

아키텍처 스타일 기반의 컴포넌트 조립 및 지원도구의 개발

이승연⁰ 이지현 권오천 신규상
한국전자통신연구원, 컴포넌트공학 연구팀
{coral, jihyun, ockwon, gsshin}@etri.re.kr

Component Assembly Based on an Architecture Style and Development of a CASE Tool

Seung-Yun Lee⁰ Ji-Hyun Lee Oh-Cheon Kwon Gyu-Sang Shin
Component Engineering Research Team, ETRI

요 약

다양한 응용 소프트웨어를 개발함에 있어 컴포넌트 기반의 개발기술이 빠르게 발전하고 있는 가운데, 실제 개발된 컴포넌트들을 재사용하고, 이들을 조립하여 커다란 시스템을 개발하려는 노력이 꾸준히 있어왔다. 컴포넌트의 개발과 컴포넌트 조립을 통한 응용 소프트웨어 시스템 개발을 위해서는 이들이 서로 정확하게 결합하여 작동할 수 있는 아키텍처 기반의 조립환경이 지원되어야 한다. 본 논문은 Enterprise JavaBeans (EJB)의 조립을 지원하는 아키텍처 기술언어 및 이의 지원도구에 대하여 기술한다. 아키텍처 스타일의 하나인 Chiron-2 (C2) 스타일을 기반으로 아키텍처를 기술하고, 이를 다이어그램 형식과 텍스트 형식으로 표현할 수 있는 아키텍처 모델러의 구조 및 이의 프로토타입에 대해 설명한다.

1. 서론

복잡한 응용 프로그램을 빠르게 개발하고 이의 유지보수가 용이하도록 하기 위해 재사용 가능한 컴포넌트 개념이 대두되었고, 이와 관련되어 EJB(Enterprise JavaBeans), COM+, WaterBeans[1]와 같은 다양한 컴포넌트 모델과 이의 개발 및 응용에 관한 연구가 다양하게 진행되고 있다. 그러나, 다양한 컴포넌트 모델에 의해 설계되고 여러 벤더들로부터 개발된 컴포넌트를 조립한다는 것은 쉬운 일이 아니다. "Write once, run everywhere"이라는 취지아래 개발된 EJB 기반의 컴포넌트의 경우, 한번 개발된 컴포넌트는 다양한 어플리케이션을 개발하고 다양한 플랫폼에 전개될 수 있어야 하지만, 이를 위해서는 컴포넌트 간의 조립 뿐만 아니라, 컴포넌트와 프레임워크, 조립된 컴포넌트들의 상위 레벨에서의 조립 등, 다양한 조립을 지원해야 하기 때문에, EJB 컴포넌트간의 조립조차 간단히 지원되지 않는다.

본 논문은 컴포넌트의 개발과 컴포넌트 조립에 의한 응용 소프트웨어의 개발을 효과적으로 지원하기 위하여 컴포넌트들이 서로 정확하게 결합하여 작동할 수 있는 아키텍처를 기반으로 컴포넌트 조립을 지원하는 방법과 이의 CASE 도구의 프로토타입을 기술한다. EJB의 조립을 지원하는 아키텍처 기술언어를 Chiron-2(C2)[2] 아키텍처 스타일에 기반하여 정의하고 이를 다이어그램 형식과 텍스트 형식으로 기술하고 구문, 타입 및 형세의 모델검사를 지원하는 아키텍처 모델러의 프로토타입에 대해 설명한다.

2. 관련연구

2.1 컴포넌트 조립

컴포넌트 조립(Component Assembly)은 컴포넌트 기반 소프트웨어 개발(Component Based Software Development: CBSD)의 궁극적 목표로서, 소프트웨어를 구성하는 단위 컴포넌트들은 조립을 통하여 다른 컴포넌트들과 상호 작용한다. CBSD에 있어 컴포넌트 조립은 같은 모델을 기반으로 개발된 단위 컴포넌트간이나 이종 모델 컴포넌트간의 어플리케이션 레벨 조립과, 이종 서버에 전개되어 있는 컴포넌트간의 시스템 레벨 조립, 그리고 조립된 컴포넌트들간의 상위레벨에서의 조립 또한 포함된다[3].

이러한 조립을 지원하기 위해서 컴포넌트는 인터페이스(Interface)를 기반으로 결합된다. 인터페이스는 정보 은닉(Information Hiding) 원리에 따라 컴포넌트를 자유롭게 대체하고 독립적으로 사용할 수 있도록 지원한다. 그러나, EJB나 COM+ 와 같은 컴포넌트 모델에서의 인터페이스는 컴포넌트의 모듈 기능에 대한 표현은 제공하나, 컴포넌트의 비기능적인 속성들의 완벽한 지원은 제공하지 못한다. 또한, JavaBeans나 WaterBeans에서 표현되는 컴포넌트의 입출력 흐름이 제공되지 않고, 'Plug-and-Play' 방식의 손쉬운 조립을 행하는 데에도 많은 제약이 있다. 이를 극복하기 위하여 아키텍처 레벨에서 정의하는 추상화된 인터페이스를 제공해야 하며, 컴포넌트간의 메소드 호출방식

(Method Invocation)이 아닌 메시지 전달방식(Message Passing)을 통하여 컴포넌트 조립을 쉽게 지원할 수 있어야 한다.

2.2 아키텍처 스타일(Architectural Style)

아키텍처 스타일은 자주 사용되는 관용적인 소프트웨어 시스템의 구조 패턴을 의미하며 컴포넌트와 커넥터들로 모델링된다. 아키텍처 스타일은 구조적인 구성 패턴을 공유하는 시스템들의 집단을 정의하며, 아키텍처의 설계에 사용되는 컴포넌트 타입들, 커넥터 타입들, 그리고 이들의 상호 연결 패턴에 대한 규칙들과 제약 조건들을 정의한다 [4]. 아키텍처 스타일의 예로는 작업이나 데이터의 흐름을 나타내는 pipelines, pipe-and-filter, 컴포넌트의 상호작용을 나타내는 데 용이한 event 방식이나, 여러 통신 프로세스, 데이터 저장고 역할을 하여 공유할 수 있도록 지원하는 blackboard 방식 등이 있으며, 현재 이러한 아키텍처 스타일 기반의 다양한 아키텍처 기술언어가 개발되고 있다 [5].

아키텍처 스타일 중 C2 아키텍처 스타일은 미국의 UC Berkeley에서 연구 개발중인 스타일로 메시지 기반의 계층 아키텍처를 나타낸다. 모든 컴포넌트와 커넥터는 top과 bottom을 가지고 있어 C2 컴포넌트들 사이의 통신은 커넥터를 통하여 이뤄지며 이때에는 top, bottom 포트를 통하여 일정한 규칙을 따라 상호 작용할 수 있다. C2 컴포넌트의 구조는 EJB 컴포넌트를 독립적으로 만들 수 있는 wrapper를 생성하여 컴포넌트들간의 조립을 지원하는데 응용될 수 있다.

3. 컴포넌트 조립을 지원하는 아키텍처 기술언어 및 지원 환경

본 장은 조립을 통한 EJB 컴포넌트간의 상호작용을 유연하게 지원하는 모델과 이를 기술하는 언어, 이의 설계 및 구현을 지원하는 도구의 환경에 대해 기술한다.

3.1 C2 Style을 기반으로 하는 아키텍처 기술언어

EJB 컴포넌트의 조립을 지원하는 아키텍처는 C2 아키텍처 스타일을 따르며, 각 컴포넌트마다 입출력 메시지와 관련된 wrapper를 생성하여 컴포넌트 간의 메시지 요청과 응답이 원활히 이뤄질 수 있도록 한다. 본 연구에서 컴포넌트 상호작용을 기술하도록 정의하는 아키텍처 기술언어는 각 컴포넌트의 독립적인 기술뿐만 아니라, 컴포넌트들과 커넥터들의 조립 아키텍처로도 기술할 수 있다. 즉, 각 컴포넌트마다 독립적으로 가지는 메소드 및 행위(behavior)가 명료하게 기술될 수 있으며, 세부적으로 기술된 컴포넌트들은 아키텍처 레벨에서 타 컴포넌트들과의 상호관계를 추상적으로 표현될 수 있도록 지원한다. 또한, 정의하는 아키텍처 기술 언어는 컴포넌트와 커넥터의 타입검사가 가능하며 기술하는 아키텍처의 타당성 검증도 지원한다. [그림 1]은 EJB 컴포넌트의 조립을 지원하기 위해 정의한 컴포넌트 및 아키텍처의 Semantic 모델이며 이는 아키텍처 모델을 기술하는 ADN(Architecture Description Notation)

과 컴포넌트 모델을 기술하는 IDN(Interface Definition Notation) 정의언어로 나눌 수 있다.

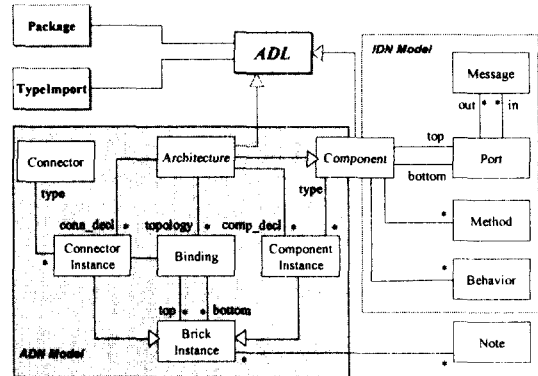


그림 1 EJB 컴포넌트의 조립을 지원하는 아키텍처의 Semantic 모델

3.2 컴포넌트 조립 아키텍처 설계를 위한 지원환경

3.1에서 정의한 아키텍처 기술언어를 기반으로 컴포넌트를 조립할 수 있도록 지원하는 CASE 도구의 요구사항은 다음과 같다.

- 1) 편집 기능: C2 스타일을 기반으로 컴포넌트 조립을 지원하는 아키텍처는 다이어그램과 텍스트를 이용하여 기술할 수 있다. 다이어그램 편집기에서 EJB 컴포넌트를 끌어다 놓아 연결한 후, 특정 편집기를 사용하여 이들간의 조립 명세를 작성하여 조립 아키텍처를 생성하거나, 텍스트 기반의 편집기에서 C2 스타일을 따르는 아키텍처 기술언어를 이용하여 조립 아키텍처를 기술할 수 있다. 동일한 어플리케이션 구축을 위해 작성된 컴포넌트 조립 아키텍처의 다이어그램 기술내용과 텍스트 기술내용은 상호 일치해야 한다.
- 2) 모델 검사 기능: 다이어그램이나 텍스트로 작성된 아키텍처 모델은 구문검사와 의미검사를 통하여 문법 및 의미검사를 수행할 수 있다. 구문검사는 아키텍처 모델이 문법에 맞게 작성되었는지를 말하며 의미검사는 타입검사와 형세(topology) 검사가 포함된다. 타입 검사란 아키텍처 모델이 아키텍처 언어의 타입규칙에 맞게 작성되었는지를 말하며 형세 검사란 아키텍처 스타일에서 규정하는 규칙에 따라 아키텍처 구성 요소가 연결되고 구조화되었는지를 말한다.
- 3) 아키텍처 모델의 컴포넌트 모델로의 변환 기능: 아키텍처를 구성함에 있어, 새로운 컴포넌트가 필요할 경우에는, 이를 생성할 수 있도록 아키텍처 모델을 컴포넌트 모델로 변환하는 기능이 필요하다. C2 기반의 아키텍처 분석 기법과 UML

기반의 컴포넌트 설계 및 구현 방법을 접목하여 아키텍처 모델의 모든 아키텍처 구성요소가 컴포넌트 모델의 EJB 컴포넌트로 변환될 수 있다.

위의 요구사항을 만족하기 위해 포함되어야 하는 아키텍처 기반 조립지원도구의 기본 구성은 크게 편집기, 모델검사기, 컴포넌트 모델 변환기로 나눌 수 있다. 편집기는 아키텍처 다이어그램 편집기와 ADL 편집기의 두 종류가 있으며 두 편집기간의 동기화를 맞춰주기 위한 작업이 필요하다. 모델검사기는 편집기에서 작성된 아키텍처의 구분 및 타입검사를 담당하며 컴포넌트 모델 변환기는 아키텍처 모델을 UML 기반의 컴포넌트 모델로 변환할 때 이용된다.

4. EJB 컴포넌트 조립 지원 아키텍처 모델러의 구현

3장에서는 EJB 컴포넌트의 조립을 지원하기 위하여 C2 아키텍처 스타일을 응용하여 아키텍처 기술언어를 정의하였고, 이를 지원하는 도구의 기본 기능을 살펴보았다. 이를 바탕으로 본 장에서는 실제 EJB 컴포넌트들의 조립을 지원하는 아키텍처 모델러의 구성 및 기능과 이의 프로토타입 구현에 대해 기술한다.

[그림 2]는 아키텍처 모델러의 전체 구성도 및 각 서브 모듈간의 관계를 나타낸 것이다.

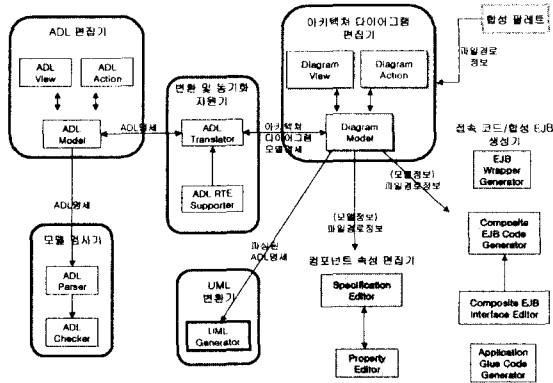


그림 2 아키텍처 모델러의 전체 구성도 및 서브 모듈간의 관계도

각 편집기는 M-V-C의 구조를 따르며 이들간의 모델일치는 변환 및 동기화 지원기에 의해 가능하다. EJB 컴포넌트의 조립을 지원하기 위하여 합성 팔레트로부터 기존의 EJB Jar 파일을 끌어다 놓거나 새로운 아키텍처 요소들을 팔레트로부터 생성할 수 있으며 컴포넌트 조립을 위하여 속성편집기와 합성 EJB 생성기의 기능을 이용하여, 3장에서 정의한 IDN을 작성하여 컴포넌트 명세를 기술하고 이로부터 생성된 IDN파일을 이용하여 전체 아키텍처를 구성하는 EJB 조립 컴포넌트를 생성한다. 작성된 컴포넌트 명세 및 아키텍처 명세는 모델 검사기를 통하여 검사할 수 있으며, 조립을 위하여 새로 생성하여야 하는 컴포넌트는 UML 변환기의 기능을 이용하여 UML 기반의 컴포넌트 모델로 변환할 수 있다.

[그림 3]은 EJB 컴포넌트 조립을 지원하는 아키텍처 모델러의 수행 화면이다.

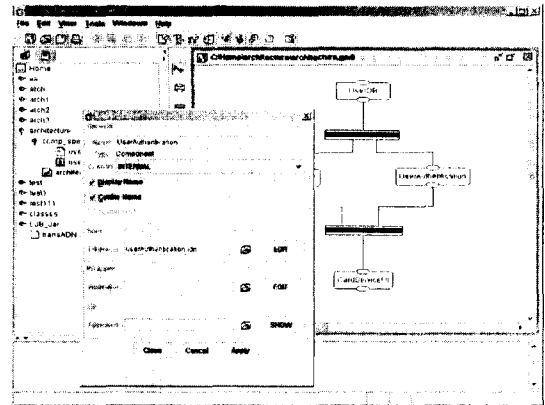


그림 3 아키텍처 모델러

5. 결론

본 논문은 EJB 기반의 컴포넌트 개발과 조립에 의한 응용 소프트웨어의 개발을 효과적으로 지원하기 위해 아키텍처를 기반으로 컴포넌트들이 서로 정확하게 결합하여 작동할 수 있도록 지원하는 컴포넌트 조립 환경을 제안하고 이를 구현한 아키텍처 모델러의 프로토타입을 기술하였다. 향후, 본 연구에서 제안한 아키텍처 기술 언어의 확장이 필요하며 같은 모델의 컴포넌트 조립 뿐만 아니라 이종 모델간의 조립 및 이종 서버에 전개되어 있는 서버레벨의 조립 또한 지원될 수 있는 도구의 확장이 필요하다.

6. 참고문헌

- [1] Plakosh, D., Smith, D., and Wallnau, K., Water Beans Component Builder's Guide, Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1999.
- [2] Nenad Medvidovic, Davis S. Rosenblum, and Richard N. Taylor, A Language and Environment for Architecture-Base Software Development, In the Proceedings of the 21st International Conference on Software Engineering (ICSE 21), Los Angeles, CA, pages 44-53, May, 1999.
- [3] Felix B., Len B., Charles B., Santiago C.D., Fred L., John R., Robert S. and Kurt W., Technical Concepts of Component-Based Software Engineering, Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 2000.
- [4] M. Shaw, "Patterns for Software Architectures", Proceedings of PLoP '94 Conference, PLoPD, Addison Wesley, pages 453-462, 1995.
- [5] Mary Shaw, David Garlan, "Software Architecture - Perspectives on and emerging discipline", Prentice-Hall, 1995.