

## 컴포넌트를 이용한 게임 개발의 분석 및 설계에 관한 연구

정해룡\*, 정광호\*\*

\* 중부대학교 전자계산학과 석사과정,  
gamastra@hanmail.net

\*\* 중부대학교 정보공학부 교수, khjung@joongbu.ac.kr

A Study on the Design and Analysis of Component based Game Development

Hae Ryong Jung\*, Kwang Ho Jung\*\*

### Abstract

The objective of this paper is to systematically establish the development of game software through the fusion of game implementation and component methodology. The development of game software has inconsistently accomplished without the consistent frame for the analysis and design of game development in the domestic environments. Therefore in case of the development of game software based on game component, although a lot of resources were required in the initial the accumulation of experience and the technology of reuse lead to the efficiency of the maintenance ultimately.

*Key Words* : Component

### 1. 서론

최근 영상, 음반, 애니메이션, 게임 등 문화콘텐츠의 중요성이 증대되면서 전 세계적으로 문화콘텐츠 산업이 급속하게 성장하고 있다. PC게임, 비디오게임, 업소용게임, 온라인게임, 모바일게임 등 게임산업의 각 장르도 그 성장률은 계속 높아지고 있다. 특히 첨단 제작기술 및 이동통신 기술의 발전에 따라 게임이 구현되는 플랫폼도 기존 오프라인과 온라인으로부터 모바일에 이르기까지 다양해지고, 이에 따라 게임제작도 더욱 첨단화되고 복잡해지고 있다. 이로 인해 게임개발 또한 최소한의 비용으로 전문적인 방법의 체계적인 절차를 거쳐서 제작되어야 한다. 그럼에도 불구하고 국내에서는 현재까지 대부분 게임의 분석과 설계에

대한 전문적인 개발방법이 없이 게임 시나리오에 의존해서 무계획적으로 게임 소프트웨어를 개발하고 있는 실정이다. 그러므로 본 연구에서는 이러한 문제를 해결하기 위해서 소프트웨어 공학 기법인 컴포넌트 방법론을 게임 제작에 접목하여 게임 소프트웨어 개발을 체계적으로 정립해보고자 한다.

### 2. 컴포넌트

소프트웨어 컴포넌트란 잘 알려진 특정 기능을 수행하도록 구현한 독립적인 단위 소프트웨어로, 구체적인 구현은 컴포넌트 사용자에게 숨기고 잘 정의된 인터페이스를 통하여 해당기능을 제공한다. 하지만 이러한 컴포넌트의 개념

및 특징에 대한 명확한 정의가 아직 제대로 이루어지지 않고 있어 다양한 시각에서 접근이 가능하지 않다.

현재 필요한 정보를 언제, 어디서나 쉽게 얻을 수 있는 인터넷 환경이 보편화되어 있다. 그리고 이 기종 컴퓨터간의 연동기술의 발전으로 인터넷상의 소프트웨어 부품을 기종에 상관없이 사용할 수 있게 되었다. 부품을 조립해서 제품을 만들어 내는 것처럼 부품화된 소프트웨어들을 조립하여 완성된 소프트웨어를 만들어 낼 수 있는데, 이러한 독립된 기능 단위의 소프트웨어에 대한 부품을 컴포넌트라 한다.

컴포넌트에 대한 관점은 여러 연구의 접근 방법에 따라서 다양한 형태로 표현된다. 컴포넌트는 하나의 클래스이거나 또는 클래스의 집합이 될 수 있다. 또한 객체의 집합이거나 프레임워크가 될 수 있다. 그리고 소스코드의 조각이거나 모델이 될 수 있고 그들의 조합이 될 수 있다. 다양한 형태로 존재하는 컴포넌트에서 가장 중요한 공통점은 바로 컴포넌트가 재사용의 단위로 쓰인다는 것이다. 컴포넌트에 대한 다양한 다른 관점에서 나타나는 공통적인 요소들을 정리하면 다음과 같은 정의를 내릴 수 있다.

- 컴포넌트는 하나의 시스템의 일부분으로서 그 시스템의 나머지 부분과 분리 될 수 있다.
- 컴포넌트는 개발자가 인식할 수 있는 개념으로 표현되어야 한다.
- 컴포넌트는 오직 그 컴포넌트를 위해서 정의된 인터페이스만을 통해서 다른 컴포넌트와 통신할 수 있다.
- 컴포넌트가 요구하는 것과 요구되는 것을 명확히 나타내야 한다.
- 컴포넌트는 다른 컴포넌트들로부터 독립적으로 분산 될 수 있어야 한다.[1, 2]

## 2.1 컴포넌트의 특성

현재 컴포넌트의 특성에 관한 연구는 다양한 각도에서 진행되고 있는데 이들을 종합하여 컴포넌트가 갖추어야 할 기본 특성을 규정하고자 한다. 일반적으로 컴포넌트는 아래 <표 1>과 같이 캡슐화, 서비스 중심, 식별성, 독립성, 신뢰성, 연결성, 일반성, 교체가능성, 개발용통성의 9가지 특성을 갖추어야 하고, 이것을 바탕으로 컴포넌트 기반 소프트웨어 개발에 대한 바람직한 시각을 가지고 컴포넌트를 평가할 수 있을 것이다.[3]

특성	내 용
캡슐화	컴포넌트는 서브 컴포넌트를 포함하여 자신을 캡슐화 시킴으로써 물리적 구현상의 경계를 표시한다. 기능성은 인터페이스를 통해 제공되는 서비스에 의해서만 노출된다.
서비스 중심	서비스는 제공되는 것만 표시한다는 점에서 필수적이다. 컴포넌트가 제공하는 서비스가 실제로 수행되는 방법은 수요자가 알 수 없어야 한다.
식별성	컴포넌트와 인터페이스는 각각 명확한 식별을 가져야 하며, 다른 컴포넌트 또는 애플리케이션에서 대체사용된 경우, 추적 가능해야 한다.
독립성	컴포넌트는 다른 컴포넌트의 구현에 의존하지 않아야 한다.

신뢰성	컴포넌트는 우선적으로 구현이 증명되어야 하며, 실행 시 부작용이 없어야 한다.
연결성	컴포넌트는 컴파일과 링크 프로세스를 통해 정적인 방법이 아닌, 실행 시간에 인터페이스를 통해 다른 컴포넌트와 역동적으로 연결할 수 있어야 한다.
일반성	인터페이스에 사용되는 기술은 다중 환경에서 사용할 수 있어야 하고, 이는 공통 대화 표준(CORBA, DCOM 등)으로 정의된다.
교체가 가능성	새 컴포넌트가 동일 인터페이스를 통한 같은 서비스를 제공하는 한 컴포넌트 대치는 간단해야 한다.
개발용 통성	컴포넌트는 여러 가지 기술을 사용하여 전개할 수 있어야 하지만 동일한 인터페이스를 유지해야 한다.

표 1. 컴포넌트의 특성

## 2.2 컴포넌트의 목적

### 2.2.1 재사용의 효과제고

컴포넌트가 시스템구축의 기본단위로 인식됨에 따라 재사용에 대한 관심도 함께 증가했으며, 여러 사람들은 컴포넌트가 재사용을 위한 최적의 단위라는 사실을 믿기 시작했다. 재사용을 통해 얻을 수 있는 여러 가지 이득 중 중요한 것들은 배포 소요시간의 절감, 개발비용의 절감, 배포된 정보시스템 내에서 발생하는 에러의 수 절감등의 이점이 있다.

### 2.2.2 소프트웨어의 유연성 확보

유연성은 애플리케이션이 요구사항의 변경을 수용할 수 있도록 하는 것으로 설명된다. 요구사항의 변경이 그 시스템의 제한된 부분에만 영향을 미친다면, 시스템의 유연성은 증가한다. 그러므로, 시스템을 여러 개의 작은 컴포넌트들

로 나누고, 어느 컴포넌트에 대한 변경이 다른 컴포넌트들에 끼치는 영향을 최소한으로 하는 방향으로 만들어야 한다. 유연성을 고려하며 시스템을 설계하는 방법에는 두 가지가 있다. 첫 번째 방법은 요구사항의 변경이 가능하도록 컴포넌트들끼리의 영향을 최소화하는 방향으로 시스템을 여러 컴포넌트들로 나누는 것이다. 이 방법에는 시스템을 컴포넌트단위로 나눌 때 미래에 일어날 수 있는 변경사항에 대한 예측력이 필요하다. 두 번째 방법은 다른 컴포넌트들에 영향을 끼치지 않는 컴포넌트들을 발견하는 것이다. 이러한 컴포넌트들의 컨트랙트들은 어떠한 타입의 변경도 허용할 수 있어야 한다.[4, 6]

### 3. 컴포넌트를 이용한 게임 분석

모든 소프트웨어는 정확한 분석 및 설계가 전제되어야 효율적으로 개발 될 수 있다. 따라서, 본 장에서는 컴포넌트 개발 기술을 이용한 고도리 게임을 예를 들어 분석을 하고자 한다.

#### 3.1 컴포넌트 기반 개발 단계

컴포넌트 기반 개발 단계는 컴포넌트 생성과 조립, 두 과정이 동시에 다른 조직에 의해 진행되고, 컴포넌트 획득에 대한 새로운 과정이 있다. 기존에 존재하는 컴포넌트는 컴포넌트를 저장소에서 획득하고 존재하지 않는 컴포넌트는

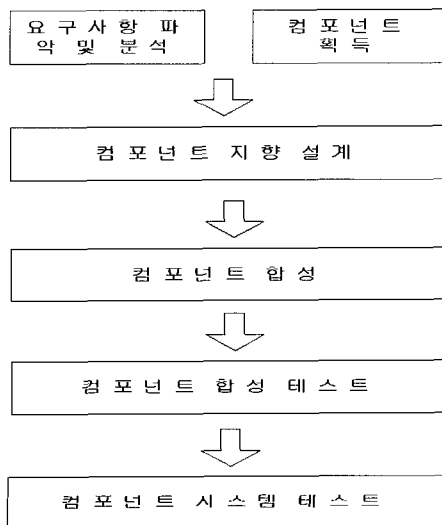


그림 1. 컴포넌트 기반 개발 방법

컴포넌트를 생성한다. 요구사항 분석으로 컴포넌트를 생성하거나 저장소로부터 컴포넌트를 획득한 후 조건에 맞게 컴포넌트 지향 설계를 하고 다른 컴포넌트에서 필요한 기능을 가져와서 컴포넌트를 합성해서 새로운 컴포넌트로 재구성한다. 일련의 컴포넌트 합성으로 이루어진 합성된 컴포넌트의 error 여부를 테스트하는 컴포넌트 합성 테스트를 한다. 그 후 전체 시스템을 대상으로 기능적인 테스트를 수행한다. 컴포넌트 기반 소프트웨어 개발 프로세스의 생명 주기는 요구사항 파악, 객체 지향 분석, 컴포넌트 설계, 컴포넌트 구현, 테스트라는 각 단계가 반복적으로 이루어지는 형태로 구성된다.

#### 3.2 일반적인 요구사항

어떤 일을 할 때 목표를 확립하고 행동하는 것과 그렇지 못한 것은 매우 큰 차이를 보인다. 게임 소프트웨어를 개발하기 위한 노력의 성공 여부도 시스템을 만들기 위한 목표를 프로젝트의 초기에 정립할 수 있는가에 달려있다. 요구사항 파악은 시스템이 만족시켜야 할 요구사항을 발견하고, 이를 명세화하는 과정이라 할 수 있다. 이 단계에서는 계약 또는 제안서에 명기된 부분보다 세부적으로 시스템의 요구사항을 세분화하고, 그 결과를 융합하여 사용자가 원하는 시스템에 대해 서술한다. 이처럼 요구사항 파악은 시스템에 만족시켜야 할 기능, 성능, 인터페이스 등을 규명해야 하며, 일반적으로 생각되는 것보다 어려운 부분에 속한다. 시스템을 개발하는 데 있어 가장 중요한 부분은 사용자 또는 고객의 문제점을 이해하는 것이며, 어떤 것을 제공할 것인가를 정확히 파악해야 하기 때문이다.

##### 3.2.1 프로젝트 필요성 파악

관리자나 분석가 또는 개발자들은 시스템 요구사항과 기본적인 소프트웨어 아키텍처에 대한 기반을 확립하고, 많은 가정들을 긍정적으로 평가하거나 삭제한다. 아이디어를 찾아내는 여러 활동들과 위험 요소에 대한 기본적 식별, 외부 인터페이스에 대한 식별, 시스템이 제공해야 하는 주요 기능에 대한 인식, 그리고 기본적 개념을 증명하기 위한 프로토타입도 도입단계에서 나타날 수 있다.

##### 3.2.2 위험요소 파악

많은 프로젝트의 경우, 계획된 일정보다 늦어지는 경향이

있다. 이와 같은 기간의 연장에는 여러 요인들이 있지만, 가장 근본적인 이유는 프로젝트에서 생겨날 수 있는 위험요소들을 초기에 파악하고, 이를 극복하기 위한 체계적인 접근 방법이 없기 때문이다. 즉, 익숙하지 않은 도메인이나, 매우 많은 양의 데이터 처리, 개발 경험의 부족, 신기술 도입 등을 포함한 다양한 경우의 위험 요소를 초기부터 파악하고 이를 효과적으로 풀어나가야 할 것이다.

### 3.2 객체 지향 분석

분석은 요구사항 파악 단계에서 찾아진 요구사항을 모델링하는 과정을 말한다. 이 분석 단계에서는 개념적 단계의 모델이 작성되며, 설계 단계에서 실제적인 모델이 구축되기 위한 기반을 제공한다. 또한 분석이란 원하는 시스템이 어떤 일을 할 것인가에 대해 기술하면서 주어진 영역과 개발할 시스템에 대해 이해하고, 이를 모델링하는 것을 말한다. 따라서 객체 지향 분석이란 유즈케이스 모델을 바탕으로, 시스템을 객체 모델을 통해 모델링하는 것을 말한다. 이때, 시스템이 해야 하는 일들을 적은 유즈케이스 기술서에 대한 이해와 해당 분야에 대한 지식과 개발 경험을 필요하게 된다.

### 3.3 유즈케이스 다이어그램

유즈 케이스 다이어그램의 목적은 시스템을 개발하는 동안 시스템이 제공해야 할 기능 또는 서비스를 도식화하는데 있다. 그러므로 시스템에서 찾아낸 액터, 유즈 케이스, 그리고 이들간의 상호작용을 그림으로 표현한다. 액터란 시스템의 일부가 아니라 시스템과 상호 작용 하는 사람이 나 사물을 지칭하는 말로 게임에서는 일반적으로 “게임을 즐기는 사람(Gamer)”을 액터로 추출한다. 그리고, 유즈 케이스란 시스템이 액터에게 제공하는 기능을 말한다. 그러므로 시나리오를 기반으로 유즈 케이스 다이어그램을 작성

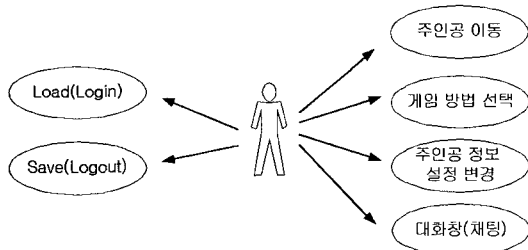


그림 2. 유즈 케이스 다이어그램

하면 (그림 2)와 같이 작성할 수 있다.

### 3.4 시퀀스 다이어그램

시퀀스 다이어그램은 객체간의 메시지 통신을 분석하기 위한 것으로 시스템의 동적인 모델을 아주 보기 쉽게 표현하고 있기 때문에 의사 소통에 매우 유용하다.

(그림 3)은 예제 게임의 시퀀스 다이어그램을 나타낸 것

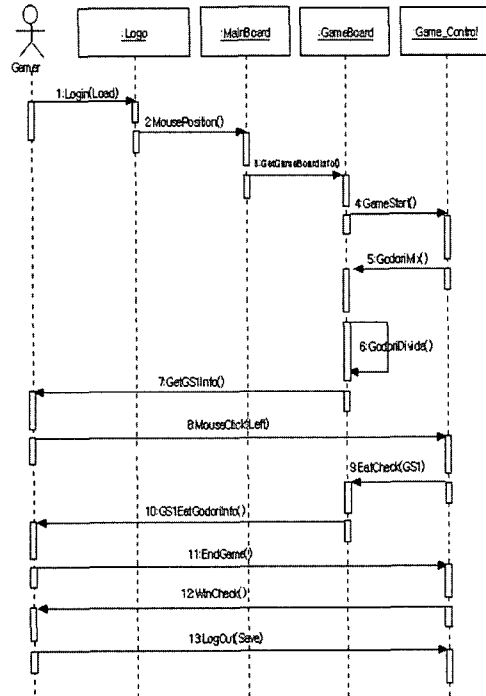


그림 3. 시퀀스 다이어그램

이다.

## 4. 게임 컴포넌트 설계

설계는 분석 과정을 통해 나온 모델을 더욱 상세하게 표현하면서 실제계의 모습들을 컴퓨터 내부로 옮기는 과정이며, 자세한 구현 과정이 나타나지는 않지만 시스템이 어떻게 동작하는지에 대해 표현하는 과정이다. 설계의 결과로 나온 모델은 실제적인 프로그래밍을 하기 위해 효율적으로 표현되어야 하며 분석 모델에서 나타나지 않은 컴퓨팅 환경을 고려한 세부적인 사항들을 잘 나타내어야 한다.

따라서 컴포넌트 설계란 객체 지향 분석 단계의 객체 모델을 바탕으로 실제적인 시스템을 위한 제반 사항을 표현하는 과정을 말한다.

4.1 게임 컴포넌트 기반 처리 모델

게임 컴포넌트 기반 처리 모델은 다음 (그림 4)처럼 크게 컴포넌트 생성하고, 컴포넌트 분류해서 컴포넌트 저장소에 저장을 하게 된다. 컴포넌트 이용자가 컴포넌트를 요구하면 컴포넌트 관리에서 컴포넌트 저장소, 컴포넌트 추출 부분, 컴포넌트 이해 부분으로 이동하여 컴포넌트 이용자가 요구한 컴포넌트가 존재하면 컴포넌트 조립에 들어가게 된다. 컴포넌트 생성 과정은 시나리오 분석, 설계, 구현, 테스트, 인증, 등록 순으로 이루어지고, 컴포넌트 조립 과정은 도메인 분석, 아키텍처 구축, 컴포넌트 획득, 컴포넌트 변

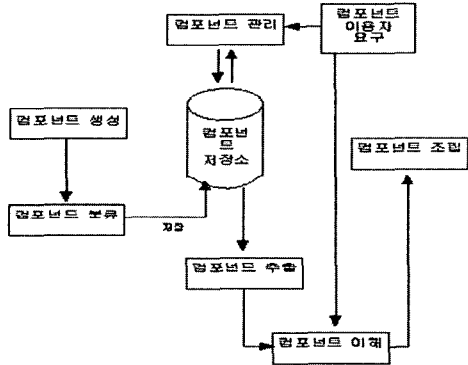


그림 4. 게임 컴포넌트 기반 처리 모델

형, 컴포넌트 합성, 컴포넌트 조립 테스트, 컴포넌트 통합 단계로 이루어진다.

4.2 게임 컴포넌트 계층도

게임 컴포넌트 계층도를 살펴보면 아래 (표 2)과 같이 게임 컴포넌트 아키텍처, 게임 컴포넌트 기반, 게임 공용 컴포넌트 개발, 게임 상세 개발 컴포넌트 개발로 이루어진다. 게임 컴포넌트 아키텍처에는 EJB, COM/DCOM, CORBA가 있고, 게임 컴포넌트 기반에는 컴포넌트 생성, 컴포넌트 합성,

게임 상세 컴포넌트 개발	맵(map), 캐릭터, 아이템, 무기와 장비, 버튼 조작, 화면 흐름 등
게임 공용 컴포넌트 개발	아케이드 게임, 롤플레이 게임, 어드벤처 게임, 시뮬레이션 게임 등
게임 컴포넌트 기반	컴포넌트 생성, 컴포넌트 합성, 컴포넌트 추출, 컴포넌트 인증
게임 컴포넌트 아키텍처	EJB, COM/DCOM, CORBA
	운영체제 및 Network Service

표 2. 게임 컴포넌트 계층도

컴포넌트 추출, 컴포넌트 인증을 하게 된다. 게임 공용 컴포넌트 개발에는 아케이드 게임, 롤플레이 게임, 어드벤처 게임, 시뮬레이션 게임 등이 있다. 게임 상세 개발 컴포넌트 개발에는 맵, 캐릭터, 아이템, 무기와 장비, 버튼 조작, 화면 흐름 등이 있다.

예제 게임의 컴포넌트 화면의 계층도를 살펴보면 (그림 5)과 같이 로고, 메뉴, 싱글플레이, 멀티 플레이, 도움말, 계

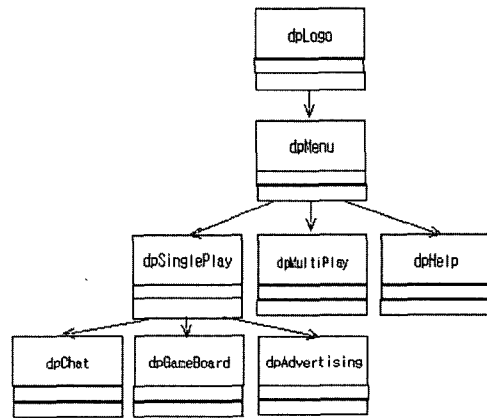


그림 5. 게임 컴포넌트 화면의 계층도

임창, 채팅창, 광고창 등으로 구성될 수 있다. 이와 같이 컴포넌트 계층도를 작성하게 되면 게임 개발자가 한눈에 전

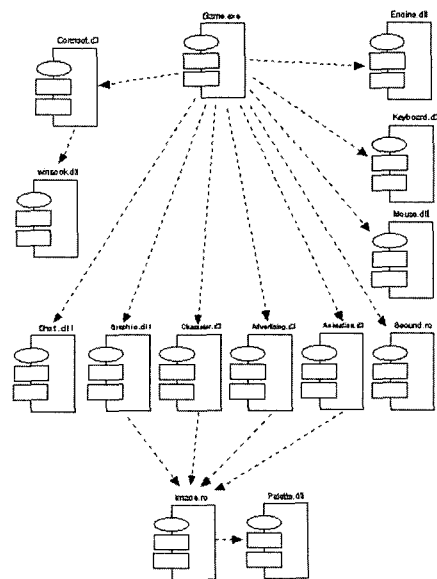


그림 6. 게임 컴포넌트 다이어그램

체적인 흐름을 파악할 수 있을 것이다.

#### 4.3 게임 컴포넌트 다이어그램

컴포넌트 다이어그램은 설계단계에서 작성하는 산출물로, 여러 파일들이나 클래스들 간의 관계를 나타낸다. 따라서, 논리적인 개념의 산출물이 아닌, 실제 구현을 통해 만들어지는 소스파일이나, 실행 파일들에 대한 관계를 표현한다.

(그림 6)는 예제 게임의 컴포넌트 다이어그램이다. 다음과 같이 게임 컴포넌트 다이어그램을 분할하여 작성할 수 있다.

#### 4.4 연구 결과

컴포넌트를 예제 게임 개발에 접목해봄으로써 체계적인 개발 단계를 설정할 수 있었고, 새로운 기능이 추가되어도 별다른 어려움 없이 삽입이 가능하였다. 그리고 효율적인 모듈화로 다수의 프로그래머가 일관성 있게 작업할 수 있었고 프로그래밍 도중에 발생할 문제를 미리 예측할 수 있어 안정성을 높일 수 있었다. 또한 컴포넌트 다이어그램의 사용으로 프로그래밍의 알고리즘 측면에서도 효율성을 높일 있었고, 제작과정이 문서화되어 있어 재사용성이 높은 클래스들을 많이 추출할 수 있었다. 하지만 단순한 아케이드 게임과 같은 규모가 적은 게임의 경우 컴포넌트를 이용할 경우 오히려 제작에 많은 시간이 걸렸고, 규모가 큰 게임일수록 제작 기간이 단축되었다.

### 5. 결론 및 향후 연구과제

컴포넌트 기반 게임소프트웨어 개발의 장점은 2가지로 요약할 수 있다.

첫째, 부품화 되어 있는 컴포넌트들을 이용하여 사용자가 원하는 소프트웨어를 조립하고 적시에 시장에 게임을 출시할 수 있다. 이를 통해 개발 시간을 최소화하여 단기간 내에 게임을 개발함으로써 경쟁력을 가질 수 있고 개발비용을 절감하는 효과도 볼 수 있다. 그리고 품질이 뛰어난 컴포넌트를 선택함으로써 개발에 따른 위험을 줄일 수 있다.

둘째, 새로운 기술에 신속하고 손쉽게 접할 수 있다. 끊임 없이 생겨나는 많은 정보 기술들을 채택한 컴포넌트들을 조립하여 소프트웨어 시스템을 만들어 냄으로써 수많은 정

보 기술이 있는 시대에 적절히 대처할 수 있다.

그리고, 규모가 크고 복잡한 게임 장르 등의 효율적인 유지보수를 위하여 게임 컴포넌트 개발 프로세스가 절실히 요구된다.

향후 연구 과제로서 본 논문에서 제안된 게임 컴포넌트 분석 및 설계를 사용하여 실현 가능하도록 구현하고, 데이터 저장소 관리에 대한 체계적인 정보 기술과 컴포넌트 관리 시스템의 자동화가 요구된다.

### 참고문헌

- [1] Szyperski C., Component Software, Addison-Wesley, 1999.
- [2] Desmond D' Souza & Alan Wiils, Object, Component and Frameworks with UML, Addison-Wesley, 1998.
- [3] Short K., Component Based Development and Object Modeling, Sterling Software, 1997.
- [4] Kirtland M., Designing Component-based Applications, Microsoft Press, 1999.
- [5] Alan W. Brown, Large Scale Component Based Development, Prentice Hall, 2000.
- [6] 정광호, 게임개발의 效率的인 維持補修 具顯을 위한 UML과 컴포넌트 應用에 關한 研究, 동국대학교 대학원 박사학위 논문, 1999.
- [7] 박현철, 분산 환경을 위한 컴포넌트 기반 소프트웨어 개발 프로세스, 중앙대학교 정보산업대학원 석사학위 논문, 1999.
- [8] John Cheesman, John Daniels, UML Components - 컴포넌트 기반 소프트웨어 명세를 위한 실용적인 프로세스, 인터뷰전, 2001.



정해룡

2000년 중부대학교 전자계산학과(학사)  
2000년 ~ 현재 중부대학교 전자계산학과(석사과정)  
관심분야: 소프트웨어공학, 게임

---



정광호

서울산업대학교 전산기기전공(공학사)  
건국대학교 산업대학원 컴퓨터응용(공학석사)  
동국대학교 대학원 전산통계학전공(이학박사)  
육군통신학교 마이크로웨이브교관(예비역 대위)  
중부대학교 전자계산소장, 학생처장, 사회교육원장등  
중부대학교 전자계산학전공 전임강사, 조교수, 부교수  
중부대학교 컴퓨터공학부 게임공학전공 부교수  
중부대학교 대학원장(현)  
관심분야: 게임공학, 소프트웨어공학

---