

Trigger Agent의 설계 및 구현

전장욱⁰, 안동언, 정성중
전북대학교 컴퓨터공학과
juchon@calhpl.chonbuk.ac.kr, {duan, sichung}@moak.chonbuk.ac.kr

A Design and Implementation of Trigger Agent

Jang-uk Cheon⁰, Dong-un An, Sung-jung Jung
Dept. of Computer Engineering, Chonbuk National University

요 약

인터넷과 웹의 발전으로 인한 정보량이 급격한 증가로 정보를 수집 하는데 있어 많은 어려움에 부딪치게 되었다. 근래 이런 복잡한 문제를 해결하고자 하는 노력이 에이전트 연구와 접목되어 시도되고 있다. 정보량의 급증으로 인해 로봇을 사용하는 시스템들은 로테이션 타임이 길어지고, 네트워크 망의 과부하를 급증시키고 있다. 이런 문제를 해결하고자 트리거(Trigger) 에이전트를 제안한다. 트리거 에이전트는 웹 서버에 에이전트를 래핑 하여 문서를 모니터링 하게 하고 문서의 갱신이 발생하면 이때 서버에 로봇을 요청하여 문서를 수집하는 아키텍처를 가진다.

1. 서론

컴퓨터와 통신 기술의 발전은 인터넷이라는 거대한 컴퓨터 망을 만들어 놓았다. 이런 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 어려워 졌다. 반면 정보검색의 필요성은 더욱 증가하여 인터넷과 웹에서 중요한 이슈가 되었다. 정보검색의 중요성이 가중되면서 패러다임의 변화를 요구하며, 이에 지능형 에이전트(Intelligent agents)는 의미 있는 한 가지 방법을 제시하고 있다.

지능을 가지고 있으며 학습하고 서로 간의 통신을 통해서 대화를 주고받으며 주어진 환경을 감시하고 그 변화에 적응하는 능력을 가진 객체를 지능형 에이전트라 정의한다. 에이전트는 기기종 간 플랫폼의 제한을 가지고 있어서는 안 된다. 따라서 기기종 플랫폼에서 자유로운 분산 개념을 도입함으로써 에이전트 발전에 많은 연구 분야를 형성하게 되었다.

본 과제는 전북대학교 정보검색 시스템 연구센터의 "정보검색 시스템 설계" 연구의 일부분입니다.

본 논문에서는 이런 정보검색과 에이전트, 분산객체를 접목하여 새로운 패러다임을 제안하고자 한다

본 논문의 구성은 다음과 같다. 2장에서는 연구배경 및 고려사항에 대해 알아보고 3장에서는 본 논문에서 제시하고자하는 트리거(Trigger) 에이전트를 소개하고 4장에서는 결론 및 향후 연구 과제를 제시한다.

2. 연구배경 및 고려 사항

2.1 에이전트

지난 반세기동안 에이전트의 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.

2.2 분산객체

분산 객체는 다른 메모리 도메인에 존재하여 접근하

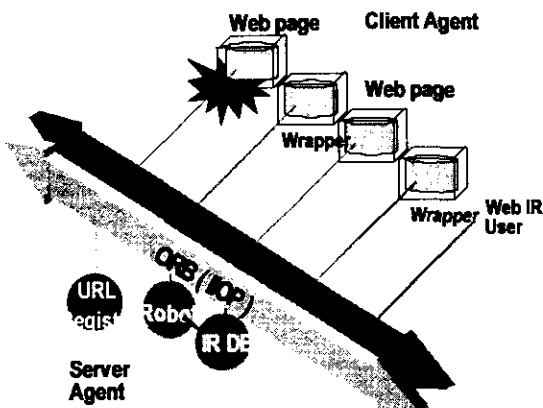
기 위해 네트워크를 이용하는 것이 필요한 객체를 말한다. 이것은 일반 객체와 마찬가지로 데이터와 메서드로 이루어져 있다. 분산 객체 기술은 다양한 환경에서 분산 객체를 사용할 수 있게 해주는 기술이다. 플랫폼과 프로그램 언어에 대한 독립성을 제공한다. 이 특징과 더불어 객체지향 기술은 분산 객체 기술을 앞으로 더 널리 쓰이게 하는 중요한 요인이 된다. 분산 객체의 배경 기술로는 객체 지향 기술, 미들웨어 기술 등이 있으며, 현재 대표적인 분산 객체 시스템은 CORBA와 DCOM 그리고 RMI 등을 이용하고 있다. 이런 시스템은 각각의 영역을 확보하고 있으므로, 시스템을 개발하는데 있어 어떤 분산환경을 선택하는가 또한 중요하다. 현재 분산객체는 OMG에서 표준을 만들어 많은 가입회원 단체가 이를 따르고 있다.

2.3 정보검색

얼마 전까지만 해도 정보검색은 지역도서관에 가서 도서관 직원에게 도움을 요청하는 것을 의미 했다. 도서관 직원은 장서를 모두 알고 있으며, 가끔 부정확한 대답을 할 때도 있지만, 대부분 필요한 도서에 대한 정보를 알려 주었다. 하지만 정보(도서)의 양의 증가는 도구를 필요로 했고, 이런 도구에 있어 가장 중요한 것이 색인이다. 색인은 제목, 저자명, 청구번호 등으로 구성할 수 있지만 색인의 구조는 근본적으로 동일하다. 현대의 정보검색은 자동화된 색인 목록에 더욱 복잡한 구조를 요구한다. 이처럼 색인의 자동화는 에이전트를 필요로 하며 본 논문에서 제안한 트리거 에이전트가 그중 하나이다.

3. 트리거 에이전트의 설계

3.1 트리거 에이전트의 필요성



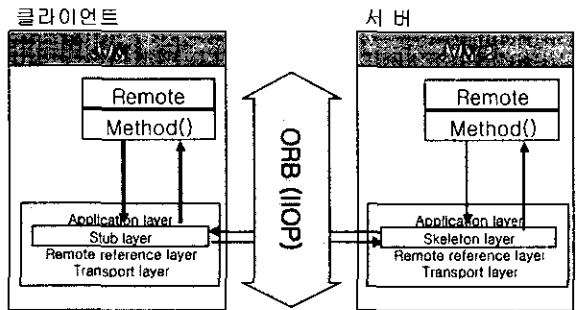
트리거는 DB가 갱신을 알려주는 사그널 이름이다. 에이전트 이름을 트리거라 한 이유도 이점과 유사하기 때문이며, 정보검색능력을 가진 분산객체 에이전트들의 패키지를 트리거 에이전트라 정의한다. 현재 사용되는 많은 정보검색 시스템중 대부분의 시스템이 로봇을 사용한다. 하지만 기하급수적으로 불어나는 정보의 양을 수집하기란 로봇의 수동성으로는 쉬운 일이 아니다. 따라서

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

트리거 에이전트를 구성하는 시스템은 기본적으로 다음 세 가지의 모델을 사용한다.

- (1) 자바 Virtual Machine
하나의 프로그램으로 이 기존 시스템을 구동하기 위해서 자바 Virtual Machine을 사용한다.
- (2) RMI(Remote Method Invocation)
RMI(Remote Method Invocation)시스템은 자바 가상 머신에서 동작하는 분산 객체를 지원한다.
- (3) HTTP 문서를 가져 올수 있는 로봇
기존에 이미 만들어진 로봇을 Wrapping 해서 사용한다.



[그림 2] RMI 계층 구조도

기존의 수동적인 웹 페이지 관리 수준에서 벗어나 능동적인 웹 문서관리를 수행 할 수 있는 트리거 에이전트는 위 세가지 모델을 기반으로 구현 되며 그림 2는 RMI 계층 구조도를 보여주며 서버와 클라이언트간 투명한 통신과 보안을 제공한다.

3.3 Trigger Agent 시스템 아키텍처

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

- 1)서버 에이전트 시스템 수집
 - 로봇 : 웹 문서를 수집
 - 문서처리 : 수집된 문서에서 색인과 요약 하는 단계
 - IR DB : 정보검색 DB
 - 관리 에이전트 : Server Management Agent
 - 2)클라이언트 에이전트 시스템
 - 문서 : 웹 문서
 - 문서 래핑 : 문서를 모니터링 하기위해 래핑
 - 모니터링 에이전트 : 래핑된 문서를 모니터
 - 3) 통신 프로토콜
 - ORB (IOP)
- 에이전트간 매개변수와 기능에 대하여 [표 1]에서 설

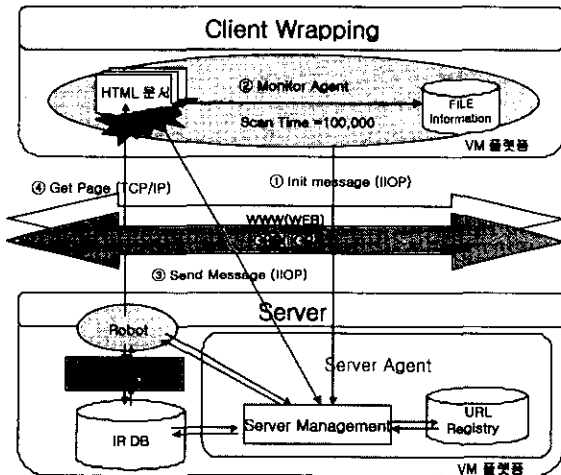
명하고 있다. 총 4개의 매개변수를 가진다. 날짜 변수는 마지막 갱신일이며 변수형 Date이다. live 와 modify 는 boolean 변수로 선언하였다. request 1과 request 2 일때 각각 live 와 modify 메시지를 요구하며, request 가 3일때 request_lastmodify를 요구한다. [표 2]

name	type	기능
live	boolean	클라이언트의 동작 여부
request	int	request
modify	boolean	문서의 갱신 여부
last_modify	Date	마지막 갱신시간

[표 1] 매개변수 타입과 정의

define	변수 값	기능
request_live	1	live 정보 요구
request_modify	2	modify 정보 요구
request_last_modify	3	last_modify 정보 요구

[표 2] request define



[그림 3] 트리거 에이전트 기능 구조도

시나리오를 가지고 트리거 에이전트의 기능들을 살펴 보자.

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

- (1) 웹 문서를 가진 컴퓨터에 트리거 에이전트를 설치하면 서버 에이전트에 메시지를 전달하고 서버의 Registry에 클라이언트의 URL을 등록한다.
- (2) 여기서는 100초마다 문서의 갱신여부를 클라이언트에 있는 래핑 에이전트가 모니터한다. 또한 장시간 서버와 클라이언트 통신이 이루어지지 않고 있다면 서버에서 Request_live(request = 1) 메시지를 보내서 클라이언트 에이전트의 작동 여부를 감시한다.
- (3) 문서의 수정이 발생하면 Send Message 이벤트 발

생 서버에 modify 메시지를 보낸다.

- (4) 서버의 로봇이 문서를 수집한다.
- (5) 문서정보 DB에 새로운 문서 정보를 보관한다.
- (6) 다시 (2)를 반복한다.

4. 트리거 에이전트 구현

플랫폼을 설계할 때 고려사항으로는 첫째 기존 시스템 위한 확장성 독립성을 제공해야 한다. 따라서 에이전트 래핑(Wrapping) 기술을 도입했다. 둘째로 구현환경에서 이기종간 자유로운 이식성을 제공해야 한다. 플랫폼 독립적이어야 한다. 셋째 인증된 에이전트는 자유로운 활동을 보장할 수 있는 기반구조와 신뢰성을 보장해야 한다. 이런 점에서 OMG의 표준인 RMI를 사용하여 구현하였다.

5. 기대효과 및 결론

트리거 에이전트의 사용함으로써 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, 빠르게 정확하게 사용자가 원하는 가장 최신 정보를 받아볼 수 있을 것이다.

5. 참고 문헌

- [1]. Intelligent Information Agents: Review and Challenges for Distributed Information Sources: D.S. Haverkamp and S.Gauch
- [2]. Mobile Agent for Distributed Information Retrieval : Brian Brewington, Robert Gray, Katsuhiko 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]. Mobile Agent Security: Christian f. Tschudin