



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

석사학위논문

위게임 모델의 소프트웨어 아키텍처
설계에 관한 연구

-상위레벨 계층구조를 기반으로 한 소프트웨어 구조를 중심으로-



한성대학교 국방과학대학원

국방 M & S 학과

국방 M & S 학전공

정 경 식

석사학위논문
지도교수 이동준

위게임 모델의 소프트웨어 아키텍처 설계에 관한 연구

—상위레벨 계층구조를 기반으로 한 소프트웨어 구조를 중심으로—

A Study on Wargame Model's Software Architecture Design
—Especially by Software Structure Based On The High Level
Hierarchical Structure—

2014년 12월 일

한성대학교 국방과학대학원

국방 M&S 학과

국방 M&S 학전공

정 경 식

석사학위논문
지도교수 이동준

위게임 모델의 소프트웨어 아키텍처 설계에 관한 연구

—상위레벨 계층구조를 기반으로 한 소프트웨어 구조를 중심으로—

A Study on Wargame Model's Software Architecture Design
—Especially by Software Structure Based On The High Level
Hierarchical Structure—

위 논문을 국방M&S학 석사학위 논문으로 제출함

2014년 12월 일

한성대학교 국방과학대학원

국방M&S학과

국방M&S학전공

정 경 식

정경식의 국방M&S학 석사학위논문을 인준함

2014년 12월 일



심사위원장 _____인

심사위원 _____인

심사위원 _____인

국 문 초 록

위게임 모델의 소프트웨어 아키텍처 설계에 관한 연구 -상위레벨 계층구조를 기반으로 한 소프트웨어 구조를 중심으로-

한성대학교 국방과학대학원

국방 M&S 학과

국방 M&S 전공

정 경 식

국내 위게임 모의모델은 1996년도부터 개발이 시작된 사·군단급 지휘관/참모 훈련모델인 창조21모델의 개발을 필두로 지금까지 합참, 육군, 해군, 공군 및 해병대에서 각 군의 연습/훈련지원, 획득 및 실험분석을 위해 지속적으로 위게임 모의모델을 개발하고 있으며, 전시작전권환수에 따라 한국군의 독자적인 전쟁 수행능력 향상을 위한 연습지원 및 실험분석 등을 위하여 신규 위게임 모의모델의 개발소요가 지속적으로 제기되고 있다.

위게임 모의모델은 모의규모에 따라 전구급, 임무급, 교전급 및 공학급으로 분류할 수 있으며, 또한 운용적인 측면에서 연습용 모델, 획득용 모델 및 실험분석용 모델로 구분할 수 있다. 위게임 모의모델에 활용되는 기술은 모델링 기술, 시뮬레이션 기술, M&S기반기술 등이다.

위게임 모의모델 개발간 최근에는 생산성 및 재사용성 향상을 위한 방법론인 국방CBD방법론을 적용하여 개발하고 있으나, 실제로는 모델별 특성에 따른 컴포넌트를 각각 개발 및 구현하여 모델을 개발함에 따라 생산성이 저하되고 또한 일부 컴포넌트는 중복하여 개발되고 있는 실정이다. 특히 전투모의엔진분야는 모든 위게임 모의모델에서 공통으로 사용하는 핵심 컴포넌트 분야로서, 전투모의엔진분야의 소프트웨어 아키텍처가 공동으로 사용될 수 있도록 표준화한다면 위게임 모의모델의 개발 생산성과 재사용은 매우 향상될 것이다.

소프트웨어 아키텍처의 표현은 구조와 뷰로 표현된다. 구조는 소프트웨어나 하드웨어에 존재하는 아키텍처 요소 그 자체이다. 뷰는 이해관계자에 의해

읽히고 쓰이는 아키텍처 요소의 일관된 집합으로 아키텍처 요소 집합과 그들 간의 관계로 이루어진다. 아키텍처 구조는 요소가 보여주는 의미에 따라 크게 모듈구조, 컴포넌트와 커넥터 구조 및 할당구조의 3가지 그룹으로 나누어진다.

위게임 모의모델의 시스템 구성을 계층구조로 분류하면 상위레벨 계층으로 데이터 관리, 모델링, 시뮬레이션, 데이터 전달 및 연동, 상황전시 및 명령입출력, 사후분석의 6개 계층으로 구분된다. 이러한 상위레벨 계층인 6개 계층을 기반으로 한 소프트웨어 아키텍처 구조를 제시한다. 소프트웨어 아키텍처는 모듈구조, 컴포넌트와 커넥터 구조 및 할당구조로 구분하여 설계안을 제시한다. 모듈구조는 분할구조, 사용구조, 계층구조, 클래스구조를 제시하되 분할구조는 상위레벨 계층인 6개 계층에 대한 각각의 상세 분할구조를 제시한다. 컴포넌트와 커넥터 구조는 클라이언트 서버 구조, 동시성 구조, 프로세스 구조와 공유 데이터 구조를 제시한다. 할당 구조는 업무할당 구조, 배치 구조, 구현 구조를 제시한다.

위게임 모의모델을 6개의 상위레벨 계층구조로 정의한 계층구조도와 이를 기반으로 한 소프트웨어 아키텍처는 위게임 모의모델 개발 간 시스템 분석 및 설계단계에서의 기간단축, 개발생산성과 재사용성 향상 및 위게임모델의 상호운용성 향상에 기여할 것이다.

【주요어】 위게임, 아키텍처 프레임워크, 소프트웨어 아키텍처, 재사용성, 표준화, 국방CBD방법론, War Game, Architecture Framework, Software Architecture, Reusability, Standardization, Military CBD Method

【 목 차 】

제 1 장 서 론.....	1
제 1 절 연구 배경 및 목적.....	1
제 2 절 연구 범위 및 방법.....	2
제 3 절 연구 제한사항.....	3
제 2 장 이론적 배경.....	4
제 1 절 M&S 및 위게임 모의모델 개념.....	4
제 2 절 아키텍처 개념.....	6
제 3 절 소프트웨어 아키텍처 개요.....	11
제 4 절 소프트웨어 아키텍처 설계 프로세스.....	16
제 5 절 소프트웨어 아키텍처 스타일.....	19
제 6 절 디자인 패턴.....	24
제 3 장 소프트웨어 아키텍처 설계를 위한 사례연구.....	29
제 1 절 창조21 모의모델 개발사례 연구	29
제 2 절 태극JOS 모의모델 개발사례 연구	32

제 4 장	위게임 모의모델 소프트웨어 아키텍처 설계	34
제 1 절	소프트웨어 아키텍처 설계를 위한 아키텍처 구조 정립	34
1.	모듈(Module) 구조	34
2.	컴포넌트와 커넥터(Component and Connector) 구조	36
3.	할당 (Allocation) 구조	37
제 2 절	위게임 모의모델 상위레벨 계층구조 설계	38
제 3 절	모듈 구조	41
1.	분할(Decomposition) 구조	41
2.	사용(Users) 구조	48
3.	계층(Layered) 구조	49
4.	클래스 구조	50
제 4 절	컴포넌트와 커넥터 구조	51
1.	클라이언트 서버 구조	51
2.	동시성 구조	52
3.	프로세스 구조	53
4.	공유 데이터 구조	54
제 5 절	할당 구조	55
1.	업무할당 (Work Assignment) 구조	55
2.	배치 (Deployment) 구조	56

3. 구현 (Implement) 구조	57
제 6 절 위게임 모의모델의 소프트웨어 아키텍처 설계 기대효과.....	58
제 5 장 결 론.....	59
【참고문헌】	60
ABSTRACT	61



【 표 목 차 】

<표 2-1> 소프트웨어 아키텍처의 4+1 뷰	15
<표 2-2> 디자인 패턴 영역	24
<표 4-3> 창조21 및 태극JOS 모델을 통한 공통 계층구조 분석	39



【 그림 목 차 】

<그림 2-1> 무기체계 및 인간의 참여에 따른 시뮬레이션 구분	5
<그림 2-2> 아키텍처 구성요소	6
<그림 2-3> 전사 아키텍처 프레임워크 구성 예	8
<그림 2-4> 전사 아키텍처와 소프트웨어 아키텍처와의 관계	11
<그림 2-5> 개발단계에서의 소프트웨어 아키텍처 위치	12
<그림 2-6> 소프트웨어 아키텍처 설계 프로세스	16
<그림 2-7> 레파지토리 스타일	20
<그림 2-8> 모델 뷰 컨트롤러 아키텍처 스타일	20
<그림 2-9> 클라이언트 서버 스타일	21
<그림 2-10> 계층 스타일	22
<그림 2-11> 파이프 필터 스타일	23
<그림 3-1> 창조21 모의모델 전장기능	29
<그림 3-2> 창조21 모델 시스템 구성도	31
<그림 3-3> 태극JOS 모델 시스템 구성도	33
<그림 4-1> 소프트웨어 아키텍처 구조	34
<그림 4-2> 위게임 모의모델 상위레벨 계층구조	40
<그림 4-3> 위게임 모의모델 분할구조	41
<그림 4-4> 위게임 모의모델 데이터 상세분할 구조	42
<그림 4-5> 위게임 모의모델 모델링 상세 분할 구조	43
<그림 4-6> 위게임 모의모델 시뮬레이션 상세 분할 구조	44
<그림 4-7> 위게임 모의모델 데이터 전달 상세 분할 구조	45
<그림 4-8> 위게임 모의모델 상황전시 및 명령입출력 상세 분할 구조	46
<그림 4-9> 사후분석 상세 분할 구조	47
<그림 4-10> 위게임 모의모델 사용 구조	48
<그림 4-11> 위게임 모의모델 계층구조	49
<그림 4-12> 위게임 모의모델 클래스 구조	50
<그림 4-13> 위게임 모의모델 클라이언트 서버 구조	51

<그림 4-14> 위게임 모의모델 동시성 구조	52
<그림 4-15> 위게임 모의모델 프로세스 구조	53
<그림 4-16> 위게임 모의모델 공유 데이터 구조	54
<그림 4-17> 위게임 모의모델 업무 할당 구조	55
<그림 4-18> 위게임 모의모델 배치 구조	56
<그림 4-19> 위게임 모의모델 구현 구조	57



제 1 장 서 론

제 1 절 연구 배경 및 목적

모델링 및 시뮬레이션(M&S, Modeling & Simulation)은 모델링과 시뮬레이션의 합성어로서, 모델링은 수학적 모델, 물리적 모델 및 과학적 모델로 구분할 수 있으며, 시뮬레이션은 실제(Live) 시뮬레이션, 가상(Virtual) 시뮬레이션 및 구성(Constructive) 시뮬레이션으로 구분할 수 있다. 모델링 및 시뮬레이션을 국방분야에 활용하는 목적은 본질적으로 미래를 예측하고자 하는 것이 아니라 결심권자로 하여금 결심을 할 수 있도록 통찰력을 제공하는 것이다. 특별히 구성 시뮬레이션이라 할 수 있는 위게임 모의모델을 통해 인류 역사 이래 끊임없이 반복되어 온 전쟁이라는 실제 체계에 대한 충분한 이해 및 분석과 더불어 이에 적절히 대비하고자 하는데 그 목적이 있다. 모델링 및 시뮬레이션을 활용하는 이유나 그 가치는 시뮬레이션이 과학적이고, 합리적이며, 시스템적인 사고의 수단이라는 것이다.¹⁾

위게임 모의모델은 저비용 고효율의 군사력 건설 및 운용의 필수수단으로 인식하고 있으며 그 활용성이 점차 증대하고 있으며, 그에 따라 교육훈련, 분석, 획득을 위한 과학화된 위게임 모의모델 수요가 점차로 증가하고 있는 실정이다. 국내 위게임 모의모델은 1996년도부터 개발이 시작된 사·군단급 지휘관 /참모 훈련모델인 창조21모델을 필두로 지금까지 합참, 육군, 해군, 공군 및 해병대에서 각 군의 연습/훈련지원, 획득 및 실험분석을 위해 지속적으로 위게임 모의모델을 개발하고 있으며, 특히 전시작전권환수에 따라 한국군의 독자적인 전쟁 수행능력 향상을 위한 연습지원 및 실험분석 등을 위하여 신규 위게임 모의모델의 개발수요가 지속적으로 제기되고 있다.

미국의 경우를 살펴보면 미 국방성은 모델링 및 시뮬레이션 분야에 있어서

1) 모델링 및 시뮬레이션 이론과 실제, 21세기군사연구소, 2008.6.20, p44

상호운용성과 재사용성 향상을 통하여 보다 경제적이고, 효율적으로 다양한 작전적 요구사항에 대해 융통성 있게 가상 전투공간을 구성하여, 대규모 부대 훈련뿐만이 아니라, 분석, 획득, 임무수행 예행연습 등 국방업무 분야에 광범위하게 활용하도록 한다는 것이다.²⁾

그러나 우리 군이 개발한 위게임 모의모델 현황을 살펴보면 모델별 특성에 따른 컴포넌트를 각각 개발함에 따라 중복 개발 및 생산성이 저하되고 있다. 또한 재사용가능 소프트웨어 및 산출물이 미약하여 유사기능의 중복개발 등 생산성과 재사용성 측면에서 매우 미흡한 실정이다.

따라서 개발 생산성 및 재사용성 향상을 위하여 위게임모의모델의 계층구조를 정립하고 이를 기반으로 한 소프트웨어 아키텍처를 설계하여 제시하고자 한다.

제 2 절 연구 범위 및 방법

위게임 모의모델 개발 생산성 및 재사용성 향상을 위한 소프트웨어 아키텍처 설계연구는 연습/훈련 모델의 시뮬레이션 소프트웨어 아키텍처를 중심으로 하되 연구 범위는 이론적 배경, 소프트웨어 아키텍처 설계를 위한 선행연구, 위게임 모의모델 소프트웨어 아키텍처 설계 및 결론으로 각 장을 편성하여 연구한다.

이론적 배경에서는 M&S 및 위게임 모의모델 개념에 대하여 살펴보고 아키텍처 개념, 소프트웨어 아키텍처 개요, 소프트웨어 아키텍처 설계 프로세스, 소프트웨어 아키텍처 구조, 소프트웨어 아키텍처 스타일, 디자인 패턴에 대하여 연구한다. 소프트웨어 아키텍처 설계를 위한 사례연구에서는 상위레벨 계층구조 정립을 위하여 창조21 모의모델과 태극JOS 모의모델의 개발 사례를 분석한다.

위게임 모의모델 소프트웨어 아키텍처 설계에서는 위게임 모의모델 시스템

2) 전게서, p207

구성 요소를 계층화 하여 파악하고 각 시스템 계층별로 소프트웨어 아키텍처 구조를 연구한다. 소프트웨어 아키텍처 구조는 모듈구조, 컴포넌트와 커넥터 구조 및 할당구조로 구분하여 연구한다. 모듈구조는 분할구조, 사용구조, 계층구조 및 클래스 구조로 구분하여 연구한다. 컴포넌트와 커넥터 구조는 클라이언트 서버 구조, 동시성 구조, 프로세스 구조 및 공유데이터 구조로 구분하여 연구한다. 할당 구조는 업무할당 구조와 배치 구조 및 구현 구조로 구분하여 연구한다. 끝으로 소프트웨어 아키텍처 설계에 대한 기대효과를 제시하고 결론을 맺는 것을 본 논문의 연구 범위 및 방법으로 제시한다.

제 3 절 연구 제한사항

구성(Constructive) 시뮬레이션에 해당하는 위게임 모의모델은 전구급, 임무급, 교전급 및 공학급으로 구분할 수 있으며, 그 사용용도에 따라서 연습/훈련용, 실험분석용 및 획득용으로 구분할 수 있다. 본 연구에서는 구성(Constructive) 시뮬레이션의 위게임 모의모델 중 연습/훈련용 모의모델에 한정하여 분석토록 하며, 연습/훈련용 위게임 모의모델의 시스템 구성요소에 대한 소프트웨어 아키텍처를 설계하고자 한다. 소프트웨어 아키텍처의 구성요소에는 계획, 규칙, 모델의 3가지가 있으나 본 논문에서는 모델분야의 어플리케이션 아키텍처를 대상으로 하되, 소프트웨어 아키텍처 구조를 6개의 상위레벨 계층구조인 데이터, 모델링, 시뮬레이션, 데이터전달 및 연동, 상황전시 및 명령입출력, 사후분석으로 구분하여 제시한다. 소프트웨어 아키텍처의 표현은 구조와 뷰로 표현될 수 있다, 본 논문에서는 구조에 해당하는 모듈구조, 컴포넌트와 커넥터 구조 및 할당구조만을 연구하여 제시한다. 모듈구조는 분할구조, 사용구조, 계층구조 및 클래스 구조로 구분하고, 컴포넌트와 커넥터 구조는 클라이언트 서버 구조, 동시성 구조, 프로세스 구조 및 공유데이터 구조로 구분하며, 할당 구조는 업무할당 구조와 배치 구조 및 구현 구조로 구분하여 제시한다.

제 2 장 이론적 배경

제 1 절 M&S 및 위게임 모의모델 개념

M&S란 모델링과 시뮬레이션의 합성어로서 기존 위게임 영역에서 그 영역을 대폭 확대하여 국방기획관리 소요제기·결정, 획득, 분석평가는 물론 군의 교육훈련까지를 과학적으로 지원하는 도구 및 수단의 총칭이다.

모델링(Modeling)이란 “군사작전(전투)을 모의하기 위하여 적과 아군부대 및 전투체계, 작전지역을 포함한 전장환경, 자연 및 인공현상과 전술교리 등 전투에 영향을 미치는 각종 요소와 모의절차, 과정 등을 물리적·수학적·논리적 표현으로 만들어 나가는 과정”³⁾을 말하며, 모델은 모델링의 결과물이다. 모델은 시뮬레이션 대상체계의 성질 또는 특성을 표현하는 방법에 따라 수학적모델(Mathematical Model), 물리모델(Physical Model), 과정모델(Process Model)로 구분된다.

시뮬레이션(Simulation)란 “모델링의 산출물인 모델을 활용하여 연속적인 시간의 흐름 속에서 군사작전(전투)을 실제와 유사하게 실행하는 것”⁴⁾으로 복잡한 군사작전과정을 간단한 수치적 물리적 모델로 표현하고 그 결과를 계산적으로 처리하는 기법이다. 시뮬레이션은 시뮬레이션 시간이 연속적으로 진행되면서 시뮬레이션 되는 연속 시뮬레이션(Continuous Simulation)과 이산시점에 따라 시뮬레이션 되는 이산 시뮬레이션(Discrete Simulation)으로 구분할 수 있으며, 이산 시뮬레이션은 시간 중심(Time Driven)과 사건 중심(Event Driven)으로 구분할 수 있는 바, 시간 중심 시뮬레이션은 이산 시점의 간격이 일정한 반면 사건 중심은 이산 시점을 사건발생 시점에 기준하여 시간 간격이 비동기적으로 발생하는 시뮬레이션이다. 시뮬레이션은 대상체계 참여에 따라 <그림 2-1> 무기체계 및 인간의 참여에 따른 시뮬레이션 구분에서 보는 것처럼 실제 인간과 실제 무기체계가 참여하는 실기동 시뮬레이션(Live Simulation), 실제 인간과 모의 무기체계가 참여하는

3) 국방 위게임모델 목록집, 합동참모본부, 2010.1.22, p3

4) 상계서, p3

가상 시뮬레이션(Virtual Simulation), 모의 인간과 모의 무기체계가 참여하는 구성 시뮬레이션(Constructive Simulation) 및 모의 인간 및 실제 무기체계가 참여하는 스마트 시뮬레이션(Smart Simulation)으로 구분할 수 있다.



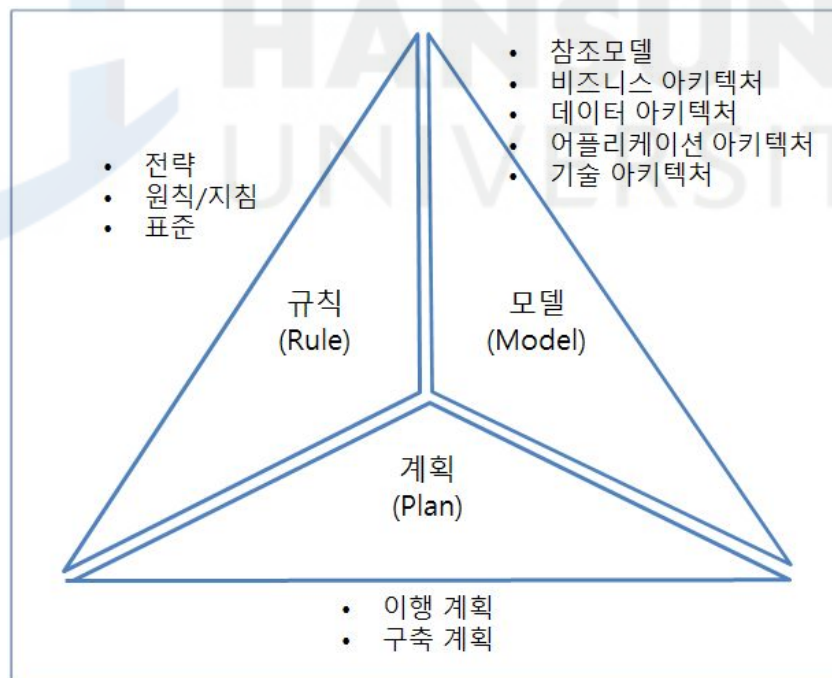
<그림 2-1> 무기체계 및 인간의 참여에 따른 시뮬레이션 구분

위게임이란 Live-Virtual-Constructive 시뮬레이션 중에서 특히 Constructive 시뮬레이션에 해당되며, 실제 또는 가상 군사상황을 묘사하기 위하여 둘 또는 그 이상의 적대세력 간에 발생하는 군사적 상황을 게임규칙, 각종 자료 및 절차를 사용하여 군사작전을 모의하는 것으로, 이런 맥락에서 군사적으로 활용되는 모델을 ‘위게임 모의모델’로 지칭한다. 위게임 모의모델은 전구/전역급 모델(Theater/Campaign Model), 임무/전투급 모델(Mission/Battle Model), 교전급 모델(Engagement Model) 및 공학급 모델(Engineering Model)로 구분할 수 있다. 전구급 모델은 다양한 전력이 개입된 전쟁을 모의하며, 장기적 관점에서 전역 및 전쟁수준의 분쟁 결과를 분석하는 데 사용된다. 임무/전투급 모델은 시간단위로 진행되는 공중우세, 차단 또는 강습 등과 같이 특정임무가 목표를 달성하기 위해 운용하는 다중 플랫폼 전력 패키지의 능력을 모의한다. 교전급 모델은 특정 표적이나 적 대상 위협 무기체계에 대한 개발 무기체계의 효과도를 평가하는데 사용되며, 공학급 모델은 체계나 부체계 및 구성품의 성능, 비용, 군수지원성 등을 분석하거나 이들 요소 사이의 상쇄분석 시 사용된다.

제 2 절 아키텍처 개념

1. 아키텍처 정의 및 구성요소

아키텍처란 고대 그리스어에서 건축 혹은 석공명인(Master)을 의미하는 'Architecton'이라는 용어에서 유래되었다. 당시의 아키텍처는 건축물의 골격을 제공하는 설계도 역할을 했다. 좋은 아키텍처는 훌륭한 건축물을 탄생시켰고, 이들은 아직까지 인류의 훌륭한 유산으로 남겨지고 있다. 고대의 아키텍처 개념이 화강암과 대리석으로 건축물을 짓는데 적용되었다면, 산업시대에는 건축뿐만 아니라 첨단 과학기술을 이용하여 항공기, 자동차, 선박 등을 개발하는데 적용되어 왔다. 또한 오늘날 디지털 정보화 시대에는 첨단 정보기술을 이용하여 정보체계, 소프트웨어 내장형 체계, 지휘통제 통신체계 등을 구축하는데 적용되고 있다.⁵⁾



<그림 2-2> 아키텍처 구성요소

5) 데이터아키텍처 전문가 가이드, 한국데이터베이스진흥센터, 2006, p23

IEEE Std. 1971에서는 아키텍처를 “구성요소의 구조, 구성요소들 사이의 관계, 구성요소의 설계 그리고 시간경과에 따른 구성요소의 발전을 위한 원리와 지침”으로 정의하고 있으며, 데이터아키텍처 전문가 가이드에서는 “구축하고자 하는 목적에 따라 복잡한 대상을 단순하게 표현하고, 구성요소의 변화에 대한 요구를 수용할 수 있게 한 청사진(Blueprint)”으로 정의하고 있다.

아키텍처 구성요소는 <그림 2-2>에서 보는 것처럼 규칙관점, 모델관점 및 계획관점으로 구분할 수 있다. 규칙관점의 구성요소는 전략, 원칙/지침, 표준이 있으며, 모델관점은 참조모델, 비즈니스 아키텍처, 데이터 아키텍처, 어플리케이션 아키텍처 및 기술 아키텍처로 구성되어 있으며, 계획 관점은 이행 계획, 구축 계획으로 구성되어 있다.

2. 아키텍처 종류

아키텍처의 종류에는 전사 아키텍처, 시스템 아키텍처, 소프트웨어 아키텍처 및 데이터 아키텍처로 구분할 수 있다.

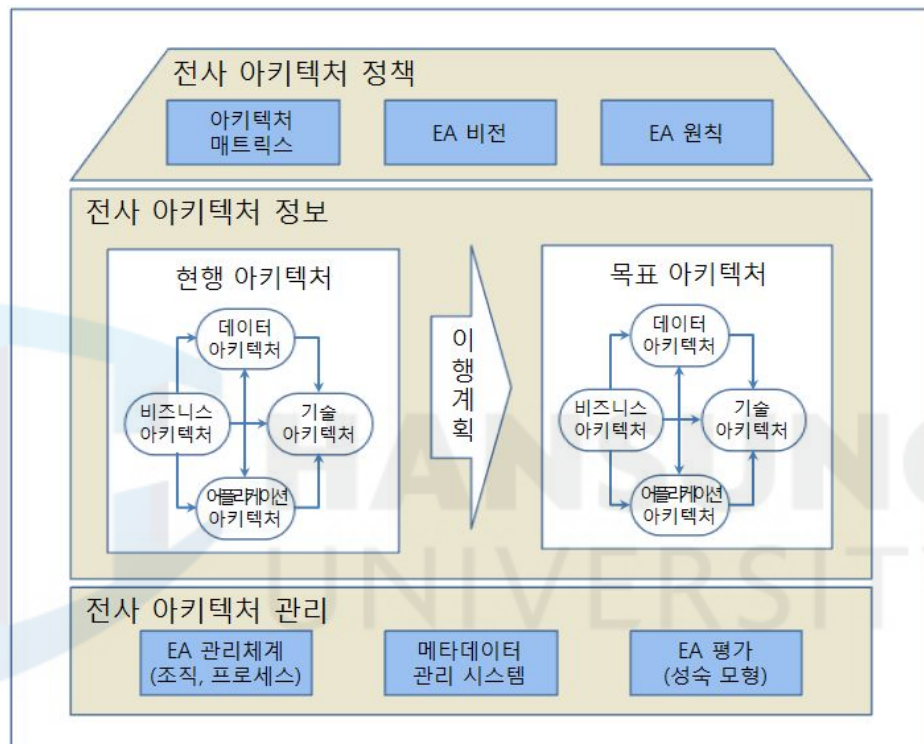
가) 전사 아키텍처 (Enterprise Architecture)

전사(Enterprise)란 “공동의 목표를 추구하기 위해 고객과 상품 또는 서비스가 존재하고 이를 지원하기 위한 조직, 자원, 기술을 보유하며 필요한 업무 프로세스를 수행하는 조직의 집합체”라 정의하고 있으며, 전사 아키텍처는 “기업의 목표와 요구를 잘 지원하기 위해 IT 인프라의 각 부분들이 어떻게 구성되고 작동되어야 하는가를 체계적으로 기술하는 것”이라 정의하고 있다.⁶⁾ 전사 아키텍처는 복잡한 기업의 모습을 다양한 측면 즉 비즈니스, 데이터, 어플리케이션, 기술 등에서 분석하고 이를 표현하여 이해하기 쉽도록 관련 정보체계를 구축하고 활용하는 것이다.

전사 아키텍처를 정의하여 관리하기 위해서는 우선 전사 아키텍처를 어떻게 표현하고 운용할 것인가에 대한 전체적인 사고의 틀을 결정해야

6) 전계서, p23

하는데, 이는 전사 아키텍처 활동에서 얻어지는 산출물을 분류하고 조직화하고 이를 유지 관리하기 위한 전체적인 틀을 정의하는 전사 아키텍처 프레임워크로 정의한다. 전사 아키텍처 프레임워크는 전사 아키텍처 정책, 전사 아키텍처 정보, 전사 아키텍처 관리 등의 3가지 영역으로 구분된다. 전사 아키텍처 프레임워크의 3가지 영역을 기반으로 한 프레임워크 구성 예는 <그림 2-3>7)과 같다.



<그림 2-3> 전사 아키텍처 프레임워크 구성 예

나) 시스템 아키텍처 (System Architecture)

시스템 아키텍처란 광의의 정의로는 어떤 분야의 산업을 유지 및 운용하기 위한 기반 산업을 총칭하는 용어이며, 협의의 정의로는 컴퓨터와 사용자를 연결하는데 사용되는 물리적인 하드웨어 및 소프트웨어의 통칭으로 정의할 수 있다. 따라서 시스템 아키텍처란 “경영 전략의 달성을 위해 필요한 업무 프로세스의 원활한 지원과 효율적 처리를 위해 필요한 하드웨어, 시스템

7) 전계서, p83

소프트웨어, DBMS, 네트워크 및 보안으로 구성된 전산 시스템의 기반이 되는 환경”으로 정의할 수 있다. 시스템 아키텍처는 User와 컴퓨팅 시스템과의 데이터 흐름 및 인터페이스까지도 정의한다.

다) 소프트웨어 아키텍처 (Software Architecture)

“프로젝트에서 설계근거가 포함된 아키텍처를 확보하지 않고 본격적인 개발에 돌입해서는 안 된다. 아키텍처를 산출물로 만들어 놓으면 개발은 물론 유지보수 단계까지 유용하게 사용할 수 있다”고 베리 보엠은 말했다.⁸⁾ 이렇듯 아키텍처는 프로젝트에 도움을 줄 뿐만 아니라 조직에 가치를 부여하는 자산이기도 하다.

소프트웨어 아키텍처 이론과 실제의 소프트웨어 아키텍처 정의를 살펴보면 소프트웨어 아키텍처는 “소프트웨어 구성요소와 그들이 지니고 있는 특성 중에서 외부에 드러나는 특성, 그리고 구성요소들의 관계를 표현하는 시스템의 구조나 구조체”라고 정의하고 있다.⁹⁾ 또한 소프트웨어 아키텍처는 ‘상위 수준의 설계이다’라고 할 수 있으며, ‘프로그램이나 시스템을 구성하고 있는 컴포넌트들의 구조, 그들 간의 상관관계, 시스템 설계를 통제하고 향후 진화에 영향을 주는 원칙이며 지침’이라고도 정의 할 수 있다.

소프트웨어 아키텍처는 고객, 사용자, 프로젝트 관리자, 개발자, 시험자 등 관련 이해관계자와의 대화 수단이며, 개발의 제약사항을 정하며, 개발 구조를 결정하고, 시스템의 품질속성을 장려하거나 억제하는 등 초기 설계의 결정사항이고, 프로젝트 초기에 아키텍처를 결정하는 요구사항과 아키텍처 그 자체 및 아키텍처 개발 경험을 재사용 할 수 있도록 타 시스템에 적용이 가능한 시스템의 추상화를 결정하기 때문에 기술적 측면에서 매우 중요하다.

소프트웨어 아키텍처는 아키텍처 구조와 뷰로 설계되고 문서화된다. 아키텍처 구조는 요소가 보여주는 의미에 따라 모듈 구조(Module Structures), 컴포넌트와 커넥터 구조(Component and Connector Structures) 및 할당 구조(Allocation Structures)로 구분되며, View는 논리적 뷰, 프로세스 뷰, 개발 뷰 및 배치 뷰로서 구분 할 수 있다. 각 구조 및 뷰에

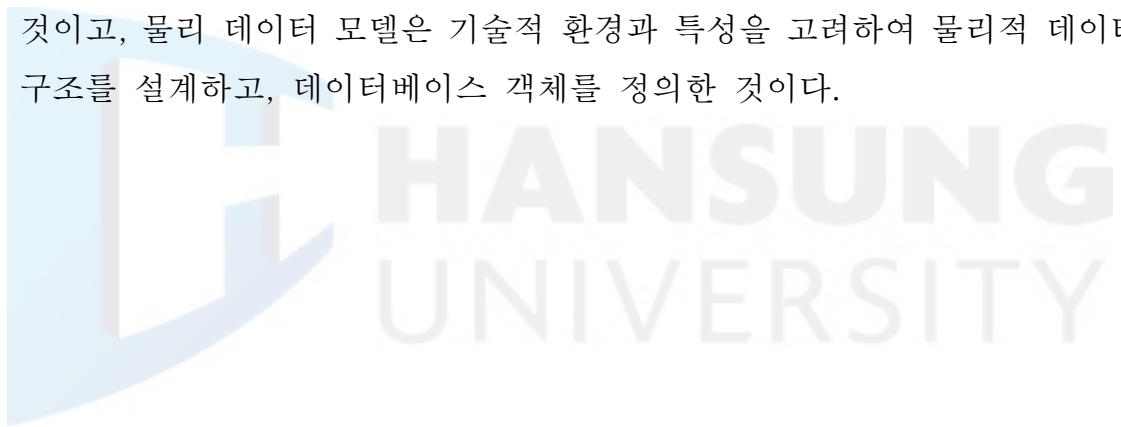
8) 소프트웨어 아키텍처 이론과 실제, 렌 베스 외, 2008, p19

9) 상계서, p21

대한 사항은 제 3 절 소프트웨어 아키텍처 개요에서 자세히 설명한다.

라) 데이터 아키텍처 (Data Architecture)

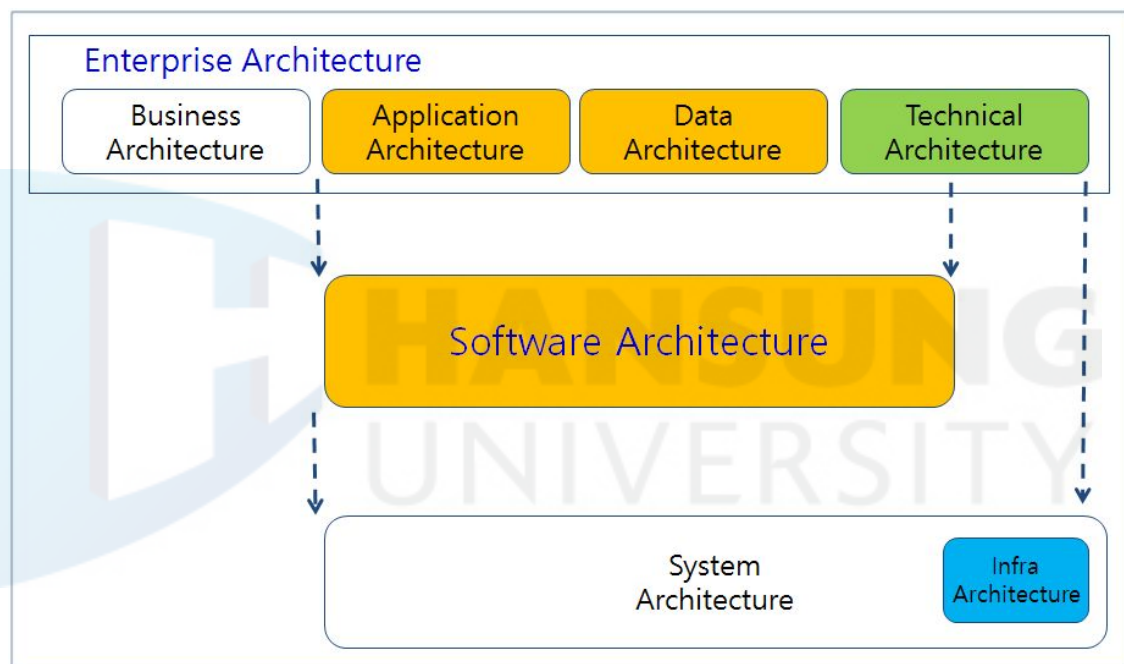
데이터 아키텍처는 기업의 업무 수행에 필요한 데이터의 구조를 체계적으로 정의하는 것으로, 전사의 데이터 영역을 분류하는데, 업무 데이터와 메타데이터를 구분하거나 운영계 데이터와 정보계 데이터 등으로 구분한다. 이를 기준으로 전사 수준의 주제 영역 모델, 개념 데이터 모델을 정의하고 영역별로는 논리 데이터 모델, 물리 데이터 모델을 정의한다. 전사 수준의 주제 영역 모델은 개괄 데이터 모델이라고도 하며 상위 수준의 전사 데이터 영역을 분류하고 표현한 것이며, 개념 데이터 모델은 단위 주제 영역 또는 핵심 엔터티 정도를 표현한 데이터 모델이다. 논리 데이터 모델은 업무 요건을 충족시키기 위한 데이터의 상세한 구조를 논리적으로 구체화한 것이고, 물리 데이터 모델은 기술적 환경과 특성을 고려하여 물리적 데이터 구조를 설계하고, 데이터베이스 객체를 정의한 것이다.



제 3 절 소프트웨어 아키텍처 개요

1. 전사 아키텍처와 소프트웨어 아키텍처와의 관계

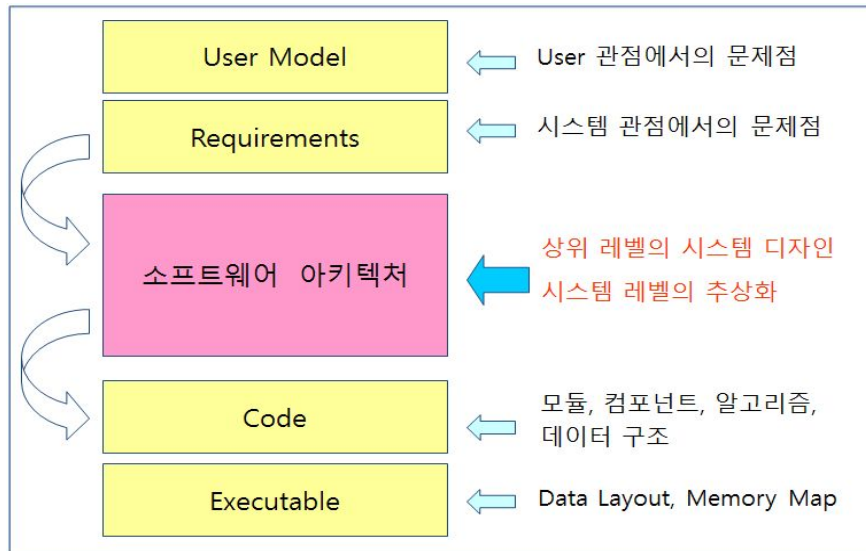
소프트웨어 아키텍처는 전사 아키텍처의 어플리케이션 아키텍처와 데이터 아키텍처와 관계있으며, 기술 아키텍처와도 일부 관련성이 있다. 전사 아키텍처와 소프트웨어 아키텍처 및 시스템 아키텍처와의 관련성을 그림으로 표시하면 아래와 같다.



<그림 2-4> 전사 아키텍처와 소프트웨어 아키텍처와의 관계

2. 개발단계에서의 소프트웨어 아키텍처 위치

소프트웨어 아키텍처는 개발단계에서 요구사항 분석단계와 코드 구현 단계의 중간 즉 기본설계와 상세설계 단계에 위치하게 된다.



<그림 2-5> 개발단계에서의 소프트웨어 아키텍처 위치

3. 소프트웨어 아키텍처란?

소프트웨어 아키텍처는 “프로그램이나 컴퓨팅 시스템에서 소프트웨어 구성요소와 그들이 가진 특성 중에 외부에 드러나는 요소의 특성, 그리고 구성요소들간의 관계를 표현하는 시스템의 구조나 구조체”를 말하는 것으로서 설계 프로세스에서 중요한 위치를 차지한다. 소프트웨어 아키텍처는 복잡한 대규모 시스템의 구조를 표현한다. 시스템을 바라보는 아키텍처 관점은 매우 추상적이어서 구현 시의 상세 내용이나 알고리즘, 데이터 모델링 등은 구체적으로 표현하지 않는다. 아키텍처 관점은 단지 블랙박스로 표시되는 구성요소 간의 동작과 상호작용을 표현하는데 주력한다. 소프트웨어 아키텍처는 시스템에 요구되는 특성을 설계하는 것으로서 시스템 개발의 첫 단계에서 수행된다.

조직의 목표에서 요구사항이 나오고, 요구사항에 따라 아키텍처가 설계되고, 아키텍처를 통해 시스템이 만들어진다.

아키텍처는 고객, 개발조직 관리자, 마케팅 담당자, 사용자, 유지보수 조직 관리자 등 많은 이해관계자로부터 영향을 받는다. 또한 개발조직의 특성이나 구조, 아키텍트의 배경, 경험, 지식, 기술적 환경 등으로 부터도 영향을 받는다.

4. 시스템 구조

시스템 구조는 정적구조와 동적구조로 구분할 수 있다.¹⁰⁾

정적구조는 설계 시 내부 요소 및 배열을 정의한다. 요소로는 모듈, 객체 지향 클래스, 서비스 등이 될 수 있다. 내부 데이터 요소로는 클래스, 관계형 데이터 베이스 개체 및 테이블, 데이터 파일 등을 포함하고 있다. 내부 하드웨어 요소는 컴퓨터나 컴퓨터 부품, 네트워크 요소 등을 포함하고 있다. 이런 요소들의 배열은 문맥에 따라 달라지는데, 문맥은 요소 간의 연결, 관계 등이 될 수 있다.

동적구조는 시스템이 실제로 어떻게 동작하는지를 보여준다. 즉, 실시간 요소와 요소들간의 상호 연동을 정의하고 있다. 이런 연동은 병렬 혹은 순차적인 관계로 표현될 수 있다.

5. 아키텍처 요소

아키텍처 요소란 시스템을 구성할 때 고려할 수 있는 기본적인 부분이다. 아키텍처 요소는 다음과 같은 주요 속성을 반드시 갖고 있어야 한다.

- 명확히 정의된 역할의 집합
- 명확히 정의된 경계의 집합
- 명확히 정의된 인터페이스의 집합

6. 이해 당사자

소프트웨어 시스템에 영향을 받는 사람은 사용에 국한되지 않는다. 사람들은 소프트웨어 시스템을 만들고, 테스트하며, 고치고, 보완하며, 이에 필요한 금액을 지불한다. 즉, 소프트웨어 아키텍처에서 이해 당사자는 아키텍처 실현화에 대해 관심 있는 사람, 그룹, 혹은 개체를 의미한다. 좋은 이해 당사자의 속성을 다음과 같은 것을 포함한다.

10) 소프트웨어 아키텍처 기반 설계 모델 및 명세기법 개발, 한국정보보호진흥원, 2007.12

- 좋은 의사 결정을 하기 위해 정보가 충분히 제공된다.
- 건설적인 방향으로 처리나 선택이 가능하도록 자발적으로 참여해야 한다.
- 의사 결정을 하기 위해 권한을 받아야 한다.
- 이해 당사자 그룹이 대표적이어서 그들의 관점이 유효하다는 것을 보여야 한다.

7. 아키텍처 명세서

아키텍처 명세서는 이해 당사자가 이해할 수 있고, 아키텍처가 이해 당사자의 관심을 확실하게 보일 수 있는 방법으로 아키텍처를 문서화해 놓은 산출물들의 집합이다. 여기서 산출물의 범위에는 아키텍처 모델, 범위 정의, 제약사항, 원칙이 포함될 수 있다. 아키텍처 명세서는 아키텍처의 핵심과 그에 관한 자세한 내용을 표현해야 한다. 비록 모든 시스템이 아키텍처를 갖고 있지만, 모든 시스템이 좋은 아키텍처 명세서를 포함하고 있는 것은 아니다. 여기서 좋은 아키텍처 명세서란 효율적이고 일관성 있게 아키텍처의 핵심 측면을 적절한 이해 당사자에게 전달할 수 있도록 기술된 것을 의미한다.

8. 소프트웨어 아키텍처 뷰와 뷰타입

아키텍처 뷰는 중점적으로 언급하고자 하는 관점으로 연관있는 아키텍처의 측면이나 요소를 기술하는 방법이다. 그러나 하나의 모델에서 전체 아키텍처의 핵심과 그에 대한 자세한 내용을 담기는 어렵다. 그러므로 래셔널 소프트웨어 사에서 발표한 것처럼 4+1 뷰 모델과 같은 뷰 포인트가 필요하다. 4+1 뷰는 논리적 뷰(Logical View), 프로세스 뷰(Process View), 개발 뷰(Implementation View), 물리적 뷰(Physical View) 및 유즈 케이스 뷰(Use case View)로 구분할 수 있다.¹¹⁾

11) 소프트웨어 아키텍처 문서화, 폴 클레멘츠 외, 2009, p16

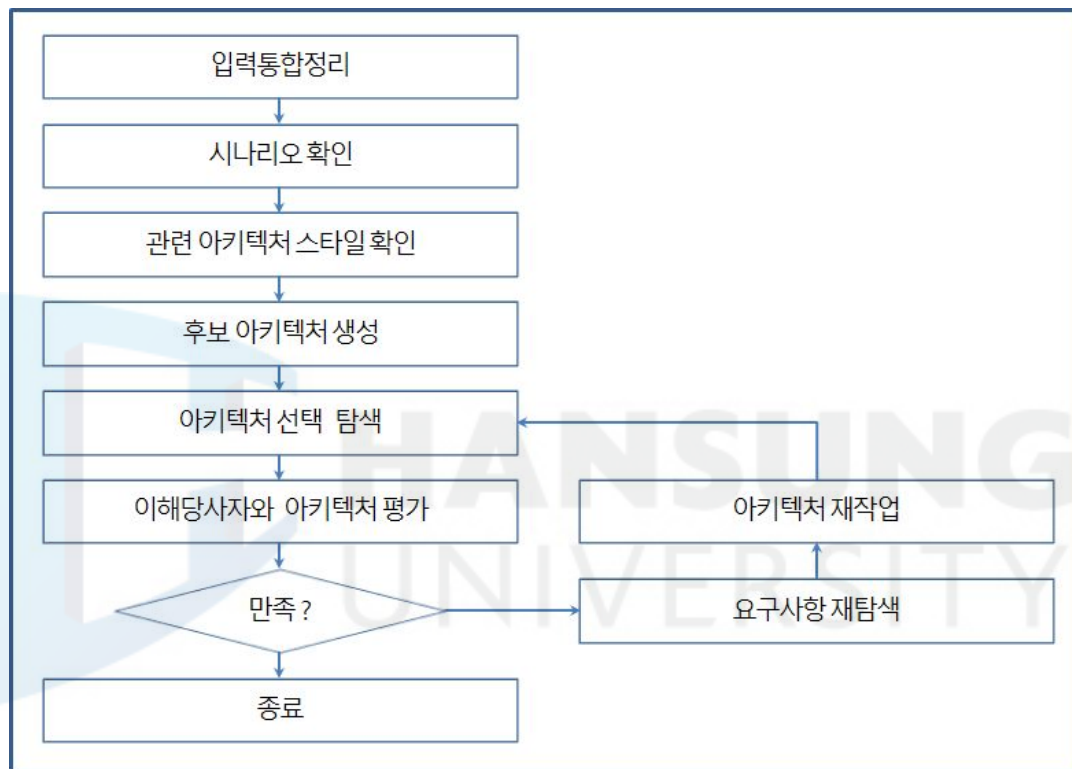
구분	설명
논리적 뷰	시스템의 기능적인 요구사항, 즉 시스템이 최종 사용자에게 해주는 것을 구조적으로 정의한 뷰
프로세스 뷰	실제 구동 환경을 살펴본 형태의 뷰
개발 뷰	개발 환경내의 정적인 소프트웨어 모듈(소스 코드, 데이터 파일, 컴포넌트, 실행 파일 등)의 구성을 나타낸 뷰
물리적 뷰	시스템이 실제로 설치되고, 배치되는 모습을 표현한 뷰
유즈케이스 뷰	위의 4가지 뷰 모두를 검증하고 통합시켜주는 뷰 (소프트웨어 아키텍처에 속하지 않음)

<표 2-1> 소프트웨어 아키텍처의 4+1 뷰

뷰타입은 뷰를 구조화 하는 방법으로, 모듈 뷰(Module View), 런타임 뷰(Runtime View), 할당 뷰(Allocation View)의 3가지 형태가 있다, 모듈 뷰는 코드 유닛들을 구성하는 방법을 제시하며, 런타임 뷰는 소프트웨어 시스템 요소들의 실행 행위와 상호작용을 표현하며, 할당 뷰는 하드웨어, 네트워크 등 비 소프트웨어 요소들과의 관계를 제공한다. 이러한 뷰들을 설계하기 위해서 소프트웨어 제약사항과 소프트웨어 구성 요소들과의 연관관계의 특별한 형태를 모아 놓은 것이 아키텍처 스타일이다.

제 4 절 소프트웨어 아키텍처 설계 프로세스

소프트웨어 아키텍처 설계 프로세스는 입력 통합 정리, 시나리오 확인, 관련 아키텍처 스타일 확인, 후보 아키텍처 생성, 아키텍처 선택 탐색, 이해 당사자와 아키텍처 평가, 아키텍처 재작업 및 요구사항 재탐색으로 이루어진다.



<그림 2-6> 소프트웨어 아키텍처 설계 프로세스

- 입력 통합 정리 : 초기 입력을 이해하고, 유효성을 확인하며, 정제한다. 입력으로는 범위, 문맥, 이해 당사자의 관심과 같은 가공되지 않은 프로세스 입력이 필요하다. 출력으로는 총합 정리된 입력이 나온다. 이를 위해서는 먼저 가공되지 않은 프로세스 인풋을 가져오고, 이들 간의 불일치성을 해결하여야 한다. 또한 공개된 질문에 대해 답해야 하고, 필요로 한 부분을 깊게 조사한다.
- 시나리오 확인 : 시스템에서 가장 중요한 요구사항을 기술할 수 있는

시나리오 집합을 찾는다. 입력으로는 통합 정리된 입력이 필요하고, 출력으로는 아키텍처 시나리오가 나온다. 이런 시나리오를 만들기 위해서는 두 가지를 고려해야 한다. 먼저, 시스템의 가장 중요한 속성을 특성화해야 한다. 또한 시나리오를 통해 아키텍처를 평가할 수 있어야 한다.

- 관련 아키텍처 스타일 확인 : 하나 이상의 검증된 아키텍처 스타일을 찾는다. 입력으로는 통합 정리된 입력과 아키텍처 시나리오가 필요하고 출력으로는 아키텍처 스타일이 나온다. 이를 위해서는 아키텍처 스타일의 기존 카탈로그를 검토하고, 시스템이 작동하는 조직을 고려한다. 또한, 예상되는 아키텍처와 관련 있는 스타일을 찾는다.
- 후보 아키텍처 생성 : 기본 아키텍처 관심을 반영하고 최초의 아키텍처를 개발한다. 입력으로는 통합 정리된 입력과 관련 아키텍처 스타일, 뷰포인트, 관점이 필요하고 출력으로는 아키텍처 뷰의 초안이 나온다. 이를 위해서는 초기 아키텍처 아이디어를 정의할 수 있는 아키텍처 뷰의 초기 집합을 만든다.
- 아키텍처 선택 탐색 : 다양한 아키텍처 가능성을 탐색하고, 그 중 하나를 선택하기 위하여 아키텍처 의사 결정을 한다. 입력으로는 통합 정리된 입력, 아키텍처 뷰의 초안, 아키텍처 시나리오, 뷰포인트, 관점이 필요하고, 출력으로는 좀더 상세하고 정확한 아키텍처 뷰가 나온다. 이를 위해서는 시나리오를 모델의 초안에 적용하여 확인해 본다. 또한 위험, 관심, 불확실성과 관련된 분야를 고려해 본다. 그리고 하나 이상의 해결 방법이 있을 때는 각각의 장단점을 평가해 본다.
- 이해 당사자와 아키텍처 평가 : 주요 이해 당사자와 아키텍처 평가를 하고 문제나 결점을 확인하며, 이해 당사자의 인수를 받는다. 입력으로는 통합 정리된 입력과 아키텍처 뷰와 관점 결과가 필요하고 출력으로는 아키텍처 검토 해설이 나온다. 이를 위해서는 대표적인 이해 당사자 집단과 아키텍처를 평가해야 한다.

- 아키텍처 재작업 : 평가 작업 동안 발생한 관심에 대해 언급한다. 입력으로는 아키텍처 뷰, 관련 아키텍처 스타일, 뷰포인트 관점, 아키텍처 검토 해설이 필요하고, 출력으로는 재작업한 아키텍처 뷰, 좀더 조사해야 할 분야가 나온다. 이를 위해서는 아키텍처 평가의 결과를 가져오고 좀더 나은 아키텍처를 만들기 위해 그 결과를 언급한다.
- 요구사항 재탐색 : 시스템 본래의 요구사항 중 변경된 것을 고려한다. 입력으로는 아키텍처 뷰와 아키텍처 검토 해설이 필요하고, 출력으로는 개정된 요구사항이 나온다. 이를 위해서는 이해 당사자와 함께 요구사항을 다시 검토하고 필요한 개정에 대해 동의를 얻어야 한다.



제 5 절 소프트웨어 아키텍처 스타일

아키텍처를 설계할 때 반복되는 유사한 조직적인 설계 스타일을 아키텍처 스타일(Architecture Style)이라 한다. 아키텍처 스타일은 작은 코드수준의 재사용 보다는 전체 시스템의 구조나 설계모형을 재사용하기 위한 구체적이고 체계적인 재사용 규약이다. 모든 종류의 시스템 설계에 적용할 수 있으며, 아키텍처 스타일별로 기본적인 특성 및 장단점이 이미 알려져 있다. 따라서 이러한 아키텍처 스타일을 중심으로 설계작업의 초기에 적절한 아키텍처 스타일을 적용하여 보고 이를 통해 시스템의 아키텍처적인 제약조건을 분석할 수 있게 한다.

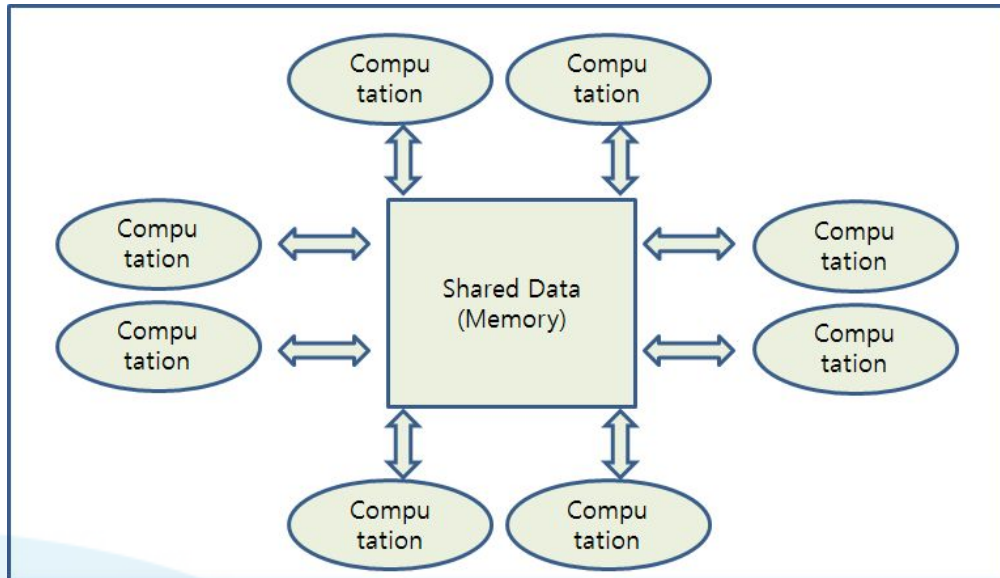
아키텍처 스타일은 저장소(Repository) 스타일, MVC(Module-View-Controller) 스타일, Client Server 스타일, 계층(Layered) 스타일, 파이프 필터(Pipe & Filter) 스타일, Publish-Subscribe 스타일, Peer-to-Peer 스타일, BlackBoard 스타일, Broker 스타일, Batch Sequential 스타일 등 여러 가지가 있으나, 본 논문에서는 위게임 모의모델의 아키텍처 스타일로 적용 가능한 저장소(Repository) 스타일, MVC(Module-View-Controller) 스타일, Client Server 스타일, 계층(Layered) 스타일, 파이프 필터(Pipe & Filter) 스타일의 5가지 스타일에 대해서만 고찰한다.

1. 저장소 스타일(Repository Style)

레파지토리 스타일은 중심이 되는 문제가 수립되고 논증되고 유지되는 복잡한 정보를 갖고 있는 어플리케이션에 적합한 스타일이다. 정보시스템, 프로그램 환경, 그래픽 에디터, 인공지능 기반, 역공학 시스템 등에 이용 될 수 있다.

레파지토리 스타일은 대량의 데이터를 저장하는데 효과적이며, 중앙 집중화를 통한 데이터 백업이 용이하고 보안적인 측면에서 강하며, 동시성 조절이 가능하다.

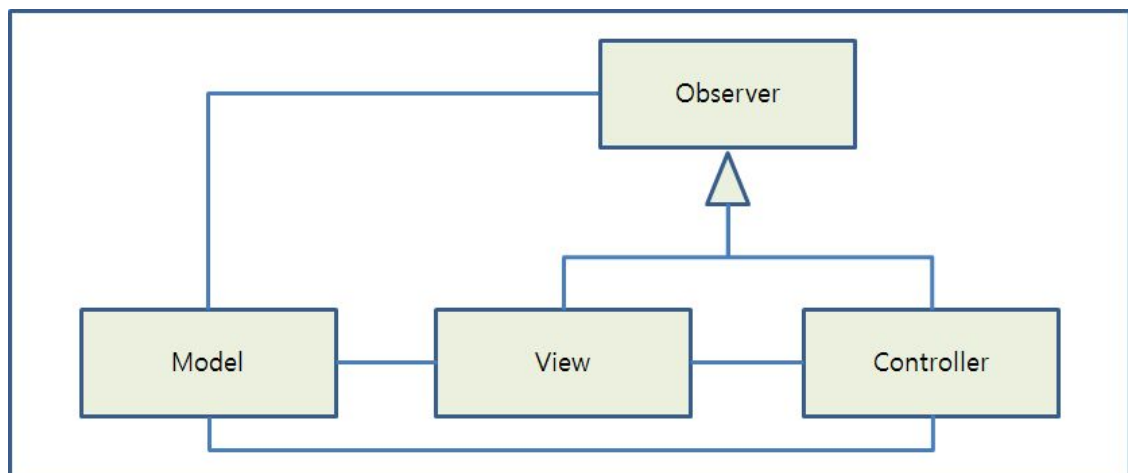
레파지토리 스타일은 <그림 2-7>에서 보는 것처럼 메모리상의 Shared Data를 Memory Access를 통해서 각 Computation 모듈들이 접근하는 구조를 보여준다.



<그림 2-7> 레파지토리 스타일

2. MVC (Module/View/Controller) 스타일

모델 뷰 컨트롤러 스타일은 상호작용 어플리케이션을 모델 컴포넌트, 뷰 컴포넌트 및 컨트롤러의 세 개 컴포넌트로 구분한다.

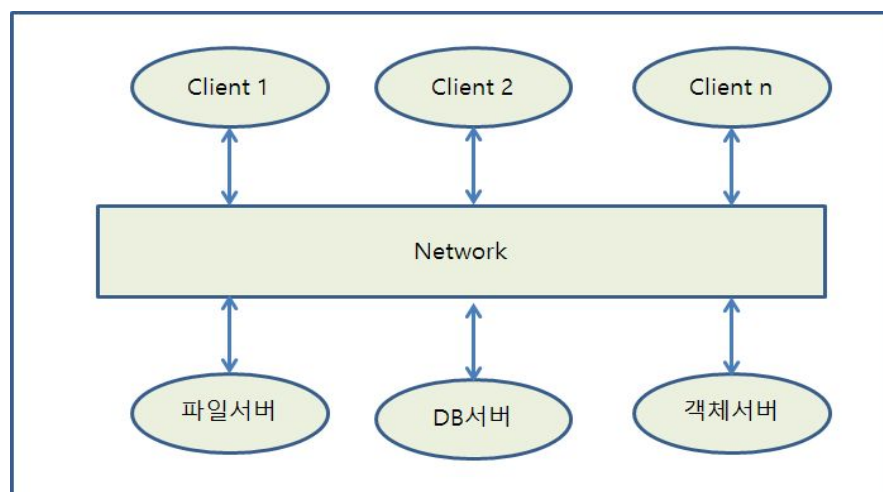


<그림 2-8> 모델 뷰 컨트롤러 아키텍처 스타일

모델 컴포넌트는 어플리케이션의 핵심 기능과 데이터를 포함하며, 적절한 데이터를 캡슐화 하고 어플리케이션에 특화된 프로세싱을 수행한다. 뷰 컴포넌트는 사용자에게 정보를 보여준다. 모델의 정보는 다양한 방법으로 다양한 뷰로서 보여준다. 컨트롤러 컴포넌트는 이벤트로서 사용자 입력을 관리하고, 뷰와 함께 UI를 구성한다. 이벤트들은 모델 혹은 연관된 뷰를 위한 요구사항들로 해석된다.

3. Client Server 스타일

클라이언트 서버 스타일은 여러 컴포넌트에 걸쳐서 데이터와 데이터를 처리하는 프로세싱 부분이 분산되어 있는 어플리케이션에 적용하기 적합하다. 각각의 컴포넌트는 다른 컴포넌트의 서비스 요청에 의해서 상호작용한다. 이 스타일에서 중요한 점은 커뮤니케이션은 클라이언트에 의해서 시작된다는 것이다. 클라이언트로부터 서비스 요청을 서비스를 제공하는 서버와 한 쌍을 이룬다. 이 스타일에서 서버는 하나 이상의 인터페이스를 통해 여러 서비스를 제공한다. 그리고 클라이언트는 시스템의 가른 서버가 제공하는 서비스들을 사용할 수 있다. 클라이언트 서버 스타일에서 하나의 중앙 서버 또는 분산된 여러 서버가 존재할 수 있다.

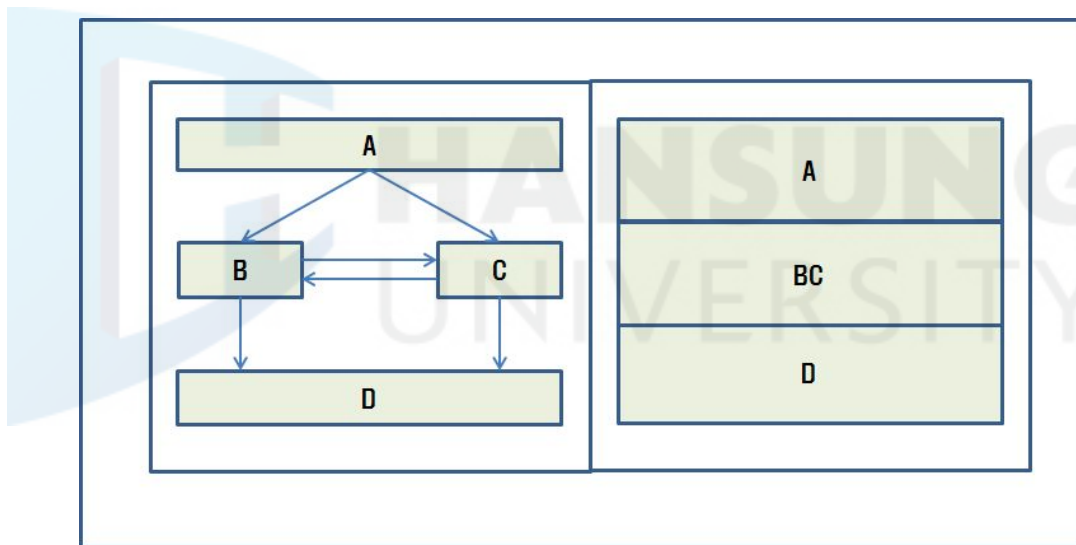


<그림 2-9> 클라이언트 서버 스타일

4. 계층 스타일 (Layered Style)

계층 구조(Layered Style)는 계층적으로 조직화될 수 있는 서비스로 구성된 어플리케이션에 적용하기 적합하다. 계층 스타일은 다음과 같은 4가지 항목(요소, 관계, 요소의 속성, 토폴로지)으로 기술될 수 있다. 계층 스타일의 요소는 계층이다. 각 계층은 모듈들의 응집된 집합이다. 계층내의 모듈들은 호출 또는 접근 될 수 있다. 계층 간의 관계는 사용 가능(Allowed to Use)의 관계이다.

계층 스타일에서는 소프트웨어를 계층이라는 단위로 나눈다. 각 계층은 하나의 가상 머신으로도 표현된다. 가상머신은 추상적인 컴퓨팅 장치로 소프트웨어와 하드웨어 사이의 인터페이스 역할을 하는 프로그램이다.

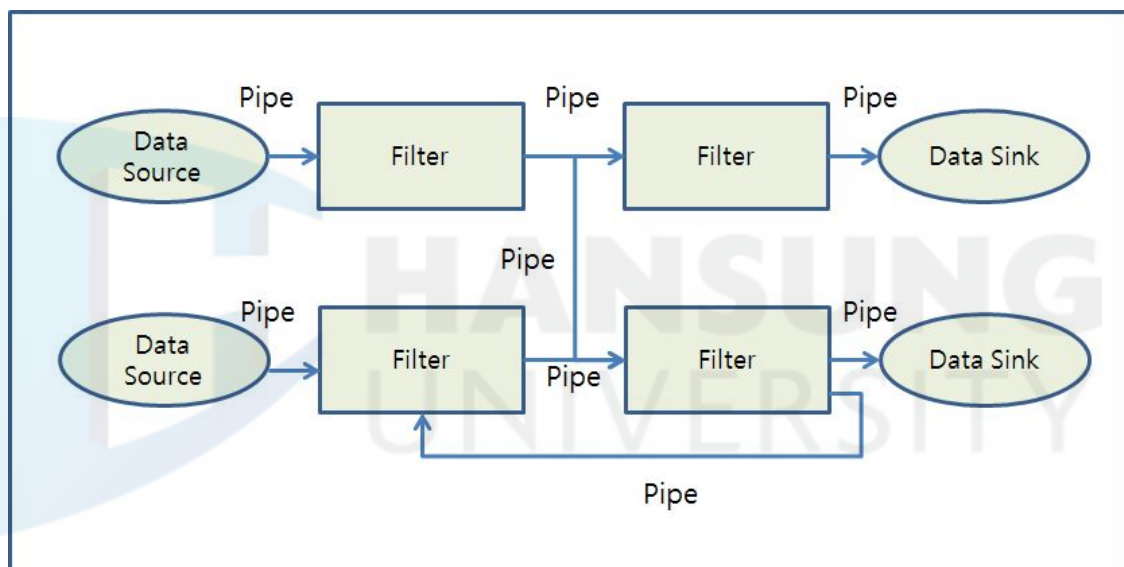


<그림 2-10> 계층 스타일

계층 스타일은 정보은닉(Information Hiding) 원칙이 적용된다. 하위계층의 변화는 상위계층의 인터페이스에 영향을 미치지 않는다. 상위계층의 인터페이스가 변경되지 않는한, 상위수준의 의존성은 인터페이스만 의존하게 된다. 또한 계층 스타일의 장점은 높은 이식성을 제공하는 것이다. 머신, 운영체제, 또는 다른 하위 레벨 의존성은 레이어 안으로 숨겨진다.

5. 파이프 필터 (Pipe & Filter) 스타일

파이프 필터 스타일은 데이터의 스트림을 처리하는 시스템을 위한 구조로서, 각 컴포넌트는 일련의 입력을 출력으로 변환하는 과정에서 입력과는 별도로 단계적인 처리를 통해 완성되는 출력집합이 독자적인 처리를 하도록 할 때 사용한다. 즉, 입력데이터를 파이프 구조를 통해 일련의 컴포넌트로 전달하면서 중간 중간에 특정 용도로 설계된 필터 역할을 하는 컴포넌트들이 각각의 작업을 통해 데이터를 변환하여 최종적으로 출력데이터를 생산하는 구조를 갖는 것이다.



<그림 2-11> 파이프 필터 스타일

또한 데이터의 처리 중에 시스템의 특성에 따라 불필요한 컴포넌트들을 거치지 않고 다음 처리로 옮기게 한다면 또는 다른 특별한 처리를 필요로 하는 데이터들에 한정적인 처리를 하는 컴포넌트를 추가한다면 하는 처리가 가능하다. 따라서 복잡한 데이터 처리나 대용량의 트랜잭션을 효율적으로 처리하면서 가변적인 유연성을 유지할 수 있게 한다.

제 6 절 디자인 패턴

디자인 패턴은 “재사용 가능한 객체지향 설계를 만들기 위해 유용한 공통의 설계 구조로부터 중요 요소들을 식별하여 이들에게 적당한 이름을 주고 추상화한 것”이다. 디자인 패턴은 패턴에 참여하는 클래스와 그들의 인스턴스를 식별하여 역할을 정의하고 그들 간의 협력 관계를 정의하고 책임을 할당한다. 각 디자인 패턴은 특정 객체지향 설계 문제에 집중하고 있다. 디자인 패턴은 언제 패턴을 적용할지, 다른 설계 제약을 고려하여 패턴을 적용할 수 있는지, 패턴을 사용했을 때의 결과가 어떻게 되는지를 기술한다.¹²⁾

구분		목적		
		생성	구조	행위
범위	클래스	Factory Method	Adapter	Interpreter Template Method
	객체	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Proxy Flyweight	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

<표 2-2> 디자인 패턴 영역

디자인 패턴의 분류는 <표 1>에서 제시한 것처럼 패턴이 무엇을 하는지 정의하는 목적 기준과 패턴의 적용을 구분하는 범위 기준의 2가지 기준을

12) GoF의 디자인 패턴, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides 공저, 2002.9, p4

사용한다.¹³⁾ 첫 번째 분류기준인 패턴의 목적기준으로 구분해 보면 패턴은 생성패턴, 구조패턴, 행위패턴 중의 한 가지 목적을 갖는다. 생성패턴(Creation Patterns)은 객체의 생성 과정에 관여하는 것이고, 구조패턴(Structural Patterns)은 클래스 또는 객체의 합성과 관련한 패턴들이다. 행위패턴(Behavioral Patterns)은 클래스 또는 객체들 간에 상호작용 하는 방법과 책임을 분산하여 처리하는 방법을 정의한다. 두 번째 분류 기준인 범위 기준으로 구분하면 패턴을 주로 클래스에 적용하는지 아니면 객체에 적용하는지를 구분하는 것이다. 클래스 패턴은 클래스들과 서브클래스들 간의 관련성을 다루는 패턴이다. 클래스 패턴의 관련성은 대부분 상속이며 컴파일 시점에서 정적으로 결정된다. 객체 패턴은 객체간이 관련성을 다루는 것으로 런타임 시점에 동적으로 변경할 수 있다. 대부분의 패턴들은 어느 정도 상속을 이용한다. 클래스 패턴으로 정의한 패턴만이 클래스 관련성을 이용하는데, <표 1> 디자인 패턴 영역에서 보는 바와 같이 일부만이 클래스 패턴이고 대부분의 패턴은 객체 영역에 속한다.

1. 생성 패턴 (Creation Patterns)

생성 패턴은 “생성에 관련된 패턴은 인스턴스를 만드는 과정을 추상화하는 것”으로 객체를 생성, 합성하는 방법이나 객체의 표현 방법에 독립적으로 시스템을 개발할 수 있게 한다. 이를 위해, 클래스 생성 패턴은 상속을 이용해서 인스턴스 화하는 방법을 다양하게 할 수도 있고, 인스턴스 생성을 다른 클래스에게 위임 할 수도 있다.¹⁴⁾

- Factory Method : 객체를 생성하는 인터페이스를 정의하지만, 인스턴스를 만들 클래스의 결정은 서브클래스가 한다. Factory Method 패턴에서는 클래스의 인스턴스를 만드는 시점을 서브클래스로 미룬다.
- Abstract Factory : 구체적인 클래스를 지정하지 않고 서로간에

13) 전계서, p11

14) 상계서, p93

관련성을 갖는 집합을 생성, 또는 서로 독립적인 객체들간의 집합을 생성 가능하게 하는 인터페이스를 제공한다.

- Builder : 복합 객체의 생성 과정과 표현 방법을 분리함으로써 동일한 생성 공정이 서로 다른 표현을 만들 수 있게 한다.
- Prototype : 프로토타입은 인스턴스를 이용하여 생성할 객체의 종류를 명시하며 이 프로토타입을 복사해서 새로운 객체를 생성한다.
- Singleton : 클래스의 인스턴스는 오직 하나임을 보장하며, 이 인스턴스에 접근 가능한 방법을 제공한다.

2. 구조 패턴 (Structural Patterns)

구조 패턴은 “더 큰 구조를 형성하기 위해 어떻게 클래스와 객체를 합성하는가와 관련된 패턴”이다. 구조 클래스 패턴의 이용기법은 상속 기법이며 인터페이스 또는 구현을 합성한다.¹⁵⁾

- Adapter : 클래스의 인터페이스를 접속한 클라이언트가 기대하는 다른 인터페이스로 변환시킨다. Adapter 패턴은 상호간에 호환성이 없는 인터페이스이기 때문에 동시에 사용할 수 없는 클래스를 변형하여 함께 작동하도록 해준다.
- Bridge : 추상화와 구현을 분리하여 각각을 독립적으로 변형할 수 있게 한다.
- Composite : 부분-전체 계층을 나타내기 위한 방법으로 복합 객체를 트리 구조로 만든다. Composite 패턴은 클라이언트가 개별적 객체와 복합 객체 모두를 동일하게 다루도록 한다.
- Decorator : 객체에 동적으로 책임을 추가할 수 있게 한다. 기능의 유연한 확장을 위해 상속 대신 사용할 수 있는 방법이다.
- Facade : 서브시스템에 있는 인터페이스 집합을 하나의 통합된 인터페이스로 제공한다. 이 패턴은 서브시스템을 좀 더 사용하기

15) 전제서, p166

편하게 하기 위해서 높은 수준의 인터페이스를 정의한다.

- Proxy : 다른 객체로의 접근 통제를 위해서 다른 객체의 대리자 또는 다른 객체에게 정보 보유자를 제공한다.
- Flyweight : 작은 크기의 객체들이 여러 개 있는 경우, 객체를 효과적으로 사용하는 방법으로 객체를 공유하게 한다.

3. 행위 패턴 (Behavioral Patterns)

행위 패턴은 어떤 처리의 책임을 어느 객체에 할당하는 것이 좋은지, 알고리즘을 어느 객체에 정의하는 것이 좋은지 등을 다룬다.¹⁶⁾

- Interpreter : 내부 표현 방법을 노출하지 않고 복합 객체의 원소를 순차적으로 접근 할 수 있는 방법을 제공한다.
- Template Method : 오버레이션에는 알고리즘의 처리 과정만을 정의하고 각 단계에서 수행할 구체적 처리는 서브클래스에 정의한다. 알고리즘의 처리 과정을 변경하지 않고 알고리즘 각 단계의 처리를 서브클래스에서 재 정의할 수 있게 한다.
- Chain of Responsibility : 요청을 처리할 수 있는 기회를 하나 이상의 객체에게 부여함으로써 요청하는 객체와 처리하는 객체 사이의 결합도를 없애려는 것이다. 요청을 해결할 객체를 만날 때까지 객체 고리를 따라서 요청을 전달한다.
- Command : 요청을 객체로 캡슐화 함으로써 서로 다른 요청으로 클라이언트를 파라미터화하고, 요청을 저장하거나 기록을 남겨서 오버레이션의 취소도 가능하게 한다.
- Iterator : 언어에 따라서 문법에 대한 표현을 정의한다. 또 언어와 문장을 해석하기 위해 정의한 표현에 기반을 두어 분석기를 정의한다.
- Mediator : 객체들 간의 상호작용을 객체로 캡슐화 한다. 객체들 간의 참조 관계를 객체에서 분리함으로써 상호작용만을 독립적으로

16) 전계서, p263

다양하게 확대할 수 있다.

- Memento : 캡슐화를 위배하지 않고 객체 내부 상태를 객체화하여, 나중에 객체가 이 상태로 복구 가능하게 한다.
- Observer : 객체 사이의 일 대 다의 종속성을 정의하고 한 객체의 상태가 변하면 종속된 다른 객체에 통보가 가고 자동으로 수정이 일어나게 한다.
- State : 객체의 내부 상태에 따라 행위를 변경할 수 있게 한다. 이렇게 하면 객체는 마치 클래스를 바꾸는 것처럼 보인다.
- Strategy : 알고리즘 군이 존재할 경우 각각의 알고리즘을 별도의 클래스로 캡슐화하고 이들을 상호 교환 가능한 것으로 정의한다. 클라이언트에 영향을 주지 않고 독립적으로 알고리즘을 다양하게 변경할 수 있게 한다.
- Visitor : 객체 구조의 요소들에 수행할 오퍼레이션을 표현한 패턴이다. 오퍼레이션이 처리할 요소의 클래스를 변경하지 않은 상태에서 새로운 오퍼레이션을 정의할 수 있게 한다.

제 3 장 소프트웨어 아키텍처 설계를 위한 사례연구

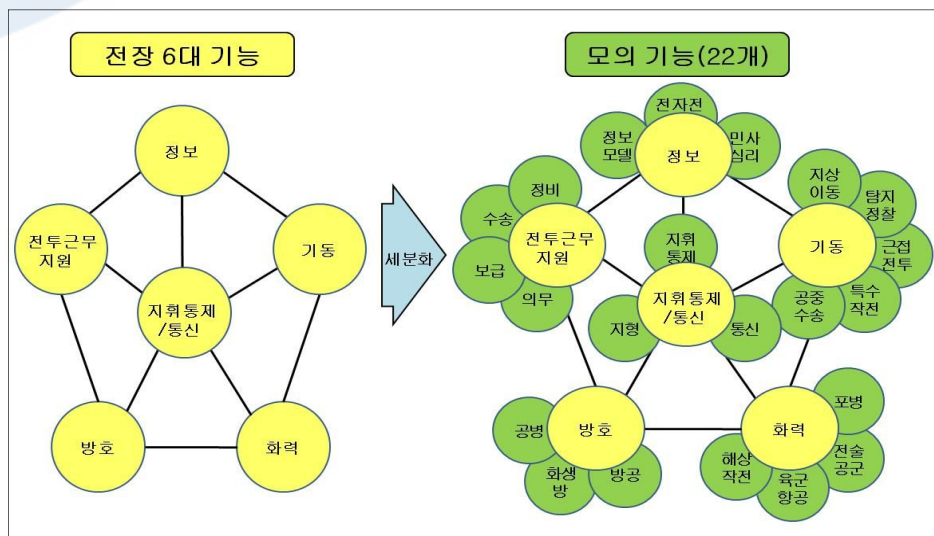
제 1 절 창조21 모의모델 개발사례 연구

창조21 모델은 사·군단급 전투지휘훈련을 지원하기 위하여 지형, 피아 무기체계 및 편성 등 한반도의 전장실상을 구체적으로 묘사하는 모의모델이다. 또한 최신 첨단기술을 활용하여 세대별 위게임 모델 체계로서의 확장성과 군수모델, 해·공군모델, C4I체계, 미군 CBS모델 등과의 연동성을 고려하여 개발되었다.¹⁷⁾

창조21 모델의 모의수준을 소대 및 분대급 규모부대 까지 모의가능하며 최대 운용부대 수는 5만개 까지 모의가 가능하다.

창조21 모델의 수행하는 기본적인 기능은 백두산 훈련 경험 부대를 대상으로 개선 요구사항을 수집하여 육군대학 및 교리 담당부서의 검토와 야전의견을 수렴하여 최종 확정되었으며, 전장 6대 기능 및 통합전투 수행에 관한 내용으로부터 장차적 양상 및 지형에 관한 내용을 포함하고 있다.

창조21 모델은 <그림 3-1>과 같이 6대 전장기능을 22개 모의기능으로 세분화하여 모의한다.



<그림 3-1> 창조21 모의모델 전장기능

17) 교육참고 25-14 창조21모델 '11모의논리 분석서, 육군교육사령부, 2011.1

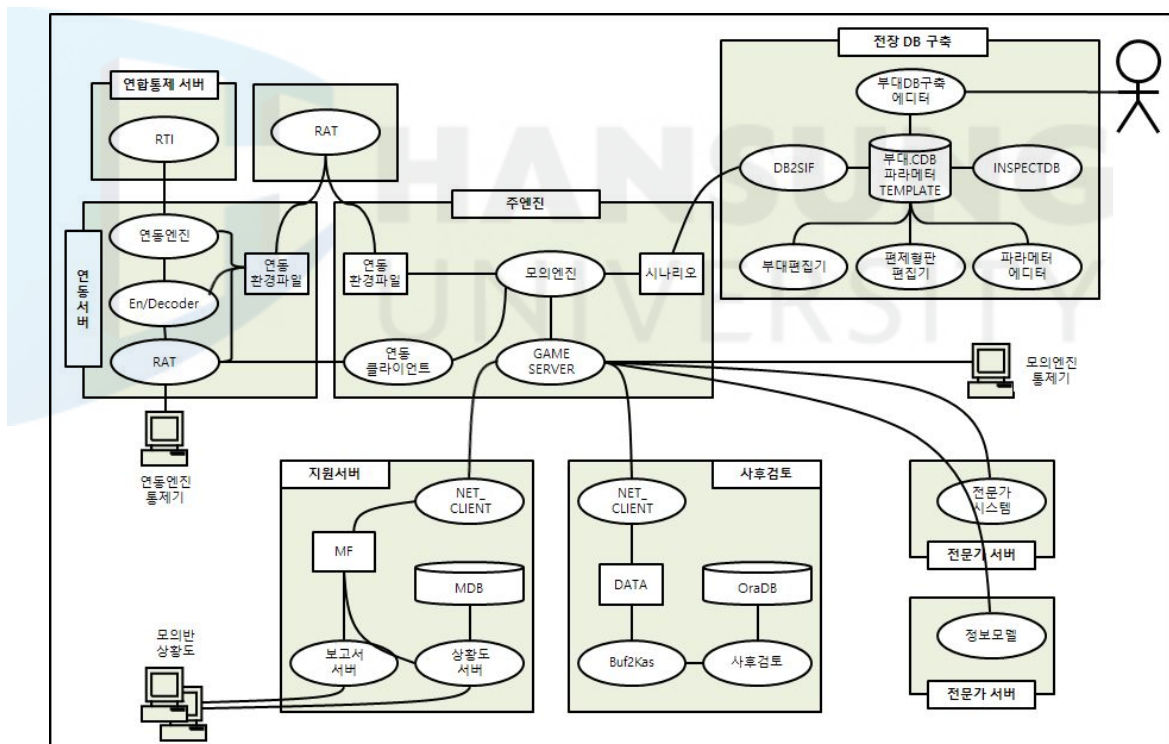
기동분야는 실 지형에서 모든 방향으로 자유롭게 기동할 수 있으며, 간접화기에 의한 근접전투 손실방법을 개선함으로써 한반도와 같은 착잡한 지형에 적합한 모델이 되도록 개발되었다. 근접전투 평가를 위해 전문가 시스템을 사용하였으며, 전문가 시스템을 다양한 전투 상황을 입력받아 기본 손실률에 계량화가 곤란한 핵심적 전투영향요소인 은폐/엄폐, 전술적 기동성, 표적획득능력, 제병협동능력 등을 평가한다. 기동분야는 지상군, 특수작전, 공중수송으로 구분된다. 지상군의 주요 기능으로는 전술/행정 이동, 도보/차량이동, 근접전투 등이다. 특수작전의 주요기능은 침투, 매복, 습격이며, 공중수송의 주요 기능은 육군항공, 공군 수송기 이용 병력/물자 공수 등이다.

화력분야는 포병의 피해평가 알고리즘과 조명탄 사격 논리를 적용하였으며, 화력분야의 주요기능은 포병, 통합화력, 공격헬기, 육군항공 작전(공격, 차장, 방호, 정찰) 등이다. 전투근무지원 분야는 우리 실정에 맞는 군수제원을 적용하고 보급수송 지원 모의방법을 적용하였으며 보급, 정비, 수송, 인사의무 분야의 제반기능을 모의한다. 공병분야는 장애물 설치/제거, 진지구축, 일반공병지원, 도하지원 등을 모의하며, 방공무기(고도별), 항공기 탐지/식별/교전 등의 방공분야와 화학 공격 및 연막탄, 제독활동의 화학기능을 모의한다. 또한 전자전, 해상지원작전 모의기능을 정보분야와 화력분야에 추가함으로써 전장의 제반 상황을 모두 모의할 수 있도록 발전시켰으며, 전술공군 분야에서는 공대지, 대공제압, 근접항공지원 및 공중수송 기능을 모의한다.

데이터베이스(DB)는 크게 모델인수DB, 부대제원DB 그리고 지형DB로 구분된다. 모델인수DB는 전장상황을 모의하기 위하여 필요한 각종 제원으로써 장비 및 보급품에 대한 세부 제원, 근접전투간 적용되는 소모계수, 임무수행에 소요되는 기준시간 등 프로그램 실행시 참조해야 하는 각종 인수를 포함한다. 부대제원DB는 훈련에 참가하는 모든 부대제원과 이동 방향, 전투력 배분, 태세전환점 등 부대의 현 상태를 나타내는 제원과 군수제원 등이 포함된다. 지형DB는 지형형태, 초목화, 도시화 등의 면정보와 도로 및 하천 등의 선정보, 그리고 교량, 장애물 등의 점정보로 구성되며, 지형편집기를 통해 훈련실시간 지형DB의 수정이 가능하다.

창조21 모델의 시스템 구성도는 <그림 3-2>에서 보는 것처럼 전장DB구축, 주엔진, 모의통제기, 모의반상황도, 지원서버, 사후검토, 전문가시스템, 정보모델 및 연동서버로 구성되어 있다. 모의통제기 프로그램은 전장모의를 실시간에, 또는 필요시 고속으로 진행시킬 수 있고, 기술통제관에 의한 모의통제를 가능하게 하며, 지원서버는 통제본부와 전방에 위치한 통제본부의 컴퓨터들을 상호 연결시켜 주는데 동시 부하를 최소화하는 것이 기술의 핵심이다.

또한 전장DB 구축분야의 데이터베이스 프로그램은 처리속도를 향상시키기 위해 독자적으로 설계하였고 정밀한 지형 묘사가 가능하도록 지형 데이터베이스를 개발하는 등 그 동안 군의 위게임 운용경험과 학계의 지원을 받아 개발 당시의 국내 최고기술을 결집하여 개발하였다.



<그림 3-2> 창조21 모델 시스템 구성도

전투모의요원 명령 입·출력을 위한 모의반 상황도는 위게임 모델과 전투모의요원을 연결시켜주는 단말기로서 최초 설계시부터 명령 입출력 및

상황도 전시를 단일 모니터로 처리하고 입출력 메뉴를 기능중심에서 부대중심체계로 설계 하여 사용자의 편의성을 최대한 도모하였다. 모의반 상황도에서 구현된 내용은 실시간에 7개 축척의 상황도를 선명하게 전시할 수 있고, 각종 전장상황을 실전처럼 나타나도록 하였 으며 또한 모든 명령 입력과 보고서 출력시 한글 및 표준 전술용어를 사용하고 도움말을 제공하는 등 전투모의요원의 부담을 최대한 감소시켰다.

지형 데이터베이스는 훈련전 또는 실시간 수정 소요가 발생되는데, 이를 고려하여 도로의 신설, 변경사항이나 하천, 교량 지형형태 등의 지형자료를 훈련 준비간 또는 훈련간에 수정 가능토록 개발함으로써 최신 실지형을 적용한 전투지휘 훈련이 가능하게 한다.

제 2 절 태극JOS 모의모델 개발사례 연구

태극JOS 모델은 태극연습 주 통제모델로서 3군 합동작전 연습과 합참 및 작전사 지휘관, 참모 훈련을 지원하며, 합참대학 합동작전 위게임 교육에 활용된다. 태극JOS 모델은 전구급 연합 및 합동작전을 모의하며 C4ISR, 기동/대기동, 화력, 전투근무지원, PGM, 핵/화학전, 전자/심리전 및 상륙전의 모의능력을 가지고 있으며, 지상전은 최대 10,000개의 부대 및 30,000개의 표적을 모의할 수 있다. 해상전은 개별 함정 및 전투단 단위로 모의하며, 공중전은 임무단위로 모의한다.¹⁸⁾

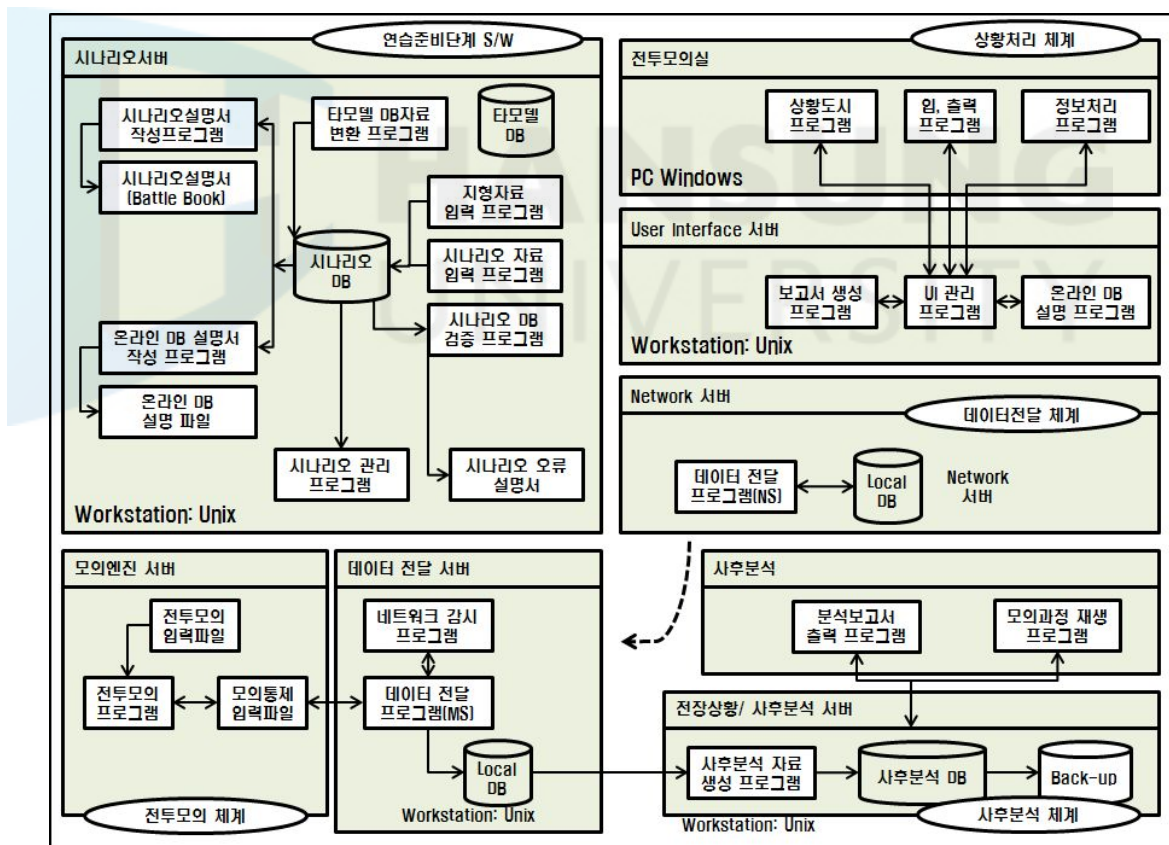
태극JOS 모델의 지·해·공군 및 군수, 정보기능은 각각 독립작전을 모의하며, 필요에 따라 이들 기능이 상호간에 기능간 연동 연동(합동) 작전을 모의한다. 태극JOS 모델의 특징은 군별·병과별 합동·협동작전을 세부모의하며, 실제 도로/야지 기동, 해상/공중 실기동 경로를 구현하고, 차량, 수륙양용, 철도 이동 등 다양한 기동효과와 정찰자산의 첩보수집을 세부적으로 모의하는데 있다.

태극JOS 모의모델의 주요 기능으로서는 지상전투 모의기능에 지상군이 보유한 제반의 전투자산병력, 무기체계, 지상부대 등을 모의하며, 해상전투는

18) 태극JOS 모의논리서, 한국국방연구원, 2012.12

해군이 보유한 제반의 전투자산함정, 함포, 유도탄, 기뢰, 어뢰, 센서, 함정탐재기 등을 모의한다. 공중전투는 공군이 보유한 제반의 전투자산 항공기, 공대지 및 공대공 무장, 활주로, 대공무기, 센서, Jammer 등을 모의한다. 군수모의기능은 지·해·공군의 임무수행을 지원하는 기능으로 보급품 소비, 보급, 초과보급품 처리, 보급품 수송, 부대간 보급품 이전, 보급품 고갈시 해당 전투체계 전투력 상실 등을 모의한다. 정보모의기능은 부대가 보유한 정보자산 정찰항공기, 레이더, 센서, 인간정보팀 및 특수전 부대 등을 모의한다.

태극JOS 모델은 연습준비단계의 소프트웨어로 자료관리 및 시나리오 관리 프로그램이 있으며, 실행단계의 소프트웨어는 모의실행, 상황도 프로그램, 분석단계는 지상전/해상전/공중전 분석 프로그램으로 구성되어 있다.

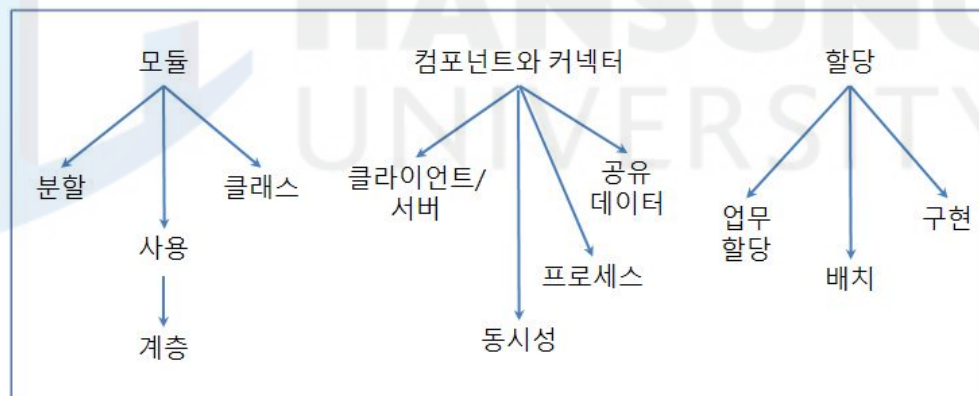


<그림 3-3> 태극JOS 모델 시스템 구성도

제 4 장 위게임 모의모델 소프트웨어 아키텍처 설계

제 1 절 소프트웨어 아키텍처 설계를 위한 아키텍처 구조 정립

소프트웨어 아키텍처의 표현은 구조와 뷰로 표현된다. 본 논문에서는 소프트웨어 아키텍처 표현 방법 중에서 위게임 모의모델의 재사용성 및 상호운용성 향상에 효율적으로 적용할 수 있는 아키텍처 구조에 대하여 연구하고 정립하고자 한다. 소프트웨어 아키텍처 구조는 소프트웨어나 하드웨어에 존재하는 아키텍처 요소 그 자체이다. 뷰는 이해관계자에 의해 읽히고 쓰이는 아키텍처 요소의 일관된 집합으로 아키텍처 요소 집합과 그들 간의 관계로 이루어진다. 아키텍처 구조는 요소가 보여주는 의미에 따라 크게 모듈구조, 컴포넌트와 커넥터 구조 및 할당구조의 3가지 그룹으로 나누어진다.



<그림 4-1> 소프트웨어 아키텍처 구조

1. 모듈(Module) 구조

가) 분할(Decomposition) 구조

분할구조의 기본단위는 “~의 하위모듈이다(is-part-of)”라는 관계이다. 분할구조는 가장 큰 모듈에서부터 충분히 이해가 가능한 작은 단위로

분할하는 것이 기본적인 방식이다. 일반적으로 분할 구조에서 설계가 시작된다. 즉 아키텍트는 소프트웨어 단위가 어떤 일을 해야 하는지 결정하고 그 단위가 어떤 모듈과 연동되는지 설계해야 한다.

나) 사용(Users) 구조

사용 구조는 모듈이나 프로시저, 모듈의 인터페이스에 있는 자원 등의 단위로 구성되며, 이러한 자원들의 의존(depends-on) 관계를 사용이라는 관계로 특화시켜 놓은 것이다. 즉 A의 정확성을 위해 B의 정확한 버전이 존재해야 하는 경우, A가 B를 사용한다고 정의한다. 사용구조는 새로운 기능을 시스템에 추가하거나 유용한 기능의 부분집합을 추출하는데 유용하다. 아키텍트는 이 스타일을 활용해 아키텍처의 구현 방식에 제약을 가할 수 있다. 이 스타일은 통해 개발자들은 자신들이 맡은 부분 외에 시스템에 어떤 모듈이 더 있어야 제대로 작업이 가능한지 파악할 수 있다. 이 강력한 스타일 덕분에 전체 시스템 중에서 일부분만 점증적으로 개발해서 배치하는 일이 가능해진다.

다) 계층(Layered) 구조

사용 구조에서 사용 관계를 특정한 방법으로 잘 조절하면 시스템의 계층이 나타난다. 계층은 서로간에 밀접한 관련성이 있는 기능들의 일관된 집합이라고 할 수 있다. 엄격한 계층구조에서는 n계층은 바로 밑에 있는 n-1계층의 서비스만을 사용한다. 계층을 사용하면 소프트웨어를 여러 부분으로 깔끔하게 나눌 수 있고, 나뉜 개별 부분들은 공용 인터페이스를 갖춘 하나의 가상기계를 형성하면서 하나의 응집된 서비스 집합을 제공 할 수 있다.

라) 클래스(Class) 구조

클래스 구조는 상속구조가 대표적인 구조이며, 상이한 코드단위들이 어떤 식으로 관계를 맺고 있는지 보여준다. 즉 일반화 스타일로 표시함으로써 어느 클래스가 어느 클래스로부터 상속받을지 결정할 수 있다. 이 스타일은 객체지향 설계 내용을 표현하는데 널리 사용 될 뿐 아니라 기타 다양한

형태의 유지보수를 지원하는 데 자주 사용된다. 재사용할 때는 클래스를 단위로 하는 경우가 많고, 새로운 함수를 추가 할 때는 주로 기존의 함수를 수정해서 하며, 오류가 발생해서 고쳐야 하는 부분도 대개 클래스 단위로 명세를 한다.

2. 컴포넌트와 커넥터(Component and Connector) 구조

가) 클라이언트 서버(Client Server) 구조

클라이언트와 서버 개념의 시스템이라면 이 구조를 사용하는 것이 좋다. 컴포넌트는 클라이언트와 서버이고 커넥터는 시스템의 기능을 수행하기 위해 클라이언트/서버가 사용하는 통신규약과 메시지다. 기능을 분화하기가 용이하고 물리적으로 분산 처리 기능을 표현하기에 유리하며 부하 분산에 유용하게 사용된다.

나) 동시성(Concurrency) 구조

이 컴포넌트의 커넥터 구조를 통해 아키텍트는 병렬 구조의 아키텍처를 설계 할 수 있고 자원 경쟁이 발생할 수 있는 지점을 찾을 수 있다. 동시성 구조에서 사용하는 단위는 ‘논리적 스레드’ 컴포넌트와 커넥터이다. 논리적 스레드란 연산 작업의 연속을 표현한 아키텍처 요소로서 이후 설계 과정에서 물리적 스레드에 할당될 수 있다. 동시성 구조는 설계 초기에 동시 수행과 관련된 요구사항을 찾아내는 데 쓰인다.

다) 프로세스(Process) 구조

컴포넌트와 커넥터 구조는 모두 모듈 구조와 상호 독립적이다. 따라서 프로세스 구조 역시 모듈 구조와 상호 독립적이다. 프로세스 구조는 시스템 실행 관점을 표현하는 동적인 모습을 표현한다. 다시 말해서 프로세스나 스레드 간의 연결이나 상태 일치, 독립된 행위를 프로세스 구조에 표시한다.

라) 공유 데이터(Shared Data) 구조

공유 데이터 구조는 저장소(Repository) 구조라고도 할 수 있다. 이 구조는 변화가 없는 데이터를 생성, 저장하고 이용하는 컴포넌트와 커넥터로 구성된다. 시스템이 하나 이상의 공유 데이터 저장소로 구성되어 있다면 이 구조로 표현하기 좋은 모델이다. 이 구조는 실행 시에 소프트웨어 요소가 어떻게 데이터를 생성하고 사용하는지를 보여준다.

3. 할당 (Allocation) 구조

가) 업무할당 (Work Assignment) 구조

업무할당 구조는 개발팀에게 구현이나 통합 모듈의 개발 책임을 적당히 할당하는 것을 말한다. 업무할당 구조를 아키텍처의 한 부분으로 수립하면 관리적인 측면뿐만 아니라, 아키텍처적으로도 누구에게 어떤 일을 시킬지 확실하게 결정할 수 있다. 업무할당 구조는 대규모 분산 프로젝트에서 기능 공통 모듈을 구분하는 수단으로 사용할 수 있고, 별도의 공통 모듈을 단일한 팀에 할당할 수 있을 것이다. 이렇게 적당히 할당하는 것이 개발자가 필요한 공통 기능을 스스로 개발하는 것보다 훨씬 효과적이다.

나) 배치 (Deployment) 구조

배치구조는 소프트웨어가 하드웨어/프로세스 혹은 대화 요소에 어떻게 할당되는지 보여준다. 할당 구조의 단위는 소프트웨어(대개 컴포넌트와 커넥터 뷰의 프로세스)와 하드웨어 프로세서, 통신 경로이다. 단위들 간의 연결은 “~에 할당되는” 이나 “~에 전이되는” 이란 의미를 가진다. “~에 할당되는”의 의미는 소프트웨어 요소가 물리적 단위에 어떻게 대응되는지에 대한 것이고, “~에 전이되는”은 소프트웨어 배치가 동적일 경우에 사용된다. 이런 관점을 통해 엔지니어는 성능, 데이터 무결성, 가용성, 보안에 대해 추론할 수 있다. 이 구조는 특히 분산과 병렬 처리 시스템에 강점을 보인다.

다) 구현 (Implement) 구조

구현 구조는 소프트웨어 요소(소스코드, 데이터파일, 컴포넌트, 실행 파일

등)가 시스템의 개발환경, 통합환경, 형상관리환경에서 파일 구조에 어떻게 대응되는지 보여준다. 소프트웨어는 서브시스템 등의 작은 단위로 패키지되며, 이들 패키지는 소수의 개발자에 의해 개발된다. 서브시스템은 레이어 계층으로 구성되며, 각 레이어는 상위레이어에 대하여 정의된 인터페이스를 제공한다. 이 구조는 개발 활동 관리와 프로세스 생성관리, 개발 용이성, 소프트웨어 자산 및 재사용성 등의 문제를 해결한다.

제 2 절 위게임 모의모델 상위레벨 계층구조 설계

위게임 모의모델을 활용하여 실제 연습 및 훈련을 실시하기 위해서는 위게임을 모의하기 위한 모델, 모델을 장치하고 분석할 수 있게 하는 컴퓨터 관련 장비 및 시설, 이러한 것들을 운용할 전문요원이 필요하다. 또한 모델에 들어가는 입력자료(DB)와 군 작전과 작전계획 등을 이해할 수 있는 작전운용개념이 필요하다. 특히 위게임 모의모델은 모델 논리 파악, 분석 및 설계, 모델 개발 등의 과정으로 개발되는 바, 개발 간 유사 모의논리 및 기능 대한 중복개발로 개발 생산성을 저하시키는 경우가 발생한다. 따라서 위게임 모의모델 개발간 재사용성 및 상호운용성 향상은 물론 개발 생산성을 향상시키기 위한 위게임 모의모델의 상위레벨 계층구조도를 제시한다.

위게임 모의모델을 통한 연습 및 훈련은 우선 연습 및 훈련대상에 따라 모의기능 및 모의대상 무기체계가 결정된다. 모의기능 및 무기체계에 따라 관련 제원 및 파라미터 등이 결정되고, 모의기능을 이벤트기반(Event Driven)과 시간기반(Time Driven)으로 처리하기 위한 시뮬레이션 엔진이 필요하다. 또한 시뮬레이션 결과를 훈련 참여자 및 관련 연동체계에 전달하기 위한 데이터 전달 및 연동체계, 모의결과를 가시화하여 상황도상에 표현하고 명령의 입력/출력 기능을 제공하는 전장상황도시, 연습 및 훈련 결과를 종료 후에 분석 및 발전시키기 위한 사후분석 기능이 제공되어야 한다.

따라서 앞에서 제시한 창조21모델과 태극JOS모델의 시스템 구성도와 소프트웨어 설계 및 위게임 모의모델을 통한 연습 및 훈련개념을 기반으로 위게임 모의모델의 구성요소와 그들이 가진 특성, 구성요소간의 관계를

고려하여 상위레벨 계층구조를 분석한 결과 <표 4-1>에서 보는 것처럼 창조21 모의모델은 전장DB구축, 주엔진, 지원서버, 연동서버, 연합통제서버, 모의반상황도, 사후검토의 계층구조로 구분할 수 있으며, 태극JOS 모의모델은 연습준비단계, 전투모의체계, 데이터전달체계, 연동체계, 상황처리체계, 사후분석체계로 계층구조를 구분할 수 있다.

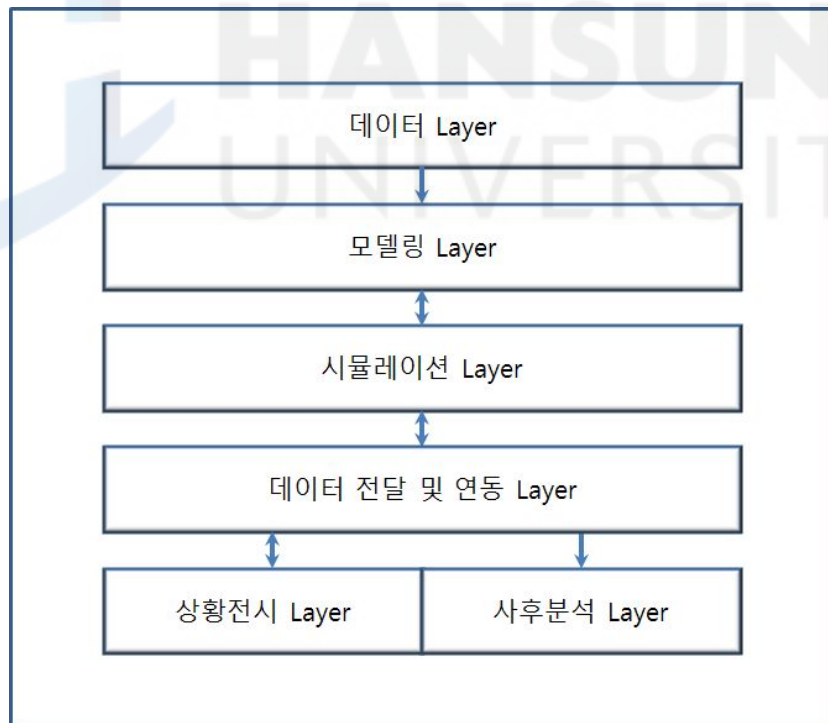
창조21모델 계층구조	공통 계층구조	태극JOS모델 계층구조
전장 DB 구축	데이터 Layer	연습준비단계 SW
주엔진	모델링 Layer	전투모의 체계
주엔진	시뮬레이션 Layer	전투모의 체계
지원서버	데이터 전달 Layer	데이터 전달 체계
연동서버, 연합통제서버	연동 Layer	연동체계
모의반상황도	상황전시 Layer	상황처리체계
사후검토	사후분석 Layer	사후분석체계

<표 4-3> 창조21 및 태극JOS 모델을 통한 공통 계층구조 분석

창조21 모의모델의 계층구조와 태극JOS 모의모델의 계층구조를 기반으로 하여 공통계층 구조를 정립하면 데이터 레이어, 모델링 레이어, 시뮬레이션 레이어, 데이터 전달 및 연동 레이어, 상황전시 레이어 및 사후분석 레이어의 6가지 레이어로 구분되며 이를 위게임 모의모델의 상위레벨 계층구조로 정립한다.

- 데이터 레이어는 위게임 모의모델 연습 및 훈련 준비단계에서 주로 사용되는 기능이며 연습/훈련에 사용되는 부대, 무기체계 및 파라미터 데이터와 코드를 관리하며, 데이터에 대한 정합성 등 데이터 검증관리를 포함하고 있다.
- 모델링 레이어는 군사작전 모의를 위하여 부대 및 전투체계, 작전지역 등을 포함한 전장환경, 그리고 자연 및 인공현상과 전술교리 등의 작전상황을 지상작전, 해군작전, 공군작전 및 해병대작전으로 구분하여 물리적·수학적·논리적 표현으로 만들어 놓은 과정이다.

- 시뮬레이션 레이어는 모델 즉 모델링의 산출물을 활용하여 시간의 연속적인 흐름 속에서 군사작전을 최대한 실제와 유사하게 실행하는 것으로 이벤트 관리, 시간관리, 모의기능관리 및 모의데이터관리 등을 포함하고 있다.
- 데이터 전달 및 연동 레이어는 위게임 모의모델의 각 서브시스템간의 데이터 전달 역할을 수행하며, 외부체계 즉 C4I체계와 M&S연동 체계와의 인터페이스 역할을 수행한다.
- 상황전시 레이어는 위게임 모의모델의 시뮬레이션 결과를 2D/3D 상황도로 묘사하고, 통제관 및 게이머의 명령 입력 및 모니터링 기능을 제공한다.
- 사후분석 레이어는 위게임 모의모델의 시뮬레이션 결과를 수집하여 모의결과를 지상작전, 해군작전, 공군작전 및 해병대작전별로 분석할 수 있는 모의결과 분석기능과 모의결과를 재생하는 기능을 제공한다.



<그림 4-2> 위게임 모의모델 상위레벨 계층구조

제 3 절 모듈 구조

구조와 뷰로 표현되는 소프트웨어 아키텍처를 위하여 위게임 모의모델 상위레벨 계층구조를 기반으로 한 소프트웨어 구조를 모듈구조, 컴포넌트와 커넥터 구조, 할당구조로 구성하면 다음과 같다.

1. 분할(Decomposition) 구조

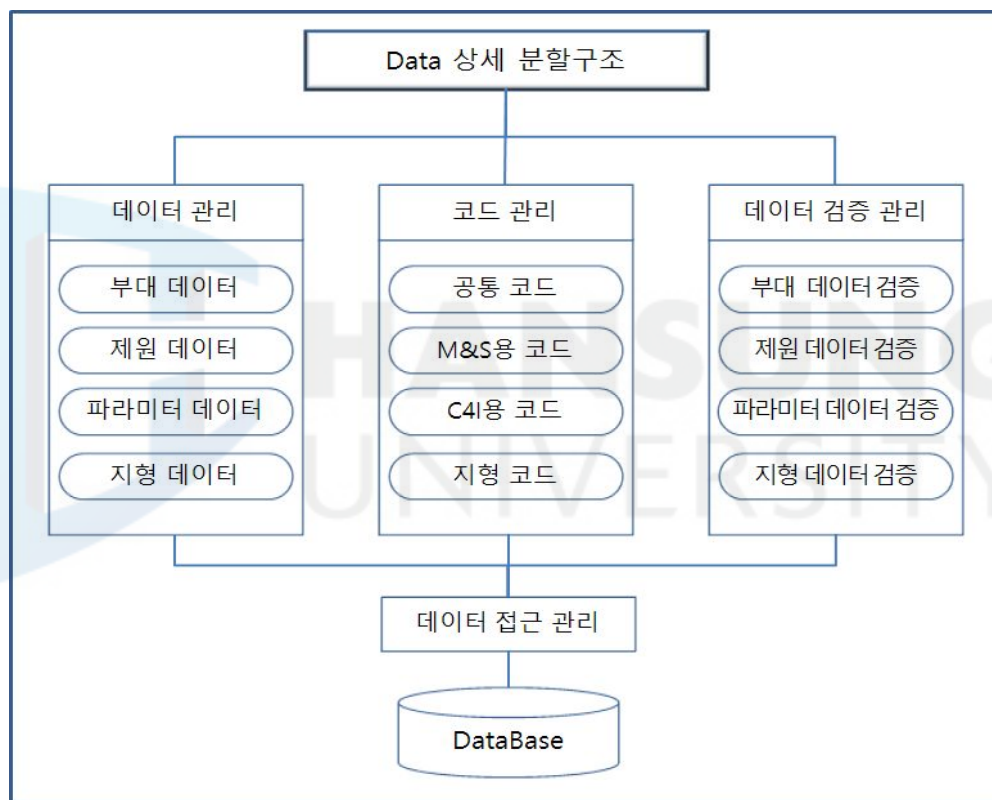
위게임 모의모델의 분할구조는 “~의 하위모듈이다(is-part-of)”라는 관계에 근거하여 위게임 모의모델의 가장 큰 모듈에서부터 각 업무단위의 작은 단위로 분할하는 것이 기본적인 방식이다. 일반적으로 분할 구조에서 설계가 시작됨에 따라 위게임 모의모델의 기본 분할구조는 소프트웨어 단위가 어떤 일은 해야 하는지 결정하고 그 단위가 어떤 모듈과 연동되는지 설계해야 한다. 앞 장에서 제시한 위게임 모의모델 상위레벨 계층구조를 기반으로 하여 각 Layer별 세부 모듈을 분할하면 아래 그림과 같다.



<그림 4-3> 위게임 모의모델 분할구조

가) 데이터 상세 분할구조

워게임 모의모델에서의 데이터는 모델링의 기초가 되는 매우 중요한 요소이다. 모델링은 부대, 무기체계 제원, 파라미터 및 지형 데이터 간의 상관관계에 의해서 대부분 이루어지며, 그 모델링의 결과가 이벤트 등으로 처리되어 사용자 및 연습/훈련 참여자에게 제공된다. 따라서 데이터 상세 분할 구조는 데이터 관리, 코드 관리 및 데이터 검증 관리로 구분할 수 있으며, 데이터는 부대 데이터, 무기체계 제원 데이터, 파라미터 데이터 및 지형 데이터로 상세 분할된다.

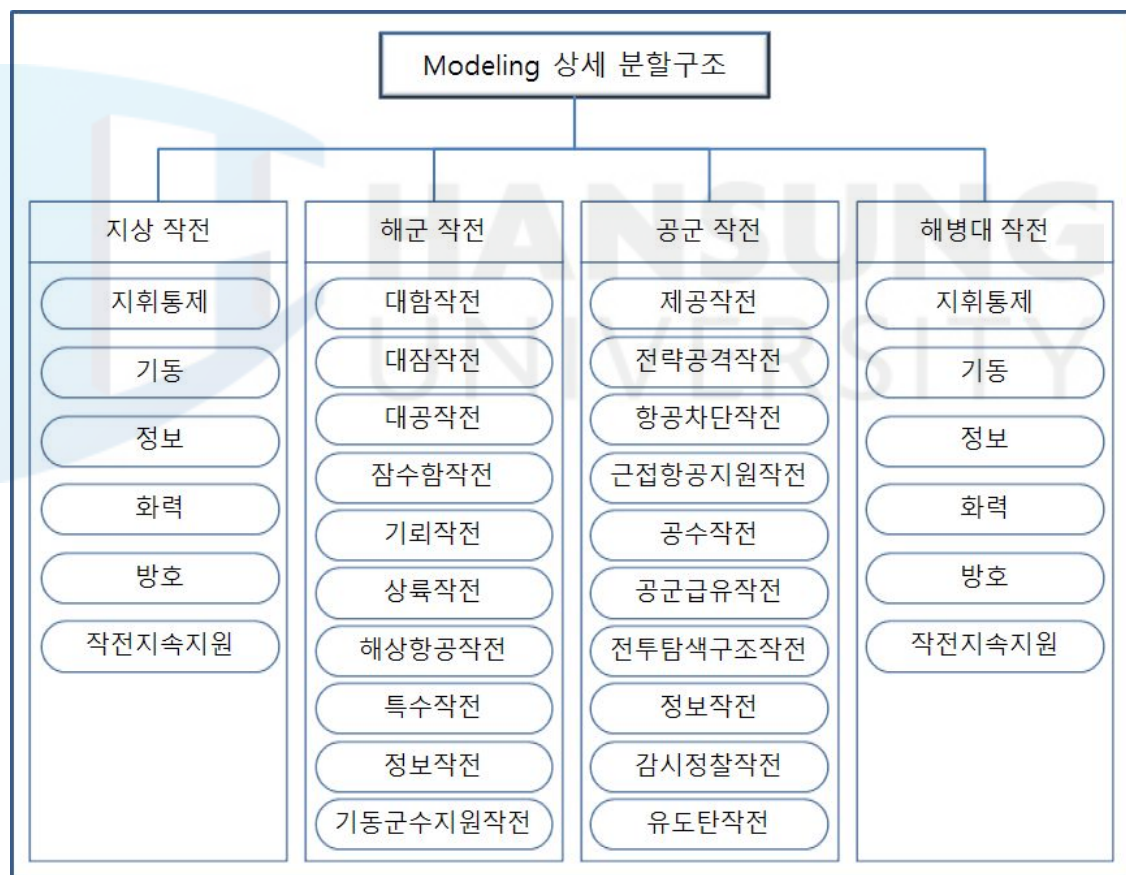


<그림 4-4> 워게임 모의모델 데이터 상세분할 구조

코드는 워게임 모의모델에서 공통으로 사용되는 공통 코드와 각 모델간 연동에 사용되는 M&S용 코드, 실체계인 C4I와 연동을 위해 사용되는 C4I 코드 및 전장환경을 구분하는 지형 코드로 구분한다. 또한 데이터에 대한 검증을 위해 각 데이터 영역별로 검증관리에 대한 분할 구조가 포함된다.

나) 모델링 상세 분할구조

위게임 모의모델은 각 군별 지휘관 및 참모의 지휘절차 연습 및 각 기능 실무자의 전장상황에 대한 각종 조치훈련을 실시하는 것이다. 따라서 지상작전과 해병대작전은 ‘지상군 기본교리’¹⁹⁾에 근거하여 지휘통제, 기동, 정보, 화력, 작전지속지원 및 방호의 6대 기능으로 분할하고, 해군작전은 ‘해군 기준교범’²⁰⁾에 의거 대함작전, 대잠작전, 대공작전, 잠수함작전, 기뢰작전, 상륙작전, 해상항공작전, 특수작전, 정보작전 및 기동군수지원작전의 10대 기능으로 분할한다. 공군작전은 ‘공군교리’²¹⁾에 의거하여 제공작전, 전력공격작전, 항공차단작전, 근접항공지원작전, 공수작전, 공군급유작전, 전투탐색구조작전, 정보작전, 감시정찰작전 및 유도탄작전의 10대 기능으로 상세 분할구조를 제시한다.



<그림 4-5> 위게임 모의모델 모델링 상세 분할 구조

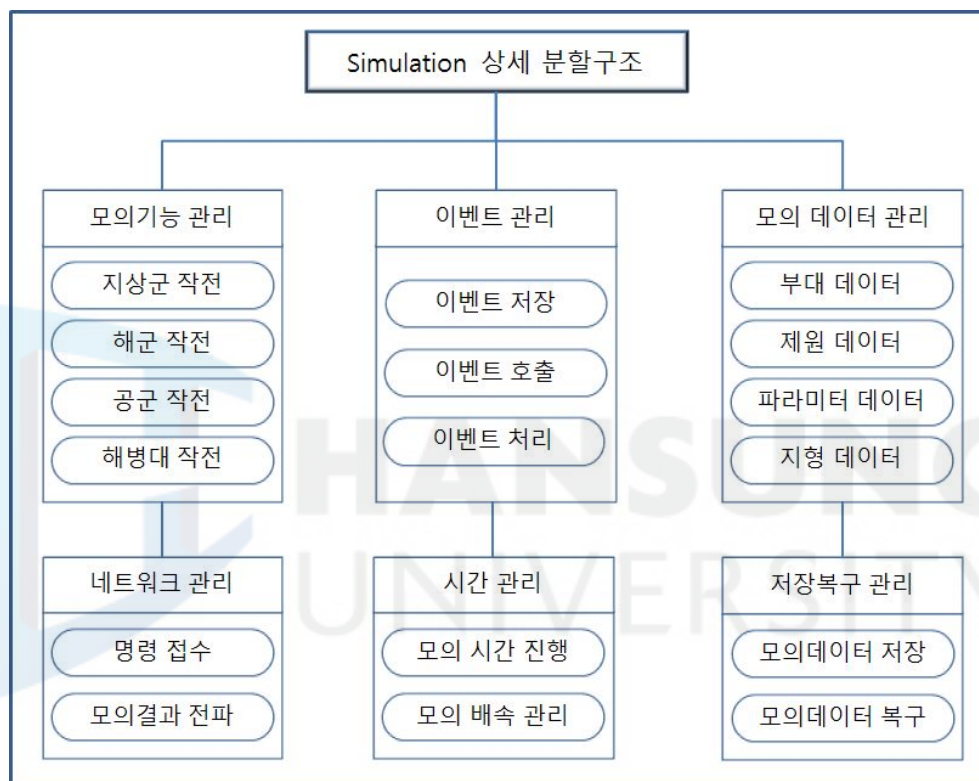
19) 야전교범0, 지상군 기본교리, 육군본부, 2011.11.1

20) 해교3, 해군 기준교범, 해군본부, 2003.3

21) 공군교리3, 공군본부, 2011.12

다) 시뮬레이션 상세 분할구조

위게임 모의모델의 시뮬레이션은 모델링의 상세 분할 구조에서 제시한 지상작전, 해군작전, 공군작전 등을 연속적인 시간의 흐름 속에서 실제 군사작전과 유사하게 실행하는 것으로, 각 작전별 이벤트는 사건발생 시점에 따라 시간 간격이 비동기적으로 생성되고 이벤트의 이산시점에 따라 호출되어 처리 된다.

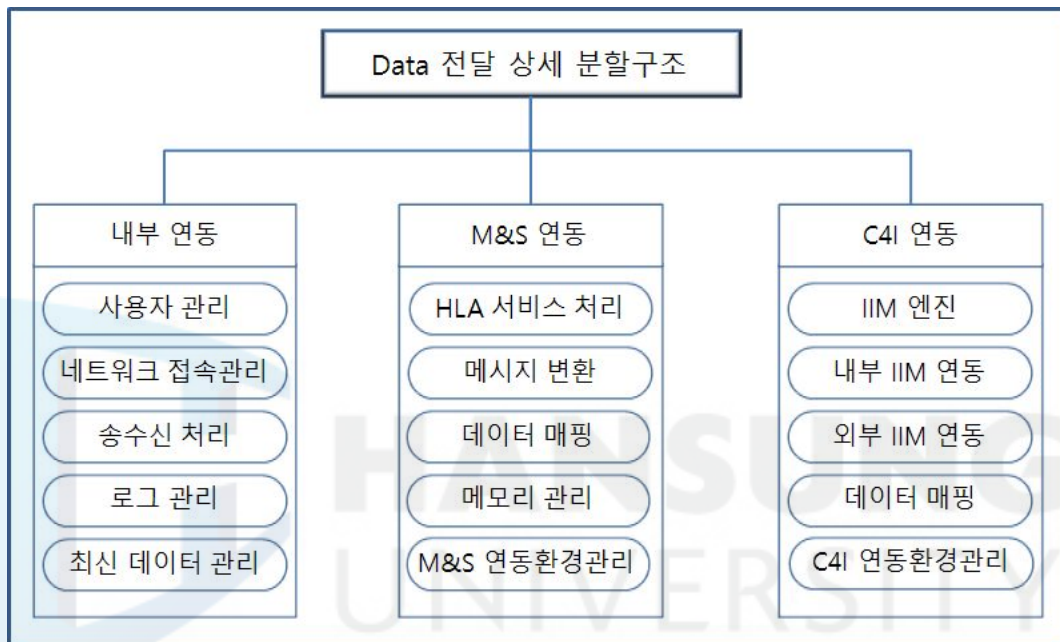


<그림 4-6> 위게임 모의모델 시뮬레이션 상세 분할 구조

시뮬레이션 상세 분할 구조는 모델링의 결과인 지상작전, 해군작전, 공군작전 및 해병대작전의 각 모의기능을 관리하는 모의기능 관리, 이 기능을 이벤트로 저장, 호출, 처리하는 이벤트 관리, 시뮬레이션의 배속과 시간을 관리하는 시간관리, 부대 및 무기체계 제원 등을 관리하는 데이터 관리, 모의 데이터의 저장 복구 관리, 명령접수와 모의결과 결과를 위한 네트워크 관리로 구분할 수 있다.

라) 데이터 전달 및 연동 상세 분할구조

데이터 전달 및 연동은 내부연동, M&S연동 및 C4I연동으로 구분할 수 있는 바, 내부연동은 시뮬레이션과 상황전시 및 명령입출력, 사후분석과의 연동을 처리하며, M&S연동은 위게임 모의모델과 M&S연동체계인 HLA/RTI 페더레이션과의 연동을 처리하며, C4I연동은 통합연동모듈(IIM)을 통해 실체계인 육·해·공군 지휘통제체계들과의 연동을 처리한다.



<그림 4-7> 위게임 모의모델 데이터 전달 상세 분할 구조

내부연동에서는 위게임 모의모델을 사용하는 각 사용자 접속을 통제하는 사용자 관리, 네트워크 접속관리, 명령자료 및 시뮬레이션 결과 자료에 대한 송수신 처리, 로그 관리 및 상황도에 최신 데이터를 제공하는 최신 데이터 관리로 구분할 수 있으며, M&S연동은 객체의 존재 및 특성을 통신하는 객체관리 등의 HAL 서비스 처리, 메시지 변환, 데이터 변환, 메모리 관리 및 M&S연동환경관리로 구분할 수 있으며, C4I연동은 통합연동모듈(Integrated Interface Module) 엔진, 내부/외부 IIM 연동, 데이터 매핑 및 C4I연동환경 관리로 구분할 수 있다.

마) 상황전시 및 명령입출력 상세 분할구조

상황전시 및 명령입출력은 위게임 모의모델의 시뮬레이션 결과를 2D/3D 상황도로 묘사하고, 통제관 및 게이머의 명령 입력 및 모니터링 기능을 제공한다.

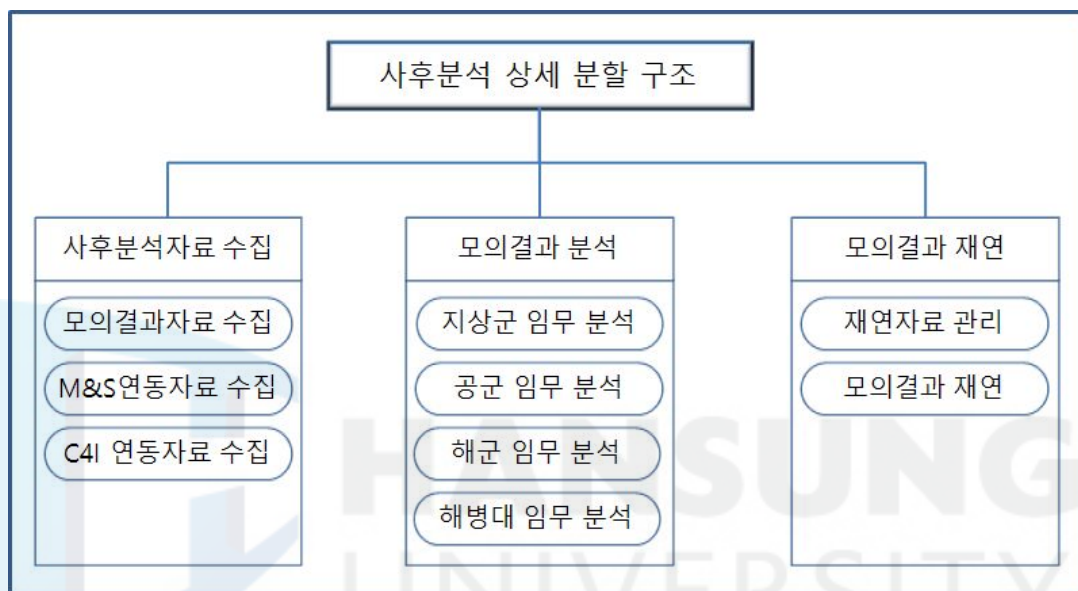


<그림 4-8> 위게임 모의모델 상황전시 및 명령입출력 상세 분할 구조

상황전시 및 명령입출력 상세분할구조는 2D/3D 지도를 도시하는 지도 도시, 지상군, 해군, 공군, 해병대 임무를 전시하는 임무현황전시, 투명도 관리, 전장환경 자료인 지형과 장애물을 도시하는 지형데이터 도시, 사전에 명령을 계획하여 시간의 흐름에 따라 자동으로 처리하는 계획 명령과 연습/훈련간 작전을 수행하기 위한 수시명령을 관리하는 입력명령관리, 시뮬레이션 결과를 전장상황 보고서 및 긴급보고서로 접수하여 관리하는 보고서 관리로 구분할 수 있다.

바) 사후분석 상세 분할구조

사후분석은 위게임 모의모델의 시뮬레이션 결과와 M&S연동체계와의 연동결과, 실체계인 C4I체계와의 연동결과를 실시간으로 수집하여 연습/훈련간 또는 종료된 후에 수집한 시뮬레이션 결과를 지상작전, 해군작전, 공군작전 및 해병대작전별로 분석할 수 있도록 제공하고, 시뮬레이션 결과를 상황도의 임무현황 전시를 활용하여 재연할 수 있는 기능을 제공한다.

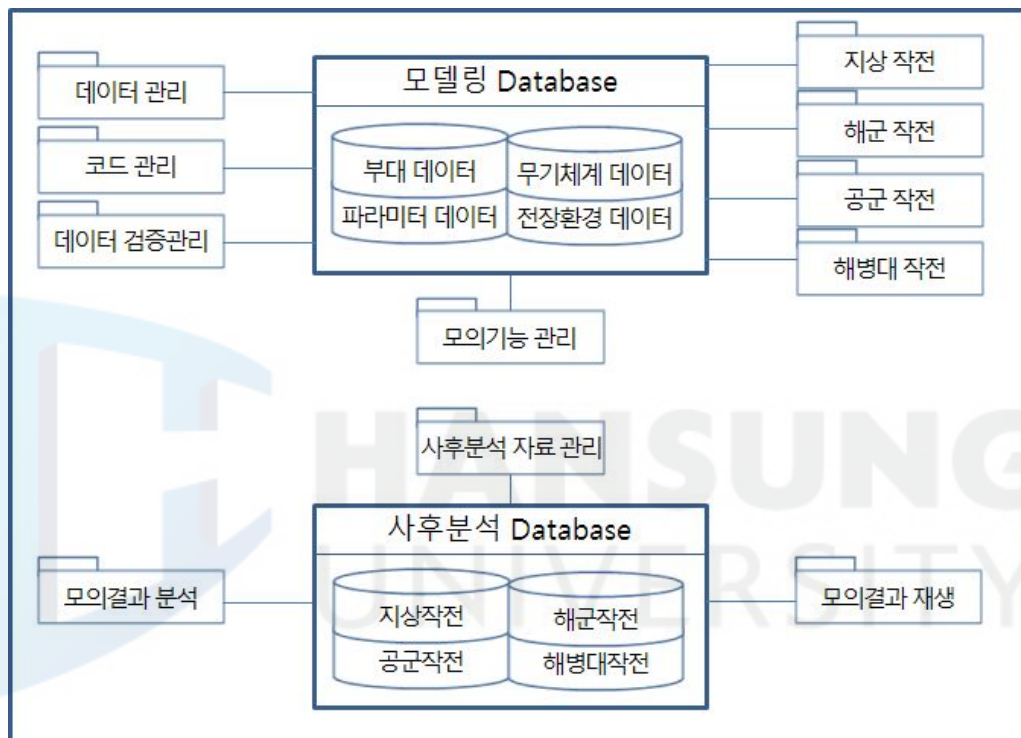


<그림 4-9> 사후분석 상세 분할 구조

따라서 사후분석 상세 분할 구조는 내부연동을 통해 송수신되는 계획명령과 수시명령과 시뮬레이션 결과로 제공되는 임무현황자료와 각종 보고서 등의 모의결과자료, 연합 및 합동연습간 HLA/RTI 페더레이션과 연동되는 M&S연동자료 및 육군 전술지휘정보체계, 전구 합동화력운용체계 등 실체계의 지휘통제를 지원하는 C4I체계와의 연동자료를 수집하는 사후분석자료 수집, 지상군임무, 해군임무, 공군임무 및 해병대임무를 분석하는 모의결과 분석 및 연습/훈련간 처리된 임무현황을 상황도의 임무현황 전시를 통해 재연할 수 있도록 재연자료관리 및 모의결과 재연으로 구분할 수 있다.

2. 사용(Users) 구조

사용 구조는 모듈이나 프로시저, 모듈의 인터페이스에 있는 자원 등의 단위로 구성되며, 이러한 자원들의 의존(depends-on) 관계를 사용이라는 관계로 특화시켜 놓은 것으로서, 본 논문에서는 앞장에서 제시한 위게임 모의모델의 분할구조에서 데이터베이스를 사용하는 모듈이 어떤 것인가를 구분하여 제시하는 것으로 사용구조를 제시한다.

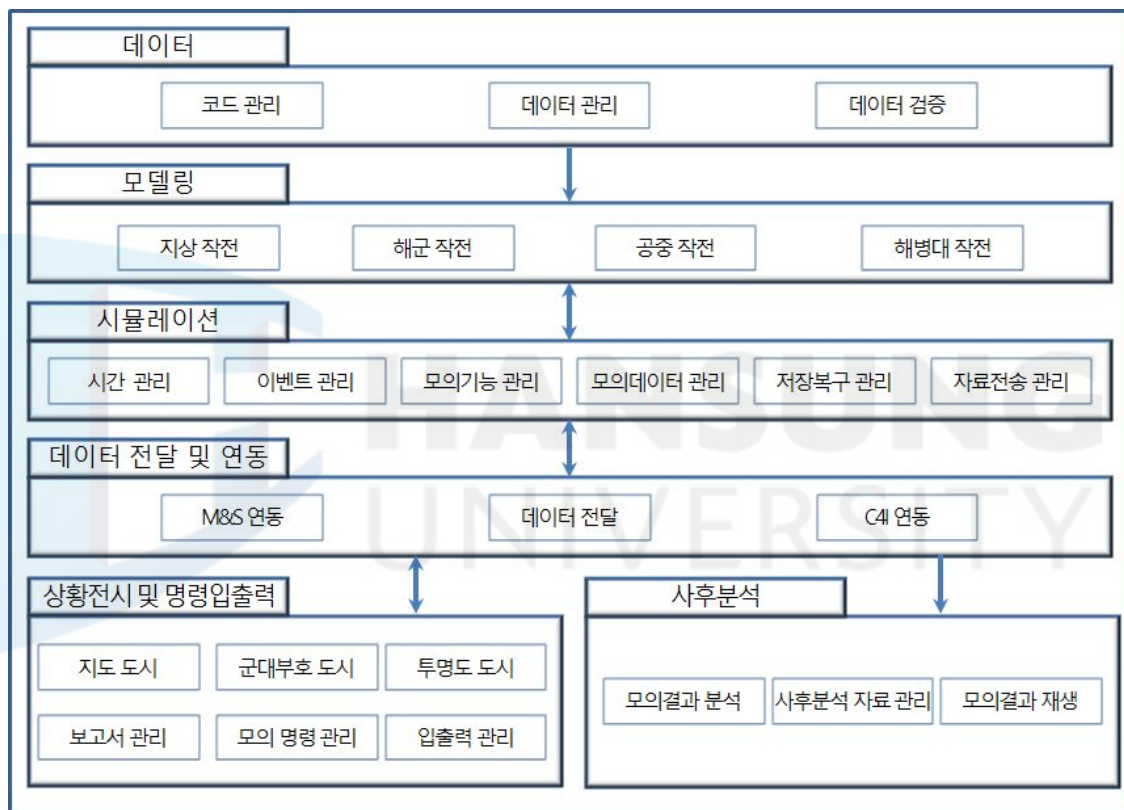


<그림 4-10> 위게임 모의모델 사용 구조

위게임 모의모델에서는 크게 2가지 종류의 데이터베이스가 사용된다. 하나는 부대 데이터, 무기체계 데이터, 파라미터 데이터 및 전장환경 데이터를 구축하여 관리하는 모델링 데이터베이스와 지상작전, 해군작전, 공군작전, 해병대작전 시물레이션 결과를 저장하고 관리하는 사후분석 데이터베이스이다. 모델링 데이터베이스는 데이터 Layer와 모델링 Layer에서 사용하며, 사후분석 데이터베이스는 사후분석 Layer에서 사용된다.

3. 계층(Layered) 구조

계층은 컴포넌트들간에 밀접한 관련성이 있는 기능들의 일관된 집합이라고 할 수 있다. 위게임 모의모델의 계층구조는 앞장에서 제시한 분할구조를 데이터의 흐름과 사용관계에 따라 계층화 할 수 있다. 이러한 계층구조는 소프트웨어를 여러 부분으로 나눌 수 있고, 각 소프트웨어는 공용 인터페이스를 갖춘 하나의 가상기계를 형성하면서 하나의 응집된 서비스 집합을 제공 할 수 있기 때문에 모델 개발시 재사용 및 재활용이 가능하다.

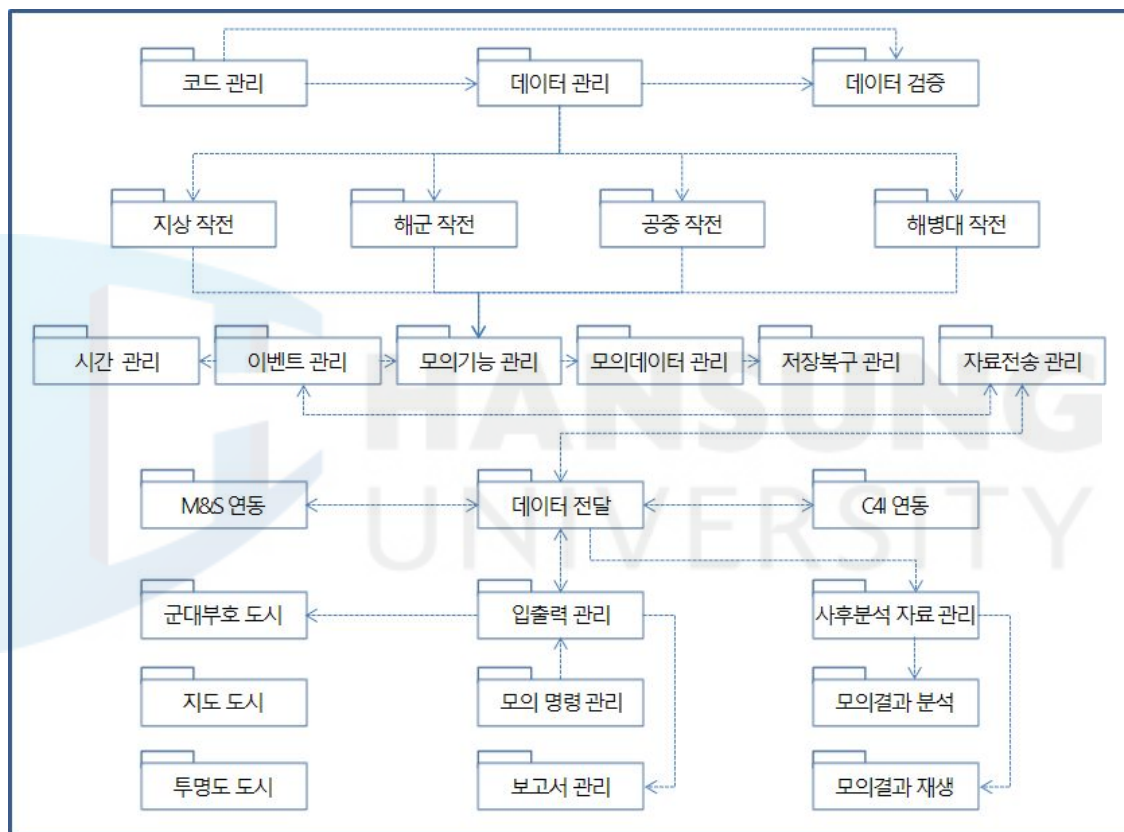


<그림 4-11> 위게임 모의모델 계층구조

위게임 모의모델의 계층구조는 앞장의 위게임 모의모델 시스템 구성에서 제시한 상위레벨 계층구조와 소프트웨어 아키텍처 구조의 분할구조에 기반하여 데이터, 모델링, 시뮬레이션, 데이터전달 및 연동, 상황전시 및 명령입출력, 사후분석의 6개 계층으로 구분할 수 있다.

4. 클래스 구조

클래스 구조는 시스템의 구조를 나타낼 때 사용하며, 객체들의 공통구조와 동작들을 추상화시킨 것이다. 객체는 위게임 모의모델이 수행되는 동안 생성되고 변경되며 소멸될 클래스의 인스턴스이다. 클래스 구조는 객체, 클래스, 속성, 오퍼레이션 및 연관관계를 이용하여 시스템을 나타낸다. 본 논문에서는 위게임 모의모델의 상위계층을 대상으로 계층별 패키지의 연관관계를 클래스 구조로서 제시한다.



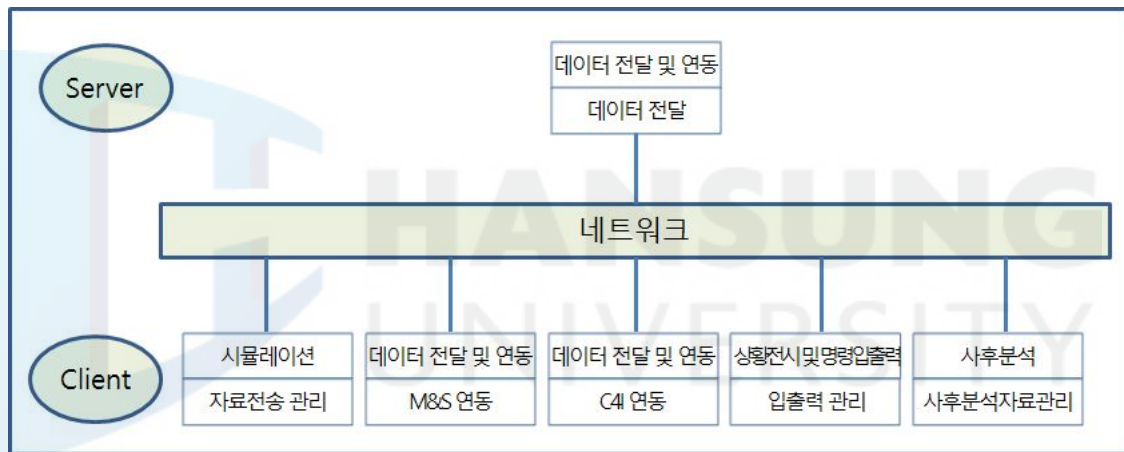
<그림 4-12> 위게임 모의모델 클래스 구조

패키지 구조는 목표에 맞는 소프트웨어 구조를 찾고 서브시스템으로 분할하여 작성되는 것이며, 패키지는 클래스를 의미 있는 관련된 그룹으로 구성하는 매커니즘이다. <그림 4-12>는 위게임 모의모델의 각 패키지가 서로 연관되는 관계도를 제시한 것이다.

제 4 절 컴포넌트와 커넥터 구조

1. 클라이언트 서버 구조

위게임 모의모델은 기본적으로 클라이언트와 서버의 구조를 가지고 있다. 최근들어 대부분의 정보시스템들이 웹 시스템을 기반으로 개발되고 있으나, 위게임 모의모델은 실시간으로 대용량의 데이터를 송수신하면서 처리하는 시스템이므로 처리속도가 시스템의 성능에 매우 중요한 요소로 대두되고 있다. 따라서 대부분의 위게임 모의모델은 처리 속도면에서 다소 유리한 클라이언트와 서버 구조로 개발되고 있다. 컴포넌트는 클라이언트와 서버이고 커넥터는 시스템의 기능을 수행하기 위해 클라이언트/서버가 사용하는 통신규약과 메시지다.

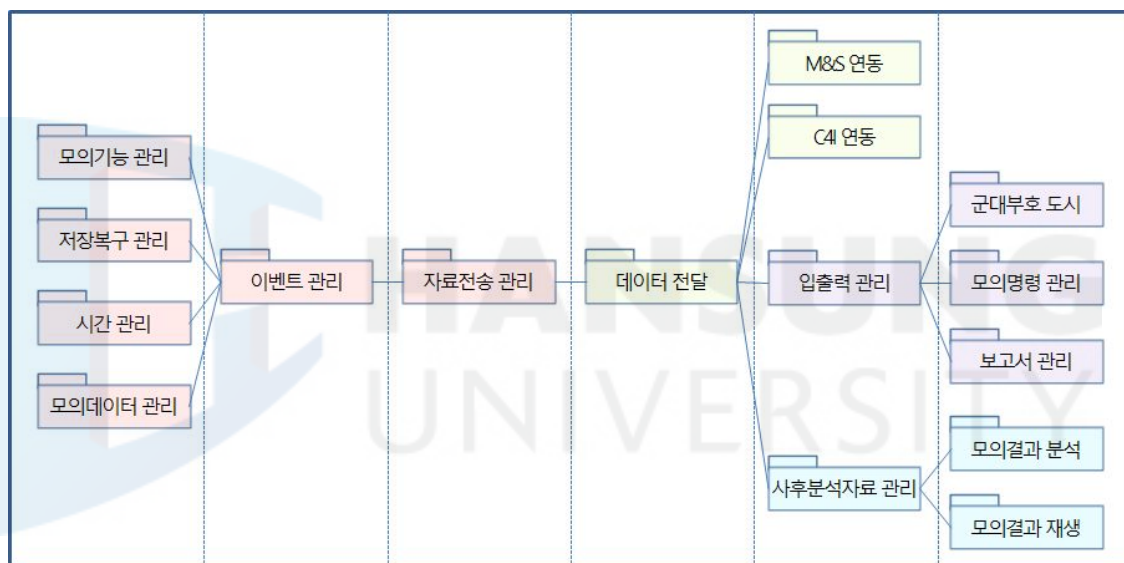


<그림 4-13> 위게임 모의모델 클라이언트 서버 구조

위게임 모의모델은 데이터 전달 및 연동의 데이터 전달 서버가 가장 상위에 위치하여 모든 서브시스템들의 접속을 통제 및 관리하며 데이터 송수신을 처리한다. 시뮬레이션의 자료전송 관리는 작전명령을 수신하고 시뮬레이션 결과를 제공한다. M&S연동과 C4I연동은 관련 연동체계와의 연동자료를 송수신하며, 상황전시 및 명령입출력은 작전명령을 생성하여 송신하고 시뮬레이션의 결과를 보고서 형태로 수신한다. 사후분석은 지상작전 등 시뮬레이션 결과자료인 내부연동자료와 M&S 및 C4I연동자료를 데이터 전달 서버를 통하여 수신한다.

2. 동시성 구조

위게임 모의모델은 시뮬레이션 계층인 이벤트 관리의 통제하에 모의기능 관리에서 모의명령 관리에서 입력된 각종 모의명령과 M&S연동 및 C4I연동 체계를 통해 입력된 각종 연동명령들이 처리되고, 그 결과가 다시 데이터 전달을 통해서 상황전시 및 명령입출력과 M&S연동, C4I연동 및 사후분석자료 관리로 전송되어 게이머 및 연동 사용자에게 가시화되어 전시 및 보고서 형태로 제공된다. 이러한 위게임 모의모델의 구조를 통해 동시성 구조 아키텍트인 병렬 구조의 아키텍처를 설계 할 수 있고, 어느 소프트웨어에서 자원 경쟁이 발생하는지 찾을 수 있다.

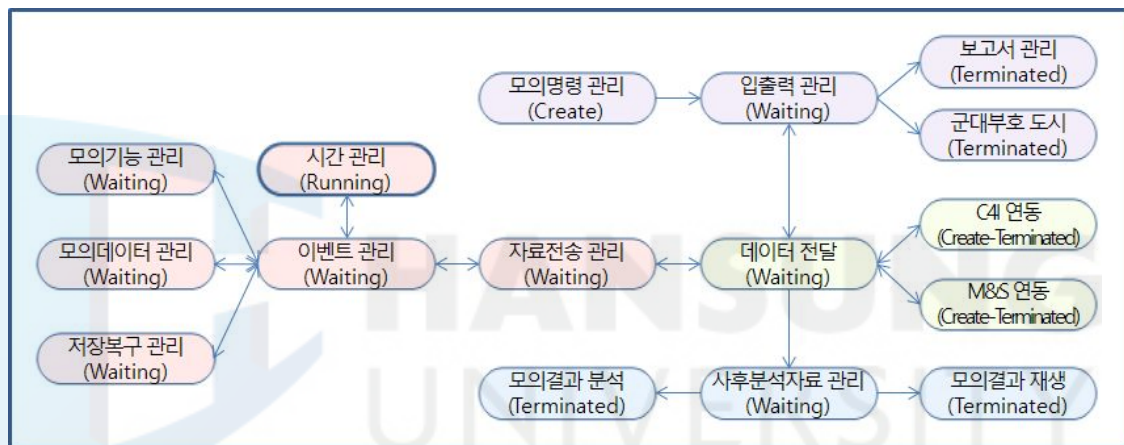


<그림 4-14> 위게임 모의모델 동시성 구조

위게임 모의모델에서 병렬로 처리 가능한 소프트웨어는 시뮬레이션 계층인 모의기능관리, 저장복구관리, 시간관리, 모의데이터관리와 데이터 전달 계층인 M&S연동, C4I연동, 입출력관리, 사후분석자료 관리 및 상황전시 및 명령입출력 계층의 군대부호도시, 모의명령관리, 보고서 관리 및 사후분석 계층의 모의결과 분석과 모의결과재생이다. 자원의 경쟁이 발생하는 지점은 이벤트 관리와 데이터 전달이다. 따라서 이 두 컴포넌트의 처리속도 향상을 위한 품질관리가 결국 위게임 모의모델의 성능에 결정적인 역할을 한다.

3. 프로세스 구조

프로세스 구조는 위게임 모의모델의 실행 관점을 표현하는 동적인 모습을 표현하는 것으로서, 프로세스나 스레드 간의 연결이나 상태 일치, 독립된 행위를 프로세스 구조에 표시하여 설계하는 것이다. 프로세스는 기본적으로 생성(Create), 실행(Running), 대기(Waiting), 종료(Terminated)의 4 가지 상태로 변경되면서 처리된다. 위게임 모의모델도 마찬가지로 모든 작전명령들이 이벤트로 생성되어 실행되고, 작전명령의 실행시간 및 데이터의 흐름에 따라 대기하고, 최종적으로 이벤트의 정보를 가시화하거나 보고서로 그 결과를 제시하면서 종료된다.

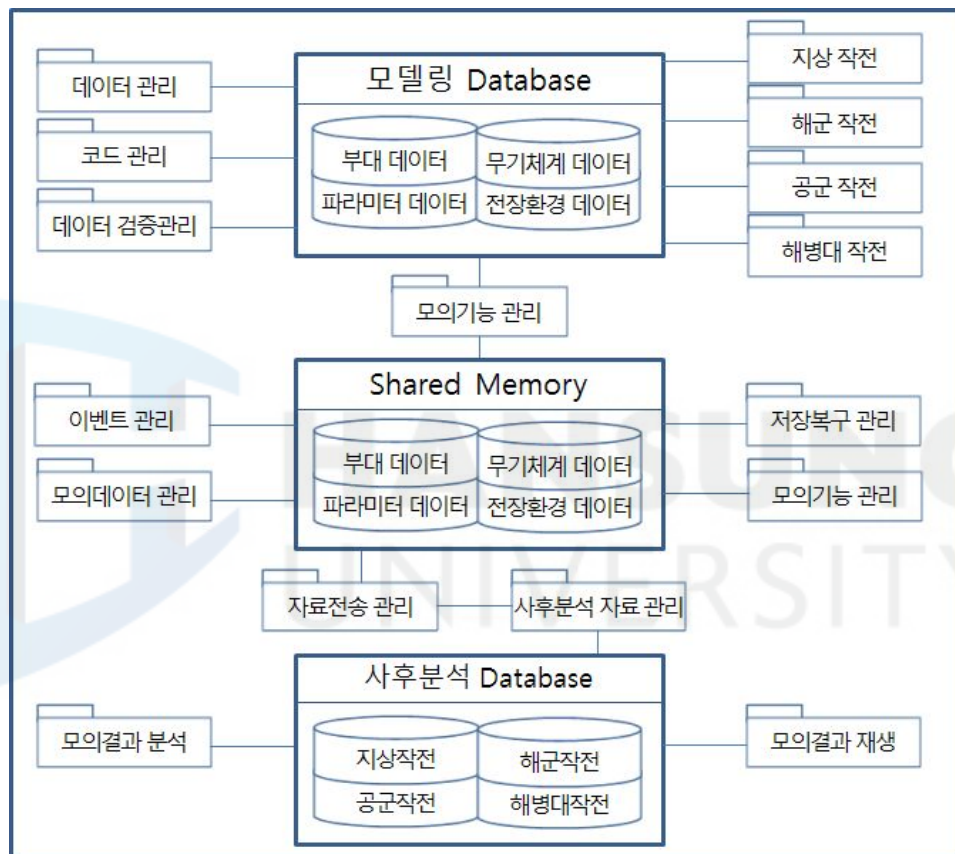


<그림 4-15> 위게임 모의모델 프로세스 구조

이러한 위게임 모의모델의 이벤트 처리 구조를 프로세스 구조로 설계하면 <그림 4-15>에서 제시한 것처럼 모의명령관리, M&S연동 및 C4I연동 컴포넌트에서 이벤트가 생성되고, 시간관리 컴포넌트에서 실시간으로 시간을 통제하면서 상시 실행상태로 실행되고, 보고서관리, 군대부호도시, 모의결과 분석, 모의결과 재생과 C4I연동 및 M&S연동 컴포넌트에서 모의명령에 의해 생성된 이벤트가 가시화 또는 보고서 형태로서 최종 종료하게 된다. 그 외의 나머지 컴포넌트들은 대기상태를 유지하다가 생성, 실행 및 다른 대기상태의 컴포넌트의 요청에 의해 대기상태에서 깨어나 해당 이벤트를 처리한 후 다시 대기상태를 유지한다.

4. 공유 데이터 구조

공유 데이터 구조는 저장소(Repository) 구조라고도 할 수 있으며, 변화가 없는 데이터를 생성, 저장하고 이용하는 컴포넌트와 커넥터로 구성된다. 위게임 모의모델은 크게 세 개의 공유 데이터 저장소로 구성되어 있으며, 따라서 공유 데이터 구조를 적용한 설계는 각 컴포넌트가 어떤 공유 데이터 저장소의 데이터를 생성하고 사용하는지를 보여주는 아키텍처이다.



<그림 4-16> 위게임 모의모델 공유 데이터 구조

위게임 모의모델의 공유 데이터 구조는 사용구조를 확장한 개념으로서 모델링 데이터베이스와 사후분석 데이터베이스 및 시뮬레이션 계층에서 사용하는 Shared Memory 구조로 구분하여 설계할 수 있다.

제 5 절 할당 구조

1. 업무할당 (Work Assignment) 구조

업무할당 구조는 위게임 모의모델 개발팀에게 구현이나 통합 모듈의 개발 책임을 적당히 할당하는 위하여 설계한다. 업무할당 구조는 관리적인 측면뿐만 아니라, 아키텍처적으로도 누구에게 어떤 일을 시킬지 확실하게 결정할 수 있다.

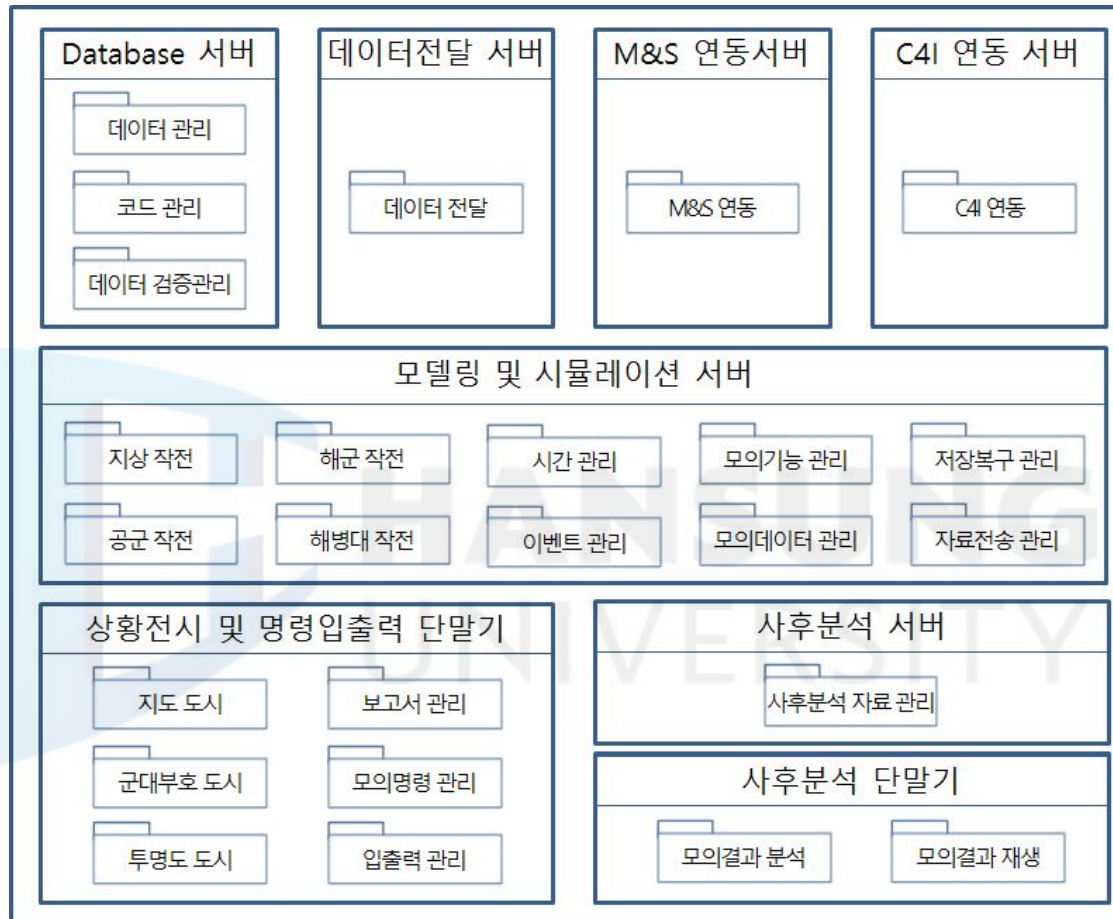
준비 단계	데이터 Layer	데이터 관리	코드 관리	데이터 검증관리
	모델링 Layer	지상작전	해군작전	공군작전
실시 단계	시뮬레이션 Layer	모의기능 관리	이벤트 관리	시간 관리
		모의데이터 관리	저장복구관리	자료전송 관리
	데이터 전달 및 연동 Layer	데이터 전달	M&S 연동	C4I 연동
	상황전시 Layer	지도 도시	군대부호 도시	투명도 도시
		보고서 관리	모의 명령 관리	자료전송 관리
사후 분석 단계	사후분석 Layer	사후분석자료 관리	모의결과 분석	모의결과 재생

<그림 4-17> 위게임 모의모델 업무 할당 구조

위게임 모의모델의 업무는 크게 준비단계, 실시단계, 사후분석단계로 구분하여 업무를 구분할 수 있다. 준비단계는 위게임 모의모델을 통한 연습 및 훈련에서 사전에 부대자료, 무기체계자료 및 각종 연동환경자료 등 연습 시나리오를 준비하는 단계이며, 실시단계는 모델을 통해 가상의 전장환경에서 가상의 부대와 가상의 인원이 실전과 유사한 전쟁을 수행하는 단계이고, 마지막으로 사후분석단계는 지휘관 및 참모 즉 연습 참여자들이 실제로 조치한 각종 상황들의 과정을 분석 및 재연하는 단계이다.

2. 배치 (Deployment) 구조

배치구조는 소프트웨어가 하드웨어/프로세스 혹은 대화 요소에 어떻게 할당되는지 보여주는 것으로서 할당구조의 단위는 소프트웨어와 하드웨어 프로세서, 통신 경로이다. 본 논문에서는 배치의 단위를 하드웨어를 기반으로 하여 각 컴포넌트들의 배치관계를 제시하고자 한다.



<그림 4-18> 위게임 모의모델 배치 구조

위게임 모의모델은 6 개의 서버와 2 개의 단말기로 하드웨어 아키텍처를 설계할 수 있다. 서버는 데이터베이스 서버, 데이터전달 서버, M&S연동 서버, C4I연동 서버, 모델링 및 시뮬레이션 서버 그리고 사후분석 서버로 구성할 수 있으며, 단말기는 상황전시 및 명령입출력 단말기와 사후분석 단말기로 구성할 수 있다.

3. 구현 (Implement) 구조

구현 구조는 소프트웨어 요소가 시스템의 개발환경, 통합환경, 형상관리 환경에서 파일 구조에 어떻게 대응되는지 보여 것으로서, 본 논문에서는 앞에서 제시한 배치구조를 기반으로 하여 서버와 단말기에서 각 컴포넌트를 어떤 OS환경에서 개발 및 구현하여야 하는지 그 방법 및 컴포넌트의 구성 방안을 제시한다.

Database 서버	데이터전달 서버	M&S 연동서버	C4I 연동 서버
데이터 검증관리	데이터 전달	M&S 인터페이스	C4I 인터페이스
코드 관리		M&S연동관리도구	C4I연동관리도구
데이터 관리		M&S연동환경관리	C4I연동환경관리
DBMS			
Linux OS	Linux OS	Linux OS	Linux OS

모델링 및 시뮬레이션 서버		상황전시 및 명령입출력서버	사후분석 서버
지상작전	모의데이터관리	지도 도시	사후분석자료관리
해군작전	저장복구관리	군대부호 도시	Linux OS
공군작전	이벤트관리	투명도 도시	사후분석 단말기
해병대작전	시간관리	보고서 관리	
모의기능관리	자료전송관리	모의명령 관리	
		입출력 관리	
Linux OS		Window OS	Window OS

<그림 4-19> 위게임 모의모델 구현 구조

위게임 모의모델에서 각 서버는 다수의 사용자 접속 및 프로세스를 처리하여야 하기 때문에 Linux OS를 탑재한 서버기반에서 개발하며, 클라이언트는 Window OS를 탑재한 PC기반으로 개발하는 것으로 제시한다. 특히 M&S연동을 위한 서버는 HLA/RTI기반으로 개발하여야 하는 바 HLA/RTI은 현재 Linux OS에서 운용되는 것을 군에서 표준으로 선정하여 사용하는 바 Linux OS에처 개발하여야 하는 당위성이 있다.

제 6 절 위게임 모의모델의 소프트웨어 아키텍처 설계 기대효과

소프트웨어 아키텍처는 프로그램이나 컴퓨팅 시스템에서 소프트웨어 구성요소와 그들이 가진 특성 중에 외부에 드러나는 요소의 특성, 그리고 구성요소들간의 관계를 표현하는 시스템의 구조나 구조체를 말하는 것으로서 설계 프로세스에서 중요한 위치를 차지한다. 특히 위게임 모의모델과 같이 복잡한 시스템을 개발하기 위해서는 설계과정에서 시스템을 바라보는 다양한 관점에서의 뷰 및 설계가 필수적이다. 그러나 지금까지 위게임 모의모델을 개발하면서 시스템 아키텍처는 물론 소프트웨어 아키텍처도 매우 추상적으로 분석 및 설계하여 개발 간 상세내용이나 알고리즘, 데이터 모델링 등의 아키텍처적인 요소를 구체적으로 표현하지 않았다. 그 동안 국내에서는 육군, 해군, 공군 및 합참에서 연습/훈련을 위해 다양한 위게임 모델을 개발하여 왔으나, 위게임 모의모델의 기본적인 시스템 및 소프트웨어 아키텍처를 설계 및 관리하지 않음에 따라 각 모델을 개발하면서 개발 초기에 모델 개발 경험자 즉 위게임 모의모델 전문 아키텍트에 의해 아키텍처 설계를 실시하고, 관련 모델에서 재사용 및 재활용 가능한 모듈 및 컴포넌트를 식별하지 못해 유사한 모듈 및 컴포넌트를 중복하여 개발하는 등 개발 생산성과 재사용성이 매우 저조한 형편이었다. 본 논문에서 제공한 위게임 모의모델의 6가지 계층구조를 기반으로 한 모듈구조, 컴포넌트와 커넥터 구조 및 할당 구조의 소프트웨어 아키텍처는 육군, 해군, 공군 및 합참의 연습/훈련을 위한 위게임 모의모델은 물론 일부 컴포넌트의 배치와 구조를 변경하면 분석/실험을 위한 모델에서도 적용 가능하다.

본 논문에서 설계한 소프트웨어 아키텍처는 비록 위게임 모의모델의 상위계층을 대상으로 한 모듈구조, 컴포넌트와 커넥터 구조 및 할당 구조에 대한 설계를 제공하였지만, 각 계층을 구성하고 있는 컴포넌트간의 동작과 상호작용을 표현하는데 주력하였으며, 이 설계안은 고객, 개발조직 관리자, 마케팅 담당자, 사용자, 유지보수 조직 관리자 등 많은 이해관계자들에게 위게임 모의개발 초기부터 의사소통의 수단 및 개발 방향을 이해하고 협의하는데 많은 도움을 제공할 것으로 판단한다.

제 5 장 결 론

위게임 모의모델은 실제 또는 가상 군사상황을 묘사하기 위하여 둘 또는 그 이상의 적대세력 간에 발생하는 군사적 상황을 게임규칙, 각종 자료 및 절차를 사용하여 군사작전을 모의하는 것으로, 실제 무기체계 및 병력을 기동시키면서 훈련 및 연습을 실시하기 곤란한 현실적인 상황을 감안할 때 그 중요성이 매우 부각되고 있는 실정이다 최근에는 육군, 해군, 공군 및 합참차원에서 전구급에서부터 교전급까지 다양한 위게임 모의모델의 개발 및 각 군의 기능을 상세히 묘사하고 지원할 수 있는 기능모델의 개발이 지속적으로 소요되고 있는 추세이다.

위게임 모의모델은 그 구조가 전구급에서부터 교전급 모델까지 개발구조가 거의 유사한 계층구조를 이루고 있기 때문에 본 논문에서 제공하고 있는 기본적인 계층구조를 대상으로 한 소프트웨어 아키텍처 설계는 연습/훈련을 위한 위게임 모의모델의 개발에 개념을 이해하고 상위 아키텍처를 설계하는데 초석을 제공할 것으로 판단한다.

위게임 모의모델의 6가지 계층 즉 데이터, 모델링, 시뮬레이션, 데이터 전달 및 연동, 상황전시 및 입출력, 그리고 사후분석의 계층은 위게임 모의모델의 사례연구에서도 파악하였듯이 국내 대표적인 위게임 모의모델인 창조21과 태그JOS모델의 구조에도 적용 가능한 구조이다. 특히 아키텍처 구조로서 제시한 모듈구조, 컴포넌트와 커넥터 구조 및 할당 구조는 위게임 모의모델 개발 경험이 없는 아키텍트도 각 컴포넌트의 행위 및 상호작용을 쉽게 내용을 이해하고 적용할 수 있을 것으로 판단한다.

향후 위게임 모의모델의 개발에 좀 더 도움을 주기 위해서 본 논문에서 제시한 컴포넌트들의 세부 아키텍처를 연구하여 제시한다면, 위게임 모의모델의 개발 간 상호운용성 확보는 물론 표준적인 아키텍처를 적용하고 개발함으로써 발생하는 컴포넌트 및 모듈의 재사용성, 재활용성은 극대화 될 것이며, 개발생산성 또한 현재보다 거의 배 이상을 향상될 것이다.

【 참고문헌 】

1. 국내문헌

- 모델링 및 시뮬레이션 이론과 실제, 21세기군사연구소, 2008.6.20
- 국방 위게임모델 목록집, 합동참모본부, 2010.1.22
- 데이터아키텍처 전문가 가이드, 한국데이터베이스진흥센터, 2006
- 소프트웨어 아키텍처 이론과 실제, 렌 베스 외, 2008
- 소프트웨어 아키텍처 기반 설계 모델 및 명세기법 개발, 한국정보보호진흥원,
2007.12
- 소프트웨어 아키텍처 문서화, 폴 클레멘츠 외, 2009
- GoF의 디자인 패턴, Erich Gamma, Richard Helm, Ralph Johnson, John
Vlissides 공저, 2002.9
- 교육참고 25-14 창조21모델 '11모의논리 분석서, 육군교육사령부, 2011.1
- 태극JOS 모의논리서, 한국국방연구원, 2012.12
- 야전교범0, 지상군 기본교리, 육군본부, 2011.11.1.
- 해교3, 해군 기준교범, 해군본부, 2003.3
- 공군교리3, 공군본부, 2011.12

ABSTRACT

A Study on WarGame Model's Software Architecture Design

—Especially by Software Structure Based On The High Level Hierarchical Structure—

Jung, Kyung Sik

Major in National Defense Modeling & Simulation

Dept. of National Defense Modeling & Simulation

Graduate School of National Defense Science

Hansung University

Domestic war game simulation model was developed starting from the ChangJo21 model that was developed from 1996. ChangJo21 model is the division and corps level commander / staff training model. So far, the Joint Chiefs of Staff, Army, Navy, Air Force and Marines are constantly developing war games simulation model in order to practice, support for training, acquisition and experimental analysis.

In order to practice and experiment analysis support for improving ROK warfighting capability of its own, development of new war game simulation model takes being constantly raised.

There wargame simulation models can be classified as a theater, Mission/Battle, engagement, and engineering class, depending on the scale simulations. In addition, operational aspects war game model is divided

into a model for training the model, the model for acquisition and analysis of experiments.

Technology that is utilized in the war game simulation model is modeling techniques, simulation technology, M&S-based technologies.

In recent years, the war game simulation model has been developed by applying the defense CBD methodology for improved productivity and reusability. Actually we develop the respective components in accordance with the model-specific characteristics. Therefore, the productivity is lowered and some components may be duplicated to develop.

In particular, simulation engines is a key component used in all areas wargame simulation model as a common. If we use a joint to standardize the software architecture of the simulation engines, development productivity and reusability of the war game simulation model will be greatly improved.

Representation of the software architecture is expressed the view and structure. Structure is an architectural elements in the software or hardware. A view is consistent set of architectural elements used reading and writing by stakeholders. The view consists of a set of architectural elements the relationship between them. According to the meaning component is showing, architecture structure is largely divided into three groups of the module structure, component and connector structure and the assignment structure.

System block diagram of a war game simulation model is divided into six layers of data management, modeling, simulation, data transfer and integration, status display and command input, after action review. In this paper, I propose a system block diagram based on a six-layer and present a software architecture structure. I identify the software architecture to the module structure, component and connector structure and the structure assignment and present a design solution. Module

structure was presented in the partition structure, using the structure, the hierarchy, class structure. Partition structure was presented in each of the divided structure detail all six layers. Structural components and connectors have been proposed to the client-server architecture, concurrency structure, process structure, and the shared data structure.

I classified the assigned structure as work assignment structure, arrangement and the implementation structure.

System configuration defined a six-layer, software architecture wargame simulation model will contribute to reduced system analysis and design period and to improve development productivity during the war game simulation model development.

【Keyword】 War Game, Architecture Framework, Software Architecture, Reusability, Standardization, Military CBD Method

