

ROCH: 워게임 모의개체 간 역할기반 협력 구현 방안 연구

김정윤 · 김희수* · 이상진

Implementation of Role-based Command Hierarchy Model for Actor Cooperation

Jungyoon Kim · Hee-Soo Kim* · Sangjin Lee

ABSTRACT

Many approaches to agent collaboration have been introduced in military war-games, and those approaches address methods for simulation entity (actor) collaboration within a team to achieve given goals. To meet fast-changing battlefield situations, an actor must be loosely coupled with their tasks and be able to take over the role of other actors if necessary to reflect role handovers occurring in real combat. Achieving these requirements allows the transfer of tasks assigned one actor to another actor in circumstances when that actor cannot execute its assigned role, such as when destroyed in action. Tight coupling between an actor and its tasks can prevent role handover in fast-changing situations. Unfortunately, existing approaches and war-game strictly assign tasks to actors during design, therefore they prevent the loose coupling. To overcome these shortcomings, our Role-based Command Hierarchy (ROCH) model dynamically assigns roles to actors based on their situation at runtime. In the model, "Role" separates actors from their tasks. In this paper, we implement the ROCH model as a component that uses a publish-subscribe pattern to handle the link between an actor and the roles of its subordinates (other actors).

Key words : Collaboration, Actor, Agent, War-game, Role, Publish-subscribe

요약

군 워게임 분야에서 많은 에이전트 협력과 관련된 연구들이 제시되었고, 그러한 연구들은 주어진 목표를 달성하기 위해 존재하는 한 팀을 이루는 모의개체간 협업을 위한 기법들을 다루고 있다. 급격히 변화하는 전장을 모의하기 위해서, 실제 전장 상황에서 빈발하는 임무 인계를 반영하기 위해 하나의 모의개체는 다른 개체의 역할을 인계 받을 수 있어야 하며, 이를 위해 개체는 자신의 과업과 느슨하게 연결되어 있어야 한다. 이 요건을 만족함으로써 모의상황에서 파괴된 개체가 자신의 과업을 수행하지 못할 경우 다른 개체에게 자신의 과업을 전달하는 것을 가능케 한다. 그러나 개체와 과업간의 긴밀한 연결은 긴박한 전장상황에서의 임무 승계를 반영하는 것을 저해한다. 기존의 문제점으로서, 기존의 연구 및 워게임들은 설계단계에서 과업을 개체에 엄격하게 지정함으로써 그러한 느슨한 연계를 어렵게 한다. 이 결함을 극복하기 위해 역할기반명령계층(ROCH) 모델은 실행상태에서 개체의 주변 상황에 근거하여 역할을 동적으로 할당한다. 이 모델에서 역할(Role)은 개체와 과업을 분리한다. 본 논문에서는 하나의 개체와 다른 부하 개체들 간의 역할과의 연결이 긴밀하지 않도록 출판/구독(publish/subscribe)패턴을 활용하여 ROCH 모델을 하나의 컴포넌트로 구현한다.

주요어 : 협력, 개체, 에이전트, 워게임, 역할, 출판/구독 패턴

* 본 연구는 국방과학연구소의 “공통SAF 시뮬레이션 구성 기술” 과제의 일환으로 수행되었다.

Received: 2 November 2015, **Revised:** 20 November 2015,
Accepted: 27 November 2015

*Corresponding Author: Hee-Soo Kim
E-mail: hskim@realtimevisual.com
REALTIMEVISUAL(주)

1. 서론

워게임은 모의개체와 모의개체의 전술과 같은 군 전투 요소를 모델링하고 모의하며 적은 비용으로 예상되는 실제 교전에 참여하는 전투 참여자 및 그들의 전술을 시험한다. 워게임에서 개체(Actor)는 구조적인 요소이고 전술

Table 1. Element Naming in Existing Models

용어	정의
Actor	<ul style="list-style-type: none"> - Agents in agent based simulation area. 이후 본 논문에서 ‘agent’는 다른 의미로 사용한다. - 개별 개체로서 물리적 요소로 상호작용을 한다. (소총수, 전자, 항공기 등) - 개체의 집합으로서 가상의 부대를 표현한다. (소대, 중대, 대대 등)
Tactics	<ul style="list-style-type: none"> - Realized into simulation scenario (reflection of the plan). 모의엔진은 시나리오(tactics)를 개체의 과업으로 변환하며 그러한 변환은 그룹의 전술을 개체의 과업으로 변환하는 방법들이 필요하다. (협업, 구체적으로 임무승계) - Plan (Tactics의 구체화 형태)은 모의에서 참여자와 참여자의 목표를 시간흐름의 순서로 정의한 것이다. 참여자(전투원)은 목표 달성을 위해 타 개체와 협력적으로 행동한다.

은 행위적인 요소로 사용된다. Table 1은 두 가지 주요 용어의 정의를 제시한다.

기존의 위게임들은 집합적 부대 또는 개별적 개체의 전술을 모델링함에 있어서 성공적이었다고 볼 수 있다. 일부는 그러한 모델링을 재사용성이 높은 조립형으로 만드는 모듈화된 모델의 개발을 따르고 있다(Logsdon, Nash, & Barnes, 2008; Ternion Corporation, 2012; VT MÁK, 2012). 그러나 이들 위게임에서 모의개체들은 그들이 수행할 수 있는 전술들과 밀접하게 결합된 형태로 구현된다. 그리고 이것은 특정상황을 모의함에 있어 몇몇 불편함을 가져온다. 예를 들어, 실제 전장에서 전투 참여자가 전투 중에 부상당하는 경우를 고려해보자. 이 경우 부상을 입은 참여자의 임무는 다른 참여자에 인계되어 수행될 수 있다 (역할승계). 하지만, 모의개체와 전술의 밀접하게 결합된 시스템에서는 이와 같은 임무 승계의 기능이 결여되어 있다. 이러한 기능의 결여는 실 전투상황을 표현함에

있어 적절하지 않다. 또한, 임무는 일련의 과업들로서 개체에 결합되어 있기 때문에 모의 상에 필히 지속되어야 할 임무를 가진 개체의 이탈이 발생하면 해당 모의 상에서 임무가 수행되지 않는 결과를 낳는다. Fig. 1의 좌측 부분은 이러한 상황을 묘사하고 있다.

이러한 문제점을 극복하기 위해 우리는 Role-based Command Hierarchy (ROCH) (Kim & Lee, 2013) 모델을 제시하였다. ROCH는 역할(role)이란 개념을 활용하여 개체로부터 과업(전술)을 분리한다. 역할은 모의개체의 전술과 개체의 하부개체(부하, 또 다른 개체)를 연결하는 논리적 연결이며 설계단계에서 정의된다. 모의개체는 실행단계에서 상황에 따라 역할을 통해 하위개체와 동적으로 연결된다. 이러한 아키텍처는 역할이 그 것의 개체에 대한 큰 변경사항이 없이 재사용될 수 있도록 하며 개체 또한 그 것의 임무를 고려할 필요 없이 재사용하는 식으로 조립성, 재사용성, 확장성을 제고한다. 우리는 본 논문을 통해 ROCH 모델에 대한 개략적인 설명과 구현된 인터페이스에 대해 논한다. 그리고 실제 군 모의체계에 통합하여 구현함으로써, 군 모의체계 분야에서 ROCH 모델의 유용성을 검증한다.

2. 관련 연구

2.1 기존 연구 요약

미육군은 교전분석, 교육훈련, 및 획득판단을 포함한 다양한 군 모의분야를 지원하기 위해 OneSAF라고 하는 모의제품라인(simulation product line)을 개발하였다(Logsdon, Nash, & Barnes, 2008). 이는 군 사용자들로 하여금 광범위한 범위의 구성모의 및 가상 시뮬레이터에 대한 솔루션을 활용할 수 있도록 한다. 또한 다양한 상용 모의 아키텍처도 집합적 또는 개별적으로 모의단위의 전술을 모델링함에 있어 매우 성공적이다. 이 중에서 FLAMES와 VR-Force는 충실도 및 해상도 수준에서 광범위한 범위의 물리적 도메인(육상, 해상, 공중, 및 우주공간)을 모델링한다

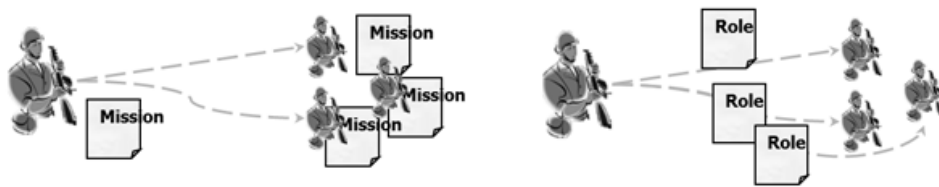


Fig. 1. Tightly coupled missions and seared missions (roles)

Table 2. Element Naming in Existing Models

Modeling Elements	OneSAF	FLAMES	VR-Forces	CSFA
Actor	Actor	Unit	Entity	Unit
Atomic actor	Entity	Unit	Entity	Entity
Aggregate actor	Unit	-	Aggregate Entity	Aggregate Unit
Physical model	Physical Agent / Physical Model	Equipment	Sensor / Actuator	Equipment Agent/ Equipment Model
Behavior logic	behavior Agent / behavior Model	Cognition Model	Controller	behavior Agent
Intra unit communication	Trigger	Specific interface	Port	Trigger
Inter unit communication	Event	Message	Message	Event
Cooperation	behavior Agent	Cognition Model	Controller	Behavior Agent

(Ternion Corporation, 2012; VT MÄK, 2012). 이 체계들은 컴포넌트 기반 개발방법론(component based development)의 개념에 기초하여 부대, 개체, 및 이들의 전술들과 같은 모의요소들이 모듈화된 컴포넌트로 구현했다.

OneSAF, FLAMES 및 VR-Forces는 모두 재사용성이 비교적 우수한 특성을 가지고 있으며 컴포넌트 기반의 모듈화 개념은 개발자들이 M&S체계를 저예산으로 신속하게 개발할 수 있도록 한다. 그러나 실제 전투에서의 전술을 모델링함에 있어 제한이 있으며, 빠르게 변하는 전투 참여자의 임무전환을 효과적으로 반영하지 못한다(예를 들어 하나의 개체는 임무 중에 파괴되어 임무에서 제외되는 경우로서 대개 개체가 제거된 임무만 남겨지게 된다).

이외에도 Hierarchical Agent Control(HAC) (Atkin, Westbrook, & Cohen, 2001), Commander Model(CB) (Vakas, Prince, Blacksten, & Burdick, 2001), Command-based Multi-Agent System(CMAS) (Song, & Yang, 2006), Tactical Team Behavior (TTB) (Bisht, Malhotra, & Taneja, 2007), Agent-Group-Role(AGR) (Ferber, Gutknecht, & Michel, 2004) 등, 기타 관련 연구에서도 유사한 문제점을 볼 수 있다. 각 모델은 집단 단위의 모의개체 전술을 표현할 뿐만 아니라 모의개체가 전술 수행을 할 수 있도록 한다. 그러나 전술이 모의개체와 밀접하게 결합되어 있으므로, 개체와 개체의 전술을 재활용하기에 곤란하게 되어 있다.

2.2 공통 모델(Common Model)

본 연구는 네 개의 모의 아키텍처를 조사하였으며 각 아키텍처 별로 유사한 요소들을 구분하였다. 이 아키텍처

들 중 세 가지는 비중 있는 위치를 점하고 있는 OneSAF, FLAMES, 및 VR-Forces이며 이들은 군사영역 또는 상업적 영역에 적용될 수 있다. 나머지 또 하나의 아키텍처는 Composable Software Framework Architecture(CSFA) (Petty, Kim, & Byun, 2014)로서 조립성 제고를 목표로 한 모듈화 모의 개발을 목적으로 국방 연구과제의 일환으로 개발한 것이다. CSFA는 한국육군의 여건을 반영한 OneSAF의 한국군 버전이라고 볼 수 있는, 벤치마크의 결과물로 간주될 수 있다. OneSAF의 경우 반 테러활동과 같은 비정규전을 포함하는 세계적 규모의 군사작전을 반영하기 때문에 그것의 광범위한 과업(행위) 목록은 한국 상황의 토착적 여건에 적절하지 않다.

이 조사의 결과로서, 본 연구는 이상의 아키텍처 요소들을 비교하고 모델링 요소들로 유사성을 바탕으로 분류함으로써 Common Model을 Table 2와 같이 도출할 수 있었다. 따라서 Common Model은 기존의 모델들과 그것들을 구성하는 요소들의 평균화 분류로 간주될 수 있다. 이들 아키텍처들은 컴포넌트 기반의 소프트웨어 제품들이기 때문에, 많은 측면에서 동일한 분류 내의 대부분의 요소들은 유사한 개념이거나 다른 이름을 가진 동일한 특성을 가지고 있다.

일부 소프트웨어 아키텍처의 요소들은 구조적 요소와 행위적 요소로 구분될 수 있다. Fig. 2는 조사로부터 도출된 Common Model의 요소들을 보여준다. 주요 요소들은 모의개체(Actor), 전장 시나리오(Warfare Scenario), 임무(Mission), 과업(Task), 및 이벤트이다. 많은 명칭은 모의개체(Actor)와 행위(Behavior)와 같이 주로 OneSAF로부터 차용하였다. 이 모델에서, 전장시나리오(warfare scenario)

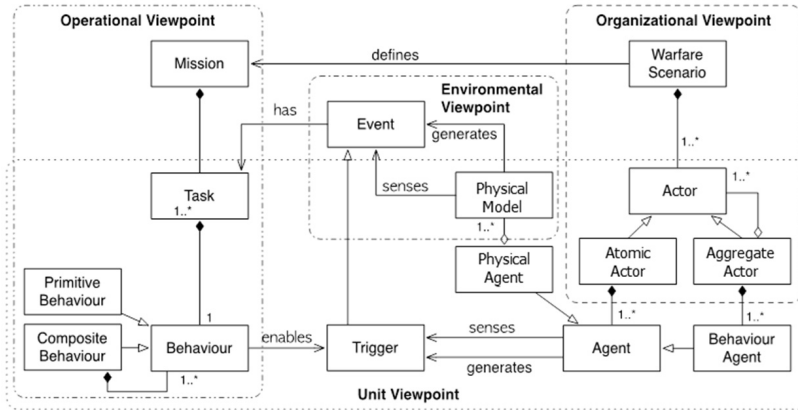


Fig. 2. Common Model from existing models

는 다양한 모의개체를 포함한다. 이것은 모의 상 더 이상 쪼개질 수 없는 것으로 간주되는 기본개체(Atomic Actor) 나 다른 집합개체(Aggregate Actor)로 구성되는 형태로 청군 또는 홍군으로 구성되는 조직 정보를 포함한다. 전장시나리오는 또한 과업(Task)으로 구성된 임무(Mission)로 정의된다. 개체는 분리되지 않거나 집합적일 수 있다: 기본개체(전장에서 개체)는 하부개체를 가지고 있지 않으며 물리에이전트 또는 행위 에이전트(physical/behavior agents)로 구성되는데, 반면에 집합개체(또는 부대 조직)는 개체의 집합을 포함하며 물리에이전트 없이 행위 에이전트만으로 구성된다. 상위개체와 하부개체 간의 관계는 지휘라인을 반영한다. 개체는 에이전트를 포함하고 있는데, 이는 컴포넌트의 인터페이스에 해당하며, 물리 에이전트, 행위 에이전트로 구분될 수 있다. 물리 에이전트는 개체의 센서나 기동, 무기과 같은 개체의 파트를 표현한다. 행위 에이전트는 해당 개체의 행위를 결정한다. 개체 내부에서 에이전트는 트리거(Trigger)를 통해서 다른 에이전트와 소통을 하게 된다. 개체는 물리모델과 물리에이전트(Physical Models /Agents)를 통해서 환경의 제반사항을 탐지하거나 변화를 일으킨다.

Common Model의 행위적 요소들은 행위(Behavior)와 과업(Task), 임무(Mission), 이벤트(Event), 트리거(Trigger)가 있다. 행위는 복합 또는 기본 행위(Composite / Primitive Behavior)가 될 수 있다. 복합행위는 정렬된 다른 행위들로 구성된다. 기본행위는 개체의 행위를 개시시키기 위해 트리거를 생성함으로써 기본행위에 적절한 행위 에이전트를 제어한다. 과업은 이벤트의 형태로 개체에 전달된다. 과업은 행위의 조합으로 정의된다. 모의개체는 행위를 수

행함으로써 과업을 수행한다. 임무는 개념적인 목표로서 개체가 반드시 수행해야 하는 것이며 일련의 과업들로 구성된다. 임무는 주로 개체가 수행하는 장기적 계획을 형성한다. 마지막으로 트리거는 개체의 외부로부터 포착된 이벤트의 다른 이름이며 개체의 행위를 개시하게 한다. 행위가 이벤트에 반응한다는 것은 예를 들면, “공격을 받다”라는 환경의 변화가 이벤트 형태로 전달이 되어서 개체 안으로 들어가게 되면 개체 내에서 트리거로 사용되는 것으로 볼 수 있다. 모의가 개시되면, 각 에이전트는 자신이 관심을 갖는 이벤트들을 등록한다. 그런 다음 만약 해당 이벤트가 발생하면 모의 엔진은 그 이벤트에 대해 등록된 에이전트에게 이 사실을 통보한다. 최종적으로 에이전트는 발생한 이벤트에 따라서 행동을 취하게 된다.

Common Model은 Fig. 2와 같이 점선으로 둘러 친 네 가지의 다른 측면으로 볼 수 있다: 작전(Operational), 조직(Organizational), 환경(Environmental), 부대(Unit)가 각각의 측면이다. 작전측면은 실제 전장에서 수행되는 전술을 의미하며, 모의의 행위적 요소를 의미한다. 조직측면은 교전에 참여하는 전투원이며 모의의 구조적 요소를 의미한다. 환경측면은 단순히 환경과 개체 간의 상호작용을 의미한다. 이벤트는 물리모델을 통해서 개체로 전달되며, 물리모델은 탐지센서, 화기, 기동성, 또는 통신장치 등과 같은 환경의 변화를 탐지하거나 환경에 변화를 초래하는 여하한의 매개체를 표현한다. 부대측면은 개체의 조합을 의미하며, 개체의 조합과 수행할 임무의 측면에서 바라본다.

2.3 기존 모델의 취약성

Common Model의 제한사항은 역할 승계와 같은 특정

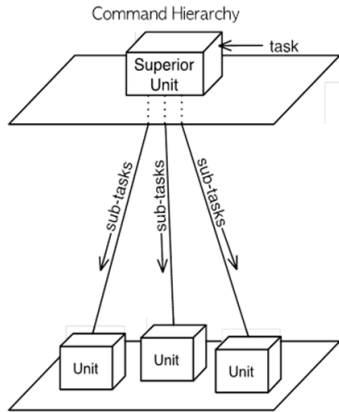


Fig. 3. Command Hierarchy in Common Model

상황을 반영함에 있어 어려움이 있다는 것이다. 전장시나리오와 임무에서 개체와 그것의 과업이 서로 뒤엉켜있기 때문에 개체나 과업 단독으로 재사용하기가 매우 곤란하다. Fig. 2에서 보인 바와 같이, 개별 개체는 자체의 임무를 가지고 있는데, Fig. 3과 같이 표현할 수 있다. Fig. 3에서 상급 부대(Superior Unit)는 과업(task)를 부여받을 때 각 부대(Unit)도 상급 부대로부터 상급 부대의 과업을 구성하는 하부과업(subtask)을 부여받는다. 일반적으로 과업과 하위 과업은 설계단계(시나리오 작성 시)에서 정의되는데, 상급 부대의 과업은 그것에 포함된 하위 부대의 과업을 사용하여 어떻게 달성할 것인가를 고려하여 작성된다. 결과적으로 과업과 하위 과업은 각각을 수행하는 개체와 밀접하게 결합된다.

Common Model의 구조 상, 과업은 다른 개체의 과업과 교체될 수 없는 구조이기 때문에, 만약 그 과업을 가진 개체가 모의로부터 파괴 등의 이유로 제거된다면 그 개체의 과업 또한 제거되게 된다. HAC(Atkin, Westbrook, & Cohen, 2001), CM(Vakas, Prince, Blacksten, & Burdick, 2001), 그리고 TTB(Bisht, Malhotra, & Taneja, 2007)도 모두 컴포넌트의 조립에 대해서는 충분한 고려를 하지 않은 전술의 모델링에 초점을 맞추고 있다. CMAS(Song, & Yang, 2006)과 AGR(Ferber, Gutknecht, & Michel, 2004)은 모의개체가 수행하는 역할들 간의 상호작용으로서 모의개체의 전술을 명시할 수 있으나 그 역할들로부터 모의개체를 분리하기 위한 기법을 명확히 제공하지는 않는다. 이러한 긴밀한 결속(tight coupling)은 상위 모의개체(부대)는 하위 모의개체에 또는 그 반대방향으로 의존적이어야 한다는 것을 의미한다.

3. 역할 기반 명령 계층(ROCH) 모델

Fig. 4는 ROCH 아키텍처의 기본개념을 설명한다. ROCH의 목적은 긴밀한 연계의 단점을 극복하기 위한 목적으로 모의개체로부터 전술요소를 분리하는 것이다. ROCH에서 가장 특이한 점은 역할(Role)이다. 이는 가상의 자리로서 실제 전장에서의 하위부대를 가리키며, 하위부대와 역할의 연계는 설계단계(시나리오 작성단계)에서 정의되지 않는다. 연계는 실행단계에서 이루어지며, 이 때 전술적으로 행위를 수행하는 상위개체에 포함된 하위개체와 하위개체가 담당할 전술이 연결된다. 하부과업(Subtask)은 상위 모의개체의 과업의 목표를 달성하기 위해 기여하는 하위 모의개체의 행위로 볼 수 있다. 역할(Role)은 역할을 소유하는 모의개체가 수행해야 하는 하위과업(Subtask)의 조합으로 이루어진다. 따라서 모든 하위과업의 조합은 상위 모의개체의 전술로 간주될 수 있다.

ROCH는 메타모델과 프레임워크(meta-model and a framework)의 두 가지 부분으로 구성되어 있다. 과거의 역할기반 연구들에서 역할은 일반적으로 모의개체의 설계단계에서 하드코딩되는 형태로 구현된다(Xu, Zhang, & Patel, 2007; Cabri, Leonardi, & Zambonelli, 2003; Becht, Gurzki, Klarmann, & Muscholl, 1999; Hahn, Madrigal-Mora, & Fischer, 2009). 이 부분이 본 연구와의 차이점이라고 볼 수 있는데, 본 연구에서는 XML을 활용하여 역할과 모의개체, 과업과 이들 사이의 관계를 정의하며, 이를 프레임워크가 읽어서 해석하여 처리할 수 있도록 한다. 메타

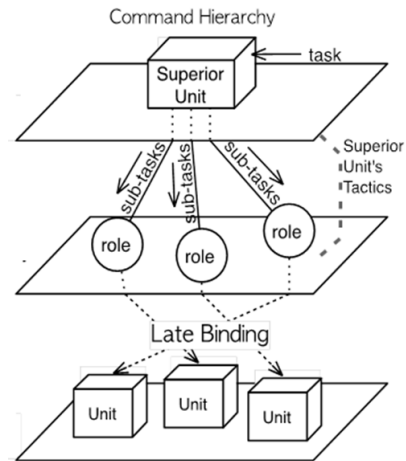


Fig. 4. ROCH Architecture

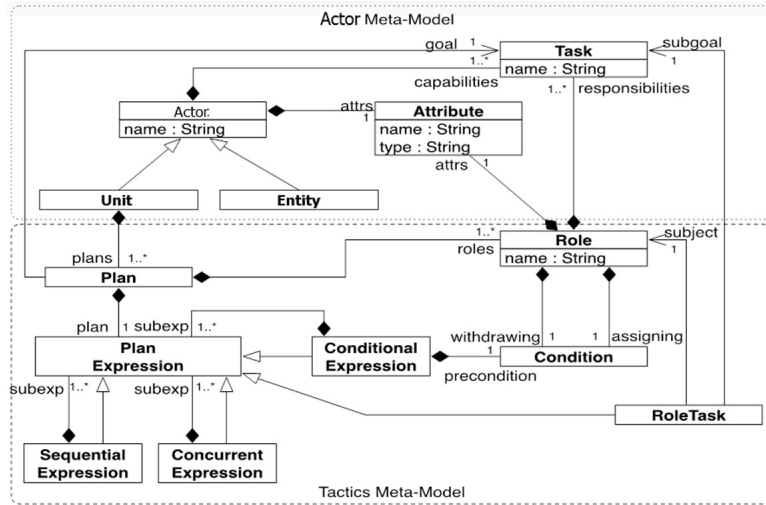


Fig. 5. ROCH Meta Model

모델은 모의개체의 조건을 모르는 상태에서 사용자에게 의해 모의개체의 전술이 자유롭게 변형될 수 있도록 하는 것을 허용한다. 프레임워크는 런타임에서의 상황을 고려하여 동적으로 역할을 할당하는 메커니즘을 이용해 모의개체의 계획을 수행시키는 기능을 제공한다.

Fig. 5는 UML로 표현된 메타모델이다. 메타모델은 모델기반 아키텍처(Model Driven Architecture)(Object Management Group, 2012)와 앞에서 설명한 Common Model의 내용에 기초를 두고 있다. 메타모델은 액터 메타모델(Actor Meta-Model)과 전술 메타모델(Tactics Meta-Model)로 구성된다. 액터 메타모델에서 Attribute는 액터의 상태를 구성하는 변수이며 과업(Task)는 액터가 수행하는 과업이다. 전술 메타모델은 모의개체와 그것의 하위개체가 명시된 계획을 표현한다. 계획(Plan)은 역할과 계획표현(Plan Expression)으로 구성된다. 역할은 ROCH에서 특이한 특성을 가진 존재로서 속성과 과업(수행해야 할 책임)을 포함하며 두 가지의 조건(할당조건과 철회조건)을 가진다. 역할은 모의개체가 수행해야 할 요건으로 간주될 수 있다. 하위 모의개체(Fig. 4의 Unit)가 역할을 받기 위해서는, 그 개체는 역할에 포함된 과업을 수행할 속성 및 역량(Capability)을 갖추어야 한다.

메타모델은 역할의 조건을 정의하는데 이 조건은 상위개체를 위한 역량들이고, 상위개체는 필요한 역량을 갖춘 하위개체들 중에서 역할을 수행할 모의개체를 동적으로 선택하게 된다. 즉, 할당 조건(Assigning Condition)은 하

위개체가 할당된 역할을 수행할 수 있어야 함을 요구한다. 철회조건은 어떤 할당된 역할을 수행하고 있던 하위모의개체가 더 이상 수행할 수 없다는 조건을 의미한다. 구현에서는 아키텍처에서 철회조건을 생각했는데, 그 이유는 사실상 할당조건을 만족시키지 못하는 상황과 동일하기 때문이다. 그러나 철회조건은 ROCH의 주요한 개념이기 때문에 편의상 ROCH 메타모델에서는 철회조건을 계속 유지하기로 한다. 계획은 상위 모의개체에 의해 달성되어야 하는 것으로 일련의 하위과업으로 구성되어 있다. 계획은 모의 사용자나 상위 모의개체가 할당한 과업을 수행하기 위해 일련, 병행, 조건(sequential, concurrent, conditional)으로 표현되거나 또 다른 Role Task expression (다른 표현)으로 표현된다. 역할과업(role-task)은 역할을 맡은 개체가 수행해야 하는 과업을 표현함에 있어 가장 기본적인 표현이다.

프레임워크(Framework)는 Fig. 6과 7에서 보여주는 바와 같이 ROCH의 또 다른 중요한 부분이다. 프레임워크의 주요 기능은 Fig. 6에서와 같이 상급부대(Unit A)의 행위 에이전트 아래에서 계획을 실행하는 것이다. 상태통보기능(State Informing Function)은 모의개체가 자신의 역량과 속성에 대한 정보를 상급부대에 알리는 것이다. 이 기능은 모의개체가 그것의 상급부대에 참여할 때 또는 이미 참여한 상태에서 자신의 역량과 속성에 변화가 있을 때 호출된다. 모의개체와 이들의 역할 간에 관계를 다루기 위한 출판/구독(publish/subscribe)패턴(1~3)에서, 상급

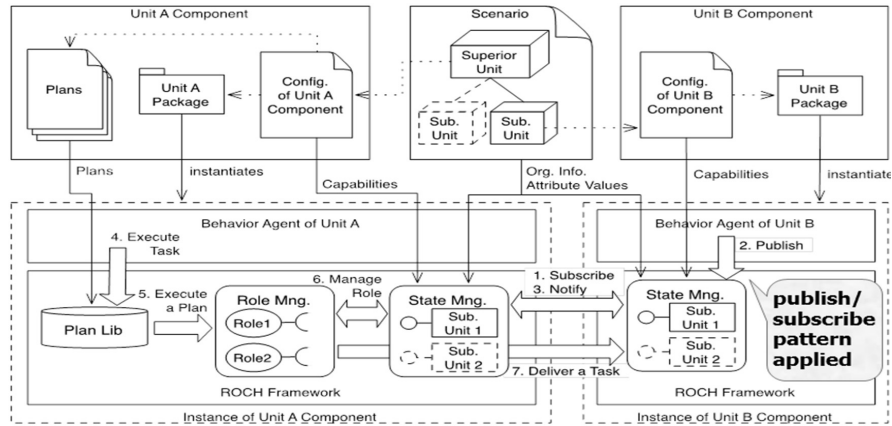


Fig. 6. ROCH Framework

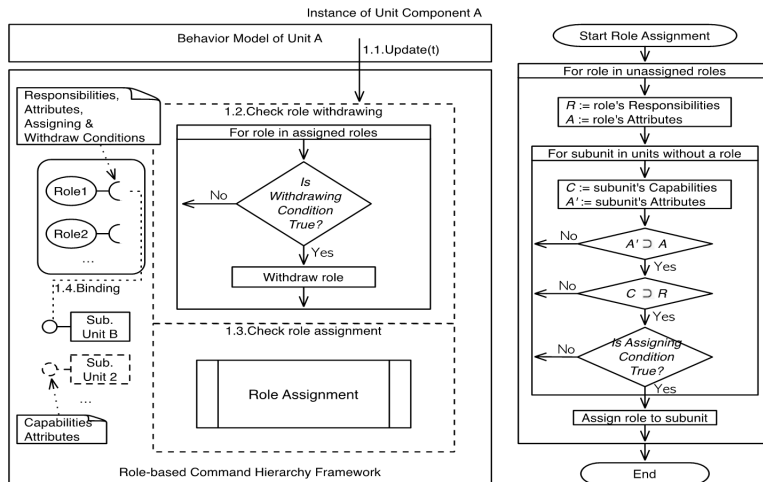


Fig. 7. Role Assignment and Withdrawing

부대는 수신자가 어떤 모의개체인지에 대한 정보가 없이 간접적으로 다른 모의개체에게 명령을 전달할 수 있다. 상급부대는 단지 관련된 역할을 알면 된다. 계획라이브러리(Plan Lib)는 모의개체의 계획을 로딩한다. 이것은 모의 사용자 또는 상급부대로부터 받은 과업을 위한 계획을 선택하고 활성화한다. 역할관리자(Role Mng.)는 할당/철회 조건에 따라 역할을 할당/철회한다. 상태관리자(State Mng.)는 그것이 포함된 모의개체와 하위 모의개체들의 상태를 관리한다. 프레임워크는 편제 정보를 포함하고 있는 시나리오를 기반으로 동작한다. 시나리오는 또한 모의개체를 구현한 모델(Config. of Unit Component)에 대한 참조를 포함한다.

계획실행기능(Plan Execution Function)은 계획을 실행하는 것이다. 부대가 어떤 과업을 수신하면(4), 그것의 계획라이브러리는 그 과업을 위한 계획을 선택하고 활성화한다(5). 역할관리자는 활성화된 계획에 명시된 역할들을 적절한 하위 모의개체에 할당한다(6). 각 역할은 하위 모의개체와 결합되고, 계획에 있는 하위과업들은 각 역할을 수행하는 하위 모의개체에 전달된다(7). 역할을 수행하는 하위 모의개체가 그 역할을 수행할 수 없으면(즉, 그 개체의 상태가 역할의 철회조건을 만족하면), 이 역할은 역할관리자에 의해 다른 적절한 하위 모의개체에게 재할당된다.

Fig. 7은 역할의 철회와 할당의 과정을 순서도를 사용

하여 표현한 것이다. Fig. 7에서, 부대의 행위 에이전트는 주기적으로 프레임워크의 업데이트 함수를 호출한다(1.1). 프레임워크의 역할관리자는 할당/철회조건을 기반으로 하위 모의개체로부터 철회하거나(1.2), 할당할 역할이 있는지(1.3)를 검사한다. 이 단계들은 상태관리자가 제공하는 모의개체와 역할의 정보를 사용한다. 마지막 단계는 역할의 할당 조건을 만족하는 하위 모의 개체에 이 역할을 할당하는 것이다. Fig. 7에 있는 우측 순서도는 역할을 할당하는 과정을 설명한다. 이 과정은 먼저 할당되지 않은 각각의 역할에 대해, 그 역할의 수행을 위해 요구되는 행위의 집합(R)과 상태를 표현하는 속성의 집합(A)을 하위부대(subunit)가 수행할 수 있는 행위의 집합(R)과 속성의 집합(A')이 각각 포함하는지를 검사한다. 그리고 마지막으로 현재 속성 값으로 표현되는 상태가 역할의 할당 조건을 만족하는지를 검사하여, 이를 만족하면, 역할을 할당한다. 역할을 할당한 후에, 프레임워크는 역할과 하위

모의개체 쌍의 정보를 생성하여 저장한다(1.4).

4. ROCH 모델의 구현

본 논문에서 위게임 모의에서 ROCH 모델의 구현이 임무 하달 상황을 실현하는 방법을 설명하기 위해, Fig. 8 과 9에서 설명되는 교전 시나리오를 활용하였다.

Fig. 8은 시나리오로서, 이 시나리오는 청군 전차소대가 홍군 전차소대와 교전하게 되는 초기 상황과 모의 중 수행해야 하는 계획을 포함하며, 각 전차 소대는 3대의 전차로 구성된다. 이를 위해서 청군과 홍군 각각은 하나의 소대 모델과 세 개의 전차 모델을 사용한다. 청군 전차소대의 계획은 X 지점 (Point X)을 확보하는 것이다. 홍군 전차소대의 계획은 청군에 대하여 현재 위치를 방어하는 것이다.

Fig. 9는 청군 전차소대의 지점확보 과업에서 사용되는

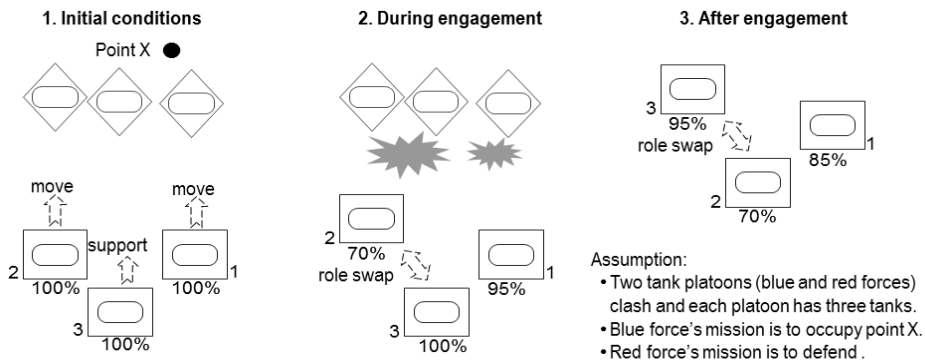


Fig. 8. Simple Scenario for Test and Expected Result

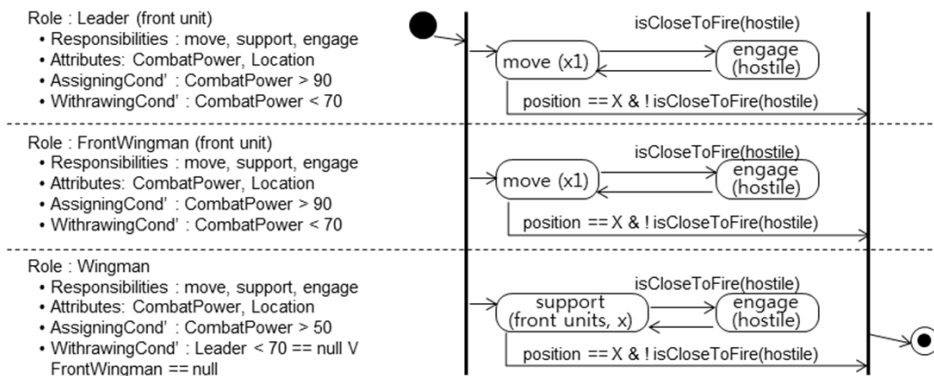


Fig. 9. Role Definition in the Scenario

계획을 보여준다. 이 계획에서는 지휘전차(Leader)과 선두전차(Front Wingman), 지원전차(Wingman)의 세 역할을 시01용한다. 적을 발견하지 않은 이동(move) 과업을 수행할 때, 지휘전차는 대형의 전열에서 소대를 이끌고, 선두전차는 지휘전차와 같은 열을 유지한다. 이 때, 지원전차는 후방에서 전열의 차량들을 지원한다. 역할의 정의에서 지휘전차와 선두전차의 할당 조건은 90%를 초과하는 전투력을 유지해야 하는 반면, 지원전차의 할당 조건은 50%를 초과하는 전투력을 유지해야 한다. 모의 중에 우리는 교전이 발생한 후, 지원전차가 지휘전차나 선두전차 중 피해가 발생한 차량의 역할을 인수받기를 기대한다.

이 시나리오는 단순하지만 ROCH가 가지는 이점의 주요 맥락을 보여줄 수 있을 만큼 충분하다. Fig. 10은 실제 모의 소프트웨어에서 Fig. 5의 ROCH 메타 모델 기반으로 개발된 행위 편집기의 일부 화면을 보여준다. Fig. 10에서 녹색의 상자는 역할을 의미하며, 역할을 할당할 때 수행해야 하는 행위의 절차를 포함한다. Fig. 9에서 설명된 역할들은 이 행위 저작기를 사용하여 다음과 같이 모델링된다. 지휘전차(Leader)는 이동(Move)를 수행한 후, 적이 사거리 내에 존재하는 적을 발견하면 교전을 수행하고, 적이 없는 경우 다시 이동을 반복 수행한다. 지원전차(Wingman)는 지원을 위해 지휘전차를 쫓으며(Follow), 지휘전차와 마찬가지로 사거리 내에 적을 발견하면 교전을 수행하고, 그렇지 않으면 지휘전차를 쫓아 이동을 수행한다. Fig. 10의 좌측 상단 패널은 각 역할과 그것의 할당조건을 보여준다. 이것은 지휘전차의 전투력(CombatPower)

이 High(90%를 초과)가 아니고 지원전차의 전투력이 High보다 높으면, 이 두 역할을 수행하는 모의개체가 역할을 교환한다는 내용을 암묵적으로 담고 있다. 이 경우에 지휘전차는 실제 전장에서 지휘를 담당하는 차량이라기보다 대형의 전열에서 다른 전차들을 이끈다는 것을 의미한다. 앞서 언급한 바와 같이, 이 구현에서 우리는 역할 할당/철회 조건으로 각각을 구현하지 않고 하나의 할당 조건만을 사용했다.

Fig. 11은 Fig. 8에서 설명된 시나리오와 Fig. 9에 정의된 역할들을 사용하여 모의할 때, 평균 전차소대의 3호 전차가 남긴 모의 로그의 일부를 보여준다. 이 로그 테이블은 3호 전차가 수행한 행위의 기록으로 각 행은 새로운 행위가 시작되거나 기존의 행위가 종료될 때, 그것의 상태와 시간을 보여주고 있다. 이 테이블에서 모의 시간은 하단에서 상단 방향으로 진행된다. 각 행은 가장 앞에 등장하는 동근 사각형으로 표현되는 전술적이동 행위(Tactical-MovementU)를 가지고 시작한다. 이 행위는 시나리오에서 주어진 X지점을 점령하기 위해 이동하는 과업으로, 소대에 포함된 전차들이 Fig. 9와 10을 통해 설명된 지휘전차(Leader)와 선두전차(FrontWingman), 지원전차(Wingman)의 역할을 수행함으로써 목표를 달성하게 된다. Fig. 11의 아래서 4번째와 5번째 행이 보여주는 바와 같이 ROCH의 프레임워크에서 역할이 변경되면, 그 내용이 로그로 출력된다. 이 실험에서 3호 전차의 역할은 지원전차(Wingman)에서 지휘전차(Leader)로, 지휘전차(Leader)에서 지원전차(Wingman)로 두 번에 걸쳐서 변경되었다. 이것은 3호

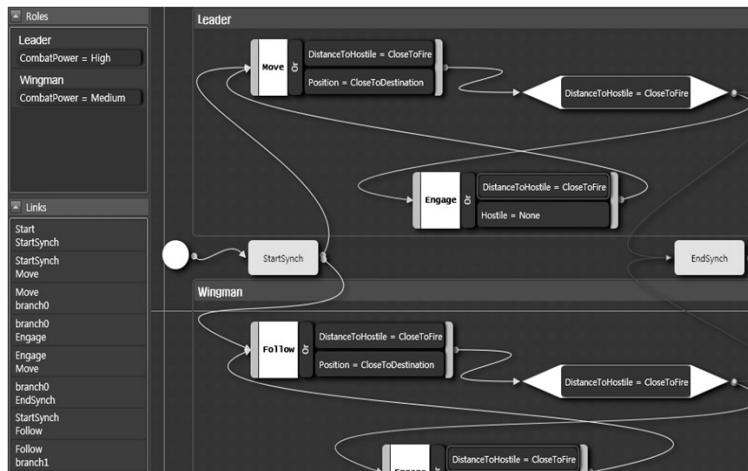


Fig. 10. Screen Capture for Behavior Editor

Drag a column header and drop it here to group by that column

Time	Behavior execution info
00:02:08.8163679	TacticalMovementU Wingman TacticalMovementW MovementU
00:02:08.8153678	TacticalMovementU Wingman TacticalMovementW
00:02:08.813677	TacticalMovementU Wingman
00:02:07.8133105	TacticalMovementU Leader
00:01:01.8075352	TacticalMovementU
00:00:59.8074208	Planned Behaviors
00:00:58.8083636	Planned Behaviors FireSupportRequestU
00:00:58.8073636	Planned Behaviors
00:00:57.8063063	Replanning <TacticalMovementU>
00:00:56.8072492	Replanning <TacticalMovementU>
00:00:55.8071920	TacticalMovementU
00:00:08.8035035	TacticalMovementU Leader TacticalMovementL Movement
00:00:08.8015034	TacticalMovementU Leader TacticalMovementL
00:00:08.8005034	TacticalMovementU Leader
00:00:07.8034463	TacticalMovementU Wingman
00:00:07.8014462	TacticalMovementU
00:00:05.8003318	Coping
00:00:04.8112752	Coping

Fig. 11. Part of Simulation Log for Tank #3

전차가 최초 지원전차의 역할로 이동하는 중에 적 전차와의 교전이 발생하여 지휘전차 역할을 수행하던 전차가 피해를 입어 이 둘 간의 역할 교환이 발생하였고, 계속된 교전에서 지휘전차 역할을 수행하던 3호차량이 피해를 입고 지원전차로 역할을 변경하는 상황이 발생했음을 보여준다. Fig. 11에서 아래로부터 8번째부터 13번째 행에 표시된 다른 색의 둥근 사각형은 (Kim & Choi, 2013)의 연구에서 수행된 재계획을 수행하는 과정에서 포병 지원사격을 수행하는 행위로, 이에 대한 설명은 본 논문의 요지를 벗어나기 때문에 생략한다.

ROCH의 구현과 실험을 통해 우리는 모델의 조립성과 재사용성이 향상된다는 우리의 주장에 대한 확신을 가질 수 있었다. 그 근거로 실험에서 역할은 전차소대와 전차를 결합하는 매개체로 사용되었으며, 이 때 기존의 모델이 유지했던 전차소대와 전차간에 직접적인 의존성이 역할을 통해 전차소대와 역할, 역할과 전차간의 역할과 속성에 대한 간접적인 의존성으로 재정리되었다. 이 과정에서 사전에 정의된 역할을 가지는 전차 모델이 내부의 수정 없이 역할을 정의 또는 수정하는 작은 노력을 통해 전차소대에서 재사용되었다.

추가로, ROCH는 Fig. 12가 보여주는 바와 같이 모의 모델의 해상도 측면에서 확장성을 제공한다. 이것은 동일한 전술 표현 방법을 사용하여 병사나 전차와 같은 개체 수준에서 연대와 같이 대규모 부대의 과업과 이를 수행하기 위한 계획을 표현하도록 지원한다. 그리고 모의 과정에서 부대(예, 연대)가 과업을 수행할 때, 이 부대는 역할

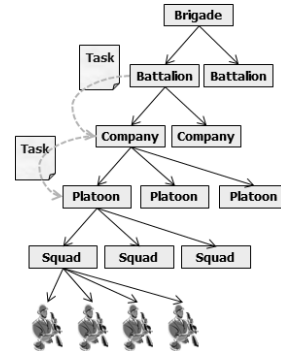


Fig. 12. Scalability in ROCH

을 고려하여 그것의 하위 수준의 과업을 하위 모의개체(예, 대대)에게 전달한다. 이 하위 모의개체가 부대인 경우, 그것은 다시 그것의 하위 모의개체(예, 중대)에게 상급부대로부터 할당된 과업을 수행하기 위한 하위 과업을 전달한다. 이와 같은 동작방법은 시나리오에 명시된 조직의 전반에 걸쳐서 적용될 수 있다.

5. 결론 및 향후 연구

본 논문에서 우리는 OneSAF와 FLAMES, VR-Forces를 포함하여 현재의 두드러진 위게임 소프트웨어에 대해 조사하고 이를 토대로 공통된 모델을 정리하였다. 이 과정을 통해 모의개체와 전술이 밀접하게 결합되어 있는 기존의 모델들을 설명하였다. 이와 같은 결합은 사전에 개발된 모의개체의 모델들과 전술의 재사용을 어렵게 만든다. 특히, 모의 중에 동적으로 변화하는 상황에서 실제 전장 상황에서 벌어지는 것과 같이 임의의 모의개체가 다른 모의개체의 임무를 맡아서 수행하는 것은 거의 불가능하다.

상기 단점을 극복하기 위해 우리는 메타모델로 기술되는 전술 모델을 실행하는 ROCH 모델을 소개했다. 이 모델은 전술과 그것을 수행하는 모의개체를 분리한다. 그 결과, 모의개체와 전술의 컴포넌트들은 느슨한 결합을 유지할 수 있어, 조립성과 재사용성을 강화하는 효과를 가져온다. 또한, ROCH 프레임워크의 역할 할당 매커니즘은 모의중에 역할을 동적으로 할당할 수 있게 한다. 이것은 사용자가 빠르게 변화하는 전장 상황을 반영하여 동적으로 적용되는 전술을 모의할 수 있도록 돕는다. 마지막으로 ROCH는 분대부터 연대, 그 이상까지의 다양한 수준의 부대에 동일한 메커니즘으로 적용할 수 있는 확장성을 갖는다.

본 연구의 주요 활동으로 수행한 ROCH를 구현/실험하는 과정에서 우리는 몇몇 예외상황들을 식별했다. 이것은 주로 모의개체가 전투능력을 완전히 상실한 경우로, 이 상황에서 상급부대는 자신이 필요로 하는 역할의 수행자를 찾을 수 없다. 이와 같은 상황을 해결하는 방법으로, 현재의 ROCH는 모든 예외상황을 다루기 위한 보다 정교한 계획을 필요로 한다. 하지만 이해결 방법은 사용자의 불편을 불러올 수 밖에 없다. 이와 같은 문제를 극복하는 방법으로, 우리는 ROCH를 기반으로 부대의 상황에 따라 계획을 수정/보완할 수 있는 자가적응형(Self-Adaptive) 계획 생성과 관련된 연구를 조사하고 있다.

References

1. Logsdon, J., Nash, D., and Barnes, M., "One semi-automated forces (OneSAF): capabilities, architecture, and processes", DoD M&S (Modeling and Simulation) Conference Presentations, Orlando, Florida, DoD M&S Conference, 2008.
2. Temion Corporation. "FLAMES Simulation Framework: Online Document Version 10.0.1", 2012. from <http://www.temion.com>.
3. MÄK., "VR-Forces: Developers Guide", 2012. from <http://www.mak.com/products/simulate/computer-generated-forces.html>
4. Atkin, M. S., Westbrook, D. L., and Cohen, P. R., "HAC: A unified view of reactive deliberation activity", Proceedings of the 5th International Conference on Autonomous Agents, pp. 92-107, 2001.
5. Vakas, D., Prince, J., Blacksten, H. R., and Burdick, C., "Commander behavior and course of action selection in JWARS", Proceedings of the 2001 Winter Simulation Conference, pp. 697-705, 2001.
6. Song, Y., and Yang, Y., "Modeling organization of multi-agent system with command mechanism", Proceedings of the 1st International Multi-Symposiums on Computer and Computational Sciences, pp. 732- 736, 2006.
7. Bisht, S., Malhotra, A., and Taneja, S. B., "Modeling and simulation of tactical team behavior", Defence Science Journal, Vol. 57, No. 6, pp. 853-864, 2007.
8. Ferber, J., & Gutknecht, O., & Michel, F., "From agents to organizations: an organizational view of multi-agent systems", Agent-Oriented Software Engineering IV, Lecture Notes in Computer Science, Vol. 2935, pp. 214-230, 2004.
9. Xu, H., Zhang, X., and Patel, R. J., "Developing role-based open multi-agent software systems", International Journal of Computational Intelligence Theory and Practice, Vol. 2, No. 1, pp. 39-56, 2007.
10. Cabri, G., Leonardi, L., and Zambonelli, F., "BRAIN: A framework for flexible role-based interactions in multiagent systems", On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, Lecture Notes in Computer Science, Vol. 2888, pp. 145-161, 2003.
11. Becht, M., Gurzki, T., Klarmann, J., and Muscholl, M., "ROPE: role oriented programming environment for multiagent systems", Proceedings of the 4th IFCIS International Conference on Cooperative Information Systems, pp. 325-333, 1999.
12. Hahn, C., Madrigal-Mora, C., and Fischer, K., "A platform-independent metamodel for multiagent systems", Autonomous Agents and Multi-Agent Systems, Vol. 18, Issue 2, 239-266, 2009.
13. Object Management Group, "OMG Unified Modeling Language (OMG UML), Infrastructure, Version 2.4.1", 2012. accessed on 30-11-2012.
14. Petty, M., Kim, J., and Byun, J., "Software Frameworks for Model Composition, Modeling & Simulation in Engineering", vol. 2014. article ID 492737.
15. Kim, H., Lee, S., "Role-based Command Hierarchy Model for War Fare Simulation", International Journal of Simulation Model Vol.12 No.4, pp. 252-263, 2013.
16. Kim, J., & Choi, D. (2013). "Implementation of Goal Oriented Behavior Planning, Re-planning for SAF", Interservice/ Industry Training, Simulation, and Education Conference, Orlando, National Training and Simulation Association, 2013.



김 정 윤 (jkim@realtimevisual.com)

1997 Texas Tech Univ. 전산학 석사
2006 한국과학기술원 전산학 박사

관심분야 : 군수정보체계, 소프트웨어 개발 프로세스, 시뮬레이션



김 희 수 (hskim@realtimevisual.com)

2005 아주대학교 컴퓨터공학과 석사
2010 리얼타임비주얼 개발팀장

관심분야 : 시뮬레이션, 인공지능, 소프트웨어 개발



이 상 진 (sangjinlee@add.re.kr)

2008 한국과학기술원 산업공학 박사
2011 국방과학연구소 선임연구원

관심분야 : 모델링, 시뮬레이션, 조립형 모의 아키텍처, 및 가상군