

# 게임엔진 개발 공정의 정형화를 지원하는 컴포넌트 저장소의 설계

김정중, 송의철, 박운재, 송호영  
경남대학교 컴퓨터공학과  
e-mail: songec@saekyung-c.ac.kr

## Design of a Component Repository Formalizing Game Engine Development Processes

Jung-Jong Kim, Eui-Cheol Song, Woon-Jai Park, Ho-Young Song  
Dept of Computer Engineering, Kyungnam University

### 요약

컴퓨터 게임은 소프트웨어 분야의 새로운 고부가가치 영역으로 발전되어가고 있으나 개발의 공정과 방법론에 있어서는 경험중심의 개발 방법을 고수하고 있으며 높은 자유도와 사용자 환경에 적용할 수 있는 빠른 패치의 개발에는 적합하지 못할 뿐 아니라 개발공정의 정형화 미비 등 많은 문제점을 가지고 있다. 본 논문은 이러한 문제를 지원할 수 있는 컴포넌트를 기반으로 한 게임엔진 개발방법에 적용할 수 있는 컴포넌트 저장소의 설계에 대하여 논하였다.

## 1. 서론

최근의 컴퓨터 게임 소프트웨어는 시스템 환경과 데이터의 구조가 복잡하고 다양한 형태로 처리되는 멀티미디어 시스템 형태로 개발되고 있다. 특히 인터넷 환경의 발전으로 인해 소프트웨어의 개발범위와 운영이 복잡하고 빠른 속도로 발전하고 있으므로 종전의 비정형화된 시스템 개발형태로서는 해결할 수 없는 많은 문제점을 가지고 있다.

따라서 본 논문에서는 정형화된 게임개발 공정과 이를 적용한 컴포넌트 기반 개발을 실현할 수 있는 컴포넌트 저장소를 설계하여 개발하는 방법을 제시하고자 한다.

이러한 방법을 적용한 소프트웨어 개발자는 게임의 기본개념으로부터 코드를 작성하는데 집중할 수 있도록 지원하고 개발기간의 단축과 개발된 엔진의 신뢰성 향상에 기여할 수 있도록 한다.

본 논문의 구성형태는 게임엔진의 재사용 가능한 영역을 살펴보고 적용 가능한 범위를 설정한다. 2장에서는 적용의 필요성에 대한 개발 시간과 단위공정

의 관계, 개발공정의 목표를 살펴보고 이를 기반으로 한 게임엔진의 재사용 가능 모듈과 컴포넌트 기반 개발형태를 정형화하며, 3장에서는 일반적인 게임공정에 이용되는 비정형화된 개발공정과 공정의 정형화과정, 개발 단계에서의 컴포넌트 참조방법, 모델적용을 위한 저장소의 설계와 적용후의 예측되는 결과에 대한 평가를 기술하였다.

4장의 결론 부분에서는 기존의 게임엔진 개발에서의 문제점을 해결할 수 있는 방안을 제시하고 이러한 시스템을 게임엔진에 국한된 형태에만 적용하는 형태에서 다양한 멀티미디어 시스템 구축에 필요한 컴포넌트 적용방법의 가능성에 대하여 논하였다.

## 2. 게임엔진의 재사용 가능 모듈

컴포넌트 기반을 적용하기 위한 참조 형태는 범용 컴포넌트 영역과 생성 컴포넌트 형태로 구분하며, 게임엔진 개발에 적용 가능한 재사용 가능한 모듈은 다음과 같다.

1) 범용 컴포넌트 영역

데이터 파일 로딩, 시스템 구성과 환경설정, 소프트웨어 구성, 코덱(CODE: Compression and Decompression: 압축과 압축 해제), 암호화와 해독 코드, GUI와 그래픽 처리요소, 기본적인 AI 요소, 사용자 인터페이스 개발 등

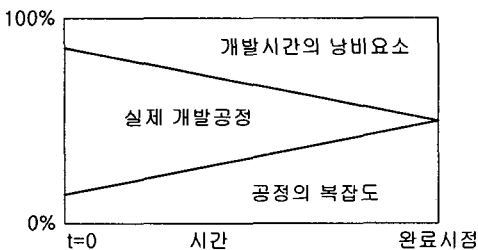
2) 생성 컴포넌트 영역

동영상 재생, 2D 그래픽 출력, 그래픽 데이터 출력, 특수효과, 광원처리, 3D 애니메이션, 카툰, 라이트 맵, 특수효과 커스틱, 게임 배경음악 재생, 효과음 처리, 클라이언트 데이터의 전송, 출력, 데이터 베이스 저장, 관리 등

3. 게임엔진과 컴포넌트 저장소

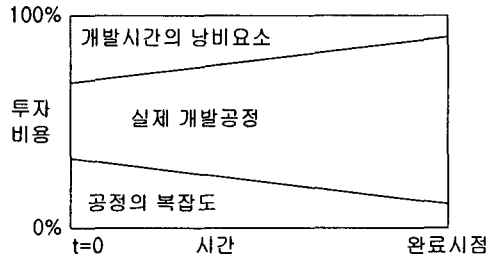
3.1 적용의 필요성

본 논문에서 제시하는 방법을 적용하기 위해서는 개발에 필요한 개발 툴과 컴포넌트가 재사용성과 확장성을 고려하여 게임엔진의 분석과 설계단계에서 구축되어야 하며, 게임엔진 개발의 전체적인 개발주기에 툴 부분과 컴포넌트 참조 내용이 포함되어야 한다. 물론 본 논문에서 제시하는 방법은 단일 시스템을 개발하는 측면에서 고려할 경우에는 낭비적인 요소로 작용할 수 있으나, 새로운 시스템을 연속적으로 개발시 적용하는 경우 많은 장점을 포함하고 있다.



[그림 1] 개발시간과 단위공정의 관계

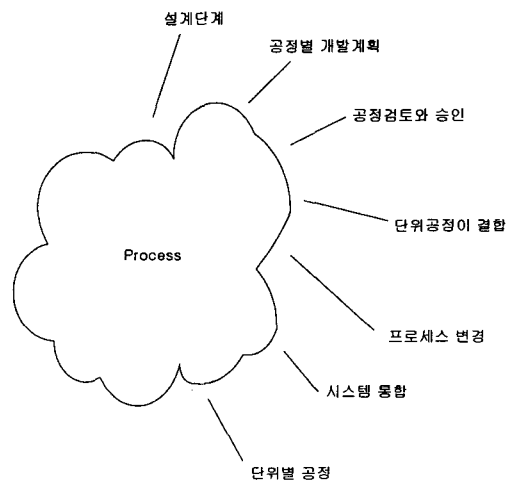
림 1]과 같이 개발 초기단계에서 수정되는 오류와 불필요한 공정을 감소시키는 방법을 제안하고 게임 엔진의 개발이 전체시스템 개발 공정 중 컴포넌트의 적용시점과 분류를 정의한다.



[그림 2] 개발공정의 목표

[그림 1]과 [그림 2]를 비교한 바와 같이 시간의 낭비요소는 감소하였으나, 종전의 비정형화된 게임 개발 공정에서는 시스템 완료시점이 공정의 복잡도가 증가하였다. 컴포넌트 기반 개발의 적용으로 공정은 단순화되고 게임개발의 주요목표인 게임성을 증대시키는 실제 개발공정을 처리하는 과정에 투자 비용을 할당할 수 있다.

3.2 공정의 분류

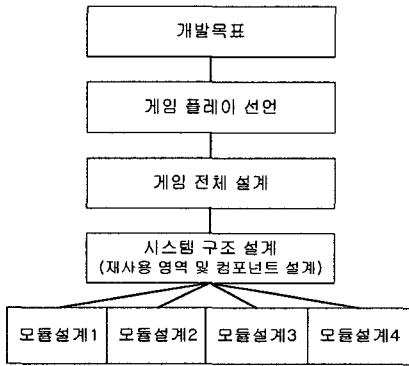


[그림 3] 비정형화된 개발공정

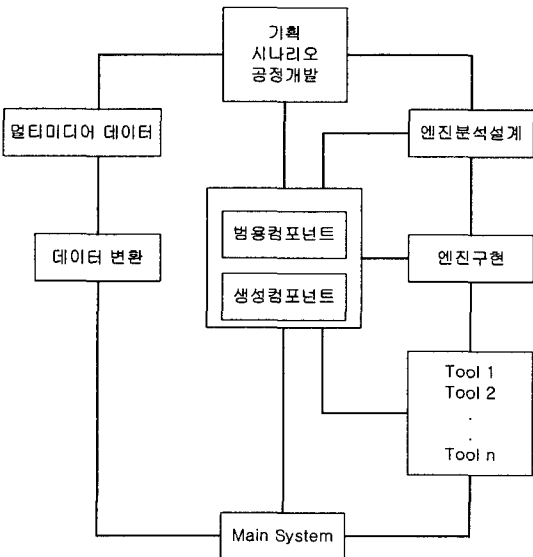
제시한 문제점을 해결하기 위한 적용 부분은 [그

[그림 3]의 일반적인 게임개발의 공정형태에서는 [그림 1]과 같은 문제점을 초래할 수 있다. 즉 공정을 더욱 세분화하여 구체적으로 분류할 때는 각 단위별 복잡도가 증가할 뿐만 아니라 재사용성이 감소함을 알 수 있다.

[그림 1]의 문제점을 개선하기 위한 목표는 [그림 2]와 같으며, 이러한 공정의 개선을 위하여 개발 단계별 과정을 [그림 4]와 같이 적용한다.



[그림 4] 공정의 정형화



[그림 5] 컴포넌트 참조단계 구분

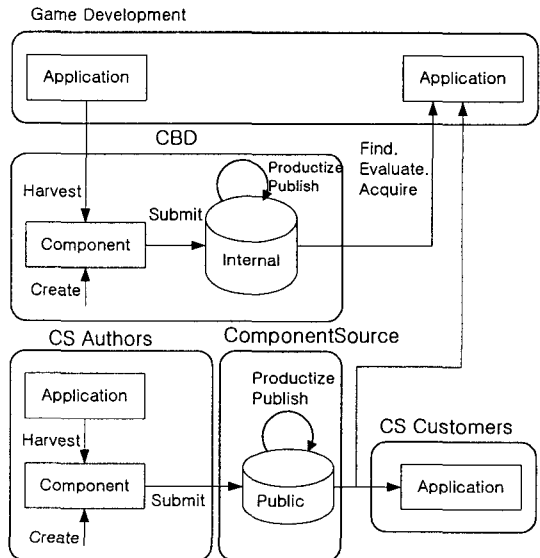
[그림 5]는 컴포넌트 참조단계를 설명하고 있으며, 간략화된 공정으로 전체 공정의 컴포넌트 적용 부분

은 단순히 엔진구현 단계에서만 적용하는 것은 아니며, 시스템의 초기단계부터 컴포넌트 적용이 가능한 부분들을 분류하여 정형화된 표준화작업을 수행하여야 한다.

엔진구현 툴의 종류는 3가지 형태로 구분할 수 있다. Map, character, event를 생성, 편집하여 생성 컴포넌트 저장소에 분류·저장하며, 저장 시 메타모델의 속성에 따라 분류방법과 저장형식은 차별화된다.

본 논문에서 제시하는 맞춘 컴포넌트 저장소는 정보의 통합과 교환, 공유를 위한 저장소의 개념이 아닌 게임소프트웨어 개발의 전 공정에서 효과적으로 이용할 수 있는 통합지원시스템 형태의 저장소이다.

즉, 아직 개발공정의 정형화가 부족한 게임소프트웨어 개발공정을 표준화하고 단위 공정별 적용 가능한 항목들을 재사용 할 수 있는 요소들의 구조를 정의하여 다양한 관점의 메타 데이터를 식별하고 분류할 수 있도록 정의한다.



[그림 6] 컴포넌트 적용 모델

[그림 6]은 컴포넌트 적용방법 모델을 제안하고 있다. CBD영역과 컴포넌트 소스영역은 게임소프트웨어 속성상 유연성을 가지고 확장이 가능하며, 중

전의 컴포넌트 기반 프로세스 적용방법 등에서 적용할 수 없는 부분들도 적용이 가능하다.

### 3.3 평가

컴포넌트 기반의 개발방법의 적용에는 [표 1]과 같은 장단점을 가지고 있다. 그러나 개발 기간과 비용적인 측면, 코드의 신뢰성 확보, 유지보수의 편리성을 고려할 경우 기존의 방법과 비교하여 몇 가지 단점을 가지고 있지만 이러한 단점들은 소프트웨어 개발비용과 신뢰성측면을 고려한다면 적용하는 편이 우수하다고 평가할 수 있다.

[표 1] 컴포넌트 기반 적용의 장단점

장점	단점
개발기간 감소	최초적용시 개발기간 증대
플랫폼의 다양화 가능	컴포넌트 사전준비
코드의 안정성보장	코드의 난이성 증대
재사용성과 관리의 편리성	최초적용시 개발기간 증대
개발자의 지식공유 가능	개발자 숙련
개발공정의 파악용이	관리비용 증대
개발자의 전문성 증대	기술의 유연성 감소

### 4. 결론

본 논문에서 제시한 컴포넌트 기반 게임엔진 개발방법의 적용은 평가에서 살펴본 바와 같이 여러 가지 장단점과 다음과 같은 문제점을 해결할 수 있다. 게임개발의 핵심적인 요소라고 볼 수 있는 전체시스템의 자유도와 창조성을 향상할 수 있으며, 소프트웨어 개발시간의 단축과 각 단위별 공정산출의 정확성을 증대시킬 수 있다. 특정한 범용 모듈개발에 집중화와 재사용성의 증대로 단순화된 개발방법의 적용이 가능하므로 정형화된 엔진개발이 가능하다.

따라서 플랫폼으로부터 독립성을 유지할 수 있으므로 코드의 안정적 유지가 가능하며, 재사용 가능성과 관리, 개발자의 지식공유, 시스템 개발주기의 공정이 보다 가시적인 적용이 가능하므로 특화된 시

스템개발 분야인 게임엔진 개발에 효율성을 가져올 수 있다.

물론 위와 같은 여러가지 장점을 가지고 있지만 컴포넌트의 수집과 최초 적용시 기술의 탄력성이 떨어진다라는 문제점도 가지고 있다. 이러한 문제점은 최초의 시스템 개발에 국한된 부분이며, 반복과 지속적인 개발에 있어서는 소프트웨어 생산성 향상에 크게 기여할 수 있다.

앞으로 이러한 시스템을 게임소프트웨어 개발에 국한된 형태에만 적용하는 형태에서 다양한 멀티미디어 시스템 구축에 필요한 컴포넌트 적용방법의 가능성에 대하여 체계적인 연구가 필요할 것이다.

### 참고문헌

- [1] Andrew Rollings, Dave Morris, "Game architecture and design", LLC, 2000
- [2] <http://componentsource.com/>
- [3] <http://www.gameonline.co.kr/>
- [4] Luqi, Jiang Guo, "Toward Automated Retrieval for a Softweware Componet Repository", IEEE Conference and Workshop on Engineering of Computer-Based Systems, March, 1999
- [5] 차정은, "컴포넌트 기반 개발 프로세스 지원을 위한 컴포넌트 저장소의 설계 및 구현", 대구가톨릭대학 박사학위논문, 2001
- [6] 진광호, "게임개발의 효율적인 유지보수 구현을 위한 UML과 컴포넌트 응용에 관한 연구", 동국대학교 박사학위 논문, 1999.