

인터넷 상에서 Prolog 프로그램 실행 환경 개발

안혜선, 창병모
숙명여자대학교 전산학과

Development of Execution Environment for Prolog Program on Internet

Hye-Sun Ahn, Byung-Mo Chang
Dept. of Computer Science Sookmyung Women's University

요 약

최근 들어 클라이언트/서버 개발의 두드러진 특징 중 하나가 인터넷 컴퓨팅 환경의 발전으로 웹에서 사용할 수 있는 다양한 기능이 추가되어 개발된다는 점이다. 웹에서 이용한 다양한 응용 프로그램 개발에 대한 요구가 증가함에 따라 공동작업 기능 및 추론(inference)을 이용한 지능형 프로그램 개발의 요구가 날로 증가하고 있다. 특히 프롤로그와 같은 논리 언어는 추론, 단순성, 견고성, 자동 메모리 관리, 코드의 이동성 등이 용이하여 현재 프롤로그를 인터넷 프로그래밍 언어에 적합하도록 확장하는 연구가 세계적으로 활발히 진행되고 있다. 본 논문에서 Prolog 응용 프로그램을 인터넷상에서 이용할 수 있는 Prolog 프로그램 실행 환경을 개발하였다. 본 연구는 클라이언트-서버 모델을 기반으로 하였으며 클라이언트와 서버간 통신은 Linda Blackboard를 사용하여 구현하였다. 구축된 환경 하에서 사용자(클라이언트)는 웹 브라우저 상에서 서버에 의해 수행되는 Prolog 응용 프로그램에 대한 질의(query)를 수행할 수 있다.

1. 서론

고성능 컴퓨터 분야와 초고속 네트워크 분야에서의 급속한 기술 발전으로 어플리케이션의 질적, 양적인 요구가 날로 증대되고 있고, 이를 충족시켜 주기 위해서는 정보의 위치, 형태, 시간적 제한에 관계없는 서비스가 제공되어야 한다. 그러나 기존의 단일 컴퓨터 내에서의 작업에 적합하도록 설계된 프로그래밍 언어는 수행속도를 최대화하기 위하여 기계어에 가까운 코드로 번역되었으므로 플랫폼에 의존적이고, 번역된 프로그램 코드들은 한 호스트내의 영역(address space)에서 실행되므로 다른 코드 이동 및 실행을 어렵게 한다. 그래서 분산환경에서 일어나는 작업을 프로그래밍 하는데 많은 어려움이 있으며, 비효율적이다. 따라서 앞으로 개발될 소프트웨어는 인터넷을 기반으로 할 것이며, 이러한 측면에서 인터넷 프로그래밍 언어의 개발 및 확장은 매우 중요하다고 할 수 있다. 특히 프롤로그와 같은 논리 언어는 추론 기능, 단순성, 견고성, 자동 메모리 관리, 코드의 이동성 등이 용이하여 현재 프롤로그를 인터넷 프로그래밍 언어에 적합하도록 확장하는 연구가 세계적으로 활발히 진행되고 있다[BPT96, CH96, CHV96].

본 연구에서는 기존의 Prolog 응용 프로그램을 인터넷상에서도 쉽게 이용할 수 있도록 클라이언트-서버 모델을 기반으로 하여 Prolog 프로그램 실행 환경을 개발하였다. 기존의 Prolog 시스템은 서버로써 기능하여 웹 상의 클라이언트의 요청(질의)에 따라 응용 프로그램을 실행하여 그 결과를 클라이언트에 알려 준다. 서버는 여러 클라이언트로부터의 요청을 순차적으로 서비스한다. 본 연구에서는 클라이언트-서버간 통신을 Linda-Blackboard 모델에 따라 구현하였다. 구축된 환경하에서 사용자(클라이언트)는 서버에서 수행되는 Prolog 응용프로그램에 대한 질의를 웹 브라우저 상에서 수행함으로써 보다 효과적으로 응용프로그램을 사용할 수 있게 하였다.

2. 프롤로그 서버

프롤로그 언어는 인터프리터를 이용하여 텍스트모드에서 수행되어지므로 웹 환경에 적합하지 않다. 이 언어를 웹 환경에서 수행 가능케 하고 다중-사용자가 사용할 수 있게 클라이언트-서버 구조로 구현한다. 본 논문의 클라이언트-서버 구조에서는 서버와 클라이언트간의 정보 교환을 위해 프로세스간의 통신 방법중 하나인 Linda-스타일[TB96, TDVR97]의 블랙보드를 사용하였다.

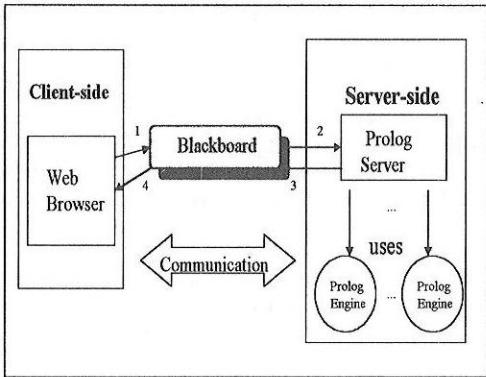
2.1 서버 구성

클라이언트-서버 환경은 네트워크로 연결된 컴퓨터들이 서버와 클라이언트로 각각의 역할을 분담하는 서비스 형태이다.

클라이언트/서버는 Linda 연산자를 이용한 블랙 보드를 사용하여 상호 통신한다. 클라이언트가 블랙보드에 연결하게 되면, 서버는 클라이언트로부터의 요구에 응답하여, 서비스를 제공하는 역할의 기능 분할이 이루어진다. 예를 들면, 서버는 클라이언트의 사용을 지원하기 위한 업무처리 기능을 제공한다. 이는 응용 서비스기능을 전체 또는 일부를 처리하기 위해 제공될 수 있다.

클라이언트의 역할은 첫째, 사용자 인터페이스를 처리하고, 둘째, 서버에게 사용자 요구를 전송하며, 셋째, 서버의 응답 대기기를 기다리고 있어야 한다. 그리고 서버의 역할은 첫째, 프로로그 엔진을 수행하여 여러 종류의 문제 해답(problem solving) 프로그램을 통하여 요청하는 질의에 결과 값을 전달하는 것이 가능하다. 둘째, 클라이언트들의 질의를 큐(queue)와 같은 버퍼(buffer)에 대기시키고, 셋째, 클라이언트의 질의를 실행하여 원하는 결과 값을 얻을 수 있도록 수행되어야 한다. 넷째, 결과를 클라이언트에게 전송 할 수 있어야 한다.

이 서비스는 사용자가 서버에 질의를 블랙보드를 통해 요청하면 서버가 Prolog를 호출하여 그 질의를 수행하고 그 결과로 생기는 값을 블랙보드에 기록한다. 즉 블랙보드와 사용자 사이를 서버가 연결하여 데이터를 전달하는 역할을 한다. 서버는 Prolog를 정의하고 있기 때문에 사용자가 요청한 프롤로그 질의를 수행하여 클라이언트에게 되돌려 줄 수 있다.



[그림 1] 블랙보드를 이용한 Prolog 수행

그림 1에서 보는 것과 같이 클라이언트-서버가 갖는 구성은 간단하다. 그림에 나타난 수행 과정을 설명하면 다음과 같다. Prolog-서버가 블랙보드로부터 입력을 받을 수 있게 Linda 연산인 out(), cin()을 이용한다. 이 때 out(), cin()에 특별한 문자를 포함하여 Linda의 고유한 연산과 구분하였다.

- (1) 클라이언트가 Linda 연산인 out()을 이용하여 블랙보드에 질의를 입력한다.
- (2) 블랙보드를 통해 입력된 질의는 Linda 연산인 cin()

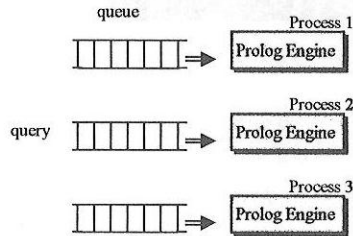
을 통해 Prolog 서버로 보내진다.

- (3) Prolog 실행 후 결과는 out()을 사용하여 블랙보드에 전달된다.
- (4) 클라이언트는 결과를 블랙보드에서 cin()을 이용하여 읽는다.

2.2 구현 모델

클라이언트들이 각각의 질의를 요청하면 서버측에서는 해당 응용프로그램에 수행하기 위해 응용프로그램마다 Prolog 엔진을 수행해야한다. 서버가 Prolog 엔진을 호출하는 방법은 그림 2에서처럼 클라이언트들의 질의에 해당하는 각각의 응용프로그램을 실행하기 위해 Prolog프로세스를 생성한다.

이때 같은 응용프로그램을 실행하는 질의는 큐에 대기시키고 순차적으로 서비스한다.



[그림 2] Prolog 엔진 생성

그래서 다음과 같은 장점을 갖는다.

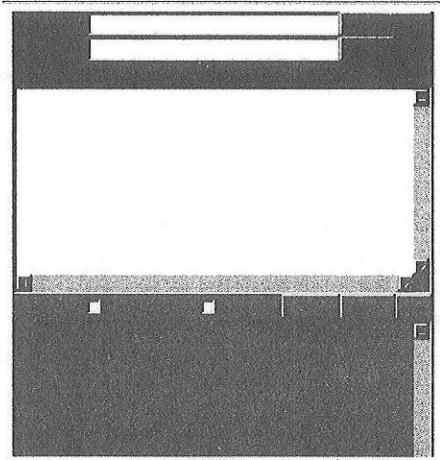
- 언어 처리기가 업그레이드되면 새로운 버전을 서버에만 설치하면 된다.
- 자바로 구현되었으므로 웹에서 그래픽 사용자 인터페이스(GUI: graphical user interface) 를 사용할 수 있다.
- 특정 하드웨어 또는 소프트웨어 플랫폼에 의존하지 않고 개방형 시스템(open system)으로 채택 될 수 있다.

3. 구현

사용자가 웹에서 사용 가능하도록 그래픽 사용자 인터페이스를 구현한다. 이 인터페이스의 구현은 UNIX의 Solaris2.5.1 환경에서 JDK1.1.6 버전(version)과 Linda-서버의 프롤로그 엔진으로는 jProlog1.02 버전을 사용하여 구현하였다.

서버에서 사용될 프롤로그 엔진(engine)으로는 여러 종류의 프롤로그가 있지만, 자바 언어로 작성된 jProlog를 서버로 작성하고, Linda 스타일 블랙보드를 이용하여 클라이언트와 통신한다. 이유는 자바 언어로 프롤로그를 작성하였으므로 이를 이용한 웹 어플리케이션의 경우 사용자에게 보다 능동적인 사용자 인터페이스와 트랜잭션 프로세싱을 제공할 수 있게 되고, 애플릿(applet)과 같은 형태의 코드의 재사용이 가능하게 되어 소프트웨어 개발이 용이해지고, 이식성과 안정성이 뛰어나 시스템을 구현하는데 용이하다.

Linda-Client Prolog



[그림 3] 클라이언트 초기화면

그림 3은 클라이언트가 웹을 통해 서버를 수행하였을 때의 초기 화면이다. 클라이언트들은 이를 수행하기 위해 자바 애플릿을 실행 할 수 있는 웹 브라우저를 가지고 있어야 한다

기능

- Query : 질의어를 쓴다.
- Results : 질의 또는 로드 프로그램에 대한 질의 결과, 에러 메시지를 보여 준다.
- Say : 서버와의 대화 창이다.

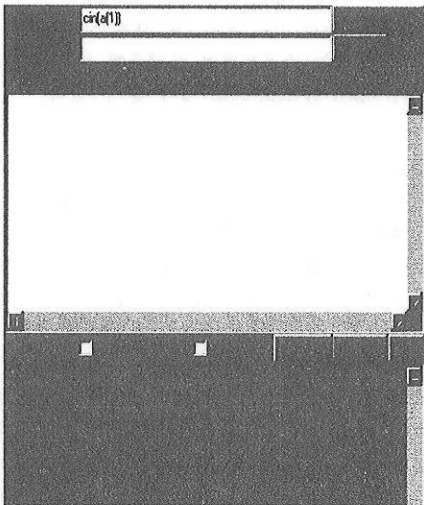


그림 4 질의 결과

그림 4는 Linda-클라이언트가 블랙보드에 있는 결과 값을

Linda 연산자인 cin()을 통해 a(1)가 있음을 Results 텍스트 필드에서 보여주는 화면이다.

4. 결론

서버 프롤로그를 자바 언어로 그래픽 사용자 인터페이스를 구현하여 넷스케이프(netscape), 또는 익스플로러(explorer)와 같은 웹 브라우저에서 사용자가 편리하게 사용할 수 있다. 근래에는 웹 상에서의 사용 문제가 대두됨에 따라 웹서비스에 대한 효율적 방법이 필요하게 되었다. 이를 위하여 Prolog을 서버에 연결하여 서버가 클라이언트들의 요청에 보다 능동적으로 응답하기 쉬워졌다.

특히 논리 언어가 갖는 지능성 등의 특징은 앞으로 개발될 에이전트와 같은 지능형 프로그램의 개발을 위해서 보다 유용하게 사용될 것이다. 하지만 분산처리 환경에 알맞은 논리 언어의 개발은 아직 초기 단계라고 할 수 있으며, 논리 언어가 인터넷 프로그래밍에 효과적으로 사용되기 위해서는 웹과의 자연스런 연동, 에이전트의 지원, 접근 제어 등에 대한 연구가 필요하다.

참고 문헌

- [BPT96] Koen De Bosschere, Daniel Perron, and Paul Tarau. "LogiMOO : Prolog Technology for Virtual Worlds", In Proceedings of PAP'96, pages 51-64, London, April 1996. ISBN 0 9525554 1 7.
- [CH96] D. Cabeza and M. Hermenegildo. html.pl : "A HTML Package for (C)LP Systems", Technical report, 1996.
- [CHV96] D. Cabeza, M. Hermenegildo, and S. Varmaa. "The PiLLOW/CIAO Library for Internet/WWW Programming using Computational Logic Systems", First International Workshop on Logic programming Tools for Internet Applications, Bonn, September 1996.
- [TB96] Paul Tarau and Koen De Bosschere. "Virtual World Brokerage, with BinProlog and Netscape", First International Workshop on Logic programming Tools for Internet Applications, Bonn, September 1996.
- [TDB97] Paul Tarau, Veronica Dahl, and Koen De Bosschere. "A Logic Programming Infrastructure for Remote Execution, Mobile Code and Agents", In Proceedings of IEEE WETICE'97, Boston, MA, June 1997.
- [TDVR97] Paul Tarau, Koen Debosschere, and Veronicadahl, Stephen Rochefort. "LogiMOO : an Extensible Multi-User Virtual World with Natural Language Control", 1997