

UML을 사용한 온라인 게임 엔진 프로젝트 설계 연구

최 성
남서울대학교 컴퓨터학과

Study On the Object Oriented Design Project of Online Game Engine Using UML

Sung Choi
Dept. of Computer Science, NamSeoul University
sstar@nsu.ac.kr

요약

게임개발 프로젝트의 설계는 프로세스 구성과 게임 소프트웨어의 프로그램 산출물과 태스크의 관리는 개발자간에 연계하여 구성해야 한다. 기존 게임개발 프로젝트 설계는 프로젝트 활동의 정의 및 변경을 지원하고, 형상관리 도구는 버전제어, 작업공간 관리, 빌드 관리 등을 지원한다. 본 논문의 설계도구는 UML을 사용하여 게임 컴포넌트 기반 개발 프로세스 모델을 설계하고, 프로세스의 진행 상태 기록, 변경 요청에 따른 처리, 태스크별 작업 진행 보고서 작성, 산출물 등록과 변경, 버전 기록, 산출물 또는 서식 검색 등의 기능을 연구하였다. 또한 시스템 설계를 사용함으로써 게임 공학 활동에 참여하는 개발자들이 분산 개발 환경에서 게임 소프트웨어 개발에 필요한 정보와 산출물 관리를 생산성 있게 할 수 있다. 이 시스템 구성에 따라 정보를 공유하면서 설계의 체계화 및 표준화를 수행 할 수 있다.

Abstract

Game Designs & Developers that system designs improve, the product and the change of tasks to all developers are essential in achieving On-Line Game project. Existent On-line Game project management designs support the definition and the change of project activities, and configuration management designs support version check, workspace management, build management, etc.

In this paper, the proposed Design defines Game component based development process model, and achieves recording of process progression, processing the request of change, reporting the progression of each task, product registering and change, version recording, artifact or form search, etc. using UML. Furthermore, study on the stake holders get the systematic management and standardization by sharing information that are necessary in Game design & development and configuration management in distributed environment using the system integrated management design.

Key Words : Game process, artifact, UML, CBD, Process system management design

1. 서론

1.1 게임 시스템 구성

온라인 게임이 제공하는 가상세계의 모델링에서는 객체

지향 개념을 사용한다. 실세계를 예로 하였을 때 객체지향 개념이 맞지만, 게임 상의 사이버세계는 실세계를 기반으로 하고 있으므로 객체지향 설계가 적합하다. 그러나 기존의 사회형 머드 게임은 실세계를 기반으로 하고 있으므로 객체 지향적으로 가상세계를 모델링할 수 있는 MOO(Mud

Object Oriented)가 개발되었다. 그러므로 구체적인 UML을 사용한 온라인 게임의 객체지향 개념은 다음과 같다.

- 1) Class, Object 개념을 적용한 데이터 추상화(Data Abstraction)를 사용한다.
- 2) 시스템 실행 중(Run time)에 클래스의 확장(생성, 삭제)이 가능하며, 메타 클래스(Meta Class) 개념이다.
- 3) 독창적인 클래스 계층구조 및 객체 관리 구조이다.
- 4) 메시지 기반의 객체지향적 운영이다.
- 5) 관리자의 개입이 최소화된 게임 환경을 위해, NPC가 적절한 난이도를 유지하며 유전자 알고리즘을 이용하여 진화하도록 하였다.

1. 2 게임 서버/클라이언트 구성

온라인 게임의 사용자는 유저 인터페이스(UI : User Interface)를 제공하는 클라이언트(OMC-U)는 객체지향 개발 방법론을 이용하여 개발한다. 서버가 알고리즘 위주에서 기반 독립적인데 반하여, 클라이언트는 UI이기 때문에 기반 의존적이다. 클라이언트는 윈도우를 기반하고 있다. 그리고 개발 도구가 제공하는 객체지향 및 컴포넌트 기반 기술을 사용하는 것이 개발 및 운영에 유리하다. 시스템의 확장성이 원활하도록 객체지향 개념을 도입하여 유연성을 높여야 한다.

2. 게임 설계의 정의

2.1 게임 객체 특성

온라인 게임은 다음 네 가지 객체(특성)로 나뉜다. 이는 실제계를 모델링한 기존 사회형 네트워크 게임에서 이루어진다.

- 1) PC (Playable Character) : 온라인 게임을 즐기는 사용자가 게임 내에서 조종하는 객체로서 사람 또는 생명체에 해당된다.
- 2) NPC (Non-Playable Character) : 게임 엔진의 조종을 받는 온라인 게임내의 생명 있는 객체로서 사람 또는 생명체(몬스터 포함)이다.
- 3) Item : PC, NPC가 소유할 수 있는 게임상의 물건 객체이다.
- 4) Map : 게임의 배경이 되는 세계, 지형, 건물 등의 객체이다.

온라인 게임은 고정된 시나리오 없이 진행되며, 클래스와 객체는 각각 확장되고 유동적으로 변한다. 하지만, 게임 운영의 기본이 되는 시나리오상의 가정이 필요하다. 클래스의 확장, 객체의 생성 소멸에도 규칙이 필요하기 때문이다. 또한, 가장 기본이 되는 초기 시나리오, 즉 초기 클래스 계층과 객체들이 제시된다. 다음은 시나리오상의 가장 기본적인 가정/개념이다. 이 개념들이 실제로 어떻게 적용되는 세부 내용은 다음과 같다.

객체 (특성)	세부 내용	비고	
PC / NPC	종족(Race)	PC / NPC 클래스 계층에 적용되는 1단계	
	직업(Duty)	PC / NPC 클래스 계층에 적용되는 2단계	
	능력(Power)	객체의 속성(attribute)에 적용	
Item	용도(Usage)	Item 클래스 계층에 적용되는 1단계	
	명칭(Naming)	Item 클래스 계층에 적용되는 2단계	
	경제	화폐 기반 경제논리를 적용	
	수명(Lives)	모든 Item은 수명과 효력이 지속되는 시기가 있다.	
Map World	OMS-S의 구분관리 단위		

(표 1 : 사회형 네트워크 게임의 단계별 특성)

2. 2 클래스 자료의 구조

UML을 사용한 온라인 게임 엔진을 설계하려는 자료의 클래스 계층 구조는 독특하다. 기존의 객체지향 분석·설계 과정에서 클래스 계층을 도출한다. UML의 경우, 클래스 계층을 도출하는 것은 정적 모델링이지만, 동적 모델링도 중요하다. 이에 클래스 계층을 객체(Object)와 행동(Action)으로 구분하여 설계한다.

1) 객체에 대한 클래스 계층(OCH : Object Class Hierarchy) : 온라인 게임의 가상세계 내의 모든 객체에 대한 클래스 계층으로, PC, NPC, Item, Map에 대한 클래스 계층을 의미한다. 당연히 이 클래스 계층으로부터 객체 생성이 가능하다.

2) 행동에 대한 클래스 계층(ACH : Action Class Hierarchy) : 온라인 게임의 가상세계 내의 모든 객체가 행할 수 있는 행동(action)에 대한 체계적인 클래스 계층이다. OCH로부터 메소드(method)를 빼낸 것으로서 ACH로부터 객체는 생성되지 않는다.

이러한 클래스 계층 분리하는 이유는 명령의 처리속도를 향상시키기 위해서다. 사용자가 자신의 PC(Playable

Character)를 가지고 한 게임상의 명령은 PC를 행동하게 한다. NPC의 경우 게임엔진에 의해서 명령을 받아 행동한다. 이러한 행동은 행동의 주체객체와 행동의 대상객체, 그리고 행동에 사용한 물건객체에게 영향을 준다. 이러한 영향을 계산하는 것은 게임의 규칙에 의거해 이루어진다. 즉, 이 영향을 명령에 대한 처리로 계산하면 종료되는 것이다. 명령에 대한 처리 방식은 다음과 같다.

① 기존 게임의 경우 이러한 영향의 처리를 Switch-Case 문이나, 많은 if 문을 사용하여 처리하였다. 이 방식의 경우 규칙이 코드 형태로 유지되기 때문에 수정이 어려우며, 규칙 적용 시간이 길게 나타난다.

② 온라인 게임의 경우처럼 메시지에 기반으로 객체지향 운영을 사용하는 경우에도 비슷한 문제가 생긴다. 명령에 의해 유발되는 객체의 행동은, 메시지 기반 운영에선 대상 객체에게 메시지로서 전달된다. 그러므로 메시지의 처리는 대상객체에서 이루어져야 하며, 이는 명령에 대한 규칙 적용 책임이 대상객체에 있다. 이 경우, 각 클래스에 메소드의 형태로 규칙적용 코드가 분산 된다. 이 방식은 기존 게임의 방식과 같이 수정이 어려우며 규칙 적용에 걸리는 시간이 길다.

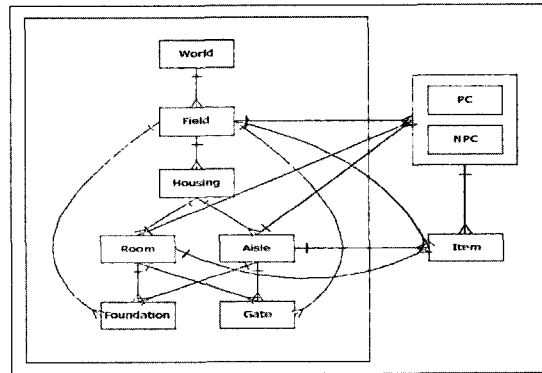
③ 기존 게임의 코드 형태로 규칙을 유지하는 방법이 처리속도 측면에서 단점이 있기 때문에 규칙을 자료구조의 형태로 따로 빼내는 방법이다. 이것을 수용하여 본 프로젝트에서도 규칙을 클래스 계층으로부터 분리하여 규칙 테이블(RT : Rule Table)시스템의 형태로 유지한다. 그리고 명령으로부터 발생하는 메시지의 처리를 대상객체에서 하지 않고, 게임엔진의 메시지 처리기가 모두 책임진다. 그렇기 때문에, 클래스 계층에는 메소드는 속성만 포함되지 않는다. 즉, 실제적인 규칙 적용 코드를 규칙 테이블의 형태로 빼어 내면, 처리 속도도 향상되고 규칙의 변화에도 유연하게 대처할 수 있다. 그러나 OCH에 굳이 불필요한 메소드들 필요가 없다. 객체가 어떤 행동을 할 수 있는지를 알기 위해 이름뿐인 메소드라도 두어야 한다.

하지만 해당 객체가 어떤 행동을 할 수 있는지는 그 객체에 의해서만 결정되는 것이 아니라, 행동에 사용되는 객체에 의해서 대상객체가 결정된다. 이것은 규칙 테이블에서 종합적으로 표현되기 때문에 OCH의 메소드는 필요 없다.

3. 객체관리 시스템

3. 1. 시스템의 구조

UML을 사용한 온라인 게임에서는 관리의 편의성을 높인 오브젝트 풀(Object Pool)을 설계하였으며, 이를 효율적으로 관리하기 위한 모델이다. 이 오브젝트는 일반적인 객체지향 기반으로 설계한다. 온라인 게임의 오브젝트는 가상 세계에서 발생하는 포함관계(Containment)가 기본이다. 즉, 어떤 사람이 무기라는 아이템을 소유하면, 그에 대한 객체들도 포함관계를 가지게 된다. PC/NPC와 Item을 고려하여 확장하면 다음[그림1]과 같다.



(그림 1 : 오브젝트 풀 구조(Structure of Object Pool))

이러한 오브젝트 풀의 관계는 PC/NPC의 움직임, Item의 주고받음/사용/취득에 따라

수시로 바뀌게 된다. 이러한 포함관계는 정적 포함관계와 동적 포함관계로 나눈다.

- 1) 정적 포함관계 : Map 객체간의 포함관계로서 실행 시간 중에는 변하지 않는다.
- 2) 동적 포함관계 : Map 객체와 PC/NPC/Item간의 포함관계, PC/NPC와 Item간의 포함 관계로서 실행 시간 중에 수시로 바뀐다.

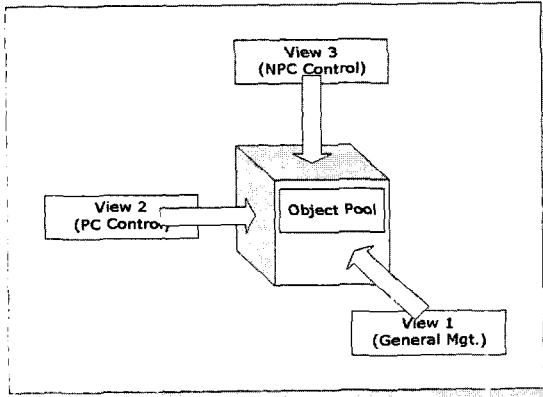
3. 2 시스템 관리(Management of Object Pool)

온라인 게임에서는 오브젝트 풀을 다음과 같은 세 가지 관점에서 관리하고 있다. [그림 2]에서 오브젝트 풀의 관리 모델을 설계한다.

- 1) View 1 : has World list for general management
- 2) View 2 : has PC list for PC Control

3) View 3 : has NPC list for NPC Control

이 세 가지 뷰는 오브젝트 풀에 대한 이용형태로 관리한다. 이 뷰로부터 오브젝트 풀의 인터페이스를 설계하도록 한다.



(그림 2 : 오브젝트 풀 관리 모델(Management Model of Object Pool))

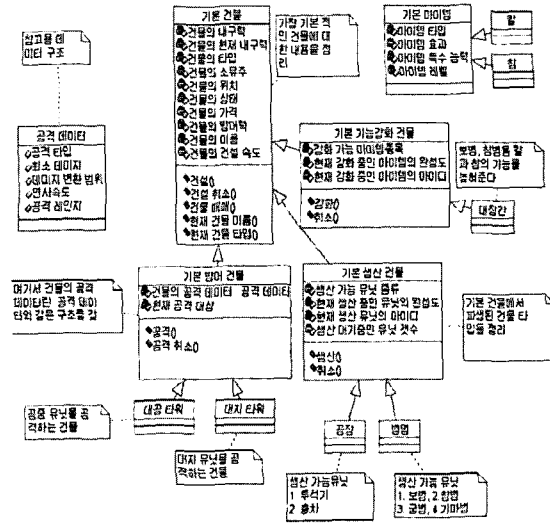
4. UML 게임 엔진 구조 설계(Structural Design of Game Engine) 사례

4. 1. MAP설계

객체지향 방법론을 일관되게 적용한 분석/디자인/개발을 위해서 UML을 사용하였다. 본 연구에서는 UML로 온라인 게임 서버/클라이언트를 설계한다. 복잡해지는 기획자의 의도 하에 프로그램으로 구현하기 위해서는 기획 단계 시에 기획자, 프로그래머, 그 외의 개발에 관계된 모든 이들이 이해하고 동의할 수 있는 방법으로 설계 과정을 조직화 하여야 한다. 이 때 UML(& GDL)을 사용하여 조직화를 할 수 있다.

UML(Unified Modeling Language)은 시스템 개발자가 자신의 비전(vision)을 구축하고 반영하는데 있어서 표준적이고 이해하기 쉬운 방법으로 할 수 있도록 도와주며, 자신의 설계 결과물을 다른 사람과 효과적으로 주고받으며 공유할 수 있는 메커니즘을 제공하는 하나의 언어(Language)이다. 이 것과 마찬가지로 게임 개발용으로 표준 언어 GDL을 개발하면 게임 제작 시 효율적이다.

UML을 이용한 게임 디자인 지원방안 UML notation을 이용하는데, UML의 각 다이어그램을 따라 설명한다.

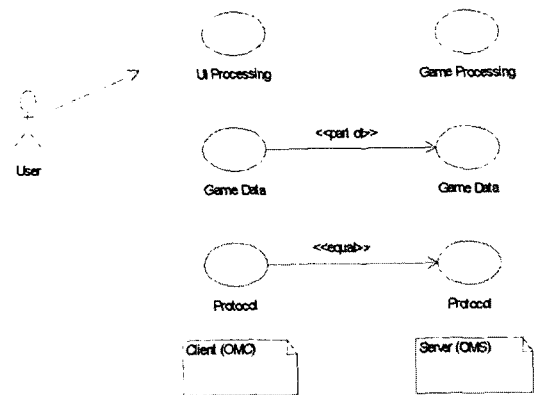


(그림 3 : 기본적인 RTS 예 - 건물 중심 MAP)

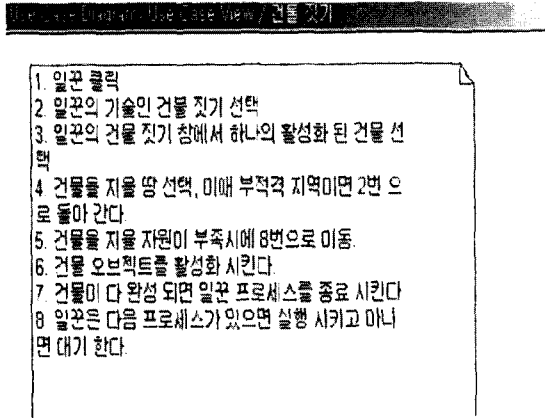
4. 2. Use Case Diagram

다음으로 서버와 클라이언트를 나누어서 생각한 (그림 4)이다. 시스템이 물리적, 논리적으로 서버와 클라이언트로 나누어지기 때문에, 개발자의 입장에서 서버와 클라이언트를 구분해야 한다. 또한 본 프로젝트에서는 클라이언트만 객체지향 방법론을 사용하기 때문이다.

사용자의 입장에서 보면 실제적인 게임 서비스는 User Interface라 할 수 있다. 이러한 UI 프로세싱은 클라이언트에서 이루어진다. 하지만, 실제 게임을 운영하는 게임 프로세싱은 서버에서 이루어지게 된다. 게임 운영에 필요한 게임 데이터 또한 서버에서 중앙집중식으로 관리하고 갱신하



(그림 4 : 서버/클라이언트로 나눈 Use Case Diagram)

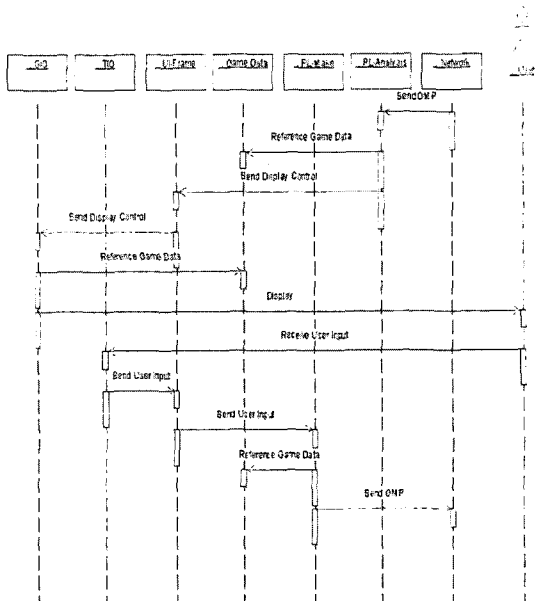


(그림 5 : 기본적인 RTS 예제 - 세부 기획)

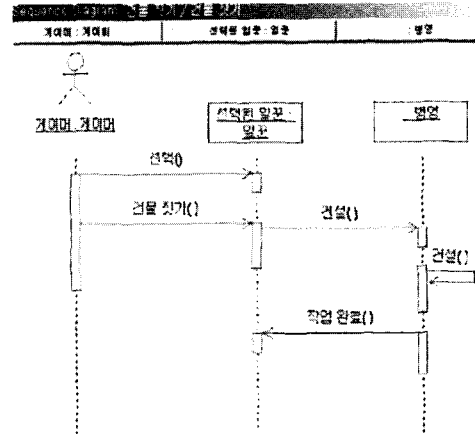
게 된다. 그렇지만 앞에서 고찰하였듯이 이 게임 데이터의 일부는 클라이언트에서 필요로 한다. 또한 서버와 클라이언트는 서로 약속된 프로토콜이 필요하다.

4. 3. Sequence Diagram

다음은 Sequence Diagram으로 각 기능 요소들의 협동 작업을 메시지에 초점을 맞추었다. 앞서 설명한 OpenMUD 클라이언트의 두 가지 실행 시나리오를 Sequence Diagram으로 기술하였다.



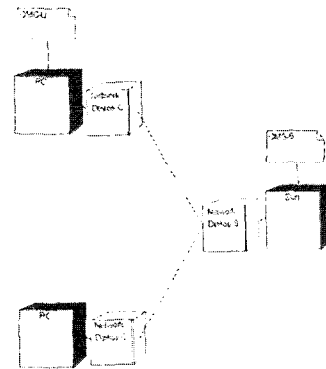
(그림 6: Sequence Diagram of Client System)



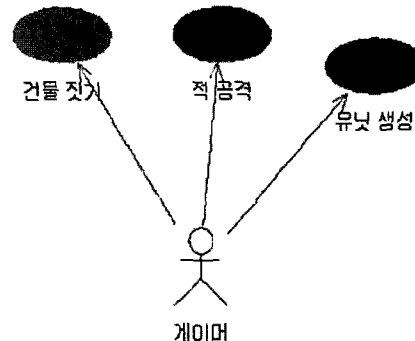
(그림 7 : 기본 RTS 예-시퀀스 다이어그램)

4. 4. Deployment Diagram

Deployment diagram은 시스템의 네트워크 구조를 단순화한 것이다. 앞에서 보인 네트워크 구조는 OMS-M, OMS-S, OMC-A, OMC-U를 모두 고려한데 반하여, (그림 8)에서는 일단 개발한 서버와 클라이언트만 고려한다. (파란색은 작업 중, 빨간색은 완성)



(그림 8 : Deployment Diagram of Client System)



(그림 9 : 기본 RTS 예-현재 기획을 시작점 표현)

4. 5. Class Diagram / Collaboration Diagram

클라이언트 시스템에 필요한 클래스를 설계한다. 우선 Class Diagram을 살펴보면, 기능 정의에서 설명했던 것처럼, 크게 네 가지로 분류되었다. 이 중 ADT 클래스는 “Game Data” 클래스와 “PL” 클래스이며, 나머지는 모두 객체 생성이 가능한 클래스들이다. 대부분의 클래스들은 앞의 설명으로 충분하며, 몇 가지 특이한 클래스들에 대해 보충설명 한다.

① UI Frame Class

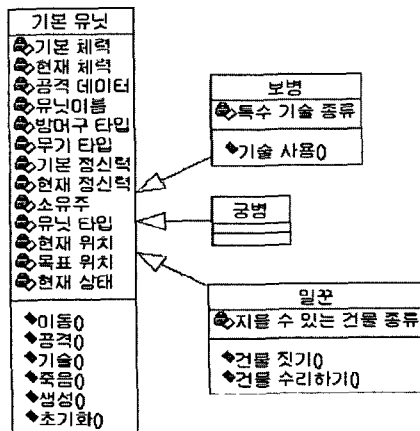
- GIO, TIO 객체들을 포함하게 된다.(Aggregation)
- GIO, TIO 객체들의 입력과 출력을 책임지는 PL-Analysis, PL-Make와 연결을 유지한다.
- User Interface의 변화가 생겨 새롭게 장르가 추가될 경우, 유연한 확장이 가능하다.

② PC, Map

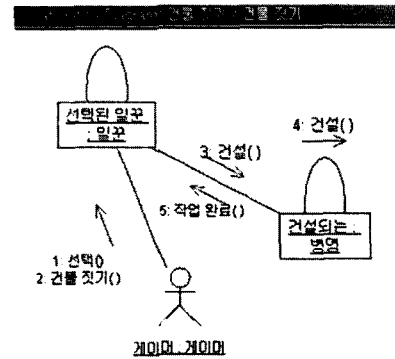
- 게임 데이터 중 PC나 Map 하나에 대한 정보를 유지하는 것이 아니다. 클라이언트가 유지하여야 하는 게임데이터 중 PC / Map에 대한 것을 총괄 관리한다.

③ Network

- 하드웨어적인 네트워크 디바이스가 아니라, TCP/IP 프로토콜을 의미한다. 물론 운영체제와 개발환경에 따라 약간씩 차이가 있으며, 윈도우의 경우 Winsock 이 이에 해당한다.



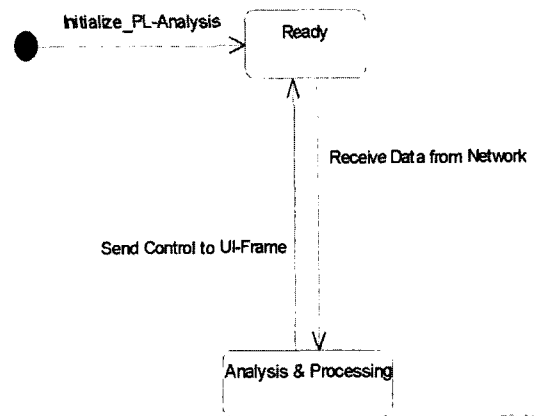
(그림 10 : 기본 RTS 예 - 유닛 중심)



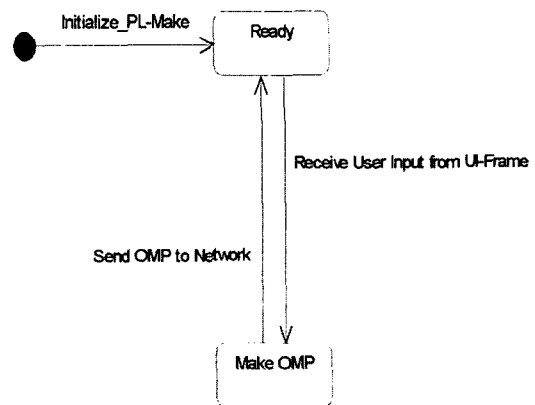
(그림 11 : 기본 RTS 예 - 협력 다이어그램)

4. 6. State Diagram

각 클래스들은 State Diagram을 가질 수 있다. 위에 설명한 클래스들 중, PL-Analysis, PL-Make, UI-Frame, Network 클래스에 대해 State Diagram을 보인다.



(그림 11: State Diagram of PL-Analysis)



(그림 12 : State Diagram of PL-Make)

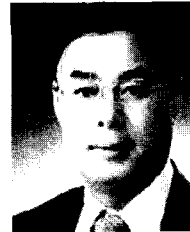
5. 결론 및 향후 연구 방향

기존 설계 방법으로 일반적인 프로그램을 개발 할 경우 품질 및 생산성에 문제가 많았으나 UML을 사용하여 객체 기반의 게임 소프트웨어를 개발하여 개발 프로젝트를 수행 하면 생산성이 증대된다. 게임프로젝트 설계와 형상관리 방법을 사용하여 프로세스 또는 결과물을 관리하고 있으며, 이렇게 설계 된 설계서는 정확한 기능 연계관계와 게임 개발자간의 관계를 가질 수 있다. 그러나 게임 개발 프로젝트를 수행하다 보면 프로세스 설계의 작업 수행 시에 필요한 입·출력물이 산출되므로 프로세스와 산출물을 동시에 관리할 수 있는 도구가 필요하다. 이에 따라 본 논문에서는 프로젝트 관리와 산출물 관리를 UML로 설계 할 수 있는 통합관리 도구를 연구하였다. 본 논문에서 제안한 UML을 사용함으로써 프로세스 모형화가 용이하고 설계 및 개발 프로세스 진행현황을 팀원 모두가 동시에 쉽게 파악할 수 있으며, 작업 개발자, 검토자, 승인자를 파악하기 쉽고, 시스템을 통한 프로세스 모델 수정이 용이하며 산출물의 표준화를 이룰 수 있다. 또한 산출물의 관리가 용이하며 산출물의 정보를 공유할 수 있고 산출물간의 추적이 용이하다. 향후 연구방향은 UML을 사용하여 프로세스 정형화를 위한 프로세스 참조 모델을 연구하고 이를 적용하여 개발 프로세스 관리를 수행함에 있어 생산성을 극대화 할 수 있으므로 자동화 및 품질의 균형을 이룰 수 있는 게임개발 CBD 도구를 연구하도록 한다.

<참고 문헌>

- [1] "2002 대한민국 게임백서", 한국게임산업개발원, 2003.4.24
- [2] 김정민, "TTA SW 품질시험인증서비스 현황", TTA저널 제 84호, 2002.11
- [3] "ISO/IEC 9126 Software Engineering- Product Quality", ISO
- [4] "ISO/IEC 14598 Information technology -- Software product evaluation", ISO
- [5] "The Test discipline", Rational Corporation, Rational Unified Process.
- [6] "게임엔진 품질보증기술 연구및 개발", 한국게임산업개발원, 연구보고서02-004, 2002
- [7] 박준영 외, "ISO/IEC 9126 품질모델 기반의 게임엔진 품질특성 도출에 관한 연구", 2002년 하계 한국게임학회 총회 및 학술발표대회, 2002.7
- [8] "PC게임SW 품질평가시스템개발", 한국게임산업개발원, 과제수행 최종보고서, 2002.11
- [9] 최 성, 2003년1월 동계게임학술대회 발표논문, 효율적인 네트워크 게임시스템 구축을 위한 알고리즘 연구 (Study on the algorism of Implementation for Network game system)
- [10] KT-CBD 방법론, KT 서비스개발연구소, 2002.12
- [11] Microsoft, CBD 도입의 비용과 개발기간의 상관 분석, 2002.5
- [12] 정보통신부, "S/W컴포넌트 기술개발계획", 1999.7.
- [13] 최성, "Together로 UML 배우기" 1권, 으뜸정보기술(주)간, 2002.6.,
- [14] 최성, "Together로 객체지향 프로그램 배우기" 2권, 으뜸정보기술(주)간, 2002.9.
- [15] 조진희, 하수정, 김진삼, 박창순, "컴포넌트 기반 시스템 개발 방법론 마르미-III", 프로젝트관리기술논문집, 한국프로젝트관리 기술협회, 제 4권 제 4호, 2001. 9
- [16] 권오천, 이우진, 천윤식, 신규상, "CBD 지원 도구의 설계 및 프로토타이핑", 한국정보과학회 정보과학회지, 제 19권 2호, 2001. 2
- [17] 김경주, 조남규, "UML Components: 컴포넌트 기반 소프트웨어 명세를 위한 실용적인 프로세스", 인터비전, 2001
- [18] Butlergroup, "Component-Based Development", Butlergroup Management Guides, Sept., 1998.
- [19] Keith Short, "Component-Based Development and Object Modeling", Sterling Software, Feb., 1997.
- [20] Alan, W. Brown and Balbir Barn, "Enterprise-Scale CBD Building Complex Computer System From Components" Sterling Software, 1999
- [21] Desmond F. D' Souza, Alan C. Wills; "Objects, Components, And FrameWorks with UML- the Catalysis Approach", Addison-Wesley, Reading, Mass, 1998.

- [22] Paul Allen, "Realizing e-Business with Components", Addison-Wesley, 2000
- [23] Alan W. Brown, "large-Scale Component Based Development", Prentice Hall, 2000
- [24] George T. Heineman & William T Council, "Component-Based Software Engineering", Addison-Wesley, 2001
- [25] Peter Herzum & Oliver Sims, "Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise", John Willey & Sons, Inc, 1999
- [26] John Cheesman & John Daniels, "UML Component : A simple Process for specifying Component-Based software", Addison-Wesley, 2000
- [27] Rosenblum, D.S. and Natarajan, R., "Supporting architectural concerns in component-interoperability standards", IEEE Proceedings- Software, Volume: 147 Issue: 6, pp.215 223, Dec. 2000
- [28] Rational Software Corporation, Rational Unified Process, <http://www.rational.com/products/rup/index.jsp>
- [29] Computer Associates, CBD96, <http://www.sterling.com/cbdedge1/cbd96.htm>
- [30] Peter Herzum and Oliver Sims, "Business Component Factory: A Comprehensive Overview of Component-Based Development for the Enterprise", John Willey & Sons, Inc, 1999
- [31] Katharine Whitehead, "Component-Based Development: Principles and Planning for Business Systems", Addison-Wesley, 2002
- [32] Ralph Keller, Urs Holzle, "Implementing Binary Component Adaptation for Java", www.cs.ucsb.edu/oocsb
- [33] Oh-Cheon Kwon, Yoo-Hee Choi and Gyu-Sang Shin, "Technique for Adapting EJB Components according to Adaptation Patterns", ISE2001, Las Vegas, USA, 2001
- [34] Rosenblum, D.S. and Natarajan, R., "Supporting architectural concerns in component-interoperability standards", IEEE Proceedings-Software, Volume:147



최성

강원대학교 대학원 컴퓨터과학과 가상현실(영상기술)전공(박사)
 연세대학교 산업대학원 전자계산학과(석사)
 동국대학교 산업시스템학과(학사)
 (전) 한국생산성본부 국장, 제주은행전산실장
 (현재) 남서울대학교 컴퓨터학과 교수
 관심분야: EC/ERP, 소프트웨어공학, VR영상게임
 저서: 비즈니스리엔지니어링, ERP시스템기초, 게임PD가 되는 길, 게임수업서 등 33권
 E-mail: sstar@nsu.ac.kr