it527 Distributed System
Survey Report :
Distributed Intelligent Agents

담당교수 : 홍 원 기 교수님

작성자 : 9651M19 이 수 철

1996년 5월 4일

요 약

최근 Software계에 새로운 문제 해결 방안으로 급성장하고 있는 Distributed Intelligent Agents에 관련된 기술(Intelligent Agents의 속성, 동작 모델 밑 개발 방법, Distributed Intelligent Agents의 구조 등) 및 연구현황을 살펴 봄으로, 기술을 이해하며 이에 대한 현장 적용가능성을 타진해 보고자 함.

목 차

제 1 절 Introduction

사무를 보고 있는 중 사무비서 (Personal Computer내 Software Agent)로부터 Message를 받았는데, 그 내용인즉 지금 현재 매우 중요한 다섯개의 Mail이 도착해 있으며, 내일 아침 9시에 서울에서 고객과 회의가 있다는 것을 상기시켜 주는 것이였다. 이에 따라 사무비서를 통하여 총무처에 출장품의서를 fax로 송부하고, 내일 고객과의 회의자료를 만들도록 작업을 시켰다. 여행 Agent로 하여끔 오늘 저녁 8시 반 서울행 비행기와 호텔 및 렌트카를 예약토록 하고 이를 확인했다.
이러한 시나리오는 미래의 공상이야기 일까 ? 그렇지 않다.
실제 이같은 일들을 가능하게 하기 위한 시도가 다방면으로 진행되고 있다. 최근, 사용자들을 가상세계속으로 데려다 줄 Intelligent Agents(Software Intelligent)를 개발하기 위해 거의 모든 기술 역량이 동원되고 있다. 이처럼, Agent-oriented computing 기술은 "the next siginificant breakthrough in software development" (Sargent, 1992) 또는 "the new revolution in software" (Ovum, 1994) 라고 외쳐지며, 광범위한 분야에 걸쳐 새로운 문제 해결 방안으로 대두되고 있다. 또한 영국의 어떤 시장 분석가는 전세계에 걸쳐 Agent Software 시장은 1994년 년간 370만불에서 2000년에 있어서는 26,000만불로 급신장 할 것으로 예측하고 있다. 이처럼 Intelligent Agents 시스템이 Intelligent Interfaces, Adaptive Interfaces, knowbots, knobots, Softbots, userbots, taskbots, Personal agents, network agents등 수 많은 신조어를 창출해 가며 급증하고 있는 반면, 분산환경에서 이기종 Agents간 상호 운영할 수 있는 참조모델이 아직 존재하지 않을 뿐 더러, Agent Software에 대한 설계나 구현을 위한 표준도 아직 정해지지 않는 등 많은 해결점이 남아 있다.
여기서는 지금까지 발표된 자료들을 근간으로 Distributed Intelligent Agent와 관련된 기술 및 그 의미를 살펴 보고자 한다.

제 2 절 Intelligent Agents 기술

1 개 요

최근에 있어 Intelligent Agent란 용어는 매우 일반화 되어 있으며, Agent에 대한 개념 역시 새로운 것이 아니다. 수십년에 걸쳐 Mavin Minskey, Oliver selfridge, Alan Kay, Nicholas Negroponte, Rodney Brooks, Pattie Maes와 같은 연구자들이 여기에 대해 연구해 왔으며 Agent에 대한 동작형태 몇가지를 실험적으로 구현해 왔다.
Agent 연구에 대한 기본 사상은 "모든 형태의 최종 사용자들에게 고용되어 도와 줄 수 있는 소프트웨어 시스템 개발"에 두고 있다 할 수 있겠다. 이처럼 Agent란 용어가 " 사람이나 어떤 것을 대리해 사전에 주어진 요구나 행동을 처리해 주는 자"와 같은 의미를 갖고 있는 바와 같이, Intelligent Agent는 " Computer에 의해 창조되는 인물(Software)로서, 가상환경 속에서 사용자의 일을 대행해 주는 자"라고 정의(By 브렌다 로렐)하고 있다.
이러한 관점에서 볼때 Intelligent Agent란 가상현실속에서 사용자의 일을 대신 처리해 주기 위해 개발된 특수한 형태의 Software 툴로 볼 수 있다. 즉, Intelligent Agent는 가상 환경속에 살고 있는 존재라고 할 수 있으며, Agent는 가상 현실 시스템의 사용자를 위해 작업을 수행한다. 재미있는 게임의 상대방이 될 수도 있으며, 항공편과 렌트카를 예약하는 여행 대리인이 될 수 있는

등, 많은 중요한 일을 대신 해 줄 수 있다.
Agent는 가상현실속의 인물을 Graphic으로 재현한다든지 하며, 프로그램의 기능을 나타내는 오브젝트를 처리하기 위해 인공지능(A.I) 프로그램과 함께 적용되는 것이 일반적이다. 예를 들면 지능형 Database 검색 Agent는 셜록 홈즈의 모습으로, 과학정보에 대한 자문 Intelligent Agent는 아인슈타인으로 나타내는 식이다.
과거 A.I 기술이 논문에 거치며, 연구의 실험 대상이 되거나, 전문가 시스템과 같이 특정 영역의 사용자에게만 적용되었던 것 과는 달리 Intelligent Agent는 모든 사용자를 대상으로 그들 손에 결과를 직접 쥘 수 있도록 하고 있다.
특히 오늘날과 같이 Internet과 W.W.W (World Wide Web)가 급증함으로 인해 발생된 정보의 홍수, 정보의 과부하를 줄여주기 위한 특정 정보 검색 Intelligent Agents System이 주목받고 있다.

2 Agent의 분류

Agent를 분류하는데 전문가에 따라 여러가지 방법이 있으나, 여기서는 Jennings와 Wooldridge가 분류한 것을 소개하고자 한다. 이는 Agent의 Intelligence 차이에 따라 분류한 것이다.

1. "gopher" Agent
   : 가장 단순한 레벨의 Agent로 미리 명세화된 룰과 가정에 따라 TASK를 곧 바로 실행시킨다. (예: 매주 금요일 아침 9시에 팀 업무 회의가 있다는 것을 상기시켜라.)

2. "Service performing" Agent
   : 조금 복잡성(지능)을 갖고 있는 Agent로, 사용자의 요구에 대해 잘 정의된 고차원 TASK를 실행시킨다. (예: 회의 일정을 정리하거나, 적당한 비행편을 예약시키도록 하는 일)

3. "Predictive/Proactive" Agent
   : 사용자의 요청이 없어도 그 정보나 Service가 적절하다고 여겨질때 그것을 지원해 주는 Agent. (예: Agent가 Internet상의 News 그룹을 모니터링 하고 있다가 사용자에게 유용한 항목이 있을 경우, 토론장으로 사용자를 바로 연결시켜 주는 Agent)

이외에도 Simple Agent, Intelligent Agent 또는 Weak Agent, Strong Agent등 Agent들을 분류하는 데는 여러가지 안이 있다.

3 Agent의 속성

앞서 이야기 한 바와 같이 아직 구체적인 표준이나 모델이 정착되지 않은 관계로 속성을 여러가지로 표현하고 있으나, 여기서는 Jennings가 정의한 내용을 위주로 설명하고자 한다.

1. 자율성 (Autonomy)
   : Agent는 사용자나 다른 Agent의 개입없이 자율적으로 자신의 문제를 해결할 수 있는 능력을 갖고 있어야 한다.

2. 사회성 (Social Ability)
   : Agent들은 적절하게 다른 Intelligent Agent 또는 사용자와 상호 작용하여 문제해결 목적을 달성하거나, 다른 Agent의 요청이 있을때 도와주어야 한다. 이를 달성하기 위해 Social Ability는 다른 Agent들과 통신할 수 있는 최소한의 수단을 갖추어야 한다.

3. 신속한 응답성 (Responsiveness)
   : Agent는 Agent가 처한 환경 (실제 물리적 환경, 사용자, 다른 Agent들의 정보, Internet 상황등) 을 인지하여, 환경변화에 대해 즉각적으로 적절하게 대응해야 한다.

4. 진취성 (Proactiveness)
   : Agent는 단순히 그가 처한 환경에만 대응해서 반응해서는 안되며, 목적지향적인 행동과 다른 선택기회를 제시할 수 있어야 하며, 어떤 것이 적절한지 부가 설명해 줄 수 있어야 한다.

5. 적응력 (Adaptability)
   : 시간 변화에 따른 환경 조건 변화에 Agent는 행동방법을 변경시킬 수 있는 능력을 갖추어야 한다.

6. 이동성 (Mobility)
   : 문제해결을 위해 Agent는 Agent의 물리적 위치를 변경할 수 있는 능력을 갖어야 한다.

7. 정확도, 성실성 (Veracity)
   : Agent는 고의로 거짓정보를 제공하지 않는다는 전제사항이 있어야 한다.

8. 합리성 (Rationality)
   : 문제해결을 위해서 Agent는 정확한 근거하에 동작해야 한다.

이외에도 IBM은 Intelligence, Agency 및 Mobility로 Agent의 속성을 규정하고 이에 따라 Agent를 차별화하고 있다.

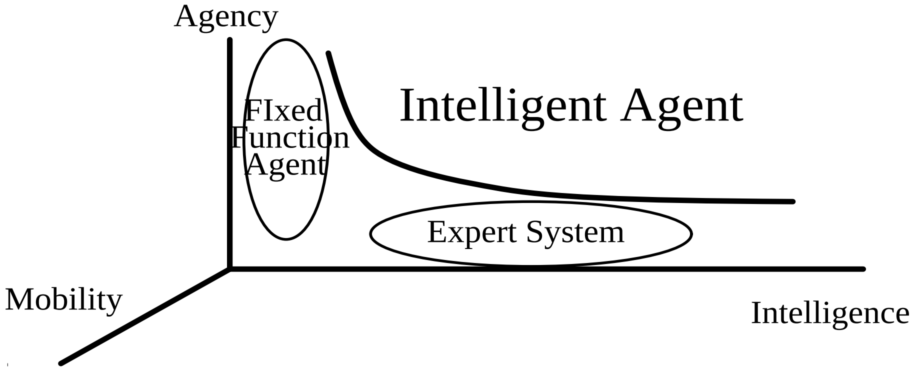

그림 1: Intelligent Agent 영역

4 Intelligent agent의 동작 모델

Intelligent Agent에서 기본 동작 모델을 아래 그림에서 보여주고 있다. 사용자는 Graphic User Interface (GUI)를 통해 Intelligent Agent Software Application을 Access하며, Intelligent Agent Software Application은 Internet과 같은 NETWORK를 경유해 공급자의 Gateway에 도달하게 된다. 최종적으로 Agent는 공급자의 Database나 다른 저장매체에 있는 정보원을 Access하여, 사용자가 요구한 필요한 정보를 추출, 필요로 하는 정보를 완성하여 사용자에게 응답한다.

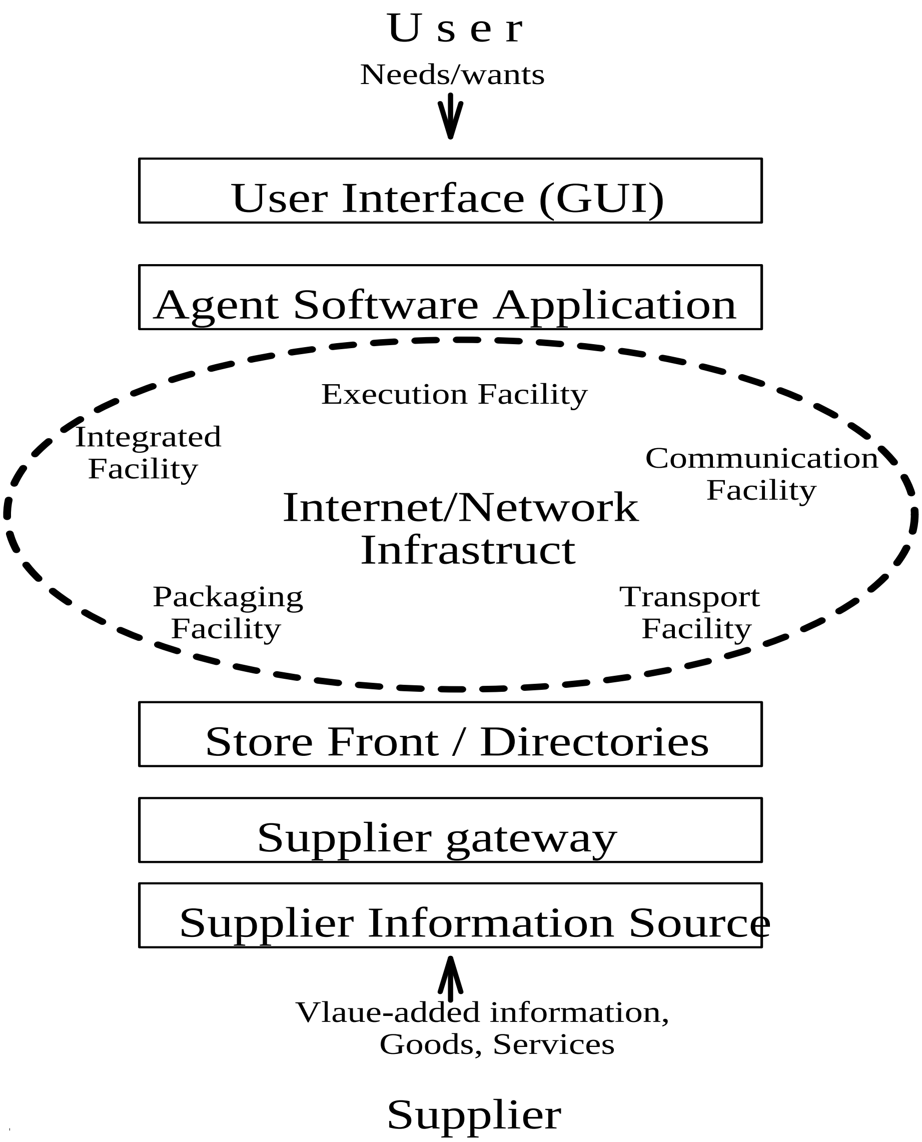

그림 2: Agent 동작 모델

타원으로 표시되는 부분이 Agent 시스템의 하부구조(Infrastructure)를 보여주고 있는데, 이들 구성요소들을 좀더 상세하게 살펴보면 다음과 같다.

4.1 Executing Facility

Execution Facility는 Agent의 Run-Time환경을 의미한다.(일반 Process에 있어서 O.S가 Real-Time환경을 제공해 주듯이) 즉, Agent 환경에서 Agent Program을 실행시켜 주는, Agent의 거주 및 작업 수행환경이 되는 Hardware, Software를 말한다.
H/W는 P.C, Workstation, Main Frame computer등 어느 것이라도 가능하다. 현재 또는 향후의 Agent 개발 Language Compiler는 Multiple Hardware Platform으로 개발 될 것이므로 현재 있는 시스템들과 호환성만 확인하면 된다.
Software 요소로는 Agent Software Programming Language를 들 수 있는데, 아직 학문적으로 연구단계에 있어 미비한 점이 많다. 아직 미진한 이유는 Intelligent Agent Application이 기존 응용 program과 많이 틀린점, 기술이 아직 개발중에 있다는 점, 그리고 전문가들이 Intelligent Agent Application 개발용 programming Language에 대한 표준에 합의를 못 하였다는 점을 들 수 있다.
Agent Programming Language는 일반적으로 다음 세가지 유형으로 분류된다.

1. General Purpose Programming and Scripting Language
   -. 범용 programming Language : C, C++, LISP 등 -. 범용 scripting Language : SQL (Structured query Language), Perl (Web-based scripting Language)

2. General Purpose Mobile Code Language
   Mobile Code는 Network를 경유해 그곳에서 실행되어 져야 하는 부가적인 기능을 갖고 있다. Java는 현재 발표되어 있는, 대표적인 Mobile Code Language이다. SUN사의 Java는 C++과 유사하나, 이동성을 갖고 있다 . HotJava는 최근 Netscape Web Browser로 통합되어짐에 따라 World Wide Web계에 표준 Mobile Code로 되어 가고 있다. 이외에도 Safe-Tcl, albeit, Python, scheme48, Guile, Obliq, Logicware, ScriptX, Plantom등이 있다.

3. Intelligent Agent Application용 Mobile Code Language
   앞서 언급된 Mobile Code language가 Mobile Intelligent Agent에 필요로 하는 기능을 제공해 주고는 있지만 Intelligent Agent Application을 위해 개발된 것은 아니다. 좀더 학문적이고 상업적인 관점을 갖고 Mobile Intelligent Agent 개발을 위해 제작된 것이 있는데 다음의 것이 이 부류에 속한다.

   (a) AGENT0 : Stanford 대학의 Yoav Shoham 교수가 그의 Agent Oriented Programming 개념에 근거해 개발하고 있는 것으로 아직 완성되지는 않았다.
   (b) object.REXX : IBM사가 Agent-based computing 기술에 근거해 개발한 제품으로 Agent Programming Language의 표준을 목표로 하고 있다.
   (c) Telescript : General magic사가 개발한 것으로 현재 Agent-based Communication에 있어 표준으로 부상하고 있다. 이것은 Object-oriented 기반, Remote-programming Language로 실행환경 뿐만 아

니라, Communication, security 특성까지 제공해 주며, Agent가 자유로히 여행할 수 있도록 지원해 주어 다른 분산 Network 응용 Program에도 유용하게 활용되고 있다. 지금 현재 Telescript의 구현 예는 Sony Magic Link Patform상에 구현된 PersonaLink Service를 들 수 있다.

당분간 C 또는 C++ code로 개발될 것이며, 이들 대부분은 세션계층에서 기동될 것으로 예상된다.

(Remote Programming (RP))

Intelligent Agent 기술에 있어 Agent와 또는 Agent간 통신 및 상호작용 기능을 제공해 주는 것으로 RPC (Remote procedure Call)을 개선시킨 방식이다. Remote Programming은 Computer와 Computer간 통신에 있어 한 Computer가 상대측 Computer의 Procedure를 Call할 뿐 아니라, 그 Procedure가 실행되도록 지원해 준다. Network를 경유해 전송되어지는 Message 각각에는 수신측 Computer가 실행해야 할 Procedure와 Data인수로 구성되어져 있다.

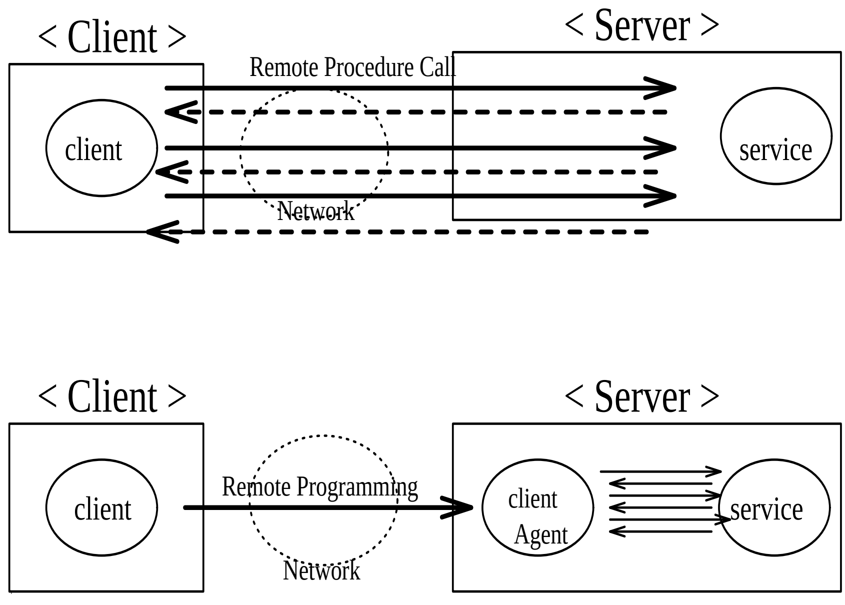

그림 3: Remote Programming

그림에서 살펴 볼 수 있듯이, Server에게 의뢰해야 할 일을 갖고 있는 사용자측 Computer는 Server Site에서 상태정보 (예: "두달")에 기인하여 요구하는 일을 수행할 수 있는 Procedure (예 : "Delete")를 만들 수 있는 Agent를 Server측으로 송신한다. Agent는 Server Site에서 삭제될 File을 선별(두달 이상 된것)하여 삭제작업을 수행한다.
그림에서 볼 수 있듯이 삭제될 File이 얼마가 되던간에 Computer와 Computer간의 통신에는 한개의 Message로 충분하여, Remote Procedure Call에 비해

훨씬 효율적으로 운영할 수 있다.
이러한 통신 방식은 Intelligent Agent의 이동성을 지원해 줄 뿐 아니라, 원격 Intelligent Agent 기술의 기본이 되고 있다.

4.2 Communication Facility

Communication Facility는 Agent간 정보 교환을 지원해 준다. 즉, Agent들간 거래에 있어 교환되어 져야 할 정보를 처리하는 것이다. Object-oriented 시스템에서와 같이 교환되어 지는 정보 Message에는 내용(Content)과 구문(Context)을 포함하고 있다.
Communication Facility는 동기 통신방식과 비동기 통신방식 둘 다를 지원해야 하며, Multiple Agent들과 동시에 통신할 수 있도록 지원해 주어야 한다.
아직까지 정해진 표준 Communication Facility는 나오지 않았다.

4.3 Transport Facility

Communication Facility는 Agent간 통신에 있어 어떠한 것을 사용할 것인가 하는데 관여하는 반면, Transport Facility는 Agent들간 통신에 있어 어떻게 할 것인가 하는 방법을 취급하고 있다. 즉, Agent는 목적하는 바 업무를 완료하기 위해 한 실행환경으로 부터 다른 실행환경으로 이동하게 되는데 이러한 이동 자체를 지원해 준다.
HTTP, SMTP 및 TCP/IP와 같은 Data 전송 표준 Protocol을 사용한다.

4.4 Packaging Facility

Packaging Facility는 Agent를 관련 정보와 같이 "Wrapping" (및 "Unwrapping")하는 표준 방법을 제공한다. 즉, Agent가 한 실행환경에서 다른 실행 환경으로 옮겨 갈때 사정의 준비작업의 일환인 자신의 Packaging에 관한 방법을 말하는 것이다. 내부구조가 별도로 정의되어 있음에도 불구하고 모든 Agent들은 그들의 상태정보(현재 수행하고 있는 TASK에 대한 상태정보, 작업수행정도, 이동경로등), 인증 정보(소유자 ID, 자신 Home Address등), Goal정보, Agent능력(수행기능, 범위등) 및 계획(Agent의 Goal을 얻기 위해 임시로 순서화시킨 잘정의된 일련의 작업 절차) 정보등을 캡슐화시켜야 한다. 그러나 아직 표준화된 packaging Facility는 아직 나오지 않았다.

4.5 Integrated Facility

Agent에 대한 신분 명세 및 확인기능은 Agent Architecture Infrastruct의 본질적인 속성이다. Security는 앞서 언급된 모든 요소에 관련되는 분야이다. Agent Infrastruct는 Agent에 대한 소유자와 출처를 정하는 안전한 방법을 제공해 주어야 한다. 대부분의 Application에 있어서 개별 Agent들은 이를 정보를 코드화한 "Passport"를 갖고 있다. Agent Infrastruct에서 사용되어 지는 Security 방안에는 자원에 대한 보호(Host Computer나 Network를 오용하지 않도록 하는 등), 요청한 서비스가 거절되지 않도록 하는 배려, 정보에 대한 비밀보장 등도 고려되어 져야 한다.
일반적으로 개별 Intelligent Agent Application은 Application에 대한 특정 요구를 처리하기 위해 자신의 보안 프로토콜을 사용한다.

5 Intelligent agent에서 관련 정보 흐름

지금 현재 구현되어 있는 Intelligent Agent들은 비교적 간단하다. 이들중 대표적인 것이 사용자가 필요로 하는 정보를 찾도록 도와 주는 것인데, 대부분 Hypertext Link를 사용하여 TEXT에서 TEXT로 뛰어 넘어 가며, 정보를 수집할 수 있도록 해준다. 다음 그림은 이러한 환경에서 Agent가 동작하는 모습을 보여주고 있다. 아래의 Agnet 응용 프로그램은 DATA를 수집하고 탐색(Search engine)을 위한 Index를 구축하도록 되어 있다. 사용자가 Searching할 때마다, Agent가 생성해 주는 새로운 Searching File을 대할 수 있다. (Searching Engine은 Agent의 한 부분이 아니다.)

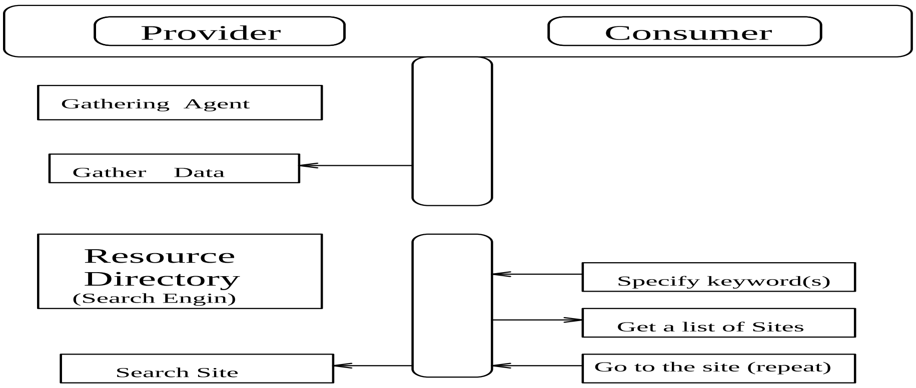

그림 4: Agents in action

다음 지금 현재 구현되어 있는 Agent 응용 프로그램중, Andersen Consulting사의 "BargainFinder"의 동작 모습을 보여 주고 있다. 사용자 CD 제목을 입력하면 최적 가격의 가게를 알려준다.

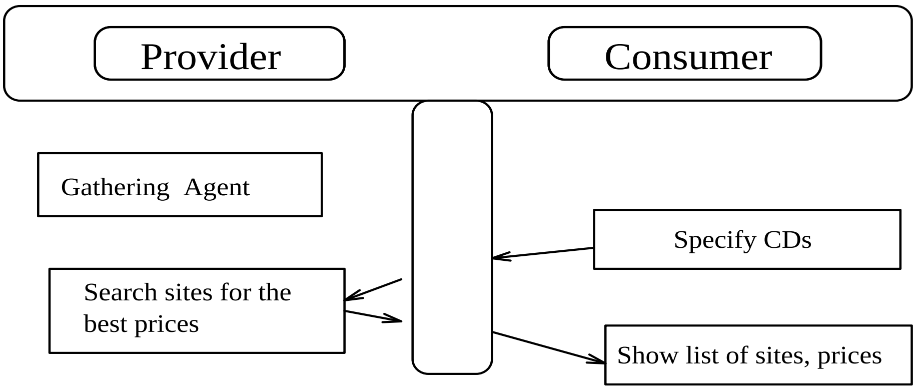

그림 5: Commercial Agents

6 Agent의 개발

앞서 언급된 바와 같이 Intelligent Agent는 인공지능 기술과 Object-oriented 기술 및 Computer Graphic 기술의 결합에 바탕을 두고 있다. 여기서는 Intel-

ligent Agent의 행동양식과 상태를 개발하는 과정을 살펴 보기로 한다. 먼저 Intelligent Agent의 행동양식을 통제하기 위한 가장 일반적인 방식은 종전의 A.I (Artificial Intelligent) 연구가들이 사용해 온 기법에 의존하는 것이다. 예를 들어 전문가시스템 (Expert System), 계획시스템 등의 기법을 들 수 있다. 이들 중 가장 많이 활용되고 있는 기법이 계획시스템이다. 계획시스템은 해법을 찾기 전에 먼저 가능한 대안들을 여러가지로 시도해 보는 비교적 간단한 문제해결 기법인데, 여기에서 계획이란 주어진 문제에 대한 해법을 찾기 위한 일련의 과정을 의미한다. 대부분의 계획시스템은 몇개의 계획안을 먼저 제시한 다음 시행착오를 통해 최소경비로 가장 바람직한 대안을 결정하는 구조로 되어 있다. 계획시스템은 통상 정확한 환경모델을 완전히 Access할 수 없기 때문에 계획시스템은 불확실성과 불완전한 심지어는 일관되지 않은 정보까지도 취급할 수 있어야 한다. 게다가 Intelligent Agent는 대상영역이나 Agent 행동이 예상 하지 못한 결과에 대한 불확실성으로 인해 계획이 실패로 돌아가더라도 상황복구 능력이 있어야 한다는 문제도 있다.
전통적인 인공지능 기법이 나름대로 유연성을 갖고 있다고는 하지만 이러한 접근방식들을 실시간의 제약적인 환경에서 그대로 적용한다는 것은 무리가 있다. 이에 따라 신 A.I라고 불리우는 접근방법이 브룩스나 매스등으로 대표되는 학자들에 의해 대안으로 제시되고 있다. 신 A.I 기법은 인지와 행동간의 보다 긴밀한 연관성의 확보 방안과 분산연산, 환경과의 동적 쌍방향처리, 자원의 제약성과 불완전한 지식들을 취급할 수 있는 적절항 메커니즘의 확보에 촛점을 맞추고 있다. 신 A.I의 경우 System에 대한 지식은 종전과 같이 상징적 형태로 제시되는 것이 아니라 환경 그 자체가 스스로 최상의 모델이 된다. Intelligent Agent 연구중, 가장 흥미로운 분야는 자기학습과 발전능력을 갖춘 Agent 개발에 관한 것이다.
이와 관련하여 많은 연구결과가 제시되었지만, 그중 인공 신경망이 가장 최근 주목 받고 있는 분야이다. 인공 신경망은 상호연결 여부와 강도(가중치)에 의해 행동양식이 결정되는 간단한 처리 객체들로 연결된 시스템이다. 사람의 신경망 조직에서 힌트를 얻은 이 시스템은 Intelligent Agent Software의 구성을 이루는 Object 인식능력, 필적인식 및 기타 다양한 기능들을 개발하는데 적용되고 있다.

7 Agent-based 컴퓨팅과 Client/Server Model과의 차이

Client/Server기술은 영역을 Service 소비자들(Clients)과 제공자들(Servers)로 나누며, 이들간 통신 채널을 통하여 정보를 교환하고 있는 반면, Agent-based 컴퓨팅 환경은 이들의 혼합형태 (Intelligent Agent 의 가장 기본적인 기능은 Client나 Server상에 존재하여 Data를 수집하고, 그것을 해석하며, 사용자가 정의한 동작과정을 구현함으로 그에 대한 응답을 하는 것이다.)를 유지한다. 예를 들어 Client/server Model위의 Database Server는 Remote Client들이 질문(queries)을 통해 Database Server를 access할 수 있다. 이 후 NETWORK를 통해 질문에 대한 결과를 받으며, Client들은 이 결과를 분석해 필요한 정보를 추출해 낸다. 이러한 과정에서 Clients는 보다 정형화된 질문을 보냄으로 search범위를 좁힐 수 있다. 이 보다 효율적인 방법은 Client가 정보추출 Agents를 Database Server에게 보내, 정보추출 작업을 그곳에서 실행토록 하는 것이다. 이런 Agent들은 Client의 지능(Intelligent)을 갖고 Server로 옮겨가기 때문에 시간과 NETWORK의 대역폭을 절약할 수 있다.

8 Object-oriented 개념과의 차이

Agent-oriented 개념은 종종 Object-oriented 개념과 비교된다. Object-oriented programming 환경에 있어 Object들은 실세계의 대상물을 표현하고 있다. (재고목록, 회계거래등) Object들은 문제영역 컴퓨팅 환경으로 Mapping이 가능하며, 상태(State)와 정의된 동작(Behavior)으로 특성화되며, Message를 통하여 상호 작용하고 있다.

Agent-oriented programming은 Object와 Agent 둘 다를 취급한다. 어떤 의미에 있어 Agent는 Object를 포함한다고 할 수 있다. 왜냐하면, Agent역시 상태와 동작으로 특성화 되며, 내부 Data 구조와 동작에 독립된 Message 기반의 Interface를 제공하기 때문이다. 그렇지만 Object 와는 달리 Agent는 Goal을 갖고 있으며, Goal을 획득하기 위해 진취적으로 동작한다.

object 개념에서는 Programmer가 Pocedure를 구성할 때 Message종류에 따라 일반적으로 정해진 동작을 취하도록 Procedure를 작성하기 때문에 간단하다 할 수 있다. 반면, Apent는 Programmer가 예측치 못한 돌발사태에서도, 지난 경험을 토대로 동작토록 되어야 하므로 Procedure작성이 복잡하다 하겠다. (예를 들며, 개 Object는 궁지에 몰리면 짖는다. 개 Agent는 계속해서 짖어대면 먹이를 빨리 준다는 것을 알고 있다.)

Object-oriented programming의 기본 사상이 정보의 은닉(encapsulation), 계승(inheritance)에 있는 반면, Agent-oriented Programming은 목적지향 실행(goal-directed execution)원리에 기본을 둔다고 할 수 있다.

제 3 절 Distributed Intelligent Agent 관련 기술

1 개 요

앞서 살펴 본 바와 같이 Intelligent Agent는 지식과 추론을 이용하여 사용자의 행위적 관점에서 TASK를 수행하는 Software요소이다. Agent에게는 어떠한 목적이 주어지고 이 목적을 수행하기 위해 자기 자신을 동작시키는 것이다. Agent는 사용자로 하여끔 보다 쉽고 효율적으로 Computer를 취급할 수 있도록 해 주지만 한개의 Intelligent Agent가 전체 일을 수행하기에는 역 부족인 경우가 있다. 특히, 실세계에 있는 TASK들을 많은 다른 활동을 필요로 한다. 이에 따라 최근에는 이러한 문제를 해결코자 Multiagent System에 대한 연구가 한창이다.

Multiagent System은 자율성을 가진 몇개의 Agent들이 주어진 공동의 목적을 달성하기 위해 상호동작 또는 협조하도록 되어 있다. Multiagent 시스템이 갖는 큰 잇점은 개별 Agent의 지식과 능력을 자신의 구조나 내부 로직을 변환하지 않고 다른 Agent의 지식과 능력을 공유하는 것 만으로도 확장할 수 있다는 것이다.

이와 같이 Multiagent 시스템들은 일반적으로 어떠한 공동 영역내에 Network로 묶여져, 서로 상호간 협조하면서 사용자가 요구한 작업을 수행토록 되어 있다. 이러한 시스템 구성을 Multiagent 시스템 혹은 Distributed Intelligent System이라 불리운다.

따라서, Distributed Intelligent Agents는 기존 Intelligent Agents에다 network 통신 기능을 부가한 것으로 볼 수 있다. 물론 단순히 통신 기능만 추가된 것으로 분산된 Agent들간 작업이 순순히 진행되는 것은 아니다.

이는 일반 프로세스와 달리, 개별 Agent들이 자율성을 갖고 동작하기 때문에 이들간의 조율이 가장 중요하다고 여겨진다.

물론 하나의 큰 목적이 단순히 작은 목적 단위로 나누어져, 이들을 개별 Agent의 동작결과로 부터 얻어지는 것은 쉬운 것이 아니다. 따라서, Multiagent 시스템에 있어 가장 중요한 요소는 Agent들간의 동작에 있어 공동의 목적을 달성할 수 있는 상호 협조체계, Agent들간의 통신이 되는 것이다.

지금 현재 Distributed Intelligent Agent에 대한 연구가 한창 진행중인 관계로 기술이 아직 완전히 정착되지 않은 것 같다. 여기서는 Distributed Intelligent Agent에 관련된 연구 자료들을 간단하게 정리해봄으로써 현재 Distributed Intelligent Agent 기술이 어디까지 와 있으며, 어떤 것이 있는 지를 개략적으로 살펴보고자 한다.

다음에 보여지는 그림은 Distributed Intelligent Agents 시스템 구성에 대한 이해를 돕기 위해, 최근 ETRI(Electronics and Telecommunications Research Institute)에서 발표한 자료중 발췌한 것이다. 여기서 각 Agent들은 Computer 비서 역할을 수행토록 되어 있다.

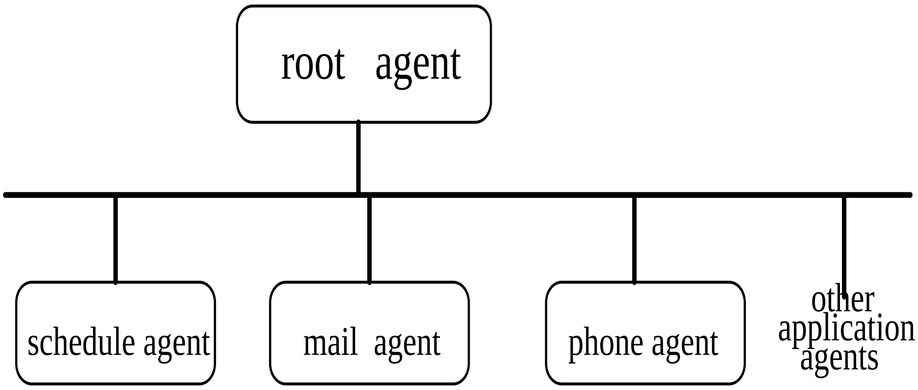

그림 6: ETRI의 개인비서용 Agent 시스템 Architecture

2 Distributed Intelligent Agents의 구조

Distributed Intelligent Agents 시스템이 갖는 구조는 앞서 Intelligent Agent시스템의 동작 모델에서 살펴본 바와 같은 동일한 Infrastructure를 갖고 있다. 즉 다음 그림에서 보여주는 Facility는 Executing Facility, Communication Facility, Transport Facility, Packaging Facility, Integrated Facility 모두를 포함하게 된다.

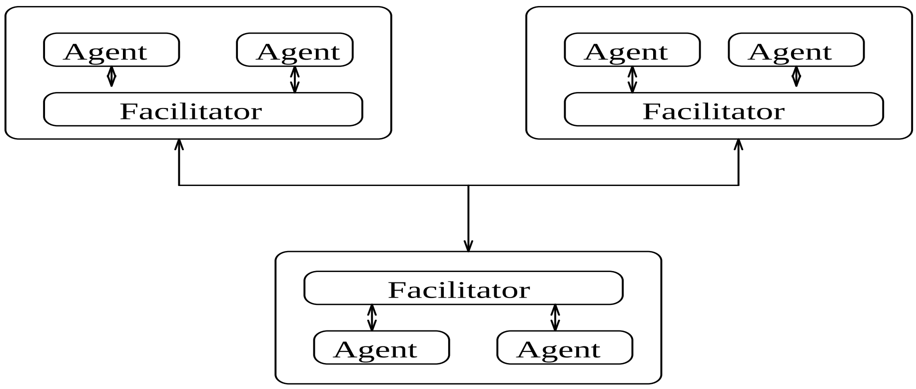

그림 7: Architecture of Multi-Agent system

3 Distributed Intelligent Agent System에 있어 통신

Distributed Agent가 가져야 할 속성중 제일 중요한 속성이 사회성이라 할 수 있다. 즉, 다른 Agent들을 포함하는 환경속에 존재하기 때문에 이들과 어떤 방법을 통해서라도 상호 협조하여 문제해결을 위한 목적을 달성해야 한다. 분산 intelligent Agent System에 있어 Agent는 multi-Agent 환경에 놓여있다. 이들은 초기 A.I분야중 하나인 전문가 시스템(Expert System)과 같이 분리 격리된 intelligent가 아니다.

이러한 관점에서 볼때 Agent에 있어 통신은 매우 중요하다.

Agent들간 협조작업에 있어 일관성을 유지하기란 매우 어려운데 이 문제의 가장 큰 요인으로는 정보의 부족에 기인한다. 즉, 일반적으로 Agent는 그들의 환경에 대해서 완전하고, 정확하며, 현재의 지식을 유지하는 것은 거의 불가능하다. 즉, 어떤 Agent도 다른 모든 Agent가 어떤 일을 하는지 모른다는 것이다. 대신 Agent들은 부분적인 지식을 유지하면서 Communication을 통해 필요로 하는 정보(Plans, Patial Result, Synchronization Information등)를 주고 받으며 상호 협조 작업을 하는 것이다.
Werner는 분산 agent환경에 있어 통신방안을 다음과 같이 다섯 가지로 분류했다.

1. No communication
   : 가장 간단한 방법으로 통신을 하지 않는 것이다. 이방법에 있어서 Agent들은 당연히 각기 다른 Agent들의 계획사항을 통신하지 않고 추론할 수 있거나, 모든 Agent들이 문제해결 목적을 달성하기 위해서는 미리 정해진 안(Pre-established harmony)에 따라 일관성 있게 동작되어 져야 한다.
   이러한 방법은 실행시간중에 통신이 필요 없어 간편하게 보이나, 실제 다른 Agent들의 계획사항을 추론하는데 어려움이 있고 시간도 많이 소요됨으로 극히 한정된 환경에서 사용되어 질수 있다.

2. Signalling
   : Multi-Agent 공동작업시 동기를 Semaphore 같은 Signal을 사용하여 얻는 방법으로, 동기 정보만 제공 되어지므로 매우 잘 다듬어진 Agent들을 요구한다.

3. Message passing
   : Hewitt과 Agha가 개발한 방식으로, 지금 현재 가장 많이 활용되고 있는 방식이다.

4. Plan passing
   : 앞서 언급된 바와 같이 Agent간 협조작업에 있어서는 일관성을 유지하는 것이 매우 중요한데 이를 계획 교환(Plan-exchange)을 통해 해결하도록 하는 방법이다.

5. Speech acts
   : Speech acts의 기본 원리는 동작(action)의 형태로 통신하는 것으로, 철학자와 언어학자들이 사람의 통신 수단(형태)를 이해하고자 하는 시도로부터 연구가 진행되고 있다.

4 Multiagent System의 분류

앞서 논한 바와 같이 Multiagent System은 일정 범위 공동영역내에 몇개의 Intelligent Agent 시스템으로 구성되어 있다. 이들 Agent System들은 상호간 계산자원과 정보를 공유해야 한다.
일반적으로 시스템은 세가지 측면에서 볼 수 있다. 즉, 시스템이 수행하는 기능에 따라 기술할 수 있으며, 시스템의 구성요소에 따라 구조화시킨 관점에서 또한, 시스템 구성요소가 어떻게 시간적으로 동작하는가 하는 행위적 관점에서 시스템을 묘사할 수 있다.
여기서는 기능적인 측면에서, Multiagent 시스템들이 자원을 공유하는 방법에

따라 Multiagent 시스템을 분류해 놓은 것을 살펴보고자 한다.
여기서 기본을 두고 있는 것은 Genesereth와 Nillson이 정의해 놓은 8-tuple형 Intelligent Agent의 기능 모델이다.

(D, S, T, A, perceive, infer, select, act)

- D : 예측 계산용 Database의 집합
- S : 외적 (External world) 상태 정보의 집합
- T : S의 부분집합으로 Agent가 구분 가능한 상태정보의 집합
- A : 동작(Action)들의 집합

Perceive, infer, select 및 act 의 기능은 다음과 같은 법칙에 따라 정의되어진다.

- perceive : S -> T
  (환경을 감지하고 이를 인식하여 agent가 가용할 수 있는 정보로 만드는 기능)
- infer : D * T -> D
  (Robot에게 인지능력을 부여하는 것으로 T에 기초하여 새로운 예측계산용 Database를 재구성 하는 기능)
- select : D * T -> A
  (Select 기능은 주어진 환경에서 Agent가 취할 수 있는 적절한 동작들을 선정할 수 있는 능력)
- act : A * S -> S
  (Act 기능은 Agent가 환경에 영향을 주는 동작들을 수행할 수 있는 능력)

이들 기능요소들을 모델로 재배치해보면 그림 8.과 같으며,

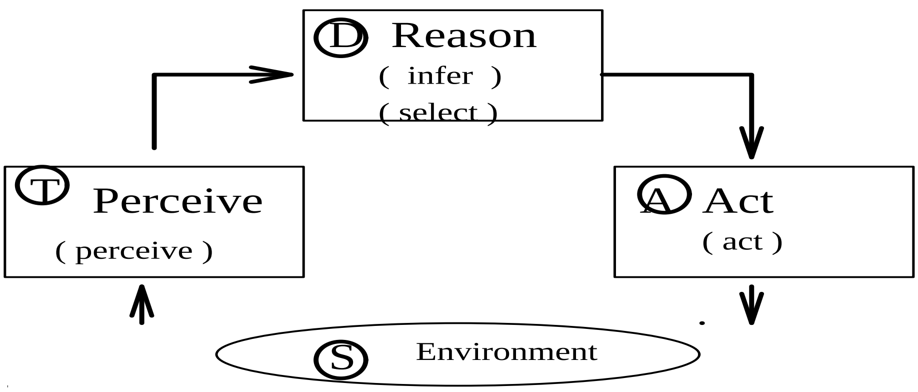

그림 8: Intelligent Agent 시스템의 기능 모델

이를 Graphical 표기법으로 표현하면 그림 9.와 같이 표현할 수 있다.

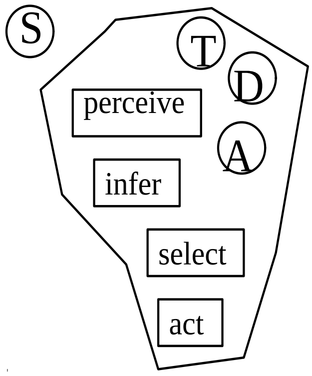

그림 9: Intelligent Agent 시스템의 기능 모델

원은 집합을, 네모박스는 기능을 가르키며 전체 몸체가 Agent를 가르킨다. Agent 테두리선 안에 있는 모든 것은 Internal로 바깥쪽에 있는 것은 External로 보는 것이다. 외적 환경정보를 나타내는 집합군 S가 밖에 있는 것을 알 수 있다. 이는 Multiagent 시스템을 구성하는 공동 영역내의 환경정보를 나타내는 것이다. 이로 부터 우리는 가장 간단한 Multiagent 시스템 구성형태를 알 수 있다. 즉, 외적 환경정보 S만 서로 공유하는 형태이다. 이런 결합 형태를 "Autonomous agents"라고 하며, 다음 그림 TYPE 1.에서 볼 수 있다.
Intelligent Agent의 기본 기능 모델에서 살펴 본바와 같이 Agent는 3개의 집합(T, D, A)과 (S는 이미 Multiagent system들간 공유하는 것으로 정의되었으므로) 4 개의 기능(perceive, infer, select, act) 구성되어 있으므로, Agent들간 결합되어 질 수 있는 경우의 수는 128 가지 (2 ** 7)가 될 수 있다.
여기서 좀 더 분석을 간단히 하기 위해 살펴보면, 예측계산용 Database의 집합을 공유한다면 이에 관련되는 기능도 자동적으로 공유됨을 알 수 있다. 예를 들어 T가 Multiagent 시스템에 있어 공유되어 진다면 perceive 기능은 자동적으로 공유되어 진다.
이를 통해 구분되어 질 공유자원을 T, D 및 A 세 가지로 압축시킬 수 있다. 이들 자원을 공유할 수 있는 총 방법의 수는 8 가지가 된다. 다음 그림은 이들의 형태를 하나 하나 보여 주는 것이다.
이에 따라 Multiagent 시스템들은 다음 8 가지 방법으로 구분된다.

- TYPE 1 : "Autonomous agents"
- TYPE 2 : "perceptually coupled"
- TYPE 3 : "inferentially coupled"
- TYPE 4 : "distributed cognition"
- TYPE 5 : "cognitively coupled"
- TYPE 6 : "distributed inference"
- TYPE 7 : "distributed perception"
- TYPE 8 : "single agent"

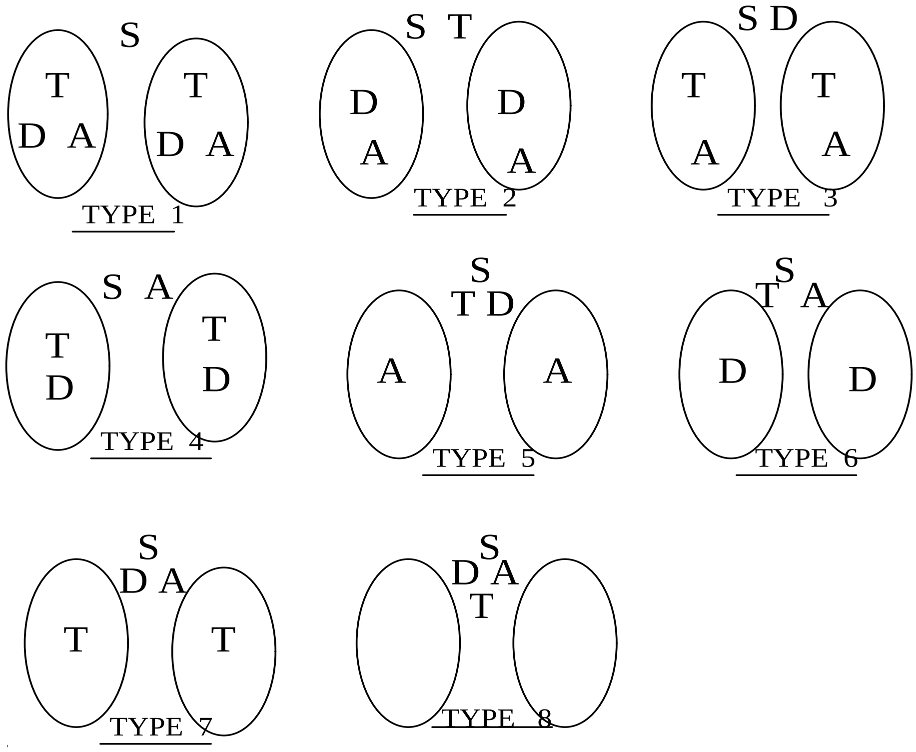

그림 10: Intelligent Agent 시스템들의 결합 형태

제 4 절 결 론

지금까지 Distributed Intelligent Agents 시스템에 관련하여 Intelligent Agent 기술과 Multiagents 기술들을 개략적으로 살펴 보았다. Distributed Intelligent Agents 기술의 기본이 되는 Intelligent Agent 기술은 인공지능(A.I)기술에 근거하고 있는 것 같으며, 이를 현재 유행처럼 확산되고 있는 분산처리 기술과 접목시킨 것 같다. 따라서, 전반적으로 기존 인공지능을 연구하던 사람들이 위주가 되어 연구 활동이 지속되고 있는 것 같다.

물론 조사가 충분히 이루어 지지 않았지만, Distributed Intelligent Agent기술을 현장에서 적용하기에는 시기 상조인 것 같다. 많은 분야에서 연구 및 세미나, Workshop이 계속 활발하게 진행중이며, 일부 제품이 구현되고 있으나 아직 Agent 개발용 언어, Agent 모델, 개발 방법론 등에 대한 표준화는 큰 진전이 이루어 지지 않고 있는 것 같으며, 특히 현장 응용 프로그램에서 중요시되는 실시간 보장이 아직은 어려운 것 같다.

지금 현재 현장에서 시스템 구성시 제일 문제가 되는 것은 시스템의 개방성이다. 이를 Distributed Intelligent Agent System에서도 똑같은 고민거리로 생각된다. 지난 통신 프로토콜의 표준화 이력을 살펴 볼 때 표준화까지는 많은 시간이 소요될 것 같다. 24 시간 연속 운전에 대한 안정성과 실시간성을 요구하는 제어환경에서는 새로운 기술을 적용시키기가 매우 힘이 든다. 실패에 대한 위험 부담이 너무 크기 때문이다.

그러나, 이러한 기술이 현장에 적용될 수만 있다면 제어의 꿈인 무인화 공장을 구성하는데 크나큰 기여를 할 것 같다. 즉, 공장 제어요소별로 제어 Agent들을 구축하며, 이들을 제품 생산 Agent, Mainternance Agent, 제품 수주 Agent, 제품 판매 Agent, 생산계획 Agent등과 같이 Distributed Intelligent Agent 시스템으로 전 제철소 전산시스템을 구축한다면 현장의 무인화, 자동화는 물론 많은 분야에서 성력 효과를 거둘 수 있을 것 같다.

Distributed Intelligent Agents 기술의 기본이 되는 인공지능 기술이 미흡한 상태에서 금번 조사 활동을 수행하기에는 매우 힘이 든 것 같다. 그러나, 본 조사 활동을 통해 새로운 기술을 접하는 방법을 배웠으며, 이러한 것을 충분히 활용한다면 현장에서 크나큰 도움이 될 수 있다는 점을 느낀것이 가장 큰 수확이 아닌가 생각해 본다.

참고 서적

[1] ichael Wooldridge and Nicholas R. Jennings (1995). "Applying Agent Technology"

[2] on Fulbridght and Larry M.Stephens. "classification of multiagent systems"

[3] athryn Heilmann, Dan Kihanya, Alastair Light and Paul Musembwa (1995). "Intelligent Agent : A Technology and Business Application Anaysis"

[4] oner , Lenny. "What's an Agent, Anyway ? A Sociological Case Study"

[5] ichael Wooldridge and Nicholas R. Jennings (1995). "Intelligent Agents : Theory and Practice"

[6] yeong-Wuk Jang, Joongmin Chio and Young-Hwan Lim. "Controlling Agent Interactions in a tightly Coupled Multiagent Agents"

[7] BM (1995). "Intelligent Agent Strategy"

[8] ichael R. Genesereth and Steven P. Ketchpel. "Software Agents"

[9] 지능형 에이전트" 앞세워 네트웍 통합관리 앞당겨 , 오픈컴퓨팅, (95년 1월) 110 - 153 Page

[10] 사이버스페이스 시대의 동반자 소프트웨어 에이전트", 김 태규, 컴퓨터 월드 (95년 5월) 168 - 175 Page

[11] 소프트웨어 에이전트란 무엇인가" 김 태규, 컴퓨터타임즈 (95년 4월) 62 - 65 Page