

소프트웨어 에이전트 분류 및 기술동향

박수현* · 김태석**

1. 에이전트 개요

근래에 들어 사전적으로는 “대행해 주는 사람”, “대리인”의 뜻을 가진 에이전트란 용어는 스포츠, 증권, 금융, 예술, 컴퓨터공학, 전산학 등 여러 분야에서 사용되고 있다. 특히 인공지능, 컴퓨터 네트워크, 분산시스템, 데이터마이닝 및 레지스트리와 같은 컴퓨터관련 분야에서는 “어떤 기능을 대신하여 수행하는 응용 소프트웨어”를 의미한다. 에이전트는 이용자가 자세한 지시사항을 일일이 제시하지 않고 작업에 필요한 일부지식만을 제공해 주면 그것만으로도 목적에 필요한 작업을 스스로 판단해 처리하는 기능을 갖게 하고 있다[1]. 이렇듯 에이전트가 거의 모든 컴퓨터 분야에서 응용되고 있는 이유는 과연 무엇일까? 가장 커다란 이유는 컴퓨터 네트워크의 발달이라고 할 수 있다. 또한 에이전트 자체가 가지는 여러 가지 특징들, 자율성, 이동성, 사회성, 지능 등에 기인한다고 할 수 있다[2,3]. 위의 특징들은 마치 에이전트가 하나의 유기체와 같은 속성을 가지고 있는 것과 같은 착각에 빠지게 한다. 자율성이란 에이전트가 사용자나 다른 프로그램의 직접적인 지시나 간섭 없이 스스로 판단하여 작업을 수행하는 능력을 의미하며 이는 에이전트가 갖는 가장 기본적이고도 중요한 기능이다. 사회성은 에이전트들이 다른

에이전트와 협력을 통하여 작업을 수행할 수 있는 능력을 말한다. 이동성은 데이터와 그 처리 순서를 하나로 정리해 컴퓨터 통신망을 통해 상대방에게 전송하여 필요한 기능을 수행하도록 하는 속성을 의미하며 지능은 각각 일정한 지식을 갖게 하여 에이전트가 다음에 수행하여야 할 기능을 스스로 학습하며 에이전트들 사이의 협조 체제를 구축하는 데 필요한 속성이다.

에이전트에 대한 분류는 지원기능, 기능원리 등에 따라 여러 종류로 구분할 수 있다. 우선 에이전트의 지원기능으로서 전자비서 에이전트, 상업 에이전트, 로봇 에이전트, 정보검색 에이전트 등으로 구분할 수 있으며 기능원리에 따른 분류로는 협력 에이전트, 이동 에이전트, 인터넷 에이전트, 지능형 에이전트 등으로 구분할 수 있다. 이외에도 반응성 에이전트, 이형 에이전트, 혼합형 에이전트 등이 있다. 네트워크의 부하를 줄이고 전체적인 네트워크 시스템의 효율성을 향상시키기 위해서는 이동 에이전트의 개념이 적합한데 이동 에이전트는 이동 컴퓨팅의 개념을 바탕으로 에이전트를 네트워크내의 상대 컴퓨터로 보내어(여기서 상태 컴퓨터는 또 다른 클라이언트나 네트워크 내의 서버가 될 수 있다.) 그곳에서 작업을 수행한 다음 처리된 결과만을 되돌려 받는 개념이 된다. 이러한 개념을 지원하기 위하여 클라이언트 에이전트를 이형질(heterogeneous) 서버 플랫폼으로 파견(dispatch)하거나 이형질의 다른

*동의대학교 컴퓨터공학과 교수

**중신회원, 동의대학교 컴퓨터공학과 교수

네트워크로 파견 시 이주(migration) 관련 기술과 에이전트 환경 정합 및 보호 문제에 대하여 연구가 진행 중에 있다[4]. 다행하게도 플랫폼 독립형 언어인 JAVA와 이형질 리저시 시스템의 통합을 지원하는 CORBA의 출현은 이와 같은 이주관련 및 에이전트 환경정합 문제에 대한 해결책을 제시해 주고 있다. 하지만 이형질 환경으로 파견된 에이전트 자신의 보호문제는 아직도 많은 문제점을 가지고 있다. 이와 같은 여러 문제들은 결국 에이전트 사이의 통신언어의 표준화와 관련이 된다. 이는 특히 협력 에이전트(collaborative agent) 분야에서 매우 중요한 부분으로서 에이전트들이 상호정보를 교환하기 위하여 사용하는 메시지 형식 및 의미를 정의한다. 이에 대한 연구는 미국방성의 KSE(Knowledge Sharing Effort)가 주축이 되어 진행한 KQML(Knowledge Query and Manipulation Language)이 가장 널리 알려져 있다[5]. 본 고에서는 이러한 에이전트들의 국내외 기술동향에 대하여 알아보기로 한다.

2. 에이전트 특징

에이전트가 가져야하는 가장 중요한 특징은 무엇보다도 자율성(autonomy)이다. 자율성은 에이전트가 다른 소프트웨어와 구분할 수 있는 대표적인 특징으로서 에이전트가 사용자나 다른 프로그램의 직접적인 지시나 도움없이 스스로 판단하여 주어진 임무를 수행함을 의미한다. 에이전트 사용자의 지시에 따라 사용자를 대신하여 주어진 임무를 수행하기 위하여는 스스로 판단하여 적절한 시기에 적절한 프로세스를 추진하고 필요한 경우에는 사용자의 지시가 없어도 상황을 판단하여 새로운 임무를 추가로 수행할 수 있다. 이를 위하여 에이전트가 갖는 여러 중요한 특성은 협동성(cooperation), 사회성(sociality), 지능(intelligence),

학습능력(learning), 이동성(mobility) 등을 들 수 있다[3].

협동성(cooperation) 및 사회성(sociality)은 네트워크 내에 있는 여러 에이전트들이 상호간의 협력을 통하여 주어진 임무를 수행하는 특성을 의미한다. 이는 에이전트들을 마치 하나의 유기체 공동체와 같이 인식하여 에이전트를 하나의 독립적인 소프트웨어 프로그램이 아닌 에이전트 유기체 사회의 구성원으로 인식하는 것이다. 협동성과 사회성을 통하여 에이전트는 독립적으로 임무를 수행할 때 보다 더 많은 다양한 업무를 효율적으로 처리할 수 있게 된다. 지능(intelligence) 및 학습능력(learning)은 에이전트가 사용자의 요구패턴이나 자주 사용하는 문자 및 특수기호 등을 기억하고 있다가 사용자의 요구사항을 지능적으로 분석한 후 처리결과를 사용자가 선호하는 출력패턴으로 사용자에게 제공하는 능력을 의미한다. 또한 에이전트 사용자의 지시에 따라 사용자를 대신하여 주어진 임무를 수행하기 위하여는 스스로 판단하여 적절한 시기에 적절한 프로세스를 추진하고 필요한 경우에는 사용자의 지시가 없어도 상황을 판단하여 새로운 임무를 추가로 수행하는 기능 또한 이 특성에 속한다고 볼 수 있다.

이동성(mobility)은 에이전트가 네트워크내의 다른 컴퓨터 시스템으로 파견(dispatch), 이동되어 주어진 임무를 수행하는 특성을 말한다. 예를 들어 데이터마이닝을 주목적으로 하는 시스템이 있다고 하자. 대규모데이터베이스(VLDB)내에 저장되어 있는 대규모의 데이터로부터 의사결정에 필요한 정보를 찾는 데이터마이닝을 이동성 특성을 갖는 에이전트로 이용하지 않을 경우 데이터베이스 내에 있는 엄청난 양의 자료들을 네트워크를 이용하여 모두 ftp와 같은 기능을 이용하여 가져와서 처리해야만 했다. 하지만 이동성 특성을

갖는 에이전트의 경우 에이전트(데이터마이닝 기능을 갖는 응용프로그램)가 해당 데이터베이스로 이동하여 마이닝을 수행한 후 그 결과만을 사용자에게 전달해 줌으로서 전체 네트워크 시스템의 성능뿐만 아니라 작업의 속도도 훨씬 향상시킬 수 있게 된다. 이외에도 에이전트는 상황에 따라서 자신이 가지고 있는 정보 및 지식에 따라 환경에 반응을 하는 반응성(reactivity)과 합리성(rationality)의 속성을 가져야 한다.

하나의 에이전트로 처리하기 어려운 매우 복잡한 작업의 경우, 에이전트의 협동성을 이용하여 다른 여러 에이전트들의 도움이 필요한 데 이를 위하여는 에이전트간 협동 메커니즘이 필요하다. 이와같은 특성을 지니는 에이전트를 다중 에이전트(multi-agent) 또는 분산 에이전트(distributed agent)라고 한다. 향후에는 네트워크의 부하를 줄이고 네트워크 전체의 성능을 향상시키기 위하여는 이동 컴퓨팅의 개념을 바탕으로 한 이동 에이전트의 개념이 확산될 것으로 전망되며 특히 인터넷상에서의 데이터 검색엔진 분야에서 로봇 에이전트의 개념으로 적극적으로 활용되고 있다.

3. 에이전트 분류

에이전트에 대한 분류는 지원기능, 기능원리 등에 따라 여러 종류로 구분할 수 있다. 에이전트의 분류는 매우 다양하여 플랫폼 독립적인 언어로 주로 작성되는 소프트웨어 에이전트, 하드웨어를 기반으로 동작하는 하드웨어 에이전트, 에이전트간의 협동 메커니즘을 이용하여 동작하는 복합형 에이전트가 있다[6]. 복합형 에이전트에는 다중에이전트(multi-agent)[7] 또는 분산에이전트(distributed agent) 등이 있으며 복합형 에이전트와는 달리 단독적으로 행동하는 에이전트를 단독

형 에이전트라 한다. 우선 에이전트의 지원기능으로서 분류하면 전자비서 에이전트, 상업 에이전트, 로봇 에이전트, 정보검색 에이전트[8] 등으로 구분할 수 있으며 기능원리에 따른 분류로는 협력 에이전트, 이동 에이전트, 인터페이스 에이전트, 지능형 에이전트 등으로 구분할 수 있다. 이외에도 반응성 에이전트, 이형 에이전트, 혼합형 에이전트 등이 있다. 네트워크의 부하를 줄이고 전체적인 네트워크 시스템의 효율성을 향상시키기 위해서는 이동 에이전트의 개념이 적합한 데 이동 에이전트는 이동 컴퓨팅의 개념을 바탕으로 에이전트를 네트워크내의 상대 컴퓨터로 보내어 (여기서 상대 컴퓨터는 또 다른 클라이언트나 네트워크 내의 서버가 될 수 있다.) 그곳에서 작업을 수행한 다음 처리된 결과만을 되돌려 받는 개념이 된다.

3.1 전자비서 에이전트

전자비서 에이전트는 단순 반복적이면서도 하지 않으면 안되는 일상업무와 관련하여 각종 업무를 사람을 대신하여 처리해 주는 지능형 에이전트이다. 전자비서 에이전트는 다음과 같은 역할을 담당한다.

- 일/주/월/분기 별 개인/팀 스케줄을 관리
- 전자메일처리 : 수없이 도착하는 전자메일 중 스팸메일을 판단하고 실제 업무 및 개인 생활에 필요하다고 판단되는 메일들만을 사용자가 정해진 일정한 기준에 맞추어 정리하여 사용자에게 보여주고 나머지에 대하여는 답변여부를 판단하여 스스로 응답을 해 준다.

전자비서 에이전트의 개념은 80년대 중반 몇몇 미래학자들에 의해 그 중요성이 강조되었다. 80년대 후반에 접어들어서는 미국의 휴렛팩커드, 에

플, 디지털 등 여러 컴퓨터 관련업체들이 미래의 컴퓨터 인터페이스로서 전자비서 에이전트를 소개하였다. 1987년 홍보 비디오용으로 소개된 애플사의 Knowledge Navigator는 이중 하나로, 나비넥타이를 맨 비서가 컴퓨터 화면에 나타나 사용자에게 유용한 정보를 제공해 주면서 일상업무를 처리하는데 있어서 비서와 같은 역할을 해주는 모습을 보여주었다.

최근들어 한국과학기술원, 숙명여대 등 각 대학을 중심으로 사이버 공간 내에서 활동하는 사이버 학생 도우미의 개념을 도입하여 학사일정, 병무, 학점관리, 증명서발급 신청접수, 대외홍보 등의 역할을 담당하고 있는 데 이들이 초보적 단계의 전자비서 에이전트의 예라고 할 수 있다.

지능형 전자비서 에이전트는 지식베이스, 전문가 시스템, 자연어 처리, 패턴인식 등 인공지능의 관련된 여러 분야가 조금 더 발전되어야만 활성화 되겠지만, 초보적인 형태의 전자비서 에이전트는 이미 여러 연구기관에서 활발히 연구 개발되고 있다. AT&T의 Bell 연구소의 Kautz 연구팀은 연구소를 찾아오는 방문객과의 미팅 스케줄을 짜주는 실험적인 전자비서 에이전트 시스템을 구축하였다. Visitorbot라고 불리는 이 전자비서는, 각 연구원들마다 userbot이라는 담당 에이전트를 두고 이들 에이전트와 교신함으로써 스케줄을 짜나가도록 구성되어 있다. Kautz와 그의 연구팀은 이 실험적인 에이전트 시스템을 구축하면서 몇 가지 중요한 점을 지적하였다. 그것은 사용자의 프라이버시, 보안, 그리고 개개인의 업무 방식을 에이전트가 얼마만큼 잘 고려하는가가 성공적인 에이전트의 관건이라는 것이다. 그리고 단일 에이전트로 모든 것을 해결하는 것보다는 여러 에이전트로 작업을 분할하여 수행하는 것이 보다 효과적으로 주어진 문제를 해결할 수 있다는 것이다. 즉,

visitorbot은 스케줄 짜는 부분에만 신경을 쓰고, 자료의 입력 과정 및 인터페이스는 userbot 에이전트들에게 맡기는 것이 보다 확장성이 뛰어난 전자비서 에이전트를 구축하는데 도움을 줄 것이라고 주장한다[8].

3.2 상업 에이전트

상업에이전트는 최근에 가장 주목을 받고있는 전자 상거래(EC / CALS : Electronic Commerce / Commerce At the Light Speed)에서 가장 중요한 역할을 담당하는 에이전트이다. 전자 상거래는 간단히 생각하면 상품의 주문, 주문받은 상품의 수배, 대금결제 등과 같은 단계를 거치게 된다. 또한 원하는 물품 또는 서비스를 받고자 하는 구매자의 입장에서선 전문가의 의견을 수렴하여 현명한 구입과 알찬 애프터서비스 등을 받고 싶어한다. 또한 생산자의 입장에서선 소비자의 취향과 소비자를 대상으로 한 전문적인 조언 등을 제공할 수 있어야 하며, 고객만족을 위한 서비스를 운영하여야 한다. 상업 에이전트는 온라인 상에서 고객과 생산자와의 중개역할을 해주면서 서로가 편하도록 각종 일을 자동화시키는 임무를 띄고 있는 에이전트이다. 전자 상거래 시스템의 기능 구성은 인터넷 쇼핑물을 지원하는 기능, 고객이 기존에 거래하였던 트랜잭션 자료로부터 고객의 성향을 분석하여 고객으로 하여금 원하는 상품을 최단시간에 효율적으로 선택할 수 있도록 해야한다. 또한 고객이 원하는 상품이 자신의 쇼핑물에 재고가 있는지 여부를 확인 후 없는 경우 고객의 욕구를 만족시켜 주기 위하여 인터넷상에서 해당 상품이 어디에 있는 지 수배를 해야하며, 대금결제 시 고객이 원하는 대금결제 수단에 최대한 부응해 주어야 한다. 이상 언급한 기능들은 모두 에이전트들에 의하여 수행이 가능한데 이동 / 지능 / 복합

에이전트의 특성을 지녀야 한다. 고객의 성향을 분석하기 위하여는 데이터웨어하우스 및 고객성향 분석 데이터마이닝 에이전트가 필요하다. 또한 고객이 원하는 상품의 존재여부를 확인하기 위하여는 기존의 재고관리 시스템만으로는 고객의 욕구를 만족시킬 수 없다. 만일 고객이 자신이 원하는 상품이 자신이 거래하는 해당 전자상거래 쇼핑몰에서 매번 재고부족이라는 응답만 듣는다면 고객은 다른 거래처로 신속히 접속하게 될 것이다. 치열한 경쟁이 벌어지는 전자 상거래 분야에서 이는 치명적인 시스템 결함이 된다. 따라서 대부분의 전자 상거래 시스템은 지능형 이동 에이전트를 이용하여 꾸준히 거래할 상품들에 대한 정보를 수집하여 자신의 전자 상거래 시스템 내의 저장소(repository)에 저장해 놓아야 한다. 또한 고객이 원하는 결제 수단, 즉, 현금, 신용카드, 직불카드, 전자화폐 등에 모두 부응하기 위하여는 인터넷 뱅킹만을 전담하는 신개념의 에이전트가 필요하게 된다.

3.3 로봇 에이전트

로봇 에이전트[9]란 웹을 순회하며 각 홈페이지들의 정보를 수집하는 프로그램이다. 웹 서버에 접속하여 데이터를 가져오는 기능적인 측면으로만 봐서는 웹브라우저와 같은 기능을 하는 역할을 한다. 단지 웹 브라우저는 가져온 데이터를 사용자의 취향에 맞게 화면에 보여주고 화면에 하이퍼링크가 있고 사용자가 이를 클릭하면 관련 홈페이지로 연결시켜주는 기능이 있으나, 로봇 에이전트는 사용자의 취향에 맞게 화면에 보여주는 대신 HTML을 분석하고 URL 부분을 추출하여 다른 URL로 접근하게 하는 역할이 주요 기능이다. 자동적으로 홈페이지를 찾아다니므로 로봇 에이전트를 이용한다면 사람이 할 수 있는 일이지만 지

루한 다양한 일을 할 수 있다. 가령 어느 홈페이지의 내용이 내가 원하는 내용이지만 링크되어 있는 문서가 굉장히 많고 HTML 안에 수많은 그림들이 있을 때 과연 모니터 앞에 앉아 하나씩 저장하는 과정이 필요한 것인가? 또는 홈페이지의 관리자가 자신이 관리하는 홈페이지를 하나씩 접속해보면서 잘못된 URL이 있으면 찾아가서 일일이 고칠 것인가? 로봇 에이전트는 이와 같이 웹 페이지를 돌아다니면서 할 수 있는 다양한 일들을 자동적으로 해 주는데 의미가 있다. 현재 로봇 에이전트를 이용한 검색엔진도 마찬가지로 사람이 일일이 찾아다니면서 검색을 위한 인덱스를 만드는 것이 아니라 로봇 에이전트가 자동적으로 돌아다니면서 원하는 정보를 수집하는 것이다. 현재 로봇 에이전트로 할 수 있는 일들은 다음과 같은 것이 있다.

- 통계 분석(Statistical Analysis)
- 유지 보수(Maintenance)
- 미러링(Mirroring)
- 리소스 발견(Resource discovery)
- 복합적인 사용(Combined Uses)

3.3.1 통계 분석(Statistical Analysis)

로봇 에이전트의 최초의 사용 목적은 전세계에 웹서버가 몇 개나 있는지 알아보기 위한 것이 주요 목적이었다. 웹에 소개된 많은 웹 서버들이 생겨나기 시작하였고, 어느 정도 활성화된 1993년에 MIT의 Matthew Gray (mkgray@mit.edu, <http://www.mit.edu:8001/people/mkgray/mkgray.html>)는 "World-Wide Web Wanderer" 로봇 에이전트를 이용하여 과연 전세계에 몇 개의 웹서버가 있는지 알아보기 시작하였다. Matthew Gray 뿐만 아니라 워싱턴 대학의 Brian Pinkerton도 웹서버의 수를 조사하였다. 1994년 5월 현재 약 3800여

개의 웹 서버가 있었음을 로봇 에이전트를 통하여 알 수 있었고 이후에 통계 분석용 프로그램을 서치엔진용 로봇 에이전트로 바꾸어 현재 WebCrawler (<http://webcrawler.com/>)라는 서치엔진을 운영 중이다. 웹서버의 개수 만 계산하는 로봇 에이전트뿐만 아니라 어떤 웹서버를 사용하는지 조사하는 로봇 에이전트도 있다. 국내에서도 한국 내의 웹서버가 어떤 서버 프로그램을 사용하고 있는지 알아보는 조사를 시행하였는데 1996년 3월 8일 현재 NCSA 서버가 가장 많이 사용되는 것으로 조사되었다. 이때 사용한 로봇 에이전트는 libwww-perl/0.40을 이용하여 작성되었는데 조사방법으로는 libwww-perl을 이용하여 HEAD 메소드로 국내의 WWW 서버를 접근하여 헤더 정보를 가져와 분석하는 방법으로 WWW 서버의 종류와 버전을 중심으로 조사하였다. 이렇듯 로봇 에이전트를 이용하여 웹서버의 발견, 서버의 수 카운트, 서버 당 문서의 평균수, 화일 타입의 분포 등의 통계 조사를 할 수 있다.

3.3.2 유지 보수(Maintenance)

웹서버를 운영하다 보면 웹서버의 유지 보수에 여러 어려움이 따른다. 웹서버 관리자는 개개인이 가지고 있는 화일에 대하여 일일이 어떤 부분이 잘못되었는지 어디에 데드링크(dead links)가 발생하였는지 알 수 없다. 데드링크란 원래의 URL이 변경되거나 없어져 버림으로써 기존의 사용자가 알고 있는 URL로써 접근이 불가능한 경우를 말한다. 이때 유용하게 사용할 수 있는 것이 로봇 에이전트이다. 쉽게 알 수 있는 유지보수용 로봇 에이전트는 바로 넷스케이프 브라우저 안에 내장되어 있다. 넷스케이프 브라우저의 Bookmark를 보면 File->What's New 부분이 있다. 이 메뉴의 기능은 현재 북마킹 되어 있는 URL을 브라우저가 접근하여 변경된 부분이 있는지 검사해 준다.

만약 URL의 내용이 북마킹 하였던 시점 이후에 변경되었다면 북마크 내의 URL의 타이틀 설명 옆의 마크가 변경된다.

URL-minder(<http://www.netmind.com/URL-minder/URL-minder.html>)에서는 홈페이지를 등록해 놓으면 변경되었을 때 등록된 사용자에게 변경되었다는 통보를 해 준다. 이 역시 로봇 에이전트를 이용하여 등록된 홈페이지를 검사하여 변경되었다면 - 변경된 것을 알 수 있는 방법은 다양하다. Last-modified 필드를 참조하여 알 수 있으며 Last-modified를 지원하지 않는 서버는 홈페이지의 사이즈를 비교하여 사이즈가 예전에 등록된 것과 틀려지면 변경되었다고 생각한다. - 그 홈페이지를 등록된 사용자에게 변경되었다고 통보한다.

3.3.3 미러링(Mirroring)

로봇 에이전트의 또 다른 용도는 다른 사람의 홈페이지의 내용을 복사해오는 미러링이다. 물론 저자의 허락을 받아야 하지만 각각의 페이지들을 가져오는 경우 그 내용이 많고, 그림이 많이 있다면 그림, HTML을 일일이 가져오는 것이 힘들다. 예를 들어 어떤 대학의 강의 노트가 있는데 강의 별로 수십~수백 페이지와 수많은 이미지가 있을 때 웹 브라우저로 한 페이지씩 접근하여 저장한다는 것은 많은 시간을 필요로 한다. 이때 유용하게 사용될 수 있는 것이 웹 미러링 툴이다. 많은 웹 미러링 툴이 나와 있지만, 일반적으로 미러링 툴은 웹 페이지들을 접근하고 로컬에 저장하는 프로그램이다. 현재 가장 많이 사용하고 있는 미러링 툴은 WebCopy(<http://www.inf.utfsm.cl/~vparada/webcopy.html>), w3mir(<http://www.ifi.uio.no/~janl/w3mir.html>) 등으로 이들 프로그램은 웹 페이지들을 접근하고 로컬에 저장하는 기능을 가지고 있다.

3.3.4 리소스 발견(Resource discovery)

서치엔진에 사용하는 용도로서 자동적, 주기적으로 웹 페이지를 항해하며 데이터를 수집하는 용도로 사용되는 것을 의미한다. 아래 그림은 인터넷상의 로봇 에이전트를 이용한 검색 엔진의 일반적인 형태이다. 로봇 에이전트는 URL 데이터베이스를 참조하여 WWW 서버를 항해하며 필요한 페이지를 수집하고, Index Builder는 수집된 자료를 처리하여 검색할 수 있는 형태로 변환한다. 서치엔진은 사용자와 변화된 형태의 자료간에 키워드를 받아 검색할 수 있는 중간자 역할을 한다.

그림 1은 리소스 발견과정을 보여주고 있다.

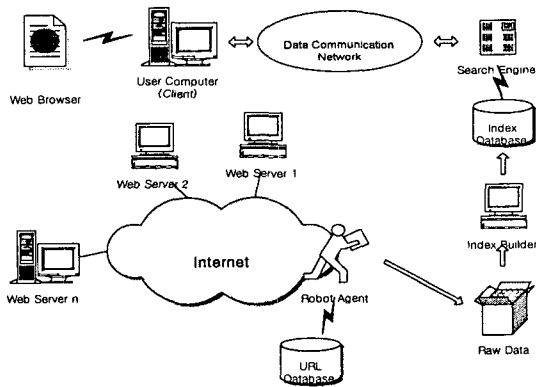


그림 1. 인터넷 상에서의 리소스 발견과정

3.3.5 복합적인 사용(Combined Uses)

현재 로봇 에이전트의 사용은 단일 목적으로 사용하기보다는 한 번의 로봇 에이전트의 동작으로 다양한 작업을 하고 있다. 단순히 통계적 목적으로 홈페이지의 수만을 검사하기보다는 서치엔진을 제공하면서 자신이 보유한 URL 데이터베이스 수를 검사하여 현재 수집한 홈페이지의 수를 알려주는가 하면, 어떤 로봇 에이전트는 특정 홈페이지의 내용을 차례차례 방문하고, 그 내용을 분석하여 알려줌과 동시에 특정 페이지 자료만을

가지고 키워드에 의한 검색이 가능하게 하기도 한다. ArchitextSpider는 통계적 목적과 인덱싱을 위한 것이고, NetCarta WebMap 엔진은 통계적 목적, 인덱싱, 유지보수, 미러링의 모든 기능을 가지고 있다.

로봇 에이전트가 다양한 분야에서 사용되고 있지만 여러 문제점을 가지고 있다. 우선 로봇에 대한 표준이 정해져 있지 않다[11]. 다양한 로봇이 작성되고 또한 로봇에 의해 데이터가 공유되려면 로봇에 대한 표준은 필수적이나 현재까지 로봇에 대한 표준은 언급된 바 없고 단지 접근에 대한 표준만이 제안되었을 뿐이다. 다음으로 로봇 프로그램은 브라우저와 달리 Clickable Image (Sensitive Map 또는 이미지 맵), Form 등을 처리하지 못한다. 일부 로봇은 이미지의 크기를 읽어 조밀하게 웹서버에 URL을 조정하며 시도하는 것이 제안되기도 하였으나 이는 서버의 부담만 가중시킬 뿐 실질적인 해결방안은 아니다. 현재 Client-side Image Map이 아닌 일반적인 이미지 맵을 사용하면 로봇이 그 정보를 읽고 판단할 능력은 없다. 세 번째로 로봇은 BBS 또는 ID가 필요시 웹서버에 접근시 사용자 ID와 패스워드를 필요로 하는 서버는 접속하지 못한다. 물론 특정 서버만 접속하는 로봇이라면 가능하지만 그 많은 웹서버에 하나씩 필요한 ID를 다 만들어 놓는 것은 불가능하다

3.4 정보검색 에이전트

최근 들어 전세계적으로 인터넷 사용자수는 불과 몇 개월 사이에 배로 증가하고 있다. 이와 같은 인터넷의 폭발적인 증가는 너무나도 많은 정보의 생산을 초래하게 되었다. 현재 인터넷을 통하여 전 세계적으로 산재한 정보를 색인, 데이터베이스화 한 정보검색 엔진들의 수는 점차로 증가하고

있는 추세이다. 정보검색 에이전트는 사용자가 검색엔진들에 일일이 접속할 필요 없이 검색엔진들을 통합하여 보다 쉽게 원하는 정보를 찾을 수 있도록 사용자에게 일관된 질의 인터페이스를 제공하고, 질의에 대하여 해당자료라고 판단되는 검색 결과를 통일된 형태로 제공함으로써 기존의 정보검색 방법의 단점을 보완한다. 이러한 서치엔진을 메타 서치엔진, 통합형 검색엔진, 멀티스레드(multithread) 검색기라고 알려진 정보검색 에이전트라고 부른다. 한편으로는 이러한 개념은 로봇 에이전트와 별로 다를 것이 없는 듯한 느낌을 주지만, 정보검색 에이전트는 자체 데이터베이스를 가질 필요가 없으므로 적은 시스템 자원으로 서비스를 개설할 수 있다. 뿐만 아니라 한번의 검색으로 여러 검색엔진을 이용하는 효과를 얻으므로 적은 시간으로 포괄적인 정보사냥을 할 수 있다는 장점을 가지고 있다. 정보검색 에이전트의 기본적인 구조는 사용자를 위한 인터페이스 에이전트와 네트워크 상에 존재하는 정보검색 엔진들과의 연계를 담당하는 담당 에이전트들로 이루어져 있다. 사용자가 인터페이스를 통하여 질의를 보내게 되면 인터페이스 에이전트는 각각의 담당 에이전트들에게 질의를 의뢰하는 형식으로 작업을 수행하게 된다. 담당 에이전트가 많이 생기다 보면 에이전트를 수행하고 있는 컴퓨터에는 프로세스 포크(fork)와 메모리 사용량의 증가에 따른 부하가 발생하기 마련이다. 따라서 대부분의 정보검색 에이전트들은 멀티스레드 프로그래밍 기법을 이용하여 담당 에이전트를 생성, 서버의 부하를 줄이도록 하고 있다. 이와 동시에 멀티스레드 기법은 적은 자원으로 여러 담당 에이전트들을 병렬적으로 처리할 수 있는 틀을 마련해준다. 인터넷에 공개된 정보검색 에이전트 중 가장 인기를 끌고 있는 것은 콜로라도 주립대학의 Daniel

Dreilinger가 개발한 Savvy Search로, 현재 알타 비스타, Lycos, Yahoo 등 14개의 검색엔진에 질의를 의뢰할 수 있도록 구성해 놓았다. 이와 반면 워싱턴 주립대학의 MetaCrawler는 6개의 검색엔진에 질의를 보낼 수 있는데, 검색 결과 중 해당자료가 실제로 그 주소에 존재하는지 미리 검사해주는 옵션이 있다. 우리나라에는 미스다찾니가 대표적인 예가 된다.

그러나 현재의 정보검색 에이전트들은 서버 쪽에서 모든 검색작업을 처리하도록 프로그램 되어 있어, 멀티스레드 기법을 쓴다고 하여도 사용자가 많은 시간대에는 여전히 과부하가 생길 수 있다는 단점을 지니고 있다. 또한 각 검색엔진으로부터 얻어진 검색결과는 질의를 의뢰한 서버 쪽의 담당 에이전트에게 송신되고, 이는 다시 인터페이스 에이전트를 통해 사용자에게 송신됨으로, 막대한 네트워크의 낭비를 초래하게 된다[8].

3.5 협력 에이전트

협력에이전트(collaborative agent)[2]는 에이전트의 특성 중 협동성(cooperation) 및 사회성(sociality)을 강조한 에이전트 시스템으로 네트워크 내에 있는 여러 에이전트들이 상호간의 협력을 통하여 주어진 임무를 수행하는 특성을 가진다. 협력 에이전트들을, 마치 하나의 유기체 공동체와 같이, 에이전트를 하나의 독립적인 소프트웨어 프로그램이 아닌 에이전트 유기체 사회의 구성원으로 인식하는 것이다. 협동성과 사회성을 통하여 에이전트는 독립적으로 임무를 수행할 때 보다 더 많은 다양한 업무를 효율적으로 처리할 수 있게 되며 지능(intelligence) 및 학습능력(learning)은 에이전트가 협력에이전트의 일원이 될 경우 사용자의 요구패턴이나 자주 사용하는 문자 및 특수기호 등을 기억하고 있다가 사용자의 요구사

항을 지능적으로 분석한 후 처리결과를 사용자가 선호하는 출력패턴으로 제공하는 능력을 제공한다. 협력에이전트(collaborative agent)[2]는 협조에이전트(co-operation agent)라고도 불리며 이에 대한 연구는 다중 에이전트 기반구조(MAF : Multi-Agent Framework)에 대한 연구와 에이전트간 통신언어(ACL : Agent Communication Language)에 대한 연구로 구성된다. MAF는 협력 에이전트를 구성하는 다중 에이전트들 사이의 구성과 이들 에이전트 사이의 메시지 송수신 체계에 대하여 정의하는 부분이다. MAF는 다시 조정 에이전트(coordinating agent, broker, facilitator)를 통하여 메시지를 주고받는 중앙 집중식 구조(centralized architecture)와 조정 에이전트 없이 각 에이전트들이 다른 에이전트와 서로간의 메시지를 주고받는 분산식 구조(distributed architecture)로 구분할 수 있다. 중앙 집중식 구조의 예로는 FIPA(The Foundation for Intelligent Physical Agent)에서 제안한 에이전트 플랫폼 [12]과 한국전자통신연구원의 EMAF(Extensible Multi Agent Framework)[13] 등이 있다. 분산식 구조의 예로는 Queen Mary and Westfield College of London University의 Archon[14] 등이 있다.

에이전트 통신언어는 협력 에이전트들 간의 프로토콜로서 상호간에 정보를 주고받기 위하여 사용하는 메시지에 대한 신택스(syntax) 및 시멘틱스(semantic)를 정의한다. 이에 대한 연구는 미국방성의 KSE(Knowledge Sharing Effort)가 주축이 되어 진행한 KQML(Knowledge Query and Manipulation Language)이 가장 널리 알려져 있으며[5], FIPA의 ACL(Agent Communication Language)[15] 등이 있다. 그림 2는 중앙 집중식 구조(centralized architecture)를 갖는 에이전트 플랫폼을 보여주고 있다.

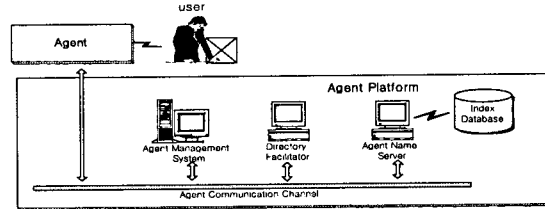


그림 2. 중앙 집중식 구조(centralized architecture)를 갖는 에이전트 플랫폼

3.6 이동 에이전트

이동 에이전트(mobile agent)[2]는 에이전트의 특성 중 이동성(mobility)이 강조된 에이전트로서 사용자의 지시사항을 수행하기 위하여 자신의 판단에 의하여 네트워크 상의 서버들로 이리저리 이동하여 작업을 수행하는 에이전트로서 네트워크 에이전트(network agent), 순회 에이전트(itinerant agent)라 불리운다. 그림 3에서 보는 바와 같이 기존의 RPC (Remote Procedure Call)를 기반으로 하는 클라이언트/서버 모델에서는 클라이언트가 임의의 기능을 수행하기 위하여 수행요구 메시지를 필요에 따라 수행에 필요한 데이터와 함께 서버로 넘겨주면 원격서버 호스트는 이를 받아 처리한 후 그 결과만을 넘겨주게 된다. 하지

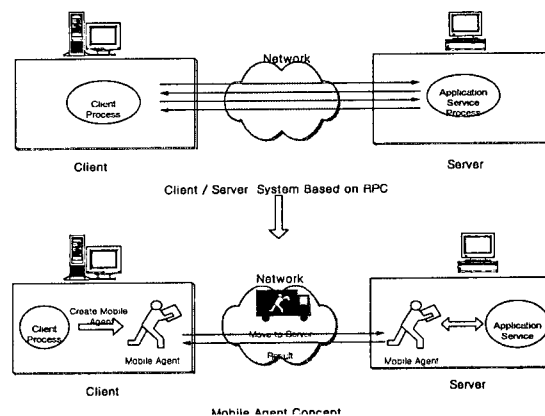


그림 3. 이동 에이전트 개념

만 이동 에이전트 환경에서는 클라이언트가 만든 에이전트를 원격 서버 호스트로 보내어 필요로 하는 기능을 서버의 도움을 받아 업무를 수행하거나 단독으로 기능을 수행하게 된다. 서버로 이주해간 에이전트는 적절한 시기에 기능을 수행할 수 있고 필요에 따라서는 또 다른 서버 플랫폼으로 이동하여 필요한 기능을 수행할 수 있다[4]. 이동 에이전트는 여러 관점에서 모델을 생성할 수 있는 데 생명주기모델, 보호모델, 대화모델, 항해모델 등으로 구분할 수 있다. 이동 에이전트가 다른 서버 플랫폼으로 이동하여 동작하기 위하여는 플랫폼, 즉 다른 서버플랫폼의 운영체제나 하드웨어, 통신프로토콜 등 이질적인 시스템과의 의사소통 수단이 필요하다. 이를 위하여 개발된 이동 에이전트 구현언어로는 제네바 대학의 MO, 디지털사의 Obliq, 코넬대학의 Telescript, MIT의 SodaBotL 등이 있다. 이동에이전트는 에이전트 분야 중 현재 가장 상업화가 많이 진행된 분야로서 General Magic, Ftp Software 등에서는 이미 시제품이 출시된 상태이다.

그럼에도 불구하고 이동 에이전트는 여러 문제점을 가지고 있다. 가장 커다란 문제점은 에이전트 소프트웨어가 이동하면서 발생하는 보안문제이다. 이동 에이전트가 활동하는 네트워크내의 서버 시스템은 이동 에이전트가 자신의 시스템으로 들어와 시스템에 어떤 영향을 줄 지에 대해 파악할 수 있어야 한다. 이동 에이전트 측면에서도 역시 서버 시스템 내에서 또는 네트워크를 이동하면서 원하지 않는 공격을 받거나 손상되어 사용할 수 없는 경우가 발생하지 않도록 안전성을 보장받아야 한다.

3.7 인터넷 에이전트

인터넷 에이전트(internet agent)[8]는 앞에서

언급한 로봇 에이전트와 유사한 특성을 갖는다. 이는 인터넷이나 웹 상에 분산되어 있는 엄청난게 많은 양의 정보를 사용자가 보다 쉽게 접근할 수 있도록 지원하는 역할을 담당한다. 인터넷 에이전트는 웹 에이전트(web agent), 웹 로봇(web robot), 정보 에이전트(information agent) 등으로 불린다. 인터넷 에이전트는 인터넷내의 정보를 탐색하기 위하여 이동 에이전트와 같이 인터넷상을 항해하기도 하지만 반드시 이동성을 요구하는 것은 아니다. 이동성을 가지지 않은 인터넷 에이전트의 예로는 Washington 대학의 Internet Softbot[16]은 ftp, telnet, mail,archie, netfind, compress 등과 같은 명령어나 응용프로그램들을 활용하여 인터넷상의 정보검색을 수행한다. 또한 보다 편리한 인터페이스를 제공하기 위하여 메뉴와 자연어 인터페이스 환경을 제공하고 명확한 명령어 입력을 위하여 모호성 제거 알고리즘을 채택한다. 인터넷 에이전트의 종류 중 사용자가 자주 방문하는 관심 있는 사이트의 정보가 갱신되었는지 여부를 주기적으로 검사하여 변경여부를 사용자에게 알려주는 기능을 갖는 웹 모니터링 에이전트가 있다.

3.8 지능형 에이전트

지능형 에이전트(intelligent agent)[8]는 학습(learning), 추론(reasoning), 계획(planning) 등의 능력을 가진 에이전트를 의미한다. 지능형 에이전트는 일반적인 개념의 에이전트를 의미하며 이들이 갖는 능력에 따라 학습 에이전트, 추론 에이전트, 계획 에이전트 등으로 분류된다. 지능형 에이전트의 예로는 COACH(COGnitive Adaptive Computer Help) 시스템[17]이 있다. 이 시스템은 사용자가 LISP 프로그래밍을 하는 과정을 관찰하여 AUM(Adaptive User Model)을 만들어 작업에 대한 지식과 적응 사용자 모델에 따라 사용자가

향후에 진행할 일을 추론하여 사용자의 작업분야 및 능력에 맞추어 도움말을 제시해 주는 역할을 수행한다. COACH 시스템은 학습기능과 추론기능을 가지는 지능형 에이전트로서 작업환경을 돕는다는 측면에서 사용자 인터페이스 에이전트라고 할 수 있다. 이외에 Syskill & Webert[18] 등이 있다. 이 시스템은 사용자의 작업성격 및 작업취향, 관심분야에 해당하는 웹 페이지 등을 구분하여 학습해내는 학습 에이전트이다.

4. 에이전트관련 보안

에이전트는 자율성에 따라 사용자나 다른 프로그램의 직접적인 지시나 간섭없이 스스로 판단하여 작업을 수행하고 협력성의 특징에 따라 다른 에이전트와의 협력을 통하여 작업을 수행할 수 있는 능력 및 데이터와 그 처리 순서를 하나로 정리해 컴퓨터 통신망을 통해 상대방에게 전송하는데 필요한 기능 등을 수행한다. 이와 같은 특성으로 인하여 에이전트 환경은 모든 에이전트 및 서버 플랫폼에 개방되어 있는 하부구조를 갖기 때문에 에이전트는 컴퓨터 바이러스 및 다른 프로그램의 고의적 훼손 등 공격당하기가 쉽다. 반대로 에이전트에게 정보를 제공해주는 네트워크상의 컴퓨터 서버의 입장에서는 악의성을 지닌 에이전트로부터 자신의 플랫폼의 자원이나 자료를 보호해야 한다. 에이전트 플랫폼 및 에이전트에게 정보를 제공하는 서버시스템에 대한 대표적인 위협요소는 정당한 권한이 없는 에이전트 초기생성자(initiator)에 의해 생성된 에이전트의 고의적인 불법행위를 생각할 수 있다. 에이전트가 이주(migration)해간 서버 플랫폼의 자료, 예를 들면 신용카드 정보나 개인신상정보 등을 서버의 허락을 받지 않은 상태에서 자료를 몰래 유출 또는 변조, 파괴할 수 있으며 서버 플랫폼의 컴퓨팅 자

원을 불법으로 사용하는 경우를 들 수 있다. 또한 정당한 권한이 있는 초기생성자인 경우에도 악의성을 지닌 에이전트를 생성하여 고의적으로 경쟁사의 자료를 유출하거나 서버 시스템의 하드웨어 플랫폼을 다운시키고 데이터베이스 시스템을 파괴하는 등의 행위를 할 수도 있다.

이와같은 에이전트의 잘못된 행위를 수행하기 위한 공격기법들은 기존의 시스템 소프트웨어 개발환경에서 사용되던 방법들을 그대로 적용할 수 있다. 도청(eavesdropping), 속임수(spoofing), 반복수행(message replay), 바이러스(virus), 스팸(spam) 등의 기법들이 이용된다.

이와 같은 문제점들을 해결하기 위하여 에이전트와 서버 플랫폼 관점의 2가지 보호대책을 고려할 수 있다[4]. 먼저 에이전트에 의한 서버 플랫폼의 위협을 해결하기 위하여는 에이전트가 서버 플랫폼내의 자원들에 대한 접근을 제한하는 방법이 있다. 이를 위하여 서버 플랫폼의 자원이용 범위를 정하여 놓고 서버 플랫폼에 접근하여 자원의 할당을 요구하는 에이전트에게 이를 배분하는 방안이 있다. 또한 플랫폼 서버 내에 안전지역(safe box)을 설정하여 에이전트를 해당 지역 내에서만 동작할 수 있도록 하며 서버 플랫폼 내에 특정한 자원(예: 데이터베이스 관리시스템)에 대한 접근은 시스템관리자 등의 허용하에서만 접근 가능하도록 하는 방안이다. 허용은 서버 플랫폼과 에이전트 사이의 상호인증(mutual authentication)을 통하여 얻을 수 있는데 이를 위하여 도착한 에이전트와 에이전트를 최초로 생성시킨 초기생성자를 확인하여 자원에 대한 접근 허용여부를 인증하거나 에이전트 플랫폼간의 상호인증 등이 있을 수 있다. 또한 감사(auditing) 기능 강화, 접근제어 정책 적용 등의 방법도 서버 플랫폼을 보호하는 좋은 대책이 될 수 있다.

서버 플랫폼이 악의를 갖고 자신에 들어온 이동 에이전트를 고의적으로 손상시키려는 경우 에이전트의 특성상 이를 보호하는 데는 한계가 있다. 에이전트를 네트워크내의 다른 서버 플랫폼으로 보내어 기능을 수행시키기 위해서는 원칙적으로 에이전트 자신의 프로그램 소스코드나 데이터를 공개해야하기 때문에 이동 에이전트를 악의적인 서버플랫폼의 공격으로부터 보호하는 것은 사실상 불가능하다. 다만 부분적인 보호나 사후에 에이전트의 손상여부를 감시하기 위하여 소극적인 방법으로서 에이전트에 체크섬(checksum)을 추가하거나 중요 데이터의 경우 암호화하거나 재수행 방지기법 또는 복사방지기법들을 적용할 수 있다.

5. 결 론

오늘날 컴퓨터 분야를 포함한 과학분야, 경제분야, 나아가 사회과학분야까지 인터넷은 엄청난 영향력을 발휘하고 있고 인터넷 사용자의 수는 가히 폭발적이라 할만큼 증가하고 있다. 정치분야에서조차 인터넷을 홍보수단 및 유권자들의 의견수렴의 장으로 사용하고 있을 뿐만 아니라 경제분야에서는 전자 상거래(Electronic Commerce)가 다가오는 세기의 주요산업으로 자리잡고 있다. 전자 상거래의 경우 관련 분야가 너무 방대하여 마케팅, 광고, 금융, 인터넷뱅킹, 데이터베이스, 데이터웨어하우스 및 마이닝, 데이터 레지스트리 등 거의 모든 경제 및 컴퓨터 관련 분야를 통괄하고 있다. 이와같이 인터넷이 매우 광범위하고 빅뱅과 같은 속도로 발전함에 따라 인터넷을 통하여 발생하여 통용되는 데이터는 매초 매시간 엄청난 양으로 넘쳐나고 있다. 따라서 이와같은 데이터의 홍수 속에서 옥석을 가리어 우리에게 진정으로 필요

한 정보를 찾아내기 위하여는 더 이상 인터넷 웹서핑을 통한 직접적인 인간의 노력은 한계에 이를 수 밖에 없다. 이와 같은 문제를 해결하기 위하여 "대리인"의 뜻을 가진 소프트웨어 에이전트가 90년대 들어 새로운 분야로서 각광을 받고 있다. 에이전트는 이용자가 자세한 지시사항을 일일이 제시하지 않고 작업에 필요한 일부지식만을 제공해주면 그것만으로도 목적에 필요한 작업을 스스로 판단해 처리하는 기능을 갖게 하고 있으며 에이전트 자체가 가지는 여러 가지 특징들, 자율성, 이동성, 사회성, 지능 등은 마치 에이전트가 하나의 살아있는 유기체와 같은 속성을 가지고 있는 것과 같은 착각에 빠지게 한다.

이와같은 특징 및 역할에 따라 본 고에서는 에이전트에 대한 분류를 지원기능, 기능원리 등에 따라 전자비서 에이전트, 상업 에이전트, 로봇 에이전트, 정보검색 에이전트, 협력 에이전트, 이동 에이전트, 인터넷 에이전트, 지능형 에이전트 등으로 구분하였다. 이외에도 반응성 에이전트, 이형 에이전트, 혼합형 에이전트 등에 대하여 언급하였다. 에이전트 환경은 기본적으로 서로 다른 형태의 시스템간의 통신과 대화를 위하여 인터넷을 통한 상호운영(interoperability)과 보호(security) 문제가 대두되었다. 인터넷 상에 분산된 응용프로그램들 간의 상호운영성을 제공하기 위하여 DCE, CORBA, DCOM 등의 미들웨어(middleware) 시스템이 출현하였다. 인터넷, 웹, CORBA, DCE 간의 상호운영 프로토콜은 데이터의 포매팅 규칙과 시맨틱을 지원하는 메시지 타입을 지원하고 있는데, CORBA의 GIOP 메시지는 TCP/IP, SNA 프로토콜 등 전송 프로토콜 상에서 보내어지며 인터넷 표준인 TCP/IP 상에서 GIOP 규격이 접속되도록 정의한 것이 IIOP이다.

에이전트 환경에서 특히 고려해야할 사항은 이

형질성(heterogeneous)의 에이전트 및 네트워크 서버 플랫폼이다. 다행히도 최근들어 이와같은 문제점을 해결할 수 있는 JAVA와 같은 프로그래밍 언어 및 CORBA와 같은 상호운용지원 미들웨어들이 출현하였다.

에이전트와 관련하여 이형질 환경으로 파견된 에이전트 자신의 보호문제는 아직도 많은 문제점으로 남아있다. 에이전트 환경은 모든 에이전트 및 서버 플랫폼에 개방되어 있는 하부구조를 갖기 때문에 에이전트는 컴퓨터 바이러스 및 다른 프로그램의 고의적 훼손 등 공격당하기 쉽다. 이와같은 여러 문제들은 결국 에이전트 사이의 통신언어 및 플랫폼간의 보안성과 관련이 된다. 나아가 네트워크의 부하를 줄이고 전체적인 네트워크 시스템의 효율성을 향상시키기 위한 노력도 게을리해서는 안되는 향후 연구분야이다.

컴퓨터 분야에서 에이전트의 개념이 본격적으로 도입/구현된 것은 최근 10여년 사이의 일이다. 따라서 에이전트 분야는 위에서 언급하였던 여러 문제점들과 표준화에 관련하여 많은 문제점을 안고 있다. 네트워크이 주요 사회 인프라스트럭처가 되는 미래에는 소프트웨어 에이전트가 현재의 소프트웨어의 흐름을 이어나갈 것은 자명한 사실이다.

참 고 문 헌

- [1] 日經産業新聞, 1995년 10월 30일
- [2] 장명욱, 이광로, 민병의, "에이전트 기술", 전자통신동향분석, 제12권 제6호, pp.58-69, 1997
- [3] 최중민, "에이전트 개요와 연구방향", 정보과학회지, 제15권, 제3호, pp.415-419, 1977
- [4] 송영기, 인소란, 김명준, "소프트웨어 에이전트의 보호기술", 전자통신동향분석, 제12권 제6호, pp.70-81, 1997
- [5] T. Finin and R. Fritzson, "KQML - A Language and Protocol for Knowledge and Information Exchange", Proceedings of the 13th International Distributed Artificial Intelligence Workshop, pp.127-136, 1994
- [6] Shaw Green et al, "Software Agents : A Review", <http://www.cs.tcd.ie/research/groups/aig/iag/pubreview.zip>, 1997
- [7] 이은석, "멀티에이전트 기술의 실제시스템으로의 응용", 정보과학회지, 제15권 제3호, pp.17-28, 1997
- [8] 승현석, "정보검색의 새로운 세계, 에이전트를 잡아라", <http://www.mochanni.com/~hseung/writing/infoage/9604/article.html>
- [9] 이강찬, 로봇 에이전트와 그 활용 <http://dblab.comeng.chungnam.ac.kr/~dolphin/publication/articles/robot.html>
- [10] A standard for Robot Exclusion, <http://info.webcrawler.com/mak/projects/robots/norobots.html>
- [11] FIPA Home Page, <http://drogo.cselt.stet.it/fipa>
- [12] Paul Obrien, "Agent Management", *FIPA '97 Draft Specification : Part I*, 1997
- [13] 백순철, 최중민, 장명욱, 박상규, 임영환, "이형분산환경에서 에이전트들 간의 이형성을 극복하기 위한 멀티에이전트 기반구조", 정보과학회논문지(C), 제2권, 제1호, pp.24-37, 1996
- [14] T. Witting, N. R. Jennings and E. H. Mandami, "ARCHON - A Framework for Intelligent Cooperation", IEE-BCS Journal of Intelligent Systems Engineering - Special Issue on Real-Time Intelligent Systems in ESPRIT, pp.168-179, Vol.3, No3, 1994
- [15] Ian Dickinson, "Agent Communication Language", *FIPA'97 Draft Specification : Part 2*, 1997
- [16] Oren Etzioni and Daniel Weld, "A Softbot-based Interface to the Internet", *Communications of ACM*, vol. 37, no.7, pp.72-76, 1994
- [17] Ted Selker, "COACH : A Teaching Agent that Learns", *Communications of ACM*, vol. 37, no.7, pp.92-99, 1994
- [18] 백혜정, 김형훈, 이강건, 한경식, 이수원, 박영택,

“적응형 에이전트”, 정보과학회지, 제15권, 제3호, pp.29-38, 1997



박 수 현

- 1988년 2월 고려대학교 이과대학 컴퓨터학과 이학사
- 1990년 2월 고려대학교 대학원 전산학 이학석사
- 1998년 2월 고려대학교 대학원 컴퓨터학과 이학박사
- 1990년~1999년 LG정보통신(주) 중앙연구소 선임연구원
- 1999년~현재 동의대학교 공과대학 컴퓨터공학과 교수
- 관심분야 : 네트워크 관리 시스템, 컴포넌트기반개발, TINA



김 태 석

- 1981년 2월 경북대학교 전자공학과 (공학사)
- 1991년 3월 일본 KEIO대학 이공학부 계산기과학전공 (공학석사)
- 1994년 3월 일본 KEIO대학 이공학부 계산기과학전공 (공학박사)
- 1993년 3월~1994년 2월 일본 KEIO대학 이공학부 객원연구원
- 현재 : 동의대학교 컴퓨터공학과 교수, 전산정보원장
- 관심분야 : 기계번역, 정보시스템, 인터넷시스템관리
- 저서 : 인터넷비즈니스, 제일 쉬운 인터넷주세요 등 다수