

에이전트의 개요와 연구방향

한양대학교 최중민*

● 목 차 ●

- | | |
|-------------------|---------------|
| 1. 서론 | 4. 에이전트 언어 |
| 2. 에이전트 개요 | 5. 에이전트 응용 |
| 2.1 에이전트 정의 | 5.1 인터넷 정보검색 |
| 2.2 에이전트 특성 | 5.2 온라인 전자상거래 |
| 2.3 에이전트 연구방향 | 5.3 사용자 인터페이스 |
| 3. 에이전트 구조 | 5.4 메시징 |
| 3.1 에이전트 구성요소와 유형 | 6. 결 론 |
| 3.2 멀티 에이전트 시스템 | |

1. 서 론

에이전트를 한마디로 표현하면 사용자를 대신해서 사용자가 원하는 작업을 자동적으로 해결하여 주는 소프트웨어라고 할 수 있다. 지능형 에이전트는 사실 인공지능 분야에서 오래 전부터 연구되어 온 개념으로, 사실상 인공지능의 최종 목표가 사람과 유사하게 생각하고 행동하는 지능을 가진 에이전트의 개발이라고 보아도 크게 벗어나는 표현은 아닐 것이다. 하지만 에이전트라는 분야가 인공지능과 분리되어 독립적인 연구 주제로 대두되기 시작한 것은 80년대말부터이며 그 이후 에이전트에 대한 독자적인 국제 학술대회가 개최되고 관련 제품이 많이 출시되면서 활발하게 연구가 전개되고 있다.

에이전트가 인공지능이라는 모체와 분리되어 독자적인 노선을 걷게 된 배경은 여러 가지가 있을 수 있겠지만 가장 두드러진 것은 분산 협동 처리(distributed cooperative processing)와 에이전트간 통신(inter-agent communication)의 개념이 대두되면서 부터이다[1]. 기존의 인공지능이 추구하는 대상은 하나의 독립된

에이전트로써 주로 인지적인 측면(cognitive aspect)에서 에이전트의 지식 표현과 추론, 자연어 처리 등에 대한 연구개발이 이루어졌고, 이를 바탕으로 전문가 시스템이나 지식 베이스 시스템 등의 응용 시스템이 개발되었다. 하지만 나날이 다양해지고 복잡해지는 사용자의 요구를 해결하기 위해서는 단독 에이전트의 능력으로는 한계가 있었고 그 해결방법으로 지역적으로 떨어진 다른 에이전트의 도움을 받아 처리하는 분산 협동 처리의 개념이 접목되기 시작하였다. 이와 관련된 분야가 분산 인공지능(distributed artificial intelligence)이며 에이전트와 밀접한 관계가 있다.

여러 에이전트의 분산 협동 처리를 위해서는 에이전트간 통신이 필수적이다. 에이전트간 통신의 목적은 정보나 작업처리의 공유와 교환에 있다. 즉, 모든 정보를 포함하고 어떠한 일도 혼자서 처리할 수 있는 완전한 에이전트를 개발하는 것은 불가능하므로 자신이 가지고 있지 않는 정보를 접근하거나 자신이 해결하지 못하는 작업을 처리하기 위해 다른 에이전트에게 도움을 청하는 단계가 반드시 필요하게 되는데, 이 도움을 청하는 방법으로 에이전트간 통신을 이용하는 것이다. 에이전트간 통신을 위

* 정회원

한 가장 큰 걸림돌은 각 에이전트가 가지고 있는 이형질성(heterogeneity)이다[17]. 즉, 각각의 에이전트는 서로 다른 사람에 의해, 서로 다른 시기에, 서로 다른 플랫폼을 바탕으로, 서로 다른 목적으로 개발되었기 때문에 이들간의 통신을 위해서는 상호 이해 가능한 언어와 프로토콜이 필요하다. 이러한 노력의 대표적인 예가 KQML(Knowledge Query and Manipulation Language)[7]이라고 할 수 있다. 하지만 KQML의 표준화 작업이 지연되는 것을 보아도 이형질성의 해결이 얼마나 어려운지 짐작할 수 있다.

90년대 중반에 들어서면서 에이전트에 대한 관심과 필요성이 증대되고 있는데, 그 이유 중에 두드러진 것은 인터넷의 폭발적인 이용 증가로 인하여 좀 더 사용자에게 정보를 쉽게 접할 수 있도록 편리함을 제공하는 기능이 요구되었기 때문이다. 즉, 인터넷의 정보 홍수 속에서 원하는 정보를 정확하게 제시간에 얻기란 쉬운 일이 아니며, 따라서 이러한 작업을 대신해주는 에이전트의 역할이 점점 커지고 있다. 또한 국가 시책으로 추진되고 있는 초고속 정보통신망과 연계하여 정보 검색뿐 아니라 온라인 쇼핑 등의 전자상거래나 메시징과 같은 이동 컴퓨팅 분야에서도 에이전트의 필요성은 대두되고 있다.

본 논문은 이러한 에이전트에 대한 이해를 돕기 위한 전반적인 사항을 요약하여 전개해 나간다. 본 논문은 다음과 같이 구성된다. 1장에서는 에이전트의 역사적 배경과 필요성, 현재 관심정도등을 기술하였다. 2장에서는 에이전트의 정의, 특성, 연구방향등 에이전트 개요에 대하여 기술한다. 3장에서는 에이전트 구성요소와 에이전트 통신 등을 다룬 에이전트 구조에 대하여 기술한다. 4장에서는 에이전트 개발을 위한 에이전트 언어를 기술한다. 5장에서는 에이전트가 적용되는 응용분야를 소개한다. 6장에서는 결론과 앞으로의 방향을 제시한다.

2. 에이전트 개요

2.1 에이전트 정의

이 논문에서 논의되는 에이전트는 기본적으로 소프트웨어 에이전트[10] 로써 큰 범주로 보면 일종의 프로그램이라 볼 수 있다(물론, 로봇과 같은 하드웨어 에이전트도 에이전트의 범주에 포함시키기도 하지만 여기서는 소프트웨어 에이전트에 대해서만 다룬다). 하지만 이때 대두되는 질문은 과연 소프트웨어 에이전트라는 것이 기존의 일반 프로그램과 무엇이 다르냐 하는 것이다. 바꾸어 표현한다면 기존의 소프트웨어(예를 들어 워드 프로세서, 스프레드 시트, 게임 등)들도 모두 에이전트라고 할 수도 있지 않느냐 하는 것들이다. 이를 위해서 단순한 프로그램이 아닌 소프트웨어 에이전트이기 위해서는 어떤 기능을 수행하고 어떤 역할을 담당해야 하는가에 대한 답변이 요구되는데 이에 대한 답변이 바로 에이전트의 정의가 될 수 있다.

에이전트에 대한 정의는 매우 다양하다. 사람마다 에이전트를 보는 시각이 다르고 연구 방향이 다르기 때문이다. [8, 16, 22] 등에서는 논의된 에이전트의 여러 가지 정의를 비교하여 공통된 부분을 다음과 같이 종합하였다.

- 에이전트는 특정 목적에 대하여 사용자 대신하여 작업을 수행하는 자율적 프로세스(autonomous process)이다.
- 에이전트는 독자적으로 존재하지 않고 어떤 환경의 일부이거나 그 안에서 동작하는 시스템이다. 여기서의 환경은 운영체제, 네트워크, 또는 MUD 게임환경 등을 지칭한다.
- 에이전트는 지식베이스와 추론 기능을 가지며 사용자, 자원(resource), 또는 다른 에이전트와의 정보교환과 통신을 통해 문제 해결을 도모한다.
- 에이전트는 스스로 환경의 변화를 인지하고 그에 대응하는 행동을 취하며, 경험을 바탕으로 학습하는 기능을 가진다.
- 에이전트는 수동적으로 주어진 작업만을 수행하는 것이 아니고, 자신의 목적을 가지고 그 목적 달성을 추구하는 능동적 자체를 지닌다.
- 에이전트의 행동의 결과로 환경의 변화를 가져올 수 있다. 에이전트의 행동은 한 번

에 끝나는 것이 아니라 지속적으로 이루어진다.

이러한 에이전트의 정의를 전자우편 에이전트(email agent)를 예로 들어 설명할 수 있다. 전자우편 에이전트의 목적은 사용자가 수작업으로 해왔던 전자우편의 분류, 검색, 삭제, 전달 등의 작업을 대신해 주는 것이며, 특성상 Unix나 Windows 등의 운영체제 환경 하에서 동작한다. 지식의 처리를 위해서 사용자의 요구 사항이 조건-작용 규칙(condition-action rule)의 형태로 지식베이스에 저장되고, 이 규칙의 조건부분은 추론 메카니즘을 통해 전달된 메시지와 매치가 되는지 결정된다. 가령 'security'에 관한 전자우편이 도착했을 때 사용자가 현재 있는 곳으로 전달해 달라는 규칙이 있을 때, 'security alert'라는 제목의 전자우편이 오면 조건이 만족되었으므로 사용자의 위치를 파악하여 전화나 팩스 등의 방법으로 메시지를 전달하게 된다. 전자우편 에이전트는 한 번의 수행으로 종료되는 것이 아니고 데몬(daemon)처럼 항상 동작되면서 새로운 전자우편 메시지를 모니터링하게 되며, 전화번호 데이터베이스나 사용자의 일정 등과 같은 시스템의 다른 자원과도 메시지 교환을 통해 협력하게 된다.

에이전트에 대한 정의는 절대적인 것은 아니지만 어떤 소프트웨어가 에이전트인지 아닌지를 나름대로 판단하는데 좋은 지표가 될 수 있다. 물론 판단대상이 에이전트다 또는 아니냐고 단정짓는 것은 좋은 접근 방향이 아닌 것 같고, 다만 위에서 정의된 항목에 얼마나 많이 부합되는가에 따라 에이전트의 역할을 좀 더 충실히 한다고 말할 수 있겠다.

2.2 에이전트 특성

에이전트의 정의를 살펴보면 기본적으로 에이전트라고 불리기 위해서 필요한 몇 가지 특성을 추출해낼 수 있는데 대표적으로 자율성(autonomy), 지능(intelligence), 이동성(mobility), 사교성(social ability) 등을 들 수 있다.

자율성은 에이전트와 다른 일반 소프트웨어를 구별해주는 가장 핵심이 되는 특성이라고 볼 수 있는데, 사용자나 다른 에이전트의 직접적인 지시나 간섭 없이도 스스로 판단하여 행

동하는 성질을 의미한다. 일반 프로그램이 수동적인 것에 비해 에이전트는 자율성을 가짐으로써 능동적으로 작업 수행을 진행하게 되며, 이를 위해 에이전트는 수행 동작이나 내부 상태변화 등에 대한 제어권도 가지고 있다. 자율성은 사용자로 하여금 상위단계 목적(high-level goal)에 집중을 하게 하고 그 목적을 달성하기 위한 세부 절차 차등은 에이전트가 맡게 된다[3]. 에서 자율성에 대한 정형화를 시도하였다.

지능은 지식 베이스와 추론 능력을 갖추고 사용자의 의도를 파악하여 계획(planning)을 세우고 학습(learning)을 통하여 새로운 지식을 스스로 터득하는 성질로 인공지능에서 많이 연구된 결과를 근거로 한다. 지능은 사실상 자율성과 밀접한 관계를 가지게 되며, 지능을 바탕으로 에이전트는 같은 작업이라도 계획과 경험을 통해 더 나은 효과를 기대할 수 있다.

이동성은 사용자가 요구한 작업을 현재의 호스트에서 수행하지 않고 실제 그 작업을 처리하는 호스트로 이동시켜 수행함으로써 수행의 효율을 높이고 네트워크 부하를 줄이는 효과를 가져온다. 이동성은 기존의 클라이언트/서버의 개념과는 판이하게 다른 개념으로 서버의 내용을 클라이언트를 통해 전송받아 정보를 얻거나 작업수행을 하는 것이 아니고 클라이언트가 필요로 하는 작업을 위해 에이전트를 서버로 보내어 수행을 시킨다. 이동성은 특히 인터넷의 보급으로 컴퓨터 네트워크를 통해 제공되는 정보의 수가 급증하면서 그 중요성이 강조되고 있으며 원격 통신시에도 통신 라인이 항상 접속되어 있을 필요가 없기 때문에 무선 이동통신을 위한 작업수행 환경에서 큰 효과를 낼 수 있다.

사교성은 에이전트간 통신 능력을 의미한다. 즉, 하나의 에이전트로는 처리하지 못하는 작업의 수행을 위해 다른 에이전트의 도움을 필요로 할 때 에이전트간 메시지 교환에 의존하게 된다. 에이전트 통신 언어(agent communication language)를 이용한 에이전트간 통신은 메시지 전달(message passing)이나 공유 메모리(shared memory)방법을 이용할 수도 있고 다른 에이전트의 메소드(method)를

불러 수행하기도 한다.

이러한 기본적인 특성 외에도 에이전트가 가지는 성질은 여러 가지 있는데 그 중에서 환경 변화에 대하여 반응할 수 있는 반응성(reactivity), 틀린 정보를 주고받지 않는 정직성(veracity), 그리고 반드시 목적을 달성하는 방향으로 작업을 수행한다는 이성적 행동(rationality) 등이 있다. 물론 이러한 성질들을 모두 갖추어야 에이전트가 된다는 것은 아니며 단지 이러한 성질을 많이 만족할 수록 좀 더 지능형 에이전트에 근접한다는 얘기가 될 수 있으며 그러한 방향으로 에이전트 연구와 개발이 진행되어야 할 것이다.

2.3 에이전트 연구방향

에이전트의 연구방향은 크게 에이전트 이론(agent theory), 에이전트 구조(agent architecture), 에이전트 언어(agent language), 에이전트 응용(agent application) 등으로 구분된다.

에이전트 이론은 앞절에서 기술한 에이전트의 정의와 특성에 대한 규정(specification)을 명확하게 결정하는 것으로, 에이전트가 무엇이며 어떤 기능을 수행하는가에 대한 정형화(formalism) 작업이 주를 이룬다. 에이전트 이론에서는 에이전트를 믿음(belief), 지식(knowledge), 소망(desire) 등의 인지 요소에 바탕을 둔 의도 시스템(intentional system)으로 간주하고 에이전트의 행동에 따라 인지 요소들이 어떻게 변하는지에 대한 추론 모델을 제시한다. 에이전트 이론에 대한 연구로 [12]는 Z 언어를 이용하여 에이전트의 각 특성을 에이전트 환경과 행동으로 연계시켜 정형화하였고, [19]는 멀티에이전트 시스템을 선언적 표현방법을 사용하여 정의하였다.

에이전트 구조는 이론에서 정형화한 특성들을 구현하기 위한 에이전트 구성요소와 제어 및 통신 프로토콜등을 연구한다. 에이전트의 특성상 분산 멀티 에이전트 환경 하에서의 에이전트간 상호작용에 대한 고려가 많이 이루어지고 있으며 시스템 자원과의 연계성 연구와 사용자와의 인터페이스에 대한 연구도 이 범주에 속한다. 이밖에 에이전트 라이프사이클과 에이전

트 네임서버, 보안, 이동성 등에 대한 연구가 이루어지고 있다. 에이전트 구조에 대해서는 3장에서 상세하게 다룬다.

에이전트 언어는 에이전트를 개발하기 위해 에이전트 특성을 고려한 특정 프로그래밍 언어를 개발하는데 초점을 맞춘다. 에이전트 언어의 기원은 Actor와 같은 병렬 객체 언어(concurrent object language)라고 간주하며 에이전트 기반 프로그래밍(agent oriented programming, AOP)[18]에서는 에이전트 이론에서 제시된 인지요소를 직접 이용하여 에이전트를 프로그래밍하는 기반구조와 언어를 연구하였다. 에이전트 언어는 크게 AOP, PLACA, ABLE와 같은 에이전트 기술언어와 Java, Tcl, Telescript [20]와 같은 에이전트 개발 언어로 구분되는데 상세한 내용은 4장에서 다룬다.

에이전트 응용은 이론과 구조 등을 기반으로 실제 사용할 수 있는 응용 소프트웨어를 개발하는 것이다. 인터넷 정보검색, 온라인 쇼핑, 메시징, 네트워크 관리등 에이전트가 필요로 하는 분야는 매우 많다. 에이전트 응용은 5장에서 좀더 상세히 소개한다.

3. 에이전트 구조

3.1 에이전트 구성요소와 유형

하나의 에이전트는 기본적으로 에이전트 엔진, 영역지식(domain knowledge), 통신 모듈 등으로 구성된다. 에이전트 엔진은 에이전트 생성과 작업수행, 에이전트 종료 등의 일련의 작업을 조정하기 위한 제어지식과 추론능력 등을 가진다. 영역지식은 특정 응용분야의 작업수행에 필요한 지식으로써 에이전트의 역할을 특장지워주는 부분이다. 에이전트는 생성시에 자신의 영역지식과 관계된 작업능력을 공개함으로써 다른 에이전트와의 작업 공유를 시도한다. 에이전트 통신은 다른 에이전트와의 메시지 교환을 담당하는데 대부분 자신이 해결하지 못하는 문제에 대하여 다른 에이전트의 도움을 청하는데 많이 이용된다.

에이전트 통신의 주된 목적이 다른 에이전트

와의 협동을 통한 문제 해결이라고 할 때 몇 가지 문제점이 발생한다. 즉, 다른 어떤 에이전트가 존재하는 지, 또는 다른 에이전트가 해결할 수 있는 능력이 무엇인지를 알고 있어야 한다는 것이다. 그렇지 않고는 누구에게 도움을 청할 지 판단할 수 없기 때문이다. 또 다른 문제점은 모든 에이전트끼리 서로 연결되어 있어야 하기 때문에 발생하는 네트워크 사용의 비용과 성능 저하를 들 수 있다. 이런 문제점을 해결하기 위한 접근 방향이 바로 멀티 에이전트 시스템이다.

분산 환경에서의 에이전트 구조하에서는 크게 3가지 유형의 에이전트가 존재하는데, 인터페이스 에이전트, 타스크 에이전트, 그리고 정보 에이전트가 그것이다. 인터페이스 에이전트(interface agent)는 사용자와 상호작용을 하며 사용자의 질의를 받아 분석하고 결과를 보여주는 역할을 수행한다. 좀더 지능을 갖춘 인터페이스 에이전트는 사용자의 습성과 기호 등의 데이터를 취득하여 이를 모델링하고 이를 바탕으로 시스템의 조정작업에 활용하는 기법을 가진다. 예를 들어 사용자의 설정에 따라 전자우편을 분류하거나 여과해주는 작업은 인터페이스 에이전트에 속한다. 타스크 에이전트(task agent)는 주어진 작업에 대한 영역지식과 함께 다른 타스크 에이전트나 정보 에이전트의 능력 정보 등을 가지고 사용자가 요구한 작업을 실제 수행하는 에이전트이다. 이와 더불어 타스크 에이전트는 여러 정보 에이전트가 제공해주는 정보들에 대한 종합과 서로 상반된 정보에 대한 해결 등의 작업도 수행한다. 정보 에이전트(information agent)는 여러곳에 흩어져 있는 이형질의 정보 소스를 지능적으로 접근할 수 있는 기능을 제공한다. 정보 에이전트는 원하는 정보를 찾는 역할 뿐 아니라 정보의 변화상황을 감시하는 모니터링도 수행한다. 예를 들어 주식 시세에 대한 데이터를 계속 모니터링해주는 에이전트라면 정보 에이전트에 속한다고 볼 수 있다.

3.2 멀티 에이전트 시스템

멀티 에이전트 시스템은 하나의 에이전트로 해결하지 못하는 복잡한 문제의 해결을 위하여

여러 에이전트간의 협동이 필요하게 되었고, 이를 효과적으로 수행하기 위해 제안되었다. 멀티 에이전트 시스템은 여러 응용 에이전트외에 조정 에이전트(coordinating agent, facilitator)라는 중재자를 통해 메시지의 전달과 각 에이전트의 제어를 수행하게 된다. 이 경우 모든 응용 에이전트의 통신 메시지는 조정 에이전트를 통해 다른 에이전트로 전달된다. 따라서 각 응용 에이전트는 조정 에이전트와만 연결된다. 각 에이전트가 어떤 처리 능력이 있는지에 대한 정보는 응용 에이전트 생성시에 조정 에이전트로 등록되며, 조정 에이전트는 이를 바탕으로 통신 메시지를 해당 응용 에이전트에게 보낸다. 응용 에이전트는 다른 에이전트가 어떤 일을 할 수 있는지에 대하여 알 필요가 없고, 단지 필요시 통신 메시지를 조정 에이전트에게 보내어 맡기면 된다. 이와 같이 조정 에이전트는 응용 에이전트의 수행 능력을 통해 에이전트의 실체를 알 수 있는 에이전트 네임 서버의 역할을 수행하는 것이다. 멀티 에이전트의 일반적으로 연합 에이전트 구조(federated agent architecture)를 가지며[5], 하나의 멀티 에이전트 시스템에 하나의 조정자와 여러 에이전트가 계층구조를 이루고, 조정자는 다른 멀티 에이전트 시스템의 조정자와 메시지를 주고받는다. [2]에서는 멀티 에이전트 시스템을 분류하는 방법론을 제시하였다.

멀티 에이전트에서 발생하는 중요한 문제는 각 응용 에이전트의 이형질성이다. 물론 멀티 에이전트 기반구조와 응용 에이전트들을 처음부터 같이 개발했다면 이러한 문제가 없겠지만 기존의 응용 프로그램을 바탕으로 에이전트 기능을 추가한 형태의 에이전트인 경우는 기존 응용 프로그램의 특성에 따라 서로 다른 형태를 지니게 된다. 예를 들어 전자우편 에이전트를 통해 전자우편의 자동 분류나 자동 전달과 같은 기능을 구현하고자 할 때 Unix 메일 시스템의 에이전트와 MS Exchange 메일 시스템의 에이전트는 서로 상이한 메일 포맷과 전달 방법으로 인해 그 역할을 수행하기 어려운 것이다. 이와 같은 이형질의 응용 에이전트간의 통신을 위해서는 표준화된 메시지 형태와 전달 프로토콜이 있어야 한다. 대표적으로 KQML

[7]이 있는데 KQML의 통신 메시지는 통신 계층(communication layer), 메시지 계층(message layer), 내용 계층(content layer)의 3단계로 나누어지고, 메시지 계층에서 표현된 언어의 종류와 영역 온톨로지(domain ontology)정보를 기술해주면 실제 내용 계층에는 서로 다른 형태로 표시되었더라도 서로 이해할 수 있고 통신이 가능하게 되는 것이다. 다음은 3가지 계층을 모두 나타내는 KQML 메시지의 간단한 예를 보여준다.

```
(PACKAGE:FROM ap001
:TO ap002
:CONTENT
(MSG
:TYPE query
:CONTENT-LANGUAGE KIF
:CONTENT (color snow-C)
```

이 메시지는 ap001이라는 에이전트가 ap002라는 에이전트에게 눈의 색깔이 무엇인지를 요청하는 것이다. 여기서, :CONTENT (color snow-C)가 내용 계층을 나타내며(MSG ...)로 표시된 부분이 메시지 계층이고, 메시지를 감싸고 있는(PACKAGE ...)부분이 통신 계층이다. 메시지 계층은 내용 계층에서 사용된 언어가 무엇인지(이 경우는 KIF [9]), 어떤 종류의 메시지인지(이 경우는 다른 에이전트에게 요청하는 것이므로 query) 등을 표시한다. 특히 query와 같은 메시지의 타임을 결정하는 것을 수행자(performative)라고 부른다.

4. 에이전트 언어

에이전트 언어는 에이전트 이론에서 제시된 여러 정형화된 개념을 이용하여 에이전트를 프로그래밍할 수 있는 시스템을 가리킨다. 에이전트 언어는 크게 에이전트 기술언어(agent specification language)와 에이전트 개발 언어(agent programming language)로 구분된다.

에이전트 기술 언어는 에이전트의 속성과 행동 등을 기술하는데 주로 에이전트가 수행할 수 있는 기능, 목적, 계획, 지식, 믿음 스키마(belief schema) 등의 형태로 에이전트를 표시

하게 된다. 대표적인 에이전트 기술 언어인 AOP [18]는 3가지 요소로 구성되는데 1) 에이전트의 mental state를 정의하는 논리 시스템, 2) 에이전트를 프로그래밍하기 위한 해석적 프로그래밍 언어, 3) 에이전트 프로그램을 컴파일하기 위한 에이전트화 프로세스 등이다. AOP의 향상된 버전으로 PLACA가 있다.

에이전트 개발 언어는 바로 수행 가능한 에이전트를 프로그래밍하기 위한 언어로써 대부분 해석 언어(interpreted language)와 객체 지향기법을 사용하여 플랫폼에 독립적인 에이전트의 개발을 가능하게 한다. 에이전트 개발 언어로는 Java, Tcl/Tk, Telescript 등을 들 수 있는데 특징으로는 스크립트 언어와 유사하며 에이전트 프로그램이 컴파일되지 않고 목적지 호스트에서 해석적으로 수행된다. 특히 Java와 Telescript는 이동 에이전트 개발과 관련되는 프로그래밍 언어로써 유사점은 모두 객체 지향이고 해석적 언어이며 가상수행 머신(virtual machine)개념을 도입하여 특정 플랫폼에 독립적인 에이전트를 생성하는 것이고, 차이점으로는 Java가 클라이언트의 요청에 의하여 서버의 프로그램이 클라이언트로 이동하여 수행되는 것인데 비해 Telescript는 반대로 클라이언트가 go 명령어 등을 사용하여 구성된 에이전트가 서버로 이동하여 수행되는 점이다. 따라서 어느 것이 더 좋다는 것 보다는 두 가지 언어가 서로 상호 보완적인 관계에 있다고 하겠다.

5. 에이전트 응용

사용자의 작업을 대신해주는 에이전트의 필요성은 우리 일상 생활 도처에서 찾아볼 수 있다. 여행사, 회사 비서, 도서관 사서, 판매대리점 등의 역할이 바로 에이전트가 적용될 수 있는 분야라고 할 수 있다. 정보화 사회의 진전으로 일상 생활의 복잡도와 다양성이 증가함에 따라 점차 모든 부분에서 에이전트의 비중이 커질 것이다. 이와 같이 에이전트가 적용될 수 있는 분야는 매우 넓지만 그 중에서도 인터넷 정보 검색, 전자상거래, 이동 컴퓨팅 분야에 대한 요구가 매우 많다고 할 수 있다. [13]은 제

품을 중심으로 에이전트 응용에 대한 전반적인 고찰을 하고 있다.

5.1 인터넷 정보검색

아마도 현재 에이전트의 필요성이 가장 요구되고 또한 많은 기술연구와 제품개발이 이루어지고 있는 분야가 인터넷상의 정보 검색과 정보 여과 등에 대한 응용 에이전트일 것이다 [15]. 인터넷, 특히 웹의 보편화로 인하여 정보를 검색하는 일뿐 아니라 자신의 정보를 알리는 작업도 용이해졌기 때문에 웹 상의 정보의 수는 상상을 초월할 정도로 많아졌으며 시시각각으로 변하고 있다. 이러한 정보의 홍수 속에서 사용자 스스로 원하는 주제에 대한 자신의 정보를 정확하고 빠른 시간에 얻는 것은 매우 어렵기 때문에 에이전트의 도움을 필요로 한다.

인터넷 정보 검색에 대한 사용자 편의를 도모하기 위해서 검색 엔진등이 국내외에 많이 개발되어 사용되고 있는데 사용자로 하여금 원하는 주제에 대한 입력을 받아서 관련 웹 사이트를 출력해준다. 특히 검색 엔진의 인덱싱 루틴은 탐색 로봇(search robot)이라 불리며 계속적으로 새로운 웹 사이트를 찾아 인덱싱 작업을 생성한다. 하지만 검색 엔진 자체를 에이전트라고 간주하기는 어렵는데 그 이유는 다음과 같은 기능이 부족하기 때문이다.

- 사용자의 검색 요구가 있어야만 정보를 가져오므로 주식정보와 같이 제 시간에 보아야 하는 실시간 정보 등에 대한 효과적인 검색이 어렵다.
- 흩어져 있는 정보의 형태가 텍스트, 그림, 음성 등으로 다양하기 때문에 관련 주제에 대한 정보를 모두 습득하기 어렵다.
- 사용자에 대한 습성이나 관심도 등에 대한 지식을 고려하지 않으므로 복잡한 질의를 반복하여야 한다.
- 단어에 대한 문맥처리등이 되지 않기 때문에 여러 의미를 지닌 단어를 통한 검색은 불필요한 정보를 함께 동반하므로 이를 여과하는데 노력을 낭비한다.
- 검색 엔진이 사용자에 의해 길들여지는 것이 아니고, 사용자가 엔진의 기능이나 형

식에 적응하도록 길들여져야 한다[14].

에이전트의 모습을 갖춘 인터넷 정보 검색으로는 WAIBA, WebAnts 등이 있다. 이 중에서 WAIBA는 변환자(transducer)와 보조자(associate)와 같은 독립 프로세스를 두어 서버와 클라이언트 중간에서 수행시킴으로써 사용자의 입력이나 서버로부터의 결과를 수정하여 여러가지 다양한 서비스를 제공한다. WAIBA가 구현한 서비스로는 웹 사이트의 내용이 변했을 때 자동으로 알려주는 기능, 웹 사이트의 내용에 대해 그룹원끼리 주석을 달아 서로 공유하는 기능, 웹 문서가 다른 웹 문서와 어느 정도 연관이 있는지를 나타내주는 기능 등이 있다. 인터넷 정보처리와 관련된 에이전트의 연구를 정리하면 다음과 같으며 [4, 21]에서 검색엔진과 인터넷 에이전트의 연관성을 비교하였다.

- Harvest(<http://mordor.transarc.com>)
- WebAnts(<http://polarbear.eng.lycos.com/webants>)
- WAIBA(<http://www.osf.org/www/waiba>)
- SAIRE(<http://saire.ivv.nasa.gov/saire.html>)
- IWAH(<http://rbse.jsc.nasa.gov/agents>)
- CIIR(<http://ciir.cs.umass.edu/info/ciir.html>)

5.2 온라인 전자상거래

온라인 쇼핑을 포함한 전자상거래는 국내외를 막론하고 이미 활발하게 진행되고 있으며 그 거래 규모또한 계속 증가하고 있다. 초고속 통신망과 가상현실 기술이 보편화되면 전자상거래는 더욱더 그 범위를 넓혀갈 것으로 기대된다. 하지만 사용자의 입장에서는 많은 상거래 사이트중에서 자신이 원하는 물품이 어디에 있는 지 찾기로 매우 어려운 일이며, 많은 온라인 쇼핑 사이트에 대한 정보를 기억하고 있어야 하고, 새로운 카탈로그에 대한 정보를 수시로 점검해야 한다. 또한 신용카드 등을 통하여 대부분 대금을 지불하므로 개인 정보 유출에 대한 불안감을 가지고 있는 것도 사실이다.

에이전트는 사용자의 전자상거래에 따른 이러한 불편함과 불안감을 해결하는데 초점을 맞춘다. 한 예로, BargainFinder(<http://bf.cstar.ac.com/bf>)는 음악 CD를 저렴한 가격에 구입하고자 하는 사용자들을 위해 7개 정도의 온라인 CD 판매 사이트를 연결하여 사용자가 원하는 CD의 가격을 한눈에 알아보도록 보여준다. 사용자 개인정보 유출을 방지하기 위해서는 이동 에이전트 개념을 사용하게 되는데, 기존의 메시지 방식으로 보내는 개인정보 전송을 지양하고 에이전트가 직접 판매 호스트로 이동한 후에야 비로소 개인정보를 이용한 작업이 이루어지게 함으로써 보안이 유지되도록 하는 연구가 진행되고 있다.

전자상거래는 비단 사용자에게만 도움을 주는 것은 아니다. 사업자 측면에서도 광고비등이 절약되므로 경쟁력이 생기게 되고 사용자의 요구사항이나 불편사항을 바로 받아서 해결해 줄 수도 있기 때문이다. 온라인 쇼핑은 다음에 열거한 것처럼 해외의 Internet Shopping Network 등을 비롯해서 국내에서도 몇몇 백화점을 포함하여 웹 브라우저 상에서 거래를 할 수 있는 서비스가 제공되고 있다.

- Internet Shopping Network(<http://www.internet.net>)
- Market Plaza Shopping(<http://bank.marketplaza.com>)
- 사이버스토어(<http://korea.directory.co.kr/shopping/cyber.html>)
- 가상서점(<http://www.bookshop.co.kr>)
- 롯데쇼핑(<http://internet.shopping.co.kr>)

5.3 사용자 인터페이스

아이콘이나 메뉴 사용으로 대변되는 그래픽 사용자 인터페이스(GUI)의 개발로 많은 컴퓨터 이용자의 편의를 가져올 수 있었지만 GUI 만으로는 사용자의 불편함을 완전히 해소하지는 못한다. 대부분의 사용자는 자신이 원하는 작업에 대한 내용(what to achieve)만 기술하고 실제 그것을 어떤 과정을 거쳐 달성하는지(how to achieve)는 컴퓨터가 알아서 해주기를 바란다. 에이전트의 역할이 바로 그것이다.

사용자 인터페이스에 대한 에이전트는 크게 다음과 같은 3가지 역할을 수행한다고 볼 수 있다. 첫째, 사용하기 쉬운 인터페이스를 제공하는데 사용자 개개인의 패턴을 반영하고 매크로 등을 사용하여 반복적인 작업에 대한 편리성을 제공한다. New Wave나 Open Sesame와 같은 제품이 이에 속한다. 둘째, 학습을 통해 사용자의 습성을 파악하여 적용시키는 학습인터페이스 에이전트이다. 주로 전자우편관리나 개인 일정관리와 같이 데스크탑 응용에서 발생하는 사용자의 처리 유형을 반영하게 된다[6]. 셋째, 멀티미디어 인터페이스를 제공하는 에이전트이다. 음성이나 필기체 입력 등을 가능하게 함으로써 자연스러운 인터페이스를 지원한다. IBM의 파워PC 시스템이 대표적이다.

5.4 메시징

메시징은 전화메시지, 전자우편, 팩스와 같은 온라인 메시지에 대하여 지능적인 연결과 메시지 여과작업을 수행한다. 주로 지역적으로 떨어진 곳으로는 정보 전달과 이동 통신 시스템의 이용증가에 따라 이동 컴퓨팅과 연계된 이동 에이전트의 연구개발이 진행되고 있다.

이동 에이전트는 기존의 원격함수호출(RPC)과 다른 개념인 원격 프로그래밍(Remote Programming)의 개념을 이용하여 에이전트를 목적하는 원격 호스트로 이동시켜 그곳에서 수행하도록 한다[11]. 이러한 원격프로그래밍은 이동 통신 체계에서 문제가 되었던 통신 회선의 점유문제를 해결하여 항상 전화선이 연결될 필요없이 이동 에이전트를 보낼때나 결과를 받을 때만 연결되게 함으로써 비용절감의 효과를 가져오게 하였다. 하지만 에이전트의 권위 정도 파악이나 자원 할당 문제 등이 시급한 해결과제로 남아있다.

대표적인 메시징 에이전트로는 PDA와 연관된 Telescript와 Newton이 있고 Personalink, SIFT, Tapestry가 상용화되고 있다.

6. 결 론

지금까지 에이전트의 정의와 필요성, 구조, 언어, 응용 등의 각 분야에 대하여 정리해보

았다. 사용자의 간섭없이 사용자의 의도대로 능동적으로 행동하는 에이전트의 기술은 이제 적용되지 않는 분야가 없을 정도로 확산되고 있다. 현재도 그렇지만 앞으로의 추세는 역시 인터넷과 관련된 에이전트 개발이 많은 비중을 차지할 것으로 예측되며 특히 초고속 정보망과의 연계를 통하여 전자상거래도 활성화될 것이다. 특히 이동 전화와 PDA 등의 발달로 이동 컴퓨팅을 지원해주는 이동 에이전트의 중요성은 더욱 커질 것이다. 이밖에도 계획이나 학습 등을 위한 지식의 보강으로 좀 더 지능을 갖춘 에이전트의 개발이 기대된다.

하지만 이러한 기대만큼 해결해야 할 과제들도 많다. 가장 우려되는 것이 보안과 사용자 식별에 대한 문제이다. 전자상거래등은 특히 이에 민감한 부분으로 신용카드번호와 같은 개인 신상에 대한 사항들이 인터넷 상을 무수히 오갈 것으로 예측되지만 이러한 정보에 대한 유출을 방지할 수 있는 기법이 절실히 요구된다. 이동 에이전트의 경우 상대방 호스트의 자원을 이용하여 작업을 수행하기 때문에 외부에서 온 에이전트에 대하여 어느 정도 자원을 허락할지, 내 자원을 파괴시키는 악성 에이전트는 아닌지에 대한 판단 루틴도 개발되어야 할 것이다. 이러한 문제점들이 보완되면서 에이전트는 이제 우리 생활의 일부분으로 자리잡을 수 있다.

끝으로 에이전트 연구에 대한 관련자료를 필요로 하는 독자를 위하여 에이전트의 소개와 연구 주제 등을 분류해 놓은 웹 사이트를 추천하고 싶다.

- UMBC intelligent agent(<http://www.cs.umbc.edu/agents>)
- intelligent software agent(<http://www.sics.se/isl/abc/survey.html>)

참고문헌

- [1] Bhandaru N. and Croft W., "An architecture for supporting goal-based cooperative work," in Gibbs S. and Verriin-Stuart A., eds., *Multi-User Interfaces and Applications*, pp 337-354, Elsevier Science Publishers B.V., North-Holland, 1990.
- [2] Bird S., "Toward a taxonomy of multi-agent systems," *Int. J. of Man-Machine Studies*, Vol. 39, pp 689-704, 1993.
- [3] Castelfranchi C., "Guarantees for autonomy in cognitive agent architecture," in Woolridge M. and Jennings N., eds., *Intelligent Agents: ECAI Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1995.
- [4] Cheong F., *Internet Agents: Spiders, Wanderers, Brokers, and Bots*, New Riders, 1996.
- [5] Cohen P., Cheyer A., Wang M., and Baeg S., "An open agent architecture," *Working Notes of AAAI Spring Symposium on Software Agents*, pp 1-8, 1994.
- [6] Etzioni, O. and Weld D., "A softbot-based interface to the internet," *Comm. ACM*, Vol. 37, No. 7, pp 72-79, 1994.
- [7] Finin T., Fritzson R., McKay D., and McEntire R., "KQML as an agent communication language," *Proc. of CIKM '94*, pp 126-130, 1994.
- [8] Franklin S. and Graesser A., "Is it an agent, or just a program? : A taxonomy for autonomous agents," *Proc. of Third International Workshop on Agent Theories, Architectures, and Languages*. 1996.
- [9] Genesereth M. and Fikes R., "Knowledge interchange format version 3.0 reference manual," Technical Report Logic-92-1, Computer Science Department, Stanford University, 1992.
- [10] Genesereth M. and Ketchpel S., "Software agents," *Comm. ACM*, Vol. 37, No. 7, pp 48-53. 1994.
- [11] Gifford D. and Stamos J., "Remote evaluation," *ACM Trans. on Programming Language and Systems*, Vol. 12, No. 4, pp 537-565, 1990.
- [12] Goodwin R., "Formalizing properties of

- agents," Technical Report CMU-CS-93-159, School of Computer Science, Carnegie Mellon University, 1993.
- [13] Guilfoyle C. and Warner E., *Intelligent Agents: the New Revolution in Software*, Ovum Limited, 1994.
- [14] Hayes-Roth B., "An architecture for adaptive intelligent systems," *Artificial Intelligence*, Vol. 72, pp 329-365, 1995.
- [15] Knoblock C. and Arens Y., "An architecture for information retrieval agents," *Working Notes of AAAI Spring Symposium on Software Agents*, pp 49-56, 1994.
- [16] Maes, P., "Artificial life meets entertainment: Life like autonomous agents," *Comm. ACM*, Vol. 38, No. 11, pp 108-114, 1995.
- [17] Schwartz D., "Cooperating heterogeneous systems: A blackboard based meta approach," Technical Report 93-112, Center for Automation and Intelligent Systems Research, Case Western Reserve University, 1993.
- [18] Shoham Y., "Agent-oriented programming," *Artificial Intelligence*, Vol. 60, No. 1, pp 51-92, 1993.
- [19] Singh M., Huhns M., Stephens L., "Declarative representations of multiagent systems," *IEEE Trans. on Knowledge and Data Eng.*, Vol. 5, No. 5, pp 721-739, 1993.
- [20] White J., "Telescript technology: The foundation for the electronic marketplace," White Paper, General Magic, 1994.
- [21] Williams J., *Bots and Other Internet Beasties*, Sams.net Publishing, 1996.
- [22] Wooldridge M. and Jennings N., *Intelligent Agents*, Lecture Notes in Artificial Intelligence #890, Springer-Verlag, 1995.

최 중 민



1984 서울대학교 컴퓨터공학과 졸업(학사)
 1986 서울대학교 대학원 컴퓨터 공학과 졸업(석사)
 1993 State University of New York at Buffalo, Computer Science 졸업(박사)
 1993~95 한국전자통신연구소 인공지능연구실 선임 연구원
 1995~현재 한양대학교 전자계산학과 전임강사
 관심분야: 지능형 에이전트 시스템, 인공지능, 정보검색, 데이터베이스, HCI

● 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