

# 게임소프트웨어 특성을 표현하기 위한 클래스 추출과 연관관계에 관한 연구

김 용 식<sup>†</sup> · 조 현 훈<sup>\*\*</sup> · 류 성 열<sup>\*\*\*</sup>

## 요 약

현재 국내 게임시장은 급속도로 발전하고 있으나 게임제작 프로세스는 체계화 되지 못하고 있다. 또한 게임기획자와 게임개발자 간의 의사소통의 문제점은 게임의 실패를 초래 하고 있다. 본 연구는 게임기획자와 게임개발자 간의 원활한 의사소통을 위하여 게임기획을 위한 산출물에서 게임의 특성을 표현하는 요소를 추출하여 클래스화 하고, 추출된 요소간의 관계를 나타내기 위하여 클래스의 연관관계를 제시한다. 사례 연구를 통해 추출된 클래스와 클래스간의 연관관계를 파악하여 체계적인 게임기획을 지원한다.

키워드 : UML, 게임, 개념설계, 클래스, 관계

## A Study on Sampling and Association Relation of Class to Express Game Software Characteristics

Yong-sic Kim<sup>†</sup> · Hyun-Hoon Cho<sup>\*\*</sup> · Sung-Yul Rhew<sup>\*\*\*</sup>

## ABSTRACT

Currently domestic game market rapidity is developmental but the game production process does not become systematization. Also it is bringing the failure of game that problem point of conversation between game planner and game developer.. The research which to conversation between game planner and game developer it sees does extracting a game element to expressed game characteristics from the product for game planning and it change extracted element into class and In order to express the relationship of element for it presents a relationship of class. Instance research it leads, It grasps a relationship among extracted classes and it supports systematic game planning.

Key Words : UML, Game, Concept Design, Class, Relation

### 1. 서 론

게임소프트웨어는 다른 소프트웨어와는 다르게 기획자와 개발자간의 작업영역이 구분되어 있으며, 불특정 다수에게 제공되는 소프트웨어이기 때문에 요구사항이 수시로 변동하게 된다. 게임제작에서 발생하는 문제점은 게임기획자와 게임개발자간의 의사소통이 원활하지 못하며, 게임기획에서 게임기획자의 의도를 정확하게 명시할 수 있는 매개체가 없다. 현재 게임을 기획하는 단계에서 게임기획자의 의도는 단순한 텍스트이거나 핸드 스케치 형식의 규칙이 없는 문서이기 때문에 보는 사람마다 다르게 해석할 수 있다. 이러한 문제점의 가장 이상적인 해결책은 게임기획자와 게임개발자

역할을 동시에 수행할 수 있는 인력의 존재이다. 그리하여 기획의 의도를 정확하게 게임에 반영할 수 있도록 개발하는 것이다. 하지만 게임개발 업체의 실정은 기획자와 개발자의 역할을 동시에 수행할 수 있는 인력이 부족하다.

본 연구에서는 게임기획자와 게임개발자간의 의사소통을 원활하게 수행할 수 있는 매개체인 UML을 이용하여 게임 요소를 표현할 수 있는 클래스를 추출하고 추출된 클래스간의 연관관계를 연구하고자 한다. 게임도 일반 정보시스템과 같이 객체지향적인 개발을 나타내고 있는데 일반적인 정보시스템의 기획과 같이 무엇(What)을 만드는 것이 중요한 것이 아니라 어떻게(How) 만드는 것이 더욱 중요시 되고 있다. 이러한 면에서 게임개발자와 게임기획자의 의사소통은 중요시 되기 때문에 본 연구를 통하여 추출된 게임의 특성을 표현하는 클래스와 클래스들간의 연관관계를 통해 좀더 원활한 의사소통을 통하여 기획된 게임이 제작되는데 도움이 될 것이다.

<sup>†</sup> 준 회 원 : 숭실대학교 일반대학원 컴퓨터학과 석사과정

<sup>\*\*</sup> 정 회 원 : 한국문화콘텐츠진흥원 CT전략센터 과장

<sup>\*\*\*</sup> 종신회원 : 숭실대학교 컴퓨터학부 교수

논문접수 : 2005년 10월 6일, 심사완료 : 2006년 3월 31일

## 2. 관련 연구

### 2.1 UML

UML(Unified Modeling Language)은 소프트웨어 시스템을 모델링 하기 위한 언어로 개발되었다. UML을 사용함으로써 소프트웨어 개발자들은 시스템에 대한 모델을 만들 수 있게 되었고, 이에 따라 소프트웨어 개발과 관련된 산출물들을 가시화하고 명세할 수 있게 되었다. 또한 UML은 모델링 수준에서 멈추지 않고 소프트웨어 설계와 구현에 관련된 사항들을 언어에 반영하여 모델을 설계와 구현에 사용할 수 있도록 하였다[9].

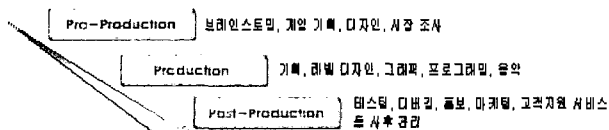
UML의 기능에는 의사소통이 있다. 예컨대 프로그래머가 개념모델을 다른 사람에게 전달하는데 여러 가지 오류가 나타날 수 있다. 그것은 참여한 모든 사람이 같은 언어를 사용하지 않기 때문이다. 일반적으로 프로젝트나 조직에서는 자신들만의 언어를 만들기는 하지만, 외부인 이거나 새로운 사람일 경우에는 무슨 일이 진행 중인지 이해하기 어렵다. 이러한 점에서 UML은 개념모델을 다른 사람에게 전달할 수 있는 공통적인 언어가 된다 [5, 6].

### 2.2 게임의 특성

게임 소프트웨어는 일반 소프트웨어와 달리 콘텐츠를 기반으로 하는 소프트웨어로 (그림 1)과 같이 3단계의 제작 단계를 통해 개발된다.

첫 번째 단계인 “Pre-Production” 단계는 개발하고자 하는 게임의 소재와 시장 조사 등의 단계를 통해 게임 소프트웨어 개발의 가능성을 검토하는 단계이다. 두 번째 단계인 “Production” 단계는 게임 소프트웨어를 개발하기 위하여 완성된 형태의 기획서와 게임을 구성할 콘텐츠 요소인 그래픽, 프로그래밍, 음악 등을 구현하는 단계이다. 마지막으로 세 번째 단계인 “Post-Production” 단계는 개발된 게임 소프트웨어를 서비스하기 위한 테스트와 디버깅, 홍보, 마케팅 등의 활동을 하는 단계이다[1].

게임 소프트웨어는 게임의 플레이 형태, 배경 사상, 재미 요소 특징을 통해 12가지의 장르로 구별을 할 수 있다. 이러한 장르적 특성은 게임 소프트웨어 개발 프로세스에 있어 상당한 영향을 줄 수 있다. 예를 들어 스토리보드나 원화 제작 같은 개발단계에서 게임 장르가 액션일 때와 시뮬레이션일때 게임의 진행 방법이나 조작, 배경이나 등장 요소의 제작 과정 등이 달라지게 된다[2].



(그림 1) 게임 소프트웨어 제작 단계

### 2.3 게임기획서 표준양식

게임기획서표준양식이란 게임기획자가 게임을 개발하는

과정에서 생각해낸 디자인 내용을 체계적으로 정리하고, 이를 각 파트 별 작업자들에게 올바르게 전달할 수 있도록 하기 위해서 만든 게임기획산출물 양식을 말한다. 이 양식은 게임 개발과정 중 발생하는 게임기획 업무의 체계를 정립하고, 게임 디자인 과정에서 나타날 수 있는 오류 또는 실수를 미연에 방지하며, 작업자들이 문서를 통해서 기획내용을 정확히 이해하고 그에 맞게 작업하여 결과물을 생산할 수 있도록 하기 위해 사용된다.

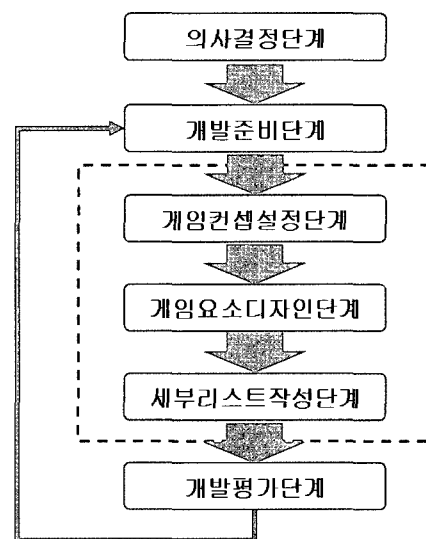
게임기획서표준양식은 게임기획 내용의 정리를 통해 게임 기획과정을 숙지하고 체계적인 게임 개발과정을 제공하며, 게임 개발인력 교육자료로 활용하는데 그 목적이 있다. 이러한 게임기획서표준양식은 특정 장르에 국한되지 않는 범용성을 가지고 있으며, 체계적인 업무단계를 제공하여 각 항목별 세부적인 상황설정 및 설명을 나타내고 있다.

## 3. 게임의 특성과 Class추출

### 3.1 게임기획프로세스와 산출물

게임기획을 위한 모델링에서 표현되는 요소를 장르에 국한되지 않는 요소로 추출하는데, 이는 게임기획서표준양식을 기반으로 요소를 추출하였다. 게임기획서표준양식은 총 6단계를 수행하고 4번의 반복이 이루어 진다. (그림 2)에서 나타내는 것은 각 단계별 순서와 반복되는 단계를 표현 한 것이다.

위에서 표현되는 각 단계에서 작성되는 게임기획산출물은 33개 이다. 각 게임기획산출물은 게임을 제작하기 위한 문서이므로 게임 내에 표현되는 요소를 상세하게 설명한다. 이러한 게임기획산출물은 게임을 제작하는 과정에서 게임기획자와 게임개발자간의 의사소통 수단이 될 수 있다. 즉, 소프트웨어 프로세스에서 설계자와 개발자간의 의사소통 수단이 산출물인 것과 같이 게임기획산출물은 의사소통이 될 수 있다. 그리하여 각 게임요소는 게임기획산출물에서 keyword



(그림 2) 게임기획산출물 작성 프로세스

중심으로 추출하게 되며, 추출된 요소는 게임을 표현하는 요소가 된다. 또한 추출된 요소는 의사소통의 핵심 항목으로 표현될 수 있다.

3.2 게임기획산출물에서 핵심어 추출

다음의 <표 1>은 게임기획서표준양식에서 각 단계별 산출물에 나타나는 요소를 keyword 방식으로 표현한 것이다.

<표 1> 게임기획산출물 리스트

단 계	게임기획문서	추출 요소(keyword)
의사결정 단계	아이디어제안서	아이디어
	브레인스토밍회의록	브레인스토밍회의
	시장조사보고서	시장조사, 경쟁제품, 용자설문
	프로젝트제안서	컨셉
	기초컨셉설정문서	게임디자인, 그래픽아트, 게임사운드, 게임프로그래밍
제품소개문서	-	
개발준비 단계	버전계획수립문서	버전
게임컨셉 설정단계	주요컨셉설정문서	컨셉
	스토리설정문서	주제, 스토리, 등장인물, 용어
	세계관설정문서	공간적배경, 시대적배경, 문화적배경, 경제상황, 존재사물
	게임진행설정문서	게임진행도
	개체컨셉설정문서	개체
배경컨셉설정문서	배경	
게임요소 디자인 단계	게임화면디자인문서	게임화면
	시스템디자인문서	시스템
	시스템공식디자인문서	시스템공식
	조작인터페이스 디자인문서	조작인터페이스
	GUI디자인문서	인터페이스
	개체디자인문서	개체, 개체규칙
	지역디자인문서	지역, 배경음악
	이벤트디자인문서	이벤트, 등장개체, 배경음악, 시나리오
	사운드디자인문서	배경음악, 효과음
	세부리스트 작성 단계	게임화면리스트문서
시스템리스트문서		시스템
시스템공식리스트문서		시스템공식
조작인터페이스 리스트문서		조작인터페이스
GUI리스트문서		인터페이스
개체리스트문서		캐릭터, 아이템
효과리스트문서		효과
지역리스트문서		지역
이벤트리스트문서		이벤트
사운드리스트문서		배경음악, 효과음
개발평가 단계	개발평가서	평가항목

위의 표와 같이 게임기획서표준양식에서 단계별 문서의 내용에서 keyword 방식으로 게임요소를 추출하였다. 추출된 요소는 문서에 나타나는 단어를 중심으로 표현한 것이기 때문에 게임에 표면적으로 나타나는 요소가 아닌 것도 있다. 게임요소는 게임제작을 위해 표면적이거나 내부적으로 표현될 수 있는 요소로 나타난다.

3.3 게임요소 Class 표현

3.3.1 게임요소의 정제

게임기획산출물에서 추출된 요소들 중에 게임제작을 위한 요소들만을 정제하도록 한다. 정제된 요소는 다시 필수요소와 부가요소로 구분될 수 있다. 필수요소는 어떠한 장르의 게임에서든지 나타나야 하는 요소이며, 부가요소는 게임의 장르에 따라 표현되거나 표현되지 않는 요소이다. 그리하여 구분을 통해 필수요소와 부가요소로 표현하였다.

<표 2> keyword 추출

추출 요소(keyword)	게임표현요소	구분
아이디어, 브레인스토밍회의, 시장조사, 경쟁제품, 사용자설문, 컨셉, 게임디자인, 그래픽아트, 게임사운드, 게임프로그래밍, 버전, 주제, 스토리, 용어, 경제상황, 게임진행도, 평가항목	해당사항없음	해당사항없음
등장인물, 개체, 개체, 개체규칙, 등장개체, 캐릭터	캐릭터	부가
존재사물, 개체, 등장개체, 아이템	아이템	부가
공간적배경, 시대적배경, 문화적배경, 배경, 지역	지역	부가
게임화면	게임화면	필수
시스템	시스템	필수
시스템공식	시스템공식	필수
조작인터페이스	조작인터페이스	필수
인터페이스	인터페이스	필수
배경음악, 배경음악, 배경음악	배경음	부가
효과음	효과음	부가
효과	효과	부가
이벤트	이벤트	부가

3.3.2 게임요소의 Class 정의

게임기획산출물에서 추출되어 정제된 게임요소는 게임에 표현될 수 있는 객체로 인식할 수 있다. 이러한 게임요소의 특징을 이용하여 UML의 Class로 표현한다. 이때 각 요소는 속성과 행위를 가지는 메타클래스의 상속을 받아 스테레오 타입으로 표현된다.

게임기획산출물에서 추출된 게임요소는 UML의 표기법을 이용하여 메타클래스를 상속받아 클래스의 스테레오 타입으로 표현하고, 각 클래스간의 관계를 나타내도록 한다. 이렇게 정의되는 게임요소는 게임기획자의 입장에서 공통된 게임요소로 정의할 수 있으며 속성과 행위를 정의하므로 의사소통의 매개체가 될 수 있다. 즉, 어떠한 요소를 기반으로

클래스가 도출되었는지 알 수 있으며, 각각의 속성과 행위에 대한 내용을 알아볼 수 있기 때문이다. 또한, 텍스트만으로 부족했던 내용을 좀더 자유롭게 표현할 수 있다. 또한 게임개발자의 입장에서 UML로 표현된 게임요소를 쉽게 파악할 수 있으며, 정의된 속성을 통해 게임요소를 구체화하도록 지원 할 것이다.

먼저 게임기획산출물을 통해 추출된 게임요소에서 핵심요소를 클래스로 표현하면 다음과 같다. 부가요소는 부록 1에 표현하였다.

〈표 3〉 메타클래스를 스테레오타입으로 정의(필수요소)

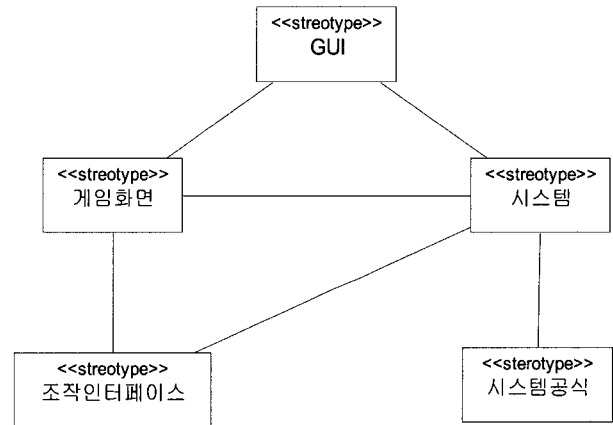
요 소	스테레오타입	프로파일
게임 화면	<pre> classDiagram     class GameView["&lt;&lt;metaclass&gt;&gt; Class"]     class GameViewSub["&lt;&lt;stereotype&gt;&gt; 게임화면"]     GameViewSub -- &gt; GameView                     </pre>	<ul style="list-style-type: none"> <li>- Description 게임이 진행되는 주요 화면의 기본적인 형태 및 구성을 디자인 하는 것을 말한다.</li> <li>- Attribute · 화면명(ViewName) · 화면번호(ViewNum) · 화면시점(ViewPoint) · 화면용도(ViewUse)</li> </ul>
시스템	<pre> classDiagram     class System["&lt;&lt;metaclass&gt;&gt; Class"]     class SystemSub["&lt;&lt;stereotype&gt;&gt; 시스템"]     SystemSub -- &gt; System                     </pre>	<ul style="list-style-type: none"> <li>- Description 게임이 갖고 있는 기본 원리에서부터 그 게임만의 특징적인 재미요소에 이르기 까지, 해당 게임을 이루고 있는 전반적인 내부 규칙들을 모두 말한다.</li> <li>- Attribute · 시스템명(SystemName) · 시스템번호(SystemNum) · 시스템내용(SystemDesc)</li> </ul>
시스템 공식	<pre> classDiagram     class SystemFormula["&lt;&lt;metaclass&gt;&gt; Class"]     class SystemFormulaSub["&lt;&lt;stereotype&gt;&gt; 시스템공식"]     SystemFormulaSub -- &gt; SystemFormula                     </pre>	<ul style="list-style-type: none"> <li>- Description 게임 내의 여러 가지 개체 혹은 요소들이 서로 맞물려 움직이는 규칙의 여러 가지 단순하거나 복잡한 모든 공식들을 말한다.</li> <li>- Attribute · 공식명(FomulaName) · 공식번호(FomulaNum) · 공식내용(FomulaDesc)</li> </ul>
조작 인터 페이스	<pre> classDiagram     class EquipInterface["&lt;&lt;metaclass&gt;&gt; Class"]     class EquipInterfaceSub["&lt;&lt;stereotype&gt;&gt; 조작인터페이스"]     EquipInterfaceSub -- &gt; EquipInterface                     </pre>	<ul style="list-style-type: none"> <li>- Description 게임을 조작하는 장치의 구성 및 조작 방법을 표현하는 클래스이다.</li> <li>- Attribute · 장치명(EquipName) · 장치번호(EquipNum) · 장치용도(EquipUse) · 장치특성(EquipChar)</li> </ul>
GUI	<pre> classDiagram     class GUI["&lt;&lt;metaclass&gt;&gt; Class"]     class GUISub["&lt;&lt;stereotype&gt;&gt; GUI"]     GUISub -- &gt; GUI                     </pre>	<ul style="list-style-type: none"> <li>- Description 게임 내용 중 화면에 표현되는 'GUI (Graphic User Interface)'의 구성 및 조작방법을 나타내는 클래스이다.</li> <li>- Attribute · 인터페이스페이지명(GUIName) · 인터페이스페이지번호(GUINum) · 인터페이스세부설명(GUIDesc)</li> </ul>

3.4 게임요소 Class간의 연관관계

추상개념을 만들어보면 혼자서 존재하는 클래스는 없다. 대부분 다양한 다른 사물과 교류한다. 게임기획산출물에서 추출된 12개의 게임요소의 클래스도 혼자서 존재하는 것은 없다. 그리하여 게임을 표현하는 클래스가 어떻게 서로 교류하면서 존재하는가를 나타내도록 한다.

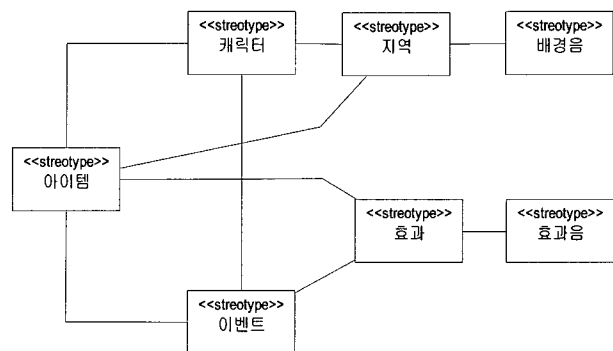
객체지향 모델링에서 특히 중요한 세가지 관계가 있다. 첫 번째는 의존(dependency)이 있다. 이것은 클래스간 사용 관계를 표현하는데 정제, 추적, 결합 관계 등을 포함한다. 두 번째는 일반화(generalization)가 있다. 일반화는 일반화된 클래스를 특수화된 클래스와 연결해 준다. 세 번째로 연관(association)은 객체간 구조 관계를 표현한다[5]. 이러한 관계들 중에 게임요소는 연관관계로 클래스의 관계를 설정하도록 한다.

위에서 정의한 게임에 표현되는 클래스의 관계는 게임기획서표준양식[4]에서 관계를 갖게 되는 요소들간의 관계를 기반으로 표현하였다. 게임에 표현되는 클래스는 모두 구조적 관계가 나타난다. 이러한 구조적 관계인 연관관계를 이용하여 위에서 추출된 게임표현요소간의 관계를 그림으로 표현하면 다음과 같다.



(그림 3) 필수요소 연관관계

아래의 연관관계는 부록1에서 나타낸 부가요소간의 연관 관계이다.



(그림 4) 부가요소 연관관계

### 4. 사례 연구

#### 4.1 게임개요

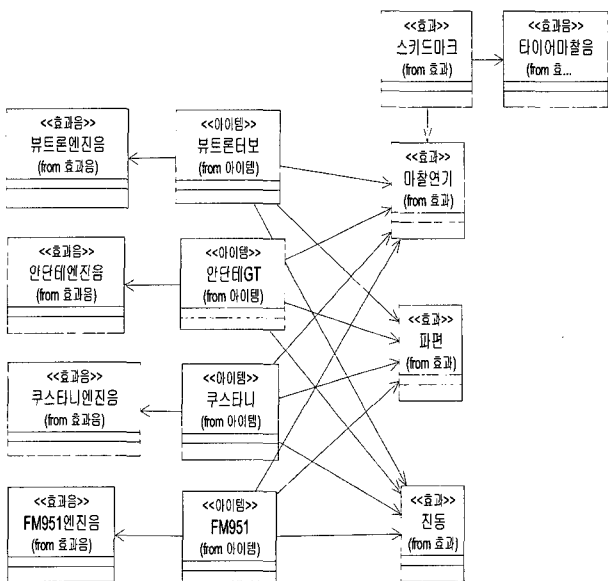
게임명	레이싱 스피리츠[부록2, 3 참조]
게임장르	온라인 레이싱 게임
게임컨셉	일반 도로에서 일반 차량으로 다른 사람들과 레이스를 벌이는 온라인 레이싱 게임으로 게임이 현실적인 레이싱 게임 분위기와 잘 매치되므로, 비현실적인 요소(부스터, 비현실적인 가속)는 배제한 게임이다.

#### 4.2 게임표현 Class

게임기획 모델링은 게임을 나타내는 요소 중에서 표현하기 위한 요소를 중심으로 여러 표현모델이 나타날 수 있다. 캐릭터를 중심으로 표현되는 모델과 아이템을 중심으로 표현되는 모델, 지역을 중심으로 하는 모델 등 여러 가지의 형태로 게임을 표현하는 모델이 나타날 수 있다. 다음은 부록3에서 제시한 게임요소에 대한 리스트에 대하여 관계를 제시한 것이다.

##### 4.2.1 아이템 중심의 Class관계

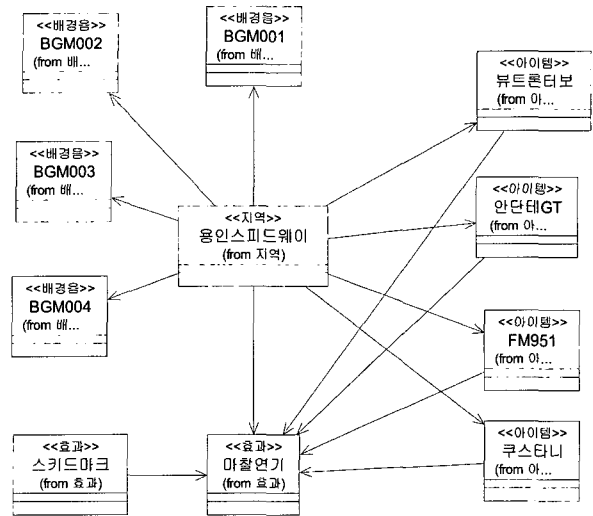
위에서 제시한 게임기획요소에서 아이템중심으로 요소간의 관계를 나타내는 게임기획 모델링을 표현하고자 한다. 이때 아이템은 게임에서 사용자가 직접 제어하는 자동차를 중심으로 나타낸다.



(그림 5) 아이템중심의 모델링

##### 4.2.2 지역 중심의 Class 관계

지역중심의 모델링은 지역에 나타날 수 있는 요소들간의 관계를 표현하여 모델링 하였다.



(그림 6) 지역중심의 모델링

### 5. 결론

체계적이지 못한 게임기획 단계에서 게임기획자와 게임개발자의 원활하지 못한 의사소통을 해결하기 위해 게임에 나타나는 요소를 추출하여 UML의 클래스 표기법을 통하여 클래스를 정의하고 정의된 클래스간의 연관관계를 나타내어 게임기획자와 게임개발자간의 원활한 의사소통을 지원하도록 하였다.

정의된 게임을 표현하는 요소는 12개로 이루어져 있으며, 각 요소는 UML의 클래스로 정의하여 속성과 관계를 정의하였다. 12개의 요소는 필수요소와 부가요소로 나뉘게 된다. 필수요소 5개와 부가요소7개로 구분되며, 필수요소는 게임의 장르에 무관하게 게임에 표현되는 요소이며, 부가요소는 게임의 장르에 따라 표현되거나 표현되지 않는 요소이다.

이렇게 표현된 게임요소는 게임기획자의 아이디어 및 의도를 표현할 수 있으며, 작성된 UML을 통해 개발자가 쉽게 인식할 수 있는 매개체가 될 수 있다. 그리하여 현재 다수의 게임개발 업체가 해결하지 못한 게임기획자와 게임개발자간의 원활하지 못한 의사소통을 해결할 것이다.

본 논문에서 표현된 요소는 어떠한 장르에 편중되지 않고 일반적인 게임에 대한 요소이다. 그리하여 추후 연구방향으로 장르에 따른 게임표현요소의 상세정의와 클래스 및 관계를 정의하여 좀더 세부적인 게임제작을 지원하도록 한다.

### 참고 문헌

- [1] 게임 제작 프로세스에 관한 연구, 한국게임산업개발원, 2003.
- [2] 게임 제작 프로세스 개선안에 관한 연구, 한국게임산업개발원, 2005.
- [3] 2004 대한민국 게임백서, 문화관광부, 한국 게임산업개발원, 2004.

- [4] 게임 기획서 표준양식 개선안에 관한 연구, 한국게임산업개발원, 2005.
- [5] Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide," Addison-Wesley, 1999.
- [6] The Object Management Group(OMG), "Unified Modeling Language," Version 1.3, OMG, June, 1999.
- [7] Martin Fowler, Kendall Scott, UML Distilled, Addison-Wesley, 1997.
- [8] Pressman, R., S., Software Engineering a Practitioner's Approach 4th Edition, McGraw-Hill, 1999.
- [9] Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language Reference Manual, Second Edition", AddisonWesley, 2004.



**김 용 식**

e-mail : Free77@nexon.co.kr  
 2004년 송실대학교 컴퓨터학과(학사)  
 2006년 송실대학교 컴퓨터학과(석사)  
 2006년 5월~현재 넥슨 포털개발실  
 포털기획팀  
 관심분야: 소프트웨어공학, UML, 게임  
 개발, 형상관리, QA, 프로세스



**조 현 훈**

e-mail : casper69@kocca.or.kr  
 1995년 광주대학교 전자계산학과(공학사)  
 1997년 송실대학교 컴퓨터학과(공학석사)  
 2003년 송실대학교 컴퓨터학과(공학박사)  
 1999년 2월~2004년 8월 신홍대학  
 컴퓨터정보계열 겸임교수  
 2001년 9월~2004년 7월 (주)크래비스  
 방법론연구팀장  
 2004년 7월~2005년 1월 (재)한국게임산업개발원 게임연구소  
 2005년 2월~현재 한국문화콘텐츠진흥원 전략기획본부  
 콘텐츠유통팀  
 관심분야: 소프트웨어공학, 객체지향방법론, 컴포넌트기반  
 개발방법론, 게임기획 및 제작, 문화산업기술(Culture  
 Technology) 및 콘텐츠유통기술 등



**류 성 열**

e-mail : syrheew@comp.ssu.ac.kr  
 1997년 2월 아주대학교  
 컴퓨터학부(공학박사)  
 1997년 3월~1998년 3월 George Mason  
 University 교환교수  
 1981년 3월~현재 송실대학교  
 정보과학대학 컴퓨터학부 교수  
 관심분야: 소프트웨어 유지보수/제사용, 소프트웨어  
 재공학/역공학, 정보보호 등