

# 웹에이전트의 관리시스템 설계

송성훈\*                      원유현  
홍익대학교 대학원 전자계산학과  
mitchell@hcc.ac.kr, won@cs.hongik.ac.kr

## The Design of Management System for Web Agent

Sung-Hoon Song\*            Yoo-Hun Won  
Dept. of Computer Science, Hongik University

### 요 약

웹서버 사이클을 왕래하며 실행되는 이동에이전트를 본 논문에서는 웹에이전트라고 명명한다. 웹에이전트가 미리 예측하기 어려운 이유로 오동작을 하게 될 때에 이를 제어하기 위한 방법이 필요하게 된다. 웹의 성격상 익명의 사용자가 웹에이전트의 적재 및 실행을 초기화하게 된다. 따라서 웹에이전트의 관리자가 누구이며, 어떠한 방법으로 관리할 것인가에 대한 정의가 필요하게 된다. 본 논문에서는 웹에이전트의 관리자 및 관리 방법, 웹서버의 관리자가 웹에이전트를 관리하기 위해 필요한 API, 시스템의 구성에 대하여 기술하였다.

### 1. 서론

지난 연구에서 웹을 기반으로 한 이동에이전트 시스템을 설계 하였다[1]. 이전의 연구에서 설계된 시스템은 자바의 서블릿과 유사한 형태의 자바 프로그램이 웹서버 사이클을 왕래하면서 필요한 작업을 수행한다는 아이디어를 기초로 한다. 기존의 시스템 설계는 이동에이전트의 실행환경과 이를 프로그램하기 위한 API설계에 중점을 두고 있다.

웹서버 사이클을 왕래하며 실행되는 이동에이전트는 익명성, 웹이라는 실행환경등 기존의 이동에이전트와 다른 특징들을 가지기 때문에 본 논문에서는 이를 구분하기 위해 웹에이전트라고 명명한다. 웹에이전트가 미리 예측하기 어려운 이유로 오동작을 하게 되거나 혹은 무한히 인터넷상을 떠돌아 다니게 되는 것과 같은 문제가 발생할 수도 있을 것이다. 따라서 이를 제어하기 위한 방법이 필요하게 된다.

본 논문에서는 웹에이전트의 관리를 위한 방법을 기술한다. 웹의 성격상 익명의 사용자가 웹에이전트의 적재 및 실행을 초기화하게 된다. 따라서 웹에이전트의 관리자가 누구이며, 어떠한 방법으로 관리할 것인가에 대한 정의가 필요하게 된다. 또한 웹에이전트의 관리자가 웹에이전트를 관리하기 위해 필요한 시스템을 구축하도록 하기 위한 API 및 시스템의 구성에 대하여 기술하였다.

### 2. 기존의 연구

기존의 연구에서는 웹상의 이동에이전트의 모델을 정의하기 위해서 실행가능한 프로그램 코드를 EU(Executing Unit)로, 프로그램 코드의 실행환경을 CE(Computational Environment)로 정의하였다[2]. 특별히, EU가 적재되어 실행이 초기화되는 CE를 HCE(Home Computational Environment)로 정의한다. 이 모델에서 EU는 웹에이전트(WA: Web Agent)에 해당하며, HCE와 CE는 웹서버에 통합된 실행환경이다. 따라서 웹에이전트는 웹서버 사이클을 이동하며 실행되는 자바 서블릿이라고 설명할 수 있다.

이러한 모델을 근거로 프로그래머가 웹상의 이동에이전트를 프로그래밍할 수 있도록 다음과 같은 연구를 수행하였다.

- ① 웹에이전트의 프로그래밍 모델 제안
- ② Jigsaw, Aglet을 참고로한 시스템 설계
- ③ 웹에이전트의 생성, 이동을 위한 API 설계
- ④ 웹에이전트간의 통신을 위한 메시징 시스템 및 API 설계

웹에이전트는 제공된 자바 API를 사용하여 프로그래밍하며, 웹서버상에 통합된 CE에서 실행된다. 기존의 연구는 서로 다른 패러다임을 가지고 있는 이동에이전트시스템과 클라이언트/서버 시스템을 결합한 응용 프로그램을 쉽게 구축할 수 있도록 웹에이전트의 프로그래밍을 위한 API와 실행환경을 제공하는 것이었다.

기존의 연구에서 W3C에서 제공하는 Jigsaw[3], IBM

Aglets[4]의 공개된 자바 소스를 참고로 하였는데, IBM Aglets의 경우 타히타라는 이동에이전트 관리용 프로그램을 제공하여 이동에이전트의 적재, 제거등의 관리가 가능하도록 하고 있다.

### 3. 웹에이전트의 관리

#### 3.1 웹에이전트의 의미성

웹에이전트는 익명의 사용자가 웹브라우저에서 실행을 초기화한다. 이는 웹서버상의 자바 서블릿을 익명의 사용자가 사용하게 되는 것과 같다. 자바 서블릿의 경우 웹서버의 관리자가 관리를 하게 되겠지만 자바 서블릿의 사용자에게 대한 인증은 관리자의 설정이나, 응용 프로그램 수준에서 해결해야할 문제이다. 웹에이전트의 실행환경이 되는 웹서버도 자신의 시스템에 어떤 웹에이전트가 수신되어 실행될 지 미리 예측하기 불가능하다는 가정을 할 수 있다. 자바 애플릿이 홈페이지를 방문하는 임의의 사용자 웹브라우저에서 실행되는 것처럼 웹서버를 익명성을 허락하는 웹에이전트 실행환경으로 보기 때문이다.

#### 3.2 웹에이전트의 관리자

웹에이전트의 관리에 관련된 사람은 다음과 같다.

- ① 홈페이지를 방문하여 웹에이전트를 실행시키는 사용자
- ② 웹에이전트의 프로그래머
- ③ 웹에이전트가 저장된 웹서버(HCE)의 관리자
- ④ 웹에이전트가 이동하여 실행중인 웹서버(CE)의 관리자

이들중 웹에이전트의 프로그래머는 웹에이전트의 운영에는 관계하지 않는다. 또한 홈페이지를 방문한 사용자의 경우 웹에이전트의 존재조차 모른다는 가정을 할 수 있다. 결국 웹에이전트의 실행에 대한 감시는 HCE, CE가 되는 웹서버의 관리자가 수행해야 할 것이다. HCE의 관리자는 웹에이전트의 주인으로서의 권한이 있으며, CE의 관리자는 웹에이전트에게 실행환경과 자원을 제공하는 입장에서 관리권한을 가진다.

#### 3.3 웹에이전트의 관리 기능

웹에이전트의 원활한 수행을 위해서 HCE와 CE의 관리자가 수행하는 많은 기능을 고려할 수 있다. 웹에이전트에 대한 자원 할당, 정보 획득과 같은 것들을 비롯하여, HCE와 CE 관리자간의 협조 방안도 고려할 수 있을 것이다.

이동에이전트는 실행과 이동을 자치적으로 결정한다는 개념을 가지고 있으며, 웹에이전트에 있어서도 마찬가지이다. 그러나, 문제가 발생했을 때, HCE와 CE의 관리자가 해당 웹에이전트를 제거하기 위해 필요한 최소한의 기능이 제공되어야 할 것이다. 이를 위해서는 아래의 세가지 기능이 관리자들에게 제공되어야 한다.

- ① Kill: 웹에이전트의 강제 종료
- ② List: 웹에이전트의 목록 표시
- ③ Check: 웹에이전트의 상태를 표시

본 논문에서는 이 세가지 기능을 API로 제공하여 웹서버의 관리자가 웹에이전트 관리용 사용자인터페이스를 웹에이전트로 프로그램하여 구축할 수 있도록 하였다.

## 4. 시스템 구성

### 4.1 웹에이전트의 관리를 위한 API

HCE와 CE가 되는 웹서버의 관리자는 먼저 자바 프로그램이나 웹페이지로 원하는 형태의 웹에이전트 관리를 위한 사용자 인터페이스를 구축한다. 이 사용자인터페이스에서는 웹에이전트 관리를 위한 API를 사용하여 프로그램된 웹에이전트를 실행하여 웹에이전트 시스템과 통신한다. 사용자인터페이스와 관리를 위한 웹에이전트는 웹서버의 관리자가 구축한다(그림 1).

웹에이전트의 관리를 위해 필요한 API는 HCE와 CE의 관리를 위한 것이 각각 제공된다(그림 2). 이들 API를 사용하여 관리용 웹에이전트를 고정에이전트로 프로그램했을 때, 이 관리용 웹에이전트는 고정된 위치의 웹서버를 HCE 혹은 CE로 하는 웹에이전트에 대한 정보를 표시하고 제거하는 일을 하게 된다.

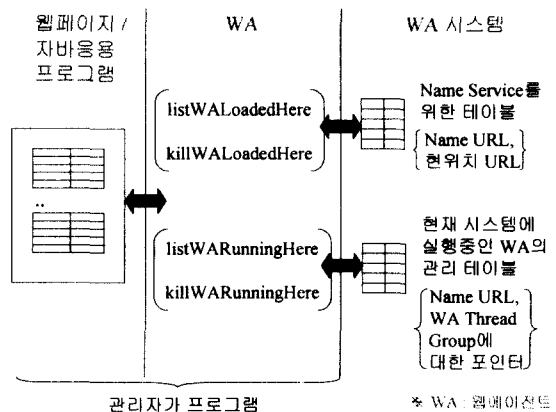


그림 1 API를 사용한 웹에이전트 관리

### 4.2 시스템 동작 방법

killWALoadedHere()를 사용할 때는 웹에이전트 시스템의 네임서비스서브시스템의 웹에이전트 관리 테이블을 참조하게 된다. 이 테이블에는 웹에이전트의 이름과 현재 위치가 URL로 보관되어 있는데, 이를 참고로 하여 웹에이전트의 현재 실행위

```

• HCE관리자용
- public static URL[] listWALoadedHere()
- public static void killWALoadedHere(URL url)
- public static boolean checkWALoadedHere(URL url)

• CE관리자용
- public static URL[] listWARunningHere()
- public static void killWARunningHere(URL url)
- public static boolean checkWARunningHere(URL url)
    
```

그림 2 웹에이전트 관리용 API

치를 알아낸다[1]. 동작 과정은 다음과 같다(그림 3).

- ① 웹에이전트를 HCE측 네임 서버의 웹에이전트 관리 테이블에서 삭제
- ② CE측에 KILL 메시지를 송신하여 웹에이전트를 삭제하도록 요청
- ③ CE측에서 웹에이전트를 제거하고 CE측의 실행 관리 테이블을 갱신

문제가 발생한 웹에이전트를 삭제할 때, 해당 웹에이전트가 반응하지 않을 수도 있다. 따라서 위의 과정에서 CE측에 KILL 메시지를 송신하더라도 CE측의 웹에이전트시스템이 이를 적절히 처리하지 못할 수도 있다. 이러한 경우는 CE측의 관리자가 해결할 수 밖에 없다. 그러나 웹에이전트의 이동은 HCE의 네임서비스서브시스템에 등록하도록 되어 있으므로 제거하고자 하는 웹에이전트는 바로 네임서비스의 웹에이전트 관리 테이블에서 삭제한다. 이렇게 하므로써 제거하고자 하는 웹에이전트의 이동을 막을 수 있다.

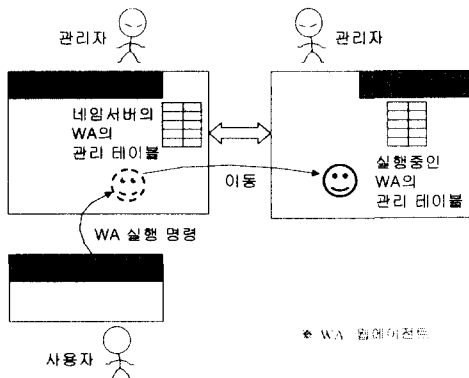


그림 3 시스템 동작 방법

웹에이전트의 실질적인 삭제는 결국 CE측에서 이루어 진다. CE는 실행중인 웹에이전트들을 URL로 되어 있는 웹에이전트의 이름과 쓰레드그룹에 대한 포인터의 쌍으로 되어 있는 데이

블로 관리한다. killWARunningHere()를 사용할 경우의 동작 방법은 다음과 같다(그림 3).

- ① CE측에서 웹에이전트를 제거하고 CE측의 실행중인 웹에이전트 관리 테이블을 갱신
- ② HCE측에 웹에이전트의 제거 사실을 통보
- ③ 웹에이전트를 HCE의 네임서비스서브시스템의 웹에이전트관리테이블에서 삭제

통신상의 문제로 인하여 HCE측에 웹에이전트가 제거되었다는 사실이 제대로 전달되지 않았을 경우 HCE의 관리테이블에는 실제로 존재하지 않는 웹에이전트에 대한 참조가 남아 있게 된다. 혹은 반대로 HCE의 관리테이블에서 삭제하고 CE측에 삭제를 요청한 웹에이전트가 실제로 CE측에 존재할 경우도 가능하다. 이 경우 관리자는 checkWALoadedHere()와 checkWARunningHere() 두가지 API를 사용하여 웹에이전트의 상태를 검사할 수 있도록 하였다. 이 두 개의 API는 각각 HCE측과 CE측의 관리 테이블에 지정한 웹에이전트가 정상적으로 관리되고 있는지를 확인하여 적절한 조치를 취할 수 있도록 하였다.

5. 결론

웹에이전트는 익명으로 실행되며, 웹서버는 웹에이전트의 실행환경으로 개방된다는 가정하에 시스템이 구축되었다. 그러나, 웹에이전트가 실행되면서 발생할 수 있는 문제에 대한 최소한의 대책이 필요하였다. 본 논문에서는 웹에이전트의 관리자를 설정하고, 최소한의 관리 방법으로 실행중인 웹에이전트를 제거할 수 있는 기능을 구현하도록 API를 제공하였다.

연구를 진행하면서, 단순히 실행중인 웹에이전트의 중단이나 제거 외에도, HCE와 CE의 관리자가 웹에이전트에 대한 자원 할당을 위해 협조하는 방법등 연구할 만한 주제들이 많다는 것을 알 수 있었다. 또한 웹서버를 웹에이전트를 위해 개방된 실행환경으로 보는데 있어 보안 모델이 보완되어야 할 필요가 있다고 생각한다.

참고 문헌

[1] 송성훈, 원유현, "웹을 중심으로한 이동코드의 메시징시스템 설계", 정보과학회 학술 발표 논문집(A) 2001.4 p.64  
 [2] G. Cugola, C. Ghezzi, G.P. Picco and G. Vigna, 1996, "Analyzing Mobile Code Languages", 2nd Int. Workshop on Mobile Object System; also in LNCS 1222, pp. 93-109  
 [3] <http://www.w3.org/jigsaw>  
 [4] <http://www.trl.ibm.co.jp/aglets/index.html>