

UML과 Petri Net을 이용한 게임 PVP 시스템 모델링 및 비교분석

Modeling and Comparative analysis of Game PVP System using UML and Petri Net

오기성¹⁾, 오석희²⁾

Gi-Sung Oh¹, Seok-Hee Oh²

요약

게임시장이 지속적으로 성장하면서 게임의 규모와 복잡도가 증가하고 있다. 이에 게임 제작의 어려움이 늘어나고 경쟁사가 많아지면서 게임의 빠른 개발과 출시가 중요한 요인이 되었다. 이로 인해 게임 기획 과정에서 시스템의 논리적인 흐름을 시각적으로 확인해야하며, 무결성을 검증하고 정확하게 설계할 필요가 있다. 본 논문에서는 게임시스템 기획과정 시 모델링 도구로 UML과 Petri Net을 제안하며, 기존 상용게임을 분석하여 PVP 시스템 모델을 구현하고 효과성을 검증하였다.

핵심어: 게임, UML, Petri Net, 모델링, 게임 시뮬레이션, 게임 시스템 디자인, 분석

Abstract

As the gaming market continues to grow and increase in size and complexity of the game. As it make difficult game production and increases number of competitors, rapid development and launch of games became important factors. Therefore, it is necessary to visually check the logical flow of the system in the game design process, and to verify the integrity and design it accurately. In this paper, we propose UML and Petri Net as modeling tools for the game system design process, and analyze the existing commercial games to implement the PVP system model and verify its effectiveness.

Keyword: Game, UML, Petri Net, Modeling, Game Simulation, Game System Design, Analysis

1) Department of Game Engineering, Gachon Univ., 1342 SeongnamDaero, Sujeong-Gu, Seongnam-Si, Gyeonggi-Do, Korea, 13120 e-mail: eustia@gc.gachon.ac.kr (First author)

2) Department of Game Engineering, Gachon Univ., 1342 SeongnamDaero, Sujeong-Gu, Seongnam-Si, Gyeonggi-Do, Korea, 13120 e-mail: seokhee5@gachon.ac.kr (Corresponding author)

* 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음" (IITP-2018-2017-0-01630)

Received(May 16. 2018), Review (June 05. 2018), Accepted(June 30. 2018)

1. 서론

게임시장이 점차 큰 폭으로 성장하고 하드웨어의 발전과 소프트웨어 개발력이 향상됨에 따라 많은 게임을 접한 소비자들의 취향도 다양해졌다. 흥미를 배가시키기 위해 게임은 그 규모가 커지고 복잡한 규칙과 요소들이 어우러지면서 그만큼 제작과정에서 어려움이 증가하고 있다[1]. 그럼에도 불구하고 포화된 게임 시장으로 경쟁자가 늘어나 점차 짧아지는 게임수명으로 인해 경쟁사 보다 빠른 개발과 출시가 필요한 것이 현실이다.

따라서 게임 디자인의 영역에서 시스템 기획은 어느 때보다 더 빠르고 정확한 설계를 요구받고 있다. 게임의 규모 및 구성이 크고 복잡해짐에 따라 설계한 시스템에 오류가 있을 경우, 개발과정에서 논리적으로 어느 부위에서 문제가 일어났는지 찾기가 쉽지 않을 뿐 아니라 이 과정에서 많은 시간과 비용이 소요되므로 가시적으로 시스템을 설계하고 검증할 수 있는 도구가 필요하다.

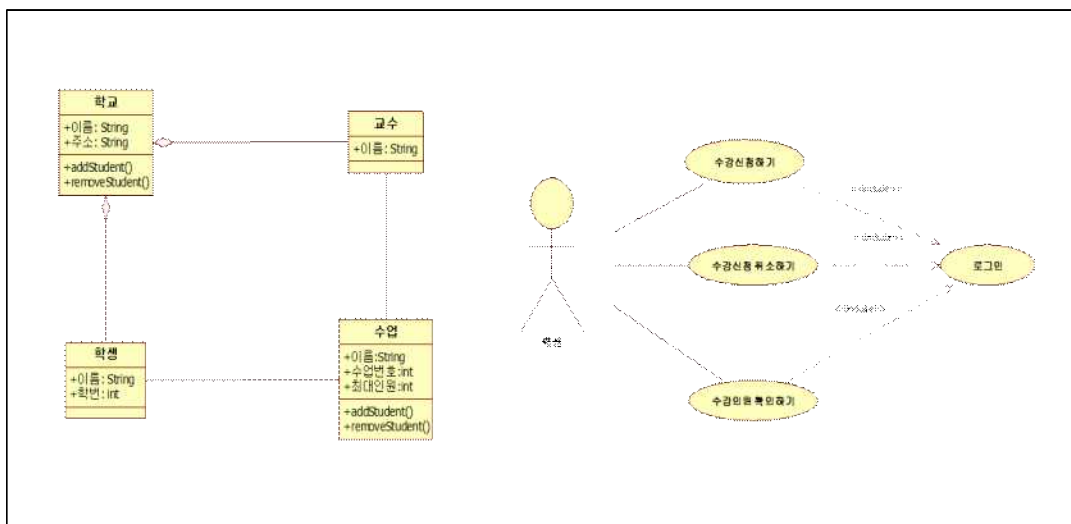
선행 연구에 따르면 게임디자인의 설계는 게임 규칙과 게임 콘텐츠의 설계로 나눌 수 있다[2]. 이중 시스템 기획은 규칙 설계로 표현하는 방식으로는 글이나 그림으로 나타내는 전통적인 문서표현(written representation) 방식[2,3], 소프트웨어 집약 시스템을 개발할 때 사용하는 UML로 표현한 방식[4,5,6], 이산사건 시스템을 모델링하기 위한 Petri Net으로 표현하는 방식들이 있다[7,8,9]. 시스템 설계 및 검증을 위한 도구로 고려해 볼 때, UML과 Petri Net이 적합하다고 판단하였다. 본 연구에서는 기존의 게임 시스템 기획에 관련된 연구를 바탕으로 2가지 도구를 이용하여 직접 게임 내 PVP 시스템을 모델링 하는 방법을 제안하고 검증하여 비교분석한다.

본 논문은 2장에서 관련 연구를 소개하고, 3장에서 선택한 게임의 PVP 시스템을 해당 도구를 이용, 직접 모델링하여 비교분석한 후, 4장에서 결론을 내린다.

2. 관련 연구

2.1 UML(Unified Modeling Language)

UML은 소프트웨어 시스템 개발 과정에서 산출된 내용을 명세화, 시각화, 문서화하기 위한 목적으로 개발된 객체지향 적 범용 모델링 언어이다. 미국 Rational 사의 Grady Booch, James Rumbaugh에 의해 개발되었으며, 1997년 11월 OMG(Object Management Group)에 의해 표준 모델링 언어로 채택되었다. UML의 표현방식은 구조 다이어그램 7종, 행위 다이어그램 7종의 총 14종류의 다이어그램을 기반으로 [그림 1] 과 같이 소프트웨어 시스템의 설계내용을 가시적으로 표현이 가능하다. 또한 객체지향 개념을 이용하여 소프트웨어 설계 외의 시스템도 모델링이 가능하도록 되어 있으며 의미와 표기의 통일로 개발자간의 의사소통의 불일치를 해소 하였다[4].



[그림 1] UML 다이어그램 예시: 클래스 다이어그램(좌), 유스케이스 다이어그램(우)

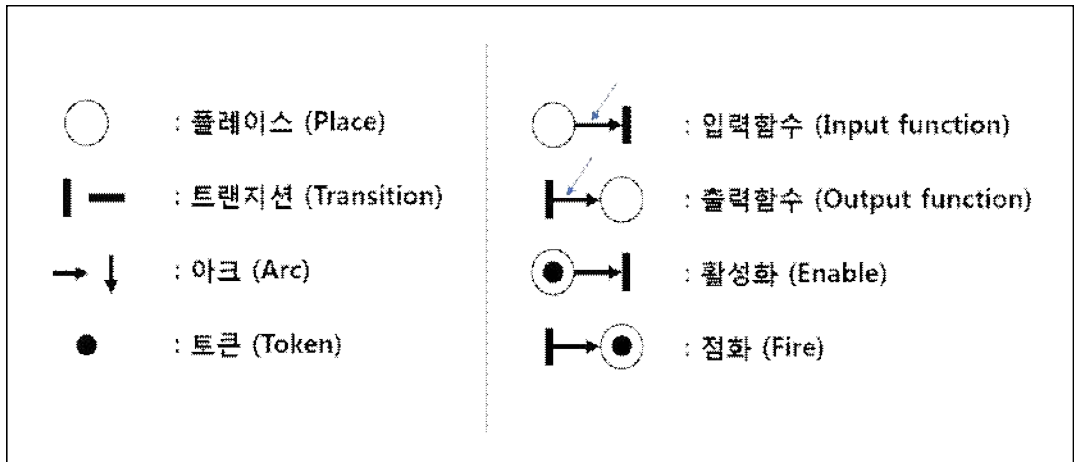
[Fig. 1] UML Diagram examples: Class Diagram(Left), Use-case Diagram(Right)

UML은 각 다이어그램의 특성에 따라 다양한 관점에서의 모델링을 제공한다[2]. 구조 다이어그램(Structure diagram)은 관계성을 중시하여 시스템의 정적인 구조를 보여준다. 각 객체의 정보나 그룹화 된 형태, 관계를 표현하는데 그 목적이 있으며 클래스 다이어그램(Class diagram), 객체 다이어그램(Object diagram) 등이 있다.

반면에 행위 다이어그램(Behavioral diagram)은 시스템의 객체간의 동적인 형태를 보여준다. 특정 사용자 관점의 동작표현이나, 시간의 흐름에 따른 상호작용 표현, 수행과정 등을 표현하며 대표적으로 유스케이스 다이어그램(Use-case diagram), 순차 다이어그램(Sequence diagram) 등이 이에 속한다.

2.2 Petri Net

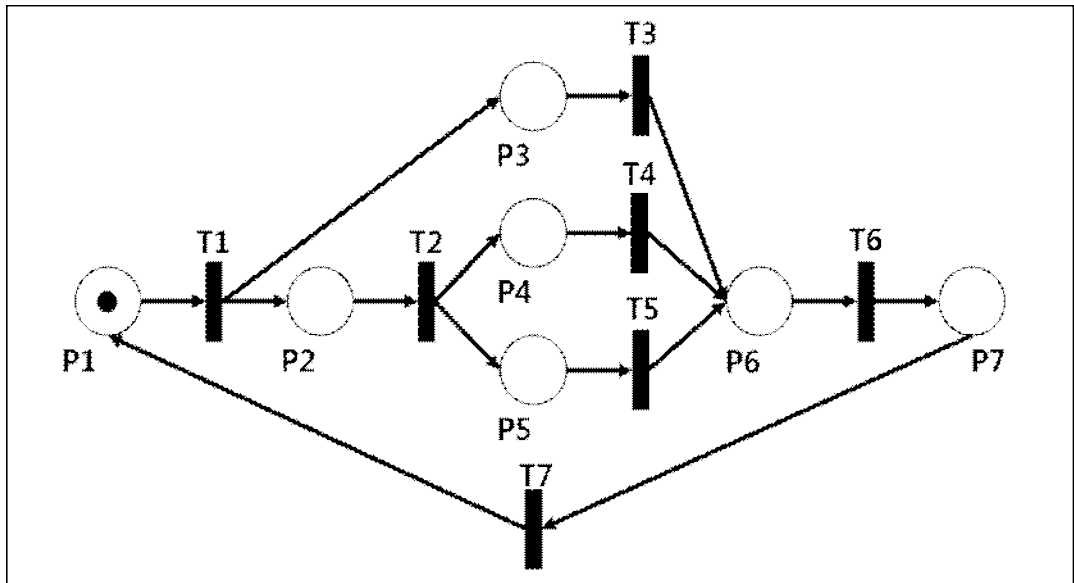
Petri Net은 1960년대 C.A. Petri에 의해 이산사건 시스템(Discrete event system)의 해석으로 처음 제시되었다. Petri Net은 동시성(Concurrency)과 동기적인 사건(Synchronized Event)을 가시적으로 모델링하고 설계하는 도구로 시스템을 상태와 사상의 관계에 의해 기술한다[7,10]. 상태는 시스템에서 일련의 조건이 성립하는 상황을 말하며 하나의 상태에 대응하는 사상이 발생 가능하고, 이때 발생한 사상은 다시 새로운 상태로 이동한다. 이러한 인과관계에 의해 모형화 된 것이 Petri Net으로 총 5가지의 요소로 구성된다. 상태를 나타내는 플레이스(P: Place), 사상을 나타내는 트랜지션(T: Transition), 플레이스로부터 트랜지션으로의 인과관계(아크: Arc)로 표현되는 입력함수(I: Input function), 반대로 트랜지션으로부터 플레이스로 이어지는 인과관계로 표현되는 출력함수(O: Output function), 마지막으로 시스템의 동작을 표현하는 각 플레이스에 부여된 상태 값인 토큰의 개수인 마킹(M: Marking)이 있다[10].



[그림 2] Petri Net 구성요소

[Fig. 2] Petri Net component

Petri Net은 사건의 일련의 흐름을 시각적으로 확인할 수 있다. 시스템의 동작을 표현하기 위해 토큰의 개념을 사용하며 토큰은 각 플레이스에 부여된 상태 값을 뜻하며 그래프 상에서는 플레이스 원 내부에 까만 점으로 표시한다. 특정 시점에서 토큰의 분포를 해당 시점의 마킹이라고 하며 마킹은 해당 시점의 정보의 흐름 및 제어 상태를 나타낸다. Petri Net의 토큰은 활성화(Enabled)와 점화(Fire)에 의해 시스템의 동작을 표현한다. 트랜지션의 입력 플레이스에 토큰이 있을 때, 조건이 만족되면 입력 플레이스에 있던 토큰이 트랜지션으로 이동하는 데, 이를 활성화(Enabled)라고 말하며 활성화된 트랜지션은 다시 연결된 출력 플레이스에 토큰을 옮기는데 이것을 점화(Fire)라고 한다[10]. 즉, 활성화 된 트랜지션의 점화 동작이 바로 Petri Net 동작이며 이때 움직이는 토큰의 위치를 통해 현재 시스템에서 적용되고 있는 사건들을 확인할 수 있다. 게임 시스템 기획에서는 이러한 요소를 통해 구성한 시스템을 Petri Net 모델을 통해 시뮬레이션을 할 수 있으며 실시간으로 제어되는 시스템의 각 부분을 점검하고 동시적인 사건과 병렬적, 확률적으로 처리되는 사건들을 시각적으로 확인함으로써 시스템 무결성 확인 및 성능평가를 할 수 있다.



[그림 3] Marked Petri Net 예시

[Fig. 3] Marked Petri Net Example

2.3 리그오브레전드(League of Legends)

리그오브레전드는 라이엇 게임즈에서 개발, 서비스 중인 온라인 MOBA(Multiplayer Online Battle Arena) 장르의 게임이다[11]. 5:5 팀 대전을 기본으로 소규모 교전이 빈번하게 일어나는 길(라인)과 정글이라 불리는 몬스터들이 등장하는 지형이 갖춰진 맵에서 게임이 이뤄진다. 유저는 캐릭터 하나를 선택하여 역할군 탑, 정글, 미드, 원딜, 서폿 중 하나를 수행하게 된다. 게임당 소요시간은 약 40분 ~ 1시간 내외로 각 팀의 중앙건물인 넥서스를 파괴하거나 어느 한 팀이 항복을 할 경우 게임이 종료된다. 타 게임들과 같이 교전에 영향을 끼치는 변수로 캐릭터의 스테이터스와 착용한 아이템, 지형, 오브젝트 등 다양하게 있으나 공격과 이동 등의 판정 시 고려해야 할 점이 비교적 명확하여 교전 시 일어나는 캐릭터의 동작 중심으로 시스템 모델링에 적용하면 될 것으로 판단하여 본 논문의 사례로 채택하였다.



[그림 4] 리그오브레전드 플레이화면

[Fig. 4] League of Legends Play Scene

3. PVP 시스템 모델링 적용 및 비교분석

3.1 게임 PVP 시스템

PVP는 Player versus player의 줄임 말로 플레이어 간에 서로 벌어지는 전투나 그런 행위가 담긴 콘텐츠를 의미한다. PVP는 환경요소, AI 등과 이루어지는 전투를 의미하는 PVE(Player versus environment)와는 다르게 상대하는 사용자의 조작에 따라 예측이 힘든 변수들이 많다. 따라서, 기획자들은 게임의 기획의도나 타겟층에 따라 시스템 구현 시 PVP에 규칙을 부여하였다.

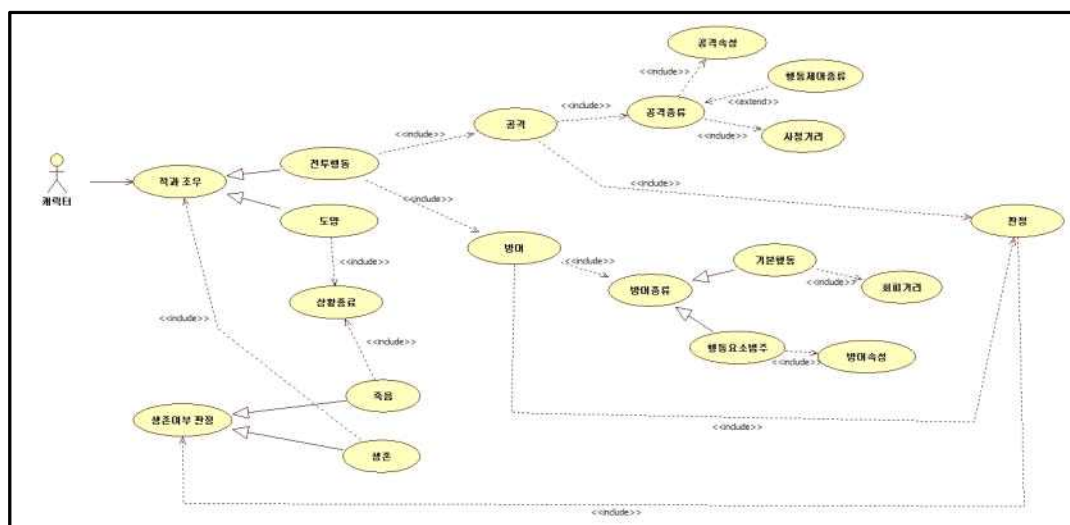
PVP는 크게 제한된 규칙 없이 게임 내에서 자유롭게 유저 간에 전투를 할 수 있게 한 자율적 PVP 유형과 일정한 규칙 아래 특정 공간에서만 전투를 허용한 제한적 PVP 유형, 진영 간의 전투를 의미하는 대규모 PVP인 RVR(Realm versus realm) 등의 유형

등으로 분류된다. 흔히 다른 플레이어들을 죽이는 행위인 PK(Player killing)과의 차이는 사전에 전투가 합의가 되어있는지의 유무로 나뉘며 일부 게임에서는 자율적 PVP 유형의 범주 내에 PK를 넣기도 한다.

본 논문에서는 리그오브레전드 게임의 1:1 상황에서의 PVP 시스템을 UML과 Petri Net을 이용하여 모델링하고 비교분석한다. 모델링 시 고려해야 할 1:1 상황에서의 PVP 시스템 게임을 구성하는 요소로 캐릭터의 기본 공격 형태, 스킬유형과 전투상황 종료 요건 등을 선정하였고, 동일 시스템을 UML과 Petri Net으로 모델링하여 비교분석하기 위해 아바타, 데미지 등의 수치에 영향을 주는 변수는 전부 동일한 조건으로 가정하였다.

3.2 UML(유스케이스 다이어그램) 모델링 적용사례

플레이어를 캐릭터에 투영시켜 사용자관점에서 1:1 전투 상에서 일어나는 시스템 내 동작들을 표현하기 위해 UML 다이어그램 중 유스케이스 다이어그램을 이용하여 모델링 하였다. 모델링을 위해 사용한 UML 프로그램은 StarUML 5.0이다[12].



[Fig. 5] League of Legends 1: 1 PVP system expressed in use-case diagram

[그림 6]와 같이 각 상태 및 동작들을 21개의 플레이스로 나타내었고, 트랜지션들은 플레이스로 표현된 상태 및 동작들을 처리·입력하는 의미로 시스템 흐름에 따라 조건에 맞게 부여하였다. 초기에 배치되는 토큰은 실행 시 아크로 연결된 트랜지션으로 이동하여 활성화 시키고 다시 아크로 연결된 플레이스에 점화되어 이동된다. 최종적으로 PVP 상황이 종료되는 조건은 상황종료 플레이스로 도달하게 되는 조건을 달성하게 될 경우로 적과 대치 시 도망을 가거나 생존여부 판정 시 죽음 판정이 일어났을 때 종료되며 이외 상황에서는 위와 같은 과정을 계속해서 반복하게 된다.

3.4 비교 분석

본 사례연구를 통해 UML과 Petri Net 기반의 1:1 PVP 시스템 모델링 결과를 제시하였다.

위의 검증 결과를 토대로 게임 기획 시 적합성을 기준으로 두 모델링 도구를 비교분석하여, [표 1]과 같이 정리하였다.

[표 6] 게임 기획 시 모델링 도구로서의 UML과 Petri Net의 비교분석표

[Table 1] Comparison of UML and Petri Net as modeling tools for game design

	UML Use-case Diagram	Petri Net
시각화	시각적 표현 우수, 단순화, 직관적	시각적 표현 가능
시스템 검증	제한적(외부 프로그램 사용)	자동화, 로직 흐름 확인 우수
표현력	요구사항, 관계성 표현	시스템 흐름까지 표현
장점	편리한 작성, 직관적인 표현	동작 시뮬레이션 가능, 밸런스 통계 분석 가능, 논리적 오류 발견 우수
단점	시스템 검증이 제한적	규칙 학습 필요

UML 유스케이스 다이어그램을 이용한 모델링의 결과, 캐릭터를 중심으로 1:1 PVP 상황에서 필요한 시스템 기능 등을 시각적으로 정의하였다. 표현하기 위해 사용한 기호들로 인해 한눈에 직관적으로 내용구성을 파악할 수 있는 것으로 나타났으며, 각 정의

된 유스케이스 간의 관계를 표현 할 수 있어 전반적인 시스템의 흐름을 이해를 도왔다. 그러나 시스템의 무결성을 검증하는 과정에 있어 UML과 연동되어 자동화된 소스코드 생성 프로그램이 없을 경우나 상세한 명세가 필요할 경우에는 직접 코드 입력을 통한 프로그램 작성이 불가피하다는 점을 확인하였다. 따라서 UML을 통한 시스템 모델링은 시각화와 논리적 표현력이 우수하며 작성 및 수정이 편하다는 장점이 있고, 시스템 검증 과정에 있어 가능한 하나 다소 불편함이 요구되는 것이 단점이다.

Petri Net을 통한 모델링의 결과, 각 기능과 상태를 중심으로 1:1 PVP 시스템의 정보 흐름을 시각적으로 표기하였다. UML과 마찬가지로 전체 시스템의 흐름을 파악 할 수 있고, 각 플레이스 간의 관계를 아크를 이용해 표현이 가능하다. 그러나 Petri Net의 요소와 기능 및 동작을 이해해야 전반적인 흐름에 대해 이해할 수 있다는 점을 확인하였다. 시스템 무결성 검증 분야에서는 토큰을 부여하여 별도의 추가 작업 없이 직접 시뮬레이팅을 통해 동시적인 동작과 동기 적으로 일어나는 동작들을 확인하고 논리적 오류를 쉽게 발견 할 수 있다는 점에서 매우 우수하다. 또한 각 플레이스에 도달한 횟수를 기준으로 데이터화 하여 통계분석을 하여 확률적 측면에서의 밸런스 확인이 가능하다. 하지만, 설계한 시스템 내용을 수정할 경우 각 플레이스와 트랜지션의 관계성과 토큰의 이동규칙을 고려해야 한다는 점에서 불편함이 있다.

5. 결론

게임 시스템 기획자들은 게임의 짧은 수명으로 인한 빠른 개발과 출시로 이어지고 있는 현 시장의 흐름의 영향으로 이전 어느 때 보다도 정확한 설계를 요구받고 있다. 또한 시스템 기획은 개발에 들어가기에 앞서 설계도를 그리는 중요한 과정이기에 기획 단계에서부터 형상관리 및 검증이 필요하며 이를 위해 유용하게 사용할 수 있는 방법이 UML, Petri Net을 이용한 모델링이다. 본 논문에서는 실시간으로 이루어지는 게임 내 PVP 시스템을 UML과 Petri Net을 통해 모델링 하였고 그 적용사례를 이용하여 2가지 모델링 도구의 비교분석을 해보았다.

실제 시스템 상에서 고려해야 할 요소인 수치 디자인 요소를 동일한 조건으로 가정하여 동작 및 상태 중심으로 고려하여 설계 한 결과, UML 유스케이스 다이어그램의

유스케이스와 Petri Net의 플레이스로 표현이 가능하였다. 둘 다 관계성을 가시적으로 표현하여 논리적인 흐름을 직관적으로 확인이 가능 했으며, UML은 편리한 작성과 관계표현에 더욱 강점이 있었고, Petri Net은 시뮬레이팅을 통해 동시적 사건의 흐름 확인과 무결성 검증에 효과적이라는 것을 확인할 수 있었다. 요구사항 및 관계 정의와 같은 선행 설계를 UML 유스케이스 다이어그램으로 한 뒤 이를 다시 Petri Net으로 변환하여 시스템의 무결성 검증과정을 거친다면 서로의 장·단점을 보완하여 정확한 시스템 설계를 기대할 수 있을 것이다.

본 연구에서 모델링한 사례는 실제 시스템 상에서 고려해야 할 요소인 수치 디자인 요소를 동일한 조건으로 가정하여 설계하였기에 다양한 변수들이 포함되지 않은 한계점이 있었다. 향후 연구에서는 다양한 개념들과 변수들을 추가하여 더 큰 범주로 고도화된 시스템을 모델링 하는 방법에 대한 연구가 필요할 것이다.

References

- [1] Jung Yoon Kim. "A Testcase Generation Techniques for Dynamic Game Map using TFD." *Journal of The Korean Society for Computer Game* 26.1 (2013): 229-237
- [2] Hee Dong Chang. "A Survey of Representation Methods of Game Rules in Game Design." *Journal of Korea Game Society* 6.4 (2006): 39-45
- [3] Bethke, Erik. "Game development and production". Wordware Publishing, Inc., (2003)
- [4] Hee Dong Chang. "A Representation Method of Game Mechanics Using UML Notations in Game Design." *Journal of Korea Game Society* 6.4 (2006): 47-53
- [5] Sung Choi. "Study On the Object Oriented Design Project of Online Game Engine Using UML." *Journal of Korea Game Society* 5.1 (2005): 33-40
- [6] Cheol Jung Yoo, and So Young Jeong. "Software Engineering: Use Case Diagram Extraction Technique from Requirements Specification" *The KIPS Transactions: Part D* (2002): 639-650
- [7] Hee Dong Chang. "A Method of Gameplay Analysis by Petri Net Model Simulation." *Journal of Korea Game Society* 15.5 (2015): 49-56
- [8] Hee Dong Chang. "A Survey of Game Mechanics Simulation Methods." *The KIPS Transactions: Part A* 12.5 (2005): 441-450
- [9] Hee Dong Chang. "An Object-oriented Design Method of Game System for Game Designers." *Journal of Korea Game Society* 16.3 (2016): 17-26
- [10] Boo Hee Nam. "Petri nets." (The) *proceedings of KIEE* 43.3 (1994): 10-17
- [11] League of Legends Official Homepage, <https://kr.leagueoflegends.com/>, Retrieved March 22 2018
- [12] StarUML 5.0 Homepage, <https://sourceforge.net/projects/staruml/>, Retrieved July 11 2014
- [13] HiPS Tool Homepage, <http://hips-tools.sourceforge.net/wiki/>, Retrieved November 19 2017

