

유비쿼터스 웹서비스 환경에서 이벤트 기반의 룰 처리 기법

이강찬* · 이원석* · 전중홍* · 이승윤* · 박종현**

*한국전자통신연구원, **서울대학교

Event based Rule Processing in Ubiquitous Web Services Environments

Kangchan Lee^{*} · Wonsuk Lee^{*} · Jonghong Jeon^{*} · Seungyun Lee^{*} · Jonghun Park^{**}

^{*}Electronics and Telecommunications Research Institute, ^{**}Seoul National University

E-mail : {chan, wslee, hollobit, syl}@etri.re.kr^{*} · jonghun@snu.ac.kr

ABSTRACT

웹서비스의 이벤트를 수신하기 위한 Web Services Eventing과 WS-Notification과 같이 현존하는 관련 표준은 하나의 이벤트 묶음을 무조건적으로 수신하거나 하나의 이벤트에 하나의 조건을 제시할 수밖에 없다는 것이 문제점으로 지적되고 있다. 그러나 실제로 이벤트들을 처리하기 위해서는 하나의 이벤트를 수신할 때도 다양한 조건을 제시할 수 있도록 하여, 동일한 이벤트를 받았을 때에도 다양한 조건식에 따라 원하는 작업을 수행할 수 있도록 하는 유연성을 제공하여야 한다. 본 논문은 그러한 유연성을 제공하는 유비쿼터스 웹서비스 환경에서 이벤트 기반의 룰 처리 기법에 대해서 살펴 보도록 한다.

키워드

Web Services, ECA rule, ubiquitous devices, action constraints

1. 서론

인터넷 및 웹의 급속한 발전에 힘입어, 최근 들어 다양한 형태의 정보 기술이 등장하게 되었다. 특히, 웹서비스는 1999년에 태동된 이후, 전 세계의 학계 및 산업계에서 차세대 서비스형 산업을 구현하기 위한 핵심 기술로 주목 받고 있으며, 이에 대한 연구가 다양한 분야에서 매우 빠르게 진행되고 있다. 웹서비스는 웹 상에서 모듈화된 소프트웨어 컴포넌트로서, 개방형 표준 데이터 표현 기법인 XML(Extensible Markup Language)과 인터넷 프로토콜을 결합시킨 새로운 패러다임에 의해서 탄생된 차세대 분산 컴퓨팅 기술이다. 특히, 웹서비스의 상호운용성(Interoperability)은 다양한 종류의 웹서비스들을 동적으로 발견하고 결합함으로써 부가가치를 가진 새로운 형태의 복합 웹서비스의 창출을 가능하게 하는 바, 웹서비스는 궁극적으로 기업 내, 기업 간은 물론 공공 기관 간의 프로세스 통합 및 협력 자동화를 이룩하기 위한 핵심 기술이라 할 수 있다. 웹서비스의 응용 분야는 매우 다양하여, 무역 분야, 전자 구매 분

야, 디지털 콘텐츠 분야뿐만 아니라[6], 기업 내 애플리케이션 간의 통합, 기업간 프로세스의 통합, 나아가서 전자 정부와 같은 공공 기관의 인터넷 서비스의 구축에도 성공적으로 이용될 수 있는 IT 도구이다. 또한 웹서비스의 상호운용성은 플랫폼, 언어 등 구현 환경에 독립적이기 때문에 비즈니스 분야뿐만 아니라 유비쿼터스(Ubiquitous) 환경을 구축, 운용하는데 있어서도 매우 효율적으로 사용될 수 있다[3][4].

1999년에 웹서비스가 태동된 이후, 웹서비스는 전 세계의 학계 및 산업계에서 차세대 서비스형 산업을 구현하기 위한 핵심 기술로 주목 받고 있으며, 이에 대한 연구가 다양한 분야에서 매우 빠르게 진행되고 있다. 특히, 사용자가 활용하고자 하는 웹서비스를 발견하고 사용하기 위해 다양한 표준안들이 제시되었는데, 이 중 현재 그 입지를 굳히고 있는 것이 WSDL(Web Services Description Language)과 UDDI(Universal Description, Discovery, and Integration)이다. WSDL과 UDDI는 현재 웹서비스 제공자가 웹서비스를 서비스 레지스트리에 등록하고, 웹서비스 사

용자가 서비스 레지스트리에서 원하는 웹서비스를 발견하는 과정에서 사용되고 있다.

하지만 UDDI는 유비쿼터스 웹서비스 환경에서 활용하기엔 많은 문제점을 안고 있다[5]. UDDI는 기본적으로 비즈니스 웹서비스를 기술하는데 초점을 맞추고 있기 때문에 웹서비스를 기술하는데 White Page 관련 정보(회사의 이름, 주소, 전화번호 등)와 Yellow Page 관련 정보(산업 분류, 웹서비스 분류, 지역별 회사 목록 등)를 요구한다. 하지만 이런 정보들은 개인이 자신의 사적인 공간에서 활용하는 유비쿼터스 웹서비스를 기술하기에 적합하지 않다. 또한 UDDI와 WSDL에 기술되는 웹서비스 정보들을 다루는 메커니즘은 정적이기 때문에, 웹서비스 제공자가 웹서비스를 공개할 때 작성한 정보 이외의 정보들을 얻을 수 없다. 일례로, 웹서비스의 endpoint 정보나 현재 웹서비스의 정책이나, 수용 능력, 상태와 같이 사용자에게 따라 수시로 바뀔 수 있는 정보들을 기술하거나 수정할 수 없기 때문에, 사용자가 UDDI와 WSDL을 이용해 원하는 웹서비스를 발견했다고 해도 그것이 최신 정보인지, 웹서비스가 올바르게 작동하고 있는지, 현재 자신이 웹서비스를 활용할 수 있는지에 대해 보장받을 수 없다. 또한 UDDI를 이용해 서비스 레지스트리에서 웹서비스를 중앙 집중적으로 관리하는 경우, 유비쿼터스 웹서비스는 endpoint만 다르고 동일한 서비스를 제공하는 경우가 많기 때문에 모든 정보를 관리하는데 비효율적이며 그 한계가 있다. 또한 사용자 개개인의 프라이버시 문제와 특정 사업자의 파워 게임을 막을 방법이 없다.

따라서 본 연구에서는 중앙 집중적이고 정적인 정보만 다룰 수 있는 UDDI와 WSDL의 한계를 극복하기 위해선 1) 서비스의 발견이 사용자의 위치, 환경, 필요에 따라 동적으로 이루어져야 하며, 2) 서비스가 자신의 정보, 상태 등을 사용자에게 스스로 알릴 수 있어야 한다. 3) 또한 사용자가 서비스를 쉽게 호출해서 사용할 수 있어야 하고, 4) 사용자가 원하는 방식으로 가용한 서비스를 조합하여 사용할 수 있어야 한다. 우리는 이러한 조건들을 만족시키기 위해 ECA 기반의 룰 기술 언어를 정의하고 이것을 바탕으로 보다 효율적으로 웹서비스들을 활용할 수 있는 방법을 제안하고자 한다.

II. ECA 기반의 룰 기술 언어 활용 기반의 네트워크

ECA[7] 기반의 룰 기술 언어를 활용하기 위해선 기존의 웹서비스 사용 구조 (service provider, service repository, service requestor)를 확장, 변화시킬 필요가 있다.

이 네트워크는 논리적으로 서비스 요청자(service requester), 서비스 제공자(service provider), 이벤트 소스(event source), ECA 룰 시스템(Rule

System - Device, Global Rule Manager)의 4가지 역할자가 필요하다.

- 서비스 요청자 : 웹서비스를 사용하려는 클라이언트로, ECA 기반의 룰을 정의하고, 서비스를 활용
- 서비스 제공자 : 일반적인 웹서비스를 의미하며, 이것은 이전의 웹서비스 활용 구조에서의 서비스 제공자와 동일한 역할을 수행
- 이벤트 소스 : 웹서비스 중에서도 자신의 상태 변화를 이벤트 알림 메시지(event notification message)를 통해 전달하는 것을 의미
- ECA 룰 시스템 : ECA 기반의 룰을 저장하고, 이벤트 소스로부터 이벤트 알림 메시지(를 받으며 (즉, event sink의 역할을 수행한다.) 해당 이벤트에 맞는 룰을 실행하는 역할을 수행

ECA 기반의 룰 기술 언어를 정의함으로써 현재 가능하지 않거나, 비표준적으로 수행되는 다음과 같은 작업들을 수행할 수 있게 된다.

2.1 이벤트의 조건적 수신/반응 (event filtering/conditional response)

웹서비스의 이벤트를 수신하기 위한 Web Services Eventing[2]과 WS-Notification과 같은 표준들의 한계는, 하나의 이벤트 묶음을 무조건적으로 수신하거나 하나의 이벤트에 하나의 조건을 제시할 수 밖에 없다는 것이다. ECA 기반의 룰 기술 언어에서는 하나의 이벤트를 수신할 때도 다양한 조건을 제시할 수 있도록 하여, 동일한 이벤트를 받았을 때에도 다양한 조건식에 따라 원하는 작업을 수행할 수 있도록 하는 유연성을 제공한다. 다음의 예에서, event e1을 받았을 때, 조건식 (c1, c2)에 따라 각기 다른 액션 (a1, a2)을 수행하는 모습을 확인할 수 있다.

```
on e1 if c1 do a1
on e1 if c2 do a2
```

2.2 Event forwarding/broadcasting/multicasting

ECA 기반의 룰 기술 언어는 웹서비스들이 주고 받는 이벤트 알림 메시지를 기반으로 하고 있기 때문에 이벤트 알림 메시지에 효율적으로 접근하고, 활용할 수 있는 방법들을 제공하고 있다. 그 중 하나가 event passing으로, 이것은 event sink가 받은 이벤트 알림 메시지를 원하는 곳으로 전달할 수 있도록 한다. 이 기능을 이용하면 event sink가 받은 이벤트 알림 메시지를 특정한 하나의 위치로 전달(forwarding)하거나, 특정 다수로 멀티캐스팅(multicasting) 할 수도 있고, 임의의 다수에게 브로드캐스팅(broadcasting)할 수도 있다.

2.3 논리적 규칙의 표현 (conjunction, disjunction, negation)

이벤트, 조건, 액션의 기술에 있어 논리 연산자

(e.g. AND, OR 등)를 활용하여 다양한 논리적 규칙을 거의 무제한적으로 표현할 수 있게 된다.

2.4 추이 특성(Transitive property)

ECA 기반의 룰 기술 언어에서는 내부 이벤트(internal event) 개념을 도입함으로써 룰에 추이 특성을 부여했다. 추이 특성을 가짐으로써 룰 간의 관계가 발생될 수 있으며, 이 관계를 통해 각각 독립적으로 의미 있는 룰들이 하나의 거대한 목적을 향해 조직적으로 작업을 수행할 수 있게 된다. 다음의 예는 event e1이 발생했을 때 조건 c1을 만족하면 새로운 event e2를 기반으로 정의된 새로운 룰과 조합되는 모습을 보여주고 있다.

```
on e1 if c1 do a1(e1)
on e1^e2 if c2 do a2
```

ECA 기반의 룰 기술 언어는 다음에서 정의하는 이벤트 관련 정보를 얻을 수 있는 방법을 제공한다. 이벤트 알림 메시지를 주고 받는 메시지 프로토콜에 독립적으로 사용된다. 단, 이 보고서에서는 이벤트 알림 메시지를 주고 받는 메시지 프로토콜인 Web Services eventing과 WS-Notification 중에서 Web Services Addressing을 기반으로 정의되어 우리에게 필요한 정보를 제공하고 있는 Web services eventing을 사용한다고 가정하고 있다. 또한, ECA 기반의 룰 기술 언어와 비슷한 목적으로 가지고 있는 WS-BPEL (Web Services Business Process Execution Language)과 SCXML (State Chart XML)[1]과 비교해 다음의 차이를 가리게 된다.

표 1. WS-BPEL, SCXML, WS-ECA 비교

	WS-BPEL	SCXML	WS-ECA
적용 범위	외부 서비스	내부 시스템	외부 서비스
상태	상태 전이(instance 생성)	상태 전이	비유지
목적	요청에 의한 서비스 처리 과정을 정의, 일련의 서비스 구성(choreography)	외부 이벤트에 의한 내부 시스템의 상태 변화를 통제(control)	외부 이벤트들에 의한 외부 시스템의 서비스 제재(coordination)
설명	한 WS-BPEL은 하나의 비즈니스 로직을 가짐	한 SCXML은 하나의 시스템 상태 변화물 묘사	한 WS-ECA는 한 사람 주변에서 일어나는 서비스 운영 매커니즘을 정의

II. ECA 기반의 룰 기술 언어

ECA 기반의 룰 기술 언어는 웹서비스를 복합해서 활용하는 과정을 이벤트(event), 조건(condition), 액션(action)으로 구분하여 사용자가 원하는 방식으로 웹서비스를 복합적으로 활용할

수 있는 방법을 XML을 이용하여 기술하도록 하는 언어이다. ECA 기반의 룰 기술 언어는 기본적으로 다음과 같은 포맷을 갖는다.

```
on event
if condition
do action
```

즉, 특정한 event가 발생하면 condition을 검사하고, 이 조건을 만족하는 경우 action을 수행하도록 한다.

3.1 이벤트(event)

ECA 기반의 룰 기술 언어에서 웹서비스 기반의 이벤트는 'event notification', 'internal event'와 'temporal event'의 집합으로 정의된다. Event notification은 Event source가 자신에게 발생한 이벤트를 event sink에게 전달하는 event message로, event sink는 이 메시지에서 필요한 정보를 이용해 자신이 관심 있어 하는 이벤트가 발생했는지 알 수 있다. Internal event는 ECA 기반의 룰 기술 언어 내부에서 정의되는 event이며, temporal event는 사용자가 지정한 특정 시각이 만족된 경우를 뜻한다. 이 시각은 특정 시각을 나타내거나 (in time), 주기적으로 계속 반복되는 시간일 수도 있고 (periodically), 다른 이벤트 발생에 상대적으로 (relative to other event) 정의되기도 한다.

이벤트들을 결합하기 위해 사용하는 논리 연산자에는 conjunction, sequential, disjunction, negation의 4가지가 존재한다. Conjunction은 'n개의 이벤트가 주어진 time interval 동안 일어나면' 참이다. 이때 n개의 이벤트들의 발생 순서는 무시된다. Sequential은 conjunction과 마찬가지로 'n개의 이벤트가 주어진 time interval 동안 일어나면' 참인데, 이때 n개의 이벤트들은 주어진 순서대로 일어나야 한다. Disjunction은 'n개의 이벤트 중 하나가 발생하면' 무조건 참이며, negation은 특정한 이벤트가 주어진 time interval 동안 발생하지 않는 것을 의미한다.

3.2 조건(condition)

조건은 웹서비스가 특정 작업을 수행하기 전에 그 작업을 수행해야 하는지 의사 결정을 할 수 있도록 하며, XPath의 boolean expression으로 표현된다.

3.3 액션(action)

이벤트가 발생하고, 조건을 만족하는 경우 웹서비스는 특정 작업을 수행하게 되는데, 이 작업을 액션이라 한다. ECA 기반의 룰 정의 언어에서 정의하고 있는 액션은 assign, delay, invoke, pass event, create internal event, create external event의 6가지로 구성된다. Assign은 정의된

variable에 특정한 값을 대입하는 것을 말하며, delay는 주어진 시간 동안 아무런 작업도 수행하지 않는다. invoke는 service provider가 제공하는 서비스를 호출하는 것을 의미하며, pass event는 event notification message를 다른 event sink로 전달해주는 것을 말한다. create internal event는 internal event를 호출하는 것이고, create external event는 시스템 외부로 event notification message를 전달한다.

Action의 논리적 조합을 표현하기 위해 conjunction, sequential, disjunction의 3가지 연산자가 사용된다. Conjunction은 'n개의 액션을 동시에 수행하는 것'을 의미하는데 여기에는 1) n개의 액션에서 invoke, pass event, create internal event와 create external event에서 input argument로 사용되는 변수로의 assign, 2) delay 액션은 존재할 수 없다. Sequential은 'n개의 액션을 순서대로 수행하는 것'을 뜻하며, disjunction은 'n개의 액션 중 하나를 수행하는 것'을 의미한다. Disjunction에서는 액션에 priority를 두어 어떤 액션의 수행을 먼저 시도할 것인지 지정할 수 있으며, 이 메커니즘을 이용해서 error handling을 수행할 수도 있다.

IV. 결론

본 논문에서는 웹서비스들의 이벤트를 처리하기 위해서는 하나의 이벤트를 수신할 때도 다양한 조건을 제시할 수 있도록 하여, 동일한 이벤트를 받았을 때에도 다양한 조건식에 따라 원하는 작업을 수행할 수 있도록 하는 유연성을 제공하여야 하는 WS-ECA를 위한 웹서비스 확장 구조와 ECA의 주요 컴포넌트에 대해서 살펴보았다. 향후, 본 연구는 구체적인 ECA 기반의 룰 기술 언어 정의와 함께 룰 시스템에 대한 구현이 수반되어야 한다.

또한, 룰들은 다양한 디바이스에 저장되면서, 또는 다중 사용자에 의하여 룰이 정의되면서 서로 상충되거나 사용자가 원하지 않는 결과를 가져올 수가 있다. 이를 방지하기 위하여 룰 충돌 정의 및 충돌 시 이를 해결하는 메커니즘의 개발이 추가로 필요하다.

References

[1]A. Friday, N. Davies, N. Wallbank, E. Catterall, and S. Pink. Supporting Service Discovery, Querying and Interaction in Ubiquitous Computing Environments. *Wireless Networks* 10:631-641, 2004.
 [2]Y. Huang, H. Garcia-Molina. Publish/Subscribe in a Mobile Environment. *Wireless Networks* 10:643-652, 2004.
 [3]S. Vinoski. Integration with Web Services.

IEEE internet computing, 7(6): 75-77, 2003.
 [4]A. Carter and M. Vukovic. A Framework For Ubiquitous Web Service Discovery. In Proc. of the 6th UbiComp, 2004.
 [5]A. Sashima, N. Izumi, and K. Kurumatani. Location-Mediated Coordination of Web Services in Ubiquitous Computing, in Proc. of IEEE Int'l Conf. Web Services (ICWS'04), pages 109-114, 2004.
 [6]N. Bassiliades, and I. Vlahavas. DEVICE: Compiling production rules into event-driven rules using complex events. *Information and Software Technology*, 39:331-342, 1997.
 [7]K. Liu, L. Sun, A. Dix, and M. Narasipuram. Norm Based Agency for Designing Collaborative Systems. *Information Systems Journal*, 11(3): 229-247, 2001.
 [8]S. Calo, and M. Sloman, Policy-Based Management of Net-works and Services. *Journal of Network and Systems Management*. 11(3):249-252, 2003.
 [9]J. Lobo, R. Bhatia, and S. Nagvi. A Policy Description Language. In Proc. of National Conference of the American Association for Artificial Intelligence, Orlando, FL, 1999.
 [10]M. Cilia and A. Buchmann. An Active Functionality Service for E-Business Applications. *ACM SIGMOD Record*, 31(1): 24-30, 2002.