

# 한미 지상모델 연동체계 구축방안

*박헌근/김용호(육군교육사)*



# 적응형 페더레이트 미들웨어 (Adaptable Federate Middleware)



육군 교육사 체계분석실  
중령 김용호, 중령 박헌근

-1-

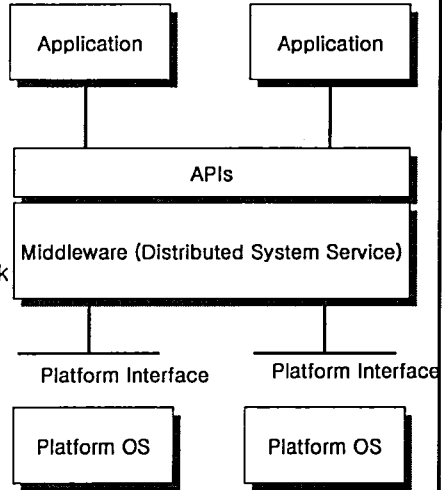
## 순서

- 분산컴퓨팅 지원 미들웨어
- HLA 기반 미들웨어 구조
- Federate 설계 이슈
- Adaptable Federate Middleware
- 차세대 창조21 모델의 Middleware 설계
- 결론

-2-

## 분산 컴퓨팅 지원 미들웨어

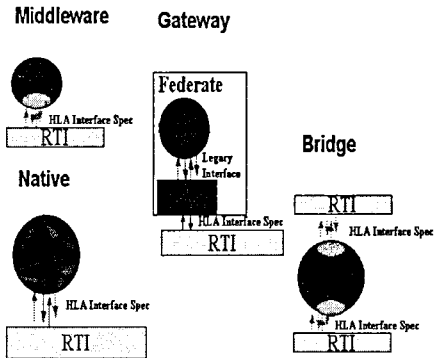
- 분산객체(Distributed Object)
  - 다른 메모리 영역에 존재하며, 접근하기 위해 네트워크를 이용하는 것이 필요한 객체
- 미들웨어는 응용 프로그램과 운영체제 사이에 존재
  - 다양한 이기종의 H/W, OS, Network 존재를 추상화
  - 특정 컴포넌트를 찾기 위한 디렉토리 및 명명서비스 제공
  - 특정 도메인에서 필요한 라이브러리 서비스 제공



-3-

## HLA 기반 미들웨어 구조

- 구조 1 : 시뮬레이션 모델과 RTI사이에 미들웨어 설계
  - 미들웨어에 페더레이션 구축시 통신서비스, 데이터 전송서비스 및 관리서비스 포함
- 구조 2 : 시뮬레이션 모델내부에 연동서비스 관련 코드를 직접구현하는 직접코딩 방식
  - RTI의 HLA서비스가 변경되거나 발전되는 경우 반영하기 곤란
- 구조 3 : 게이트웨어 방식
  - 이미 다른 인터페이스를 포함하고 있을 경우 적합한 방식



- Daniel J. Paterson

-4-

## 페더레이트 설계 이슈

*How do I get my system, simulator or application to be able to participate as a federate within a federation?*

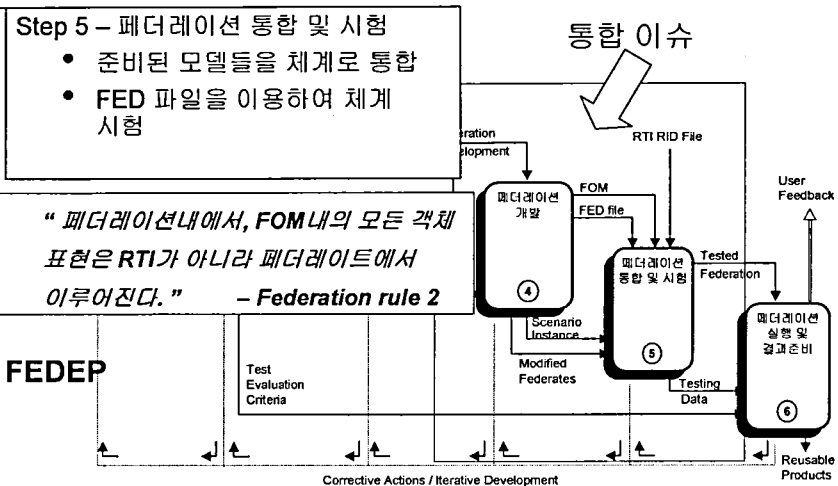


*How can my federate participate in a number of unique federations?*

- Quickly?
- Easily?

- Jane T. Bachman

## 페더레이트 설계 이슈(계속)



## 페더레이트 설계 이슈(계속)

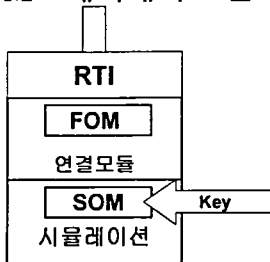
- 다중 페더레이션 지원 필요 (Plug & Play 개념)
  - 코드변경 (Code Modifications)
  - 재컴파일 (Re-compilations)
  - 재연결 (Re-linking)
- 특정 페더레이션의 특성에 빠른 전환 필요
  - 페더레이트는 SOM에 의해 완전히 표현되어야 함
  - 객체모델 매핑을 통해 가능

**해결책:**

- Base Object Models (reusable OM Components)
- "Adaptable Federate" Middleware Approach ←
- Object Model (OM) Mapping

## 적응형 페더레이트 미들웨어

- SOM - 페더레이트 인터페이스 식별

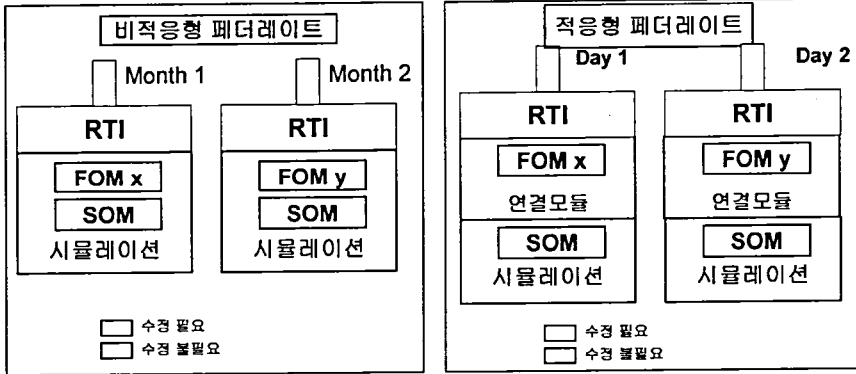


- FOM 개발을 위한 가이드 제공
- 페더레이션 실행시 FOM과 페더레이트간 인터페이스 정의를 위한 기초 제공

SOM 은 다양한 FOM에 매핑하기 위해 사용될 수 있다

## 적응형 페더레이트 미들웨어(계속)

### - 적응형 페더레이트 미들웨어 접근

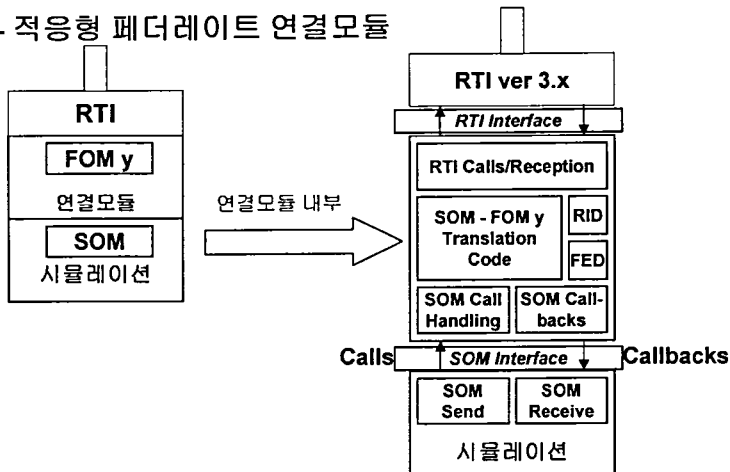


기대효과

- 시뮬레이션 코드변경 최소화
- 재컴파일 및 재링킹 최소화

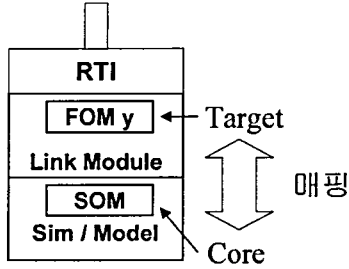
## 적응형 페더레이트 미들웨어(계속)

### - 적응형 페더레이트 연결모듈



## 적응형 페더레이트 미들웨어(계속)

### - 객체모델 매핑

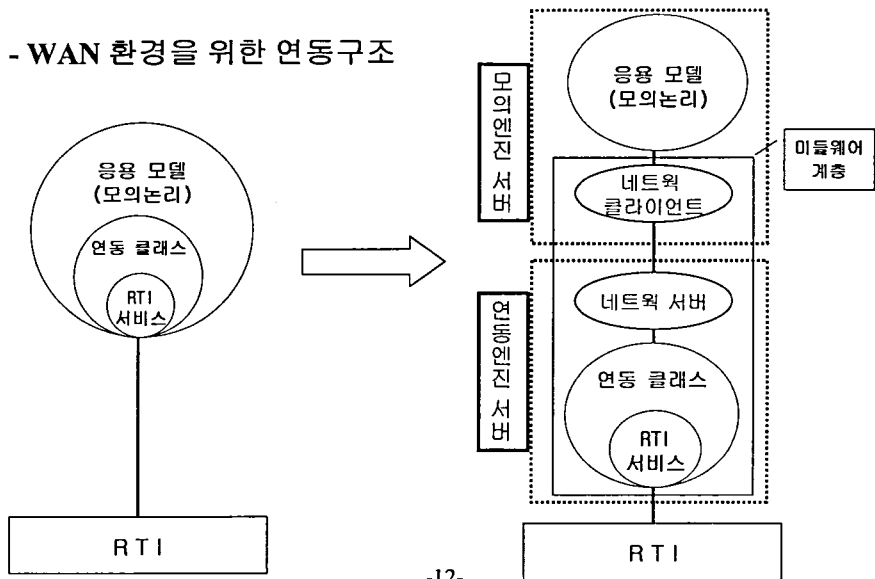


- 객체모델간 구문차이 식별
  - Object / Interaction Linking
  - Attribute / Parameter Linking
- 번역 대상 결정
  - Attribute / Parameter Element Inspection
- 페더레이트 역할 및 책임 결정
  - FEDEP 4.2 성공 보장

**"How can my federate participate in this federation?"**

## 차세대 창조21 모델의 미들웨어 설계

### - WAN 환경을 위한 연동구조

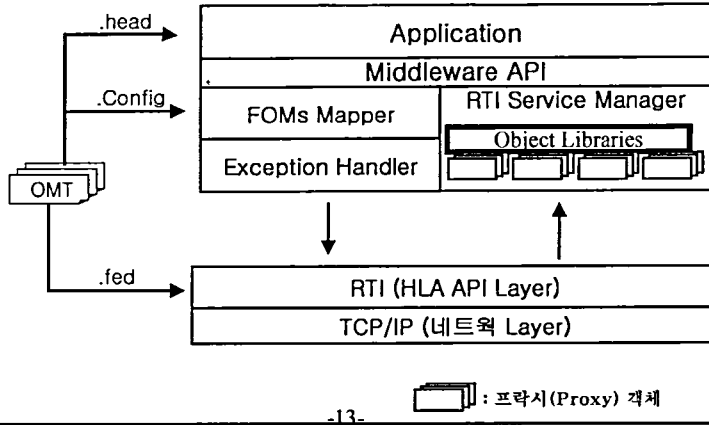




## 차세대 창조21 모델의 미들웨어 설계(계속)

### - 프레임워크 설계

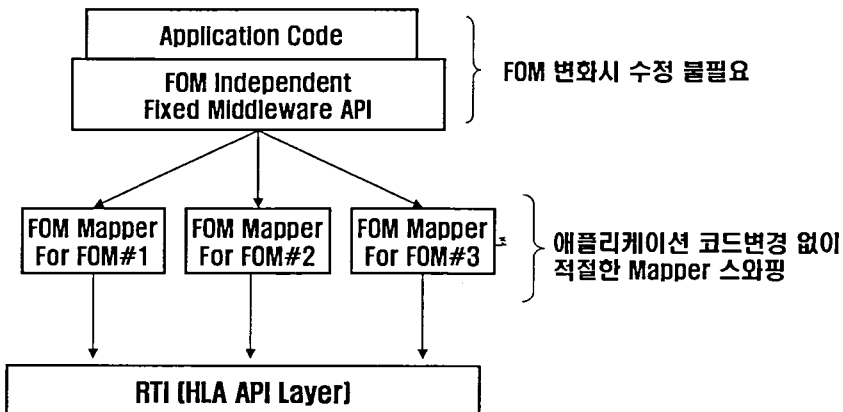
- 미들웨어 방식과 게이트웨이 방식 응용한 Hybrid 방식 적용
- FOM 변경시, 새롭게 추가되는 객체들 유지,관리
- 변경된 FOM을 읽어서 API변화없이 다양한 FOM에 매핑



## 차세대 창조21 모델의 미들웨어 설계(계속)

### - FOMs Mapper 설계

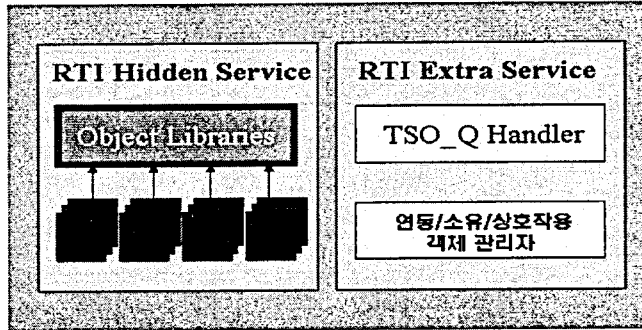
- 시뮬레이션 모델의 SOM과 페더레이션 객체모델인 FOMs를 매핑하는 기능



## 차세대 창조21 모델의 미들웨어 설계(계속)

### - RTI 서비스 관리자 설계

- 페더레이트 개발자들의 추가적인 구현작업 개선방안으로 설계
- RTI 은닉(Hidden) 서비스, RTI 추가(Extra) 서비스로 구성



-15-

## 차세대 창조21 모델의 미들웨어 설계(계속)

### - 다중 FOM과 효율적으로 인터페이스 하기 위한 방법 포함

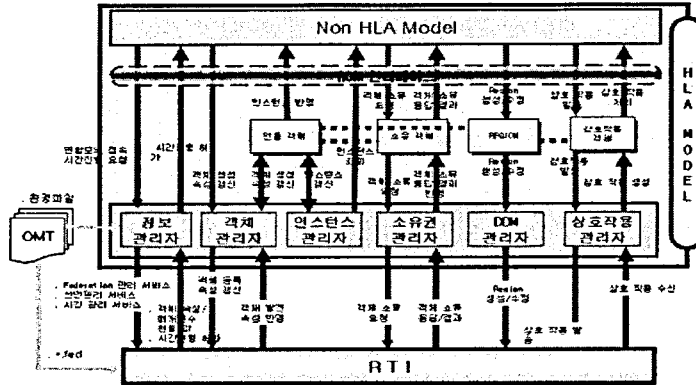
- HLA 객체 클래스와 상호작용의 빠른 생성
- 템플릿 재사용
- 정적 객체 모델 표현
- HLA 객체 클래스와 상호작용 저장
- 원격 페더레이션 객체들을 위한 지역 프록시
- 객체 속성의 자동 갱신과 반영
- 실시간 시간관리
- RTI 서비스의 개별적인 쓰레딩
- 데이터 기록 지원

-16-

## 차세대 창조21 모델의 미들웨어 설계(계속)

### - RTI 추가 서비스 관리자

- HLA API Layer 위에 존재
- RTI에서 수신한 정보 처리능력 향상을 위한 TSO\_Q 핸들러, 6개의 연동정보처리 기능으로 구성

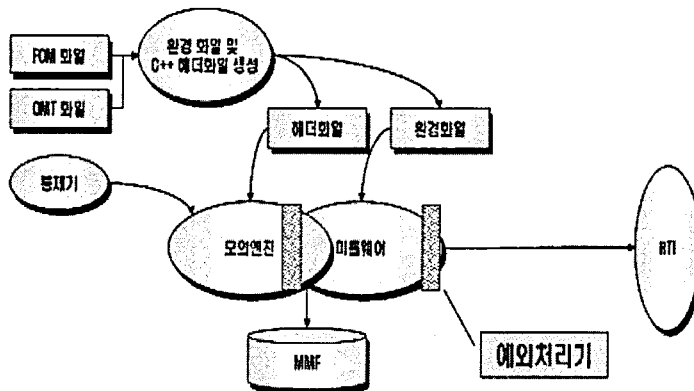


-17-

## 차세대 창조21 모델의 미들웨어 설계(계속)

### - 예외처리기 (Exception Handler) 설계

- 페더레이션 실행상태에서 비정상적으로 종료시에 대한 대비



-18-

## 결 론

- 모든 시뮬레이션 모델은 미래의 잠재적인 페더레이션 멤버로 인식
  - HLA의 기술적 난이도에 얽매이지 않도록 구성
- 시뮬레이션 엔진과 연동엔진을 분리한 병렬 프로세스 연동구조 제안
- FOMs 변경시 응용 프로그램에 좀 더 유연한 서비스 제공 가능한 재구성 미들웨어 제안

*연동에 필요한 모든 데이터 관리의 일관성 유지하여  
데이터의 변화로부터 모델의 수정소요를  
최소화할 수 있는 메커니즘 제공*