트는 여기에도 활용될 수 있다. 예컨대, 이동 에이전트를 강력한 계산 능력이 있는 호스트로 보내어 계산을 하도록 한 다음 접속을 끊고 단말기의 전원을 꺼버린다. 한편 호스트로 이동한 에이전트는 계산을 끝낸 다음 그 결과를 전자 메일로 알려지거나 약속된 위치의 화일에 저장한다.

이와 같은 시나리오의 에이전트 컴퓨팅 모형은 객체 지향 모형과 달리 자동, 능동성이 강조된 것으로서, 일을 행하는 절차(how) 대신 일의 목적(what)을 지정함으로써 어디서 계산할 것인지 결정권은 에이전트에게 맡길 수 있다. 그리고 앞서 말한 것처럼 사용자와 에이전트 사이에는 필요할 때에만 연결하는 단속적인 채널로 충분하다.

## 4. 에이전트 서비스

웹 공간의 에이전트는 기본적으로 웹 사용자의 편의를 도모하기 위해 존재한다. 이 절에서는 웹 에이전트가 해줄 수 있는 서비스를 몇가지 들어보기로 한다.

### 4.1 정보 검색

에이전트가 할 수 있는 일에는 기술적인 한계 외에는 아무런 제약이 있을 수 없다. 하지만 인터넷과 월드와이드 웹 환경에서 에이전트가 해줄 수 있는 것 중에서 가장 근본적이고

중요한 일은 바로 웹 공간에서 필요한 정보를 찾아오는 검색 기능이라 생각된다. 에이전트가 정보를 검색하는 방법은 기존의 검색 엔진을 활용하여 필요한 자료를 찾거나 특정 키워드를 중심으로 직접 찾아나설 수 있다.

첫번째 방법은 하나 또는 여러개의 검색 엔진에서 나온 검색 결과 URL 리스트를 바탕으로 각 웹 문서를 읽어와 불필요하다고 판단되는 것은 버리고 사용자가 원할 만한 정보만을 걸러 찾아 오는 형태이다. 이러한 정보 검색 에이전트는 사람이 웹 브라우저를 이용해서 검색하는 과정을 흉내낸 오프라인 검색이라고 할 수 있다. 그리고 여러 검색 엔진을 하부 구조로 하고 그 결과를 재정리한다는 점에서 메타 검색 엔진이라고도 할 수 있다.

오프라인 검색 에이전트보다 확장된 형태는 어떤 씨앗 URL(예를 들면 북마크 페이지)에서 출발하여 웹을 '탐험'하는 형태를 생각할 수 있다. 이 경우 특정 웹 문서를 중복해서 검색하는 것을 피하기 위해 이미 방문한 사이트 리스트를 기억해두어야 하며, 시간 제약 또는 탐색 깊이에 제약을 두어 함흥차사가 되지 않도록 해야 한다. 일반적으로 위와 같은 선별적 검색 능력이 있는 에이전트가 실현되기 위해서는 에이전트가 사용자의 관심사를 알고 있으며 정보를 분석할 수 있을 정도의 능력이 있어야 한다.

## 4.2 정보 검색 응용

정보 검색 에이전트가 실현되면 여러가지 응용 에이전트를 설계할 수 있다. 예를 들어 단순 검색 기능만을 쓴다면 검색 엔진의 로봇이나 특정 장소에 있는 다량의 정보를 복제하는 미러링(mirroring) 에이전트, 어떤 분야의 뉴스를 대상으로 한다면 개인용 맞춤형 뉴스 에이전트 또는 맞춤형 신문 에이전트를 만들 수 있다.

## 4.3 지능형 정보 분석

인터넷 메일을 분류하고 걸러주는 여과 서비스는 인터넷 메일을 많이 사용하는 사람에게는 절실히 필요하다. 전자 우편 여과 에이전트는 사용자의 기호, 프로필 등을 이용해서 관리해

주는 비서 역할을 해주는 에이전트이다. 여기에서 한걸음 더 나아가 웹 정보를 알아서 취사선택, 분류하고 쓸만한 것을 추천해주는 에이전트를 위한 연구가 진행되고 있다. AT&T 연구소의 아리스트렐레스 과제는 웹 자원을 자동적으로 분류하는 기술을 개발하기 위해 성립되었다. 주목적은 여과 시스템, 텍스트 추출과 분류, 학습 등의 기능이 있는 프로토타입을 개발하는 것이다. 현재의 검색 엔진에서 받을 수 있는 서비스가 앞으로 발전된다면 이와 같은 기능을 필요로 할 것으로 추측된다[13].

## 4.4 전자상거래

전자상거래(Electronic Commerce)는 하나의 가상 시장을 구성하고 거기서 안전한 상업적 거래가 이루어지도록 해주는 것을 가리킨다. 물건 또는 서비스 고객과 판매자가 만나는 가상 시장은 현재 인터넷의 이슈로 떠오르고 있다. 이와 같은 상거래는 사용자(고객과 판매원)가 웹 브라우저를 통해 직접 상거래를 성사시킬 수도 있지만 그 양쪽 당사자가 소프트웨어 에이전트도 가능하다.

사이버 공간의 전자상거래가 이루어지기 위해서는 해결해야 할 과제는 우선 완벽한 망의 보안과 전자 결제 방법 등을 개발하는 것이다. 그 기술 기반이 형성된 다음에야 비로소 가상 쇼핑몰, 인터넷 은행, 인터넷 경제가 가능할 것이다. 워싱턴 대학에서 개발된 ShopBot는 컴퓨터 소프트웨어나 CD를 주문할 수 있는 시험적 웹 에이전트 프로토타입으로서, 상거래 행동을 흉내내는 간단한 에이전트였다. 이와 달리 현재 커다란 연구 과제로 진행중인 CommerceNet은 인터넷 기반의 전자상거래를 위한 기술과 하부 구조물을 개발하기 위해 여러 기관과 업체의 지원하에 형성된 연구 과제이다. 자세한 내용은 [14]를 참고하길 바란다.

## 4.5 인터페이스 에이전트

인터페이스 에이전트가 담당하는 역할은 사용자 인터페이스를 고도화하는 것이라고 할 수 있다. 예컨대 그래픽/가상현실 기술의 하나를 응용한 살아있는 사람 또는 동물처럼 애니메이션과 합성음으로 응답하며, 자연어를 이해하거

나 음성을 이해하는 에이전트를 들 수 있다. 마이크로소프트사의 Persona 에이전트는 자연어 대화와 애니메이션이 가능하며, Argo사의 Genie 에이전트도 마이크로소프트사의 Active-X를 이용하여 개발된 비슷한 대화자이다. 향후에는 인간과 컴퓨터의 인터페이스는 이와 같이 멀티미디어에 의한 지능형 멀티모달 인터페이스로 나아갈 것으로 예상된다.

### 5. 에이전트 플랫폼

사람은 다재다능하지만 텅빈 공간에 홀로 살아가지는 못한다. 자연이라는 공간 속에서 의식을 헤쳐하며 사회라는 공동체 환경 속에서 의미있는 일을 하면서 살아간다. 마찬가지로 자동 소프트웨어 에이전트도 쓸모있는 대리자 역할을 하기 위해서는 활동할 수 있는 환경 또는 시설물이 필요하다. 예를 들어 에이전트가 다른 에이전트와 접촉을 하기 위해서는 그 에이전트를 찾을 수 있는 장소와 방법에 관한 정보가 필요하며, 또 어떤 정보를 얻기 위해서는 그런 정보가 어디에 있는지 알아야 한다. 이동 에이전트의 경우에는 목적지 호스트로부터 출입 허락/인증을 받아 실제로 그 호스트로 이동하고 이동 후에는 다시 소생하여 주어진 임무를 계속 수행하는 등 이동 에이전트가 활동하

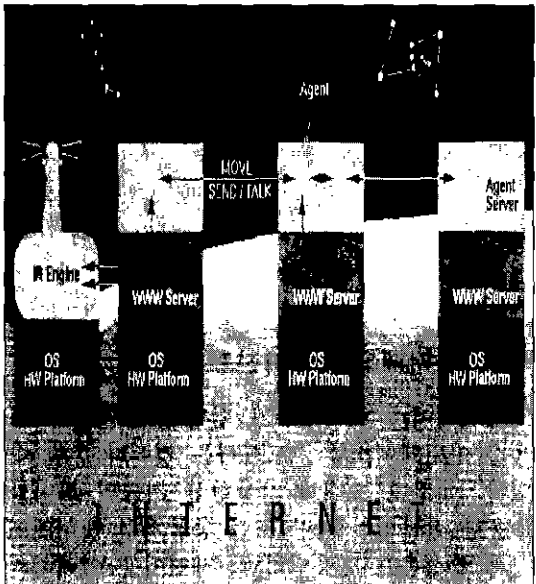
는데 필요한 기본적인 기능과 여건이 필요하다. 이와 같은 정보, 환경 또는 시설을 통틀어 에이전트 하부구조물(agent infrastructure)라고 한다.

에이전트 공간은 웹 공간과 마찬가지로 노드의 집합과 노드간의 연결로 구성된다. 각 노드의 환경을 정의하는 에이전트 플랫폼은 에이전트 서버와 인터프리터의 두가지로 구성된다. 에이전트 서버는 에이전트가 생성되어 작업을 수행할 수 있도록 기본적인 공통 기능을 지원해주는 프로세스이다. 웹 서버의 기능은 사용자 고객과 CGI(Common Gateway Interface) [15]와 URL(Uniform Resource Locator)[2] 지정 문서 사이의 일회용 채널을 제공하는 것에 불과하다. 반면 에이전트 서버는 에이전트가 각 호스트에서 작업을 완료할 때까지 계속 통신 및 대화 환경을 지원하며 정보를 제공하고 위법 행위 유무를 감시하는 등 서버의 기능을 다해주어야 한다.

에이전트 서버에 의하여 탄생된 에이전트는 인터프리터에 의하여 수행된다. 인터프리터는 프로세스 큐에 들어온 에이전트를 적절한 우선순위에 따라 실행하게 된다. 에이전트 공간에 여러가지 언어로 표현된 에이전트가 있는 경우 각개 언어에 대한 인터프리터를 필요로 한다.

‘돈있는 곳’에 사람이 몰리는 것처럼 “정보가 있는 곳에 에이전트가 간다”라고 말할 수 있다. 이동 에이전트는 통신망을 타고 이동하면서 곳곳의 호스트 컴퓨터의 자원과 정보를 사용한다. 그런데 불완전한 에이전트는 오동작함으로써 컴퓨터 자원을 낭용할 수 있으며 해악을 끼칠 수도

있다는 점에서 큰 우려의 대상이 되고있다. 이와 같은 측면에서 볼 때 이동 에이전트는 사실 컴퓨터 바이러스와 다를 바가 없다. 에이전트는 데이터뿐만 아니라 자신의 코드를 스스로 변화시킬 수 있는 능력, 자신을 한없이 복제할 수 있는 능력을 가질 수 있다. 그리고 그 능력이 서버의 감시망을 피해 시스템을 파괴하는데 쓰일 가능성도 얼마든지 있다. 따라서 에이전트로부터 컴퓨터 환경을 보호하는 문제는 에이전트 시스템의 확산을 위해 반드시 해결하고 넘어가야 할 중요한 장벽이다. 뿐만 아니라 환



경으로부터 에이전트 자신을 보호하는 자기 방어 문제도 무시할 수 없다. 그래서 인터넷 선풍이 몰아치기 시작할 즈음 아이히만D. Eichmann은 시의적절하게 웹 에이전트를 위한 윤리 강령[8]을 제안하기도 했다. 그러나 Issac Asimov의 세개의 로봇 법칙[9]과 마찬가지로 이 강령이 필연적인 자연 법칙도 아니고 또 법률과는 달리 구속력 있는 강제 조항도 아니므로, 에이전트의 존재로 인한 해악이 발생할 가능성이 항상 존재한다는 사실을 염두에 두어야 한다.

## 6. 결 론

에이전트는 인터넷, 월드와이드 웹의 각광과 함께 관심을 받기 시작했다고도 할 수 있다. 인터넷이라는 분산 환경과 무궁무진한 웹의 정보가 한데 어울렸을 때 에이전트의 모델의 효용성이 비로소 인식되기 시작한 것 같다. 웹 에이전트가 지금 현실화 되어가고 있지만 그 가능성이 모두 드러나기 위해서는 해결해야 할 과제는 아직 여러가지가 남아 있다. 본 고에서는 그러한 점을 중심으로 에이전트의 기능과 가능한 서비스, 그리고 에이전트를 위한 환경으로서의 에이전트 플랫폼을 기술하였다.

## 참고문헌

- [1] T. Berners-Lee and R. Fielding and H. Frystyk, Hypertext Transfer Protocol-HTTPW/1.0, Internet Draft draft-ietf-http-v10-spec-04.html, HTTP Working Group, Work in progress, 1995.
- [2] T. Berners-Lee and L. Masinter and M. McCahill, Uniform Resource Locator (URL), RFC 1738, Network Working Group, 1994.
- [3] J. K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley, Reading, MA, 1994.
- [4] T. Finin, Agents and security, UMBC AgentWeb(<http://www.cs.umbc.edu/agents/security/>).
- [5] N. S. Borenstein, EMail with a mind of its own: The Safe-Tcl language for Enabled Mail, Distributed as part of the Safe-Tcl 1.2 distribution available over the Internet.
- [6] L. Wall and R. L. Schwartz, *Programming Perl*, O'Reilly & Associates, Sebastopol, CA, 1990.
- [7] J. Gosling and H. McGilton, The Java language environment: A white paper, Technical Report, Sun Microsystems, 1995.
- [8] D. Eichmann, Ethical Web Agents, In *Proc. 2nd International WWW Conference*, Chicago, IL, Dec. 1995.
- [9] I. Asimov, *I, Robot*, Fawcett Crest Publications, Greenwich, Conn, 1970.
- [10] IBM, Programming Mobile Agents in Java(tm) (<http://www.trl.ibm.co.jp/aglets/>).
- [11] W. Li and D. Messerschmitt, Java-To-Go Iterative Computing Using Java (<http://ptolemy.eecs.berkeley.edu/dgm/javatools/java-to-go/>).
- [12] M. V. Nagendra Prasad and Thomas Haynes, Learning in Multi-Agent Systems Webliography(<http://dis.cs.umass.edu/research/agents-learn.html>).
- [13] AT & T Laboratories, Project Aristotle (sm): Automated Categorization of Web Resources(<http://www.public.iastate.edu/~CYBERSTACKS/Aristotle.htm>).
- [14] Yuh-Jong Hu, Intelligent Agent & Electronic Commerce web page(<http://www.cs.nccu.edu.tw/~jong/agent/agent.html>).
- [15] 이택경, 고급 CGI 프로그래밍, 웹코리아 제 3회 WWW 워킹샵, pp. 35-49, 1996.



신 봉 기

1985 서울대학교 공과대학 자원 공학과 졸업  
1987 한국과학기술원 전산학과 졸업(공학 석사)  
1995 한국과학기술원 전산학과 졸업(공학 박사)  
1987~현재 한국통신 근무  
관심분야: 패턴인식 및 모델링, 지능형 에이전트, 인공지능



김 영 환

1981 경북대학교 전자공학과 학사  
1983 한국과학기술원 전산학과 석사  
1983 한국전기통신공사 전임연구원  
1986~1990 한국과학기술원 전산학과 박사과정 파견, 공학박사  
1991~1995 한국통신 소프트웨어 연구소 선임연구

원

1995~현재 한국통신 멀티미디어 연구소 책임연구원, 인터넷 연구실장

관심분야: 인공지능, 정보검색, 멀티미디어 통신서비스 인터넷

● HPC ASIA '97 학술대회 ●

- 일 자 : 1997년 4월 28일~5월 2일
- 장 소 : 서울 힐튼호텔
- 주 최 : 병렬처리시스템연구회
- 문 의 처 : 대회사무국  
T. 02-501-7065