韓國情報敎育學會論文誌
第2卷 第1號 pp.130~137.

보안 서비스를 제공하기 위한 이동 에이전트 시스템의 설계에 관한 연구

김 현 배
부산교육대학교 컴퓨터교육과

A Study for Mobile Agent System to Support Security Service

HyunBae Kim
Dept. of Computer Education, Pusan National University of Education

Abstract

An agent is a computer program that acts autonomously on a computer system. A mobile agent is not bound to the system where it begins execution. It has an ability to transport itself from one system in a network to another. Mobile agent system solve the network traffic problem. Moreover, the agent may utilize the object services of the destination agent system. In this study the mobile agent system is introduced to support security service.

Ⅰ.서 론

네트워크 기술의 발전으로 기존의 데이터 처리 기능이 주 업무였던 컴퓨터가 강력한 통신 매체로 부상하게 되었다. 이러한 변화에 맞추어 데이터 이동만으로 서비스를 제공하던 기존의 통신 응용 프로그램들은 서버의 과부하를 줄이고, 네트워크 이벤트 발생을 줄이면서 필요한 서비스를 사용자에게 지원하기 위한 하나의 방법으로 이동 에이전트를 통하여

서비스를 지원하고 있다.

이동 에이전트란, 현재 객체의 상태와 실행 가능한 코드를 포함하는 객체로서, 분산 환경 하에서 특정 서비스를 제공하기 위하여 서로 통신할 필요가 있는 호스트 간을 직접 이동하면서 실행되는 객체를 말한다. 즉, 이동 에이전트는 에이전트가 수행을 시작한 시스템에 한정되지 않고, 서로 다른 시간에 다른 시스템에서 수행될 수 있다.

이동 에이전트는 이질적인 네트워크에서 자신의 제어로 호스트 사이를 옮겨 다니고, 각 호스트의 다른 에이전트와 상호 동작하거나 자원을 이용하여 맡겨진 임무를 수행하고, 수행이 끝난 경우 처음 시작된 시스템으로 되돌아온다. 따라서 코드는 이동될 모든 호스트에 똑같은 형식으로 동작되어야 하므로 이식성 있는 중간 언어로 작성되어 이동될 호스트에서 다시 인터프리트나 컴파일을 하지 않고 수행될 수 있어야 한다.

본 연구에서 제안하는 보안 서비스를 지원하기 위한 이동 에이전트 시스템은 각 호스트들의 보안에 관계된 점검 사항들을 관리자가 일일이 확인하는 방식이 아닌 관리자를 대신한 이동 에이전트를 사용하는 시스템을 말한다.

Ⅱ.이동 에이전트 시스템

이동 에이전트란 사용자를 대신하여 특정 작업을 수행하는 자율적인 프로그램으로서 이질적인 네트워크 상에서 자신의 제어로 호스트 사이를 이주하고, 각 호스트의 다른 에이전트와 상호 동작하거나 맡은바 임무를 수행한 후 자신이 출발한 곳으로 되돌아온다. 여기서 이동 에이전트 시스템이란 이동 에이전트를 생성, 이동, 수행, 전송, 해석 및 폐기 등 에이전트의 생명주기를 관리할 수 있는 플랫폼이다.

1.이동 에이전트의 개념

이동 에이전트는 에이전트(Agent), 플레이스(Place), 여행(Travel), 모임(Meeting), 연결(Connection), 권한(Authority)등과 같은 개념으로 이루어지며 이러한 개념들을 에이전트 프로그래밍 언어로 구현되어 진다.

에이전트는 사용자를 위해 자동적으로 행동하는 프로세스로서, 이동 에이전트는 수행을 시작한 시스템을 떠나 자신의 의지에 따라 하나의 플레이스에서 다른 플레이스로 옮기고 서로 다른 시간에 서로 다른 플레이스를 사용한다.

플레이스는 에이전트를 수행시키는 에이전트 시스템의 수행환경을 제공한다. 홈 플레이스는 이동 에이전트의 출발점과 도착점 기능을 수행한다. 플레이스는 이동 에이전트를 받아들이며, 수행 환경을 제공하고 사용자 인터페이스를 포함한다.

여행은 에이전트가 하나의 플레이스에서 다른 플레이스로 이동하는 기능을 제공한다. 에이전트가 원격에서 제공된 서비스를 받기 위하여 필요한 때 이동하고, 서비스를 받은 후 홈 플레이스로 되돌아온다. 이를 위하여 프로시저와 상태가 이식되어 처리될 수 있어야 한다.

모임은 같은 플레이스에 있는 두 개의 에이전트가 만나는 기능으로 에이전트들이 서로의 프로시저를 호출하는 것이다.

연결은 서로 다른 플레이스에 있는 2개의 에이전트가 서로 통신할 수 있도록 연결해 주는 것이다. 연결은 대화식 응용의 사용에서 사용된다.

권한은 에이전트 또는 플레이스가 다른 것들과 구별하기 위한 것이다. 파일 접근을 통제하려는 것을 예로 들면, 파일 서버는 파일을 열람하거나 삭제하려는 프로시저의 권한을 알아야 통제할 수 있다. 어떤 에이전트가 플레이스에 들어오려고 시도한다면 그 에이전트의 권한을 파악하여 특정 권한을 가지지 않았을 경우 플레이스의 방문을 통제할 수 있다. 또한 특정 권한의 에이전트들 사이의 통신만을 허용할 수도 있다.

2.관련연구

에이전트 기술 중 본 연구에서 적용될 이동 에이전트 기술은 RPC와 RP 형태의 확장으로 볼 수 있다. RPC는 사용자가 표준화된 프로시저 호출기법을 사용하여 제공되는 시스템의 오퍼레이션을 호출하도록 하는 것이며 RP는 사용자가 하나의 서브프로그램을 서버에게 보내고, 그 서브프로그램은 서버에서 수행이 되고, 그 결과를 사용자에게 전달해 주는 기법이다. 이동 에이전트 기술은 RP를 확장한 개념으로서 에이전트를 임의의 장소로 이동시킨다.

이동 에이전트 시스템은 에이전트를 프로그래밍하기 위한 언어, 에이전트를 설치하기 위한 라이브러리, 에이전트를 수행시키기 위한 인터프리터, 에이전트를 전송하기 위한 프로토콜, 에이전트를 보호하기 위한 보안등의 측면에서 분류될 수 있다.

(1)에이전트 언어 측면에서의 구분

본 연구에서는 이동 에이전트 언어 측면으로 분류할 수 있는 Java에 기반한 에이전트 시스템을 목표로 한다.

Java를 기반으로 하지 않는 이동 에이전트 시스템으로는 General Magic사가 전자 상거래를 위한 이동 에이전트 시스템으로 개발한 Telescript가 있다. 이 시스템은 Telescript언어를 사용하였다. 독일의 Kaiserslautern대학에서 스크립트 언어인 Tcl/Tk를 이용하여 Ara를 개발하고 있으며 Dartmouth 대학에서 정보 검색을 효율적으로 수행하도록 Tcl/Tk를 이용하여 이동 할 수 있는 에이전트 시스템 Agent Tcl을 개발하고 있다.

Java를 기반으로 한 이동 에이전트 시스템으로는 General Magic사가 Telescript 기술을 Java에 근거하여 다시 개발하고 있는 Odyssey가 있다. 또 다른 시스템으로는 IBM Tokyo Research Lab에서 개발한 Agent Workbench라는 것이 있다. OSF RI(Open Software Foundation Research Institute)는 MOA라는 프로젝트를 수행중이다.

(2)각 시스템의 특징

General Magic사의 Odyssey는 오디세이 클래스 라이브러리를 제공하고, 사용자는 이를 이용하여 자신의 이동에이전트 응용을 작성할 수 있다.

IBM Tokyo Research Lab에서 개발한 Aglet Workbench는 네트워크 기반 응용을 작성하는 시각적인 환경을 도입하고 있다. 이동 에이전트를 작성하는데 visual builder를 사용하여 개인의 취향에 맞게 빨리 작성할 수 있다. Aglet은 이동할 때 수행중인 자바 프로그램을 임의의 장소로 전송할 수 있기 때문에 자바 애플릿을 확장한 것으로 보면 된다. 본 연구에서도 에이전트 시스템을 작성할 때 IBM Aglet을 사용한다.

IBM Aglet은 위에서도 본 것처럼 사용자 인터페이스 작성에 시각적인 환경을 제공하므로 개발의 편의성과 신속성을 제공한다.

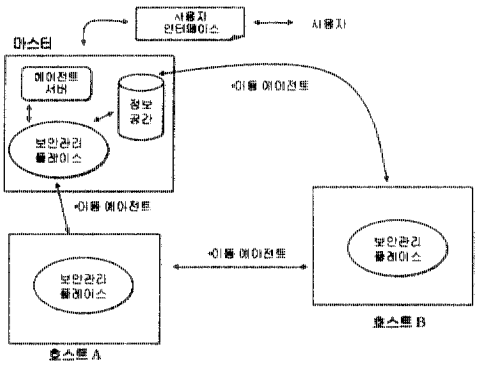
OSF RI의 MOA는 이동 객체가 쓰레드를 갖고 있어 능동적으로 동작하고, 상태 정보를 유지하면서 지능적인 결정을 하는 특성을 갖는다.

Ⅲ.시스템 설계

본 연구의 이동 에이전트 시스템은 보안 기능을 갖는 에이전트가 각각의 호스트들을 이동하며 각 호스트들의 보안사항을 점검하는 형태를 갖는다. 각 호스트를 점검하다가 보안상의 이상여부를 발견하거나 해킹의 흔적을 발견하였을 경우 해당되는 기능을 갖는 또다른 에이전트의 파견을 마스터에게 의뢰하게 된다. 의뢰를 받은 마스터에서는 그 기능을 담당한 에이전트를 해당 호스트에 파견하고 파견된 에이전트는 현재 호스트의 상태를 점검하여 마스터에 보고하게 된다. 보고를 받은 마스터에서는 보고된 내용을 관리자에게 알리게 되며 이 때 일정한 리포트 형식을 취하게 된다.

1.시스템 구성

본 연구에서 제안하는 시스템은 [그림3-1]과 같다. 각각의 호스트는 이동 에이전트를 받아들이기 위한 플레이스를 기반으로 구성되며 에이전트에 대한 모든 관리는 마스터를 통해 이루어진다.



[그림3-1] 시스템 구성

에이전트 서버는 마스터 상에 존재하며 에이전트의 생성, 이동, 실행 및 관리를 위한 것으로 호스트에 파견할 에이전트를 생성하고 에이전트에 대한 임무할당 및 정책 등을 생성, 관리한다. 다시 이것은 에이전트 관리자와 보안 관리자로 구성되어 진다. 보안관리 플레이스는 이동 에이전트의 개념 구조에서 플레이스와 같은 역할을 하는 것으로서 이동 에이전트의 실행 환경을 제공한다. 이러한 보안관리 플레이스는 각각의 호스트에서 존재하며 각 기능에 따라 여러 개의 보안관리 플레이스

가 만들어 질 수 있다. 정보 공간은 에이전트의 생명주기를 관리한다.

(1)마스터

마스터는 이동 에이전트를 생성하고 임무를 부여하며 이동 에이전트가 각 호스트들에 대한 정보를 취합한 후 그 내용을 보고하는 일종의 관리 서이다.

마스터는 에이전트 서버와 정보 공간 및 보안관리 플레이스로 구성된다. 그밖에 관리자를 위한 사용자 인터페이스 부분과 정보에 대한 데이터베이스를 갖는다.

1)에이전트 서버

에이전트 서버는 에이전트에 대한 모든 관리를 하는 부분으로 에이전트에 대하여 에이전트 생성, 에이전트 이동, 에이전트 실행, 에이전트 관리, 그리고 에이전트 폐기와 같은 일들을 수행한다.

에이전트는 목적에 따라 각각 다른 형태로 생성 가능하다. 즉, 에이전트의 임무에 따라 우선순위를 가지며 또한 생명주기도 각기 다르게 적용한다. 또한 데이터를 처리할 호스트의 리스트를 가지며 리스트에 따라 호스트를 방문하여 주어진 임무를 수행하고 임무가 완료되었을 때 다음 호스트로 이동한다. 다음은 에이전트의 임무에 따른 우선순위를 세 단계로 나타낸다.

ㄱ) 최상위 등급(Top Secret) : 가장 높은 우선순위를 가지며 호스트의 모든 정보를 취합가능한 등급이다.

ㄴ) 상위 등급(Secret) : 호스트의 정보를 선별하여 상위등급 정책에 합당한 자료만을 취합가능한 등급을 의미한다.

ㄷ) 중위 등급(Confidential) : 호스트의 일반정보 및 많은 양의 데이터를 취합하여 분석하기 위한 등급이다.

- 에이전트 우선순위

  중위등급 < 상위 등급 < 최상위 등급

- 에이전트 주기

  중위등급 > 상위 등급 > 최상위 등급

- 에이전트 할당 임무량

  중위등급 > 상위 등급 > 최상위 등급

따라서 중요한 데이터의 분석을 요구할 때는 최상위 등급의 에이전트를 생성하며 많은 양의 데이터를 분석할 때는 중위등급의 에이전트를 생성하여 전송한다.

2)정보 공간

정보 공간은 에이전트의 생명주기에 대한 일을 관리하는 부분으로 에이전트의 생성 이후 각 에이전트의 생명주기를 점검하여 주기를 다한 에이전트에 대해 호출을 요청하며 에이전트 상태를 주기적으로 파악한다. 주기가 다한 에이전트에 대해서는 에이전트 서버에 요청하여 폐기를 신청하고 에이전트 서버는 상황에 따라 에이전트를 재생성 또는 생명주기 변경 등의 행동을 취한다. 또한 에이전트의 환경설정값이나 또는 제한 사항이 변경되었을 경우 이를 에이전트에 알리는 역할을 수행한다.

3)보안 관리 플레이스

보안 관리 플레이스는 개념 구조에서의 플레이스와 같은 역할을 하는 것으로 이동 에이전트의 실행 환경을 제공한다. 에이전트 서버로부터 생성된 에이전트는 일정 목적지로 이동된 후 보안 관리 플레이스를 통해 마스터로 접속한다. 따라서 외부로부터 연결을 요청한 에이전트에 대한 관리 공간을 일컫는다. 호스트의 보안 관리 플레이스는 외부로부터 에이전트의 연결을 요청 받고 에이전트와의 모임을 생성하기 위하여 에이전트의 우선순위 등의 권한을 점검한다. 이러한 권한에 위배될 경우 에이전트의 요청을 거절하며 권한이 만족될 경우는 에이전트가 활동할 수 있는 영역 즉, 시스템 자원을 제공한다. 보안 관리 플레이스에 안착한 에이전트는 보안 관리 플레이스에 주둔한 에이전트와 모임을 설정하여 에이전트 임무에 대한 데이터를 취합한다.

(2)호스트

호스트는 실제적으로 사용되는 사용자 시스템이다. 호스트는 에이전트의 연결요청에 대한 프레임을 제공하기 위하여 보안 관리 플레이스를 기반으로 구성된다. 보안 관리 플레이스는 이동 에이전트를 받아들이며, 수행 환경을 제공한다.

(3)이동 에이전트

본 연구에서 실제적인 일을 담당하는 부분이 바로 이동 에이전트이다. 즉, 이동 에이전트는 에이전트 서버로부터 생성되어 주어진 활동을 위해 자동적으로 행동하는 프로세스로서 수행을 시작한 시스템을 떠나 자신의 의지에 따라 하나의 플레이스에서 다른 플레이스로 옮기고 서로 다른 시간에 서로 다른 플레이스를 사용한다. 이동 에이전트는 에이전트 우선순위, 목적지 리스트, 취합 자료 및 점검사항, 비상 환경정보, 자체 데이터베이스, 그리고 에이전트 생명주기와 같은 내용을 포함하여 수행된다.

이동 에이전트는 위와 같은 정보를 바탕으로 마스터를 이동하여 각 주어진 리스트의 호스트를 방문하여 권한정보에 대해 인증을 거친 후 자료를 취합하여 할당된 점검사항에 대해 분석한다. 이 때 비상환경 정보에 대한 자료가 검출되었을 경우 마스터로 이동하여 자료에 대한 정보를 보고한다. 이때 자료에 대한 자체 데이터베이스를 유지한다.

이런 상태로 각 호스트를 방문하며 그 주기는 에이전트의 생명주기에 의존한다. 생명주기는 위에서 언급한 정보 공간에서 주기적으로 점검하여 생명주기에 대한 정보를 이동 에이전트에 통보한다. 주기가 다한 이동 에이전트는 마스터의 보안 관리 플레이스로 이동하여 에이전트 서버의 명령에 따라 폐기 또는 재 생성된다.

(4)사용자 인터페이스

사용자 인터페이스는 관리자와 마스터 사이에서 행해지는 것으로 이동 에이전트가 각 호스트를 돌고 온 후 얻어지는 결과를 보고하는 형식이다. 사용자 인터페이스에는 이동 에이전트가 돌고 온 모든 호스트에 대한 정보와 각 호스트의 보안상태에 내용이 담기고 이것을 웹 형식으로 구현한다. 관리자는 사용자

인터페이스에서를 통해 이동 에이전트의 우선 순위를 설정하거나 이동 에이전트 파견 순서를 조절하는 등의 일이 가능하다. 향후 연구 과제는 보안 정보들에 대한 데이터베이스의 구축을 통하여 이동 에이전트의 보고된 사항을 분석하여 각 호스트의 상태에 따른 대체방안을 관리자에게 보고하는 기능도 사용자 인터페이스에 추가할 수 있다.

2.구현시의 고려 사항

(1)에이전트간의 통신

현재 시스템 설계의 기본 구조는 마스터로부터 파견된 각각의 에이전트가 호스트 시스템을 점검하다가 필요로 하는 다른 기능의 에이전트가 있을 때에 이것을 마스터로 보고하여 마스터에서 다시 에이전트를 생성하여 파견하는 구조로 되어있다. 이런 구조에서 많은 수의 에이전트가 생성되고 활동할 경우 필연적으로 마스터의 에이전트 서버에 많은 노드가 걸리게 되고 이것이 시스템 자체의 성능을 저하시키는 요인이 될 수도 있다. 그러므로, 에이전트간의 통신을 통하여 인접한 호스트에 위치한 에이전트의 도움을 받는다거나 같은 호스트에서 필요로 하는 에이전트가 존재 할 때 이것을 이용할 수도 있다.

(2) 에이전트 보안

이동 에이전트 시스템의 보안 문제는 크게 두 가지 방향으로 고려되어 진다. 하나는 호스트 컴퓨터의 보안이며 다른 하나는 이동 에이전트 자체의 보안이다.

호스트 컴퓨터의 보안은 이동 에이전트 개념구조에서 언급되었던 권한을 사용하여 대처할 수 있다. 또 다른 문제인 이동 에이전트의 경우 만약, 이동 에이전트 자체가 공격을 받아서 악의적인 코드를 갖는 에이전트로 둔갑하거나 또는 가짜 에이전트의 등장으로 인해 호스트 및 전체 시스템이 공격을 받을 수도 있다. 이러한 것을 방지하기 위해 에이전트가 가지고 있는 데이터의 기밀성이 유지되어야 하는 문제와 에이전트 실행 상태에 대한 감사를 해야 문제를 신중히 고려하여야 한다.

Ⅳ. 결론

최근 광범위해진 통신과 인터넷의 발전에 따라 이전보다 더 쉽고 빠른 정보 교환이 이루어지고 있다. 또한 사용자들은 언제 어디서든 정보를 참조할 수 있다. 이에 따라 보안 관리 시스템에 대한 요구는 날로 증가를 더하고 있다. 따라서 유용성과 성능을 저하시키지 않고 보안 관리 시스템에 보안 서비스를 갖추는 것이 주요 과제이다.

본 연구에서는 보안 서비스를 지원하기 위한 이동에이전트 시스템의 모델을 제안하여 보안 관리 서버의 과부하를 줄이고, 네트워크 이벤트 발생을 줄이면서 필요한 보안 서비스를 지원하도록 하였다.

본 연구의 에이전트는 각 호스트들의 보안에 관계된 점검 사항들을 관리자가 일일이 확인하는 방식이 아닌 관리자를 대신한 이동 에이전트를 사용하여 보안 서비스를 제공하는 이동 에이전트 시스템을 설계하였다.

향후의 연구과제로서 이동 에이전트의 보호

에 대한 연구를 계속적으로 연구하여 악의적인 이동 에이전트로부터 안전한 이동 에이전트의 운영을 보장할 수 있는 노력을 기울여야 할 것이다.

참고 문헌

[1]이영화, 이남용, "이동에이전트의 보호", 통신정보학회지 제8권 1호, 1998.3

[2]김평중, 윤석환, "이동 컴퓨팅을 위한 이동 에이전트 시스템", 정보처리학회지, 제4권 5호, 1997.9..

[3]박지은, 김상욱, "이동 에이전트를 지원하는 프로그래밍 언어", 정보처리학회지, 제4권 5호, 1997.9..

[4]D. B Lange and D. T. Chang, "IBM Aglets Workbench : Programming Mobile Agents in Java, A White Paper", IBM White paper, Sep. 1996

[5]J. White, "Mobile Agents White Paper", General Magic White paper

[6]Stan Franklin and Art Graesser, "Is it an agent, or just a Program?: A Taxonomy for Autonomous Agents", Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, 1996

[7]James Gosling, Henry McGilton, "The Java Language Environment: A White Paper", Technical Report, Sun Microsystems, October 1995