

소규모 교전을 위한 시나리오 생성 프로세스 제안

안의국¹ · 고민석¹ · 천상욱² · 박상철^{1†}

¹아주대학교 산업공학과, ²대우조선해양 중앙연구소 정보기술팀

A Proposal of Scenario Generation Process for Small Scale Engagement

Euikoog Ahn¹, Minsuk Ko¹, Sang Uk Cheon², and Sang Chul Park^{1†}

¹Department of Industrial Engineering, Ajou University

²Daewoo Shipbuilding & Marine Engineering

Received 14 February 2012; received in revised form 12 January 2013; accepted 15 February 2013

ABSTRACT

Computer-based modeling and simulation (M&S) techniques have become an essential component in the development of new weapons systems. M&S techniques provide a means to simulate military training, strategies, military doctrines, and weapons acquisition processes. This paper proposes a small scale engagement scenario generation method. This work also includes a process for scenario generation and visualization. The proposed scenario generation methodology employs the Timed-FSA (finite state automata) and DFS (depth first search) algorithms. The proposed scenario generation method is verified using a one-on-one combat engagement scenario between two submarines. In addition, we suggest a scenario generation process including whole scenario generation and scenario visualization.

Key Words: Defense M&S, Scenario generation, Small scale engagement

1. 서 론

세계의 각국은 21세기 변화된 안보환경에서의 미래 전에 효과적인 대응을 위해 선진국을 중심으로 군사력을 새롭게 정비하고 있다. 따라서 미래의 전장환경은 복잡한 임무와 임무공간 확대, 무기체계 및 작전 계획 복잡도가 높아지고 있다. 또한 미래전은 정보통신, 센서 유도항법 기술의 발전에 따라 네트워크 중심의 작전환경이 조성되고, 이를 적용해 복합무기체계가 운용될 것이다^[1]. 따라서 요구되는 무기체계도 과거보다 복잡해졌

으며, 운용시나리오 역시 다양하게 요구된다^[2]. 이러한 요구의 충족을 위해 최근에 주목받는 기술이 Modeling & Simulation(M&S)이다.

국방 M&S 기술은 군사훈련, 전력분석, 국방획득 분야를 아우르는 개념으로 다양한 전투 요소에 대한 연구, 조사, 분석 수단을 제공한다^[3]. 하지만 국방 M&S를 위한 운용 시나리오는 적군의 예측하기 힘든 모든 행동에 대해 미리 준비하기는 어렵다. 따라서 지휘관이 미처 예상하기 힘든 시나리오에 대한 시나리오에 대하여 미리 생성하고, 이를 사전 모의하는 기술에 대한 연구가 필요하다.

본 연구에서는 소규모 교전을 대상으로 시나리오 생성을 위한 방법론을 제시한다. 제시한 방법론을 통해 지휘관이 생성 가능한 모든 시나리오의

[†]Corresponding Author, scpark@ajou.ac.kr
©2013 Society of CAD/CAM Engineers

생성을 지원한다. 또한 생성한 시나리오 중 지휘관이 주로 관심을 갖고 있는 시나리오의 추출 방안을 제시한다. 또한 이렇게 생성한 시나리오의 시물레이션 및 가시화를 방안을 제안한다.

시나리오 생성에 대한 기존 연구를 살펴보면 주로 사회 과학 분야에서 수요예측, 미래의 경향 예측 등을 위한 의사 결정 지원을 위해 시나리오 생성 프레임워크를 제안한 연구로 위험 요소를 미리 측정하고, 관리하는 방법론을 제안했다^[4,5].

또한 제품을 설계하기 위해 가능한 시나리오를 목적과 특성에 따라 구분하고, 시나리오의 관계를 정의하는 시나리오 기반 제품설계 방법론을 제안한 연구^[6]와 시물레이션 기반 교육을 위해 시나리오 자동 생성 프레임워크를 제안한 연구가 있다^[7]. 하지만, 개념적인 프레임워크를 주로 하였기 때문에 도메인의 특성을 효율적으로 반영하기에는 한계가 있다.

게임 분야에서 이뤄진 시나리오 생성에 대한 연구로는 시나리오가 갖는 인터랙티브 요소를 중심으로 게임 시나리오 작성을 위한 방안을 소개한 연구^[8]와 시나리오 저작도구를 개발하기 위해서 캐릭터의 세계관, 시대 배경 등 일반적인 시나리오를 표현하는 시놉시스와 일정한 목표를 달성했을 때 다음 단계로 이동하는 퀘스트로 구성된 시나리오를 표현 방안에 대한 연구가 있다^[9].

그리고, 국방 분야의 시나리오 생성 저작도구의 개발 사례는 캐나다의 eNGENUITY에서 개발한 시나리오 기반 M&S도구인 STAGE(Scenario Tactic and Generation Environment)^[10]와 미국 MAK에서 개발한 VR-Forces^[11]가 있다. STAGE의 시나리오는 GUI기반으로 모델을 배치하고, 상세한 플랫폼 모델을 표현해 시나리오를 생성, 관리, 수정하는 기능을 제공한다^[12]. 또한 VR-Forces는 시나리오 생성을 위해 계층화, 세분화된 작전과 행위 표현을 제공하며, 맵을 기반으로 플랫폼의 배치, 행위를 정의해 시나리오를 생성한다^[11]. 하지만 위의 저작도구는 데이터베이스, 시물레이션 엔진, 개발 Toolkit 등 방대한 분야를 다루고 있어 시나리오 생성에 특화된 저작도구로 보기는 힘들다.

또한 국방 분야에서 소규모 교전을 대상으로 시물레이션하기 위해 시나리오를 생성한 연구^[12,13]가 있으나 시나리오 생성이 주목적인 연구가 아니므로 단순한 교전 시나리오를 UML 등을 이용해 표현하였다.

본 논문의 목적은 소규모 교전을 위한 시나리오 생성 방안을 제시하는 것이다. 이후 본 논문의 구성은 2장에서 Timed-FSA(Finite State Automata) 방법론을 이용한 접근 방법과 DFS(Depth First Search) 알고리즘을 이용해 상태를 탐색해 시나리오를 생성하는 방안을 소개하고, 3장에서 소규모 교전 시나리오 생성 방안을 설명하고, 이렇게 생성한 시나리오의 시물레이션 및 가시화를 방안을 제안한다. 마지막으로 4장에서 결론을 맺는다.

2. 접근방법

본 연구에서는 Timed-FSA(Finite State Automata)를 이용해 객체를 모델링 한다. Timed-FSA는 상태와 이벤트의 집합으로 시스템을 기술하는 형식론으로, 원자모델(Atomic model)로 표현한다. 원자모델은 입력 사건에 의한 외부상태전이(external transition)와 시간 진행에 따른 내부상태전이(internal transition)의 기술을 위해 7개의 tuple로 정의된다^[14,15]. 원자모델은 시간명세천이(Timed State Transition) 레벨에서 시스템의 동작을 표현하며, 원자모델은 시스템의 상태를 State로 표현한다. State는 외부에서 입력이 들어오거나 미리 정의한 시간이 흐르면 다른 State로 변한다^[16]. 원자모델 M의 정의는 다음과 같다^[14,15].

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

X: input events set;

S: sequential states set;

Y: output events set;

δ_{int} : $S \rightarrow S$: internal transition function;

δ_{ext} : $Q * X \rightarrow S$: external transition function

$Q = \{(s, e) | s \in S, 0 \leq e \leq t_a(s)\}$: total state of M;

λ : $S \rightarrow Y$: output function;

t_a : $S \rightarrow Real$: time advance function.

다음은 잠수함이 정찰하다가 적을 발견하면 보고하고, 다시 정찰하는 것에 대한 모델링의 한 예를 보여주는 것으로 아래에 tuple에 의해 정의된 것을 나타내며, Fig. 1은 이를 상태 천이 다이어그램으로 표현한 것이다.

X = {found}

Y = {report}

S = {Discover, Search}

$\delta_{int}(\text{Discover}) = \text{Search}$
 $\delta_{ext}(\text{Search, found}) = \text{Discover}$
 $\lambda(\text{Discover}) = \text{report}$
 $t_a(\text{Report}) = \text{reporting time (random number)}$

Timed FSA를 통해 생성한 시나리오의 탐색을 위해 DFS(Depth First Search) 알고리즘을 이용했다. DFS알고리즘은 시작 노드를 선택하고, 이 노드에서 탐색순서를 수직으로 깊은 곳에 있는 자식 노드를 검색한 후 옆 노드를 검색하는 탐색 방식을 따르는 알고리즘이다^[17]. 이러한 과정을 목표로 하는 노드에 도달할 때까지 반복한다. Fig. 2는 DFS알고리즘에 의해 탐색하는 방법을 보여주는 그림이다. DFS의 시작은 1번 노드에서 시작되어 더 이상 자식 노드가 없는 Leaf노드가 될 때까지 탐색을 반복하고, Leaf노드이면 상위의 노드를 탐색한다. Fig. 2의 탐색 순서는 1, 2-1, 3-1, 4-1, 4-2, 3-2, 2-2이다.

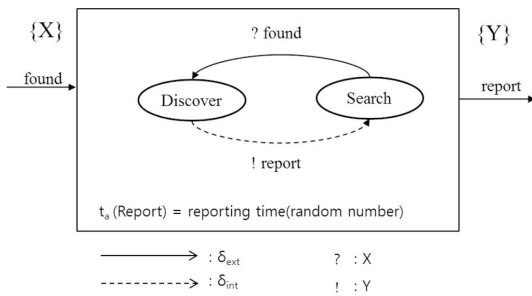


Fig. 1 The example of state transition diagram

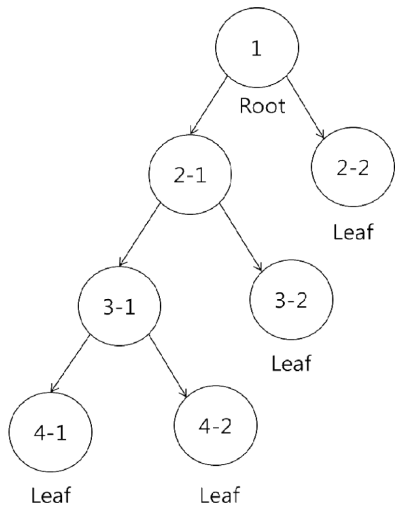


Fig. 2 The example of DFS

3. 소규모 교전 시나리오 생성 방안

3.1 시나리오 생성 방안

본 논문에서는 공격함 M1과 정찰함 M2에 대하여 Timed-FSA 방법론을 이용해 정의하였다. M1에 대한 상태전이 다이어그램은 Fig. 3과 같다.

M1은 공격함이며, 출발하여 목적지까지 도착하는 것이 목표이다. 이때 정찰함을 발견하면 그 정찰함을 공격하는 과정이 포함되는 모델이다. 모델 M1의 이벤트는 Arv(Attack sub is Arrived), Es(Effect search of reconnaissance sub), Efl(Effect to discovery of reconnaissance-c sub), Atk(attack of reconnaissance sub)으로 구성되며, 상태는 AS(Attack Start), APr(attack pre-attack), AEDA(damage assessment of attack effect), AMC(attack sub mission complete)이다. 이 모델은 AS에서 정찰함을 발견하는 Efl 이벤트가 입력 값으로 들어올 때 공격 준비 중인 상태를 의미하는 APr로 천이 된다. 또한 APr 상태에서 공격 준비하는 시간이 지나면 정찰함을 향해 공격을 하는 출력 값인 Atk이 내부 상태천이가 발생한다. 그 후 만약 정찰함이 파괴되었다는 이벤트 값인 Es를 입력 받으면 다시 APr 상태가 되어 정찰함을 향해 공격하는 과정을 반복한다. 그리고, AS, APr, AEDA 상태에서 목적지에 도착하여 Arv이벤트가 입력되면 AMC 상태가 되어 상황이 종료된다. M1에 대한 정의는 다음과 같다.

- X = {Arv, Efl, Es}
- Y = {Atk}
- S = {AS, APr, AEDA, AMC}
- $\delta_{int}(\text{APr}) = \text{AEDA}$
- $\delta_{ext}(\text{AS, Efl}) = \text{APr}$
- $\delta_{ext}(\text{AS, Arv}) = \text{AMC}$
- $\delta_{ext}(\text{APr, Arv}) = \text{AMC}$
- $\delta_{ext}(\text{AEDA, Es}) = \text{APr}$

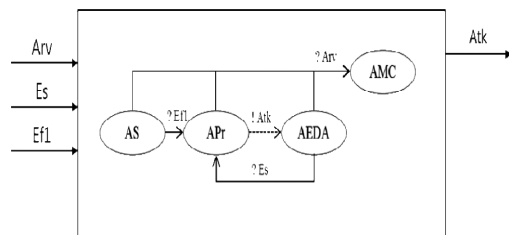


Fig. 3 State transition diagram of M1

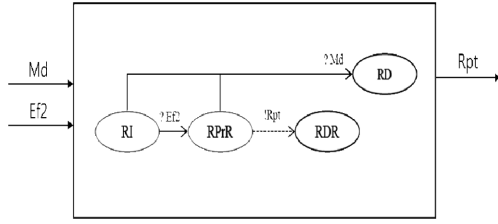


Fig. 4 State transition diagram of M2

$$\delta_{ext}(AEDA, Arv) = AMC$$

$$\lambda(APr) = Atk$$

$$t_a(Atk) = \text{Attacking time (random number)}$$

M2에 대한 상태전이 다이어그램은 Fig. 4와 같다.

M2는 정찰함으로 공격함을 발견하면 본부에 보고한다. 만약 공격함의 공격을 받아 최대의 피해를 입으면 정찰함은 파괴된다. 모델 M2의 이벤트는 Md(reconnai-ssance sub is maximum damaged), Ef2(Effect to discovery of attack sub) and Rpt(report of attack sub), 상태는 RI(reconnaissance sub initiation), RPrR(reconnaissance sub prepare report), RDR(reconnaissance done report), RD (destroyed reconnaissance sub)로 구성된다. 이 모델은 초기 상태인 RI에서 공격함을 발견하는 Ef2 이벤트가 입력 값으로 들어와 보고 준비 상태를 의미하는 RPrR로 천이 된다. 또한 RPrR 상태에서 보고를 준비하는 시간이 지나면 공격함이 발견되었음을 보고하는 출력 값인 Rpt가 내부 상태천이

가 발생하여 보고가 완료된 상태인 RPrR 상태가 되어 상황이 종료된다. 만약 RI, RPrR 상태에서 Md이벤트가 입력되면 정찰함이 파괴되었음을 의미하는 RD 상태가 되어도 역시 상황이 종료된다. M2에 대한 정의를 살펴보면 다음과 같다.

$$X = \{Md, Ef2\}$$

$$Y = \{Rpt\}$$

$$S = \{RI, RPrR, RDR, RD\}$$

$$\delta_{int}(RPrR) = RDR$$

$$\delta_{ext}(RI, Ef2) = RPrR$$

$$\delta_{ext}(RI, Md) = RD$$

$$\delta_{ext}(RPrR, Md) = RD$$

$$\lambda(RPrR) = Rpt$$

$$ta = \text{time advance function}$$

$$t_a(Rpt) = \text{reporting time (random number)}$$

본 연구에서 제안하는 시나리오 생성 방안은 다음과 같다. 우선 위와 같은 Timed-FSA 방법론을 이용해 객체에 대한 모델링을 수행한다. 그리고 모델링 한 객체를 합성해 상태와 이벤트의 관계를 표현한다. 원자모델의 합성을 위해 상태는 병렬로 붙여서 상태를 재정의하고, 각 상태에서 다음 상태로 천이되는 이벤트들을 연결한다. 이렇게 표현한 결과는 Fig. 5와 같다.

그 이후 두 객체 간의 제약사항을 고려해 Table 1과 같이 제약 테이블(Conflict table)을 작성한다. 예를 들어 공격함이 AS인 상태로 아직 출발 중인 상태일 때 정찰함이 최대 피해를 입는 이벤트인

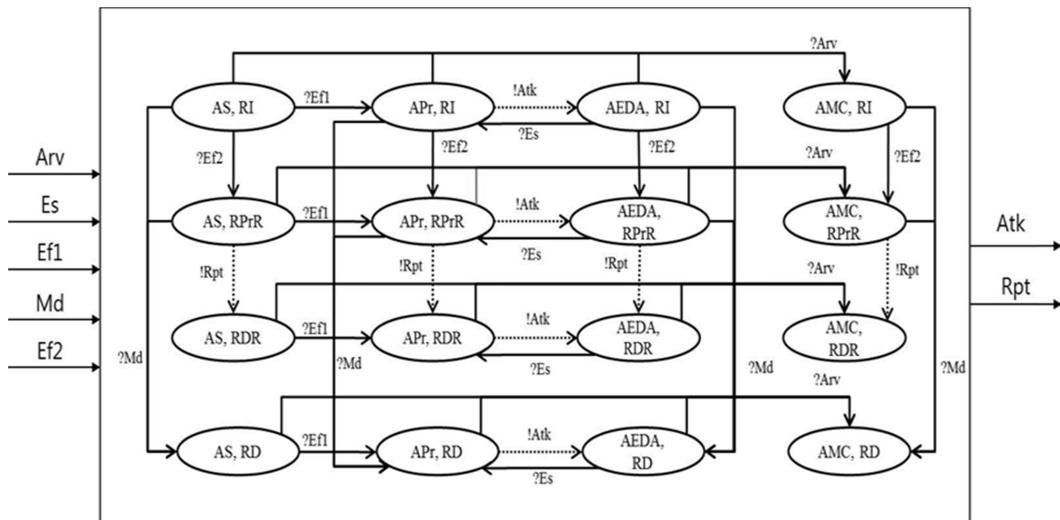


Fig. 5 Result of the composition of state transition diagram in M1 and M2

Table 1 Conflict table

	Ef2	Rpt	Md
AS	○	○	×
APr	○	○	×
AEDA	○	○	○
AMC	×	×	×

	Ef1	Atk	Es	Arv
RI	○	○	○	○
RPrR	○	○	○	○
RDR	×	×	×	×
RD	×	×	×	×

Md 이벤트의 발생은 논리적으로 불가능하게 된다. 따라서 이러한 상황을 제약 테이블을 통해 입력 받아 두 객체 간의 불필요한 상태와 이벤트를 제거할 수 있다. 제약 테이블의 구성은 Table 1과 같다.

여기에 ○로 표시된 것은 상태와 이벤트가 서로 연결되어 있음을 의미하며, ×는 서로 연결하지 않은 것을 의미한다. 예를 들어 AS상태의 경우 Ef2, Rpt와는 연결되었으나 Md와는 연결되지 않았기 때문에 Fig. 5에서 AS상태와 연결된 Md 이벤트를 모두 제거한다. 이 과정을 모든 상태에 대해 반복해 수행하면, 상태 별로 의미있는 이벤트에 대한 연결만 남게 된다. 만약 상태가 어느 이벤트와도

연결되지 않았다면 그 상태 역시 제거한다. Fig. 6은 제약 테이블을 적용한 결과이다.

이 결과에 대하여 두 객체 모두 시작하는 시점인 AS, RI를 Root노드로 정의하고, M1, M2 중 하나의 상태가 종료하면 상태를 목표 노드로 간주해 DFS 알고리즘에 의해 탐색하면 M1과 M2 사이에 발생 가능한 모든 시나리오를 생성할 수 있다. 이때 DFS 알고리즘은 사이클이 없는 경우에 대해서만 표현하는 것이 가능하다. 따라서 루프 상황에 대한 표현 방안이 필요하다. 모든 이벤트의 부모 노드와 자식 노드의 정보를 가져온다. 이때 DFS 알고리즘은 사이클이 없는 상황 대한 탐색 기법이기 때문에 사이클 형식으로 표현하기 위해 이벤트를 기준으로 부모 노드와 자식 노드를 모두 가져와 이벤트를 서로 비교하여 이벤트의 부모 노드와 자식 노드가 서로 반대인 경우를 루프 상태로 정의했다. 예를 들어 이벤트 a의 부모 노드가 P고 자식 노드가 C인데 다른 이벤트 b의 부모 노드가 C이고, 자식 노드가 P인 경우가 있다면 P노드와 C노드는 서로 루프 상태로 표현할 수 있다. 이와 같이 루프 상태인 노드의 연결 상태는 따로 저장하여 루프 상태인 경우는 중괄호로 묶고 *를 붙여 구분하였다.

[(AS, RI) (AS, RPrR) (AS, RDR)]
 [(AS, RI) (AS, RPrR) (AMC, RPrR)]

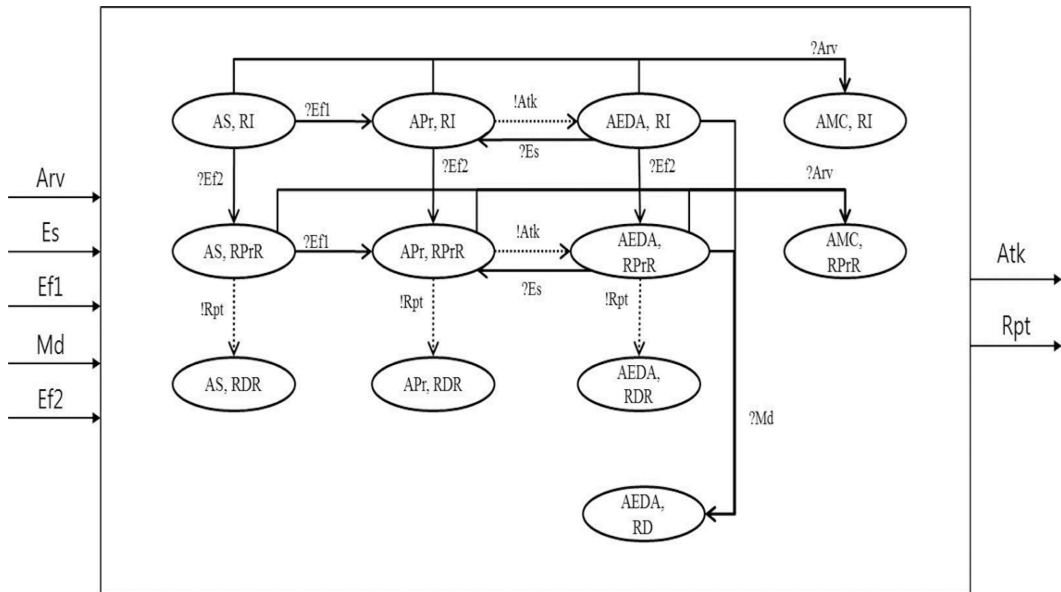


Fig. 6 Result of the composition of state transition diagram , except for the constraint between M1 and M2

[(AS, RI) (AS, RPrR) (APR, RPrR) (AMC, RPrR)]
 [(AS, RI) (AS, RPrR) (APR, RPrR) (APR, RDR)]
 [(AS, RI) (AS, RPrR) {(APR, RPrR) (AEDA, RPrR)}*(AMC, RPrR)]
 [(AS, RI) (AS, RPrR) {(APR, RPrR) (AEDA, RPrR)}*(AEDA, RDR)]
 [(AS, RI) (AS, RPrR) {(APR, RPrR) (AEDA, RPrR)}*(AEDA, RD)]
 [(AS, RI) (AS, RPrR) {(APR, RPrR) (AEDA, RPrR) (APR, RPrR)}*(AMC, RPrR)]
 [(AS, RI) (AS, RPrR) {(APR, RPrR) (AEDA, RPrR) (APR, RPrR)}*(APR, RDR)]
 [(AS, RI) (AS, RI)]
 [(AS, RI) (APR, RI) (AMC, RI)]
 [(AS, RI) (APR, RI) (APR, RPrR) (AMC, RPrR)]
 [(AS, RI) (APR, RI) (APR, RPrR) (APR, RDR)]
 [(AS, RI) (APR, RI) {(APR, RPrR) (AEDA, RPrR)}*(AMC, RPrR)]
 [(AS, RI) (APR, RI) {(APR, RPrR) (AEDA, RPrR)}*(AEDA, RDR)]
 [(AS, RI) (APR, RI) {(APR, RPrR) (AEDA, RPrR)}*(AEDA, RD)]
 [(AS, RI) (APR, RI) {(APR, RPrR) (AEDA, RPrR) (APR, RPrR)}*(AMC, RPrR)]
 [(AS, RI) (APR, RI) {(APR, RPrR) (AEDA, RPrR) (APR, RPrR)}*(APR, RDR)]
 [(AS, RI) {(APr, RI) (AEDA, RI)}*(AMC, RI)]
 [(AS, RI) {(APr, RI) (AEDA, RI)}*(AEDA, RD)]
 [(AS, RI) {(APr, RI) (AEDA, RI)}*(AEDA RPrR) (AMC, RPrR)]
 [(AS, RI) {(APr, RI) (AEDA, RI)}*(AEDA RPrR) (AEDA, RDR)]
 [(AS, RI) {(APr, RI) (AEDA, RI)}*(AEDA RPrR) (AEDA, RD)]
 [(AS, RI) {(APr, RI) (AEDA, RI) (APr, RI)}*(AMC, RI)]
 [(AS, RI) {(APr, RI) (AEDA, RI) (APr, RI)}*(APr, RPrR) (AMC, RPrR)]
 [(AS, RI) {(APr, RI) (AEDA, RI) (APr, RI)}*(APr, RPrR) (APR, RDR)]
 [(AS, RI) {(APr, RI) (AEDA, RI) (APr, RI)}*(APr, RPrR) (AEDA, RPrR)}*(AS, RPrR)]

[(AS, RI) {(APr, RI) (AEDA, RI) (APr, RI)}*
 {(APr, RPrR) (AEDA, RPrR)}*(AEDA, RDR)]
 [(AS, RI) {(APr, RI) (AEDA, RI) (APr, RI)}*(APr, RPrR) (AEDA, RPrR)}*(AEDA, RD)]
 [(AS, RI) {(APr, RI) (AEDA, RI) (APr, RI)}*(APr, RPrR) (AEDA, RPrR) (APR, RPrR)}*(AMC, RPrR)]
 [(AS, RI) {(APr, RI) (AEDA, RI) (APr, RI)}*(APr, RPrR) (AEDA, RPrR) (APR, RPrR)}*(APr, RDR)]

예를 들어 생성한 시나리오 중 {(AS, RI) (AS, RPrR) (APr, RPrR) (AS, RPrR)}를 보면 처음 공격함과 정찰함 모두 대기 중인 상태에서 공격함은 여전히 대기 중인데 정찰함이 공격함을 발견해 보고할 준비 중인 상태로 바뀐 후 공격함 역시 정찰함을 발견해 공격함이 타격 준비 중인 상태로 바뀐 반면 정찰함은 여전히 보고 준비 중인 상태에서 이후 정찰함이 여전히 보고 준비 중인 상태에서 공격함이 본부의 명령에 의해 목적지로 도착하게 되어 시나리오가 종료되는 상황을 표현한다.

3.2 시나리오 생성기 구현

제안한 방안을 이용해 생성 가능한 모든 시나리오를 생성하는 시나리오 생성기를 개발했다. 개발한 시나리오 생성기는 Microsoft Visual Studio 2008 MFC를 이용했다. 개발한 시나리오 생성기는 사용자가 객체에 대해 모델링하고, 모델링해 입력하면 이 입력값에 대하여 합성하고, 제약 테이블을 입력 받아 의미없는 두 객체 사이에 불필요한 상태와 이벤트를 제거하고, 생성 가능한 시나리오를 출

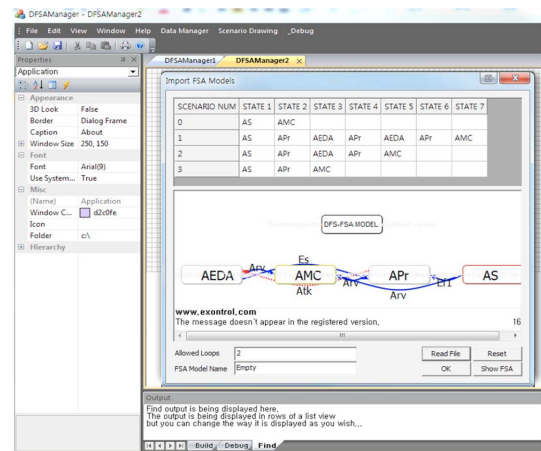


Fig. 7 Screen-shot of scenario generator

SCENARIO_01	AS:RI	APr:RI	AEDA:RI	APr:RI	AMC:RI	AMC:RD
SCENARIO_11	AS:RI	APr:RI	AMC:RI	AMC:RD		
SCENARIO_2	AS:RI	APr:RI	AEDA:RI	APr:RI	AMC:RI	AMC:RPfAMC:RD
SCENARIO_13	AS:RI	APr:RI	AMC:RI	AMC:RPfAMC:RD		
SCENARIO_4	AS:RI	APr:RI	APr:RPfR	AEDA:RPfAPr:RPfR	AMC:RPfAMC:RD	
SCENARIO_5	AS:RI	AS:RPfR	APr:RPfR	AEDA:RPfAPr:RPfR	AMC:RPfAMC:RD	
SCENARIO_6	AS:RI	APr:RI	AEDA:RI	APr:RI	APr:RPfR	AMC:RPfAMC:RD
SCENARIO_7	AS:RI	APr:RI	APr:RPfR	AMC:RPfAMC:RD		
SCENARIO_8	AS:RI	AS:RPfR	APr:RPfR	AMC:RPfAMC:RD		
SCENARIO_9	AS:RI	APr:RI	APr:RPfR	APr:RD	AEDA:RD	APr:RD AMC:RD
SCENARIO_10	AS:RI	AS:RPfR	APr:RPfR	APr:RD	AEDA:RD	APr:RD AMC:RD
SCENARIO_11	AS:RI	AS:RPfR	AS:RD	APr:RD	AEDA:RD	APr:RD AMC:RD
SCENARIO_12	AS:RI	APr:RI	APr:RPfR	AEDA:RPfAPr:RPfR	APr:RD	AMC:RD
SCENARIO_13	AS:RI	AS:RPfR	APr:RPfR	AEDA:RPfAPr:RPfR	APr:RD	AMC:RD
SCENARIO_14	AS:RI	APr:RI	AEDA:RI	APr:RI	APr:RPfR	APr:RD AMC:RD
SCENARIO_15	AS:RI	APr:RI	APr:RPfR	APr:RD	AMC:RD	
SCENARIO_16	AS:RI	AS:RPfR	APr:RPfR	APr:RD	AMC:RD	
SCENARIO_17	AS:RI	AS:RPfR	AS:RD	APr:RD	AMC:RD	
SCENARIO_18	AS:RI	AS:RPfR	AS:RD	AMC:RD		
SCENARIO_19	AS:RI	APr:RI	AEDA:RI	APr:RI	AMC:RI	AMC:RPfAMC:RDR
SCENARIO_20	AS:RI	APr:RI	AMC:RI	AMC:RPfAMC:RDR		
SCENARIO_21	AS:RI	APr:RI	APr:RPfR	AEDA:RPfAPr:RPfR	AMC:RPfAMC:RDR	
SCENARIO_22	AS:RI	AS:RPfR	APr:RPfR	AEDA:RPfAPr:RPfR	AMC:RPfAMC:RDR	
SCENARIO_23	AS:RI	APr:RI	AEDA:RI	APr:RI	APr:RPfR	AMC:RPfAMC:RDR

Fig. 8 Result of scenario generation

력하도록 했다. 이때 루프 상황은 숫자로 입력 받아 해당 숫자만큼 반복하는 것을 허용하도록 했다. 생성한 시나리오에는 이는 Fig. 7은 구현한 시나리오 생성기를 보여주고 있으며, Fig. 8은 생성한 시나리오의 결과를 출력한 것이다.

3.3 시나리오 가시화

본 연구에서는 객체에 대하여 모델링 한 결과를 입력 값으로 받아 시나리오 생성기를 이용해 자동으로 생성하고, 이렇게 생성 시나리오의 로그파일을 입력받아 가시화했다. 가시화를 위해 미 해군 대학원의 MOVES Institute에서 개발한 Delta 3D를 이용해 가시화했다. Delta 3D는 전장 시뮬레이션에 필요한 전투 객체, 지형, 3차원 특수효과 기능 등을 제공하며, 엔진의 모든 소스가 개방되어 있기 때문에 높은 접근성을 가지며, 상용화된 API들과의 연동이 용이하다^[18]. 예제 시나리오에는 [(AS, RI), {(APR, RI) (AEDA, RI)}*(AEDA, RD)]에 대하여 구현하였다. Fig. 9는 공격함과 정찰함

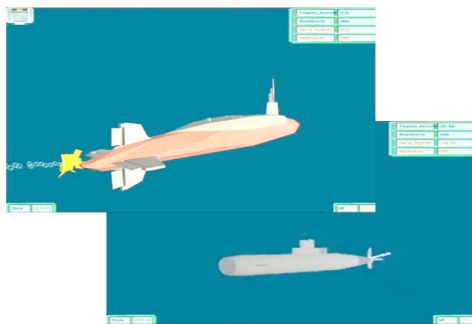


Fig. 9 STEP 1 (AS, RI)

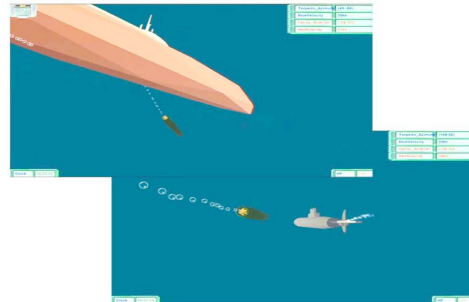


Fig. 10 STEP 2 {(APR, RI) (AEDA, RI)}*



Fig. 11 STEP 3 (AEDA, RD)

모두 초기 상태를 의미하며, Fig. 10은 공격함이 정찰함을 발견해 공격 준비 후 타격하는 것을 표현한 것이다. Fig. 11은 정찰함이 파괴되어 시나리오가 종료된 것을 의미한다. 이와 같이 생성된 시나리오의 로그파일을 입력 받아 가시화하여 사용자에게 직관적으로 시나리오 제공할 수 있다.

3.4 시나리오 생성 프로세스 제안

본 연구에서는 객체에 대하여 모델링 한 결과를 입력값으로 받아 생성 가능한 모든 시나리오를 시나리오 생성기를 이용해 자동으로 생성하고, 생성한 시나리오의 로그파일을 입력 받아 가시화 하는 일련의 과정을 고려한 시나리오 생성 프로세스를

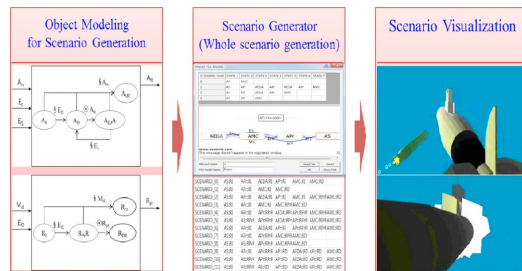


Fig. 12 Process of scenario generation

Fig. 12와 같이 제안한다.

이 시나리오 생성 프로세스는 사용자가 각 객체의 상태를 모델링하고, 제약 테이블을 통해 제약 사항을 정의하면 자동으로 시나리오를 생성하고, 생성된 시나리오가 어떻게 진행되는지 일련의 과정을 가시화하여 미리 예측하지 못한 시나리오의 검증이 가능하다.

또한 일련의 과정을 가시화하여 사용자가 원하는 파악하기 원하는 다양한 시나리오에 대한 직관적인 제공이 가능하다.

4. 결 론

본 논문에서는 교전급 모델에 대하여 효과적으로 구축하기 위해 교전 중 발생할 수 있는 모든 시나리오를 미리 생성하는 방법론을 제안했다. 시나리오 생성을 위해 Timed-FSA 이론을 이용해 객체의 시나리오를 모델링하고, 모델링을 통해 생성한 객체의 시나리오를 조합한 후, 두 객체 사이의 제약사항을 제약 테이블로 입력 받아 의미없는 상태와 이벤트를 제거하였다. 그리고 DFS 알고리즘을 이용해 발생 가능한 시나리오를 생성하는 방안을 제시했다. 또한 예제 시나리오에 대해 제안한 방법에 따라 시나리오 생성기를 구현해 생성 가능한 모든 시나리오를 생성하고, 생성한 시나리오를 로그 파일로 입력 받아 Delta 3D를 이용해 가시화하였다.

본 연구에서 제안한 방안을 이용하면 생성 가능한 모든 시나리오를 생성하여 다양한 상황에 대한 시나리오를 효과적으로 생성할 수 있으며, 제약 테이블을 이용해 의미 없는 시나리오는 생성하지 않을 수 있다. 하지만 현재의 연구 결과에서는 단순한 상황에 대한 시나리오 생성 방안을 제시했다. 따라서 좀 더 복잡한 상황에 대한 시나리오의 생성을 본 연구에서 제시한 상태 다이어그램으로 모두 표현하게 된다면 상태 폭발 현상의 발생이 일어날 수 있다. 이를 위해 고려해 볼 수 있는 연구로는 모든 시나리오의 표현할 때 그 중에서는 사용자가 원하는 시나리오를 좀 더 선별하는 방안이 있다. 예를 들어 시나리오의 생성을 위해 상태를 탐색할 때 특정 상태를 탐색하는 경우만 추출해 준다거나 혹은 특정 상태는 탐색하지 않는 시나리오의 생성하는 방안 있다. 또한 시나리오 생성을 위한 모델링을 할 때 여러 개의 상태를 하나의 상

태로 계층적으로 표현하는 방안이 있다. 이에 대한 구체적인 연구는 향후에 수행할 예정이다.

감사의 글

본 연구는 방위사업청(UD110006MD), 국방과학연구소(UD100009DD), 한국연구재단(2010-0021040)의 지원으로 수행 되었습니다.

참고문헌

1. Sung, K. J., Kim, W. S. and Kang, S.-J., 2007, Synthetic Environment Modeling Methodology for National Defense Modeling and Simulation, *Journal of Computing Science and Engineering* 25(11), pp. 32-38.
2. Park, S. C. and Seong, K. Y., 2010, A Synthetic Environment Based Engagement Simulation Model, *Transactions of the Society of CAD/CAM Engineers*, 15(4), pp. 271-278.
3. Yoon, S., 2004, Analysis of Current Military Modeling/Simulation Technology and Ways to Improve”, *Study of Korean Defense Policy*, Vol. 63, pp. 1-32.
4. Diane Reynolds, 2001, A Framework for Scenario Generation, *ALGO Research Quarterly*, 4(3), pp. 15-36.
5. Daud M. Ahmed, David Sundaram and Selwyn Piramuthu, 2010 Knowledge-based Scenario Management - Process and Support, *Decision Support Systems*, 49, pp. 507-520.
6. Irene Anggreeni and Mascha van der Voort, 2007, Tracing the Scenarios in Scenario-Based Product Design a Study to Support Scenario Generation, *Design Principles and Practices*, 2(4), pp. 123-136.
7. Glenn A. Martin and Charles E. Hughes, 2010, A Scenario Generation Framework for Automating Instructional Support in Scenario-based Training, *10 Proceedings of the 2010 Spring Simulation Multiconference ACM New York, NY, USA*.
8. Lee, J., 2005, A Study Concerning the Technicality of Game Scenario Writing, *The Journal of The Korean Society of Broadcast Engineers*, 10(1).
9. Song, H.-J. and Rhee, D.-W., 2009. A Development of Game Scenario Authoring Tool, *Journal of Korea Game Society*, 9(2).
10. Virtual Prototypes Inc, STAGE Technical Overview, 1999.
11. Mak, V. T., “VR-Forces Users Guide” VRF-4.0-

- 20-110120, 2011. 2.
12. Shin, J.-H., 2007, The Federation Development for Underwater Warfare Simulation, *Korean Simulation Association*, 16(3), pp. 11-18.
 13. Park, S. Y., Shin, H. Y., Lee, T. S. and Choi, B. W., 2010, Design of the Agent-based Network-Centric Warfare Modeling System, *Korean Simulation Association*, 19(4), pp. 271-280.
 14. Zeigler, B. P., 1976, Theory of Modeling and Simulation, Wiley Interscience.
 15. Zeigler, B. P., 1990, Object-Oriented Simulation with Hierarchical, Modular Models, Academic Press.
 16. Tag Gon, 2009, "DEVSIM++ v3.0 User's Manual", KAIST.
 17. Shin, "Game Programming", Daelim, pp.12, 2004. 10.
 18. Perry McDowell, 2006, "Delta3D: A Complete Open Source Game and Simulation Engine for Building Military Training Systems", *JDMS*, 3(3), pp. 143-152.



안 의 국

2008년 아주대학교 산업정보시스템공학부 학사
 2011년 아주대학교 산업공학과 대학원 석사
 2011년~현재 아주대학교 산업공학과 박사과정
 관심분야: Digital Manufacturing System, CAD/CAM, CAPP, Manufacturing System Modeling & Simulation, Defense modeling and simulation, Scenario Generation, Scheduling



고 민 석

2007년 아주대학교 산업정보시스템공학부 학사
 2013년 아주대학교 산업공학과 대학원 박사
 관심분야: Digital Manufacturing System, CAD/CAM, CAPP, Manufacturing System Modeling & Simulation



천 상 욱

2008년 KAIST 기계공학 박사
 2008년~현재 대우조선해양 중앙연구소 정보기술팀
 관심분야: Ship CAD, Freehand sketch-based modeling, Geometric modeling



박 상 철

1994년 한국과학기술원 산업공학과 학사
 1996년 한국과학기술원 산업공학과 석사
 2000년 한국과학기술원 산업공학과 박사
 2004년~현재 아주대학교 산업정보시스템공학부 부교수
 관심분야: Digital Manufacturing System, CAD/CAM, CAPP, Manufacturing System Modeling & Simulation