지능형 에이전트를 이용한 트리거 에이전트의 구현

Jang-uk Cheon, Dong-un An, Seung-jong Chung, Gil-su Doo, Hoo-byong Yun

Beob-kyun Kim, Young-sun Her

Dept. of Computer Engineering , Chonbuk National University

Design and Implementation of Trigger Agent Using Intelligent Agent

천 장 욱, 정 성 종, 안 동 언, *두 길 수, 윤 후 병, 김 법 균, 황 호 전, 허 영 선

전북대학교 컴퓨터공학과, *서남대학교

전화 : 063-270-2412 / 핸드폰 : 018-394-3466

Abstract

There are over a billion web sites on the Internet and the number of web sites is still growing.

This paper describes an information system. We have designed Trigger Agent System based on RMI (Remote Method invocation) and distribution object system.

In this paper, we introduce Trigger agent system which is one of Intelligent agent system.

Keyword : Intelligent Agent, Trigger Agent, IR

Ⅰ. 서론

인터넷 기술의 발전과 더불어 정보양이 급증하고 있으며 정보 환경의 변화는 정보를 풍족하게 하기도 하지만 다른 한편으로는 이처럼 많은 정보의 홍수 속에서 양질의 정보만을 선출해 내는 부담을 안겨 주고 있다. 게다가 정보의 형태 또한 다양해지고 복잡해지면서 사용자가 원하는 정보를 정확하고 신속하게 알아내기란 더욱 어려워져 정보검색의 중요성이 더욱 강조되고 있다. 더불어 소프트웨어 개발개념 중 근래에 가장 많은 주목을 받고 있는 것이 객체지향 이다. C와 같은 순차언어에서 벗어난 객체지향언어는 각 프로그램이 객체로 구성되어 있고 그 객체간 메시지 전송으로 실행된다. 정보검색과 객체지향 프로그램을 이용하여 새로운 정보검색의 패러다임인 트리거 에이전트 정보검색 시스템을 제안하고자 한다.

먼저 지능형 에이전트는 지능을 가지고 있으며 학습하고 서로 간의 통신을 통해서 대화를 주고 받으며 주어진 환경을 감시하고 그 변화에 적응하는 능력을 가진 객체로 정의한다. 이기종 간 플랫폼의 제한을 가지고 있어서는 안 된다. 따라서 이기종 플랫폼에서 자유로운 분산 개념을 도입 함으로서 에이전트 발전에 많은 연구 분야를 형성하게 되었다.

본 논문에서는 이런 정보검색과 에이전트, 분산객체를

접목하여 새로운 패러다임을 제안하고자 한다 구성은 다음과 같다. 2장에서는 연구배경 및 고려사항에 대해 알아보고 3장에서는 본 논문에서 제시하고자 하는 트리거(Trigger) 에이전트를 소개하고 4장에서는 결론 및 향후 연구 과제를 제시한다.

Ⅱ. 연구배경 및 고려 사항

에이전트

지난 반세기동안 에이전트[1][2]의 동향을 살펴보면 50년대 초창기의 인공지능에서 70년대 후반 분산인공지능 (Distributed Artificial Intelligence)이라는 새로운 연구 분야가 형성되어 다양한 종류의 분산 인공지능 문제들이 에이전트라는 추상적 단위와 에이전트간의 상호작용을 토대로 해결하려는 연구가 시작되었다. 90년대 들어서 웹(Web)과 인터넷(Internet)의 급속한 확장으로 웹 환경에 맞는 소프트웨어 에이전트의 출현을 맞게 된다. 이런 에이전트의 종류로 대행자(Proxy)와 중개자(middleware)역할을 수행하도록 발전했다. 이처럼 90년대 들어 많은 연구가 이루어지고 있는데 현재는 에이전트가 다른 여러 응용 분야와 접목되어 사용되는 추세이다.

RMI(Remote Method invocation)

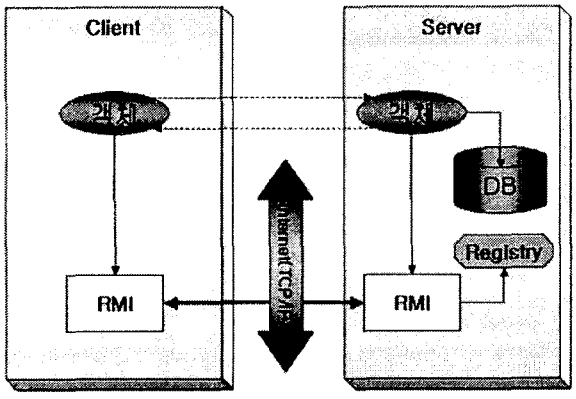

그림 1 RMI와 객체 구조도

RMI[5]를 사용하는 목적은 객체가 어느곳에 존재하는가에 관계없이, 객체의 메소드를 사용하여 통신할 수 있다고 하는 것이다. 다시말하면 객체들이 서버에 있던지 클라이언트가 수행되는 로컬 컴퓨터에 있던지 간에 그것들이 동일한 런타임 환경에서 수행되는 것처럼 해주는 의미이다. 그림 1은 RMI의 구조도이다.

분산시스템

소프트웨어 개발에서 가장 주목받고 있는 것이 객체지향 개발[5]이라면 시스템분야에서 가장 주목받는 것은 분산시스템의 개념이다. 어쩌면 이 분산시스템의 개념이 객체지향 개발을 절대적으로 필요하게 했을지도 모른다. 분산시스템이란 기본적으로 다수의 컴퓨터가 하나의 시스템처럼 작동하며 사용자에게 그 차이를 느끼지 못하게 하는 시스템을 말한다. 분산시스템의 중요한 개념은 명령실행의 투명성 (transparency), 효율성 (efficiency), 일관성 (consistency), 유연성 (flexibility), 그리고 robustness 등을 들 수 있는데 그 중 가장 많이 강조하는 것이 투명성이다. 즉, 사용자가 어떤 명령을 실행시켰을 경우 그 명령이 지금 사용자가 사용하고 있는 컴퓨터에서 실행되고 있는지, 아니면 네트워크에 연결된 다른 컴퓨터에서 실행되고 있는지 분간할 수 없음을 말한다. 대표적인 분산시스템으로 UNIX의 NFS (Network File System), AFS (Andrew File System), CODA 등이 있다.

Ⅲ 트리거 에이전트의 설계

3.1 트리거 에이전트의 필요성

트리거는 DB가 갱신되었음을 알려주는 시그널 이름이다. 에이전트 이름을 트리거라 한 이유도 이점과 유사하기 때문이며, 정보검색능력을 가진 분산객체 에이전트들의 패키지를 트리거 에이전트라 정의한다. 현재 사용되는 많은 정보검색 시스템 중 대부분의 시스템이 로봇을 사용한다. 하지만 기하급수적으로 불어나는 정보의 양을 수집하기란 로봇의 수동성으로는 쉬운 일이 아니다. 따라서 능동적인 웹 문서 관리가 필요하다. 이점에 착안하여 트리거 에이전트의 능동적으로 클라이언트 중심의 웹 문서 관리를

시도한다. [그림 1]은 트리거 에이전트의 기본 구조를 보이고 있다.

3.2 트리거 에이전트 구성 시스템

표 1 분산객체간 통신 매개변수

Name	Type	기능
live	boolean	클라이언트의 동작여부
request	int	request
modify	boolean	문서의 갱신여부
last_modify	Date	마지막 갱신시간

Define	변수값	기능
Request_live	1	Live 정보 요구
Request_modify	2	Modify 정보 요구
Request_last_modify	3	Last_modify 정보 요구

표 2 request definition

트리거 에이전트를 구성하는 시스템은 기본적으로 다음 세 가지의 모델을 사용한다.

(1) 자바 Virtual Machine

하나의 프로그램으로 이 기종 시스템을 구동하기 위해서 자바 Virtual Machine을 사용한다.

(2) RMI(Remote Method Invocation)

RMI시스템은 자바 가상머신에서 동작하는 분산객체로 보안과 투명성을 제공한다.

(3)Web 문서 랩핑 에이전트 시스템[4]

레거시시스템(Legacy System)을 에이전트 시스템으로 환원해주며, Web 문서 갱신 때 문서 수집에 관여한다.

3.3 Trigger Agent 시스템 아키텍처

트리거 에이전트의 아키텍처는 크게 서버 에이전트와 클라이언트 에이전트로 구분 할 수 있다.

1)서버 에이전트 시스템 수집

- 문서처리 : 수집된 문서에서 색인과 요약 하는 단계
- *IR DB* : 정보검색 *DB*
- 관리 에이전트 : Server Management Agent

2)클라이언트 에이전트 시스템

- 문서 : Web 문서
- 래핑 에이전트 : 문서를 모니터링 하기위해 래핑
- 모니터링 에이전트 : 래핑된 문서를 모니터

3) 통신 프로토콜

- RMI (IIOP)

표 3 request definition

에이전트간 매개변수와 기능에 대하여 [표 1]에서 설명하고 있다. 총 4 개의 매개변수를 가진다. 날짜 변수는 마지막 갱신일이며 변수형 Date 이다. live 와 modify 는 boolean 변수로 선언하였다. request 1 과 request 2 일때 각각 live 와 modify 메시지를 요구하며, request 가 3 일때 request_lastmodify 를 요구한다.

Ⅳ. 시나리오

시나리오: 웹 서버에 트리거 에이전트를 설치하며, 일정시간이 지난 후에 서버의 문서의 갱신이 발생한다.

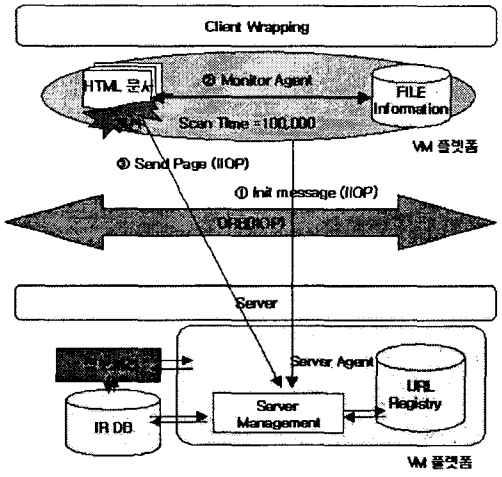

그림 2 트리거 에이전트 시스템

(1) 웹 문서를 가진 컴퓨터에 트리거 에이전트를 설치하면 서버 에이전트에 메시지를 전달하고 서버의 Registry에 클라이언트의 URL을 등록 한다.

(2)여기서는 100 초마다 문서의 갱신여부를 클라이언트에 있는 래핑 에이전트가 모니터한다. 또한 서버에서 Request_live (request = 1) 메시지를 받으면 에이전트는 live 메시지를 전달한다.

(3) 문서의 수정이 발생하면 Send Message 이벤트 발생 서버에 modify 메시지와 갱신된 문서를 서버에 보낸다.

(4) 문서정보 DB에 새로운 문서 정보를 보관한다.

(5) 다시 (2)를 반복한다.

그림 3은 트리거 에이전트를 두대의 컴퓨터로 트리거 에이전트를 구현 결과를 보여주는 화면이다.

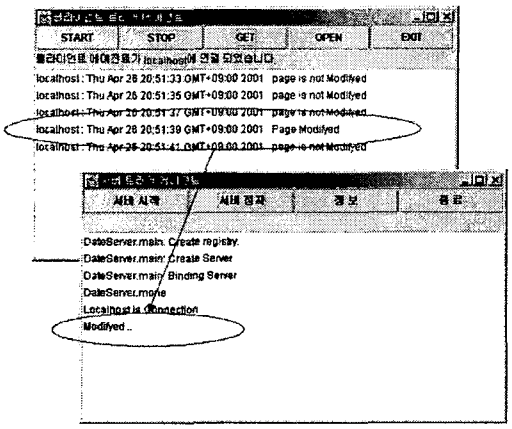

그림 3 구현

V. 결론

트리거 에이전트의 사용 함으로서 다음과 같은 장점을 제공한다. 첫째 망의 과부하를 줄일 수 있다. 기존 서버 중심의 ROBOT 에이전트는 망의 과부하는 신경 쓰지 않고 문서를 가져오는 부분에만 중심적이었는데, 트리거 에이전트는 클라이언트 중심으로 서버에 통보 방식을 따르고 있어 갱신될 때에만 통신이 이루어진다. 둘째 웹 문서의 즉각적인 갱신을 추적할 수 있다. 클라이언트 쪽에 데몬은 주기적으로 계속 갱신을 검색하고 있으므로 갱신되는 순간 바로 문서의 변화를 감지할 수 있다.

셋째 분류된 웹 문서는 각각의 서버를 두어 따로 관리할 수 있다. 이것은 분산객체에서 제공하는 장점으로 분산된 웹 문서 관리가 가능하여 지역적이고 전문화된 문서 관리를 할 수 있다.

실제 프로토타입 시스템으로 구현한 트리거 에이전트로 웹 문서를 모니터 할 수 있었다. 향후 연구 과제로는 시스템 자원관리와 트리거 에이전트를 확장하여 로봇의 사용을 배제하는 것이다. 따라서 트리거 에이전트의 몇 가지 단점을 보완 한다면 망의 부담을 줄이고, 빠르며 정확하게 사용자가 원하는 가장 최신 정보를 받아볼 수 있을 것이다.

참고문헌

[1].Intelligent Information Agents: Review and Challenges for Distributed Information Sources: D.S. HaverKamp and S.Gauch

[2]. Mobile Agent for Distributed Information Retrieval : Rrian Brewington, Robert Gray, Katsuhiro Moizumi, David kotz,

[3]. From Bussiness Processes to Cooperative Information System: An Information Agents Perspective

[4]. Adaptive Choice of Information Sources : Sandip Sen, Anish Biswas , and Sumit Ghosh

[5]. Java Network Programming : Merlin Hughes, Michael Shoffner, Derek Hamner, Umesh Bellur