

# 온라인 게임 환경에서 FSM 기반 테스트 자동화 시스템

정훈영\*, 정현준\*\*, 백두권\*\*†

\*고려대학교 소프트웨어공학과

\*\*고려대학교 컴퓨터전파통신공학과

e-mail : redcatt33@nate.com, junghj85@gmail.com, baikdk@korea.ac.kr

## A FSM based Test Automation System in An Online Game Environment

Hoon-Young Chung\*, Hyun-Jun Jung\*\*, Doo-Kwon Baik\*\*†

\*Dept. of Software Engineering, Korea University

\*\*Dept. of Computer And Radio Communications Engineering, Korea University

### 요 약

이 논문에서는 FSM(Finite State Machine) 기반 온라인 게임 서버 테스트 자동화 시스템을 제안한다. 게임 서버의 테스트 자동화는 테스트로 인한 비용, 시간적 제약, 테스트 자원의 재사용 측면에서 이득이 있다. 기존의 테스트 자동화 방식은 사용자간의 상호작용을 고려하지 않기 때문에 게임 서버의 성능에 대한 정확한 측정이 어렵다. 또한, 실제 서비스 시에 수용할 수 있는 동시 접속 인원수를 예측하기 어렵다. 이 논문에서 가상 유저를 이용한 테스트를 할 경우 FSM 을 이용하여 가상유저간의 상호작용이 가능하게 한다. 이를 이용하여 게임 서버의 성능 측정의 정확도 개선이 가능하다. 제안 시스템을 검증하기 위해 상호작용을 고려하지 않은 테스트 방법과 비교 평가 하였다.

### 1. 서론

온라인 게임 서버 테스트란 게임 서버의 다중 접속 처리에 대한 안정성 검사와 게임 콘텐츠의 오류 검사 등의 서비스 대상의 기능들의 성능을 검증하는 품질 보증 절차이다[1]. 기존의 온라인 게임 서버 테스트는 기능을 수행 동시에 얼마만큼의 사용자를 수용할 수 있는가에 대한 평가로 이루어진다[2]. 이러한 테스트 과정은 일정 인원의 사람이 일정 시간 동안 진행하기 때문에 인건비, 테스트 자원의 재사용 불가, 장비 운용 부담 과 같은 문제점이 발생한다[3]. 이를 해결하기 위하여 최근에는 온라인 게임 서버의 성능 테스트 자동화 연구가 진행되고 있다. 온라인 게임 서버 테스트 자동화 기술로는 가상 유저를 이용한 게임 서버 부하 테스트 기술이 있다[4]. 이 기술은 플레이어를 대체할 가상의 사용자를 컴퓨터 상에서 구현하여 서버의 성능을 측정하는 방법이다. 적은 비용으로 게임 서버의 서비스 가능 여부를 테스트 할 수 있는 장점이 있다. 그러나, 이 방법은 게임 서버의 네트워크 부하 테스트를 집중적으로 하고 있어서 한계가 있다.

기존의 테스트는 성능에 대한 정확한 측정이 어렵다. 실제 사용자가 보낼 수 있는 다양한 이벤트와 게임 서버의 알고리즘 수행 시간 등을 고려한 콘텐츠 제공 성능을 측정하기 어렵다[4]. 또한 가상 유저간의 상호작용이 고려되지 않았다. 이로 인해 게임 서버의

동시 수용 가능한 최대 인원과, 실제 서비스에서 원활한 서비스를 위한 동시 접속 인원과 차이가 발생한다. 예를 들어, 사용자 간의 전투를 위한 게임 서버의 성능은 상호 작용에 따른 사용자 행동에 기반하여 공격 유효 판정, 전투 공식, 공격 범위 계산 알고리즘이 반영된 결과로 도출해야 한다. 즉, 가상 유저의 상호 작용을 고려한 테스트가 이루어져야 한다.

제안 시스템에서는 FSM 을 가상 유저에 적용하여 개선된 온라인 게임 서버의 테스트 자동화 시스템을 제안한다. 사용자의 상호 작용을 고려한 가상유저를 생성하여 테스트하여 실제 서비스와 비슷한 결과를 얻을 수 있다.

이 논문은 다음과 같이 구성된다.

2 장에서는 게임 테스트 자동화에 관련된 기존 연구에 관하여 서술한다. 3 장에서는 사용자의 상호 작용이 적용 가능한 테스트 시스템을 제안한다. 4 장에서는 제안 시스템을 구현하여 실험 및 평가를 하고, 5 장에서 결론 및 향후 연구에 대해 기술한다.

### 2. 관련 연구

가상 유저란 게임 클라이언트의 그래픽 및 사운드 처리는 하지 않고 네트워크 통신만을 수행하여 서버로 하여금 접속된 사용자로 인식하도록 하는 경량화된 테스트 클라이언트를 의미한다[4]. 온라인 게임의

† 교신저자 : 백두권

특성상 게임 클라이언트의 자원을 많이 필요로 하기 때문에 현대의 PC 에서 둘 이상의 게임 클라이언트가 실행되기 어렵다. 그리고 실제 구동 환경 같은 테스트는 비용 등의 문제를 발생시킨다. 이러한 문제를 해결하기 위해 가상 유저를 이용한 테스트 자동화 방법들이 연구되어 왔다. 대표적인 테스트 자동화에 관한 연구로 에이전트에 의한 온라인 게임 서버 테스트 자동화[3]와 가상 유저에 의한 온라인 게임 서버 부하 테스트 기술[4]을 들 수 있다.

에이전트에 의한 온라인 게임 서버 테스트 자동화 [3]는 가상 유저의 역할을 하는 Virtual Client(VC), 가상 유저를 생성 하는 VC Agent, 가상 유저의 제어를 위한 Central Engineering Station(CES) 로 구성되어 있으며, 가상 유저를 제어해 대규모 사용자의 온라인 게임 환경을 시뮬레이션 할 수 있다. 그 결과 테스트에 의한 비용 절감, 자동화를 실현하였으나, 게임 서버로의 로그인, 로그아웃, game zone 이동 등의 단순한 이벤트에 집중된 테스트 기법으로 다양한 이벤트를 통한 게임 서버의 테스트가 이루어 지지 않았다. 이에, [3]연구에서도 향후 Virtual Client(VC)를 지능적으로 구동시키기 위한 인공지능 기술의 적용이 필요함을 언급했다.

가상 유저에 의한 온라인 게임 서버 테스트 부하 테스트 기술[4]에서는 현대의 PC 에 300 개 이상의 가상 유저를 생성하고 여러 대의 PC 를 통합하여 관리 할 수 있는 시스템을 구축 하였다. 이 시스템은 게임 서버에 대용량 부하를 발생시켜 서버의 안정성 및 용량을 점검하여 온라인 게임의 품질 검증(Quality Assurance)를 용이하게 하였다. 에이전트에 의한 온라인 게임 서버 테스트 자동화[3]에서 해결하지 못한 가상 유저의 지능적 구동을 위하여 가상 유저 제어 도구를 통해 시나리오 적용 기능을 제공하였으나, 가상 유저간의 상호작용을 부여하여 다양한 행동을 자동으로 발생시키지는 못했다. 가상 유저의 상호작용을 고려하지 않은 게임 서버 테스트는 측정된 최대 접속 인원과, 원활한 서비스를 위한 동시 접속 인원의 차이가 발생한다. 그리고, 서버에서 제공하는 게임 콘텐츠를 충분히 소비하는 상황에서의 성능 테스트가 수행되기 어렵다. 따라서, 가상유저간의 상호작용, 가상유저와 서버와의 상호작용을 부여한 실제 플레이어의 자연스러운 행동에 기반한 이벤트를 발생 시키는 방법에 대한 개선이 필요하다.

### 3. FSM 기반 온라인 게임서버 테스트 자동화 시스템

제안하는 시스템은 가상 유저간의 상호작용을 고려한 온라인 게임 서버의 성능 테스트를 가능하게 한다. 대규모 가상유저를 통한 게임서버의 성능과 용량 테스트와, 가상유저를 지능화 하여 현실성 있는 이벤트를 발생시켜 원활한 게임플레이를 위한 테스트를 자동화 한다. 제안하는 시스템을 위한 가상유저 지능화 방안으로, FSM 을 이용한 알고리즘을 적용하기로 한다.

FSM 은 유한한 수의 상태를 가지고 외부로부터의 입력을 받아들이며, 현재의 상태와 외부로부터의 입

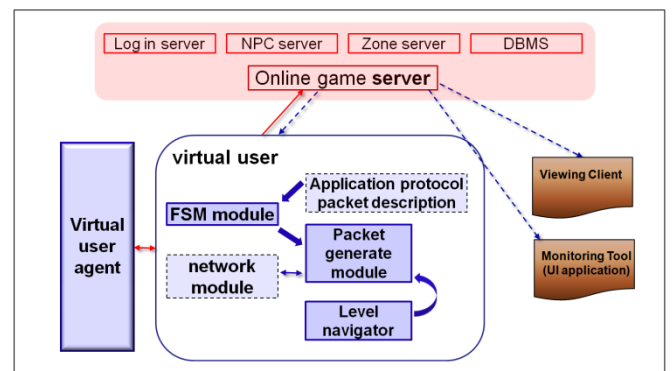
력에 따라 주어진 규칙에 의해 상태 변화를 일으킬 수 있는 규칙 기반 시스템으로 다양한 종류의 게임 분야에 널리 사용 되는 기법이다. FSM 은 상태 다이어그램을 통해 행동을 정의 하기가 쉽고, 조건과 분기문을 통해 쉽게 구현할 수 있다는 장점이 있다 [6,7,8].

FSM 이외에도 FuSM[6], 유전자알고리즘[7], 신경망 알고리즘[9] 등 게임 인공 지능에서 많이 사용되는 기법이 있지만, 이는 모두 플레이어의 재미를 목적으로 개선된 방법으로, 다양한 이벤트를 발생시키기 위한 가상 유저의 지능화 방안 범위를 벗어난다. 따라서, 가장 구현이 용이하고 온라인 게임 서버 테스트의 목적에 충분히 부합하는 FSM 을 이용한 알고리즘을 적용하기로 한다.

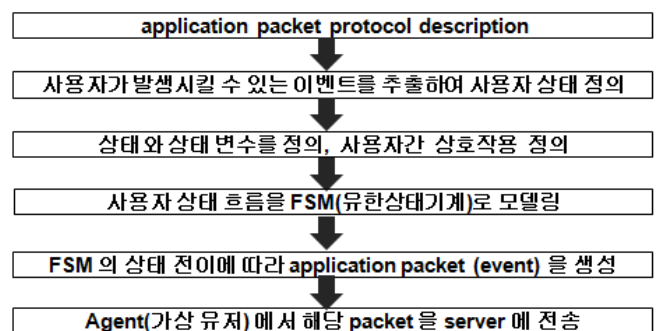
#### 3.1 테스트 자동화 시스템 구성

이 논문에서 제안하는 FSM 기반 온라인 게임 서버 테스트 자동화 방안을 위한 시스템의 구조는(그림 1)과 같다.

제안하는 시스템은 “virtual user agent”, “FSM 모듈”, “Packet generator 모듈”, “Level navigator 모듈” 로 구성된다. “Virtual user agent”에서는 가상의 클라이언트를 생성하고 관리 한다. “FSM 모듈”은 가상유저 또는, 가상유저의 상호작용에 의한 상태전이를 제어한다. “Packet generator 모듈”은 “FSM 모듈”에서 생성된 상태를 기반으로 온라인 게임 서버에 전송할 Application packet 을 생성하고, “Packet generate module”을 이용하여 서버로부터 전송 받은 메시지를 해석하여 “FSM 모듈”로 넘겨준다.



(그림 1) 제안 시스템 구조



(그림 2) FSM 구현 방법

“Level navigator 모듈”은 움직일 수 있는 게임 유닛들의 위치 또는 이동 불가능 지역 등의 지형데이터를 관리하고 유닛들이 이동할 좌표를 생성한다. 제안하는 모델의 이벤트 발생은 다음의 과정을 통해 이루어진다.

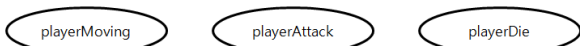
- 1 단계 : 가상유저가 생성되고 게임서버에 접속한다.
- 2 단계 : “FSM module”에서 가상유저의 상태를 전이시킨다. 이때, 이동과 관련된 상태로 전이되면 “Level navigator 모듈”로 좌표를 생성한다.
- 3 단계 : “FSM module”로 전이된 상태에 따라 “Packet generator 모듈”에서 Application packet 으로 구성하여 “network module”로 보낸다.
- 4 단계 : “virtual user”는 생성된 Application packet 을 게임 서버로 전송하고, 게임 서버서부터 받은 Application packet 을 “Packet generate module”로 분석하고 “FSM 모듈”에 제공한다.
- 5 단계 : 2 단계 이후부터 “FSM 모듈”은 게임 서버로의 응답 상황에 따라 가상유저의 상호작용을 고려한 상태전이를 일으키고 이벤트를 발생시킨다.

### 3.2 가상 유저의 상호 작용을 위한 FSM 구현 방법

가상 유저간의 상호 작용을 부여하기 위한 FSM의 구현 방법은 (그림 2)와 같다.

정의된 게임 시스템의 application packet protocol description 으로부터, 사용자가 발생시킬 수 있는 이벤트를 packet 단위로 구분하여 사용자의 상태를 정의한다. 온라인상에서 사용자의 상태는 게임 시스템이 정의한 application packet protocol description 에 명시된 FSM 에 정의될 유한 개의 상태가 된다. 이렇게 정의된 상태를 기반으로, 상태간의 전이를 발생시킬 수 있는 입력으로 작용하는 상태 변수를 정의한다. 이때 하나의 개체에서 정의된 상태가 다른 개체의 상태 전이를 일으킬 수 있는 상태 변수가 될 수 있는데, 이들을 쌍으로 묶어 개체간의 상호 작용을 통한 전이로 구분한다. 연관된 상태들을 포함한 외부 입력이 서로의 전이를 일으킬 수 있는 상태변수가 되도록 FSM 을 구성한다.

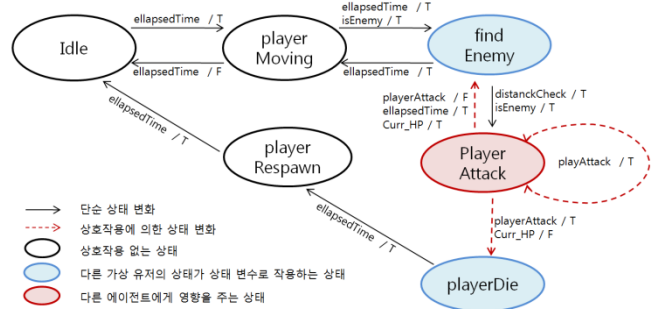
<b>playerMoving</b>	p_size	protocol_name	cid	location_curr	location_tar	speed			
	int	int	int	location	location	double			
<b>playerAttack</b>	p_size	protocol_name	cid	tid	attack_type	chp	cmp	tmp	
	int	int	int	int	int	int	int	int	
<b>playerDie</b>	p_size	protocol_name	cid	location_curr	curr_time				
	int	int	int	location	double				



(그림 3) Packet 단위의 상태 정의 예시

<표 1> 상태 및 상태변수 정의 예시

State	Input	Output
<b>playerMoving</b>	<b>Idle</b> + elapsedTime	fineEnemy
<b>findEnemy</b>	elapsedTime	playerAttack
<b>playerAttack</b>	distanceCheck+isEnemy	playerAttack
<b>playerAttack</b>	<b>playerAttack</b>	playerAttack
<b>playerAttack</b>	<b>playerAttack</b>	playerDie
<b>playerDie</b>	<b>playerAttack</b> +currHP	playerRespawn
<b>playerRespawn</b>	elapsedTime	Idle
<b>Idle</b>	ElapsedTime	playerMoving



(그림 4) 상호작용을 고려한 FSM 설계 예시

구성된 FMS 는 게임서버로부터 전송된 특정개체의 상태에 따라 전이가 가능하다. 상태 전이를 일으킨 개체는 상호작용이 종료되는 상태로 전이될 때까지 자신에게 입력으로 작용한 개체의 id 를 포함한 또 다른 상태변수로 게임서버에 전송되며 이는 상호작용하는 개체의 입력으로 사용된다.

## 4. 실험 및 평가

### 4.1 실험

가상 유저는 C++ 과 winsock, 으로 windows 7 환경 하에서 구현 한다. 테스트를 위한 서버는 MMORPG 를 위해 개발된 온라인 게임 서버로 winsock 을 이용한 TCP/IP 프로토콜의 IOCP 모델로 windows server 2008 환경에서 구현되었다. 게임 서버에서 고려되어야 할 주요 알고리즘으로는 사용자의 전투시 전투 공식에 의한 피해량 산정, 최단 거리 알고리즘, 확률 공식에 의한 아이템 생성이 있다.

컴퓨터 한대에 200 개의 가상유저를 실행시켰으며, 테스트 결과를 측정하기 위해 전체 게임 월드에 분포된 가상 유저의 군집을 점으로 표현할 수 있는 dummy 클라이언트인 Client Monitor 를 구현하여 10 초마다 128 byte 의 채팅메시지를 서버로 전송하고, echo 되는 시간을 측정하였다.

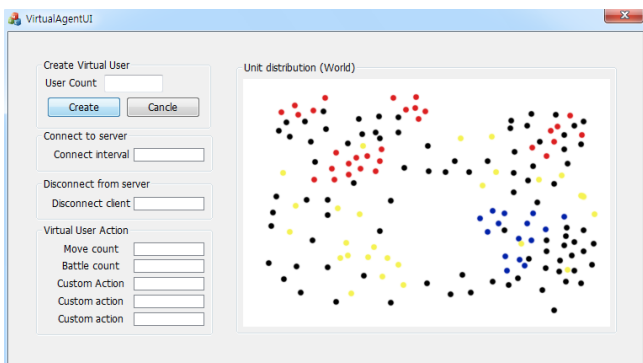
비교를 위해 가상 유저의 상태전이를 고려 하지 않은 접속 및 단순 패킷 전송을 하였다. 128 byte 의 채팅 메시지를 서버로 전송하고, 게임 서버에서는 echo 하였다. 128 byte 를 전송한 이유는 테스트 환경에서 클라이언트와 서버가 통신할 수 있는 최대 패킷의 크기가 128 byte 를 넘지 않기 때문이다. 위 실험과의 비교를 위해 상태의수를 3 개, 5 개, 10 개로 정의하여 각각 수행하였으며, 총 5 대의 컴퓨터에 가상 유저를 순차적으로 실행시키고 가상 유저와 상태 수에 따른 Client Monitor 의 채팅 메시지 반응 속도를 측정하였

다. Client Monitor 에서는, 각 상태 별로 색깔을 구분하여 각각의 상태가 전이되고 있는 지를 확인하게 하였다. 측정 결과는 <표 2>와 같다. 가로 축은 가상 유저의 수를, 세로 축은 적용한 상태의 개수를 나타낸다. 가상 유저의 상호작용을 고려하지 않은 No trans 상태에서는 반응속도 0.12 로 1000 명 정도를 수용해도 안정적인 서비스를 할 수 있는 성능으로 측정되었다. 그러나, 10 개의 상태를 정의하고 상호작용에 의한 상태전이를 적용하여 테스트 했을 때는 800 명의 가상유저에서 1.2 의는 반응속도를 보인다.

4.2 평가

실제 게임에서 1 초 이상의 지연은 서비스 불만족을 야기시킨다. 따라서, 해당 서버에서 동작하는 사용자 상호작용에 관한 알고리즘의 처리시간을 고려해 콘텐츠를 원활하게 제공하기 위한 최대수용인원은 600 명 정도가 적당하다는 결과가 나온다. 이 결과는 가상 유저의 상호 작용을 고려하지 않은 상태에서의 게임 서버의 수용 인원으로 측정된 1000 명과 400 명의 차이를 보인다. (그림 6)은 1 초 지연을 기준으로 한 최대 수용 인원의 도달점을 보여주고 있다.

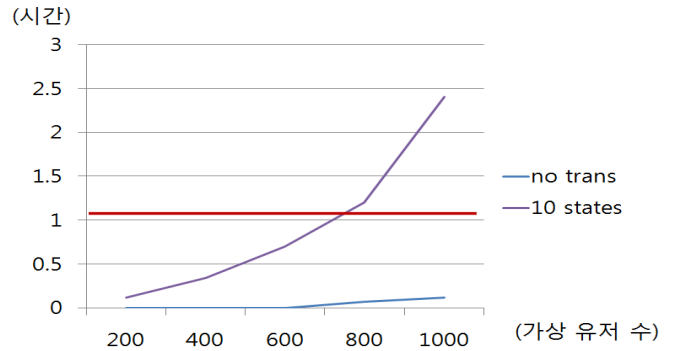
제안 시스템은 가상 유저의 상호작용을 고려한 상태의 수에 따라 지연 시간을 초과하는 접속 인원의 차이를 보인다. 이는 기존의 테스트 방법이라 할 수 있는 no trans(상호작용에 의한 상태 전이 없음)와 차이를 보인다. 즉, 기존의 가상 유저의 상호작용을 고려하지 않은 게임 서버 테스트의 방법을 개선했다 할 수 있다. 게임 시스템에서 적용 가능한 모든 상태들을 적용하지 않더라도 출현 빈도가 높을 것으로 예상되는 상태들로 FSM 을 구성하여 실제 안정적인 서비스를 위한 게임 서버의 적절한 수용인원을 산정 할 수 있다.



(그림 5) Client Monitor

<표 2> 상태와 가상 유저 수에 따른 응답 속도 비교

가상유저수 상태수	200	400	600	800	1000
No trans	0.0	0.0	0.0	0.07	0.12
3 states	0.0	0.0	0.1	0.17	0.24
5 states	0.04	0.17	0.3	0.6	0.9
10 states	0.12	0.34	0.7	1.2	2.4



(그림 6) 개선된 테스트 시스템과의 비교

5. 결론 및 향후 연구

이 논문에서는 안정적 서비스를 위한 온라인 게임 서버의 성능을 테스트하기 위해 FSM 을 이용해 가상 유저간의 상호작용을 부여한 온라인 게임 서버 테스트 자동화 방안을 제안 한다. 가상 유저의 다양한 이벤트 발생은 클라이언트를 통해 확인 할 수 있다. 제안한 모델은 실제 사용자와 같은 가상 유저의 행동변화를 발생시켜 사용자간 상호작용을 재현할 수 있어 게임서버의 실제 수용인원에 근접한 성능을 측정할 수 있다. 그 결과 서버의 콘텐츠 제공에 관한 기능적 측면에서 기존의 단순한 게임 서버 부하 테스트 자동화보다 효율적이다.

향후 연구로 제안 시스템의 FSM 을 개선하고, 관련 연구와의 정량적인 비교 평가를 하여 제안 시스템의 우수성을 보이고자 한다.

참고 문헌

- [1] ISO/IEC 9126-2 : Information Technology Software product quality – Part 2 : External Metrics, ISO/IEC JTC1/SC7/WG6, 1999.
- [2] 최용준,박성수,김재원,이범렬, “온라인 게임 서비스 안정화” 전자통신동향분석 Vol.22, No.4, 2007.
- [3] 이현주,정용우,임범현,심광현, “에이전트에 의한 온라인 게임 서버 테스트 자동화” 한국통신학회 논문지 Vol.31, No.5B, 2006.
- [4] 배수영,김정일,손강민,박창준, “가상 유저에 의한 온라인 게임 서버 부하 테스트 기술” 정보과학회지 Vol. 27, No. 10, 2009.
- [5] 유건아, “게임 캐릭터의 전술적 행동을 위한 위치 평가 함수의 학습” 정보과학회논문지 Vol.38, No.2, 2011.
- [6] 박사준, ”규칙과 신경망의 혼합형 모델을 이용한 게임 캐릭터의 행동 제어” 한국 인터넷 정보학회 논문지 Vol 11, No. 2, 2010.
- [7] 권기덕, “계층적 유한 상태 기계를 이용한 NPC 의 행위 제어”, 정보처리학회 Vol.7, No.3, 2009.
- [8] 이만재, ”게임에서의 인공지능기술” 한국정보처리학회지 Vol. 9, No. 3, 2002.
- [9] 조병현, “신경망을 이용한 지능형 게임 캐릭터의 구현”, 한국정보처리학회지 Vol.7, No.96, 2004.