

효과적 이동성 확장을 위한 URN 적용 이동 에이전트 언어의 설계

강동남⁰ 김성근 유재우

송실대학교 컴퓨터 학부

{lovehis, skkim}@ss.soongsil.ac.kr cwyoo@comp.soongsil.ac.kr

A Design of the Mobile Agent Language using URN for Effective Mobility Extension

Dong-Nam Kang⁰ Sung-Keun Kim Chae-Woo Yoo
School of computing, Soong-Sil University

요약

에이전트의 객체가 사용자의 요구에 맞는 에이전트 시스템으로 이동하여 사용자의 요구를 충족시키기 위한 이동성은 이동 에이전트 시스템에서 가장 기본적이고도 중요한 문제이다. 이동성을 증가시키기 위해 URL을 이용한 많은 연구가 이루어져 왔다. 하지만 URL은 위치 의존성을 가지고 있기 때문에 이동 하고자 하는 에이전트 시스템의 위치가 달라지면 에이전트의 이동이 불가능해지고 가지고 있는 URL정보는 아무 쓸모가 없어진다는 문제를 가지고 있다. 이에 본 논문에서는 기존의 URL을 이용한 이동성 구현의 문제점을 보완하고자, URN 그리고 URC 데이터 베이스를 이용한 에이전트 프로그램 언어를 설계하려 한다. 또한 에이전트 이동을 위하여 URN을 URC 데이터 베이스를 이용하여 URL로 매핑 시키는 과정에 있어서는 Resource Monitor이라는 개념을 이용하여 좀더 효과적인 방법을 제시한다. 이러한 이동성이 극대화된 에이전트는 사용자에게 보다 나은 서비스를 제공할 것으로 기대된다.

1. 서론

에이전트란 원래 단순히 사용자가 하는 반복 작업이나 복잡한 일 등을 대신 수행해 주는 형태로, 각각의 에이전트마다 고유하고 특정한 역할이 주어진 특화된 에이전트 상태로 개발되어 지다. 같은 플랫폼 안에서 사용자의 즉흥적으로 표현해 내는 간단한 형태의 명령들을 실행 할 수 있는 플랫폼 의존적인 범용적인 에이전트 형태로 진화를 하였다. 그리고 요즘은 다시 네트워크 컴퓨팅이라는 새로운 환경의 변화를 맞이하여 에이전트는 다시 한번의 이동 에이전트로의 진화의 과정을 거치고 있는 상태이다. 이동 에이전트란 사용자의 명령을 받아 자율적으로 다른 플랫폼으로 자유롭게 이동하여 사용자의 요구에 맞는 일을 실행 할 수 있는 플랫폼 독립적인 프로그램을 말한다. 이런 이동 에이전트에서의 가장 큰 문제 중 하나는 에이전트의 효율적인 이동성을 달성하는 것이다. 이에 본 논문에서는 URN과 URC 데이터 베이스를 이용하여 이동성을 극대화한 에이전트 언어를 설계하고자 한다.

이 URN Resolver는 사용자의 요구에 맞는 에이전트의 목적지를 효율적으로 탐색하기 위하여 구현 되어 있는 것으로 에이전트 시스

템의 요구에 따라 그에 맞는 최적의 다른 에이전트 시스템의 URL을 찾아 주는 역할을 한다. URN은 기존의 위치 종속적인 URL과는 달리 고유한 이름으로 위치 정보를 기술하는 방법을 말한다. 또한, URC 서버에 자원 감시자(Resource Monitor)라는 개념을 도입하여 에이전트 시스템을 감시하고 각 에이전트 시스템은 현재 자원의 상태를 유지하고 있어 이용자의 요구에 맞는 에이전트의 이동 목적지를 효과적으로 구할 수 있다.

2. 이동 에이전트 언어

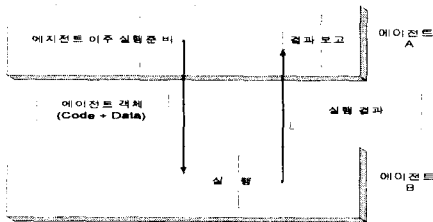
이동 에이전트는 분산시스템 환경에서 자율적으로 이동 사용자를 대신해서 어떤 일을 행할 수 있는 능력을 가진 실행 가능한 코드와 자료(data)를 포함한 객체를 말한다. [4] 에이전트 객체는 에이전트 언어형태로 기술될 수 있다. 에이전트 언어는 크게 Agent Specification Language(ASL)와 Agent Programming Language(APL)로 구분된다. APL은 초기에 Tcl, Perl과 같이 인터프리터를 통해서 해석되고 실행되는 스크립트 언어가 주된 연구의 대상이었지만, 스크립트 언어가 가지는 문법 오류 검사나 의미 분석을 할 수 없다는 단점을 극복하기 위해 JAVA와 같은

스크립트 언어의 성격을 가진 컴파일 언어의 형태로 발전해 가는 경향을 보이는데, 이는 이동 에이전트의 특징인 플랫폼 독립성을 유지하기 위해서는 특정 플랫폼에서 컴파일 된 코드가 아닌 어디서나 통용(실행)될 수 있는 코드로 표현되어야 하기 때문이다. 결과적으로 APL에 요구되어 지는 본질적인 속성은 첫째, 코드가 포함된 객체들을 조작, 전송, 수신, 그리고 실행을 지원해야 하고, 둘째 이 기종 시스템을 지원해야 한다. 또한 바람직한 에이전트 프로그램을 위해서 속성을 추가하면 확장 이동성(Extend Mobility), 원격 자원 접근(Remote resource access), 그리고 에이전트간의 통신 등의 APL의 속성에 추가 될 수 있다.

3. 이동 에이전트의 이동성

이동 에이전트에서 이동성이란 여러 다른 하드웨어 플랫폼으로 이동할 수 있는 능력을 말한다. 이동성은 크게 두 가지로 구별될 수 있는데, 약한 이동성을 가진 원격 실행(remote execution)방식 과 강한 이동성을 가진 객체 이주(Object migration)방식 이다. 본 논문에서 다루는 객체 이주 방식은 에이전트가 다른 위치로 이주해 현재의 실행을 계속하는 것이다. 객체 이주방식은 에이전트에 의해서 초기화되고 이주 명령으로 에이전트의 수행이 현재 위치에서 중단된 후(혹은, 에이전트의 이주 실행의 준비가 끝난 후) 다른 위치(다른 플랫폼)로 전송되어 실행된다.

에이전트이주 실행 시에는 실행 가능한 프로그램 코드, 인스턴트 변수의 내용, 실행상태(즉 data)등을 포함한 에이전트 객체가 필요하다. [그림 1]은 이동 에이전트의 객체 이주 방식의 메커니즘을 나타낸 것이다.



[그림 1] 객체 이주방식의 메커니즘

4. URN, URC 데이터 베이스

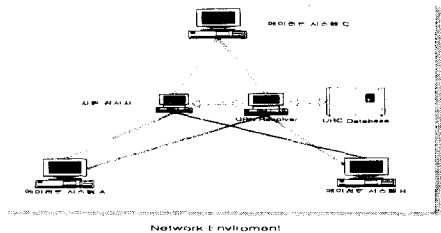
URN의 기본적인 사상은 어떤 정보를 어디에 있다는 기호로서가 아니라 그 정보 자체의 고유한 기호로 구별해자는 것이다. 또한 위치에 독립적이므로 여러 장소에 동일한 정보를 분산할 수 있어 백본(backbone)의 폭주를 줄일 수 있으므로 액세스 속도를 개선할 수

있다. URC는 인터넷에서 액세스할 수 있는 인터넷상의 정보원의 코드화 방법으로, 속성과 값의 집합이다. 정보는 원 정보에 없는 정보이기 때문에 메타정보라고 한다. 속성값으로서는 다양한 URL의 집합이 될 수도 있다. URC는 메타정보를 가지고 있는 하나 이상의 URN 혹은 URL을 가지고 있으며, 정보원에 대한 URN과 그 URL집합간을 연결하는 메타데이터의 컨테이너의 역할을 한다. 따라서 URC는 "URL 집합과 관련된 메타정보의 표현을 위한 전달수단이나 구조를 융통성 있게 제공하는 것"이라 할 수 있다.[2]

이와 같이 특성이 있는 URN과 URC를 에이전트 시스템에 적용시키면 에이전트가 요구하는 리소스(다른 에이전트)의 위치정보(URL)에 대하여 효과적인 식별과 검색이 가능하다.

또한 좀더 효과적인 검색을 위하여 사용자의 에이전트 시스템에서 URN Resolver로 보내진 URN을 URL으로 변환하는 과정에 자원 감시자의 개념을 추가할 수 있다. 자원 감시자 개념은 에이전트가 이동할 수 있는 URL의 에이전트 시스템 중에 현재 가장 여유 있는 에이전트 시스템을 탐색하기 위하여 만들어 졌다. 자원 감시자는 에이전트 시스템 서비스에 정의된 모든 종류의 자원에 대해 포괄적인 감시를 하여, 최적의 이동 위치정보 검색 시 요구되는 여러 가지 정보를 얻어낼 수 있다. 이런 URL를 얻기 위한 작업을 순서대로 살펴보면 다음과 같다.

- ① 사용자의 요구에 따라 한 에이전트 시스템에서 URN을 URN Resolver로 보낸다.
- ② URN Resolver는 URC 데이터 베이스를 검색하여 조건에 맞는 다른 에이전트의 URL을 얻어낸다.
- ③ URN Resolver는 얻어낸 URL을 가진 에이전트 시스템의 상태를 자원 감시자에게 문의한다.
- ④ 자원 감시자에게서 얻어온 각 에이전트 시스템의 상태를 참고하여 이중 최적이라 판단되는 에이전트 시스템의 URL을 작업이 요구한 에이전트 시스템으로 돌려준다.

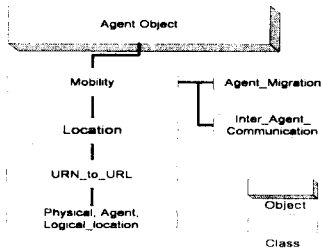


[그림 2] 에이전트 시스템의 구조도

5. URN을 이용한 위치개념 클래스 설계

(1) 위치개념 클래스

본 논문에서 설계한 APL중 URN이 들어간 위치개념의 상속 구조는 [그림3] 과 같다.



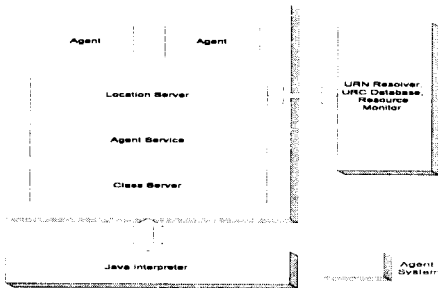
[그림 3] 위치개념 클래스들의 상속도

[그림 3]에서 보면, 이동성(Mobility)클래스는 에이전트간 통신 및 이주를 담당하는 클래스로 하위 클래스로 위치(Location)클래스를 가지고 있고, 위치 클래스는 에이전트의 실행 장소를 정의하는 클래스이다. 또한, URN_to_URL클래스는 URN Resolver와의 통신과 관련된 클래스이다.

(2) 에이전트 시스템의 하부 구조

본 논문에서 설계한 이동 에이전트시스템의 하부구조는 [그림 4]와 같다.

에이전트 시스템의 하부 구조는 구현 시 필요한 다수의 서버를 통칭한 클래스 서버(Class Server), 에이전트의 서비스를 정의한 에이전트 서비스 서버(Agent Service Server), 그리고 이동 에이전트의 이동성을 위해 구현된 위치 서버(Location Server)등으로 이루어져 있다. 이 중 위치 서버는 URN Resolver와 연결되어 에이전트의 이동시에 필요한 위치 정보를 얻을 수 있도록 설계되었다.



[그림 4] 이동 에이전트 System 하부구조

URC 데이터 베이스를 좀더 효과적으로 이용하기 위하여 사용된 자원 감시자에 대한 개념을 적용하였다. 이러한 방식을 이용하여 에이전트의 이동성 증대를 유도해 나간다면 사용자를 위하여 보다 나은 이동 에이전트 서비스를 제공할 수 있을 것이다.

향후 연구과제는 URC 데이터 베이스의 유지 보수 방법 설계, URC 데이터 베이스와 자원 감시자 사이의 효과적인 통신 프로토콜 고안, 자원 감시자의 효과적인 관리 방안의 등의 연구가 요구되어 진다. 또한 이번 설계를 뒤받침 삼아 이동성이 극대화된 APL구현은 물론 본 논문에서 개념을 정리한 URN, URC 데이터 베이스 그리고 자원 감시자사용에 맞는 새로운 이동 에이전트연구 개발이 뒤따라야 할 것이다.

7. 참고 문헌

- [1]J. M. Bradshaw(Ed.), *Software Agents*, AAAI Press/The MIT Press, 1997.
- [2]최석두, "인터넷 정보의 식별과 검색을 위한URIs", HTTP://www.nanet.go.kr/nal/3/3-1-4/nal98122.htm#4.
- [3]황선호 외, "이동 에이전트를 이용한 망 관리 시스템에서의 통합된 관리 인터페이스 설계 및 구현", *한국 정보과학회 '99 봄 학술발표 논문집(26,1)* pp.524-526, 1999.
- [4]A. Tripathi, "Mobile Agent Programming in Ajanta", HTTP://www.cs.umn.edu/Aj-anta.
- [5]R. S. Gray, *Agent Tcl: A flexible and secure Mobile-Agent System* Dartmouth College, 1997.
- [6]K.Sollins and L.Masinter, *RFC 1737: Functional Requirements for Uniform Resource Names*, ftp://ftp.isi.edu/in-notes/rfc1737.txt, Dec 1994.
- [7]R. Moats, *RFC 2141:URN Syntax*, HTTP://www.cis.ohiostate.edu/htbin/rfc/rfc2141.html.
- [8]P. Agre and D. Chapman, "Perigi: An implementation of a theory of activity", *In proceeding 6th AAAI*, pp.268-272, 1987.

6. 결론 및 향후 연구과제

본 논문에서는 URN를 이용한 이동 에이전트의 이동성 극대화라는 것에 초점을 맞추어 에이전트 언어를 설계하였다. 또한 URN과