

서비스 컴퓨팅 환경에서의 문법기반의 대화모델을 통한 소프트웨어 프로세스 관리

김웅섭
동국대학교 컴퓨터정보통신공학부
e-mail : woongsup@dongguk.edu

Software Process Management in Service Computing Using Grammar-Based Dialog Models

Woongsup Kim
Dept. of Computer & Information Communication Engineering
Dongguk University, Seoul, Korea

요 약

서비스기반 아키텍처 (SOA)는 웹기반의 표준화된 인터페이스 및 데이터 교환을 통하여 다양한 기능을 지원하기 위하여 유연성있는 비즈니스 시스템의 통합이 간편하다는 장점을 가지고 있어 현재 많은 IT 시스템에서 사용되고 있다. 하지만 기본적으로 SOA 는 call-and-return 방식으로 구현되어 있어 소프트웨어 개발 프로세스과 같이 시스템 또는 시스템 참여자간의 복잡한 상호작용이 수반되는 프로세스를 지원하는 시스템을 구축하기에는 한계를 보이고 있다. 따라서 개발 프로세스에 적용되는 시스템에서 복잡한 상호작용을 구현할 수 있는 SOA 기반 시스템이 필요하다. 우리는 이 논문에서 복잡한 상호작용을 지원할 수 있는 SOA 기반 프레임워크를 소개한다. SOA 기반 프레임워크에서 다양한 상호작용을 지원하기 위하여 우리는 semantic web 과 dialog act 이론을 사용하였으며 개발프로세스에 적용할 수 있는 ontology 를 개발, 사용함에 의하여 SOA 환경에서 다양한 메시지를 표현할 수 있도록 하였다. 또한 마지막으로 소프트웨어 개발 프로세스 사례연구를 통하여 우리의 접근방법의 타당성을 보인다.

1. 서론

서비스 컴퓨팅은 접속 인터페이스 및 데이터교환 방식에 웹기반의 표준을 사용하는 시스템 통합방식으로 플랫폼 독립적인 복합 서비스 시스템의 구축과 시스템 통합을 쉽고 간편하게 할 수 있다는 장점을 가지고 있기 때문에 최근 많은 시스템 구축에 사용되고 있다. 이러한 서비스 컴퓨팅에서는 소프트웨어의 각 기능이 패키지화되어 서비스의 형태로 사용자들에게 제공되고 웹표준의 인터페이스를 통해 프로그래밍 언어, 프로토콜, 플랫폼에 상관없이 접속이 가능하게 된다. 이 서비스 컴퓨팅은 웹서비스라는 기술을 기반으로 하며 시멘틱 웹, 온톨로지 등을 통하여 다양한 목적에 맞는 맞춤형 서비스를 제공하게 된다. 따라서 최근에 주목을 받고 있는 클라우드 컴퓨팅, 그리드 컴퓨팅 및 기타 IT 서비스 플랫폼 개방형 시스템에 적용되기 좋다는 장점을 가지고 있다 [1].

소프트웨어 개발 프로세스에는 여러 작업들이 수반되며 이들 작업에는 다양한 도구들이 사용되고 있으며 여러 부서에서 분산 진행되는 경우가 많다 [2]. 따라서 소프트웨어 개발 프로세스에는 다양한 도구들을 통합하고 다양한 형태의 인력들이 정보를 공유, 교환할 수 있는 플랫폼이 필요하며, 서비스 컴퓨팅 패러다임을 도입하게 될 경우 정보시스템 통합의 측면에

서 큰 이점을 가지고 있다고 파악된다.

본 연구에서는 다양한 이종의 시스템을 통합하여 소프트웨어 개발 프로세스 진행할 수 있고 이를 통하여 개발 프로세스의 최종 결과물의 품질을 보장할 수 있는 시스템을 구축하는 데 서비스 컴퓨팅 패러다임을 응용하는 방향으로 연구를 진행하여 왔다.

하지만 서비스 컴퓨팅의 주요 핵심기술로 사용되고 있는 웹서비스의 플랫폼 독립적인 인터페이스 및 데이터교환 방식은 그 특성상 개발 프로세스를 통합하는데 몇 가지 주요 한계점이 내재함을 발견하였다. 서비스 컴퓨팅 환경은 플랫폼 중립적인 동작환경을 제공하고 있기는 하지만 웹 서비스의 경우 call-and-return 방식의 인터페이스와 데이터에 대한 표준화가 진행되어 있어 소프트웨어 개발 프로세스의 개발 진행내용을 모니터하고 이를 제어할 수 있는 요소들을 구축하는데 어려움을 가지고 있다[3].

본 연구는 소프트웨어 개발에 필요한 다양한 요소들을 모니터하고 제어할 수 있는 서비스 기반의 프레임워크를 제안하는데 목적을 가진다. 이를 위해서 우리는 두 가지 방향으로 연구를 진행하였다. 우선 첫 번째로 우리는 협업 환경에서 개발 프로세스의 품질관리 및 기타 소프트웨어 프로세스에 나타나는 다양한 액티비티(activity)상태 및 정보를 교환할 수 있도록

록 하는 대화형 모델(dialog model)을 제안하였으며, 두 번째로 개발 프로세스에서 생성되는 다양한 정보들을 통합표현할 수 있는 온톨로지를 정의하는 것이다.

본 논문에서 우리는 대화형 모델은 dialog act 이론을 응용하여 정의하였으며 이러한 모델을 통하여 웹 서비스 기반의 메시지 표현 및 교환이 이루어지며, 소프트웨어 프로세스에 필요한 온톨로지를 제시함에 의하여 서비스 컴퓨팅 환경에서 다양한 메시지의 의미를 이해하고 처리할 수 있도록 하는 프레임워크를 제시한다.

2. 대화형 시스템 모델

우리가 제안하는 시스템의 모델은 speech act theory [4] [5] [6] [7]에 근거하여 서비스 컴퓨팅 환경에 적용할 수 있도록 설계되었다. Speech act theory 는 대화의 참가자들이 교환하는 메시지를 크게 inform 과 request 의 두 가지의 의도를 가지는 conversational action 들로 파악하고 메시지 내용 추론 작업(grounding)을 통해 상대방이 전달하는 메시지의 내용을 파악하게 하는 의도로 제안된 이론으로 개발 프로세스에서 산출되는 여러 메시지들을 표현하고 대화참가자들 간의 메시지 교환과정을 모델링하는 데 응용할 수 있다는 장점을 가진다.

정의 1. 대화형 메시지 모델

우리가 제안하는 대화형 모델은 speech act 이론에 따라 정의하였으며, 이 모델에 의한 메시지는 triple 로 아래와 같이 구성되어 있다.

$$M = (I, A, T)$$

- $I = \{assert, reassert, info-request, action-request, offer, opening, closing, accept, reject, hold\}$ 는 메시지의 intention 을 의미하는 것으로 전달하는 메시지의 구분자를 의미하며, 전달하는 메시지를 grounding 하는 데 필요한 tag 로서 작용한다. 이 intention 은 현재 전달하는 메시지가, 상대방에게 약속된 또는 참고할 만한 특정 사항을 전달하거나 (assert, reassert), 또는 상대방에게 특정 행동을 강제하거나 (action-request, info-request, hold), 상대방에 강제하지는 않지만 답변을 요구하거나 (offer), 대화의 시작 또는 끝을 알리거나 (opening, closing), 상대방이 어떤 요청을 했을 경우에 해당하는 답변 (accept, reject, hold)인지를 나타내어 메시지 수신자가 전달된 메시지에 대해 적절한 행동을 할 수 있도록 하는 역할을 한다.

- A 는 실제 전달하는 dialogue action 을 의미하며, 메시지의 내용을 의미하는 것으로 상대방에게 제공하는 정보 또는 상대방에게 해주기를 원하는 (상대방의 action 을 유도하는) 내용을 나타낸다. 실제 전달하는 메시지는 production rule 의 형태로 정의되며 production rule 은 온톨로지로 정의하여 사용할 수 있도록 하였으며 knowledge base 에 저장되어 공유할 수 있도록 한다. production rule 은 정의 2 에서 기술하였다.

- T 는 A 의 production rule 의 좌측 비단말기호에 대한 URI 또는 knowledge base 의 reference 지점을 나타

내며 메시지의 수신자는 T 를 통하여 knowledge base 에서 단말 또는 비단말 기호의 의미를 추출하여 메시지의 의미를 파악할 수 있다.

정의 2. Production Rule

사용자가 전달하는 메시지에서 기술하고자 하는 activity 는 우리 모델에서는 production rule 로 표현한다. Production rule 은 문맥 자유 문법(context free grammar) 의 형태로 표현되며 $T \rightarrow_M a$, T 는 정의 1 에서의 target 을 말하며 $T \in TNF$ 이고, a 는 정의 1 의 action 으로 $a \in FT, \cup NFT$ 의 형태를 가지고 있다. 여기서 A 는 프로세스의 한 activity 를 의미하며 FT 로 표현된다. a 는 구체적인 내용을 의미하고 TNT 와 FT 로 표현된다. \rightarrow_M 는 mapping 함수로서 단말기호를 대응하는 함수로써 $A \rightarrow_M a$, $a \in TF, \cup TNF$ 로 표현된다. mapping 함수의 예를 들면 온도 모니터링 결과는 $CURRENT_TEMP \rightarrow_M celcius:integer$ 의 형태로 표현된다.

정의 3. 기능적 요소

기능적요소(FT)는 본 시스템에서는 비단말기호로 표현되며 서비스의 기능적 요소를 표현하기 위하여 사용되며 실행환경의 변화에 따른 변화가 없는 서비스의 요소들의 집합 또는 knowledge base 에 저장된 production rule 로 정의된다. 따라서 기능적 단말기호의 경우 다음과 같이 정의되는 웹서비스의 집합으로 정의된다.

$$FT = \langle p, s, q \rangle$$

- s 는 서비스가 제공하는 내용을 의미한다. o 는 실제 서비스가 호출되어 작업을 하는 부분으로 호출인자와 리턴인자로 구성된 함수로 표현된다.

- p 는 pre condition 을 나타내며 pre condition 은 서비스 s 가 시행되기 위한 제약 조건을 의미하고 p 가 만족될 경우에만 p 를 사용자가 호출할 수 있다는 것을 의미한다.

- q 는 post condition 으로 서비스 실행후의 effect 와 서비스 사용 후의 상태를 표현한다.

정의 4. 비기능적 요소

비기능적 요소(NFT)의 경우 본 시스템에서는 단말기호로 표현되며 비기능적 데이터를 표현하는 것으로 서비스의 QoS 요소들을 포함한 사용자의 실행 context 를 나타내는데 사용된다. 서비스의 실행 context 에 대응하는 문법 규칙은정의 1 의 모델의 M 에 정의되며 비 기능적 단말기호는 다음과 같이 정의되는데 가드 함수에 의하여 어떤 단말기호로 치환될지 결정하게 된다.

$$NFT = \langle D, AT, AN \rangle$$

- D 는 특정 비기능적 요소들의 상황을 표현하는 문장들의 집합으로 QoS 와 관련된 부분들을 나타내며 범주형으로 표현되며 유한집합의 특성을 가진다. 예를 들면 서비스 예약 상태를 모니터링하는 것을 나타내는 production rule 이 $\langle reservation_status \rangle \rightarrow service_full$

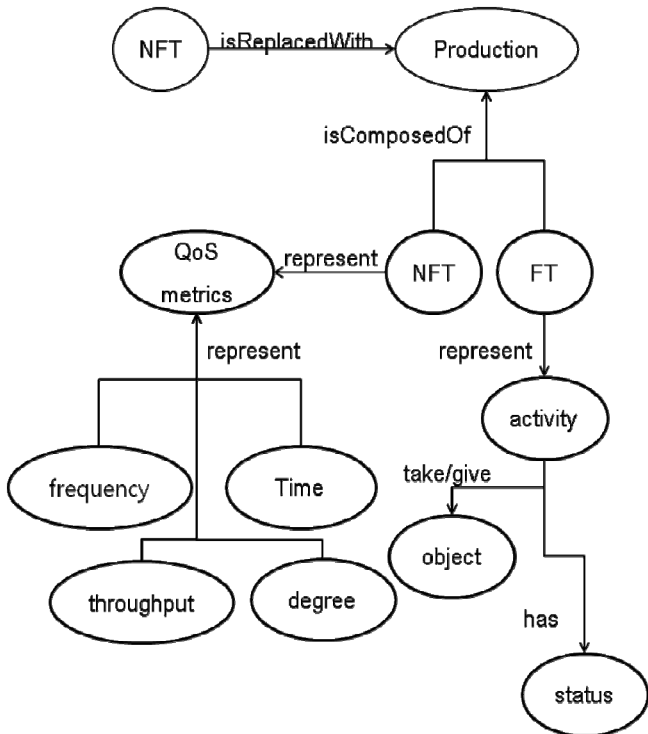
| service_empty | service_available | service_delay 등과 같이 정의될 때, 우측에 있는 요소들은 모두 비기능적 요소를 나타내는 비단말기호로 정의되며 D { service_full, service_empty, service_available, service_delay }와 같이 나타낸다. 메시지 생성시 해당하는 데이터가 없을 경우 null로 표현한다.

- AN 은 메시지의 annotation 을 의미하며 추가정보나 코멘트를 표현하는 공간이다. 추가내용이 없을 경우 null로 표시한다.

- AT 는 시스템의 action 들의 집합으로 특정 상황이 인지될 경우 시스템이 취해야 하는 action 을 말한다. 이러한 action 은 서비스 사용자에게 공개될 수 있고 아닐 수도 있다. 공개되지 않는 action 의 경우 메시지에는 null로 표현한다. action 은 규칙이 context 의 변화에 따라 새로운 규칙으로 치환되는 도중에 시스템간의 또는 시스템과 사용자간을 관리하기 위한 시스템 내부의 action 을 정의하기 위하여 사용된다.

3. 서비스 온톨로지

온톨로지(ontology)는 서비스 컴퓨팅 환경에서 개발 프로세스에 참여하는 서비스 또는 참여자들간에 공유하는 있는 정보를 정의하는데 사용되며 상호교환되는 메시지들을 grounding 하여 메시지의 실제 의미를 파악하는데 사용된다. 이러한 온톨로지는 다양한 언어로 구현되거나 표현되는데 RDF[8], OWL[9], SHOE[10]와 같은 언어로 표현되며 소프트웨어 개발과정에 생성, 사용되는 다양한 객체, 개념들을 표현하는데 사용한다.



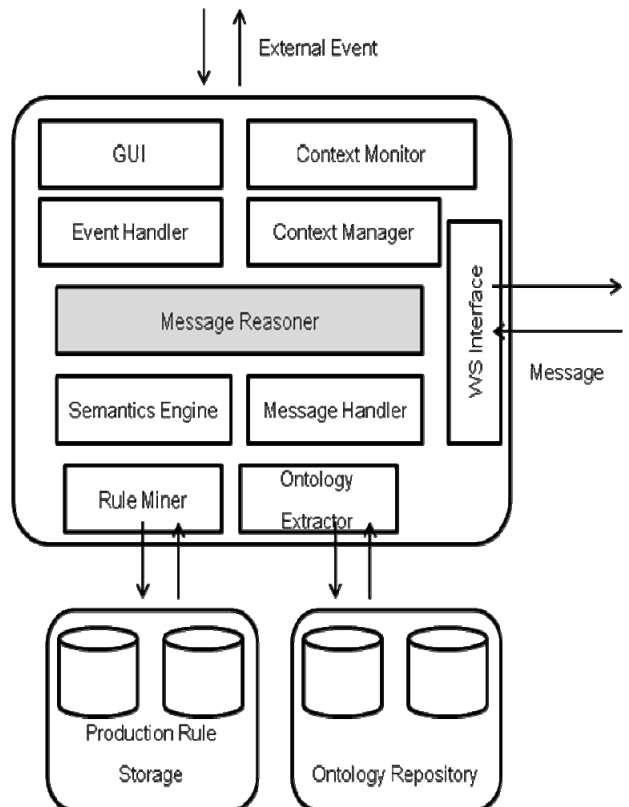
(그림 1) 대화형 모델을 위한 서비스 온톨로지

우리가 제안하는 시스템에서 온톨로지는 입력된 메

시지의 한 요소인 production rule 의 기능적 또는 비기능적 요소들을 grounding 하기 위하여 시스템에 사용된 온톨로지는 그림 1 과 같이 나타낸다. 그림 1 에서 보이다시피 production rule 은 NFT 와 FT 로 구성되며 NFT 는 QoS 관련 내용을 FT 는 기능적 요소들을 표현하고 있다. QoS 관련 부분은 수치로 표현되며 현재는 frequency, time, throughput, degree 로 구분되어 있다. 또는 FT 의 경우 실제 작업에 해당하는 activity 로 나타나 있으며 입출력 object 및 실행 subject 를 가질 수 있다.

4. 시스템 구성

본 시스템은 자바, Apache Axis 2 [11] 및 Semantic Web Service Framework (SWSF) [12] 기반으로 개발되었으며 그림 2 와 같은 구성을 가지고 있다. 시스템간 메시지 교환은 Web Service 기반으로 진행되며 Web Service Interface 모듈을 통해 메시지가 교환된다. Web Service Interface 는 Axis Framework 기반으로 작성되었으며 전달된 메시지는 Message handler 를 통해 intention 과 production rule, target 으로 파싱작업이 이루어진다. 파싱 작업후 각 token 은 Semantics Engine 으로 전달되는데 production rule 은 Rule Miner 와 Ontology Extractor 를 통하여 그 내용과 의미를 파악할 수 있도록 하였다. Message Reasoner 는 이러한 일련의 작업을 관리하는 모듈이며, 메시지의 의미는 context manager 를 통해 어플리케이션과 연결, 어플리케이션 custom 정보로 변환되며 Event Handler 와 Context Monitor 를 통해 사용자에게 전달되고 관리되도록 한다.



(그림 2) 시스템 아키텍처

5. 사례연구

소프트웨어 개발 프로세스 중에는 다양한 내용의 이벤트가 생성되며 이러한 이벤트의 처리를 요청하는 메시지들이 생성된다. 본 섹션에서는 이러한 이벤트가 어떤 형태로 본 시스템의 메시지 모델로 표현되는지, 또한 이러한 메시지가 어떻게 처리되는지를 보인다.

소프트웨어 개발 프로세스 중에 특정 사용자가 패스워드를 변환할 필요가 있다는 것을 시스템 관리자가 발견했을 경우, 패스워드를 변환해야 한다는 메시지를 보내야 하는데, 시스템 관리자는 “USER→<CHANGE PASSWD>”로 표현되는 production rule 을 찾아 시스템 관리자는 사용자에게 다음과 같은 메시지를 보낸다.

```
<action-request, CHANGE PASSWORD, USER>
```

이때 메시지 수신시스템의 rule extractor 는 메시지의 USER 에 대한 production rule 중에서 user 와 관련된 activity 중에서 change 와 password 에 대한 production rule 을 찾은 후, change 와 password 에 대한 의미를 온톨로지 repository 를 통해 찾으려 한다. 이 때 change password 에 대한 production rule 은 패스워드를 변화하기에 필요한 절차를 나타내며 “CHANGE_PASSWORD→DisplayWarning EnterOldPassword EnterNewPassword SaveNewPassword GetPasswordConfirmation” 와 같이 표현된다.

이때 사용자는 production rule 에 표현된 내용대로 차례로 실행을 하게 되는데 그 실행을 하기 위해서 DisplayWarning 에 대한 action 을 하게 되는데 password 에 대하여 DisplayWarning 에 대한 모듈이 사용자 application 내에 있다면 그 내용을 바로 실행시키고 아니라면 일반적인 warning display 모듈을 사용하도록 한다. 사용자 모듈에 해당 Production rule 을 실행할 수 있는 모듈이 있다면 해당 모듈을 실행시키도록 하는데 그렇지 않다면 reject 메시지를 보내도록 한다. Reject 에 대한 메시지 형식은 다음과 같다.

```
<reject, CHANGE PASSWORD, Not Available>
```

6. 결론

우리는 본 연구에서 semantic web 과 web service 를 통하여 소프트웨어 개발 프로세스 상에서 다양한 메시지가 생성, 교환, 인식 될 수 있도록 하는 시스템의 프로토타입을 구현하였다. 이를 위하여 우리는 메시지 교환 모델을 제안하였으며 메시지의 의미를 표현, 이해 할 수 있도록 하는 온톨로지를 제시하였다. 이러한 모델은 자바 및 apache axis2 를 통해 구현되었으며 사례 연구를 통해 우리 모델의 타당성을 보였다.

본 연구를 통하여 서비스 컴퓨팅의 패러다임이 서비스의 검색, 조합을 통해 복잡하고 심화된 서비스를 제공하는 것에 그치지 않고 그 응용범위를 복잡한 소프트웨어 사용환

경에의 적용으로 넓히는 것으로 확장될 것을 기대한다.

참고문헌

- [1] Ivona Brandic, Dejan Music, Schahram Dustdar. “Service Mediation Bootstrapping as First Achievements Towards Self-adaptable Grid and Cloud Services,” In Proc. Of GMAC’09, 2009.
- [2] I. Sommerville, *Software Engineering* 7th Edt., Addison Wesley.
- [3] L. O’Brien, P. Brebner, J. Gray, “Business Transformation to SOA,” In Proc. Of SDSOA’08, 2008.
- [4] D. R. Traum, J. F. Allen, “Discourse obligations in dialogue processes,” In proc. of the 32nd Annual Meeting of the Association for Computational Linguistics, pp 1-8, 1994.
- [5] D. Novick, “Control of Mixed-Initiative Discourse Through Meta-Locutionary Acts: A Computational Model,” Ph.D. thesis, University of Oregon, 1988.
- [6] J.Ginzburg, “Dynamics and the semantics of dialogue,” Logic, Language and Computation, Vol 1. CSLI publication.
- [7] J.L. Austin, “How to do things with Words,” Harvard University Press, MIT press, 1962.
- [8] G. Klyne and J.,J. Carroll, “Resource Description Framework (RDF): Concepts and Abstract Syntax,” <http://www.w3c.org/TF/rdf-concept/>, 2004
- [9] D.L. McGuinness and F.V. Harmelen, “OWL Web Ontology Language Overview,” <http://www.w3c.org/TF/owl-features/>, 2004.
- [10] S. Luke and J. Heflin., SHOE 1.01 Proposed Specification. SHOE Project 2000.
- [11] Apache Axis Framework, <http://ws.apache.org/axis/>
- [12] Semantic Web Services Framework, <http://www.w3.org/Submission/SWSF/>