



# 네트워크 기반 게임의 다중시스템 운영 기술

박 철 제\*, 최 성\*\*

● 목 차 ●

1. 네트워크 게임의 현재
2. 기존 게임의 운영형태 고찰
3. 네트워크 기반 게임의 구조 및 운영 형태
4. 결 론

## 1. 네트워크 게임의 현재

### 1.1 네트워크 게임 산업의 중요성

게임 산업은 네트워크 기반 기술, 멀티미디어, 가상현실, 3차원 그래픽 등을 바탕으로 산업 전반의 미래 첨단 산업 기반 기술이 활용된다. 또한 멀티미디어 응용 부분들이 사용되면 정보통신기술의 발달에 기여뿐만 아니라 고부가가치 산업으로서 경제적 이윤 창출을 이룰 수 있다. 그리고 몰입과 반복적인 사용을 요하기 때문에 그 어떠한 것보다 강력한 문화적 영향력을 행사한다. 이에 부작용이 없도록 문화적 측면에 대한 충분한 고려가 필요하나 국가적, 상업적 측면에서 많은 이익을 창출할 수 있다.

### 1.2 네트워크 게임 기술의 발전 양상

(그림 1)의 분류에서는 구현 기술의 측면은 고려되지 않았다. 하지만, 요즘 많이 개발되는 “다중사용자 지원 멀티미디어 네트워크 게임”에서 사용자의 관리와 적정QoS를 만족 하기 위하여, 고속네트워크와 같은 구현기술이 일정 수준이상 뒷받침 되

어야 한다.

연도	90	91	92	93	94	95	96	97	98	99	00	01	02.
게임 제품	가정용 2차원 게임		가정용 3차원 게임				네트워크 지원 가정용 3차원 게임		가정용 네트워크 가상현실 게임				
	PC용 2차원 게임				PC용 3차원 게임		3차원 그래픽 네트워크 게임						
	텍스트 네트워크 게임			2차원 그래픽 네트워크 게임									

(그림 1) 네트워크 게임 제품의 발전 전망

### 1.3 게임 기술의 현황

게임의 표현 및 운영 측면에서는 실제 이미지와 그래픽, 3차원 음향을 적극적으로 사용하였으며, 시각, 청각뿐 아니라 촉각까지 이용한 체감형 게임이 적극 도입되고 있다. 또한 네트워크를 이용하여 다중 사용자가 동시에 게임을 진행할 수 있게 되었으며, 이는 시나리오의 자유도를 크게 높여주었다. 게임의 구성 및 시나리오 측면에서는 아직 정해진 시나리오에 얽매어 게임을 진행하지만, 사용자가 직접 게임의 시나리오를 만들어 가는 동적인 시나리오가 점차 제안되고 있다. 또한 네트워크 기반에서 다중 사용자가 동시에 게임을 진행 함으로서 시나리오 예측이 불가능한 네트워크 게임이 최근 개발되어 그 가능성을 검증 받고 있다.

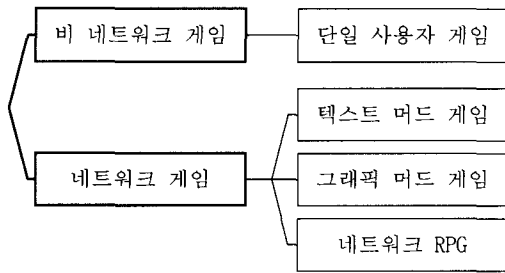
\* 동의대학교 경영정보학과 조교수

\*\* 남서울대학교 컴퓨터학과 교수

## 2. 기존 게임의 운영형태 고찰

### 2.1 기존 게임의 분류

게임은 보통 아케이드(Arcade)게임, 어드벤처(Adventure)게임, 롤플레이(Role Playing) 시뮬레이션(Simulation)게임, 교육용게임, 통신용게임으로 분류한다. 이러한 분류는 네트워크를 배제한 1인용 게임이 주류를 이루었을 때에 적합하다. 하지만, 미래의 게임은 필수적으로 네트워크를 기반으로 하기 때문에 (그림 2)와 같이 분류하여 그 구조와 기술을 고찰한다.



(그림 2) 구조를 기준으로 한 게임의 분류

### 2.2 비 네트워크 게임의 구조 및 운영형태 고찰

비 네트워크 게임은 네트워크를 사용하지 않는 게임을 말한다. 이에 비 네트워크 게임의 사용자는 컴퓨터와 단독으로 게임을 하게 되는 단일 사용자(Single-User) 게임이다. 또한 단일 시스템에서 운영되며, 사용자 역시 단일시스템에서 게임을 사용하게 된다.

비 네트워크 게임은 이러한 구조 때문에 시나리오 상의 제약을 많이 받지만, 네트워크 부하와 같은 다른 제약조건은 없다. 즉, 탄탄한 시나리오 구성이 이루어진다면, 게임을 원활히 서비스하는 데에 문제가 전혀 없다.

## 3. 네트워크 기반 게임의 구조 및 운영 형태

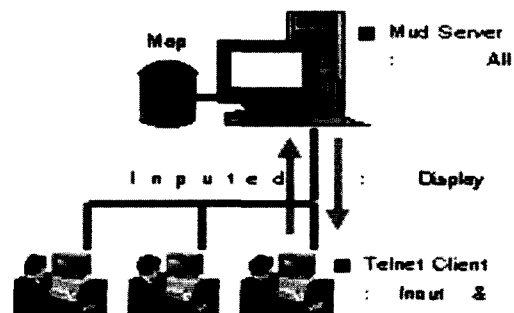
### 3.1 텍스트 머드(Text Mud)

#### 3.1.1 텍스트 머드의 정의

머드게임은 인터넷상에서 채팅(Chatting)을 하던 사람들이, 유닉스(Unix)기반에서 텍스트 채팅을 이용하여 만든 다중 사용자(Multi-user)게임이다. 이는 대전형 머드게임과 사회형 머드게임으로 나뉘는데 현재 개발되는 많은 머드게임들은 사회형 머드게임의 시나리오를 채택하고 있으며, 첨단기술 네트워크 게임 역시 사회형 머드게임의 사회성을 기반으로 한다.

#### 3.1.2 텍스트 머드의 구조 및 실행 시나리오

모든 맵(Map)데이터 등의 자료구조들을 머드 서버에서 일관되게 관리하며, 프로세싱이 단 하나의 서버에서 일어나는 중앙 집중식 구조이다. 머드 서버와 머드 클라이언트가 통신을 통해 연결되어 있지만 각 사용자의 텔넷 프로그램에서 어떠한 프로세싱도 일어나지 않기 때문에 진정한 클라이언트 서버구조라고 보기는 어렵다.



(그림 3) 텍스트 머드의 구조

[운영시의 시나리오]

- 1- 텔넷 클라이언트는 사용자로부터 입력을 받아 머드 서버로 전송
- 2- 머드 서버는 프로세싱을 거쳐 디스플레이 데이터를 텔넷 클라이언트로 전송
- 3- 텔넷 클라이언트는 디스플레이 데이터를 사용자에게 출력

3.1.3 텍스트 머드의 특징

이러한 text mud는 여러 가지 이유로 설계, 구현, 운영상의 제한점을 가진다.

① 설계상의 제한점

- 전체적인 게임 벨런스를 고려하여 시나리오를 설계해야 한다.
- 원하는 시나리오대로 가상세계를 설계하는 map editing의 과정이 복잡하다.
- 가상세계의 구성이 World, Zone, Room으로 엄격히 나누어져야 한다.

② 구현상의 제한점

- 모든 게임자에게 공평한 time 체계를 제공해야 한다.
- 연성 실시간(Soft real time)의 개념을 지원해야 한다.

③ 운영상의 제한점

- Map이 compile time때 고정되어, 운영 중에는 map editing이 불가능하다. 이는 설계상의 제한 때문에 생기는 문제이다.
- Input과 display는 텍스트를 기반으로 이루어진다. 이와 같은 방식은 굉장히 다양한 명령어를 지원하고 자유스럽다. 하지만, 이러한 명령은 텍스트를 기반으로 하기 때문에 직관적이지 못하고, 일정한 문법에 기반하고 있기 때문에 익히기가 어려우며 복잡한 단점이 있다.
- 텍스트 머드의 구조를 분산의 개념에서 보았을 때, 모든 processing이 mud server에 집중되어 있으므로, bottle neck 등의 문제로 접속 가능한 사용자의 수가 제한될 수 있다. 이러한

문제는 구조를 바꾸지 않고서는 해결하기가 불가능하다.

3.1.4 Mud server와 Mud client간의 통신

Mud server와 Mud client간의 통신은 일반적으로 Unix에서 사용하는 telnet protocol을 사용한다. Telnet protocol은 연결 지향(connection oriented)이다. 텍스트 머드의 경우 request와 reply가 용량이 작고 빈번하게 일어나는 transaction의 성격을 띠고 있다. 일반적으로 크기가 큰 파일 전송 등에는 연결 지향 방식이, 크기가 작은 transaction 등의 경우에는 비연결 지향 방식이 적합하다고 말한다. 하지만, 텍스트 머드와 같이 요구가 빈번하게 일어나고 즉각적인 반응을 필요로 하는 경우에도 연결 지향 방식이 적합하다. 여기서 말하는 연결 지향은 session level의 연결 지향 방식이다.

[운영시의 시나리오]

- 1- Mud client는 사용자로부터 입력을 받아 Mud server로 전송
- 2- Mud server는 processing을 거쳐 control data를 mud client로 전송
- 3- Mud client는 받은 control data와 자신이 가진 display resource를 이용하여 사용자에게 출력

3.2 그래픽 머드(Graphic Mud)

3.2.1 그래픽 머드의 정의

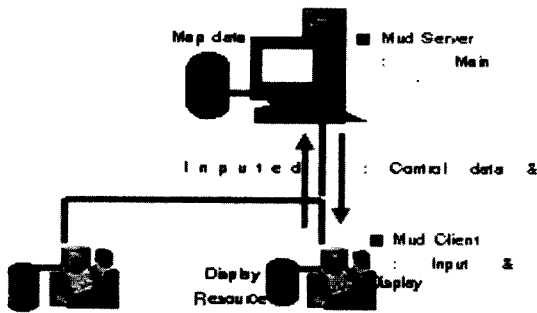
텍스트 머드가 가지는 표현의 한계 때문에 개발된 그래픽 머드는 그래픽 유저 인터페이스를 위해 각 사용자측에 그래픽자원을 예상되는 지연시간을 고려하여 미리 설치하고, 이를 그래픽을 사용하여 사용자 인터페이스를 처리한다. 여기서 그래픽 자원은 멀티미디어 데이터로서 그래픽 데이터뿐만 아니라, 애니메이션과 배경음악, 텍스트까지도 포함한다.

3.2.2 그래픽 머드의 실행 시나리오

Mud server와 Mud client사이에 주고받는 control data는, Mud server가 processing한 결과를 Mud client

에게 보여주기 위해 사용된다. Graphic mud이므로 graphic data 를 이용하여 display가 이루어져야 하는데, 네트워크 delay 때문에 직접 주고받지는 못하고 Mud client에게 미리 배포가 되어야 한다. Mud client는 가지고 있는 display resource를 이용하여, mud server에게 받은 control data를 인식하여 사용자에게 display 해주게 된다.

### 3.2.3 그래픽 머드의 구조



(그림 4) Graphic Mud의 구조

각 사용자의 mud client program에서 display를 하기 위한 processing이 일어나기 때문에, 이 경우 Two-tier Client /Server 방식이라고 볼 수도 있다. 하지만 이 경우 역시 많은 부분이 mud server에서 일어나며, mud client 부분은 I/O만을 담당하고 있다. Mud client에서 일어나는 processing은 단지 display를 위한 것이기 때문에 진정한 client / server 방식이라고 보기는 힘들다.

그래서, 이 경우에도 중앙 집중방식에 가까우며, 높은 효율을 위해서는 적절한 형태의 분산이 이루어져야만 한다. 또한 텍스트 머드와 같이, 머드를 진행하기 위한 모든 map data등의 자료 구조들 역시 mud server에서 일관되게 관리한다.

### 3.2.4 그래픽 머드의 특징

Mud Server가 텍스트 머드의 게임 엔진을 거의 수정 없이 사용하기 때문에, 앞에서 설명한 텍스트

머드가 가지는 특징/제한점을 그대로 물려받는다.

기존 텍스트 머드와 다른 점은 미리 배포되는 display resource를 이용하여 mud client가 display를 한다는 점 뿐이며, input 이나, processing 부분은 크게 다르지 않다.

이러한 이유로 텍스트 머드와 마찬가지로 구조상의 이유 때문에 같은 문제점이 발생 한다. 마찬가지로, 이러한 문제는 그래픽 머드의 구조적인 면에 기인하기 때문에 구조를 바꾸지 않고서는 해결하기가 불가능하다. 이 bottle neck의 문제와 processing의 중앙 집중 문제 때문에 이를 해결한 네트워크 RPG의 구조가 각광 받고 있다.

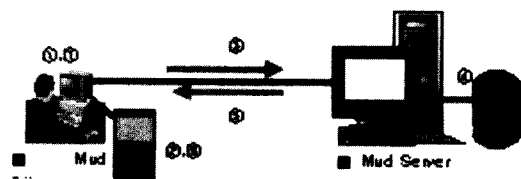
### 3.2.5 Mud server와 Mud client 간의 통신

텍스트 머드에서 client가 telnet program 으로 충분했던 것과는 달리, 그래픽 머드의 mud client는 display processing 때문에 mud server와 더불어 만들어져야 한다. 또한 Mud client와 mud server 사이의 protocol 역시 기존의 telnet protocol 대신 독자적인 protocol이 필요하다.

그래픽 머드를 위한 protocol은 텍스트 머드와 마찬가지로 연결 지향 방식이 적합하다. 그래픽 머드의 경우 역시 request와 reply가 용량이 작고 빈번하게 일어나는 transaction의 성격을 띠고 있으며, 요구가 빈번하게 일어나고 즉각적인 반응을 필요로 하는 경우에도 연결 지향 방식이 적합하므로, 연결 지향 방식의 protocol을 사용한다.

## 3.3 네트워크 RPG

### 3.3.1 네트워크 RPG의 실행 시나리오

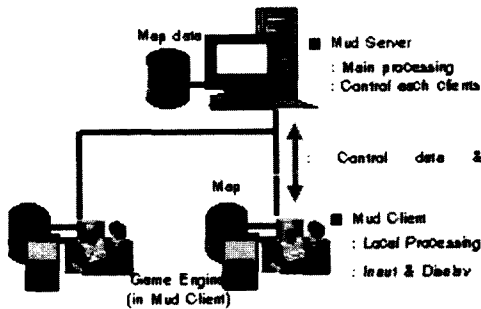


(그림 5) 네트워크 RPG의 실행 시나리오

Processing은 mud server와 mud client에서 적절하게 나누어져 이루어진다. 운영시의 시나리오를 살펴보면 (그림 5)와 같다.

- [운영 시나리오]
- 1- Mud client는 사용자로부터 입력을 받는다.
  - 2- 자신이 처리할 수 있는 요구이면 자신이 처리
  - 3- Mud server가 처리해야 할 요구이면 mud server에게 전송
  - 4- Mud server는 받은 요구에 대한 processing을 한다. Mud server가 처리해야 하는 대부분의 요구는, 여러 mud client간의 동기화를 위한 부분이거나, message 전달 등이 해당
  - 5- 사용자의 요구에 대한 processing의 결과를 다시 mud client에게 전송
  - 6- Mud client의 게임 engine은 control data의 형식으로 받은 processing의 결과를 이용하여 processing하게 된다.
  - 7- Mud client는 게임 engine에서 processing 한 결과를 사용자에게 적절한 형태로 보여준다

3.3.2 네트워크 RPG의 구조



(그림 6) 네트워크 RPG의 구조

네트워크 RPG의 구조는 진정한 형태의 Two-tier Client/Server 방식이라고 말할 수 있다. 네트워크 RPG의 client는 게임 engine에 기반 하여, I/O 뿐만 아니라 많은 processing을 하기 때문이다. 네트워크 RPG의 mud server와 mud client는 게임 진행을 위한 processing을 적절히 분산하여 나눠 가지고 있다. mud를 진행하기 위한 map data등의 자료구조 들은 mud server와 mud client가 각자 가지고 있게 된다.

3.3.3 네트워크 RPG의 특징

Mud server는 mud client간의 동기화, message 전달, 몇 가지 중요한 processing 등 만을 하기 때문에, bottle neck등의 문제는 상대적으로 덜 생기게 된다. 또한 그래픽 머드와 마찬가지로 input과 display는 전적으로 mud client에서 이루어진다.

일반적인 adventure 게임은 RPG에 기반하여 만들어졌기 때문에, 각 mud client들은 게임 engine을 내장하고 있으며, 이를 이용하여 일인용으로 게임을 즐기는게 가능하다. 이 같은 사항은 네트워크 RPG와 그래픽 머드의 큰 차이점 중의 하나이다.

Input과 display는 자체적인 게임 엔진을 이용하여 이루어진다. 텍스트 머드와 그래픽 머드와 같은 text command 방식의 명령 입력 대신 icon 등을 이용한 GUI를 이용하여 명령을 입력 하게 된다. 이와 같은 방식은 기존 adventure 게임/ RPG에서 사용되는 방법이며, 텍스트 머드와 같은 자유스러움은 없지만, 직관적이고 간단하며 사용이 편리하다.

3.3.4 Mud server와 Mud client간의 통신

그래픽 머드와 마찬가지로 mud client와 mud server 사이에는 나름대로 정한 적절한 protocol이 사용된다. 이 경우 텍스트 머드, 그래픽 머드와 마찬가지로 연결 지향 방식이 적합하다.

네트워크 RPG의 경우 역시 request와 reply가 용량이 작고 빈번하게 일어나는 transaction의 성격을 띠고 있으므로, 즉각적인 반응을 필요로 하는 연결 지향 방식이 적합하므로, 연결 지향방식의 protocol을 사용한다.

3.4 네트워크 기반 게임들 간의 비교

3.4.1 전반적인 비교

	Text Mud	Graphic Mud	네트워크 RPG
구조	중앙 집중형 Client / Server 구조	중앙 집중형 Client / Server 구조	분산형 Client / Server 구조
Map data	Server	Server	Server, Client

	게임 Engine	게임 Engine: Server	Server	Server, Client
	Display Resource	Client에 존재 않음	Client에 존재함	Client에 존재함 (게임 engine이 관리)
가상 세계의 설계방식	엄격한 World / Zone/ Room방식 (Scroll 불가능)	엄격한 World / Zone / Room방식(Scroll 불가능)	Scroll이 가능한 방식 (기존 RPG 방식)	
UI / Input	Text 입력방식	Text 입력방식	Icon 입력방식	
UI / Output	Text 이용	Graphic data 이용(Scroll 불가능)	Graphic data 이용(Scroll 가능)	
Bottleneck 문제	있을 수 있다.	있을 수 있다.	심각하지 않다.	
구현 / server와 client간의 통신 방식	연결 지향 방식(Telnet protocol 이용)	연결 지향 방식(독자적인 protocol 이용)	연결 지향 방식(독자적인 protocol 이용)	

#### 4. 결론

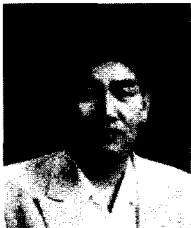
본 논문에서는, 다양한 형태의 네트워크 게임에 대해 연구하였으며 구현, 구조, 운영 측면까지 전반적이면서도 실제적인 내용을 소개하였다.

이러한 연구를 통하여 첨단 게임의 큰 축을 이룰 네트워크 게임에 대해 해석할 수 있다. 나아가 본 연구 결과는 네트워크 게임 (온라인, 아케이드, 모바일 등) 개발에 필요한 새로운 운영 체계 개발 시에 기반 연구로서 사용될 수 있다.

#### 참고문헌

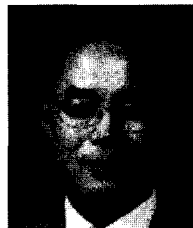
- [1] Rick D. Bess, MultiGen, Incealtime 게임 Development, 2000.
- [2] 시스템 공학 연구소, [2]의 기사 중 “게임 기술의 발전 양상”이라는 도표.
- [3] 신 동원, 문 제근, 김 형욱, “EM Mud 를 이용한 협동적이고 통합적인 주제별학습방안”, pp. 85-179, 정보통신부주최.
- [4] 아주대학교, “미래 첨단 게임의 흐름-네트워크 게임” 1998년 1월.
- [5] 한국 pc게임 개발사연합회, “게임백서”, 1997.
- [6] 아주대학교, “A study of Operating System for High-Tech 네트워크 게임”, 1998.
- [7] 한창호, “DirectX 게임프로그래밍”, PRESS, 1998.
- [8] 정상원, “게임 on the net :네트워크 게임의 세계로 가자”, 마이크로소프트웨어, 1997년 12.
- [9] 국내 온라인 게임 보고서, (사)한국첨단게임산업협회, 2000년도.
- [10] H. Hirabayashi, K. Akao, The University of Computer Gaming World, Media Factory, 1996.

## 저자약력



**박 철 제**

1986년 연세대학교 수학과 (이학사)  
1991년 일본 와세다대학교 정보과학전공 (이학석사)  
1996년 일본와세다대학교 정보과학전공 (공학박사)  
1996년-1998년 현대정보기술연구소 책임연구원  
1998년-현재 동의대학교 경영정보학과 조교수  
관심분야: 자연언어처리, 인터넷/인트라넷, 인공지능,  
에이전트  
e-mail : cjpark@dongeui.ac.kr



**최 성**

1976년-1994년 기업은행, 조선대학교 전자계산학과, 제  
주은행 전산실장, 한국생산성본부 OA  
추진사무국장 역임  
1983년 연세대학교 산업대학원 전자계산학과(공학석사)  
1999년 강원대학교 대학원 컴퓨터학과(이학박사)  
1994년-현재 남서울대학교 컴퓨터학과 교수, 한국정  
보기술전문가협회 회원담당이사, 한국첨  
단게임협회 지도교수, 영상물등급위원회  
게임분과심의위원  
관심분야: EC/ERP, DB/DC, HCI, 영상VR게임, 소프트  
웨어엔지니어링