

컴포넌트 결합 컨트랙트를 지원하는 컴포넌트 아키텍처

박성은, 이정태, 류기열
아주대학교 정보 및 컴퓨터 공학부
e-mail : aceoface@madang.ajou.ac.kr

A Component Architecture for supporting Component Composition Contracts

Sung-Eun Park, Jung-Tae Lee, Ki-Yeol Ryu
College of Computer and Information, Ajou University

요 약

현재 대부분의 컴포넌트 프레임워크들이 특정 형태 컴포넌트와의 결합 기능 및 이에 요구되는 각종 서비스 기능의 제공을 위주로 하는데 대하여, 좀 더 효과적인 컴포넌트 기반 개발 환경의 구축을 위해서는 재귀적 컴포넌트 결합 및 다양한 결합형태의 지원, 그리고 다중 티어드 아키텍처를 지원할 수 있는 방안의 필요성이 제기되고 있다. 따라서 본 논문에서는 재귀적 컴포넌트 결합을 지원할 수 있는 컴포넌트 결합 컨트랙트에 기반 하여 컴포넌트 결합 컨트랙트를 연결자로 하여 구성된 복합 컴포넌트 자체가 컴포넌트 또는 프레임워크로써 기능 하게 됨으로써 다중 티어 개념의 컴포넌트 시스템 아키텍처를 구성할 수 있음을 보이고자 한다.

1. 서론

현재 대부분의 컴포넌트 프레임워크들은 특정 형태의 컴포넌트를 결합시키는 방식 및 결합된 컴포넌트들 간의 통신 기능은 잘 제공하고 있으나 컴포넌트와 컴포넌트간의 재귀적 결합 및 이렇게 재귀적으로 결합된 복합 컴포넌트를 다시 컴포넌트로 보고 재사용하는 방안에 대해서는 그 필요성은 제기되고 있으나 구체적인 방안에 대한 연구는 아직 미흡하다.

그러나 최근 본격적인 컴포넌트 기반 시스템의 구축을 위해서는 컴포넌트 시스템의 아키텍처가 컴포넌트 프레임워크 자체를 일관된 결합방식에 의하여 결합된 컴포넌트로 간주하는 다중 티어 개념의 아키텍처를 지원해야 하며, 이를 위해서는 컴포넌트 배치(component deployment), 프레임워크 배치(framework deployment), 단순 결합(simple composition), 이질간의 결합(heterogeneous composition), 프레임워크 확장(framework extension), 그리고 컴포넌트 어셈블리(component assembly)등의 다양한 컴포넌트 결합을 허용해야 한다는 제안이 제기되고 있다 [6].

본 논문에서는 연결자(connector)로써의 컴포넌트 결

합 컨트랙트를 사용함으로써 다중 티어드 컴포넌트 아키텍처(N-tiered component architecture)[3,4]를 만족하며 재귀적 컴포넌트 결합(recursive component composition) [5]과 현재까지 알려진 다양한 컴포넌트간의 결합 형태들[6]을 수용할 수 있는 컴포넌트 시스템 아키텍처를 제안하고자 한다.

2. 컴포넌트 결합 컨트랙트

기존의 컴포넌트 컨트랙트는 컴포넌트가 프레임워크에 배치될 때 컴포넌트가 제공할 수 있는 서비스와 컴포넌트가 프레임워크에 배치되기 위해서 필요한 내용들이 기술되는데 반해서 컴포넌트 결합 컨트랙트는 다음의 세 부분으로 구성된다.

- 결합에 참여하는 컴포넌트 목록: 결합에 참여하는 컴포넌트는 컴포넌트 자체의 모습을 기술하는 자체 컨트랙트를 가질 수 있으며 결합 컨트랙트는 목록에 기술되어 있는 모든 컴포넌트들의 컨트랙트를 계승 받게 된다.
- 결합에 참여하는 컴포넌트의 제공 또는 요구 서비스 목록: 결합에 참여하는 컴포넌트의 컨트

랙트에 기술된 기능들을 요구 기능과 제공 기능으로 분류 기술하며, 이들 기능들 중 어떤 기능들이 다시 새롭게 만들어지는 복합 컴포넌트의 인터페이스가 되는지를 기술한다.

- 결합에 참여하는 컴포넌트간의 상호동작 패턴 제약 조건: 컴포넌트는 서로 독립적으로 개발되고 배치되기 때문에 개발단계에서 고려되지 않은 컴포넌트와도 결합되어서 상호동작을 할 수 있다. 이러한 경우 결합되는 컴포넌트간에 반드시 지켜져야 하는 상호동작 패턴을 기술한다. 컴포넌트 결합 컨트랙트는 둘 이상의 컴포넌트가 결합될 때 각 컴포넌트간에 발생하는 상호동작(interaction) 패턴에 대한 제약 조건을 기술할 수 있으며 컴포넌트간의 연결자(connector) 및 결합에 의해서 생성된 복합 컴포넌트에 대한 인터페이스의 역할도 수행한다.

본 논문에서는 컴포넌트와 컴포넌트간에 결합을 위한 컨트랙트를 기술하고 이 컨트랙트를 연결자로 하는 복합 컴포넌트를 만들 수 있으므로 해서 기존의 컴포넌트 아키텍처에 어떤 변화가 있는지를 보이는 것이 주 목적이므로 결합 컨트랙트의 구체적인 기술 방향에 대해서는 [7]을 참조하기 바란다.

3. 재귀적 컴포넌트 결합

결합 컨트랙트는 기존의 컨트랙트와는 달리 그림 1과 같이 컴포넌트들이 결합될 수 있도록 연결자(connector)의 역할을 수행한다. 연결자로써의 결합 컨트랙트는 컴포넌트 C1과 C2를 연결시키면서 두 컴포넌트간에 일어나는 상호동작 패턴을 관리한다. 결합 컨트랙트에서 기술되어지는 상호동작 패턴의 제약 조건을 이용하면 기존의 특정 컴포넌트에 대한 컨트랙트에서는 기술할 수 없었던, 독립적으로 개발되어진 컴포넌트들간의 상호동작 패턴(interaction pattern)을 관리할 수 있다.

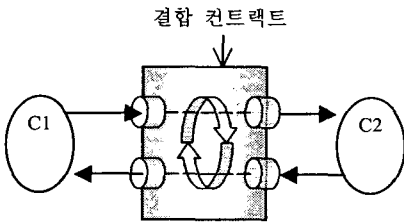


그림 1. 연결자로써의 결합 컨트랙트

상호동작 패턴의 관리는 하나의 컴포넌트가 작성되는 시점에서 고려하지 않은 다른 컴포넌트와 결합되어서 상호동작을 할 때, 각 컴포넌트들이 제공하는 서비스들이 다른 컴포넌트들에 의해서 올바르게 사용될 수 있도록 할 수 있다.

그림 2와 같이 재귀적 컴포넌트 결합에 의해서 만들어진 복합 컴포넌트가 다른 결합 컨트랙트에게 컴포넌트로써 보이기 위해서는, 먼저 결합 컨트랙트가

연결자로써의 역할을 수행해서 컴포넌트들을 결합시킨 다음에 결합 컨트랙트는 컴포넌트 결합 후에 생성되는 복합 컴포넌트에 대한 컨트랙트로써의 역할도 수행해야 한다. 이와 같이 되면 결합 컨트랙트에 의해서 만들어진 복합 컴포넌트는 다시 다른 결합 컨트랙트에 의해서 다른 컴포넌트들과 결합될 수 있다.

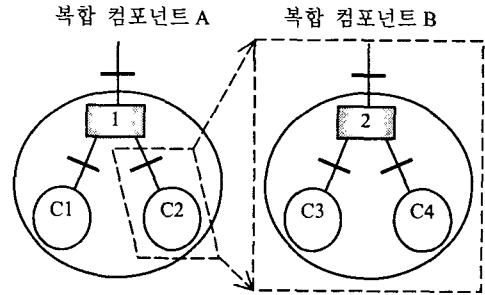


그림 2. 결합 컨트랙트에 의한 재귀적 컴포넌트 결합

그림 2에서 컴포넌트 C3과 C4는 결합 컨트랙트 2에 의해서 결합되어 복합 컴포넌트 B가 생성된다. 복합 컴포넌트 B는 결합 컨트랙트 2에 의해서 제공되어진 컨트랙트를 이용, 컴포넌트 C2로써 결합 컨트랙트 1에 의해서 컴포넌트 C1과 결합되어서 복합 컴포넌트 A를 만들게 된다. 마찬가지로 방법으로 복합 컴포넌트 A는 복합 컴포넌트 B가 결합 컨트랙트 1에게 일반 컴포넌트 C2로 보여질 수 있었던 것처럼 다른 결합 컨트랙트에게 일반 컴포넌트처럼 보여질 수 있다.

4. 다중 티어드 아키텍처

다중 티어드 아키텍처의 특징은 프레임워크 자체를 컴포넌트로 볼 수 있는 것이다. 다시 말하면 하위 티어(low tier)는 상위 티어(high tier)에게 컴포넌트로 보이지만 하위 티어 자체는 프레임워크일 수도 있는 것이다. 이와 같은 특징은 컴포넌트 결합 컨트랙트에 의해서 재귀적으로 만들어진 복합 컴포넌트를 프레임워크로 보면 만족시킬 수 있다.

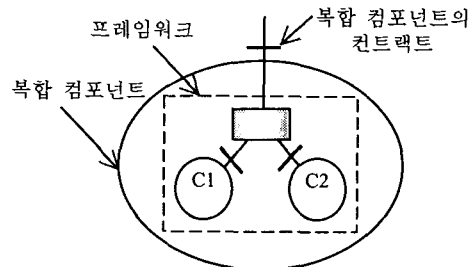


그림 3. 컴포넌트 결합 컨트랙트와 프레임워크

그림 3의 컴포넌트 결합 컨트랙트가 컴포넌트 C1과 C2를 결합시키면서 재귀적 컴포넌트 결합에 의한 복합 컴포넌트를 생성하고 복합 컴포넌트의 컨트랙트를 만들기 때문에 하나의 컴포넌트로써 상위 티어에

배치될 수 있다(상위 티어에게 컴포넌트로 보임). 또한 복합 컴포넌트의 내부적인 측면에서는 컴포넌트 C1, C2 와 이 둘을 연결시킨 컴포넌트 결합 컨트랙트가 하나의 프레임워크를 구성함으로써 다중 티어드 컴포넌트 아키텍처를 지원한다.

그림 4 와 같이 현재 대부분의 프레임워크들은 프레임워크가 만들어질 때부터 그 프레임워크가 제공할 수 있는, 도메인에 특화 된 서비스(vertical service)와 도메인 전반에 걸쳐 공통으로 사용되는 서비스(horizontal service)의 종류와 수준이 결정되며, 나중에 각 서비스의 기능 강화나 새로운 서비스의 추가가 쉽지 않다(정적 컴포넌트 프레임워크).

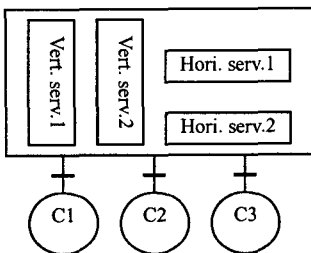


그림 4. 정적 컴포넌트 프레임워크

그러나, 기존 프레임워크에서 제공되는 도메인 특화 서비스와 도메인 공용 서비스들을 모두 컴포넌트로 만들 수 있다면 즉, 도메인 특화 서비스 컴포넌트와 도메인 공용 서비스 컴포넌트를 만들 수 있다면 기존 서비스 컴포넌트의 교체 및 새로운 서비스 컴포넌트와의 결합을 통해서 각 서비스의 기능 강화와 새로운 서비스의 추가를 쉽게 할 수 있는 동적 컴포넌트 프레임워크를 구성할 수 있을 것이다. 이와 같이 서비스 컴포넌트들을 이용해서 컴포넌트 프레임워크를 만들기 위해서는 재귀적 컴포넌트 결합과 다중 티어드 아키텍처의 개념이 반드시 지원되어야 한다.

결합 컨트랙트에 의해서 만들어진 복합 컴포넌트는 앞에서 설명된 바와 같이 상위 티어에게는 컴포넌트로 보여지고, 복합 컴포넌트의 내부는 프레임워크로 볼 수 있기 때문에 그림 5 와 같은 프레임워크를 구성할 수 있다.

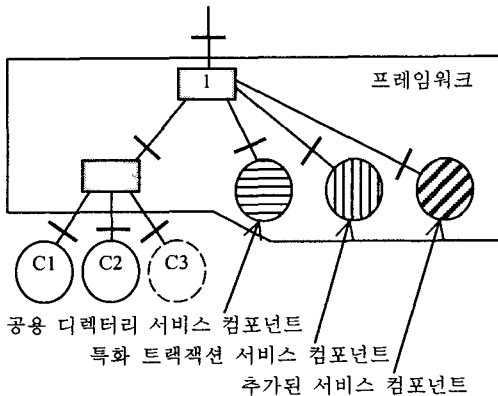


그림 5. 동적 컴포넌트 프레임워크

그림 5 의 프레임워크 초기상태가 도메인 공용 서비스인 디렉터리 서비스를 제공하는 컴포넌트와 도메인 특화 서비스인 트랜잭션 서비스를 제공하는 컴포넌트, 그리고 프레임워크에 배치되어서 실행되는 컴포넌트 C1 과 C2 로 구성되어 있었다고 가정한다. 나중에 사용자의 요구에 의해서 프레임워크에 새로 추가된 컴포넌트 C3 가 새로운 서비스를 요구한다면 컴포넌트 C3 의 요구사항을 구현한 서비스 컴포넌트(컴포넌트 내부는 요구되는 서비스가 구현된 프레임워크일 수도 있다)를 결합 컨트랙트 1 의 수정을 통해서 프레임워크에 결합시킨다.

5. 다양한 결합 형태

현재까지 알려진 다양한 결합형태[6]는 결합 컨트랙트에 의한 재귀적 컴포넌트 결합과 다중 티어드 아키텍처의 지원으로 프레임워크 배치 결합형태는 컴포넌트 배치 결합형태로 처리할 수 있으며 이질간의 결합, 프레임워크 확장, 그리고 컴포넌트 어셈블리는 단순 결합형태로 처리될 수 있다.

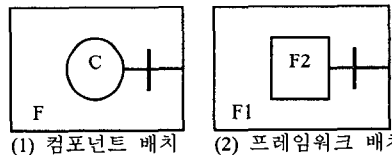


그림 6. 컴포넌트 배치와 프레임워크 배치 결합형태[6]

프레임워크 배치 결합형태는 그림 6 의 (2)와 같이 하나의 프레임워크가 컨트랙트를 제공함으로써 해서 다른 프레임워크에 결합될 수 있는 형태를 나타낸다. 이때 프레임워크가 제공하는 컨트랙트는 컴포넌트가 프레임워크에 배치되기 위해서 컴포넌트가 가지고 있는 컨트랙트와 같다. 이와 같이 되기 위해서는 프레임워크를 컴포넌트로 볼 수 있는 다중 티어드 아키텍처의 개념이 필요하다.

그림 3 에서 설명된 것과 같이 결합 컨트랙트를 이용하면 프레임워크를 컴포넌트로 볼 수 있기 때문에 그림 6 의 (2)는 다음의 그림 7 처럼 표현될 수 있다.

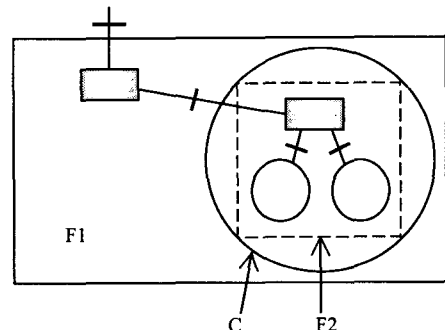


그림 7. 결합 컨트랙트를 이용한 프레임워크 배치

결합형태

그림 8의 이질간의 결합, 프레임워크 확장, 그리고 컴포넌트 어셈블리 결합형태도 재귀적 컴포넌트 결합과 다중 티어드 아키텍처의 개념이 지원된다면 단순 결합형태로 통합시킬 수 있다.

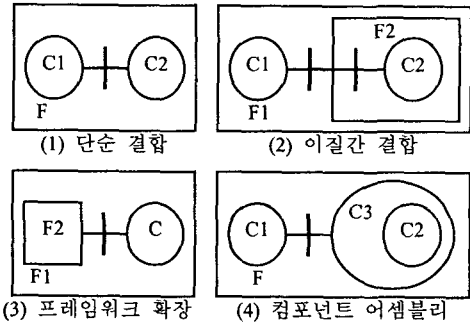


그림 8. 다양한 결합형태[6]

이질간의 결합형태에서 프레임워크 F2를 내부에 컴포넌트 C2를 포함한 복합 컴포넌트로 보면 컴포넌트 어셈블리와 같은 형태의 결합이 되며, 컴포넌트 어셈블리 결합형태는 컴포넌트 C3가 하나의 복합 컴포넌트로써 C1과 결합되는 형태이기 때문에 단순 결합형태와 같아진다. 또한 프레임워크 확장의 형태는 프레임워크 F2가 컴포넌트로서 컴포넌트 C와 결합될 수 있기 때문에 이 역시 단순 결합과 같은 형태의 결합 형태가 된다.

단순 결합과 컴포넌트 어셈블리 결합형태는 그림 2와 3을 통해서 지원됨을 알 수 있다. 프레임워크 확장 결합형태는 그림 5의 컴포넌트 C3과 같이 기존의 결합 컨트랙트를 수정해서 프레임워크와 컴포넌트를 결합시키는 방법이 있을 수 있으며, 그림 9와 같이 새로운 결합 컨트랙트를 이용해서 프레임워크 F2에 컴포넌트 C를 결합시키는 방법으로 지원할 수도 있다.

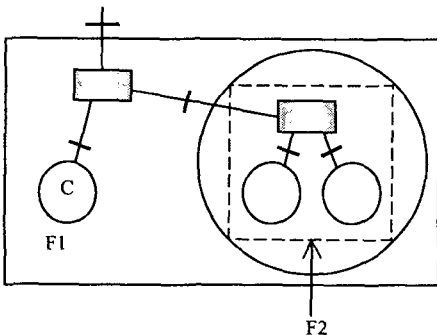


그림 9. 결합 컨트랙트를 이용한 프레임워크 확장 결합 형태

6. 결론 및 향후 계획

본 논문에서는 재귀적 컴포넌트 결합과 프레임워크 배치, 단순 결합, 프레임워크 확장, 그리고 컴포넌트

어셈블리 결합형태 등을 지원할 수 있는 컴포넌트 결합 컨트랙트를 기반으로 프레임워크를 컴포넌트로, 컴포넌트를 프레임워크로 볼 수 있는 다중 티어드 아키텍처를 만족하는 컴포넌트 아키텍처를 제안하였다.

컴포넌트 결합 컨트랙트를 지원하는 컴포넌트 아키텍처를 사용하면 기존의 컴포넌트 프레임워크에서 제공하는 모든 서비스를 서비스 컴포넌트로 만들어서 동적 컴포넌트 프레임워크를 구성할 수 있을 뿐만 아니라 경량 컴포넌트(lightweight component)[3,4]를 좀 더 잘 지원할 수 있는 경량 컴포넌트 프레임워크(lightweight component framework)[3,4]를 작성할 수도 있을 것이다. 그리고 결합 컨트랙트에 의해서 컴포넌트 명세(component specification)와 컴포넌트 구현(component implementation), 더 나아가 결합 내용까지도 분리시키기 때문에 컴포넌트 기반 소프트웨어 개발(CBSD)의 또 다른 목적인 컴포넌트 명세와 컴포넌트 구현의 철저한 분리를 통한 프로그램의 개발 환경을 좀 더 잘 지원할 수 있을 것이다.

향후 계획으로는 결합 컨트랙트의 내용을 기술하는 스크립트 언어와 표현 방법, 재귀적 컴포넌트 결합을 통한 새로운 기능의 생성 및 그 기능의 기술 방법에 대한 연구, 그리고 결합 컨트랙트와 프레임워크의 실제 구현을 통해서 제안된 컴포넌트 아키텍처를 검증하는 과정이 진행될 예정이다.

참고문헌

- [1] Johannes Leon Marais, "Design and Implementation of a Component-Architecture for Oberon", thesis, 1996
- [2] Sametinger J., "Software Engineering with Reusable Components", Springer-Verlag, Town., 1997
- [3] Szyperski C., "Component Software: Beyond Object-Oriented Programming", Addison-Wesley, 1998
- [4] Szyperski C., "Component Software and the Way Ahead", In Gary T. Leavens, Murali Sitaraman, "Foundations of Component-Based Systems", Cambridge University Press, 2000
- [5] Francisco Curbera, Sanfiva Weerawarana, Matthew J. Duffler, "On Component Composition Language", IBM T. J. Watson Research Center, 2000
- [6] Felix Bachman, Len Bass, Charles Buhman, Santiago Comella-Dorda, Fred Long, John Robert, Robert Seacord, Kurt Wallnau, "Volumn II: Technical Concepts of Component-Based Software Engineering", CMU/SEI-2000-TR-008, 2000
- [7] 백경원, 이정태, 류기열, "상호작용 제약 조건을 기술할 수 있는 컴포넌트 결합 컨트랙트", 제 16 회 한국정보처리학회 추계학술발표대회, 2001 [예정]