

# Cougaar Architecture 활용 다중 에이전트 기반 소규모 교전 시물레이션 Framework

함원경<sup>1</sup> · 정용호<sup>1</sup> · 박상철<sup>2†</sup>

## A Simulation Framework of Multi-Agent Based Small Engagement Using Cougaar Architecture

Won. K. Hwam · Yongho Chung · Sang. C. Park

### ABSTRACT

M&S in the field of national defense is a battle system has been highly spotlighted for obtaining weapon systems, analyzing and experimentation of battle effects to reduce costs, time, and risks. It is classified as Campaign, Mission, Engagement, and Engineering levels by detail of description. In engagements, many situations on the battle field which are really unpredictable are required to be considered on the view of diverse tactics. Thus, engagement simulation is in demand to use for forecasting real-world battle situations by inserting various components which consists of real engaging situations into virtual local battle field. While developing the engagement simulation, adopting the concept of agent-based simulation gives it benefits which are improved autonomy, composability, and reusability of entities. It means reducing the time, cost and effort to develop the simulations. This paper concentrates on the framework of multi-agent based engagement simulation using Cougaar Architecture.

**Key words** : Multi Agent-Based simulation, Engagement, National defense, Cougaar

### 요약

국방 분야에서의 M&S는 무기 체계 획득 및 개발에 투입되는 비용, 시간, 노력 및 위험을 감소하기 위한 전투 효과 실험 등에 사용되며 그 중요성이 높아지고 있다. 국방 M&S에서 시물레이션 모델은 묘사수준에 따라 전구급, 임무급, 교전급, 공학급 모델로 분류할 수 있는데, 그 중 교전급 모델은 다양한 자원들을 가상의 전장환경에 투입하여 시물레이션을 수행하며, 이것을 통하여 사용자들이 예측 불가능한 국지 전장에서의 많은 상황들에 대하여 다각적인 시야로 전술을 고려할 수 있게 하는 역할을 한다. 이런 교전급 시물레이션을 개발할 때, 에이전트 기반 시물레이션 개념을 도입하면 개체의 향상된 자율성, 조합성 및 재사용성을 가질 수 있다. 그것은 시물레이션 모델의 개발에 소요되는 시간, 비용 및 노력을 절감할 수 있음을 의미한다. 본 논문은 쿠키아 아키텍처를 활용하여 다중 에이전트 기반의 교전급 시물레이션 프레임워크를 설계하는데 그 목적을 둔다.

**주요어** : 다중 에이전트 기반 시물레이션, 교전, 국방, Cougaar

## 1. 서론

IT의 발달과 미래 전장환경 및 전쟁 패러다임의 변화, 새롭고 다양한 작전적 요구에 비해 상대적인 국방 가용자

접수일(2011년 6월 27일), 심사일(1차 : 2011년 9월 9일, 2차 : 2011년 9월 28일), 게재 확정일(2011년 9월 28일)

<sup>1)</sup> 아주대학교 산업공학과

<sup>2)</sup> 아주대학교 산업정보시스템공학부

주 저 자 : 함원경

교신저자 : 박상철

E-mail; scpark@ajou.ac.kr

원의 제약, 그리고 환경오염 문제등으로 인해 한국군에서 국방 모델링 및 시물레이션(M&S)에 대한 중요성이 점점 높아지고 있다(차진섭, 2007; 최상영, 2010). 국방 M&S는 실세계의 전장, 자연, 지리, 인간에 대한 수학적, 논리적 모델 등으로 구성되며, 전투 환경에 대해 분석하여 작전을 지원하고 평가하는 분석분야, 무기체계를 개발하고, 시험하는 획득분야, 장비 운전자 또는 지휘자 등 전투 구성원들의 훈련 분야에 주로 사용된다. 국방 M&S를 획득, 분석, 훈련 분야에 사용함으로써 계획을 미리 검증하여,

표 1. 묘사 수준에 따른 국방 M&S 분류

	모의전력	주요묘사	산출물	사용목적
전구 (Campaign)	국가	위협, 작전형태, 과업	전쟁결과, 전력손실	전력 평가, 작전 분석
임무/전투 (Mission/Battle)	군단, 사단	위협, 작전형태, 과업	임무효과도, 손실률	부대 구성, 임무 효과도
교전 (Engagement)	연대, 대대 및 소부대	과업, 체계 기능	명중확률, 생존성/취약성	체계효과도, 비용/성능/기술상쇄분석
공학 (Engineering)	무기체계 플랫폼, 구성모델	체계 기능	성능, 탐지거리, 사거리	최적공학설계 비용/생산성 분석

타당하지 못한 계획의 수행에 의해 발생하는 비용, 시간, 노력 및 위험을 감소시키는 것이 가능하다(John 등, 2000).

국방 M&S에서 시뮬레이션 모델은 묘사수준에 따라 전구급, 임무급, 교전급, 공학급 모델로 분류할 수 있는데, 각 모델의 특성은 표 1과 같다. 묘사 수준에 따른 모델 중 교전 시뮬레이션 모델은 과업과 체계 기능이 자세히 묘사되며, 위협이나 작전형태는 부분적으로 모델링 되어진다(차진섭, 2007; 최상영, 2010). 교전급 모델은 다양한 자원들을 가상의 전장환경에 투입하여 시뮬레이션을 수행하며, 이것을 통하여 사용자들이 예측 불가능한 국지 전장에서의 많은 상황들에 대하여 다각적인 시야로 전술을 고려할 수 있게 하는 역할을 한다. 또한 최근 국방 M&S에서 교전급의 모델을 무기체계의 획득에 사용하려는 시도가 계속 되고 있으며, 체계개발 초기단계부터 요구사항과 효과도를 분석하여 요구의 적절성과 무기체계 사양 및 적절성에 대한 검토 하는 것부터 실전 배치 전에 기존의 시스템과 호환 방안을 창출할 목적으로도 사용한다(John 등, 2000; Park 등, 2010). 그러나 기존의 교전급 모델은 복잡한 무기체계를 각각의 구성 시스템 별로 모델링하여 현실에 근접하게 표현하는 기법이 부족하였고, 과거에 모델링한 전투 개체를 다양한 전장 시나리오에 맞춰 신속하게 재사용하는 것과 그것의 확장 및 변경에 한계가 있어 시뮬레이션이 요구되는 무기체계에 대해서 빠르고 신속한 실험 및 분석에 무리가 있었다. 이와 같은 기존의 교전급 전투 실험의 문제를 해결하기 위해 에이전트 기반의 시뮬레이션이 대두되었다(지승도 등, 2008; Ibrahim, Murat, 2009).

본 연구는 언급한 기존 교전급 M&S의 한계점을 해결하기 위해 Cougaar Architecture를 활용한 교전 시뮬레이션 프레임워크를 제안하여, 개발된 모델의 재사용성, 확장성 및 조합성을 갖는 다중 에이전트 기반의 교전 시뮬레이션을 설계한다. 본 논문의 구성은 다음과 같다. 1장에서 연구 배경 및 연구 목적에 대해 설명하고, 2장에서 본 연

구의 접근 방법인 다중 에이전트 기반 시뮬레이션, 3장에서 본 연구의 해결 방법인 쿠가 아키텍처에 대해 설명한다. 그리고 4장에서 쿠가 아키텍처로 다중 에이전트 기반 시뮬레이션의 프레임워크를 예제와 함께 제안하며, 5장에서 연구의 결론 및 토의로 마무리 짓는다.

## 2. 다중 에이전트 기반 시뮬레이션

다중 에이전트 기반 시뮬레이션이란 시뮬레이션을 구성할 때 모델을 에이전트 조합으로 설계한다는 의미의 소프트웨어 엔지니어링 개념이다. 에이전트 설계에 대한 정의는 의견이 분분하나 일반적으로 에이전트는 자율성, 조합성, 사회성 등의 특성을 갖고 있어야 한다. 자율성이란 개체가 사전에 규정된 규칙과 지식에 의거하여 의사결정을 수행하며, 규칙 리스트와 이것을 바탕으로 의사결정을 수행할 추론엔진이 포함되는 의미이다. 이것은 에이전트의 가장 중요한 특성으로서 에이전트가 독립적인 행위의 수행을 가능하도록 한다. 조합성은 에이전트가 쉽게 다른 에이전트 또는 전체 시뮬레이션 프레임에 연동되어 작업을 수행할 수 있어야 한다는 것을 의미하며, 이것은 Plug & Play 개념과 유사하다. 사회성은 유사한 상황에 있는 비슷한 기능의 에이전트들은 행위의 수행에 있어서 그 전체적 결과가 일관성이 있어야 한다. 그리고 같은 상황에 있는 서로 협력하는 에이전트들은 서로 정보를 공유하며 상황에 대응하는 행위를 수행한다는 것을 의미한다(Thomas 등, 2004; Wikipedia, 2011).

에이전트의 특성들을 가진 모델들이 조합되어 하나의 모델을 만들고, 조합된 모델들이 계층적 구성 관계를 갖도록 하는 것을 다중 에이전트 설계라 한다. 그림 1은 다중 에이전트 시뮬레이션의 구조를 표현하고 있는데, 상위 레벨의 에이전트들은 하위 레벨의 에이전트들이 모여 구성된다(Ibrahim, 2009). 또한 에이전트는 모듈과 규칙을 포

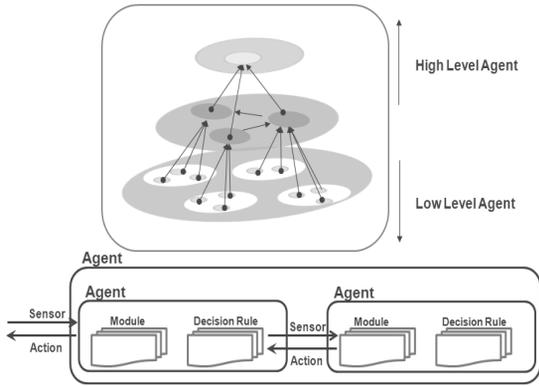


그림 1. 다중 에이전트 시뮬레이션 구조

함하는데, 이런 에이전트들이 모여 다시 하나의 에이전트를 구성하는 것에 대해 표현하였다. 예를 들어 S라는 잠수함을 구성하기 위해 필요한 에이전트는 공격체계, 방어체계, 해군 인적 자원, 추진동력 체계, 생존 체계 등으로 나눌 수 있다. 그리고 각 에이전트는 다른 세부 에이전트로 구성될 수 있다. 공격체계의 경우 미사일, 어뢰, 기뢰 등으로 구성될 수 있으며, 방어체계의 경우 소나, 기만기 등으로 구성될 수 있다. 또한 미사일에도 종류가 있기 때문에 미사일도 세부적으로 에이전트 분류가 가능하다. 필요한 에이전트를 Component 단위의 라이브러리에서 가져와 미사일을 구성하고 미사일 에이전트를 모아 공격 에이전트를 구성하는 방식으로 계층적으로 Component를 구성함으로 보다 쉽게 모델링할 개체를 표현할 수 있으며, 비용 및 시간이 크게 절감될 수 있다(박세연 등, 2010; 지승도 등, 2008). 에이전트 개념을 전투 실험에 가장 처음으로 도입한 시스템은 ISAAC이며(Ilachinski, 2000), 그 이후로 MANA 등과 같은 시스템들이 개발되었다(Babilot, 2005). 또한 이와 같은 시스템들이 보다 발전하여 인공생명(Artificial Life) 개념이 도입된 EINSTEIN 등이 탄생하였다(Ilachinski, 2000). 하지만 이러한 시스템들은 단순히 에이전트를 이용하여 Script 기반의 자율성이 부족한 SAF(Semi-Automated Force) 수준에 머무른다. 현재 인간과 에이전트 모두 개입된, 훈련/분석을 위한 BattleModel과 같은 시스템들도 개발되어 주목받고 있으며(KESEM, 2004), 지휘관을 위한 에이전트 시스템으로 TAO ITS 등이 존재한다. 그러나 인간의 개입을 배제하고, 에이전트의 특성을 충족시키는 시스템이 현재까지도 부족한 실정이다(정찬호 등, 2009).

### 3. Cougaar Architecture

에이전트를 모델링하여 시뮬레이션 하기 위해 쿠가(Cougaar : Cognitive Agent Architecture)를 사용한다. 쿠가는 자바 기반 오픈 소스 에이전트 아키텍처로 인지적 에이전트들을 생성하고 동작을 정의할 수 있으며, 인프라스트럭처 및 핵심 서비스들이 포함되어 있는 소프트웨어 아키텍처이다. 본 아키텍처는 다이내믹한 확장성을 지원하며 복잡한 분산 시뮬레이션을 구축하는데 성공적임이 DARPA(미국방위고등연구계획국: Defence Advanced Research Projects Agency) 산하 Advanced Logistics Program에 의해 증명되었고, 높은 수준의 분산 시뮬레이션과 생존성 있는 에이전트 설계를 지원하며, 쉽게 시뮬레이션 설계 및 조정이 가능하여 확장이 용이하다(Mark, 2004; Todd, 2007). 쿠가는 개략적으로 Component-Based로 만들어진 분산 에이전트 엔진이며, 에이전트들은 쿠가에서 제공하는 비대칭 메시지 전송 프로토콜을 사용하여 통신한다. 이것을 통해 쿠가 에이전트들은 특정 문제를 해결하기 위해 협동할 수도 있고, 그 해결책을 다른 에이전트들과 공유할 수도 있다. 에이전트는 관련된 기능 모듈들의 조합으로 이루어지며, 기능 모듈들은 동적이고 연속적으로 주변 환경, 제약 사항 값 등과 소통하며 문제를 해결한다(Aaron, 2004; BBN; Daniel, 2004).

간단히 쿠가의 기능들에 대해서 설명하면 다음과 같다. 쿠가의 단위 에이전트는 플러그인(Plug-In)들과 블랙보드(Blackboard)로 이루어져 있는데, 플러그인은 Component처럼 에이전트에 조합된다. 플러그인은 상세한 도메인 규칙 또는 프로세스가 구현된 하나의 기능을 담당하는 에이전트의 에이전트라 볼 수 있으며, 이것들이 모여 단위 에이전트의 행위를 구현할 수 있다. 블랙보드는 에이전트 내부의 메모리로 에이전트 내부의 플러그인들은 블랙보드에 정보를 게시/구독(Publish/Subscribe)하는 개념으로 메모리를 공유하며 통신한다. 에이전트 외부로 메시지를 보내는 에이전트간의 통신은 블랙보드 릴레이(Blackboard Relay)라고 불리는 메시지 통신을 통하여 이루어진다. 그림 2는 분산 환경에서 에이전트 내부 및 외부로의 통신에 대한 개념을 표현하고 있는 그림으로, 에이전트가 플러그인과 블랙보드를 갖고 있으며, 에이전트 내부의 플러그인들이 블랙보드를 통하여 내부 데이터를 공유함을 표현한다. 그리고 에이전트 외부로 데이터를 보낼때는 블랙보드에 릴레이 데이터를 게시하여, 다른 에이전트의 블랙보드에 메시지를 보내는 것을 표현하였다. 또한 그림 2는 에

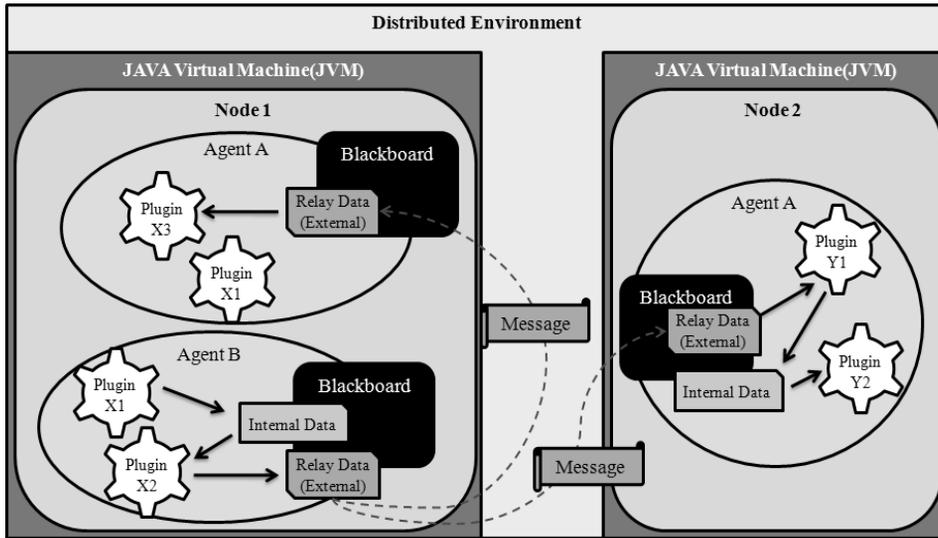


그림 2. 분산 시스템 환경에서 에이전트 구성 및 블랙보드, 릴레이 데이터 전송 개념도

이전트들이 하나의 노드에 포함되어 있고, 노드는 다시 JVM(JAVA Virtual Machine)에 포함되는 것을 설명하고 있는데, 노드는 하나 또는 여러개의 에이전트들을 운영하고, 해당 노드에 속해있는 에이전트들의 작업들을 관리한다. 하나의 노드는 하나의 JVM에 속하며, JVM은 가상의 컴퓨터를 생성하는 것으로 분산 시뮬레이션을 가능하게 해준다(Todd, 2007).

앞에서 설명한 쿠가의 핵심 기능 외에 추가적인 기능으로 메시지 통신 기능의 보조 서비스로 WP(White Page)/YP(Yellow Page)기능을 지원하는데, 이것은 네이밍(Naming) 서비스로 WP는 DNS와 같이 에이전트의 이름을 네트워크 주소와 연결시켜주며, YP는 전화번호부와 같이 속성 값에 근거하여 에이전트를 검색해 준다. 이와 같은 기능을 이용해서 커뮤니티라는 에이전트들의 집합을 형성할 수 있는데, 커뮤니티의 개념을 이용하여 에이전트들의 계층 구조를 설계할 수 있다. 이외에도 지원하는 여러 기능들을 사용하면 다중 에이전트 기반 시뮬레이션을 쉽게 구성할 수 있다. 또한 UI의 역할로 Servlet Service 기능의 엔진을 제공하며, XML 기반 시뮬레이션의 조정 기능을 갖는다(BBN).

쿠가 사용의 구체적인 이점으로는 다음과 같다. 첫째, 쿠가는 하나의 플러그인을 설계하여 구현하면 이 플러그인을 여러 에이전트에 등록하여 사용할 수 있어 그 확장성(Scalability & Expandability)이 매우 크고, 이런 플러그인들이 물리적으로 거리가 있어도 시뮬레이션이 가능하도

록 하는 분산 시뮬레이션 시스템을 지원한다. 둘째, 각 플러그인은 모듈화되어 있으므로, 병렬적 플러그인 소프트웨어의 개발이 가능하다. 셋째, 플러그인의 조합에 따라 다른 역할을 수행하는 에이전트의 개발이 가능해져 조합성(Composability)이 매우 높으며, 시스템 변경(Configure)이 쉽다. 마지막으로, 하나의 플러그인이 오작동을 일으켜 심각한 오류를 일으켰을 경우에도 전체 시스템에 영향을 미치지 않으며, 오류를 일으킨 플러그인만 종료되거나 재시작된다. 이와 같은 이점들을 갖고 있는 쿠가 아키텍처와 기존의 에이전트 지원 시스템(WebLogic, J2EE, Spring, JADE 등)들과의 차이점은 쿠가가 다른 시스템을 하위에 포함하여 시뮬레이션을 진행할 수 있는 통합적인 솔루션을 제공한다는 것이다(BBN; Mark, 2004; Todd, 2007).

지금까지 언급된 쿠가의 특징 및 이점을 종합하여 정리하면, 쿠가는 특정 분야에 맞는 에이전트와 환경에 적합하게 설계하여 자율적 행위를 수행하는 전투개체의 구현 및 정의된 에이전트 간 통신 규칙을 이용하여 서로간의 정보를 주고받으며 에이전트 간 자율적 전투 행위 수행에 의한 전투 실험을 수행하는데 용이한 아키텍처이다.

#### 4. 프레임워크

쿠가 아키텍처를 활용한 다중 에이전트 기반 교전 시뮬레이션이란 쿠가 아키텍처에서 제공하는 에이전트 설계 기능을 이용하여 다중 에이전트 기반의 교전급 시뮬레이

선 시스템을 구축하는 것이다. 일반적으로 교전급 모델의 구성은 세 종류의 개체로 나눌 수 있는데, 각각의 개체는 시뮬레이션 내에서 전투를 수행하는 물리개체, 물리개체를 관리하거나 명령을 내리는 역할을 수행하는 논리개체, 그리고 시뮬레이션이 진행되는 전장환경에 관한 정보를 관리하는 환경개체로 구성된다(Park 등, 2010). 본 절에서는 쿼가를 활용하여 다중 에이전트 기반의 개체를 생성하여 교전 시뮬레이션 및 구성 개체를 설계하기 위한 프레임워크를 제안하며, 프레임워크를 설명하기 위해서 시뮬레이션 할 시나리오를 가정하고, 쿼가를 활용하여 시나리오를 구현한다. 표 2는 쿼가를 적용하는 교전급 시뮬레이션 개발 절차를 설명하고 있으며, 본 절은 표와 같은 개발절차에 따라서 설명이 진행된다.

#### 4.1 시나리오 생성

시뮬레이션을 구축하기 위한 예제 시나리오를 정의한다. 예제 시나리오는 잠수함 A와 B가 서로를 탐색하며 이동하며, 적이 일정 거리 안에 존재함을 인지하면 적을 사정거리 안으로 두고 사격하기 위해 이동한다. 적이 사정거리 안에 존재하면 사격을 하고 피격되었는지 확인한다.

#### 4.2 시나리오 분석

정의된 시나리오를 분석하여 참여개체를 에이전트로, 개체의 기능을 플러그인으로 분해한다. 예제 시나리오에서 참여개체 중 물리개체는 잠수함 A와 B가 있으며, 논리개체는 Location Referee가 존재한다. 각 개체의 기능은 잠수함 A와 B의 경우 탐색, 인지, 공격, 이동, 손상을 가지

며, Location Referee는 물리 개체 위치들을 관리한다. 각 기능별 담당 플러그인의 구성은 표 3과 같이 설계하였다.

#### 4.3 플러그인 설계

플러그인의 구성은 크게 행위(Method) 정의부, 블랙보드(Blackboard) 데이터 게시/구독 정의부, 릴레이(Relay) 데이터 전송/수신 정의부로 구분할 수 있다. 행위 정의부

표 2. 쿼가 활용 교전급 모델 개발 절차

Procedure
1. 시나리오 생성
2. 시나리오 분석
- 참여개체: 에이전트
- 개체기능: 플러그인
3. 플러그인 설계
- 에이전트 내부 플러그인 사이의 정보공유 설계
- 에이전트 사이의 릴레이를 통한 정보 전송 설계
4. 플러그인 구현
5. 에이전트 구현
6. 시뮬레이션 실행

표 3. 기능 담당 플러그인 구성

기능	플러그인
탐색	Radar
인지	Observer
공격	Shooter
이동	Move
손상	Target
개체 위치 관리	Position Manager

표 4. 플러그인 구현을 위한 상세 설계

Plug-in Name	Main Method	Blackboard		Relay	
		Publish	Subscribe	Send	Receive
Radar	Receive Position Value	Targets` Position Values	-	-	Targets` Position Values
Observer	Calculate Distances	Fire or Not	Targets & This Position Values	-	-
Shooter	Fire	-	Fire or Not	Fire	Hit
Move	Move	This Position Value	-	This Position Value	-
Target	Reply	Hit	-	Hit	Fire
Position Manager	Receive & Send Position Values	-	-	All Position Values	All Position Values

는 시뮬레이션 내에서 플러그인이 수행하는 행위를 정의하며, 블랙보드 데이터 게시/구독 정의부에서는 에이전트 내부에서 플러그인 사이의 데이터 공유를 위한 정의를 명시한다. 릴레이 데이터 전송/수신 정의부는 외부 에이전트에 전송할 데이터 또는 외부 에이전트로부터 수신할 데이터에 대해 정의한다. 표 4는 예제 모델의 상세 설계로서, 주요 행위와 블랙보드/릴레이 데이터 관리에 대해 명시하고 있다. 표 4의 플러그인 중 Radar의 주요 행위는 위치값을 받는 것이다. 블랙보드에 게시할 데이터로서 목표들의 위치값을 가지며, 외부 에이전트로부터 수신할 데이터로 목표들의 위치값을 갖는다. 다른 플러그인들도 Radar과 같이 설명될 수 있다.

#### 4.4 플러그인 구현

플러그인의 역할 및 블랙보드/릴레이를 통한 정보 획득 및 제공의 설계가 완료되었으면 제공되는 JAVA기반의 쿼가 라이브러리를 사용하여 플러그인을 구현한다. 그림 3은 플러그인의 기본구조를 표현하고 있는 그림으로, 에이전트의 생성 시점에서 입력치(Arguments)들을 받는다. 이런 입력값들은 에이전트의 특성을 설정하는데 용이한 기능으로 같은 플러그인을 다르게 사용할 수 있다. 에이전트가 생성되면 시뮬레이션이 진행되는 동안의 행위부분으로 상태가 전이된다. 구독설정(Setup Subscription)은 해당 플러그인이 블랙보드 및 릴레이 데이터에서 플러그인 내부로 정보를 전달받을 수 있도록 정의하는 부분이다. 단일서술(Unary Predicates)은 구독설정에서 어떤 정보를 전달받을 것인가에 대해 상세하게 명시하는 기능으로, 구독설정에서는 단일서술들을 참조한다. 그림 4는 플러그인이 구독 설정에서 관리하는 정보들에 대해서 표현하고 있는 그림으로, 단일 서술에 의해 정의되는 정보들의 종류 및 그 교환을 기술하고 있다. 실행(Execute)에서는 플러그인의 정의된 기능들(Methods)을 시뮬레이션 내에서 사용한다. 정의된 기능들은 릴레이 데이터 전송 또는 블랙보드 데이터 게시 기능이 될 수도 있으며, 매 회의 루프마다 값을 변동시키다 특정 조건이 만족되면 알려주는 기능 등 사용자의 정의에 따라 기능이 다양하게 결정된다.

#### 4.5 에이전트 구현

행위를 모델링하는 플러그인들의 집합과 입력 특성치(Arguments), 그리고 에이전트 내부 메모리인 블랙보드로 에이전트가 구성되며, 플러그인들은 블랙보드를 공유하며 정보를 교환하게 된다. 에이전트의 블랙보드 사이에서 내부 및 외부로의 정보교환이 진행되며, 플러그인이

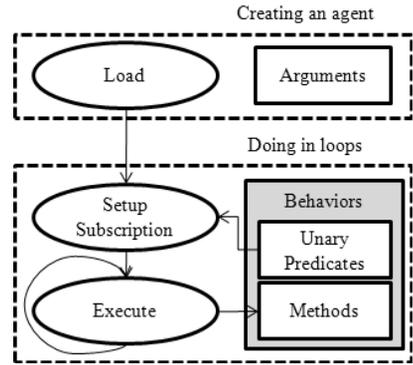


그림 3. 플러그인의 기본 구조

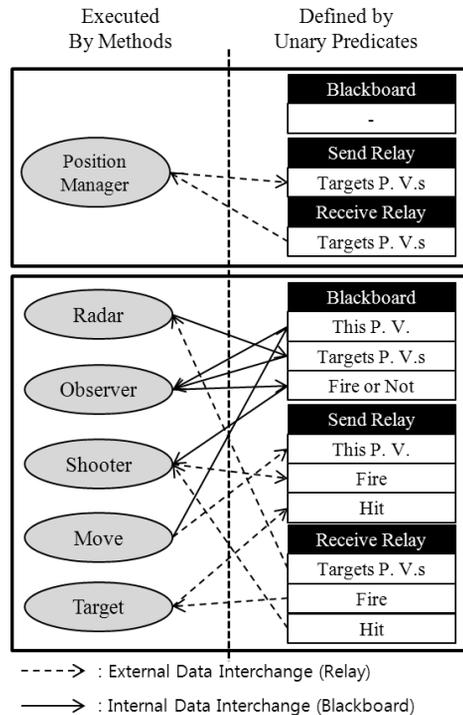


그림 4. 플러그인 데이터 처리구조

블랙보드에 정보를 게시하면 그 정보를 필요로 하는 플러그인들은 블랙보드의 정보를 구독하여 행위에 관한 결정을 내린다. 그림 5는 플러그인 라이브러리에서 플러그인을 선택 및 조합하여 에이전트를 생성하고 시뮬레이션 환경인 사회(Society) 설계에 대해 표현하고 있다. 에이전트의 설계는 그림 6과 같이 XML을 이용하여 플러그인들을 조합함으로써 이루어진다. 에이전트들은 하나의 플러그인을 공유하는 것이 가능하며, 공유된 플러그인은 속해있는 에이전트에 따라서 다른 행위를 수행한다.

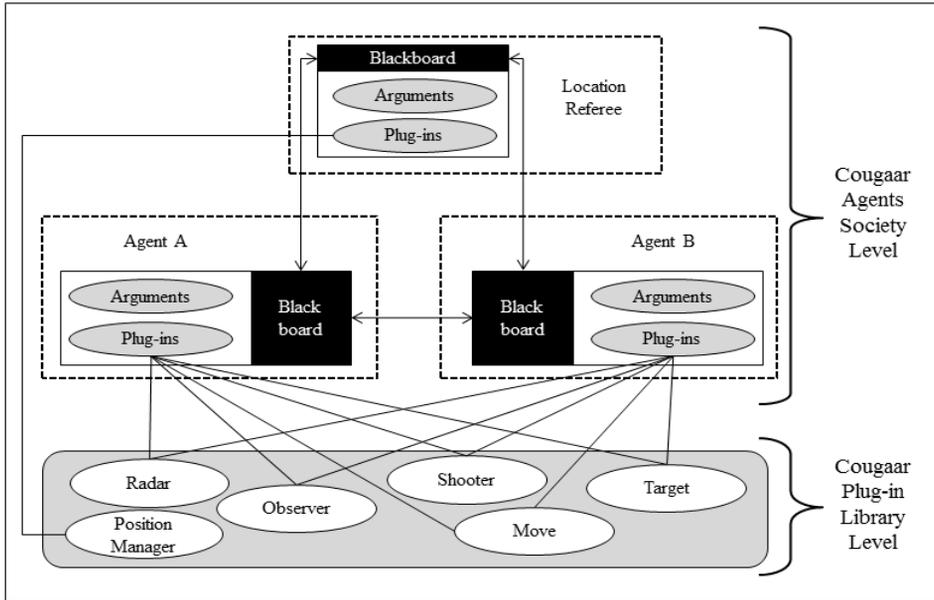


그림 5. 전체 Society의 설계

```

<society>
  <node name = "Node1">
    <agent name = "A">
      <component class = "Engmt.T1.T1Shooter">
        <argument name = "target" value = "B"/>
      </component>
      <component class = "Engmt.T1.T1Target">
        <argument name = "path" value = "/responseA"/>
      </component>
      <component class = "Engmt.T1.T1MOE"/>
      <component class = "Engmt.T1.T1Move">
        <argument name = "target" value = "B"/>
        <argument name = "XInitPos" value = "120"/>
        <argument name = "YInitPos" value = "15"/>
        <argument name = "ZInitPos" value = "0"/>
      </component>
      <component class = "Engmt.T1.T1Radar">
        <argument name = "target" value = "LocationInfoReferee"/>
      </component>
      <component class = "Engmt.T1.T1Observer">
        <argument name = "EngagingDistance" value = "200"/>
      </component>
    </agent>
  </node>
</society>
    
```

그림 6. XML기반 환경 설정

```

SHOUT [TIRadar] - B: VecVal: -84,5 , 5,0 , 0,0
SHOUT [DOTS] - +--+
SHOUT [T1Shooter] - A: Enemy is just detected.
SHOUT [T1Shooter] - A: Will shoot a gun the 1th bullet to the target
SHOUT [TIRadar] - A: VecVal: 114,5 , 15,0 , 0,0
SHOUT [DOTS] - +--+
SHOUT [T1Shooter] - A: Firing a torpedo to the target B
SHOUT [DOTS] - ++
SHOUT [T1Target] - B: This attack is disturbed by decoy.
SHOUT [DOTS] - -
SHOUT [TIRadar] - B: VecVal: -84,0 , 5,0 , 0,0
SHOUT [DOTS] - +--+
SHOUT [TIRadar] - A: VecVal: 114,0 , 15,0 , 0,0
SHOUT [DOTS] - +--+
SHOUT [DOTS] - +--+
SHOUT [TIRadar] - A: VecVal: 109,5 , 15,0 , 0,0
SHOUT [DOTS] - +--+
SHOUT [TIRadar] - B: VecVal: -79,0 , 5,0 , 0,0
SHOUT [DOTS] - +--+
SHOUT [T1Target] - B: Damaged by 1th shooting from A
SHOUT [DOTS] - ++
SHOUT [T1Shooter] - A: Will shoot a gun the 2th bullet to the target
SHOUT [TIRadar] - A: VecVal: 109,0 , 15,0 , 0,0
    
```

그림 7. 시뮬레이션 실행 로그 출력

#### 4.6 시뮬레이션 실행

구현된 에이전트 사회를 쿠가를 이용하여 시뮬레이션을 실행하였다. 그림 7은 시뮬레이션이 진행됨에 따라 변경되는 위치 값과 적의 인지 및 공격하는 과정을 로그를 출력하여 알려준다.

### 5. 결론 및 토의

본 연구는 교전급 시뮬레이션의 구현에 있어 개체에

이전트의 재사용성, 확장성 및 조합성을 갖도록 설계하기 위해 쿠가 아키텍처를 활용하여 다중 에이전트 기반 교전급 시뮬레이션의 프레임워크를 설계하고 구현하였다. 기존의 연구 중 DARPA에서 물류 시뮬레이션 모델에 쿠가를 적용한 사례가 있지만, 전투 실험에 활용된 사례는 드물다(Cougaar, 2011). 따라서 본 연구는 컴포넌트 기반 분산 에이전트 아키텍처인 쿠가를 이용해 에이전트 기반의 교전급 시뮬레이션에 적용하는 것에 초점을 맞추어 진행하였고, 그 적합성을 확인하였다.

쿠가를 활용하여 교전 시물레이션을 구현하는 것의 가장 큰 이점은 XML을 이용하여 생성된 플러그인을 쉽게 조합해서 에이전트를 생성할 수 있으므로, 높은 확장성 및 변경성을 갖는 것이다. 따라서 구현되어 있는 플러그인을 조합하는 것으로 에이전트 기반의 전투 개체의 생성이 용이하며, 신속한 전투 실험이 가능해진다. 또한 플러그인의 동시 개발이 가능하며, 각 플러그인은 개별적이어서, 시물레이션 시스템 중 하나의 플러그인에 오류가 발생하더라도 전체 시스템이 영향을 받지 않는다. 하지만 에이전트의 가장 특성 중 하나인 자율성의 구현에 있어서는 쿠가 아키텍처에서 지원하는 기능이 없으며, 플러그인의 기능 구현 부분(Methods)에 임의로 정의하여 실행(Execute)되도록 해야 한다.

선행 연구들에서 개발된 에이전트 기반 시물레이션 도구들과 쿠가의 차이점은, 앞선 도구들은 전투 실험을 위해 설계된 시물레이션 도구인 반면, 쿠가는 분산 시물레이션에서 미들웨어의 성격을 갖는다. 즉, 쿠가는 분산된 플러그인들을 조합하여 에이전트를 생성하는 기능을 제공하고, 조합 가능한 플러그인을 설계 및 구현할 수 있도록 지원하므로, 교전급뿐만 아니라 군수 개념 등이 복합적으로 결합된 전구 및 임무급 시물레이션에도 적용이 가능하다. 반면, 에이전트 특성 중 자율성의 구현에 있어서, 자율적인 전투 개체의 생성을 지원하는 기능은 존재하지 않으며, 따라서 설계자의 플러그인 및 에이전트 설계능력에 의존적이다.

비록, 쿠가 아키텍처가 자율성의 구현에 한계를 갖고 있으나, 플러그인의 재사용성 및 에이전트의 확장성을 활용하여 특정 분야에 종속되지 않는 신속하고 다양한 시물레이션을 가능하게 한다. 향후, 본 연구의 예제 모델 구현 중에 사용되지 않은 여러 가지 쿠가의 기능들, 예를 들면 웹서블릿(Web Servlet) User Interface 또는 어셋(Asset)을 통한 같은 종류의 다른 에이전트 제작 등을 적극 활용하여 플러그인 라이브러리를 구축하고, 분산 환경에서 실험을 통해 쿠가를 활용한 다중 분산 에이전트 기반의 교전급 시물레이션 시스템을 기대할 수 있다.

### 감사의 글

본 연구는 방위사업청과 국방과학연구소(UD100009DD, UD080042AD)의 지원으로 수행되었습니다.

### 참고 문헌

1. 박세연, 신하용, 이태식, 최봉완(2010), “에이전트 기반의 NCW 전투모델링 시스템 설계”, 한국시물레이션학회 논문지, 19(4), pp. 271-280.
2. 정찬호, 유용준, 류한열, 이장세, 김재익, 지승도(2009), “계층구조적 다중에이전트를 이용한 다대다 합정전투 M&S 시스템”, 한국시물레이션학회 논문지, 18(4), pp. 117-125.
3. 지승도, 유용준, 정찬호, 이장세, 김재익(2008), “에이전트 기반의 인간 미개입형 합정전투 M&S 시스템 설계 및 서해교전 사례연구”, 한국시물레이션학회 논문지, 17(2), pp. 49-61.
4. 차진섭(2007), “국방 M&S의 활용”, 정보과학회지, 25(11), pp. 102-108.
5. 최상영(2010), *국방 모델링 및 시물레이션 총론*, 북코리아, pp. 10-25.
6. Aaron, H. (2004), “Cougaar: A Scalable Distributed Multiagent Architecture”, 2004 IEEE SMC, Hague.
7. Babilot, M. (2005), “Comparison of a Distributed Operations Force to a Traditional Force in Urban Combat”, The Master's Thesis of Navl Postgraduate School, September, 2005.
8. BBN, “Cougaar Architecture Document”, A BBN Technologies Document.
9. Cougaar (2011), General Documentation of Cougaar Agent Architecture, <http://cougaar.org/twiki/bin/view/Main/GeneralDocumentation>
10. Daniel, C., Antony, R. and Jeffery, B. (2004), “A Cougaar-Based Logistics Modeling Tool for Highly Adaptable Military Organizations”, Open Cougaar 04 paper.
11. Ibrahim CIL, Murat MALA (2009), “MABSIM: A Multi Agent Based Simulation Model of Military Unit Combat”, proc. of IEEE ICADIWT2009, London, ISBN 978-1-4244-4457-1, pp. 731-736.
12. Ilachinsk, A. (2004), “Artificial War: Multiagent-based Simulation of Combat”, World Scientific Publishing Co. Pte. Ltd.
13. John, F. K., Robert, R. L., Stephen, E. M. and James, E. C. (2000), “An Architecture for Simulation Based Acquisition”, Johns hopkins APL Technical Digest, vol. 21, no. 3, pp. 348-358.
14. KESEM, “BattleModel:Supporting Capability Lifecycle Decisions”, Richmond, Australia: KESEM International, 2004.
15. Mark Barger (2004), “Introduction to Cougaar Programming”, DARPA.
16. Park Sang C., Kwon Yongjin, Seong Kilyeong, Pyun Jaijeong (2010), “Simulation framework for small scale

- engagement”, Computer&Industrial Engineering, 59, pp. 463-472.
17. Thomas, M. C., Thomas, W. L. and Susan, M. S. (2004), “Military applications of agent-based simulations”, Proceedings of the 2004 winter simulation conference.
18. Todd Wright (2007), “Cougaar Overview”, BBN.
19. Wikipedia (2011), Agent-based Model, [http://en.wikipedia.org/wiki/Agent-based\\_model](http://en.wikipedia.org/wiki/Agent-based_model)



**함 원 경** (lunacy@ajou.ac.kr)

2011 아주대학교 산업정보시스템공학부 학사  
2011~현재 아주대학교 대학원 산업공학과 석사과정

관심분야 : 모델링&시뮬레이션, 국방 시뮬레이션, 분산 시뮬레이션, 합성 환경 표현



**정 용 호** (yongho1230@ajou.ac.kr)

2011 아주대학교 산업정보시스템공학부 학사  
2011~현재 아주대학교 대학원 산업공학과 석사과정

관심분야 : 모델링&시뮬레이션, 국방 시뮬레이션, 분산 시뮬레이션, 합성 환경 표현



**박 상 철** (scpark@ajou.ac.kr)

2000 Ph.D. in Industrial Engineering, Dept of I.E., KAIST, Korea  
1994 B.S. in Industrial Engineering, Dept of I.E., KAIST, Korea  
1996 M.S. in Industrial Engineering, Dept of I.E., KAIST, Korea  
2000~2001 큐빅테크, 선임연구원  
2002~2004 DaimlerChrysler ITM Dept. Research Engineer  
2008~현재 아주대학교 산업정보시스템 공학부, 부교수

관심분야 : CAD/CAM, 시뮬레이션, PLC, 이산사건 모델링