

컴포넌트 관리 및 검증도구 설계와 동적 재구성 아키텍처 기반 SAM 시뮬레이터 개발

석지범¹ · 이재오¹ · 이재진¹ · 서윤호^{1*}

A Design of Management and Verification Tool of Component and A Development of SAM Simulator based on Dynamic Reconfiguration Architecture

Jeebeom Suk · Jaeh Lee · Jaejin Lee · Yoonho Seo

ABSTRACT

Modeling and simulation techniques construct experimental environment considering battlefields and are able to analyze performance of components of weapon system that closely resemble reality. However, developed model has low scalability and not cared reusability because it has been used only in a limited range of domain. In this paper, we develop a verification tool to verify reusability of developed component for dynamic reconfiguration and to judge scalability of it and a management tool to control data of it effectively. In addition, dynamic reconfiguration architecture of guided weapon systems designed in the previous study has been applied to SAM(Surface to Air Missile) System Simulator, and we study effectiveness of the developed component. Thus the user can configure various guided weapon systems through simulation application of dynamic reconfiguration architecture of component.

Key words : M&S(Modeling and Simulation), SAM(Surface-to-air missile), Component, Simulator

요약

모델링 및 시뮬레이션 기술은 전장 환경을 고려한 실험 환경을 조성하여 무기 체계의 구성 요소에 대한 성능 분석을 실제와 유사하게 할 수 있다. 그러나 이전까지 모델링 및 시뮬레이션에서 개발된 무기체계 모델은 제한된 범위 내에서만 활용되어 확장성이 낮고 재사용성은 크게 고려하지 않았다. 이에 본 연구에서는 재사용성을 고려하여 개발된 컴포넌트에 대한 사용 검증 및 확장성 판단을 위한 검증도구와 컴포넌트를 효율적으로 관리하기 위한 관리 도구를 개발한다. 또한 이전 연구에서 설계한 유도무기체계의 동적 컴포넌트 재구성 아키텍처를 지대공유도무기 SAM(Surface to Air Missile) 교전 시뮬레이터에 적용하여 기 개발된 컴포넌트에 대한 효과도를 확인한다. 그리하여, 동적 컴포넌트 재구성 아키텍처의 시뮬레이션 적용을 통해 컴포넌트 기반의 재사용으로 유도무기체계에 대한 다양한 시스템을 구성하게 되었다.

주요어 : 모델링 및 시뮬레이션, 지대공미사일, 컴포넌트, 시뮬레이터

1. 서론

1.1 연구 배경 및 필요성

모델링 및 시뮬레이션 기술은 다양한 분야에 적용되고 있고 특히 국방 분야에서는 전력분석, 성능평가, 교육훈련 등에 특화되어 사용되고 있다. 최근 무기체계는 점점 규모가 커지고 다양해짐에 따라 단일 시스템 중심에서 복합

*본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다. (UD080042AD).

접수일(2012년 11월 2일), 심사일(2013년 5월 30일),

게재 확정일(2013년 5월 30일)

¹⁾ 고려대학교 산업경영공학과

주 저 자: 석지범

교신저자: 서윤호

E-mail; yoonhoseo@korea.ac.kr

시스템으로 다양한 개발이 요구되며, 이에 대한 시뮬레이션 기술향상이 필수적이다^[13]. 현실에서의 제약사항들을 극복하기 위한 시뮬레이터 기술 향상은 무기체계 성능평가를 위해 오래전부터 진행되어왔으며, 다양한 무기체계를 개발하고 적용하기 위해 재구성 기술이 요구된다.

재구성 기술을 통해 기존의 무기를 분석하여 공통적인 측면은 재사용하고 가변적인 측면만 추가적으로 적용하여 무기체계의 다양한 요구사항을 충족시키고 빠르게 프로토타입을 구성할 수 있다^[14, 15]. 이를 위해 소프트웨어 프로토타입 라인 공학을 적용하여 물리 기본 컴포넌트와 행위 기본 컴포넌트를 개발하였다^[1]. 이를 통해 사용자가 원하는 컴포넌트로 재구성하여 다양한 유도무기로 개발할 수 있다. 하지만 어떻게 수많은 컴포넌트들이 생성되고 관리되는지와 생성된 컴포넌트들이 바로 사용될 수 있는 지에는 문제가 있다. 그리하여 한정된 컴포넌트들의 결합이 아니라 모든 컴포넌트들의 결합이 요구되며 결합된 컴포넌트들이 하나의 시뮬레이터에서만 사용되는 것이 아니라 다양한 시뮬레이터에서 사용하기 위한 컴포넌트 상호연동을 위한 테스트가 필요하다. 이를 위해 본 연구에서는 컴포넌트 생성, 변경, 삭제 등을 위한 컴포넌트 관리 도구와 컴포넌트를 사용가능성을 판단하는 컴포넌트 검증도구를 개발하였다.

또한 기존 연구에서 설계한 동적 재구성 가능한 유도 무기체계 아키텍처를 지대공유도무기 SAM 교전 시뮬레이터에 적용하여 개발된 컴포넌트들의 효과 및 성능을 시각적으로 확인할 수 있다. SAM 교전 시뮬레이터에 개발함으로써 동적 재구성 아키텍처의 컴포넌트들의 조합으로 지대공뿐만 아니라 공대공(Air to Air Missile)등 다양한 유도무기와 그 시스템을 개발할 수 있음을 보인다.

1.2 연구 내용의 개괄

본 논문은 소프트웨어 프로토타입 라인 공학 기법을 이용하여 설계된 유도무기체계의 동적 컴포넌트 재구성 아키텍처를 기반으로 하나의 소프트웨어 어플리케이션으로 지대공미사일(Surface-to-Air Missile, SAM) 교전 시뮬레이터를 개발하여 컴포넌트 재구성을 통한 시뮬레이터의 동적 재구성과 재사용성을 확인한다. 또한 프로토타입 재구성을 위한 컴포넌트들을 관리하고 검증하는 도구를 개발하여 생성된 컴포넌트가 시뮬레이터에서 이용가능한지를 확인하며 이를 체계적이고 관리하고 이용한다.

1.3 논문의 구성

본 논문의 구성은 다음과 같다. 먼저 2장에서는 소프

트웨어 재사용에 관련된 기존 연구들에 대해 살펴본다. 3장에서는 유도무기체계 컴포넌트의 재사용 환경을 위한 컴포넌트에 대한 관리 및 검증도구를 설계하여 구현하고, 4장에서는 동적 재구성 유도무기체계 아키텍처를 기반으로 하나의 어플리케이션인 SAM 교전 시뮬레이터를 개발하여 컴포넌트들의 동적 재구성 및 재사용성을 검증한다. 마지막으로 5장에서 결론 및 향후 연구 방향으로 끝을 맺는다.

2. 관련 연구

2.1 기존 소프트웨어 재사용 연구

기존 소프트웨어 재사용 연구로는 컴포넌트 기반과 제품 라인 기반 방법론이 있다.^[2]

2.1.1 컴포넌트 기반 소프트웨어 개발 방법론

컴포넌트 기반 소프트웨어 개발(CBD : Component Based software Development) 방법론은 요구분석을 통해 컴포넌트 자체를 개발하는 방법과 개발된 컴포넌트들을 사용하여 새롭게 소프트웨어를 만드는 방법이다. 따라서 CBD 방법론을 통해 높은 품질의 소프트웨어를 효율적으로 개발할 수 있다.

2.2.2 제품 라인 소프트웨어 공학 방법론

제품 라인 소프트웨어 공학(PLSE : Product Line Software Engineering) 방법론은 소프트웨어 집합에 속하는 멤버들 사이에 공통점과 차이점을 분석하여 도메인 모델링 단계부터 이를 고려하여 효율적인 재사용이 이루어지게 하는 것이다. 프로토타입의 특징들 중에서 공통적인 특성을 이해하고, 가변적인 특성을 명백하게 구별시켜 응용 프로그램들 사이의 가변성을 지원하는데 있다.

2.2 기존 연구의 한계점

CBD 방법론과 제품 라인 공학 방법론의 목표는 각 개발 방법론이 제공하는 지침에 따라 각 도메인의 전문 개발자들이 컴포넌트를 개발하고 다른 소프트웨어 개발에 그 컴포넌트들을 재사용하는 것이다. 하지만 컴포넌트 개발 과정에서 여러 시스템에서의 재사용을 고려하여 개발되고 있지는 않다. 그리고 다양한 컴포넌트 개발 도구들이 개발되고 발전되어 왔지만 컴포넌트 재사용 노력은 한정된 범위에서 나타났고, 개발자들이 필요에 따라 다양한 컴포넌트들을 개발하여 다양한 도메인의 유도무기체계를 구성할 필요가 있다^[3-5]. 그리고 유도무기체계의 시뮬레이

터 연구 중에서 개발 단계에서부터 소프트웨어 프로덕트 라인 공학을 적용하여 컴포넌트 동적 재구성을 고려하여 컴포넌트를 개발하고 아키텍처를 설계하고 이를 구현한 연구는 찾아 볼 수 없었다.

또한 본 논문은 구성된 컴포넌트들이 조합되어 프로덕트로 동적 재구성되어 유도무기체계 시뮬레이터란 환경에서 자신의 정보를 표현하고 정확히 반영할 수 있도록 검증하고 추가, 삭제, 수정 등의 컴포넌트들을 관리를 하는 컴포넌트 관리 및 검증도구를 소개한다. 관리 도구를 통해서 컴포넌트를 생성, 수정, 삭제 등이 가능하고 이를 검증 도구를 이용하여 검증한다. 그래서 최종적으로 프로덕트 재구성 요소에 사용하여 사용자가 원하는 물리 및 행위 모습의 재구성을 지원한다.

2.3 동적 재구성 유도무기체계 아키텍처

유도무기체계 아키텍처는 유도무기 시뮬레이터 개발에 참여하는 M&S 관련 사람들 간의 의사소통 수단이며, 유도무기 시뮬레이션 설계 구성을 위한 일반적인 시스템 추상화이다. 아키텍처는 시스템의 구조와 그 시스템을 구성하는 프로덕트와의 관계, 프로덕트의 구조와 프로덕트를 구성하는 컴포넌트와의 관계, 컴포넌트의 구조를 나타낸다. 이는 시뮬레이터 개발 시 시뮬레이터의 품질에 중요한 영향을 미치는 설계요소들을 포함하고 있다. 따라서 컴포넌트를 조립하여 시뮬레이터로 개발하기 위해서 컴포넌트가 어떻게 사용될 것인가에 대한 이해가 필요하고, 이 과정은 유도무기시스템의 아키텍처에 의해 결정된다.

컴포넌트의 재사용에 의한 시뮬레이터 개발을 위하여 소프트웨어 프로덕트 라인 공학을 이용하여 공통적인 부분과 가변적인 부분으로 나눈다. 이를 바탕으로 아키텍처를 구성한다. 소프트웨어 프로덕트 라인 공학은 여러 시스템이 가지는 공통적인 특성과 각 시스템이 가지는 다양한 특성들을 포함하는 컴포넌트를 구성, 개발하기 위한 방법이다. 그래서 공통성과 가변성을 분류하기 위하여 시스템의 요구사항들을 인식하고 분석한다. 분석한 내용을 핵심자산으로 만들고 아키텍처 구조에서 기본 뼈대를 이루는 필수요소로 지정한다. 각 시스템의 특성을 나타낸 가변적 요소들은 따로 분류하여 시스템의 다양성을 표현할 수 있도록 컴포넌트의 조립에 이용되어 재구성 가능하도록 한다.^[6,8,16]

따라서 국방 M&S 분야에서는 크게 육군, 해군 공군으로 도메인을 구분하여 각 군의 대표적인 유도미사일 체계를 분석하고 표현한 아키텍처는 Fig. 1과 같다^[9]. 컴포넌트 계층, 프로덕트 계층, 시스템 계층으로 수직적인 구조

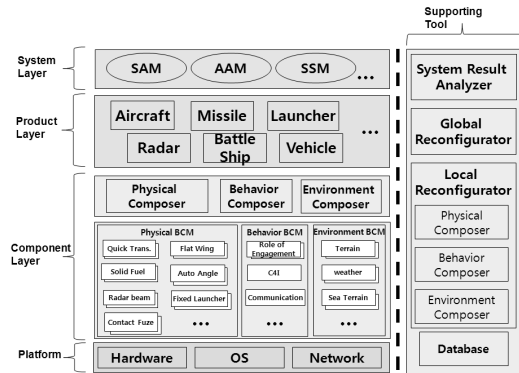


Fig. 7. Dynamic reconfiguration architecture of guided weapon system

를 갖는 이 아키텍처는 이전 동적 재구성 컴포넌트 개발 연구를 통해 설계한 것이고 이를 바탕으로 하나의 어플리케이션을 설계하기 위한 기본으로 한다. 컴포넌트 생성부터 이 아키텍처를 이용한 어플리케이션인 SAM 시뮬레이터에 적용하여 동적 재구성이 가능하다는 것을 보인다.

3. 컴포넌트 관리 및 검증도구 설계 및 구현

3.1 컴포넌트 관리 및 검증 도구 설계

유도무기체계를 컴포넌트별로 재구성하기 위해 컴포넌트를 관리하거나 검증 할 수 있는 도구가 필요하게 된다. 관리 도구를 통해 사용자는 인터페이스를 통해 컴포넌트를 쉽게 관리할 수 있게 되고 검증도구를 통해 컴포넌트 사용의 가능 여부를 확인 할 수 있게 된다.

유도무기체계에 대한 실제 데이터를 얻을 수 없는 보안상의 한계가 있어서 현실을 고려한 가상의 데이터를 생성 및 적용하여 향후 실제 데이터를 입력하였을 때 검증 및 재구성을 통해 컴포넌트의 성능을 평가 할 수 있는 가능성을 보이고자 한다.

3.1.1 컴포넌트 관리도구 설계

컴포넌트 관리 도구는 재구성된 컴포넌트들이 유도무기 시뮬레이터에서 다양한 성능이나 효과를 발휘 할 수 있게 동적 재구성 컴포넌트들을 효율적으로 관리한다. 이를 위하여 다음과 같이 필요한 기능들을 정의 하였다.

컴포넌트들을 효율적인 관리하기 위한 기능으로 크게 생성, 삭제, 수정, 조회가 가능하여야 한다.

- a. 생성 : 컴포넌트를 생성할 위치를 직접 선택하여 원

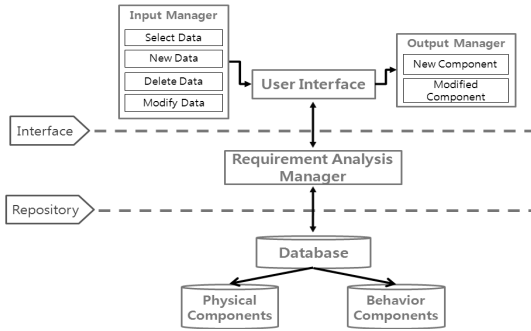


Fig. 2. Data flow of the component management tool

하는 형식에 맞게 재구성 컴포넌트를 생성할 수 있다.

- b. 삭제 : 삭제할 컴포넌트의 위치를 찾아 삭제할 수 있고 또한 연결된 실제 데이터도 삭제되어진다.
- c. 수정 : 선택한 컴포넌트에 대하여 ‘Window’를 통해 직접적으로 수정할 수 있다.
- d. 조회 : ‘Structure’를 통해 트리구조로 관리되는 실제 데이터의 입력된 컴포넌트 정보(성능, 기능, 유형 등)와 메타데이터(작성자, 날짜, 저장위치, 성능 등)를 확인할 수 있다.

이러한 관리도구의 기능들을 고려한 데이터 흐름은 Fig. 2와 같다. 사용자는 Input Manager를 통해 물리 컴포넌트 또는 행위 컴포넌트 Database에서 데이터를 불러와서 컴포넌트의 생성, 선택, 수정, 삭제 등의 기능을 통해 원하는 작업을 수행할 수 있고 User Interface를 통해 사용자가 기존 컴포넌트의 내용을 확인할 수 있다. 또한 Output Manager를 통해 새로운 컴포넌트와 기존 컴포넌트를 수정한 컴포넌트들을 생성하고 볼 수 있다.

3.1.2 컴포넌트 검증도구 설계

컴포넌트 검증을 위해 데이터 가용성과 적합성 검증, 시뮬레이터와의 연동성 검증을 통해 데이터가 특정 어플리케이션에서 시뮬레이션하기에 적합한 지에 대한 활동을 수행한다.

데이터 가용성은 해당 컴포넌트의 데이터가 목적에 맞게 언제라도 재구성 가능해야 하며 또한 데이터 적합성은 데이터가 요구된 형식에 맞아야 한다. 따라서 해당 컴포넌트들에 대한 요구 형식과 요구된 충실도에 맞는 기준을 구성할 필요가 있다. 또한 시뮬레이터와의 연동성은 해당 컴포넌트들이 재구성되어 시뮬레이터에 바로 사용될 수

있어야 한다는 것이다.

이를 바탕으로 우리가 개발한 컴포넌트 검증도구는 관리 도구를 통해 생성 또는 변경된 컴포넌트에 대해 유도 무기체계 시뮬레이터의 동적 재구성요소로서 사용가능한지 여부를 최종적으로 판단 할 수 있게 설계하였다.

우리는 3가지의 컴포넌트 검증 기준을 사용하여 사용자가 재구성 할 때 적용 가능 한 컴포넌트인지 구별 할 수 있다.

- a. 데이터 가용성(Consentaneity) : 다른 컴포넌트로 구분하기위해서 필요한 항목들이 포함되어있는지 확인한다.
- b. 데이터 적합성(Tolerance) : 컴포넌트 항목별로 인식 또는 효과를 나타낼 수 있는 허용치가 있는데 사용자가 정의해놓은 허용치에 포함되는지를 판단하게 된다. 허용치는 교전 시뮬레이터에서 인식 할 수 있는 것을 근거로 작성 하게 되고 사용자의 필요에 따라 변경하여 적용 할 수 있다.
- c. 시뮬레이터와의 연동성(Interoperability) : 같은 그룹 내의 컴포넌트를 나타내는 이름을 정해두고 번호로 추가하여 항목별로 다른 컴포넌트를 생성 하게 된다. 그래서 컴포넌트 그룹을 정의한 단어와 일치하는지 확인한다.

이러한 관련 기준을 바탕으로 검증도구의 데이터 흐름은 Fig. 3과 같다. 사용자는 Input Manager를 통해 새로운 컴포넌트와 수정된 컴포넌트를 입력하게 된다. 이 컴포넌트 정보는 User Interface를 통해 나타나고 Physical 과 Behavior 로 구분하여 내용을 판단하게 된다. 판단 기준은 3가지로 위에서 정의한 내용으로 가용성, 적합성, 연

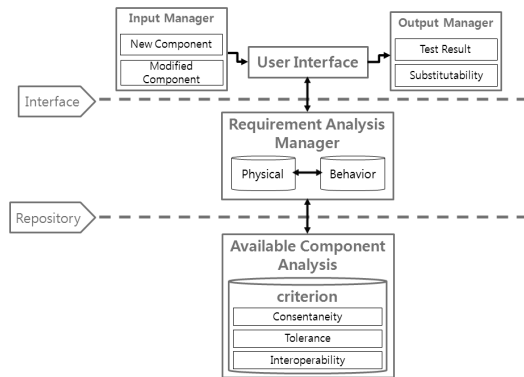


Fig. 3. Data flow of the component verification tool



Fig. 4. Component management tool

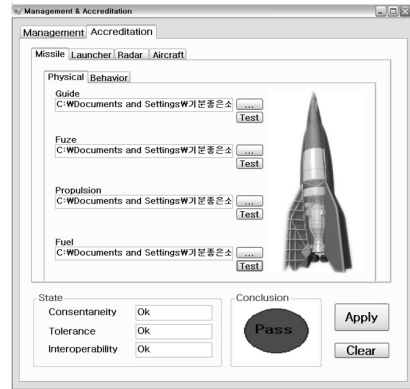


Fig. 5. Component verification tool

동성을 판단하게 된다. 이후 Output Manager를 통해 테스트하고 컴포넌트에 관한 기준별 세부 결과를 나타내고 최종적으로 컴포넌트를 이용 할 수 있는지 종합 평가를 내린다.

3.2 컴포넌트 관리 및 검증도구 개발

설계를 바탕으로 C# 언어를 이용하여 컴포넌트 관리 및 검증도구를 개발하였다. 컴포넌트 관리 도구에 필요한 기능들을 구현하였고 생성 및 수정된 컴포넌트들이 재구성에 이용될 수 있는지 판단 기준에 따라 검증할 수 있다.

3.2.1 컴포넌트 관리도구 개발

Fig. 4의 관리도구 화면의 왼쪽 ‘Structure’에서는 트리 구조로 구성된 폴더를 통해 컴포넌트들을 탐색할 수 있다. 오른쪽 ‘Window’에서는 원하는 컴포넌트를 선택하면 해당 컴포넌트 정보를 확인하고 직접 수정 할 수 있다. 확인할 수 있는 정보에는 컴포넌트 이름, 위치, 내용, 효과 등의 내용과 메타데이터에 이용되는 작성자, 날짜, 위치 등의 내용을 포함 하고 있다. 마지막 아래 ‘Information’에서는 형식에 맞춰진 메타데이터에 표시되는 내용들을 사용자가 쉽게 확인할 수 있다. 사용자는 컴포넌트에 대한 중요한 정보인 작성자, 생성된 날짜, 기능 및 효과 등을 확인 할 수 있다. 따라서 관리 도구를 이용하여 사용자는 간편하게 컴포넌트들을 관리 할 수 있다. 또한 메타데이터의 체계적인 관리를 통해 실제 컴포넌트와 연계하여 통합적인 컴포넌트 운영 ·관리가 가능하다.

3.2.2 컴포넌트 검증도구 개발

Fig. 5의 상단에서 검증하고자 하는 컴포넌트의 유형

에 따라 Missile, Launcher, Radar, Aircraft 탭을 선택하고 세부적으로 Physical 또는 Behavior 탭을 선택하게 된다. 각 내부적으로는 프로덕트에 맞는 항목이 존재하며 검증 기준도 다르다. 검증하고 싶은 해당 컴포넌트를 불러들여 Test 버튼을 눌러 검증기준에 따라 세부 검증을 실시하고 최종적으로 해당 유형에 맞는 재구성 컴포넌트로 사용가능 여부에 대한 종합 검증결정을 내린다.

검증기준은 데이터의 적합성과 가용성 그리고 시뮬레이터와의 연동성을 고려하여 정의하며 각 사항마다 세부 기준을 확립하였다. 세부 기준에 따른 컴포넌트 검증 결과에 따라 사용(Pass) 또는 불가(Fail)의 최종 검증결정을 내린다. 향후 실제 데이터에 관한 검증을 위해서 검증 기준은 사용자가 재설정이 가능하다.

예를 들면, Missile 개체를 물리적인 측면에서 핵심적인 기능으로 나눈 유도방식(Guide)에 대하여 컴포넌트를 검증하면 해당 기준에 따라 최종 사용 결정이 내려진다. 그래서 이 컴포넌트는 유도미사일을 구성하는 동적 재구성 컴포넌트로 사용될 수 있는지 없는지 판단 할 수 있게 되는 것이다. 따라서 검증 도구를 통해 사용자의 의도에 따라 생성 및 수정한 컴포넌트를 재구성에 투입하여 에러 발생을 사전에 방지하고 어떤 요소가 잘못 되었는지 확인할 수 있어서 컴포넌트 재구성을 위해 꼭 필요하다.

4. 동적 재구성 유도무기체계 아키텍처 적용사례 : SAM 시뮬레이터

4.1 SAM 교전 시뮬레이터 프레임워크 설계

컴포넌트의 재사용을 통한 시뮬레이터 개발을 위해서 동적 재구성 유도무기체계 아키텍처를 바탕으로 하나의

어플리케이션인 SAM 교전 시뮬레이터 프레임워크를 설계하였다. 프레임워크는 시스템의 주요 목적과 사용되는 개념적인 모델에 관한 구조를 나타낸다. 또한 구성하는 컴포넌트들의 그룹화 및 상호관계도 포함한다. 이러한 프레임워크는 시스템 개발 시, 요구사항을 만족하는 요소들을 선택하고 이들 간의 관계를 고려하여 개발 프로세스에 반영하여야 동적 재구성 가능한 컴포넌트 기반이 된다.

4.1.1 SAM 교전 시뮬레이터 프레임워크

SAM 교전 시뮬레이터를 개발하기에 앞서 시뮬레이터에 들어가는 컴포넌트와 프로덕트의 구성을 확인한다. 컴포넌트 구성을 위해 동적 재구성 유도무기체계 아키텍처에서 정의한 공통성과 가변성 분류를 바탕으로 우리는 시스템 층에 있는 하나의 SAM 교전 시뮬레이터를 구현하여 동적 재구성을 보이고자 한다.

개발한 어플리케이션의 프레임워크는 총 3단계의 층으로 구성한다. 시스템(System)층, 프로덕트(Product)층, 컴포넌트(Component)층으로 이루어진다. 하나의 시스템 내부에는 여러 프로덕트들로 이루어져있고 그중 하나의

프로덕트 내부에는 많은 컴포넌트들이 존재하는데 SAM 교전 시뮬레이터 프레임워크는 Fig. 6에서 보여주고 있다.

SAM 교전 시뮬레이터는 시스템층에 속하는 하나의 시스템이다. 기본적으로 SAM 교전 시뮬레이터에 필요한 프로덕트들은 Missile, Radar, Launcher, Aircraft가 필요하다. 이 프로덕트를 중심으로 내부에는 각각 여러 컴포넌트들이 존재한다. 각 프로덕트 내부에 하나의 프로덕트가 완성되기 위해서 구성되어야 할 공통적 요소들이 있고 그 요소에 해당하는 컴포넌트들은 각기 독립적으로 구성된다. 이전 시뮬레이터와 달리 SAM 교전 시뮬레이션에만 국한되어있지 않고 다양한 조합을 통해 여러 시스템에서 구성되어 사용 할 수 있다.

Fig. 7과 같이 동적 재구성 가능한 아키텍처를 이용하여 기존 컴포넌트들과 프로덕트들을 재조합하여 SAM 뿐만 아니라 ‘AAM (Air-to-Air Missile)’, ‘SSM (Surface-to-Surface Missile)’, 등 많은 시스템들을 추가로 생성할 수 있다. 이는 다양하게 재구성되어 사용자의 의도에 따라 목적에 맞는 시뮬레이터를 구성 할 수 있다.

4.1.2 SAM 교전 시뮬레이터 데이터 흐름도

컴포넌트 관리 도구를 통해 생성 및 수정된 컴포넌트를 관리하게 되고 이를 검증도구를 통해 검증하여 동적 재구성 컴포넌트로 쓰일 수 있게 된다. 다음 컴포넌트들을 조합하여 하나의 프로덕트로 만들어지게 되고 여러 프로덕트들이 모여서 사용자의 의도에 맞게 재구성되면 하나의 시스템(SAM 교전 시스템)으로 완성된다. 본 연구에서 개발한 SAM 교전 시뮬레이터는 완성된 시스템을 실행하여 재구성한 프로덕트들의 효과를 확인 할 수 있다. 즉 컴포넌트부터 시스템까지의 전체를 다루는 데이터 흐름을 프레임워크를 통해 확인 할 수 있고 동적 재구성하여 다양한 시스템 형성을 보여준다.

SAM 교전 시뮬레이터 내부의 데이터 흐름은 Fig. 8과 같다. User Interface, Requirement Analysis, Simulator

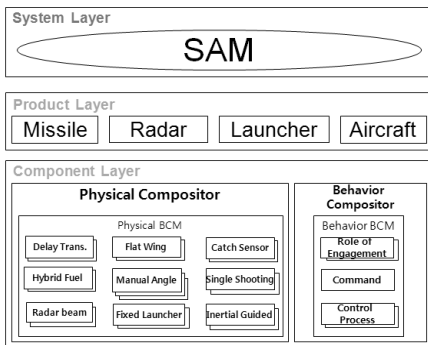


Fig. 6. Framework of SAM simulator

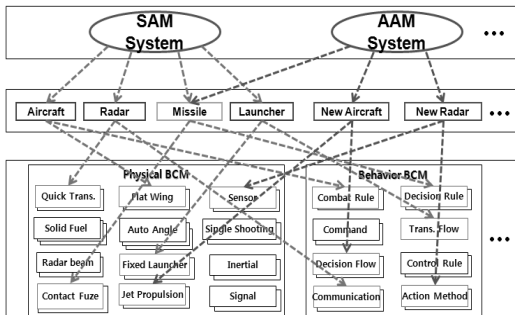


Fig. 7. Architecture-based reconfiguration of component

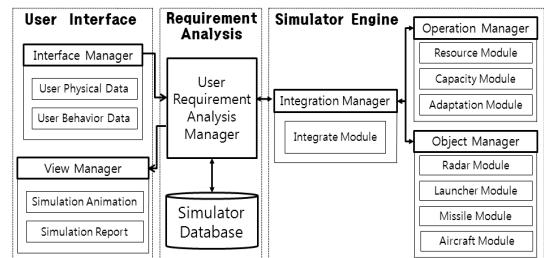


Fig. 8. Data flow of SAM simulator

Engine의 3가지 영역으로 구성된다. 먼저 컴포넌트들로 조합된 프로덕트들의 Physical 데이터와 Behavior데이터가 User Interface 영역으로 입력된다. 다음 Requirement Analysis로 전달하고 기존의 저장되어 있는 Database의 이전 데이터와 새로운 사용자 데이터를 비교하여 새로운 데이터이면 Simulator Engine에서 해당 프로덕트를 표현하는 데이터 위치에 새로운 데이터를 적용하여 통합하게 된다. 그 후 다시 통합된 데이터는 User Interface로 전달되어 애니메이션으로 실시간 효과를 확인하고 최종 결과내역을 저장 할 수 있다.

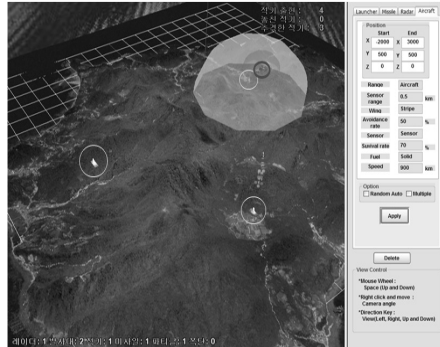


Fig. 9. Demonstration of SAM simulator

4.2 SAM 교전 시뮬레이터 정의

교전 시뮬레이터에서는 컴포넌트 조합에 따른 프로덕트들 간의 효과 및 성능을 확인 할 수 있다. 우리는 아키텍처를 바탕으로 미사일(Missile), 레이더(Radar), 발사대(Launcher)와 항공기(Aircraft)를 기본 구성으로 교전 시뮬레이터를 개발하였다. 여러 교전 방식 중 우리는 4가지 프로덕트의 구성으로 SAM 교전 시뮬레이터를 구성하고자 한다^{10, 11, 17}.

교전에서 아군을 육군, 적군을 공군으로 설정하여 SAM 교전 시뮬레이터를 구성하였다. 총 프로덕트는 4가지(미사일, 발사대, 레이더, 항공기)로 구성하면서 아군의 주요시설을 목표로 지정하여 파괴하기 위하여 이동하는 적군 항공기를 생성함에 따라 시작된다.

아군의 활동으로는 레이더가 탐지거리 안으로 적 항공기가 접근하면 표적을 추적하고 정해진 최대 탐지거리까지의 표적에 대한 정보를 실시간으로 발사대로 전송한다. 실시간으로 들어오는 적기의 정보를 바탕으로 발사대에서는 발사여부가 결정되고 미사일은 발사한다. 적군의 활동은 적 항공기가 지상의 목표물을 향해 이동하고 추격하는 미사일에 대한 레이더 감지와 회피가 가능하다.

따라서 SAM 교전 시뮬레이터 체계의 운용구성 및 임무를 요약하면 다음과 같다.

- ① 레이더 : 적기 감지 및 정보 송출
- ② 발사대 : 적기 정보 확인하여 미사일 발사
- ③ 항공기 : 목표물 폭파 및 미사일 회피
- ④ 미사일 : 항공기 추적 및 격추

이후 명중의 유무를 실시간으로 화면에 출력하고 결과에 대한 데이터를 저장 할 수 있다.

4.3 SAM 시뮬레이터 개발

물리 및 행위 컴포넌트들을 동적으로 재구성하여 필요한 프로덕트로 생성되면 SAM 교전 시뮬레이터에 필요한

Missile, Radar, Launcher, Aircraft 4개의 프로덕트가 완성되어 시뮬레이터에서 가상 교전 테스트를 할 수 있다. 이러한 프로덕트 교전 테스트가 가능한 시뮬레이터 모습은 Fig. 9와 같다.

관리 도구를 통해 사용자의 목적에 맞는 컴포넌트들을 생성 및 수정 할 수 있고 각 프로덕트에 관한 물리 컴포넌트와 행위 컴포넌트를 별개로 저장 할 수 있다. 그래서 저장한 물리모델 파일과 행위모델 파일을 불러와서 SAM 교전 상황을 테스트 할 수 있다. 또한 사용자의 필요에 의해 물리모델과 행위모델을 동적으로 재구성하여 다른 컴포넌트 조합으로 새로운 프로덕트로 만들 수 있고 재실행 가능하다. 특히 컴포넌트들로 독립적으로 구성되어 있기 때문에 이전에 재구성된 몇몇 프로덕트는 그대로 사용하면서 테스트하고 싶은 프로덕트만 재구성하여 동일 조건에서 테스트 가능하다. 그래서 컴포넌트 효과 및 성능을 개별적으로 확인가능하다.

4.4 SAM 시뮬레이터 결과 및 가상운용 실험

4.4.1 SAM 시뮬레이터 결과

SAM 교전 시뮬레이터를 통해 테스트하여 Input Data와 Output Data는 Fig. 10과 같다.

각 프로덕트가 가지는 물리/행위(P/B) 데이터가 공통적으로 입력되고 발사대, 레이더, 항공기의 시작, 끝 그리고 고정 위치를 사용자가 직접 입력하여 테스트 할 수 있다. 또한 적군의 항공기가 파괴해야할 목표물을 선택하여 전략적인 행동을 결정 할 수 있다.

교전 테스트 후 교전 결과 데이터를 출력 할 수 있다. 데이터에는 출현 한 항공기 수와 격추 된 항공기 수를 바로 확인 할 수 있고 특히 가장 중요한 항공기를 격추하였

Input Data	Output Data
<ul style="list-style-type: none"> ●Missile -P/B data 	<ul style="list-style-type: none"> ●Missile -Speed -Rotation Angle
<ul style="list-style-type: none"> ●Launcher -P/B data -Initial Position 	<ul style="list-style-type: none"> ●Launcher -Initial Position
<ul style="list-style-type: none"> ●Radar -P/B data -Position 	<ul style="list-style-type: none"> ●Radar -Detect Position
<ul style="list-style-type: none"> ●Aircraft -P/B data -Start/End Position -Target Position 	<ul style="list-style-type: none"> ●Aircraft -Interception data -Start/End Position -Target Position

Fig. 10. Input/Output data of SAM simulator

을 때 항공기와 미사일에 대한 속도, 위치 등을 확인 할 수 있다¹²⁾.

4.4.2 SAM 시뮬레이터 가상운동 실험

SAM 교전 시뮬레이터를 이용하여 사용자의 목적에 맞는 물리모델 컴포넌트들과 행위모델 컴포넌트들이 교전 시뮬레이터의 애니메이션 창을 통해 격추 테스트된다. 이를 사용자가 실시간으로 3D화면으로 확인 가능하며 결과를 확인 할 수 있다.

이에 따라 SAM 교전 시뮬레이터가 직접 운용되는 사례를 보인다. Fig. 11의 특정 물리모델 컴포넌트와 행위모델 컴포넌트들의 조합을 통해 지대공 미사일 교전상황을 나타낸 것이다. 이 상황은 아군이 아군의 기지를 목표로 하여 적 항공기가 접근하는 것을 레이더로 탐지하고, 아군의 지대공 미사일을 발사하였다. 하지만 적 항공기는 지정한 전략적 행위 방식에 따라 미사일을 좌우기동을 통하여 회피하고 아군의 기지에 폭탄을 낙하하여 목표물을 파괴하였다.

따라서 아군의 시뮬레이터 운용자는 적항공기의 전략적 행위 방식을 타파하여 아군 기지를 보호하기 위한 아군의 물리모델 컴포넌트와 행위모델 컴포넌트를 재구성한다. 앞에서 개발한 컴포넌트 관리 도구를 사용하여 기존 컴포넌트의 수정 및 새로운 컴포넌트 생성을 하며, 이를 또한 컴포넌트 검증 도구를 통해 검증하였다. 이 컴포넌트를 Fig. 12(A)와 같이 이전 연구에서 개발한 컴포넌트 재구성기를 이용하여 적 항공기의 좌우 기동에 대처 가능한 유도부기의 물리모델 및 행위모델을 구성한다. 이를 통해 변경된 아군의 물리모델이나 행위모델을 SAM 교전 시뮬레이터에 적용하여 교전상황을 재현하면 아군은 Fig. 12(B)와 같이 적 항공기의 좌우 기동에 대비하여



Fig. 11. Demo of SAM simulator in situation 1- Fail of shooting down an aircraft



(A)



(B)

Fig. 12. Demo of SAM simulator in situation 2; Dynamic reconfiguration of the component (A), Success of shooting down an aircraft (B)

재구성된 미사일 모델의 발사를 통해 적기의 격추에 성공한다. 이와 같이 추가적으로 물리모델이나 행위모델의 변경이 가능하기 때문에 테스트하고자 하는 프로덕트를 세부 컴포넌트 별로 조합하여 애니메이션 창을 통해 효과 및 결과를 확인 할 수 있다.

5. 결론 및 향후 연구 방향

그래픽기반의 컴포넌트 관리 도구를 통해서 사용자는 다양한 컴포넌트들을 쉽게 조립, 구성할 수 있게 되었고, 컴포넌트 검증 도구를 통해 컴포넌트가 다른 시뮬레이터에 바로 재사용될 수 있는지를 판단 할 수 있다. 또한 검증 기준을 다양화하여 사용자가 조건에 맞게 변경하여 적용할 수 있는 장점이 있다.

이전 연구의 동적 재구성 유도무기체계 아키텍처의 어플리케이션으로 구현된 SAM 시뮬레이터 연구를 통해 컴포넌트 생성부터 시스템 구성과 실험 결과까지 컴포넌트를 기반으로 한 동적 재구성 과정에 대한 사용자가 원하는 시뮬레이터 개발과 실험 사례를 보였다.

SAM 교전 시뮬레이터 구현을 통해 컴포넌트들의 효과를 개별적으로 확인 할 수 있고 결과 데이터에 대해서도 비교적 자세하게 얻을 수 있어서 분석용 시뮬레이터로의 이용 가능성을 보였다. 또한 SAM 교전 시뮬레이터뿐만 아니라 AAM, ASM 등 여타 교전 시뮬레이터로의 컴포넌트 재구성을 통한 개발 가능성을 확인 하였다.

향후 유도무기체계 컴포넌트 기반의 시뮬레이션을 국한된 범위를 벗어나 효율적으로 관리 및 수행하기 위하여 다양하고 많은 컴포넌트 및 프레임워크에 대한 표준화 연구와 저장소에 대한 연구가 필요하다.

References

1. J.O. Lee, J.J. Lee, J.B. Suk and Y.H. Seo (2010), "A Development of The Dynamic Reconfigurable Components based on Software Product Line : Guided Weapon System", Journal of the Korea Society for Simulation, Vol. 19, No. 4, pp. 179-188.
2. I.Y. Ko, H.K. Jo, H.J. Choi, S.D. Kim, S.J Park (2007), "Design of a Defense-Software Component-Grid Environment", Journal of KIISE, Vol. 25, No. 9, pp. 5-17.
3. Agency for Defence Development (2006), Defense-CBD Methodology, ADDMe Ver1.1.
4. S.Y. Choi (2003), "Research of a Development Methodology of Software of Weapon System", Korea National Defence University.
5. H.J. Ha, M.K. Moon, K.H. Yeom (2010), "A Method of Domain Analysis and Design for Component - based Software Development", Journal of KIISE, Vol. 28, No. 10, pp. 743-756.
6. H.G. Moon, S.G. Yu, T.H. Kim (2008), "A Practical Application to OenSAF Model", Agency for Defence Development, pp. 130-140.
7. S.Y. Lee, O.C. Kwon, G.S. Shin (2003), "A Method for Architecture-based Design and Implementation of Component Assembly and its Tool Support", Journal of KIISE, Vol. 25, No. 9-10, pp. 812-820.
8. T.S. Kim, H.J. Chang, J.M. Lee, K.S. Lee (2010), "A Modeling & Simulation Engine for Analyzing Weapons Effectiveness : Architecture", Journal of the Korea Society for Simulation, Vol. 19, No. 2, pp. 51-62.
9. Korea Military Academy (2002), "Engineering of the Weapon System", Bookshill, pp. 53-79.
10. H.R. Kim, J.Y. Lee, S.Y. Choi (2010), "A Proposal of Defense Modeling and Simulation Development Framework and Application for the Development of Distributed Air Defense Simulator", Proc. of the Conference on Korea Society for Internet Information, pp. 583-588.
11. B.G. Cho, S.H. Kim, C Youn (2005), "A Design and Application of HLA-Based Air Defense Simulation Framework", Journal of KIISE, Vol. 12-D, No. 5, pp. 709-718.
12. K.H. Kim, Y.H. Seo (2010), "Development of Simulator for Analyzing Intercept Performance of Surface-to-air Missile", Journal of the Korea Society for Simulation, Vol. 19, No. 1, pp. 63-71.
13. B. Zeigler, Tag Gon Kim, H. Prachfer, "Theory of Modeling and Simulation", Academic Press, 2000.
14. Chou, S.C. and Chen, Y.C., "Retrieving reusable components with variation points from software product lines", Information processing letters, 99(3), pp. 106-110, 2006.
15. Clements, Paul and Northrop, Linda, Software Product Lines: Practices and Patterns, Addison-Wesley, pp. 5-50, 2002.
16. Henderson, C. and Rodriguez, A., "Modeling in OneSAF", Computer generated forces and behavioral representation, pp. 337-348, 2002.
17. Giampapa, J. A., Sycara, K., Owens, S., Glington, R., Seo, Y.-W., Yu, B., Grindle, C. E. and Lewis, M., "Extending the OneSAF Testbed into a C4ISR Testbed", Simulation, 80(12), pp. 681-691, 2004.



석 지 범 (sjb1010@korea.ac.kr)

2010 고려대학교 산업공학과 학사
2010~현재 고려대학교 산업경영공학과 석사 과정

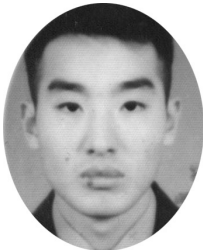
관심분야 : 모델링&시뮬레이션, 제조시스템설계, 메타휴리스틱



이 재 오 (etozt@korea.ac.kr)

2009 인제대학교 산업공학과 학사
2011 고려대학교 정보경영공학부 석사

관심분야 : 소프트웨어 프로덕트 라인, 모델링&시뮬레이션, 스케줄링



이 재 진 (ongs6455@korea.ac.kr)

1999 육군사관학교 토목공학 학사
2008 University of Texas at Austin 운영분석 석사
2010~현재 고려대학교 산업경영공학과 박사 과정

관심분야 : Supply chain management, 메타휴리스틱, 모델링&시뮬레이션



서 윤 호 (yoonhoseo@korea.ac.kr)

1984 고려대학교 산업공학과 학사
1990 미국 Pennsylvania State University 산업공학과 석사
1993 미국 Pennsylvania State University 산업공학과 박사
1993~2003 울산대학교 산업공학과 교수
2003~현재 고려대학교 산업경영공학과 교수

관심분야 : 제조, 조립 및 물류 시스템의 VR적용 스케줄링