

## 지능형 아바타 에이전트 시스템의 모델링

### Modeling of Intelligent Avatar System

김대수

Dae Su Kim

한신대학교 컴퓨터학과

#### 요 약

최근 들어 인간과 컴퓨터를 연결시켜주는 지능형 시스템에 대한 관심이 커지고 있으며, 컴퓨터 사용자들의 다양한 요구사항들을 수용할 수 있는 지능형 에이전트 시스템에 대한 연구가 가속화되고 있다. 본 논문에서는 사용자 인터페이스의 역할을 담당하는 지능형 아바타 에이전트를 도입함으로써 사용자에게 보다 편리하고 친숙한 환경설정을 제공하며 다양한 분야에 응용할 수 있는 기반을 마련하였다. 지능형 아바타 에이전트 시스템이 모델링 되었으며 이에 따른 몇 가지의 구현 결과 아바타 에이전트, 메일 체크 기능의 면에서 좋은 결과를 나타내었다.

#### Abstract

Recently, many people are interested in the intelligent system that connects human and computers. And the studies on the intelligent agent system that can manage computer user's diverse requirements are now being accelerated. In this paper, we study the modeling of intelligent avatar system that can provide the user more convenient and user-friendly environments by adopting intelligent avatar agent, and it can be applied to various fields. Modeling of intelligent avatar agent is accomplished and part of it is experimentally implemented, and it showed successful results in the field of avatar agent, mail check functions, and scheduler.

**Key Words :** 지능형, 아바타, 에이전트, 사용자 인터페이스, 스케줄러

#### 1. 서 론

컴퓨터 기술의 획기적인 발전과 폭넓은 보급은 현대인들에게 있어서 정보처리와 정보검색을 통한 컴퓨터의 이용이 생활의 일부가 되었고, 컴퓨터를 사용하는 시간이 점차 증가함에 따라 사용자 중심의 다양한 콘텐츠가 쏟아지고 있다. 과거의 콘텐츠는 보다 많은 정보와 작업의 효율성을 중점적으로 고려하고 있었지만 근래에는 사용자의 다양한 욕구 즉, 사용자 중심의 인터페이스와 사용자 중심의 정보화 등을 만족시키기 위한 새로운 대안의 필요성이 강조되고 있다[1, 2, 6].

본 논문은 그러한 사용자 중심의 인터페이스와 정보화에 초점을 맞춘 지능형 에이전트 시스템의 모델[4, 5, 9, 10, 11]로서 기본적인 스케줄, 알람 및 메일 기능 이외에 사용자가 관심 있는 주식 시세 등에 대한 실시간 정보를 제공할 수 있도록 하였다[3, 7, 8]. 또한 전역 후킹(hooking)을 사용하여 활성화된 윈도우에 따라다니는 아바타(avatar)가 중요한 정보나 항상 체크해야 할 내용을 애니메이션을 통해 정보를 표시하도록 하였다. 기존 컴퓨팅 환경에서 주요 일정이나 자신이 알람을 받고자 하는 날에 알람을 받거나, 자신이 관심 있는 항목에 대해서 지속적인 정보를 받고 싶을 때는 그 한가지

작업에 관심을 집중시킬 필요가 있었다. 자신이 증권에 관심이 있으면 증권 사이트에 접속하여 자신이 원하는 정보를 인터넷 익스플로러 등의 표시 프로그램을 이용하여 윈도우를 띄워 놓고 자신이 하고 있는 작업과 번갈아 가며 멀티캐스팅(multi-tasking)을 해야 하며, 알람을 받고자 하는 내용이 있을 경우 항상 그 부분에 대하여 집중하고 있어야 한다.

본 논문에서는 이러한 일련의 작업들을 지능적으로(intelligently) 프로그래밍하여 지능형 아바타 에이전트 시스템을 구성하였다. 이 시스템은 아바타를 이용하여 사용자에게 좀더 친숙한 환경설정을 제공하며, 제공하는 정보들은 각각 사용자에게 맞게 설정된 환경설정 데이터나 다른 알고리즘을 이용하여 정보가 변형, 가공, 삭제, 추가되어지며 알람이 되어지는 내용은 다시 아바타를 통하여 사용자에게 알려지게 된다.

#### 2. 지능형 아바타 에이전트 시스템의 모델링

지능형 아바타 에이전트 시스템은 사용자의 개인비서로 활용되어지도록 모델링 되었다. 주요 기능으로는 스케줄, 알람, 약속과 같은 일반 개인비서 프로그램의 기능과 웹서버(web server)에서 메일(mail)을 체크하고 관심 있는 주식에 대한 정보를 실시간으로 출력할 수 있도록 하였다. 또한 아바타 인터페이스를 통해 보다 편리하고 안정적인 인터페이스를 제공한다. 아바타 캐릭터는 디렉토리에 각 행동과 관련된 연속 그림이 비트맵으로 저장되어 있기 때문에 사용자가 원하는

접수일자 : 2003년 8월 16일

완료일자 : 2003년 10월 6일

이 논문은 2003년도 한신대학교 특별연구비 지원에 의하여 연구되었음.

다면 인터페이스 자체를 다른 그림으로 교체하여 활용할 수도 있다.

지능형 아바타 에이전트 시스템의 전체적 구조는 그림 1과 같이 사용자와의 인터페이스 역할을 담당하는 Agent System 부분과 설정 및 제어를 담당하는 Management 부분으로 나뉘어진다. 획일화된 네모난 윈도우 환경 속에서 캐릭터를 데스크탑 속에 비서처럼 활용할 수 있도록 하는 부분이 Agent System이다. Agent System은 사용자 취향에 맞게 캐릭터들을 골라서 사용할 수 있으며 캐릭터들은 인터넷을 통하여 쉽게 업데이트될 수 있다.

Management 부분은 이런 Agent System과 통신을 하며 비서 기능을 제공하는 부분이다. Management 부분은 크게 설정 부분과 스케줄 부분으로 나뉘어져 있으며 설정된 내용과 스케줄 내용에서 얻은 데이터를 가지고 Management 부분이 삭제 및 추가할 수 있으며, Agent System과 통신을 통하여 그 내용을 적절히 가공하며 사용자가 원하는 환경에 맞추어 원하는 정보를 제공하게 된다. Configuration 부분은 크게 Character Information, Pop Setting, Stock으로 나뉘어진다. Character Information은 Agent Interface에서 캐릭터가 나타나는데 필요한 여러 가지 환경을 설정하는 부분으로 사용자가 컴퓨터를 활용하는데 있어 불편을 최소화하면서 주목도가 가장 높은 부분을 사용자에게 맞추어 설정하는데 있다. Desktop 옵션은 컴퓨터 바탕 화면을 기준으로 캐릭터가 나타나게 할 수 있고, Window 옵션은 창을 기준으로 캐릭터를 나타나게 해주며, Hooking 옵션은 전역 후킹을 통하여 현재 사용자가 작업을 하고 있는 창위에 나타날지의 여부를 선택할 수 있도록 해준다. Pop Setting 부분은 사용자의 메일을 검색하여 새로운 메일이 있을 경우 메일의 도착 여부를 알려주며 사용자 각 개인들에게 네트워크 라인에 맞는 Pop Setting으로 사용자의 환경에 따라서 쾌적한 컴퓨팅 환경을 제공할 수 있도록 한다.

Stock 항목은 주식에 관심 있는 사용자를 위하여 Agent System이 설정해 놓은 해당 주식 종목들을 실시간으로 검색하여 변동유무를 알려주는 항목으로서 사용자가 일일이 증권 정보제공 홈페이지 및 사이트에 접속하여 화면을 집중하지 않고서도 마치 주식전용 비서가 있는 것처럼 사용자들에게 필요한 정보를 알려주는 역할을 한다.

이 지능형 아바타 에이전트 시스템은 기본적으로 마우스의 오른쪽 클릭으로 나타나는 서브 메뉴를 통해 각 기능 및 설정 옵션에 대한 제어가 가능하다. 각 기능에 대한 구현은 각 기능별 설명 부분에서 따로 서술하였다.

아바타 인터페이스는 윈도우 후킹을 통해 현재 활성화 되어 있는 윈도우 위를 이동한다. 즉 아바타가 활성화 되어 있다면 아바타는 다른 작업을 하고 있는 사용자를 언제나 따라다니며 스케줄, 알람, 약속, 메일, 주식에 관련된 정보를 사용자에게 보여주게 된다.

대부분의 기능이 클라이언트 종속적으로 설계된 아바타 시스템은 차후 메신저 서비스 등 서버 종속적인 부가기능을 가질 경우 기존에 웹사이트에서 동작하는 아바타의 경우에 발생하는 문제인 서버에 대한 부하를 줄이고 서버-클라이언트간의 싱크를 통해 보다 양질의 콘텐츠를 제공할 수 있다.

아바타는 메일 체크와 스케줄러, 주식에 관련된 서비스를 제공하고 있다. 그림 2는 아바타 실행시 화면의 중앙에 놓이도록 설정된 아바타의 모습이다. Agent System은 사용자가 보기에 가장 편한 환경에서 나타나도록 설정될 수 있다. Agent System의 서비스는 사용자 별로 다른 설정을 통해서

다양한 정보를 각기 다르게 보여주게 된다. 설정을 위해서는 아바타에서 오른쪽 버튼을 누름으로써 서브 메뉴를 호출하게 되고, 원하는 설정 메뉴에 들어가서 환경을 바꾸면 Agent System은 설정된 환경에서 원하는 데이터를 가공하여 아바타를 이용하여 알람을 주게 된다. 그림 3은 아바타의 캐릭터 위에서 오른쪽 버튼을 클릭하여 서브 메뉴를 나오게 한 화면이다.

### Intelligent Agent System

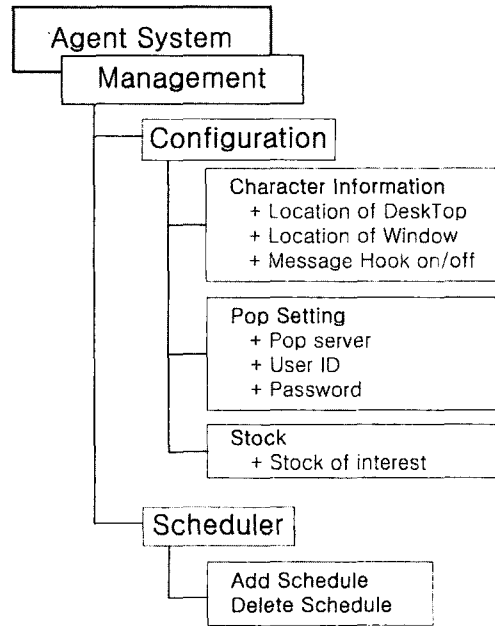


그림 1. 지능형 아바타 에이전트 시스템  
Figure 1. Intelligent avatar agent system

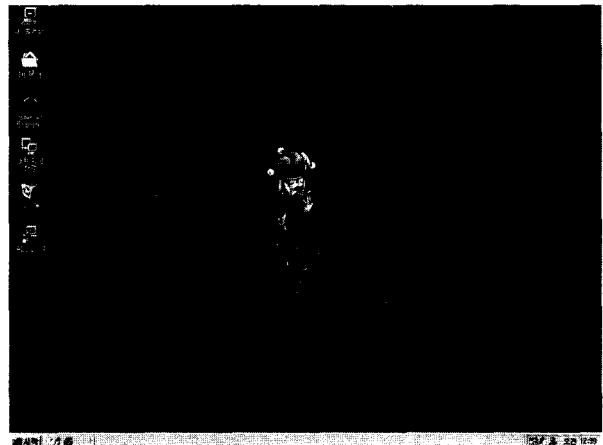


그림 2. 설정된 아바타  
Figure 2. Selected avatar

메일 체크 기능은 저장되어있는 사용자의 정보를 참고한 후 클라이언트 PC가 POP(Post Office Protocol)3 프로토콜의 메일 서버에 접속한 후, 인증을 거쳐 메일을 받아오는 과정을 거쳐 동작한다. 사용자는 자신이 사용하고 있는 메일의 Pop



그림 3. 아바타의 서브 메뉴  
Figure 3. Sub-menu of avatar

server 주소, account, password 및 자신이 메일 서버를 검색할지의 여부를 설정해 놓으면 Agent System은 설정되어 있는 시간 단위로 메일 서버를 검색하여 새로운 메일이 도착할 경우 그 내용을 아바타를 통하여 알림 기능을 수행한다. 사용자는 이 기능을 통하여 메일 서버에 직접 접속하여 신규 메일의 도착 여부를 확인하지 않아도 메일의 유무를 알 수 있으며, 중요하지 않은 메일을 자주 처리할 필요 없이 한번에 처리할 수 있도록 해준다. 사용자는 아바타의 알림을 보며 메일 수신 여부를 실시간으로 확인할 수 있다. 메일 수신여부에 대한 출력 화면은 그림 4와 같다. 메시지 출력은 애니메이션과 다이얼로그 박스를 통하여 출력된다.



그림 4. 메시지 출력  
Figure 4. Message output

아바타는 사용자가 현재 작업 중인 윈도우를 중심으로 가장 편한 위치에 나타나게 설정할 수 있다. 윈도우 메시지를 후킹하여 현재 활성화되어 있는 윈도우를 찾아낸 다음 윈도우 옆쪽으로 방해가 되지 않을 만큼 겹침 효과를 줌으로써 좀 더 친숙하게 하고 또한 캐릭터를 통한 기능의 사용에 방해되지 않게 된다. 그림 5는 현재 사용자가 메모장 윈도우를 열어 놓은 상태에서 아바타가 이를 인식하여 메시지를 전역 후킹한 다음 캐릭터의 위치를 종전의 배경화면 가운데에서 사용자가 사용하고 있는 메모장 창의 좌측 상단으로 이동한 형태이다. 이와 같이 재설정함으로써 사용자는 항상 아바타를 보면서 작업을 할 수 있고, 자신이 설정해 놓은 약속이나

데이터들이 표시될 때 가장 보기 편한 위치에서 그 내용을 볼 수 있으며 작업에 있어서 캐릭터가 방해가 되지 않게 된다. 메시지 후킹을 위해 아바타의 Management와는 별개인 멀티스레드가 띄워지게 되며 스레드는 윈도우의 창 변화 및 활성화의 변화를 감지하여 이를 아바타 Management에게 알리고 Management는 아바타의 인터페이스 부분을 다시 그려 줌으로써 그 기능을 구현한다.

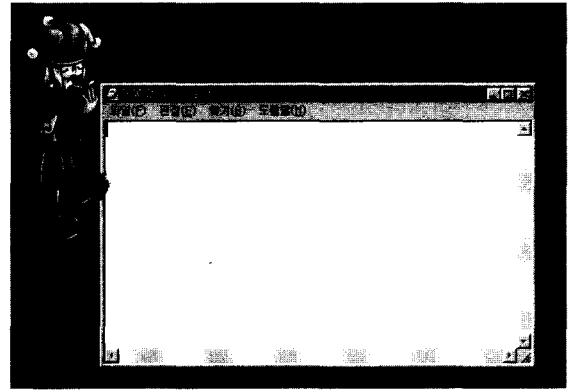


그림 5. 재설정된 아바타  
Figure 5. Re-adjusted avatar

스케줄러 기능은 사용자의 스케줄 및 약속과 관련된 정보를 레지스트리에 저장하고 있다가 일정 시간 간격으로 체크하여 위에서 제시한 아바타 인터페이스의 애니메이션과 다이얼로그 박스를 통하여 출력한다. 설정된 내용은 파일 형태로 저장되며 보조기억장치에 변화가 있을 때마다 다시 저장하게 된다. 저장된 내용은 컴퓨터가 리셋 되거나 종료되었을 경우에도 그 내용을 잃지 않고 계속적인 알림에 사용되며 사용자가 원하는 횟수만큼 알림을 받을 수 있도록 한다. 스케줄의 개수는 정의된 형식의 데이터 구조를 행렬(array) 형태로 가지고 있다가 조건에 만족하는 스케줄 데이터 형식을 불러내어 번역 작업을 한 후 캐릭터 인터페이스를 통해 그림 6과 같은 형태로 사용자에게 알려준다.

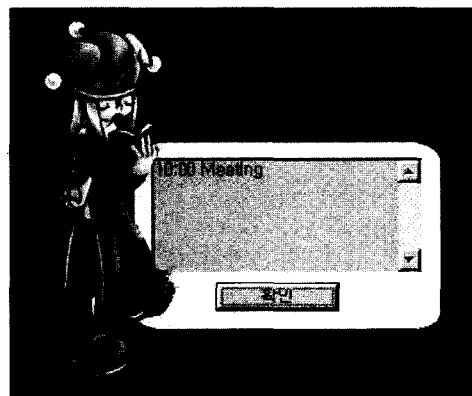


그림 6. 스케줄러 기능  
Figure 6. Scheduler function

### 3. 지능형 아바타 에이전트 시스템의 설계

메일 서비스는 POP3 프로토콜을 이용하고 있다. POP3 프

로토콜은 시스템 성능이나 경제성, 보안 등의 문제로 각각의 PC를 메일 서버로서 사용하기에는 적합하지 않기 때문에 만들어진 프로토콜로서, 메일 서버로 지정된 시스템이 외부로부터 메일을 받으면 네트워크의 PC 클라이언트들이 이 메일 서버에 접속해 자신의 메일을 받아온다. 현재는 3버전을 주로 사용한다.

메일 체크 시스템은 그림 7과 같이 옵션을 통해 사용자로부터 메일 주소와 인증 키인 ID, PASS를 얻는다. POP3 서버는 사용자가 사용하고 있는 메일 주소의 POP3 Server 주소를 넣고 account와 password는 각각 사용자 ID와 암호란에 넣는다. Agent Manager의 메일 부분을 담당하는 Mail Check System Module은 사용자가 지정해 놓은 Server 주소와 account, password를 가지고 서버에 접근하여 새로운 메일의 도착 여부를 판별한 다음, 새로운 메일이 있을 경우 그 내용을 아바타를 통하여 알려주게 된다. 이때 사용자는 메일 클라이언트를 통하여 메일이 새로 왔는지 매 분마다 확인할 필요 없이 아바타가 확인하여 연결하여 주므로 번거로움을 피할 수 있으며, 중요한 메일이 도착할 시 빼 먹는 경우를 방지할 수 있다.

사용자는 사용자의 네트워킹 상태에 따라 메일 확인 횟수를 지정할 수가 있으며 적절한 확인 횟수 지정으로 보다 좋은 컴퓨팅 환경에서 작업을 할 수 있다. 또한 메일 클라이언트를 확인할 때보다 시스템의 리소스(resource)를 적게 쓰면서도 더욱 빠르고 경제적인 메일 환경을 만들 수 있다. 테스트 상에서 활성화 되어있는 윈도우 위에 있던 아바타가 메일 수신을 확인하고 알리는 결과 화면은 그림 8과 같다.

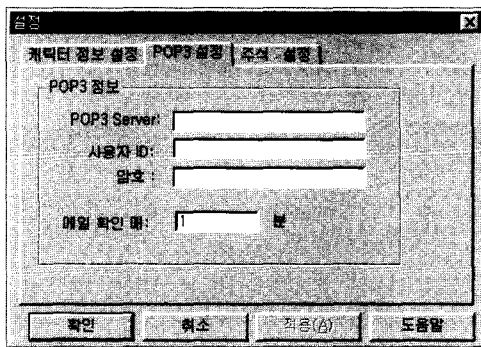


그림 7. 메일 체크 시스템  
Figure 7. Mail check system

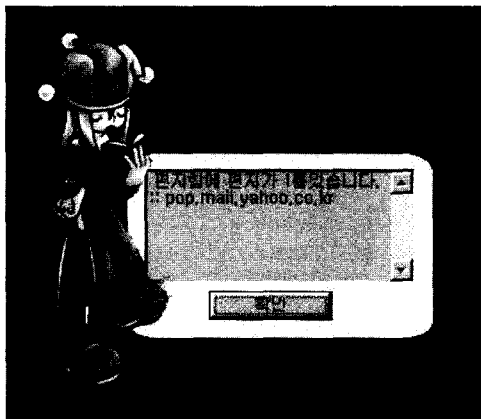


그림 8. 아바타가 메일 수신을 확인하고 알리는 화면  
Figure 8. Output of e-mail by avatar

아바타 인터페이스는 그림 9의 메일 체크 시스템 모듈과 같이 데이터베이스에 저장되어 있는 사용자의 메일 주소와 인증키를 참고하여 웹 서버에 접속한다. POP3 접속시 메일 내용은 가져오지 않고 메일의 개수만 가져오는데 메일의 확인 및 재전송 등의 작업 시에는 좀더 높은 보안 수준의 메일 클라이언트를 사용하게 된다. 서버를 통해 받은 데이터는 파싱되어 사용자에게 필요한 메일 수신여부 확인과 처리 절차를 거치게 되어 확인된 메일이 있을 경우 아바타 인터페이스를 통해 출력된다.

### Mail Check System Module

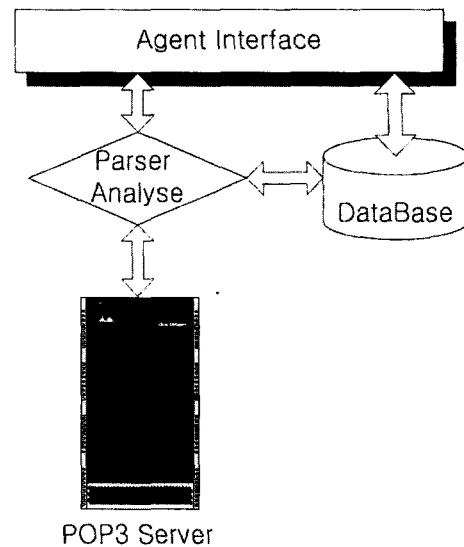


그림 9. 메일 체크 시스템 모듈  
Figure 9. Mail check system module

인증 과정에서 소켓을 통한 네트워크와 아바타의 중간에 위치하여 상호간에 연동되도록 하는 것이 파서 클래스이다. Parser Analyse는 기존의 정보를 통해 웹 서버에 응답을 요청하고 응답여부를 분석하여 처리하는 역할을 담당하고 있다. 파서 클래스는 소켓 클래스에서 사용되며 서버로부터 받은 첫 번째 토큰을 숫자로 변경하여 메일 수를 체크하고 서버와의 통신시 발생된 서버의 응답을 반환하여 연결상태를 확인한다.

소켓 클래스는 동기화 소켓으로 구성되어 있으며 다른 기능과의 싱크를 위해 크리티컬 섹션(critical section)을 사용하여 다른 메시지와 관련된 출력이 있을 경우 현재 아바타 상태를 체크하여 메시지 출력에 관련된 처리를 잠시 보류하여 대기한다. 파서 클래스의 기능 구현은 그림 10과 같다.

네트워크에는 POP3 메일 서버가 있어 게이트웨이를 통해 아바타 클라이언트와 통신한다. 이 연구에서 구현된 메일 서버와 메일 클라이언트의 통신 순서는 다음과 같다.

최초 메일 서버가 TCP 110번 포트에서 메일 클라이언트의 소켓 연결을 기다리고 있다. 메일 서버는 메일 클라이언트의 소켓 연결 요청을 받았을 때 인사말과 함께 긍정적인 응답(+OK)을 보낸다. 다음으로는 인증 단계에 들어가는데, 메일 클라이언트는 사용자명과 암호를 서버에 차례로 전송해 메일 서버로부터 사용 인증을 받는다. 인증 단계가 끝나면 작업 단계로 연결된다. 이 단계에서 클라이언트는 메일 서버에

게 명령을 내릴 수 있다. 테스트에서는 단순히 현재 서버에 저장된 메일의 개수만을 얻어온다.

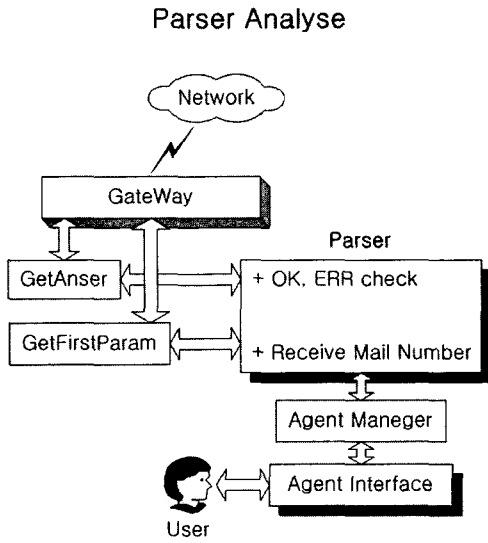


그림 10. 파서 클래스  
Figure 10. Parser class

원하는 메일 개수의 정보를 받고 메일 클라이언트에서 종료를 요청하면 메일 서버가 외부로부터 오는 메일을 저장할 수 있는 단계인 변경 단계로 바뀐다. 변경 단계까지 진행된다면 한번의 메일 체크가 끝난 셈이다. 메일 서버로부터 종료 요청에 대한 승인 및 소켓종료 메시지를 받게 되면 메일 서버와의 통신은 끝난다.

파서는 위의 과정을 거쳐 메일 서버로부터 받은 메시지 (+OK, +ERR)를 분석하여 처리하는 기능과 메일 개수에 대한 정보를 가지고 있는 스트링을 분석하여 첫 번째 파라미터인 메일 개수를 추출하는 역할을 하고 있다. 분석된 메시지는 +OK의 경우 다음 통신 단계로 넘어갈 수 있는 정보가 되며 +ERR 메시지의 경우 잘못된 통신 연결이라는 메시지가므로 서버와의 통신을 취소하고 소켓을 종료하도록 진행한다.

메일 개수를 얻을 수 있는 단계가 되어 GetFirstParam 단에서 정보를 습득하였다면 이 정보는 아바타의 여러 기능들의 제어와 관리를 담당하는 Agent Manager를 통해 관리된다. Agent Manager는 이 정보가 사용자에게 필요한 정보인지, 즉 이미 확인했거나 사용되어진 중복 정보인지에 대한 체크와 다른 알람이나 스케줄 또는 주식 서비스와 동시에 연동되어 중복 효과가 일어나지 않도록 총체적인 관리를 해주며 다른 메시지를 표시해야한다면 그 동안 잠시 처리를 멈추고 사용자가 새로운 메시지를 받을 준비가 되기를 기다린다. 메시지는 아바타 인터페이스를 통한 애니메이션과 다이얼로그를 통해 사용자에게 전달되기 때문에 중계자 역할을 할 수 있는 매니저 클래스가 없다면 확인되지 않은 정보를 잃어버릴 수 있는 문제를 내포하고 있다. 사용자에게 최종적으로 전달된 메시지는 사용자 승인 후 리셋 된다.

메일 체크 기능은 현재 기본적인 메일 개수의 실시간 확인 기능만 가능하다. 메일 서버와 메일 클라이언트 사이의 통신 과정에서 인증을 얻은 경우 메일 내용을 직접 확인하는 것이 가능하기 때문에 불필요하게 메일이 왔다는 것을 확인한 후 다시 그 내용을 확인하기 위하여 메일 사이트에 접속해야 하는 불편을 줄일 수가 있다.

#### 4. 결론

본 논문에서는 사용자 인터페이스의 역할을 담당하는 지능형 아바타 에이전트를 도입함으로써 사용자에게 보다 편리하고 친숙한 환경설정을 제공하며 다양한 분야에 응용할 수 있는 기반을 마련하였다.

지능형 아바타 에이전트 시스템이 모델링 되었으며 이에 따른 몇 가지의 구현 결과 아바타 에이전트, 메일 체크 기능, 그리고 스케줄러의 면에서 좋은 결과를 나타내었다. 특히 아바타 에이전트의 윈도우 상황에 따른 사용자 편의성이 구현 되었으며, 중요한 메일이 도착할 경우에 지능적으로 이 사실을 제때에 알려주고 알람을 해주는 기능이 구현되었다.

지능형 아바타 에이전트 시스템의 전체 설계 및 구현, 스케줄러에 관한 보다 정교한 설계와 구현, 그리고 증권 정보와 관련된 모듈 등에 관한 추후 연구가 현재 진행중이다.

#### 참 고 문 헌

- [1] S. Bird, "Toward a taxonomy of multi-agent systems", International Journal of Man-Machine Studies, Vol. 39, pp. 689-704, 1993.
- [2] F. Cheong, Internet Agent : Spiders, Wanderers, Brokers, and Bots, New Riders, 1996.
- [3] R. Goodwin. "Formalizing Properties of agents", Technical Report CMU-CS-93-159, School of Computer Science, Carnegie-Mellon Univ., 1993.
- [4] Thorsten Graf and Alois Knoll, "A Multi-agent Approach to Self-Organizing Vision System", Proceeding of the 1st Asia-Pacific Conference on IAT, 1999.
- [5] T. B. Ho, T. D. Nguyen and N. B. Nguyen, "An Agent-Based Architecture in Knowledge Discovery and Data Mining", Proceeding of the 1st Asia-Pacific Conference on IAT, 1999.
- [6] Dae Su Kim and Chung Suk Kim, "Modeling of Bank Asset Management System based on Intelligent Agent, International Journal of Fuzzy Logic and Intelligent Systems", Vol. 1, No. 1, June 2001.
- [7] Dae Su Kim, Chang Suk Kim, and Kee Wook Rim, "Modeling and Design of Intelligent Agent System", International Journal of Control, Automation, and Systems, Vol 1, No. 2, June 2003.
- [8] Dae Su Kim, et. al., "Design and Implementation of Intelligent Agent System for Pattern Classification", International Journal of Fuzzy Logic and Intelligent Systems, Vol. 1, No. 2, Nov. 2001.
- [9] Pattie Maes, Designing Autonomous Agents, MIT Press, 1994.
- [10] Mart T. Maybury(editor), Intelligent Multimedia Interfaces, MIT Press, 1993.
- [11] Horia-Nicolai Teodorescu(editor) etc., Intelligent Systems and Interfaces, Kluwer Academic Publishers, 2000.

저 자 소 개



김대수(Dae Su Kim)

제 11권 9호 참조

Phone : +82 - 031 - 370 - 6784

Fax : +82-031-372-3343

E-Mail : daekim@hanshin.ac.kr