XSchema 를 이용한 EJB 배치설명파일의 확장 방안 연구

o 공재원, 심우곤, 백인섭
아주대학교 정보통신공학과
{jwgong, startear, ispaik}@madang.ajou.ac.kr

(A Study of Extension of the EJB Deployment Descriptor File with XSchema)

o Jae-Won Gong, Woo-Gon Shim, In-Sup Paik
*Graduate School of Information and Communication, Ajou University.

요 약

컴포넌트는 소프트웨어 재사용의 핵심 기술로 인식되고 있으며, 현재 많은 수의 컴포넌트가 개발되고 사용되고 있다[9]. 많은 수의 컴포넌트들 중에서 특정 도메인에 이미 적절하다고 판단된 컴포넌트를 검색해서 사용하는 과정이 필수적이며[6], 이를 위해서 컴포넌트에 대한 정확한 명세서가 뒷받침되어야 한다. 본 논문에서 다루고 있는 컴포넌트 모델의 하나인 썬(Sun)社의 EJB ver1.1 은 배치설명파일(Deployment Descriptor)을 XML 로 기술하고 있으며, DTD 로 Validation 체크를 하고 있다. 그러나 DTD 는 표현할 수 있는 데이터 타입에서 한계를 가지며 하나의 XML 은 여러 개의 DTD 파일을 가질 수 없기 때문에 확장성에서도 취약함을 나타낸다. 이를 해결하기 위해서 XSchema 로 변환하였다. 또한 현재 EJB 의 배치설명파일에서는 컴포넌트의 결합 및 의존성에 대한 표현이 부족하기 때문에 이를 보완하기 위해서 컴포넌트 Contract 에 대한 속성들을 기반으로 하여 새로운 태그를 지정해보도록 한다.

1. 서 론

컴포넌트는 바이너리 코드 형태로 개발되고 있는 소프트웨어로서, 쉽게 바인딩되어 수행될 수 있으며 독립적으로 재사용 가능한 소프트웨어 단위를 말한다[9]. 컴포넌트 이용의 중요성에 대해서는 이미 많은 사람들이 인식 하고 있고 컴포넌트에 대한 이용도도 점차 커지고 있다. 하지만 아직 컴포넌트의 표준화된 모델이 제시된 것이 없으며 컴포넌트의 명세에 대해서도 마찬가지이다. 이 때문에 현재 컴포넌트 시장은 크게 썬(Sun)社의 JavaBeans/Enterprise JavaBeans(EJB), 마이크로소프트(MS)社의 COM/DCOM, ActiveX/OLE, OMG 그룹의 CORBA 등이 각자 업계의 표준을 가지고 공존하고 있는 실정이다.

이러한 수많은 컴포넌트 중에서 가장 적합한 컴포넌트가 적시 적소에 사용될 수 있기 위해서는 컴포넌트에 검색이 쉽고 정확하게 이뤄져야 하며, 이를 위해서는 표준화된 컴포넌트에 대한 명세화가 뒷받침 되어야 한다. 각 컴포넌트 모델에서는 배치(Deployment)를 위해서 다양한 방법을 사용하고 있는데, JavaBeans 의 경우에는 BeanInfo 클래스를 통해서 기술을 하고 있으며, EJB1.1 의 경우에는 따로 XML(ejb-jar.xml)파일을 생성하여 기술 하고 있다. COM/DCOM 의 경우에는 OSD 를 자체 확장하여 사용하고 있고, CORBA 의 경우에는 Component Descriptor Language(CDL)을 이용하여 배치를 하고 있다.

본 논문에서는 여러 컴포넌트 모델 중 썬(Sun)社의 EJB1.1 초점을 맞추었으며, EJB 의 배치설명파일(Deployement descriptor)인 XML 파일이 validation 검사를 위해 사용하고 있는 DTD 를 Xschema 로 확장하는 작업을 수행하고, 컴포넌트의 인터페이스를 표현하는 데에 있어서 EJB 배치설명 파일에서 미비하게 표현되고 있는 부분을 보완하는 방법을 제안한다.

2. Enterprise JavaBeans(EJB)

EJB 는 엔터프라이즈급 어플리케이션의 개발 및 분산 배치를 위한 서버측 분산객체 컴포넌트 아키텍처이다[8]. J2EE 애플리케이션 내에서 보여주기 로직을 주로 담당하는 것이 JSP, Servlet 이라고 한다면 비즈니스 로직을 담당하는 것이 바로 EJB 이다[8].

개발자 입장에서 하나의 EJB 를 만들기 위해서는 Client Interface 부분(*Home Interface/Remote Interface* 로 나뉜다)과 Enterprise Bean Class, Primary Key Class 파일, 그리고 본 고에서 다룰 배치설명파일(Deployment Descriptor File)을 구성한다.

2.1 배치설명파일 (Deployment Descriptor)

EJB 는 분산객체이므로 분산 환경에서 필요한 통신에 관련된 코드와 EJB 컨테이너가 제공하는 각종 서비스에 관련된 코드가 런타임시에 각각의 빈클래스에 어떻게 적용되는지에

대한 정보를 갖고 있어야 한다. 이를 위해 사용되는 것이 배치설명파일(Deployment Descriptor)이다[4].

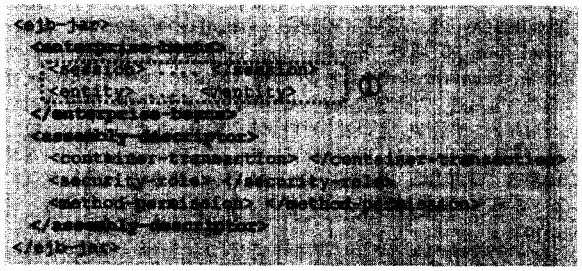

그림 1. 배치설명파일의 기본 구조

위 그림 1 에서 보는 것과 같이, EJB 배치설명파일은 <ejb-jar> 루트태그 밑에 크게<enterprise-beans>와 <assembly-descriptor> 두 개의 내용을 포함한다. <enterprise-beans>는 그림 1 의 ① 부분처럼 <session>빈 혹은 <entity>빈의 내용을 포함하며, <session> 혹은 <entity>태그 밑에는 해당 빈이 사용하는 다른 빈에 대한 정보와 빈이 접근해야하는 DB 접속과 같은 외부 자원에 대한 정보 및 해당 빈을 사용할 수 있는 role 의 이름 들의 목록이 포함된다. <assembly-descriptor>는 빈에 접근할 때에 사용되는 보안 역할에 대한 정보와 빈에 기술된 메소드들의 목록, 그리고 특정 트랜잭션 속성이 어떤 메소드에 적용되는지에 대한 내용을 포함한다. 이러한 배치설명파일은 " ejb-jar.xml" 이름으로 저장 된다.

3. DTD 와 XSchema

XML 문서의 validation 체크를 위해서 주로 Extended BNF 형식으로 기술이 되는 DTD 가 사용되어왔으나 DTD 는 데이터 타입에서 상당히 제한적이며(#PCDATA 타입만 가능) Element 의 반복 횟수에 대한 정확한 표기가 불가능(0 개이상, 1 개 이상, 1 개, 0 개/1 개 이렇게 4 가지 표현만 가능)하다는 점에서 세밀하게 문서의 Validation 을 검사하기가 어려웠다. 또한 여러 개의 Namespace 를 가질 수 없다는 점에서 확장에 많은 어려움이 있었다. 한편, 2000 년 5 월에 Recommend 된 Xschema 의 경우에는 XML 의 Validataion 을 검사한다는 DTD 의 기본적인 성격을 따르면서 XML 의 형식을 그대로 따르고 있다. Xschema 는 다양한 데이터타입과 횟수의 표시, 여러 개의 Namespace 를 사용할 수 있다는 점에서 DTD 의 확장판으로 주목을 받고 있다.

4. XSchema 를 이용한 배치설명파일의 작성

그림 2 는 현재 EJB 배치설명파일의 DTD 파일에서 Method 부분만을 가져온 것이고, 그림 3 은 이 DTD 파일을 XSchema 로 변환한 것이다. XSchema 는 자신만의 데이터 타입을 지정할 수 있고(ex. <xsd:complexType>), 사용되는 Element 의 회수를 정확하게 명시할 수 있다는 것이 (ex. minOccurs, maxOccurs) DTD 와 가장 크게 구별 되는 점이다.

컴포넌트의 의존성을 파악하고 컴포넌트 통합을 위해서는 Extra-Funcional 속성(성능, 정확성, 가능성, 잠재성, 보안성)을 나타내야 한다.이를 위한 API 를 확장할 필요가 있는데[2], Beugnard 는 이를 위해서 Syntactic, Behavioral,

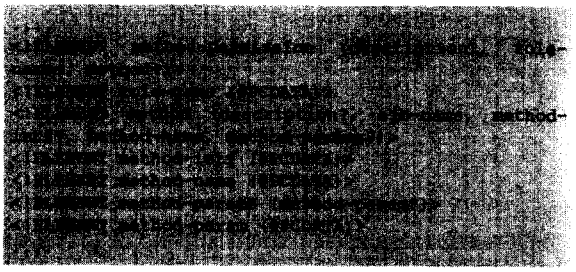

그림 2. 기존 EJB1.1 에서 제공되는 DTD 파일의 일부

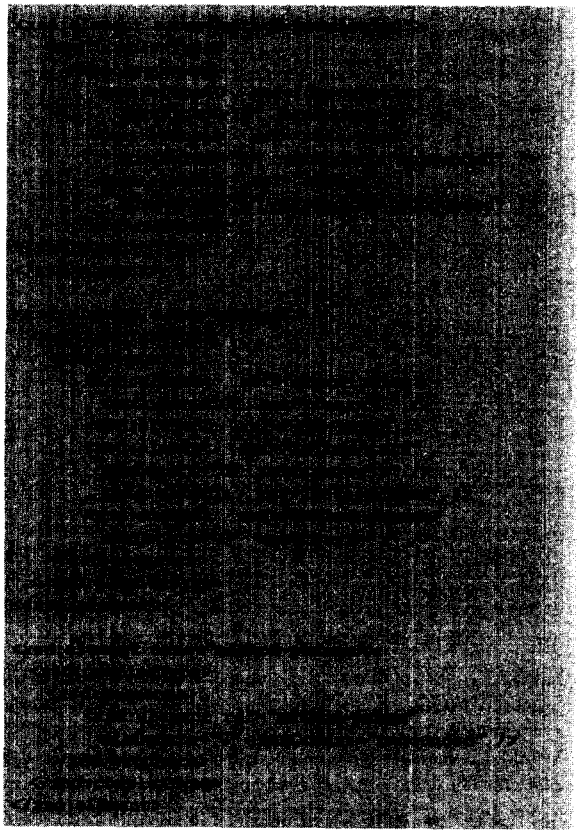

그림 3. 기존 DTD 를 Xschema 로 변환한 xsd 파일

Synchronization, Quality-Of-Service 의 4 가지 속성을 제안하였다[1]. 현재 EJB 의 배치설명파일에서는 함수에 사용되는 input 파라미터의 내용만을 기술하고 있으나, IDL 의 표현 방식과 같이 파라미터의 In/Out/InOut 속성을 표현하고 있는 형태를 이용하여 그림 4 의 ①에서 보이는 것처럼 <param-in>, <param-out>에 대한 포함하도록 확장하였다. 또한 그림 4 의 ②에 해당하는 <method-behavior> 에 대한 정보는 Beugnard 의 Level2 에 해당하는 내용으로써 pre/post condition 의 정보를 기록하는 데에 사용된다. 이를 위해서 <require>, <ensure>를 선언하도록 한다. 그림 4 의 ③에 해당하는 <method-sync>에 대한 부분은 Beugnard 의

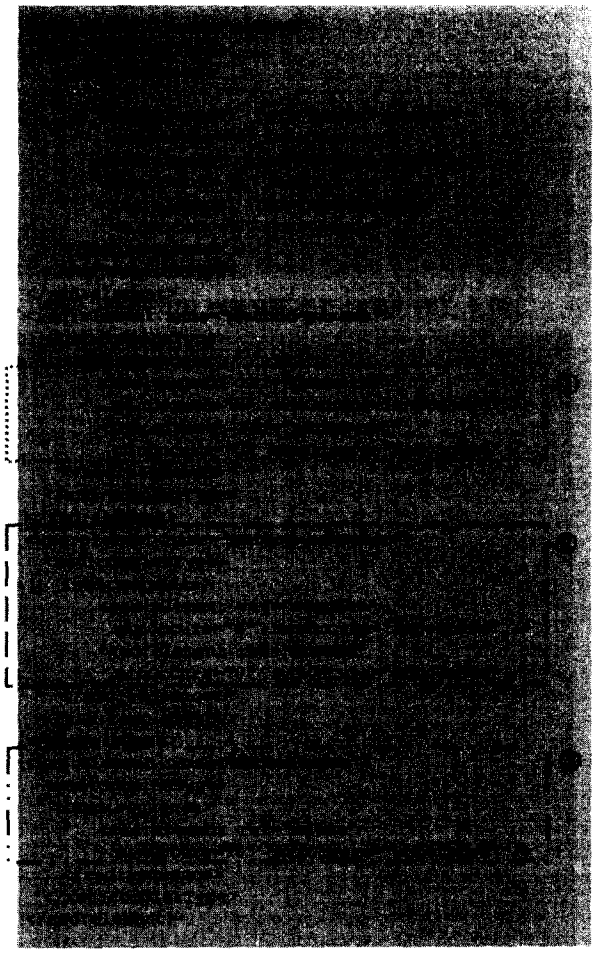

그림 4. Xschema 에 기반한 확장된 xsd 파일

Level3 에 해당하는 내용으로써 서비스들간의 Synchronization 을 맞춰야 하는 경우를 위해서 mutual exclusive(Mutex)의 상태를 만들어둔 것이다. Beugnard 의 Level4 인 Quality-Of-Service 부분에 대한 내용은 본 고에서는 생략하였다.

5. 결론 및 향후 연구과제

현재 EJB 의 배치설명파일에는 결합 및 컴포넌트 의존성에 대한 내용이 부족한 상태이며 DTD 의 한계 때문에 정확한 표기에서도 많은 취약함을 나타내고 있다.

본 논문에서는 이러한 DTD 가 갖는 제약 사항들 중에서 여러가지 형태의 데이터 타입은 XSchema 에서 제시하고 있는 <xsd:complexType>, <xsd:simpleType>등과 같이 개발자가 자체적으로 정의한 데이터 타입과 boolean, date 등의 데이터 타입을 통하여 보완을 하였다. Element 의 등장 횟수에 대해서도 minOccurs, maxOccurs 에 대한 속성을 사용하여 배치설명파일이 좀더 정확한 내용을 표현할 수 있도록 확장하였다.

한편, 기존 EJB1.1 의 명세에서 결여되어있는 인터페이스의 behavioral, synchronization 에 대한 정보를 표현하기 위해서 <method-behavior> 타입을 새로 지정하여 pre-condition 에 대한 내용은 <require>태그에서, post-condition 에 대한 내용은 <ensure>태그에서 표현하도록 하였다. Behavioral 의 표현을 위해 require 정보와 ensure 에 대한 정보를 기록함으로써 순차적인 프로그램 환경에서의 신뢰도의 수준을 올리게 된다[1]. 또한 <method-sync>타입을 새로 만들고 그 타입에서는 Mutual Exclusivce 에 대한 내용을 나타내기 위해서 <mutex>라는 태그를 포함시켰다. 이 표현은 분산환경에서의 신뢰도를 높이는데에 기여를 한다[1].

본 논문에서 언급된 EJB 의 배치설명 파일 외에도 컴포넌트 모델에서 많이 사용하고 있는 CORBA 와 COM 의 명세 언어에 대해서도 정확하게 기술 하기 위해서 XSchema 를 통해서 확장 표현하는 일을 현재 진행중이다.

참고문헌

[1] A.Beugnard, J.M.Jezequel, N.Plouzeau, D.Watkins, "Making Components Contract Aware", IEEE Computer, 32(7), pp.38~45, July 1999.

[2] F.Bachman, L.Bass, C.Buhman, S.comella-Dorda, F.Long, J.Robert, R.Seacord, K.Wallnau, "VolumeII : Technical Concepts of Component-Based Software Engineering", Carnegie Mellon Software Engineering Institute, CMU/SEI-2000-TR-008, 2000.

[3] N.H.Lassing, D.B.B.Rijsenbrij, J.C.van Vliet, "A View on Components", Proc. Of 9th Int.Workshop on Database and Expert Sys. Applications, pp.768-777, 1998.

[4] H.Richard, "Enterprise JavaBeans", O'reilly, 2000.

[5] L.Roger Costello, "XML Schema", XML Technologies Course, 2001.

[6] "Enterprise JavaBeans v1.1, Final Release", Sun Microsystem Inc., pp.239 ~ 269, 1999.

[7] "XML Schema Part 0:Primer", W3C, 2001.

[8] "XML Schema Part 1:Structure", W3C, 2001.

[9] "XML Schema Part 2:DataTypes", W3C, 2001.

[10] 김세곤, 서창근, 김민식, "EJB Bible", 정보문화사, 2001.

[11] 박서영, 신영길, 우치수, "XML 컴포넌트 명세서 기반의 컴포넌트 검색 기법", 정보과학회논문지 :소프트웨어 및 응용 제 27 권 제 2 호, pp.180~192, 2000.