

채널링을 이용한 SOFA/DCUP의

EJB컴포넌트 동적 업데이트 연구

김천호⁰ 송영재⁰⁰ 정화영⁰⁰⁰

경희대학교 컴퓨터공학과

kch1115@cvs2.khu.ac.kr⁰, jmichael@hanmir.com⁰⁰, yjsong@khu.ac.kr⁰⁰⁰

A study on the Dynamic Updating with EJB Component of

SOFA/DCUP using Channeling

Chun Ho Kim⁰, Young Jae Song⁰⁰, Hwa-Young Jeong⁰⁰⁰

School of Electronics and Information, Kyung Hee University

요 약

현재의 소프트웨어 개발 환경과 네트워킹 기술의 발전은 엔터프라이즈 소프트웨어 개발 등에서 소프트웨어 위기론과 함께, 객체 지향식 개발 방법의 한계를 드러냈으며, 이에 새로운 소프트웨어 개발 론인 CBD(Component Based Development 컴포넌트 기반개발)기술이 전 세계적으로 빠르게 확산되고 있다. 특히 네트워킹 기반의 EJB에서의 컴포넌트형 소프트웨어 개발이 활발히 이루어지고 있다. EJB로 만들어진 소프트웨어는 뛰어난 확장성과 트랜잭션을 보장하며, 멀티유저 환경에서도 그 보안성을 인정받고 있다. 최근에는 실시간으로 이루어지는 인터넷 및 인트라넷에서 EJB의 런타임 유지의 중요성이 부각 되고 있지만 런타임을 유지하는 동안 컴포넌트의 동적 업데이트, 수정, 및 삭제에 대한 지원이 미미한 상태이다.

SOFA/DCUP는 SOFA(Software Appliances)형의 아키텍처를 활용한 DCUP(Dynamic Component UPdating), 즉 동적업데이트가 이루어지는 구조이다. 동적업데이트를 할 수는 강력한 장점이 있지만, 이런 구조는 항상 SOFA형의 컴포넌트만 가능하기 때문에, 이종의 컴포넌트를 조립하는데 있어서 우리가 있으며, EJB구조에 직접적인 적용에 어려움이 존재한다. 이에 대해 본 논문은 채널을 이용하여, SOFA/DCUP기반의 컴포넌트 조립 방법을 EJB에서 런타임시에도 컴포넌트의 동적 업데이트, 수정 및 삭제가 가능하도록 연구하였으며, 이종의 컴포넌트 역시 좀더 용이하게 조립할 수 있게 되었다.

1. 서 론

오늘날 연구원들과 엔지니어들의 프로그램과 디자인기술은 컴포넌트 기반에서 많이 이루어진다.[3] 이 CBD 기술들은 컴포넌트가 라이브러리가 축적 되면서 제품의 개발기간을 훨씬 단축 시키고, 높은 응집도와 낮은 결함도를 유지시켜 시스템의 품질도 개선 시킨다. 또한 이러한 발전에 맞추어 네트워킹기반에서도 컴포넌트를 활용한 기술들이 나오고 있다. 특히 EJB의 경우 분산된 객체와 컴포넌트 구조로 이루어져 있으며,[2] 확장성이 용이하고, 트랜잭션이 안정적이다. EJB에는 여러 가지 다른 종류 컴포넌트를 그룹화 시키고 이종의 컴포넌트를 조립 가능하게 관리하는 컨테이너들이 존재하지만, J2EE에서 런타임시 여러 컨테이너들로 인해 EJB서버가 J2EE 환경안에서 종종 병목현상을 일으키기도 합니다.[1] 아직은 컨테이너가 불안한 면이 있으며 컨테이너 관리 빈은 컨테이너에 종속적

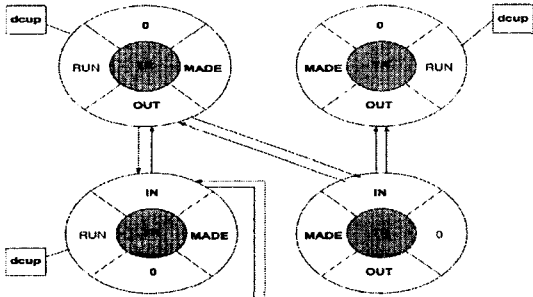
이기 때문에 빈들들에 따라 컨테이너 관리 빈을 달리 정해야 한다. SOFA/DCUP는 컴포넌트를 동적으로 업데이트 시키는 기반으로 SOFA형의 컴포넌트 자체 채널을 가지고 있어서 런타임시에도 동적인 업데이트 가능하지만, EJB에 직접적인 적용은 어렵다. 이에 자체 채널을 외에 외부 채널을 생성시켜 사용하면 EJB에서도 동적으로 컴포넌트를 업데이트 가능하며 이종의 컴포넌트를 조립이 가능하다. EJB 컨테이너 대신에 적용이 가능하기 때문에 크고 무거운 컨테이너를 생성하지 않아도 된다. 또한 채널과 채널끼리 빠른 메시지 전달을 할 수 있기 때문에 계층간의 컴포넌트를 관리하는 데에도 많은 유리한 점을 찾을 수 있다.[4]

2. 관련 연구

2.1 SOFA/DCUP

SOFA기반의 컴포넌트는 동적 업데이트 동안 컴포넌트의 안전한 상태 변화를 위한 아키텍처 제공하며, 이러한 구조는 일정한 영역 안에서의 업데이

트 하도록 하였기 때문이며, 사용자가 수정이 없이 자체 채널을 통해 동적 업데이트가 가능환경을 제공한다. [5]



<그림 1> SOFA/DCUP 구조

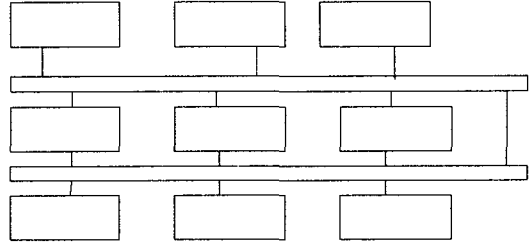
<그림 1>은 SOFA형의 컴포넌트 구조로 내부에 연결할 컴포넌트 템플릿 저장소(TR:Template Repository)가 존재하고 보안적인 요소 때문에 외부에서 직접적인 템플릿을 조작할 수 없으며 인터페이스 통해 메시지가 IN채널을 통해 접근하게 되면 RUN상태로 전환이 되고 RUN상태에서만 템플릿 저장소에 접근이 가능하며 이때 템플릿을 수정이 가능하다. 또한 다른 요청 및 결과를 OUT채널을 통해 보내게 되며, 이때 다른 컴포넌트들이 수행이 시작되고 자신이 DCUP 가능 상태로 이루어지게 된다. 마찬가지로 자체 컴포넌트가 수행 중일 때는 DCUP불능 상태로 전환되게 된다. 이러한 구조는 정확한 스케줄을 필요로 하기 때문에 바운더리 처리를 통해 정확한 업그레이드가 가능하게 한다. 바운더리는 컴포넌트 교체 여부에 따라 영구적인 영역, 교체 가능한 영역으로 나뉘며, 기능적인 면에서는 기능 영역, 제어 영역으로 나누어 바운더리를 형성하여 안정적인 업데이트가 가능하도록 되어있다.

2.2 C2

C2는 컴포넌트 조립 기법 중에 하나로 커넥터를 중점으로 컴포넌트 간에 조립이 쉬운 구조이다. 계층구조의 커넥터를 직접 연결이 가능하여 메시지 빠르게 전달 할 수 있는 구조를 가지고 있으며 이종의 컴포넌트를 조립할 때에 다른 컴포넌트와 직접적인 연결이 아니라 커넥터를 통해 연결하게 된다. 이는 커넥터에 대한 부분만 수정하면 되기 때문에 컴포넌트 조립이 용이한 구조를 되어있다. 커넥터는 메시지의 상, 하 전달 및 필터링 기능을 가지고 있으며 <그림2>에서 보듯이 계층적인 구조

를 가지고 있기 때문에 GUI구조에 적합구조를 가지고 있다.[6][7]

하지만 각 커넥터에 연결된 컴포넌트의 증가는 커넥션에 풀 대한 부하를 증가 시키고 Component의 직접적인 연결이 불가능하여 처리 속도의 저하를 가져 올 수 있으며, GUI기반이기 때문에 EJB에서 직접적인 적용은 어려운 면을 가지고 있다.[8]

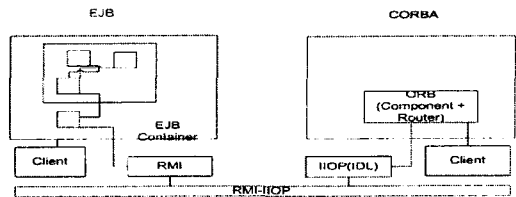


<그림 2>C2 구조간

2.3 EJB Container & CORBA

네트워크에서 컴포넌트기반의 시스템은 대표적으로 EJB와, CORBA를 들 수 있다. EJB는 자체 컴포넌트로는 이종의 컴포넌트와 연결 할 수 없지만 EJB Container를 사용하여 연결 할 수 있다. 컨테이너는 각 유형의 빈을 위한 EJB 객체와 홈 객체를 관리할 수 있다. 하지만 컨테이너 관리 빈이 컨테이너 종속적이라는 특징 때문에 다른 벤더의 컨테이너를 끼워넣기가 힘들다. 또한 이종의 컴포넌트와 연결할 때 마다 컨테이너가 필요하기 때문에 컨테이너 관리를 잘 해주어야 한다.

CORBA의 경우, ORB구조로 이루어져 있으며, 컴포넌트를 사용할 때 IDL만 사용하기 때문에 CORBA에서는 다른 언어로 작성된 컴포넌트 조립 가능하다. 그렇지만 이 기종간의 분산 환경에서 컴포넌트들을 상호 연동시키기 위한 가장 기본적인 환경만 지원 할 뿐 컴포넌트 생성, 저장 등의 기본적인 능력과 컴포넌트의 검색과 결함 또는 컴포넌트 교체 등에 관한 일련의 규칙과 지원시스템 미약하다.

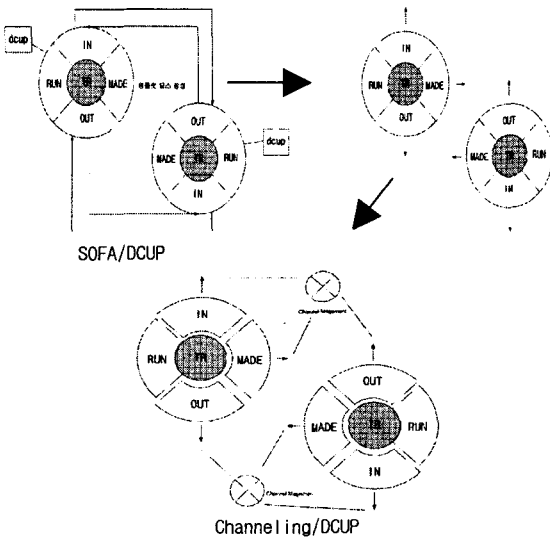


<그림 3> EJB & CORBA 구조 및 연동

3. Channeling/DCUP

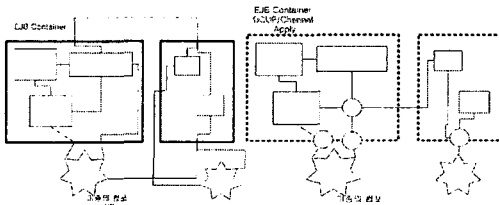
기존의 SOFA/DCUP구조는 동적업데이트라는 구조를 가지고 있지만 EJB에서 적용 및 이종의 컴포넌트의 업데이트 적용하기에 무리가 있었다. 또한 C2 형의 컴포넌트 조립 기법은 커넥터가 채널 역할을 가지고 있었기에 메시지의 전달이 용이하고 이종의 컴포넌트를 조립하기에 유리한 면을 가지고 있었지만, 역시 EJB에서 적용이 어려웠다. 또한 EJB를 지원하는 EJB 컨테이너들은 이 종의 컴포넌트는 용이하였으나, 이종의 컴포넌트가 많아질 경우에 컨테이너 역시 그에 맞게 인터페이스를 추가하거나, 아님 또 다른 컨테이너를 요구 하였다.

이에 대해 본 논문은 Channeling/DCUP 기법을 소개하는 바이다.



<그림 4> SOFA/DCUP & Channeling/DCUP 비교

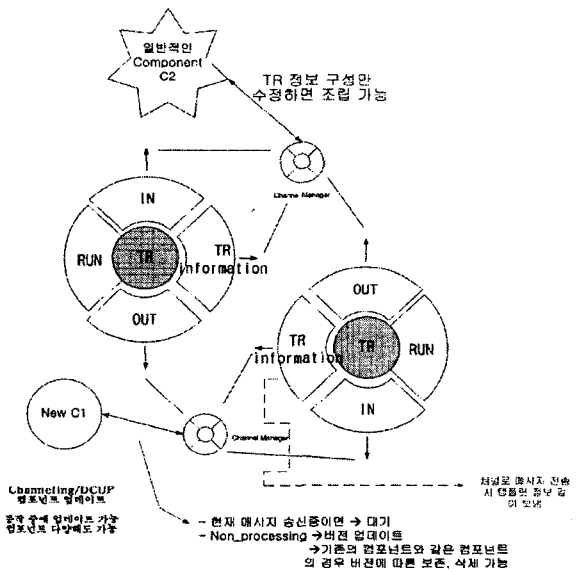
기존의 SOFA/DCUP에서 자체 채널에서 IN, OUT, MADE 대한 관리 부분을 따로 두는 방식이다. 채널을 외부에도 동으로 써 이종의 컴포넌트 연결이 가능하며, EJB에서 동적업데이트 또한 용이하게 할 수 있는 구조를 가지게 된다.



<그림 5> EJB Container & DCUP/Channeling 비교

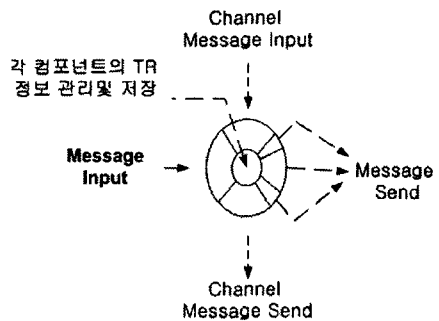
위 그림을 보면 EJB에서 이종의 컴포넌트를 조립하려면 컨테이너를 통해 해야 하며 조립하려는 컴포넌트가 많아지면 컨테이너 역시 많아 져야한다. 또한 컨테이너가 많아지면 컴포넌트들이 컨테이너를 통해 다른 컴포넌트와 메시지 전달하기 때문에 효율이 떨어진다. 그에 비해 채널을 사용하게 되면 채널을 통해 이종컴포넌트와 쉽게 접근 할 수 있으며, 컨테이너를 통하지 않고 같은 종류의 컴포넌트를 조립이 가능해진다.

Channeling/DCUP를 살펴보면 <그림6>과 나타낼 수 있다. 기존 SOFA형 컴포넌트에 채널 매니저를 두어 메시지를 제어 한다. 또한 채널은 컴포넌트를 동적 업데이트



<그림 6> Channeling/DCUP

트를 하게 될 때 컴포넌트에서 처리하는 것이 아니라 채널에서 동적 업데이트를 시킨다.



<그림 7> Channel manager

Channel manager는 컴포넌트에서 온 메시지를 필

터링을 통해 다른 컴포넌트들에 전달해주고 각 컴포넌트들의 템플릿 정보를 관리 및 저장한다. 또한 Channel manager는 컴포넌트 뿐 만아니라, Channel 끼리도 통신을 가능하게 되어 있는 구조로 되어 있고 Channel에 대한 전용 인터페이스가 존재하기 때문에 보다 빠른 메시지를 전달 할 수 있다. 만약 새로운 컴포넌트 (New C1)를 추가 할 경우 채널은 현재 메시지 송신중이면 DCUP를 대기 시켜놓고 동일 컴포넌트가 존재하는지 템플릿 저장소에서 찾는다, 여기서 동일 컴포넌트가 존재할 경우 기존 컴포넌트는 메시지 차단을 시키어 간접적인 삭제를 하는 동시에 동적인 업데이트를 이루어 지도록 한다. 새로운 컴포넌트가 이종의 컴포넌트 일 경우에는 컴포넌트에 대해 채널에 인터페이스와 템플릿 정보만 수정하거나 추가 해주면 보다 쉬운 조립할 수 있게 된다.

SOFA/DCUP에서는 이론적으로 템플릿 정보는 필수적으로 (provider_name, type_name, version)만 있으면 가능하다고 하였으나, 실제로 EJB에 적용 하는데 있어 기본적으로 (Component id, Component 이름, File위치, SQLFile위치, HelpFile위치, APIFile위치, Component version)을 필요하며 부수적으로 (OS타입, Component가격, Component크기, Upload날짜, Component종류, JNDI이름, ClientFile위치, Component 구분) 정보를 추가 할 수 있다.

4. 결론 및 향후 연구 방향

본 논문에서는 기존의 SOFA/DCUP를 활용한 동적인 컴포넌트 업데이트기법을 채널을 추가함으로써 EJB가 런타임시에서도 동적인 업데이트를 가능하게 하고, EJB컨테이너를 사용하지 않아도 이종의 컴포넌트 조립 쉬게 할 수 있었다.

향후 연구 과제로는 채널매니저의 템플릿 구성요소 증가에 따른 템플릿 정보 관리와 보다 정확한 채널링 방법을 연구해야 할 것이다.

5. 참고논문

[1] Emmanuel Cecchet, Julie Marguerite, Willy Zwaenepoel, Performance and Scalability of EJB Applications, OOPSLA, 02, November 4-8, 2002, Seattle, Washington. ACM 2002.
 [2] James White, "Enterprise JavaBean architecture and design issues", IEEE Computer Society Washington, DC, USA, 731 - 732 Series-Proceeding-Article, 2001
 [3] Wolfgang Emmerich, Nima Kaveh, "Component technologies: Java beans, COM, CORBA, RMI, EJB

and the CORBA component model" ACM Press New York, NY, USA Pages: 691 - 692, 2002

[4] Frantisek Plasil, "Behavior Protocols for Software Components" IEEE Computer Society, and Stanislav Visnovsky 1056-1076 IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 28, NO. 11, NOVEMBER 2002

[5] N. D. Hoa. Dynamic aspects in SOFA/DCUP. Technical Report Jun. 07 .2002

[6] Richard N. Taylor, "A Component-and Message-Based Architechural Style for GUI Software"

IEEE TRANSACTIONS OF SOFTWARE ENGINEERING, 390-406 VOL22, NO.6, JUNE 1996

[7] Jiaccum Wang and Yi Deng Component-Level Reduction Rules for Time PetriNets with Application in C2 System, IEEE 125-130 June. 1998.

[8] You-Hee Choi, Oh-Cheon Kwon, Gyu-Sang Shin "An Approach to Composition of EJB Components Using C2 style" Proceedings of the 28 th Euromicro Conference (EUROMICRO' 02) 1089-6503/02 IEEE 2002