제27회 한국정보처리학회 춘계학술발표대회 논문집 제14권 제1호 (2007. 5)

프롤로그를 이용한 메쉬업 서비스 시스템에 관한 연구

박성철*, 권기항**
*동아대학교 컴퓨터공학과
**동아대학교 컴퓨터공학과 교수

e-mail :lovewiz@gmail.com, khkwon@donga.ac.kr

A Study on Mash-Up Service System Using PROLOG

Sung-Chul Park*, Ki-Hang Kwon*
*Dept. of Computer Engineering, Dong-A University

요 약

공개된 웹 서비스를 가공하여 새로운 서비스를 만들어내는 매쉬업 서비스가 활발하게 개발되어 사용자에게 제공되고 있다. 그러나 이러한 매쉬업 서비스도 프로그래밍 수준에서 이루어지고 있어 매우 제한된 사람만 개발 할 수 있다는 단점이 있다. 본 논문에서는 이러한 메쉬업 서비스를 기술하고, 기술된 서비스를 효과적으로 실행할 수 있는 시스템을 제안한다.

1. 서론

지난 몇 년간 인터넷 서비스 업체의 가장 큰 이슈는 오픈 서비스였다. 오픈 서비스는 키워드 검색, 사전, 상품정보 등의 고유한 서비스를 XML 기반의 SOAP(Simple Object Access Protocol)[10]또는 REST(Representational State Transfer)등의 웹 서비스로 사용자들에게 공개하는 서비스를 말한다. 이러한 서비스를 공개함으로써 사용자의 직접 공개된 서비스를 이용하여 새로운 서비스를 개발하고, 다시 서비스 할 수 있는 방법을 제공해주고 있다.

(그림 1)은 구글[1]에서 제공하는 지도 서비스와 실시간 고속도로 정보를 제공하는 웹 서비스를 결합한 서비스 이다. 특히 (그림 1)과 같이 공개된 서비스들을 조합하여 새로운 서비스로 제공하는 서비스를 메쉬업(Mash-Up)서비스라 한다. 메쉬업 서비스는 사용자가 원하는 서비스를 서로 다른 공개된 서비스를 조합하여 새로운 서비스를 빠르고 쉽게 개발 할 수 있다는 장점이 있어 최근 큰 관심을 받고 있다. 하지만 지금까지의 메쉬업 서비스 서비스들은 직접 코드수준에서 개발해야만 한다는 단점이 있다. 즉, 프로그래밍에 대한 일정 수준 이상의 지식이 없는 사용자는 자신이 필요로 하는 서비스가 있더라도 직접 서비스를 개발하여 사용 할 수 없다는 문제점이 있다.

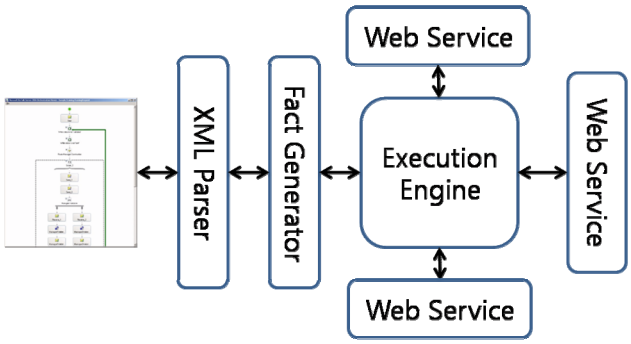
본 논문에서는 이러한 메쉬업 서비스를 효과적으로 개발하고 실행 할 수 있는 메쉬업 서비스 시스템을 제안한다. 본 논문에서 제안하고 있는 메쉬업 서비스 시스템은 프로그래밍에 대한 지식이 없더라도 서비스와 데이터의 흐름을 직관적인 표현 할 수 있는 메쉬업 서비스 디자이너와 정의 된 메쉬업 서비스를 효과적으로 실행 시킬 수 있는 메쉬업 서비스 실행엔진 등으로 구성되었다.

(그림 1) 메쉬업 서비스의 예

본 논문의 구성은 다음과 같다. 우선 2 장에서는 메쉬업 서비스 실행 시스템의 구조와 각 기능에 대해 소개하고, 3 장 에서는 메쉬업 서비스의 구현과 평가에 대해서 논의한다. 그리고 마지막으로 4 장에서는 결론 및 향후 연구과제에 대해서 기술한다.

2. 메쉬업 서비스 실행 시스템의 구조

(그림 2) 메쉬업 서비스 시스템에 대한 구조이다. 메쉬업 서비스를 정의하는 서비스 디자이너, 서비스 디자이너에 의해 생성된 XML 데이터를 분석하는 XML Parser, 분석된 XML 데이터를 이용해 프롤로그로 구현된 실행 엔진에 입력될 Fact 를 만들어 내는 Fact Generator, 그리고 웹 서비스와의 입출력 그리고 서비스들의 실행을 담당하는 실행엔진으로 구성된다.

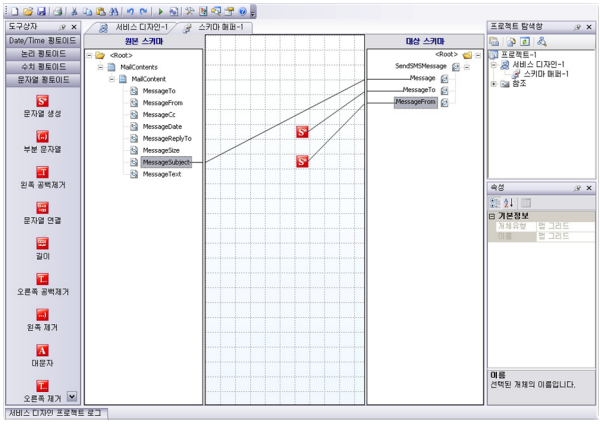


(그림 2) 메쉬업 서비스 실행시스템의 구조

이 구조의 특징은 프로그래밍을 통한 서비스의 통합이 아니라, 서비스 디나이너와 같은 비주얼(Visual) 도구를 통해 메쉬업 서비스를 정의하고, 정의된 서비스를 실행하는 구조이다. 이는 프로그래밍에 대한 지식이 없는 사람도 쉽게 메쉬업 서비스를 개발하여 사용 할 수 있으며, 빠르게 서비스를 수정하거나 확장할 수 있다는 것을 의미한다. 각 부분의 내용을 살펴보면 다음과 같다.

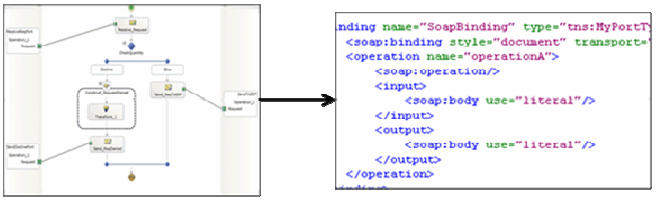
1) 메쉬업 서비스 디자니어

메쉬업 서비스 디자이너는 메쉬업 서비스에 대한 정의를 비주얼 도구를 이용하여 메쉬업 서비스를 정의한다. 메쉬업 서비스의 흐름을 나타내는 화살표와 반복, 조건, 서비스 호출, 대기 등의 기능을 표현하는 그래픽 이미지를 이용하여, 서비스에 대한 입력과 출력을 다른 서비스에 대한 입력으로 연결하는 방식으로 서비스의 조합을 정의한다.



(그림 3) 메쉬업 서비스 정의

입력과 출력의 데이터 불일치에 대한 데이터의 변환은 (그림 3)과 같은 맵핑(Mapping)방식의 디자인 도구를 이용해 데이터 변환을 정의한다. 이러한 변환정보는 XSLT(eXtensible Stylesheet Language Transformations)[15]를 생성하여, 실행시간의 데이터 변환에 이용되게 된다. 이와 같은 방법으로 메쉬업 서비스를 정의면 서비스 디자이너는 (그림 4)과 같이 XML 데이터로 메쉬업 서비스를 정의하게 된다.



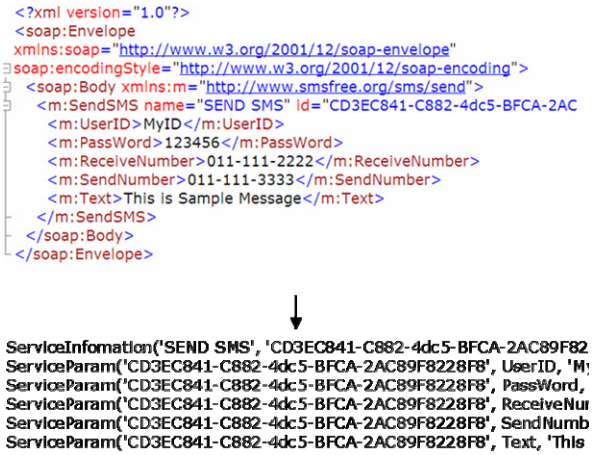
(그림 4) 메쉬업 서비스 정의

2) XML Parser

XML Parser 는 정의된 XML 데이터의 유효성을 검사하고, XML 데이터를 분석한다. 정의된 어트리뷰트(Attribute)와, 엘리먼트(Element)에서 데이터를 추출하여, Fact Generator 에 전달하기 위한 자료구조를 생성한다.

3) Fact Generator

Fact Generator 는 프롤로그 기반의 실행엔진을 위해 서비스 정의에서 정의된 XML 데이터 정보를 관계(Relation)로 표현하는 Fact 를 만들어 낸다. Fact Generator 는 실행시간에 필요한 정보와 상태를 XML Parser 에 의해 생성된 자료구조를 이용해 (그림 5)와 같은 방식으로 Fact 를 만들어 낸다.



(그림 5) XML Data 의 Fact 변환

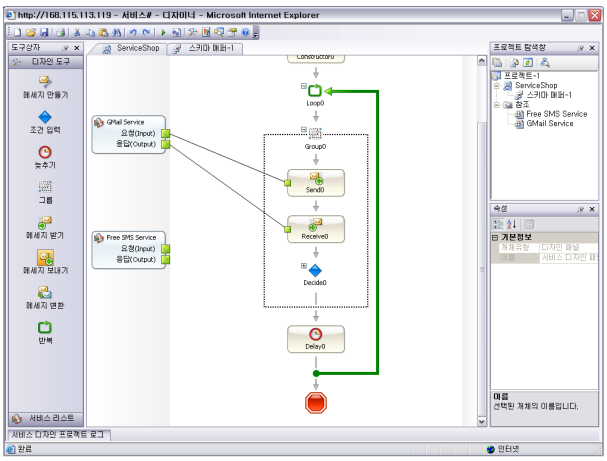
4) 실행엔진

실행엔진은 프롤로그(P#)[9] 기반이므로 실행엔진에 저장된 Fact 를 근거로 하여, 실행 시 서비스의 상태 정보, 데이터 처리의 유효성 검사, 조건 검사 등을 쿼리(Query)를 통해 필요한 정보를 검색하고 실행 모듈을 통해 서비스를 실행하는 역할을 한다.

3. 메쉬업 서비스 시스템의 구현과 평가

1) 메쉬업 서비스 시스템의 구현

본 메쉬업 서비스 시스템은 마이크로소프트의 단넷 플랫폼[14]을 기반으로 하는 C#과, 프롤로그의 단넷버전인 P#을 사용하여 개발 되었다. P#은 Dr. Jon Cook[13]에 의해 개발되었고, 프롤로그 코드를 단넷프레임워크에서 사용 할 수 있도록 해주는 컴파일러 시스템이다. P#의 코드 내에서 C#객체를 생성 할 수 있으며, C#의 그래픽, 네트워크 등 모든 단넷 프레임워크에 포함된 모든 라이브러리를 사용 할 수 있다. 그리고 P#의 런타임 라이브러리 역시 C#을 이용하여 제어 할 수 있으며, 프롤로그 코드를 실행시간에 실행 할 수 있는 특징이 있다.



(그림 6) 서비스 디자이너

사용자가 메쉬업 서비스를 정의하는 서비스 디자이너는 C#을 이용하여 개발되었으며 (그림 6)과 같은 인터페이스를 가진다. (그림 6)과 같이 서비스를 정하게 되면, 배치된 그래픽 객체정보에 따라 XSLT(eXtensible Stylesheet Language Transformations)를 이용하여, 서비스를 정의하는 XML 형태의 서비스 정의 문서를 생성하게 된다.

XML Parser 는 MSXML 4.0 을 이용하여 개발되었으며, 서비스 정의 문서를 파싱 하여, 트리기반의 데이터를 생성한다. Fact Generator 역시 XSLT 를 이용하여 서비스 정의 문서를 Prolog 코드로 변환을 한다.

실행엔진은 P#과 C#으로 구현되었고 크게 검증모듈과 실행모듈로 나뉘어진다. 검증모듈은 P#으로 개발 되었고, 서비스 상태와 조건검사와 같은 논리적 검증을 담당하고, C#으로 개발된 서비스 실행모듈은 입력 데이터를 받아 웹 서비스를 실행하고, 그 결과를 Fact 로 만들어내는 일을 담당한다.

4. 결론 및 향후 연구과제

본 논문에서는 메쉬업 서비스를 효과적으로 제작하고 실행 할 수 있는 메쉬업 서비스 시스템을 제안하고 이를 통해 효과적으로 메쉬업 서비스를 제작하고 실행하였다. 본 논문의 의의는 메쉬업 서비스 디자이너를 이용한 효과적인 메쉬업 서비스 개발과, 서비스 디자인 부분과 실행엔진을 분리하여 확장성과 재사용성을 높인 점이라 할 수 있다.

추후 연구과제로는 첫째로 Fact 정보관리를 파일시스템과 연동하여, 긴 트랜잭션(Long-term Transaction)을 효율적으로 처리하는 것이고, 둘째는 메쉬업 서비스 정의도구를 웹 기반 프로그램으로 개발하여 사용자의 접근성과 사용성을 향상시키는 것이다.

참고문헌

[1] Google APIs Available at http://code.google.com/apis/
[2] Alonso, G. Gunthor, R., Agrawal, D., El abbadi, A. and Mohan, C., AdvancedTransaction Models in Workflow Contexts, Proceedings of the Twelfth International Conference on data Enginerring, 574-581., 1996
[3] Charfi, A. and Mezini, M., Hybrid Web service Composition: Business Processes Meet Business Rules, Proceedings of the 2nd international conference on Service oriented computing, 30-38., 2004
[4] S.W. Loke and A. Davison., Logic Programming with the world-wide-web., In Proceedings on the 7th ACM Conference on Hypertext pages 235-245. ACM Press, 1996
[5] Kim, Y.H., Kang, S.H., Kim, D.S., Bae, J.S. and Ju, K.J., WW-FLOW: Web-Based Workflow Management with Runtime Encapsulation, IEE Internet Computing, 4(3), 55-64
[6] Ivan Bratko, Programming for artificial intelligence(3rd edn), Addison Weseley, 2000
[7] Michael A. Covington, Donald Nute, Andre Vellino,

Prolog Programming in Depth, Prentice Hall, 1996
[8] Leon Sterling and Ehud Shapiro, The Art of Prolog: Advanced Programming Techniques, MIT Press, 1994
[9] P# Available at http://www.dcs.ed.ac.uk/home/stg/Psharp/
[10] Simple Object Access Protocol(SOAP) Available at http://www.w3.org/TR/soap/
[11] BPEL4WS Available at http://www-106.ibm.com/developerwokrs/webservices/library/ws-bpel/
[12] BPML Available at http://bpmi.org
[13] Jon Cook's Home, http://homepages.inf.ed.ac.uk/jcook/
[14] The Microsoft developer .NET home page, http://msdn.microsoft.com/net
[15] XSLT, http://www.w3.org/TR/xslt