

실행시간 Re-planning을 지원하는 웹 서비스 에이전트

김여정^o, 김지현, 김운, 강지훈
삼성전자, 한국전력기술, 충남대학교 컴퓨터학과
yeojung99.kim@samsung.com^o, jhkim10@kopec.co.kr, {wkim, jhkang}@cnu.ac.kr

Web Service Agent with Re-Planning at execution time

Yeo-Jung Kim^o, Ji-Hyeon Kim, Yun Jin, Ji-Hoon Kang
Samsung Electronics^o, KOPEC, Chungnam National University

요 약

기존의 시맨틱 웹 서비스 에이전트는 사용자 요구에 의미적으로 부합하는 서비스를 검색하고 합성하는 플래너에 중점을 둔 반면 실행자에는 큰 비중을 두지 않았다. 그러나 기존의 다수 플래닝 알고리즘은 사용자 제약 사항의 일부만을 적용하여 불완전한 결과를 도출함에도 불구하고 플래닝을 완료하는 문제가 있다. 따라서 실행자가 단순히 플래닝된 서비스만을 실행한다면 사용자 요구를 완전히 충족시킬 수 없다. 이를 해결하기 위해 본 논문에서는 플래닝의 취약점을 보완하는 실행자를 제안하고 이를 기반으로 에이전트를 개발하였다. 본 논문에서 제안한 실행자는 서비스 실행 결과를 분석하여 사용자의 요구 사항 중 반영되지 않은 정보의 적용 또는 불필요한 정보의 제거를 위해, 플래너에게 re-planning을 요구하거나 서비스 실행 결과 자체에서 관련 정보를 추출한다. 이를 통해 사용자의 요구 사항에 대한 서비스 결과의 정확도 향상을 도모한다.

1. 서론

기존의 웹 서비스[1]와 시맨틱 웹[2]의 특징을 결합시킨 시맨틱 웹 서비스[3]는 사용자 대신 에이전트가 의미에 맞는 웹 서비스 검색(matchmaking), 검색된 웹 서비스들의 플래닝(planning), 실행(execution)을 자동으로 수행하여 최종 결과를 제공하여 사용자의 중간 개입을 최소화 시킨다. 이는 에이전트가 자원의 개념과 자원들간의 의미적 연결을 표현한 온톨로지를 추론함으로써 의미(semantics)를 부여한 서비스의 처리가 가능하기 때문이다.

현재까지의 시맨틱 웹 서비스에 대한 연구는 플래닝된 서비스를 실행하는 과정은 큰 비중을 두고 있지 않다. 그러나 플래닝은 실질적인 웹 서비스 실행 결과가 아닌 서비스 온톨로지를 통한 정적인 합성이므로, 실행자는 단지 플래닝 결과를 실행시키고 결과를 수집하는 역할에서 벗어나 플래닝 결과를 검증할 필요가 있으며, 이는 다음과 같은 문제점에서 기인한다.

첫째, 기존 다수의 플래닝 알고리즘[4][5][6]에서 발견되는 문제로 사용자 입력 사항의 일부만을 사용하였음에도 불구하고 플래닝이 완료되어, 사용자가 원하는 결과를 완전하게 충족시킬 수 없는 점이다.

둘째, 융통성 있는 서비스 검색을 위해 본 논문에서

사용한 확장된 플래닝 알고리즘[7]에서 발생하는 문제로 이 알고리즘은 사용자의 요구 사항에 부합하는 서비스의 검색 가능성을 확장시키는 장점이 있는 반면 사용자 요구에 정확히 일치하지 않는 정보도 획득하는 단점이 있다.

본 논문에서 제안하는 에이전트는 실행자에게 플래닝된 서비스들을 실행시키는 역할 뿐 아니라, 그 결과를 검증하여 불필요한 정보는 제거하고, 플래닝 과정에서 반영되지 않은 사용자의 요구 사항을 반영시키는 플래너의 보완자 역할을 수행한다. 이는 플래너에 re-planning을 요청하거나 서비스 실행 결과 자체에서 필요한 정보를 추출함으로써 가능하다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구를 알아보고, 3절에서는 에이전트 설계 및 구현에 대해 살펴보고, 마지막 4절에서는 결론 및 향후 연구에 대해 소개한다.

2. 관련연구

Sheshagiri[4]가 제안한 서비스 합성 플래너는 플래닝을 위해 STRIP[5]을 이용하여 DAML-S[8]로 기술된 단일 서비스를 액션으로 바꾼다. 플래닝 알고리즘은 본 논문에서 사용한 [7]의 알고리즘과 유사한 partial-order backward-chaining으로 최종 goal이 사용자에게 받은 의

* 이 연구는 BK21 충남대학교 정보통신 인력 양성 사업단과 소프트웨어 연구 센터의 지원을 받았음

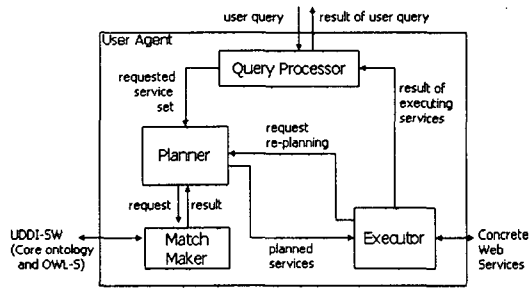
부 입력과 같을 경우 합성은 성공적으로 완료된다. 그러나 합성 완료의 조건이 외부 입력과 모두 일치할 때가 아닌 최종 goal이 외부 입력으로부터 값을 받을 수 있는 경우이므로 사용자의 외부 입력이 전부 적용되지 않고도 플래닝은 성공적으로 완료된다. 그러나 논문에서 이에 대한 문제 제기를 하지 않고 있으며 해결 방법에 관한 언급도 찾아볼 수 없다.

Sirin[6]은 입력, 출력의 매치를 통한 웹 서비스 파이 프라이닝을 제안한다. 사용자 제약 사항을 입력으로 하는 서비스를 찾고, 해당 서비스의 출력을 입력 받는 서비스를 연속적으로 찾는 방식으로 최종 선택된 서비스가 사용자의 요구 사항 일부 만족시 플래닝은 종료된다. 사용자 제약 사항의 일부만 충족되어도 매치되는 서비스로 간주하므로 불충분한 결과를 제시한다.

또한 [4][5][6]에서 제안한 시스템들은 원하는 서비스가 정확히 매핑될 때의 경우를 고려하였다. 그러나 [9]에서는 에이전트가 인터넷 환경에서 일대일 매핑되는 서비스를 찾기가 아주 어렵다고 주장한다. 본 논문에서는 이 점을 감안하여 정확히 매핑되는 서비스 외에도 의미적으로 포함되는 경우도 후보 서비스로 고려하였다.

3. Re-planning 사용자 에이전트

[그림 1]은 사용자 에이전트의 내부 구조를 보여준다. 사용자 에이전트는 사용자 질의를 분석하는 질의 처리기(Query Processor)와 서비스를 합성하는 플래너(Planner), 서비스의 일치 여부를 판단하는 매치메이커(MatchMaker)와 플래닝된 서비스를 수행하는 실행자(Executor)로 구성된다.



[그림 1] 사용자 에이전트의 내부 구조도

3.1 플래너

본 논문에서 사용한 플래닝 알고리즘[7]은 기존의 플래닝 알고리즘이 사용자의 입력 또는 출력이 정확하게 매핑되는 서비스만을 플래닝의 대상으로 했던 반면 온톨로지를 이용한 추론을 통해 개념 간에 유사한 관계가 있는 서비스도 그 대상으로 한다.

‘한국 국적(nationality)의 화가(Painter)가 2004년도에 그린(paintedYear) 회화(Painting)를 검색하라’는 질의가

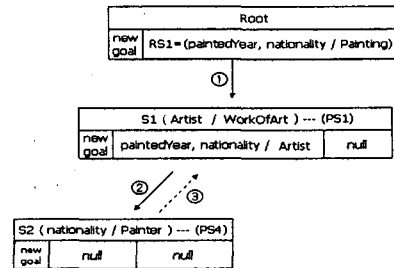
입력되면 질의 처리기에서 플래너를 위해 아래와 같은 요청 서비스 집합을 구성한다.

$RSS = \{ RS1 = nationality, paintedYear / Painting \}$

제공 서비스 목록은 다음과 같으며 RS1과 일치하는 서비스는 없음을 알 수 있다.

PS1 = Artist / WorkOfArt
PS2 = Painting / paintedYear
PS3 = Painter / nationality
PS4 = nationality / Artist

Painting을 제공하는 서비스는 없지만, [7]의 plugin 매치에 의해 Painting을 포함하는 상위 개념인 WorkOfArt를 제공하는 서비스로 대체함으로써 유연성을 제공한다. [7]의 알고리즘에 따라 플래닝을 수행하면 [그림 2]와 같은 플래닝 트리가 형성된다.

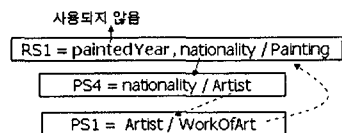


[그림 2] 플래닝 트리

3.2 실행자

실행자는 플래닝 트리를 기반으로 작성된 서비스 목록을 순차적으로 실행시키기 위해 매치메이커로부터 호출할 서비스의 정보를 얻는다. 이 정보와 WSDL[10] 문서 정보를 이용하여 메시지 타입의 매핑 여부 등을 고려하여 실제 서비스를 호출한다.

본 논문에서 제안하는 실행자의 가장 중요한 역할은 검색된 결과를 검증하는 일이다. 검증을 위해서는 플래닝 과정에서 포함된 불필요한 결과를 필터링하고 사용자의 요구 사항이 제대로 반영되었는지를 판별해야 한다. 실행자가 해결해야 할 문제는 사용자의 입력 값이 제대로 사용되지 않아 정확한 값 이외의 결과가 발생했을 경우와 정확히 일치되는 개념이 없어 plugin/non-plugin매치를 통해 확장된 서비스를 검색한 경우 결과의 범위가 사용자가 요구하는 범위와 일치하지 않아 발생하는 경우이다.



[그림 3] plugin 매치가 적용된 서비스 플래닝

[그림 3]의 플래닝 순서에 따라 먼저 사용자로부터 입력 받은 nationality를 이용하여 PS4를 실행시켜 결과

로 Artist를 얻는다. Artist를 이용하여 PS1을 실행시켜 플래닝에 의한 서비스 최종 실행결과인 WorkOfArt를 얻었다.

WorkOfArt는 사용자가 요구한 Painting의 상위개념으로 Painting뿐 아니라 Sculpture, Craft등의 부가적인 정보도 포함하고 있다. 이는 사용자의 요구사항을 벗어나므로 Painting 정보를 분리해낸다. [표 1]과 같이 서비스 실행 결과가 RDF/XML형태인 경우는 실행자가 직접 정보를 추출한다.

또한 플래닝 과정을 살펴보면 사용자가 제한사항으로 제시한 paintedYear=2004라는 항목이 적용되지 않은 채 플래닝이 성공적으로 완료됨을 볼 수 있다. 따라서 추출된 결과는 2004년뿐 아니라 다른 년도에 제작된 Painting도 포함되어 있으므로 다시 2004년에 제작된 Painting만을 추출한다.

[표 1] 서비스 실행 결과

Artist / WorkOfArt 서비스 수행 결과 (RDF/XML)
<Sculpture rdf:about="http://winslab.cnu.ac.kr/sw/sculpture1">... <Sculpture rdf:about="http:// winslab.cnu.ac.kr/sw /sculpture2">... <Craft rdf:about="http winslab.cnu.ac.kr/sw /craft">... <Painting rdf:about="http:// winslab.cnu.ac.kr/sw /painting1"> <paintedYear>1999</paintedYear> ... <Painting rdf:about="http:// winslab.cnu.ac.kr/sw /painting2"> <paintedYear>2004</paintedYear> ... <Painting rdf:about="http:// winslab.cnu.ac.kr/sw /painting3"> <paintedYear>2003</paintedYear> ...
Type 이 Painting 인 자원만 추출한 결과
<Painting rdf:about="http:// winslab.cnu.ac.kr/sw /painting1"> <paintedYear>1999</paintedYear> ... <Painting rdf:about="http:// winslab.cnu.ac.kr/sw /painting2"> <paintedYear>2004</paintedYear> ... <Painting rdf:about="http:// winslab.cnu.ac.kr/sw /painting3"> <paintedYear>2003</paintedYear> ...
paintedYear가 2004인 Painting 자원만 추출한 결과
<Painting rdf:about="http:// winslab.cnu.ac.kr/sw /painting2"> <paintedYear>2004</paintedYear> ...

그러나 아래처럼 결과가 RDF/XML형식이 아닌 경우 paintedYear에 관한 정보를 찾을 수 없다.

http:// winslab.cnu.ac.kr/sw /painting1, http:// winslab.cnu.ac.kr/sw/painting2, http:// winslab.cnu.ac.kr/sw/painting3

따라서 이 경우는 플래너에게 re-planning을 요청해야 한다. 이 때 플래너에게 실행자가 서비스 실행을 통해 획득한 정보인 Painting을 입력으로 하고 사용자의 입력 중 적용되지 않은 정보인 paintedYear를 결과로 출력하는 서비스를 요청한다.

플래너는 제공 서비스 목록에 기술된 PS2= Painting / paintedYear를 찾아준다. 실행자는 이 서비스를 실행시켜 얻은 값을 사용자가 입력한 paintedYear의 2004와 같은지 비교하여 아래와 같이 2004년의 Painting를 추출하여 사용자에게 제공한다.

검색된 정보	PS2 검색 결과	최종 결과
http://.../painting1	paintedYear = 2004	http://.../painting1
http://.../painting2	paintedYear = 1999	
http://.../painting3	paintedYear = 2003	

4. 결론

본 논문에서는 사용자를 대신하여 서비스의 검색과 플래닝을 통해 얻어진 계획된 서비스들을 실행하여 정보를 획득하고, 획득한 정보에 대한 필터링과 re-planning을 제공하는 시맨틱 웹 서비스 에이전트를 설계하고 구현하였다. 본 논문에서 제안한 에이전트는 다음과 같은 장점을 갖는다.

첫째, 부정확한 결과를 도출하는 플래닝 알고리즘에 대한 해결책을 제시한다. 실행자는 플래닝된 서비스를 실행하고 적용되지 않은 사용자의 요구 사항을 찾아 서비스 실행 결과를 기반으로 자체적으로 요구 사항에 부합하는 정보만을 추출한다. 실행 결과에서 정보를 직접 추출할 수 없는 경우는 플래너에게 re-planning을 요청하여 이전 결과와 비교하여 정보를 도출하여 플래너의 단점을 보완한다. 이러한 구조를 통해 사용자의 요구에 대한 보다 정확한 정보의 제공이 가능하다.

둘째, 획득한 정보에 대한 필터링 기능을 제공한다. 본 논문에서 제안한 에이전트는 웹 서비스 매칭에 있어서 정확한 서비스 매칭만 고려한 기존의 방법에 Plugin/non-Plugin 매칭 개념을 도입한 플래닝 알고리즘 [7]을 사용함으로써 더 확장된 서비스를 제공할 수 있다. 그러나 확장된 서비스를 제공하면 사용자 요구에 정확히 일치하지 않는 정보도 획득하므로 이것을 걸러내어 보다 정확한 정보를 사용자에게 제공한다.

서비스 실행 결과를 검증하는 단계는 비싼 시간 비용을 요구하는 작업으로 응답 시간을 줄일 수 있는 방안의 모색이 향후 연구에서 다루어져야 할 것이다.

5. 참고문헌

- [1] Web Services, <http://www.w3.org/2002/ws/>
- [2] W3C Semantic Web, <http://www.w3.org/2001/sw>.
- [3] Semantic Web Services, <http://www.daml.org/services/>
- [4] Sheshagiri, M., desJardins, M., Finin, T., "A Planner for composing Service Described in DAML-S," ICAPS 2003, Workshop on planning for Web Services, 10 June 2003, Trento, Italy.
- [5] Fikes, R., Nilsson, N., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," Artificial Intelligence, 2(3/4), 1971.
- [6] Evren Sirin, James Hendler, Bijan Parsia, "Semi-automatic Composition of Web Services using Semantic Descriptions.", ICEIS2003, 2002
- [7] Ji-Hyeon Kim, Jin Yun, Yeo-Jung Kim, Ji-Hoon Kang, "Extended Planning Mechanism to Increase Web Service Utilization", APWEB, 2005.
- [8] DAML-S, <http://www.daml.org/services/daml-s/0.9/>.
- [9] Sycara K., Widoff S., Klusch M., Lu J. G., "LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace," Autonomous Agents and Multi-Agent Systems, 5, PP173-203, 2002.
- [10] WSDL, <http://www.w3.org/TR/wsdl>