韓國 OA學會 論文誌
第6卷 第3號, 2001. 9.

2001-6-3-1-4

참조 컴포넌트 아키텍처 모델과 UML 명세화에 대한 연구

장 연 세*

A Study on the referential Component Architecture and UML Specification

Yeun-Sae Jang*

요 약

급변하는 IT 환경에서 시스템의 재사용성을 높여 라이프타임을 증가시키고 비용을 절감하기 위한 다양한 노력들이 이루어져 왔다. 구조적 프로그래밍 기법에서는 모듈에 기반한 아키텍처를 활용하여 생산성을 향상시켰다. 그러나 모듈들은 단순히 호출 빈도를 높일 뿐, 성장이나 진화를 하지 못하는 한계 상황에 직면하게 되었다. 객체지향 기법은 클래스들을 상속시키거나 메소드를 재정의 함으로써 시스템의 성장과 진화를 가능케 하여 구조적 프로그래밍 기법의 한계를 극복하였다. 최근 CORBA나 DCOM과 같은 분산 처리 기술과 객체지향 기법이 융화되어 생성된 컴포넌트 아키텍처는 고도의 재사용성이나 라이프타임의 증가뿐만 아니라 플러그-앤-플레이(Plug-&-Play)도 지원한다. 그러나 이러한 컴포넌트들을 조립하여 새로운 시스템을 구축하기 위해서는 컴포넌트의 구조와 인터페이스를 잘 정의한 명세서가 필요하다. 본 연구에서는 컴포넌트의 도입을 위한 참조 컴포넌트 아키텍처를 제시하고 UML을 이용한 이의 명세화에 대하여 제안한다.

Abstract

There has been several meaning full efforts to save costs on system development and expand the life-time of a system in changeful IT circumstance. It was a module-based architecture that empower productivity at structured programming era. But it couldn't grow nor evolve, but could raise only calling frequency of module. But OOP or OO-method overcome limit of structured programing by class inheritance and/or overloading and/or over-riding. A component centric architecture, what is mixture of distributed systems, like CORBA or DCOM with OOP, can support not only high reusability or expansion of life-time but also Plug-&-Play between them. To assemble these component to build a new system in easy way, the well-formed specification of a component is highly required. At this study, the enhanced referential component architecture and its UML specification will be suggested.

* 수원과학대학 인터넷정보과 겸임교수

Ⅰ. 서론

현재 우리가 직면하고 있는 전사적 컴퓨팅 환경(Enterprise Computing)은 복잡하고, 다양하며, 빠르게 변화하고 있다. 이러한 오늘날의 비즈니스 및 정보 기술의 추세에 능동적으로 대처하고 발생되는 문제들을 성공적으로 해결하여 생존하기 위한 정보 시스템의 구축, 운영, 유지 보수 등을 포함하는 새로운 컴퓨팅 기술이 절실히 요구되고 있다.

1990년 이후 중앙 집중적인 컴퓨팅 환경에서 벗어나 여러 대의 컴퓨터를 이용해 일정한 역할을 분배하여 네트워크 상에서 상호 협력하여 작업함으로써 언제 어디서나 필요한 정보를 얻을 수 있는 컴퓨터 시스템인 분산 컴퓨팅(Distributed Computing) 환경이 시작되었다. 소프트웨어 부문에서도 이러한 환경을 현실적으로 가능하게 하는 다양한 미들웨어(Middle/Ware)들이 발표된 후로 기존의 단순 형태에서 벗어나 분산 환경에서 다양한 서비스를 제공하는 분산 시스템이 활발히 전개 되었다.

비즈니스가 복잡해지고, 정보 기술이 급변하고 다양화하면서 소프트웨어 개발 방법에도 많은 변화를 가져왔는데, 소프트웨어 공학(Software Engineering)을 바탕으로 한 모델링 기법, 개발 프로세스, 관리 기법등의 소프트웨어 개발 방법론이 점진적으로 발전되면서 실제적인 프로젝트에 적용되고 있다.

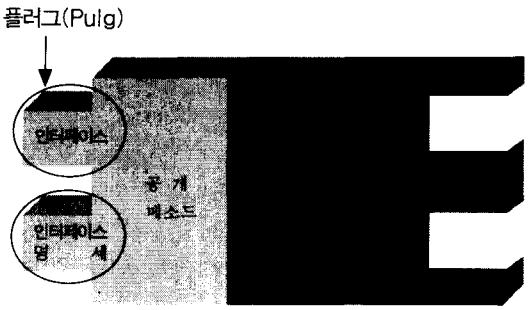
분산 컴퓨팅 환경에서의 클라이언트/서버 기술은 인터넷 기술과 객체지향 패러다임이 정보기술의 모든 분야에 적용되면서 분산 객체 컴퓨팅(Distributed Object Computing)으로의 전환이 이루어지고 있다. 분산 객체 컴퓨팅은 단지 기술만의 변화가 아닌, 오늘날의 컴퓨팅 환경의 문제를 해결하기 위한 패러다임이며, 소프트웨어 공학 프로세스를 확장하고 적용시키는 촉진제인 것이다. 이러한 기술 발전에 대한 소프트웨어 개발 기술의 성공 요소는 재사용 아키텍처를 기반으로 하는 객체지향 기술(Object Oriented Technology)로서, 이에 대한 효율적인 활용 전략이 주요 이슈로 대두되고 있다.

최근에는 객체지향 기술에 근간을 둔 컴포넌트 기반의 소프트웨어 개발 기술(CBD: Component Based Development)이 정립되어 가고 있으며, 또한 컴포넌트 기술의 프로임워크를 제공하는 업체간의 표준 전략은 기존의 DCE(Distributed Computing Environment) 기반의 환경에서, 객체 기술과 인터넷 기술이 접목된 MS사의 COM+와 OMG(Object Menegement Group)의 CORBA, SUN사의 EJB(Enterprise Java Bean) 및 J2EE(Java 2 Enterprise Edition)로 서로 보완적 관계를 유지하면서 발전하고 있다. CBD는 단순히 소프트웨어 개발 생산성에 목표를 둔 것이 아니라 비즈니스 객체를 통한 업무 재사용까지도 의미한다.

지금까지 여러 연구와 산업 현장의 프로젝트에서 많은 방법론들이 적용되어 왔지만, 같은 방법론이라고 해도, 모델과 표기법들이 서로 달라 많은 혼선과 어려움을 겪었다. 그러나 OMG에서 UML(Unified modeling Language)을 객체지향 방법론의 표준으로 제정함으로써 앞으로의 소프트웨어 개발에 있어 성공적인 확신을 갖게 하는 중요한 초석이 되었다. [2][4][5]

본 연구에서는 실질적 컴포넌트 프레임워크로 인식되고 있는 CCM(CORBA Component Model), EJB 및 J2EE와 COM+를 통해 구축할 수 있는 컴포넌트의 아키텍처를 수립하고 UML을 이용한 명세화를 제안한다.이를 위해 제 2장에서는 컴포넌트의 형상과 세가지 프레임워크를 비교 분석하고 제 3장에서 컴포넌트 아키텍처를 수립하여 참조 모델을 제시하고 제 4장에서는 컴포넌트 아키텍처의 명세서를 UML을 이용하여 작성한다. 제 5장에서 본 연구의 결론과 향후 연구 과제를 기술한다.

Ⅱ. 컴포넌트 프레임워크



컴포넌트들은 일반적으로 위의 [그림 1]과 같은 형상

표 1. 컴포넌트 플랫폼 비교표

Comparison Criteria	CORBA	EJB	COM+
Complexity of Specification	Complex(Extended IDL)	Somewhat complex	The most complex
Comp. Spec. Language	CIDL	None(tools, Java)	IDL
Separation of Public I/F	Separated IDL interface	No separated interface	Separated through interface
Communication Protocols	IIOP	RMI/IIOP, JNDI	RPC
Customizability	Fully Supported(Dynamic binding, Plug-in Objects)	The same as left colm. (at Deployment time)	Limited supported (containment, aggregation)
Scalability	Fully scalable and distributable	Fully scalable and distributable	Not supported in Win 2000
Interoperability	Source code level through IDL	Binary level within a specific EJB server	Binary level (Windows platform)
Required Runtime Engines	CIF Engine	EJB container, EJB server	COM Library
Runtime Efficiency	Not yet tested	moderate	Superior for C/S Application
Development Tools/CASE	Not available yet	A few(Web Logic,..)	Most MS Visual Studio
Market Share	Some	A small number	Thousand of products
Market Prediction	Most traditional, academic	Emerging for large grained components	Greatest market for small-grained components

으로 구성되며 이에 대한 세부 사항은 다음과 같다 :

(1) 플러그(Plug) : 다른 컴포넌트와의 조립을 위해 제공되는 인터페이스와 그에 대한 행위 명세(behaviour specification)

(2) 공개 메소드(Revealed behaviour) : 개발자가 컴포넌트를 채택함에 있어 보조하기 위해 필요한 측면에서의 공개가 요구되는 메소드와 플러그 대한 상세한 명세

(3) 비공개 메소드(Hidden behaviour) : 컴포넌트의 사용자와 무관한 구현 세부 사항과 설계 사항

(4) 실행부(Executable Part) : 컴포넌트의 기능을 실행

〔그림 1〕의 컴포넌트는 공개 메소드와 비공개 메소드를 동시에 갖기 때문에 그레이 박스(grey box)형의 컴포넌트이다. 비공개 메소드가 없는 컴포넌트는 화이트 박스(white box)형이고 플러그를 제외하고 공개 메소드를 갖지 않는 컴포넌트는 블랙 박스(black box)형의 컴포넌트이다.〔3〕〔7〕

위의 〔표1〕은 세가지 컴포넌트 프레임워크를 비교한 결과이다.

Ⅲ. 컴포넌트 참조 모델

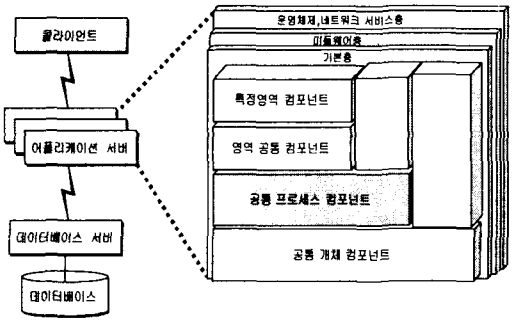
참조모델을 제시하고, 이들의 공통성과 차별성을 식별하는 영역공학을 통한 특정영역 아키텍처 모델을 구성한다. 그리고 특정영역들 간의 공통성과 차별성을 식별하여 비즈니스 공통 컴포넌트를 추출하고, 비즈니스 공통 컴포넌트를 이용하는 특정영역 아키텍처 모델을 수정, 이를 전체영역 아키텍처 모델 구성에 이용한다. 전체적인 과정은 반복과 정제의 과정을 거쳐 비즈니스 공통컴포넌트와 아키텍처 참조모델을 구성한다.

3.1 참조 모델 구성

어플리케이션을 생성하기 위한 프레임워크를 제공하도록 참조모델을 제시한다. 특정 영역의 어플리케이션을 개발하려 할 때 참조 모델에서 유도되는 소프트웨어 아키텍처를 이용함으로써 컴포넌트 기반 소프트웨어 개발을 효율적으로 수행할 수 있다.

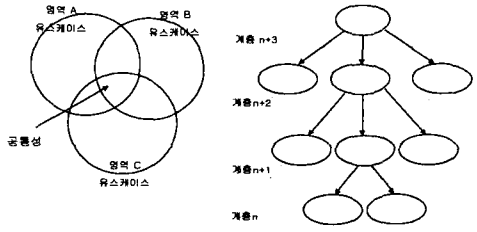
참조 모델은 하향식으로 정의하게 된다. 분산 객체 컴

퓨팅 환경을 고려하여 n-tiers에서 어플리케이션이 동작될 때, 프리젠테이션, 비즈니스 로직, 데이터를 분리한다. 비즈니스 로직은 특정 플랫폼에서 기본적인 서비스를 지원받으며 처리된다. 비즈니스 로직을 수행하는 컴포넌트는 비즈니스 공통 계층과 영역 계층으로 나눈다. 비즈니스 공통 계층은 세분 정도에 따라 공통 개체 컴포넌트와 공통 프로세스 컴포넌트로 구성되고, 영역 계층은 영역 공통 컴포넌트와 특정 영역 컴포넌트로 계층을 구분하였다.[8]

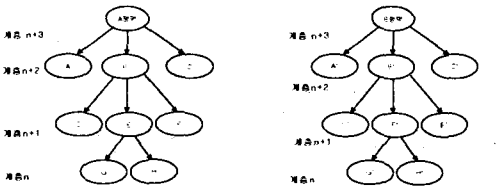


3.2 기존 시스템 분석

관련된 영역에 레가시 어플리케이션을 분석한후 이들간의 공통성을 유스케이스 중심으로 추출한다. 참조모델을 근저로 하여 컴포넌트간의 의존성과 관계를 식별해나가야한다. 이러한 여러 레가시 시스템을 분석하여 공통성과 차별성을 이용하여 해당 영역의 영역 모델을 만든다. 물론 레가시 시스템 산출물은 이를 기초로하여 새롭게 정제된다.



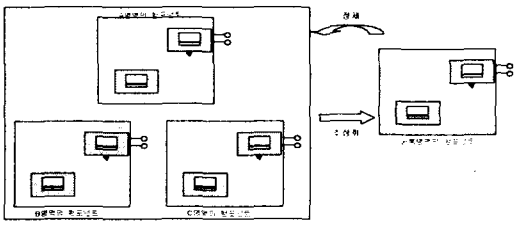
3.3 특정 영역별 아키텍처 구성



A영역의 계층$n+2$과 B영역의 계층$n+2$ 에서 유스케이스를 분석 공통성과 차별성을 분석한다. 마찬가지로 다른 영역에서도 마찬가지로 공통성과 차별성을 구별한다. 공통적인 것은 공통 컴포넌트 영역으로 묶는다. 차별성은 그나름대로 컴포넌트로 잡는다. 정확한 경계와 정제의 과정을 거쳐야한다.[8]

특정 영역에 대한 영역분석과 모델링, 아키텍처 설계의 과정은 다시 모든 영역에서 영역과의 과정을 반복적으로 수행한후 다시 만들어진 결과물들의 공통성과 차별성을 식별해야한다.

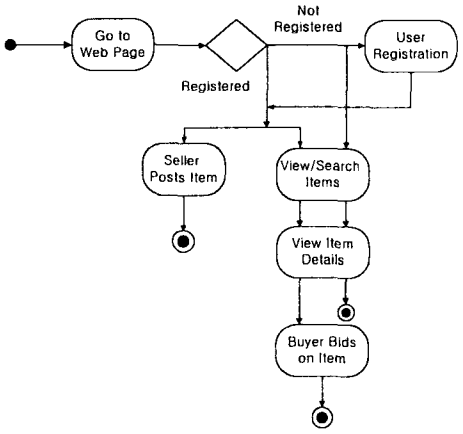
이때 컴포넌트의 가장 중요한 특징을 반영하는 재사용성과 대치가능성을 확신해주는 것은 인터페이스인데 이것에 대한 공통성을 찾는 방법은 [그림 5]와 같다.

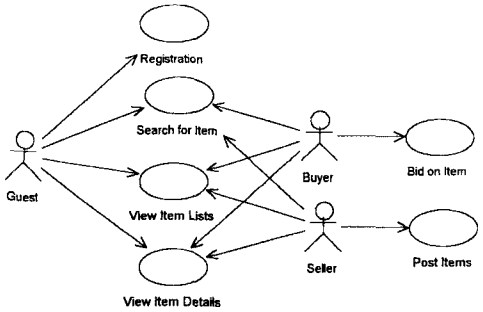


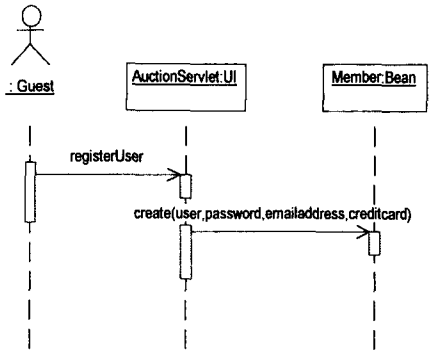
위와 같은 방법을 적용하여 정확한 컴포넌트 사이의 의존관계를 정확히 파악하고 정제하여 다음과 같은 특정 영역 영역아키텍처 모델을 구성한다. 이때 특정 영역 아키텍처 참조모델의 구성은 계층화와 참조모델의 원칙을 준해서 행해진다.

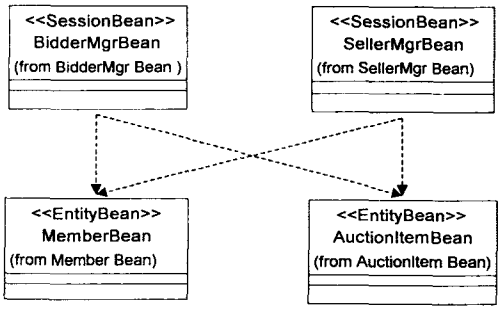
Ⅳ. 참조모델을 활용한 UML 명세

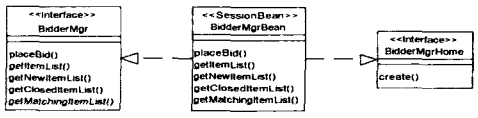
앞서 정제된 아키텍처에 기반하여 영역 분석을 통해 얻어진 활동도, 사용사례도, 순차도, 클래스 다이어그램의 일부를 [그림 6~10]에 제시하였다.[7][8]











Ⅴ. 결론 및 향후 연구 과제

컴포넌트를 활용한 어플리케이션을 생성하고 이의 명세화를 위한 프레임워크를 제공하도록 참조모델을 제시하였다. 이로써 특정 영역의 어플리케이션을 개발하려 할 때 참조 모델에서 유도되는 소프트웨어 아키텍처를 이용함으로써 컴포넌트 기반 소프트웨어 개발을 효율적으로 수행할 밑바탕을 마련하였다.

참조 모델은 분산 객체 컴퓨팅 환경을 고려하여 n-tiers에서 비즈니스 로직을 수행하는 컴포넌트를 비즈

니스 공통 계층과 영역 계층으로 나누고, 비즈니스 공통 계층은 세분 정도에 따라 공통 개체 컴포넌트와 공통 프로세스 컴포넌트로 구성하고, 영역 계층은 영역 공통 컴포넌트와 특정 영역 컴포넌트로 계층을 구분하였다.

제안된 참조 모델을 활용하면 잘 명세화된 규격에 근간하여 컴포넌트를 생성하게 되므로 엔지니어의 개발에 대한 부하를 줄이고 컴포트의 무결성과 품질을 보장할 수 있는 장치를 마련할 수 있다.

향후 연계되는 연구를 통해 제안한 참조모델과 명세서를 활용하여 CCM, J2EE와 COM+로 매핑되는 자동화 도구를 개발하고자한다.

참고문헌

[1] CORBA와 DCOM의 통합, 장연세,금영욱, 한국정보과학회, 1999년 7월호

[2] CORBA 3 프로그래밍 바이블, 장연세, 금영욱, 도서출판 그린, 2000년5월

[3] Modelling Software Components, Stuart Kent, John Howse, IEEE, 1998

[4] Reusing MS-Windows Software Applications Under CORBA Environment, Re-Chi Lin, IEEE, 1998

[5] A UML-based Method to Specify the Structural Component of Simulation-based Queuing Network Performance Models, Nunzio-Nicolo Savino, IEEE, 1999

[6] CORBA 상에서의 그룹 객체의 구현에 관한 연구, 류기열, 이정태, 변광준, 한국정보과학회, 1999년

[7] UML을 기반으로 한 실무 중심의 객체지향 방론, 조은숙,김수동,류성열, 한국정보과학회, 1999년

[8] 클래스 부품 재사용을 위한 객체의 추출과 이해, 한정수,송영재, 한국정보과학회, 1999년

저자소개

장 연 세

1993년 서울산업대학교 전자계산학과(공학사)

1995년 수원대학교 전자계산학과(이학석사)

1998년 아주대학교 컴퓨터공학과(박사과정 수료)

1998. ~ 99년 (주)필컴 기술연구소 팀장

2000. ~ 2001.4 (주)한국정보컨설팅 기술연구소장

20001 ~ 현재 한국컴포넌트컨소시엄 위원

2001.3 ~ 현재 수원과학대학 인터넷정보과 겸임교수

2001.5 ~ 현재 (주)한국후테로시스템 기술연구소장