□ 研 究 論 文 □

韓 國 軍 事 運 營 分 析 學 會 誌
第 23 卷, 第 2 號, 1997. 12 . 31

TPNSim++의 검증 및 군사 시뮬레이션 분야의 활용성
(Comparative Study for the Validation of TPNSim++ and its Applicability to Military Simulation)

최상영*, 김대운**

Abstract

TPNSim++ is the object-oriented framework for the discrete event simulation of military systems, developed by authors. The simulation world view of TPNSim++ is on the basis of activity scanning. TPNSim++ is implemented as C++ class library under Windows 95/NT. It uses the extended timed Petri nets which are called TPNSim nets for simulation modeling tool.

The aim of this study is to conduct a comparative study of TPNSim++ and SLAM II in the simulation of military maintenance systems for the validation and the applicability of TPNSim++. From this study, TPNSim++ and SLAM II have given the same results under the equivalent assumption. Thus we can get the validation of TPNSim++ and its applicability to the simulation of maintenance systems.

* 국방대학원

** 해병대 사령부

1. 서 론

현대전에서 무기체계가 고도화, 정밀화 되고 이를 지원하는 군수의 기능이 중요하게 되었다. 군수의 기능은 크게 정비, 보급, 수송으로 나눌 수 있다. 이중에서 평시 장비의 경제적 관리와 전시 장비의 대량 파괴에 대한 정비 기능은 더욱더 중요하다고 볼 수 있다.

이러한 정비 체계의 효율적인 운용을 위해서는 정비 체계에 대한 과학적이고 체계적인 분석이 필요하고 이를 통하여 전평시 정비 체계에 대한 정책을 합리적으로 수립할 수 있을 것이다.

정비 체계를 분석하는 데에는 여러 가지 모델이 요구되며, 여기에는 수식 모델, 확률 통계 모델, 그리고 시뮬레이션 모델 등이 있다. 특히, 시뮬레이션 모델은 다른 것과 비교하여 대규모 체계를 상세하게 모델링을 할 수 있는 장점을 지니고 있기 때문에 많이 활용되고 있다.

일반적으로 이러한 정비 체계를 모델링하는 데에는 SLAM II를 많이 활용하고 있다[12]. SLAM II을 사용하여 시뮬레이션 모델을 구축할 때, SLAM 네트웍 그래프를 사용하여 모델링하고 이를 SLAM II 시뮬레이션 문장으로 바꾸어 줌으로써 시뮬레이션 모델을 완성하게 된다.

그런데 최근 객체지향 소프트웨어 기술이 발전됨에 따라 시뮬레이션을 위한 여러 가지 클라스(class)를 개발하여 대규모 시뮬레이션 모델을 개발할 때 이를 재 사용함으로써 모델 개발을 용이하게 할 뿐만 아니라 모델의 확장 및 정비 유지를 향상 시키고, 객체지향 특성으로 인한 다른 응용 소프트웨어와 다양한 형태의 인터페이스를 제공받을 수 있게 되었다. TPNSim++가 이러한 것 중의 하나이다.

이는 저자에 의해 개발된 객체지향 시뮬레이션 클라스 라이버리로써 대규모 시뮬레이션을 위한 개발 환경을 제공한다. 그리고 모델링 도구로 시간 Petri nets[10]를 기반으로 하는 그래픽 도구를 사용함으로써 SLAM 네트웍 그래프, GPSS, SIMAN의 블럭 다이어그램 등과 비교할 때 표현의 명확성, 사용의 간편성, 그리고 이해의 용이성이 높다고 할 수 있다[13]. 그런데 TPNSim++는 아직 검증되지 않아 그 활용에 제한이 되어 왔다.

본 연구의 목적은 군사 정비 체계 시뮬레이션 개발시 TPNSim++를 사용하는 경우와 SLAMII언어를 사용한 경우를 상호 비교 분석해 봄으로써 TPNSim++의 검증과 군사 정비 시뮬레이션 개발의 활용성을 보고자 하는데 있다. 이를 위하여 먼저, TPNSim++에 대하여 간단히 소개한다. 그리고 군사 정비 체계에 대하여 TPNSim++ 와 SLAM II로 모델링하여 시뮬레이션한 후에 그 결과들을 통하여 TPNSim++를 비교 검증한다. 그리고 차후 활용성에 대해서 논하고 결론을 맺도록 한다.

2. TPNSim++ 일반

TPNSim++는 이산 사건 시뮬레이션(DEVS, Discrete Event Simulation)을 위한 객체지향 Framework으로써 C++의 클라스 라이버리로 구현되어 있으며 시뮬레이션 개발을 위한 환경을 제공한다[3]. TPNSim++의 시뮬레이션 월드뷰(World View)

는 시간 페트리네트(timed Petri nets)를 기반으로하는 3단계에 의한 활동 주사 방식(Activityt Scanning)이다.

TPNSim++가 기반으로하는 시간 페트리 네트는 플레이스(Place), 트랜지션(Transition), 토큰(Token), 유향에지(Directed Arcs)로 구성되어 있다. 플레이스는 트랜지션과 유향에지로 연결되어 있으며, 플레이스가 트랜지션 방향으로 유향에지가 연결되어 있을 때 입력 플레이스라고 하고 반대일 경우는 출력 플레이스라고 한다.

토큰은 플레이스 내에 존재할 수 있다. 트랜지션은 자신의 모든 입력 플레이스에 토큰이 한 개이상이 존재할 때 활성화되었다고 하며 활성화 된 이후 임의 시간후에 점화되어 토큰을 자신의 출력 플레이스로 이동시킨다.

그러면 플레이스에 존재하는 토큰의 수가 변화하게 된다. 플레이스내의 토큰 분포를 마킹이라고 하는데 이 마킹은 트랜지션의 점화에 따라 변화하게 된다. 이러한 시간 페트리 네트는 이산 사건 시뮬레이션 구도와 잘 대응된다[7,8,9]. 즉, 트랜지션의 점화는 사건이 되고 점화 시간은 사건 시간이 되며, 입력 플레이스는 사전 조건이 되고 출력 플레이스는 사후조건이 된다. 또한 토콘은 사건발생을 통제하는 제어기가 되며 마킹은 체계의 상태가 된다.

TPNSim++는 이러한 구도에서 시뮬레이션 전개 방식은 3단계로 이루어진다. 첫단계는 활성화된 트랜지션중에서 점화 예정 시간이 가장빠른 것을 찾아낸다. 뜨 번째 단계는 점화로 인한 토콘의 이동을 집행한다. 이와 같은 3단계를 순차적으로 지속시킴으로써 시뮬레이션을 진행시키게 된다.

TPNSim++을 사용하여 시뮬레이션 응용 프로그램을 개발하고자 할 때는 개발자가 먼저 시뮬레이션 대상 시스템을 분석한 후에, 이를 TPNSim nets라고 불리는 시간 Petri nets를 사용하여 가시적 그래프로 모델링한다. 그 후 모델링된 TPNSim nets 그래프를 TPNSim++의 문장으로 전환 시켜줌으로써 시뮬레이션 코드를 완성한다. 그리고 완성된 코드를 C++컴파일러를 사용하여 컴파일 한 후에 시뮬레이션 실행 화일을 얻게 된다.

TPNSim++에서는 그래픽 분석 도구를 사용하여 모델링하고 이를 TPNSim++ 문장으로 전환 시켜준다는 면에서는 SLAM II와 유사하나 그래픽 도구를 시간 페트리네트를 사용한다는 것이 다르고, 또한 C++로 컴파일 해야 한다는 것이 차이점이다.

2.1 TPNSim nets

TPNSim net는 TPNSim++로 시뮬레이션을 개발하기 위한 그래픽 분석 도구이다. 이는 시간 페트리 네트에 기반을 두고 있으며 그 형식론(formalism)은 다음과 같이 정의된다.

$$M = \langle C, , , \rangle$$

여기서,

$C = \langle P,T,I,O \rangle$ 이고

$P=\{p_1,p_2,p_3,...p_n\}$는 플레이스의 유한 집합, n0.

$T=\{t_1,t_2,t_3,...t_m\}$는 트랜지션의 유한 집합, m0.

I : TP 입력 함수, 트랜지션으로부터 플레이스들의 Bag으로의 사상.

O : TP 출력 함수, 트랜지션으로부터 플레이스들의 Bag으로의 사상.

그리고

:PN, P로부터 양의 정수 N으로 사상하는 함수.

:$T(E,D)^{card(T)}$, T로부터 $(E,D)^{card(T)}$의 공간으로 사상하는 함수로써 E(Enable), D(Disable)는 각각 트랜지션의 활성화 상태 및 비활성화상태.

:TE, T로부터 E 공간으로 사상하는 함수. E={e_i|e_i는 트랜지션 i의 점화 예정 시각(epoch)}.

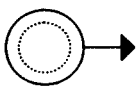
TPNSim nets는 체계를 보다 융통성 있게 모델링 할 수 있도록 플레이스와 트랜지션의 종류를 여러 가지로 구분하여 정의한다.

2.1.1 플레이스의 종류 및 표시 방법

플래이스의 종류는 다음과 같이 6가지의 종류 즉, 생성 플레이스(U형 플레이스), 제한 레이스(L형플레이스),상태플레이스(P형플레이스),자원플레이스(R형플레이스), 축척 플레이스(O형 플레이스), 메크로 플레이스 (D형 플레이스)로 구분하며 각 종류별 구체적인 정의 및 표시 방법은 다음과 같다.

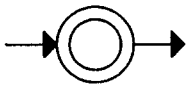
생성 플레이스(U형 플레이스)

- 정의 : 토큰을 생성하고, 생성 시간을 관리하는 플레이스
- 표시 방법 :



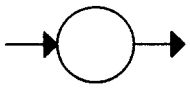
제한 플레이스(L형 플레이스)

- 정의 : 유입되는 토큰의 숫자가 제한(bounded) 되고 토큰의 유입 시간을 관리할 수 있는 플레이스
- 표시 방법 :



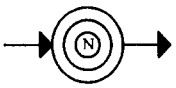
상태 플레이스(P형 플레이스)

- 정의 : 토큰에 대한 숫자만 관리하고 시간을 관리하지 않는 플레이스
- 표시 방법 :



자원 플레이스(R형 플레이스)

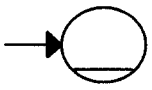
- 정의 : 시스템의 자원을 나타내는 토큰을 관리하는 플레이스
- 표시 방법 :



여기서, N은 자원의 수.

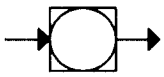
축적 플레이스(O형 플레이스)

- 정의 : 더이상 이동하지 않고 저장되는 토큰을 관리하는 플레이스
- 표시 방법 :



메크로 플레이스(M형 플레이스)

- 정의 : 모델링 세분화 정도에 따라 여러 개의 플레이스와 트랜지션을 하나의 기호로 표시한 플레이스
- 표시 방법 :

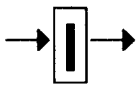


2.1.2. 트랜지션 종류 및 표시 방법

트랜지션의 종류는 생성 트랜지션(U형 트랜지션), 제한 트랜지션(L형 트랜지션), 상태 트랜지션(P형 트랜지션), 조건 트랜지션 4가지로 구분하고 다음과 같이 정의하고 표시한다.

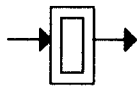
생성 트랜지션(U형 트랜지션)

- 정의 : 오직 1개의 생성 플레이스(U형)를 입력 플레이스로 갖는 트랜지션
- 표시 방법 :



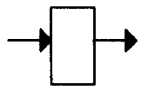
제한 트랜지션(L형 트랜지션)

- 정의 : 제한 플레이스(L형)를 적어도 1개 이상을 입력 플레이스로 갖는 트랜지션
- 표시 방법 :



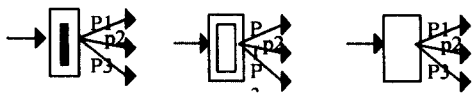
상태 트랜지션(P형 트랜지션)

- 정의 : 자신의 입력 플레이스의 형태가 P형이나 R형인 트랜지션
- 표시 방법 :



조건 트랜지션

- 정의 : 점화(firing)시 출력 플레이스에 1개의 토큰을 확률적으로 전이시키는 트랜지션
- 표시 방법 :



표시방법	확률 변수
U(a,b)	uniform 분포 확률변수 발생함수
E(lambda)	exponential 분포 확률변수 발생함수
ER(alpha, beta)	erlang 분포 확률변수 발생함수
N( )	N(0,1) 분포 확률변수 발생함수
N(nu,sigma)	N(nu,sigma) 분포 확률변수 발생함수
A(a,b,c)	삼각분포 확률변수 발생함수
T(nu)	자유도가 nu인 t 분포 확률변수 발생함수
G(alpha)	gamma 분포 확률변수 발생함수
B(alpha,beta)	beta 분포 확률변수 발생함수
C(a)	전이 시간이 a로일정한 경우

<표 1> 트랜지션의 점화시간 표시방법

2.1.3 토큰 표시 방법

토큰은 플레이스 내에 • 로 표시하는 것을 원칙으로 하나 숫자가 클 경우는 정수로도 표시한다.

2.1.4 토큰 발생 시간 및 트랜지션의 점화 시간 표시 방법

U형 플레이스나 L형 플레이스 경우 토큰의 발생 시간을 가지고, 모든 트랜지션은 점화 시간을 가진다. 이들 시간을 확률 변수로 나타내며 확률 변수의 종류와 매개변수를 플레이스 표시 안에 혹은 트랜지션 표시 안에 <표 1> 같이 표시할 수 있다.

2.2 TPNSim++ 문장

TPNSim++ 문장은 객체의 메세지 형태로 되어 있다. 문장은 아래와 같이 4가지로 구분된다.

. TPNSim nets 시스템 정의를 위한 문장
. TPNSim nets 시스템 시뮬레이션 초기화를 위한 문장
. TPNSim nets 시뮬레이션 실행을 위한 문장
. 기타 문장

TPNSim nets 시스템 정의를 위한 문장에서는 시뮬레이션 대상 시스템의 TPNSim nets에서의 플레이스의 갯수, 트랜지션의 갯수, 시뮬레이션 기간을 정의한다.
그리고 각 플레이스의 종류, 토큰의 갯수, 토큰의 발생 분포를 설정하며, 각 트랜지션의 입력 및 출력 플레이스, 점화시간 분포도 설정한다. TPNSim nets 시스템 시뮬레이션 초기화를 위한 문장에서는 시뮬레이션 초기화와 토큰 발생 시간을 초기화 한다. 그리고 시뮬레이션 실행 문장을 사용하여 시뮬레이션 실행문을 완성한다. 한편, 기타 문장에서는 통계 처리 및 결과 분석을 위해서 사용한다. 그리고 각 문장은 아래와 같다.

2.2.1 TPNSim nets 시스템 정의를 위한 주요 문장

```
aTPNSim.setEvent( a,b,c,d,e);
aTPNSim.aPlace[i].setPlace(a1,a2,a3);
aTPNSim.aPlace[i].setPlace(b1,b2,b3,b4);
aPNSim.aPlace[i].setPlace(c1,c2,c3,c4,c5);
aTPNSim.ITokenNum[i];
aTPNSim.aToken[i][j].setPDF(...);
aTransition[i].setDtransition(a,b,c,d);
aTransition[i].setInputPlaceID(a1,a2,a3,.....);
aTransition[i].setOutputPlaceID(b1,b2,b3,...);
aTransition[i].setOutputPro(p1,p2,p3,...);
aTransition[i]. setFiringPDF(...);
```

2.2.2 TPNSim nets 시스템 시뮬레이션 초기화를 위한 주요 문장

```
aTPNSim.aToken[i][j].getTimeToOccur( )
aTPNSim.aToken[i][j].timeToOccur
aTransition[i].update(aTPNSim);
```

2.2.3 TPNSim nets 시뮬레이션 실행을 위한 주요 문장

```
aTransition[i].schedule(aTPNSim);
aTransition[anEvent.ET].fire(aTPNSim);
aTransition[i].update(aTPNSim);
```

2.2.4 기타 문장

```
aTPNSim.IPlaceNum;
aTPNSim.cardOfTransition;
```

각 문장에 대한 세부 설명은 참고문헌[5]에서 상세히 취급하고 있다.

3.군 정비 시스템의TPNSim++와 SLAM II 모델링

3.1 모델링 대상 시스템

연구 목적을 위해 K-장비에 대한 정비 시스템(이하 K-정비 시스템이라고 함)을 전형적인 군 정비 시스템으로 설정하였다. K-정비 시스템에서는 overhaul 정비 뿐만 아니라 고장난 구성품에 대한 창 정비(이후 고장 정비 라고 함)를 병행하는데, overhaul 정비에서는 K-장비에 대한 엔진 정비, 차체 정비, 밋션 정비, 해수 추진기 정비, 전기 장치 정비가 이루어 지고, 고장 정비에서는 K-장비의 엔진, 포탑, 밋션, 해수 추진기, 전기 장치에 대한 창 정비가 이루어 진다.

그리고 K-정비 시스템 내에서는 6개의 정비반으로 구성되고 이들은 overal 정비 뿐만 아니라 고장 정비도 실시 하며 각 정비반의 기능은 <표 2>와 같다

구분	기능(Overhaul)	기능(고장정비)
본부반	입고장비 점검/ 출고장비 검사	
지원반	장비분해/조립, 해상추진기 정비/검사	해상추진기 분해/정비/조립/검사
기관반	엔진 분해/정비/조립/검사	엔진 분해/정비/조립/검사
동력반	밋션 분해/정비/조립	밋션 분해/정비/조립/검사
전기반	전자(기)장비 분해/정비/조립	전자(기)장비 분해/정비/조립/검사
차체반	차체 제청/도색	포탑 분해/정비/조립/검사

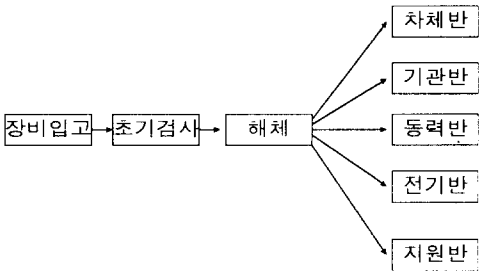
<표 2>K -정비 시스템의 정비반 기능

3.2 정비 시스템의 TPNSim nets 모델링

3.1에서 설명한 K-장비에 대하여 overhaul 정비 공정과 고장 정비 공정을 구분해서 TPNSim nets로 모델링하면 다음과 같다.

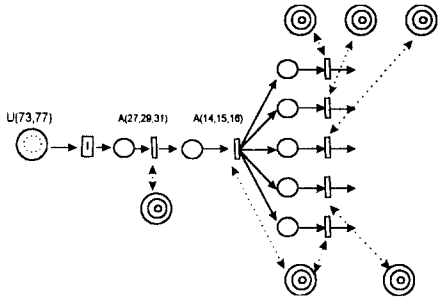
3.2.1 overhaul 정비 공정에 대한 모델링

overhaul 정비에서는 <그림 1>과 같이 최초 장비가 입고되면 본부반으로부터 검사를 받은 후에 지원반이 장비를 해체하여 차체반, 기관반, 동력반, 전기반, 지원반으로 이송되어 정비를 실시하게 된다.



<그림 1>overhaul 정비 공정도

위 과정을 TPNSim nets로 나타내면 다음과 같다.

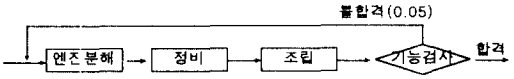


<그림2>overhaul 정비 공정에 대한 TPNSim net 모델링 그래프

여기서 도착하는 장비는 생성 플레이스로 표현하였고, 도착된 장비는 본부반에 의해서 입고 장비 점검을 실시하는데 점검 활동은 트랜지션으로 표시되면서 점검시간은 예를 들어 삼각분포(A(27,29,31))로 모델링 되었다. 그리고 점검이 완료된 장비를 지원반에 의해 분해되는데 지원반은 자원 플레이스로 표시되고 분해 활동은 트랜지션으로 표시되며 분해시간은 예를 들어 삼각분포(A(14,14,16))로 모델링 되었다. 그리고 분해된 장비는 각 정비반별 분기되는데

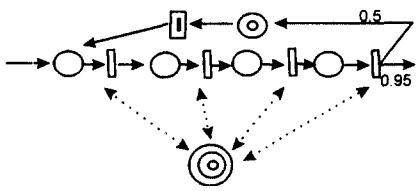
이는 조건 트랜지션을 이용하여 모델링 되고, 여기서는 정비반으로 이송되는 확률이 100%가 된 경우이다.

그러면 각 정비반에서 해당 정비 대상물에 대하여 분해 조립 혹은 검사를 하게 된다. 예를 들면 기관반에서 이루어 지는 정비 공정은 아래 그림과 같다. 여기서 정비후 기능 검사를 실시하여 불합격된 경우는 재 정비를 실시하는데 아래 그림의 경우 불합격이 될 확률은 5%임을 의미한다.



<그림 3> 구성품 각 정비반별 정비 공정도

위의 정비 과정을 TPNSim nets로 나타내면 다음과 같다.

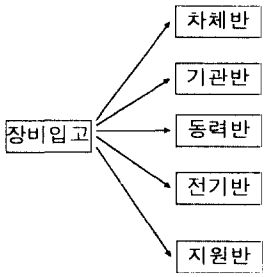


<그림4> 구성품 각 정비반별 정비 공정 TPN-Sim nets 모델링 그래프

정비반별 정비시 검사후 feedback되는 장비는 L형 플레이스를 통과함으로서 새로운 시뮬레이션 시간을 지니게 되고, 다른 정비반에서 이루어 지는 정비과정도 위와 같이 모델링할 수 있다. 그리고 모든 정비반에서 정비가 완료되면 지원반에 의하여 재조립되어 최종 검사후 출고 되고 최종 검사시 불량이 발생하면 재정비가 이루어 지도록 모델링 된다.

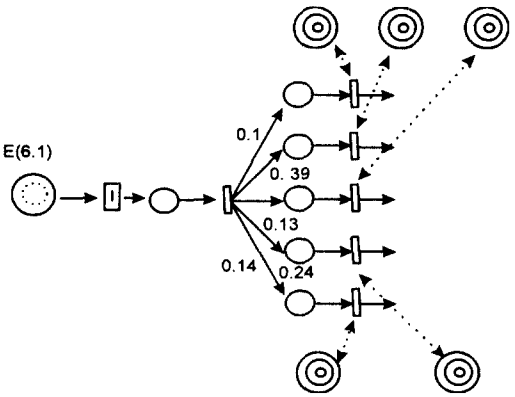
3.2.2 고장 정비에 대한 모델링

고장 정비에서는 창 정비 수준에서 정비가 이루어 지기 때문에 장비가 도착하는 즉시 고장종류에 따라 아래 그림에서와 같이 해당 정비반으로 이송된다.

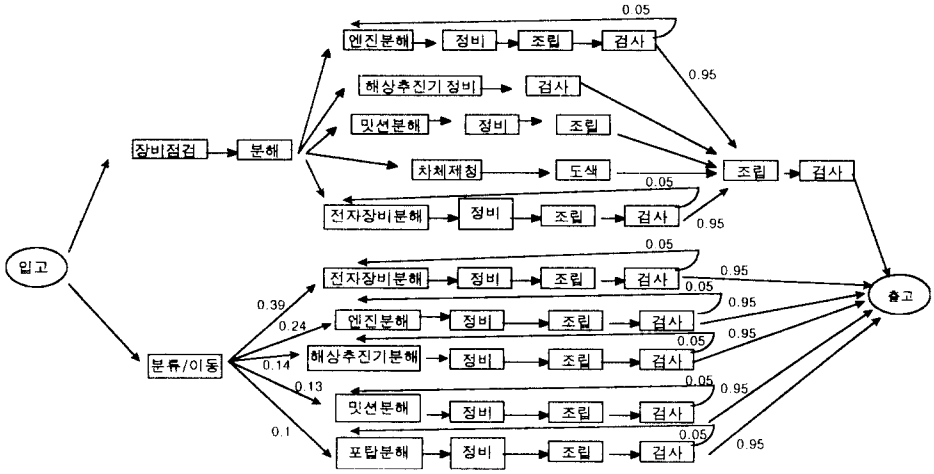


<그림 5> 고장 정비 공정도

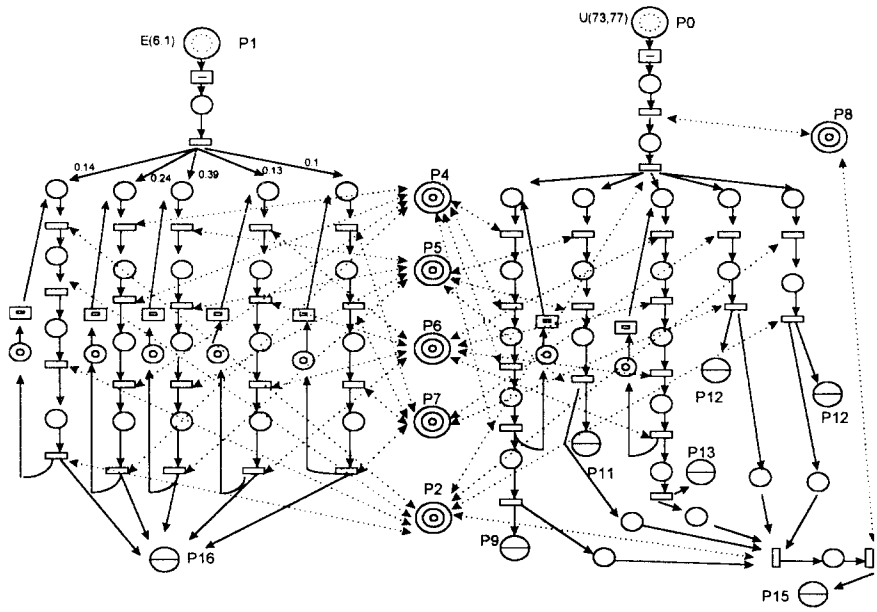
이를 TPNSim nets로 표시하면 다음과 같다.



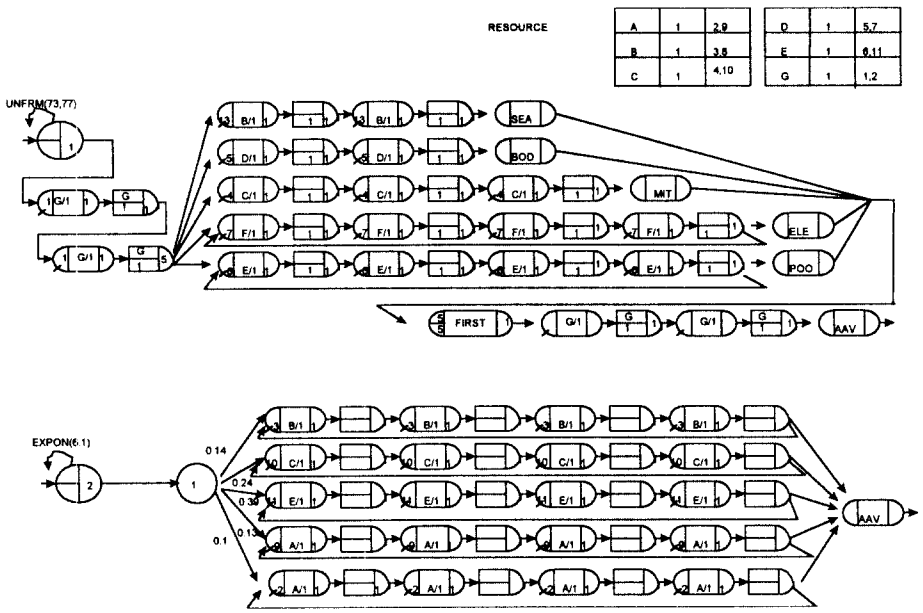
<그림 6> 고장 정비 공정 TPNSim nets 모델링 그래프



<그림 7> K-정비 체계 전체 공정도



<그림 8> K-정비 체계 전체 공정 TPNSim nets 모델링 그래프



<그림 9 > K-정비체계 전체 공정에 대한 SLAM 네트워크

여기서 고장 종류는 확률적으로 이루어 지기 때문에 조건 트랜지션을 이용하여 모델링이 된다.

3.2.3 전체 공정에 대한 TPNSim nets

이상에서 분석한 정비 체계의 전 공정은 <그림 7>과 같으며 이를 TPNSim nets로 나타내면 아래 <그림8>과 같다.

3.3 정비 시스템의 SLAM 네트워크 모델링

3.2 에서 TPNSim nets로 모델링 된 것을 SLAM II의 네트웍 그래프로 모델링 할 수 있는데, 이는 <그림 9>와 같다.

4. 시뮬레이션 결과 비교 분석

앞에서 모델링한 정비체계를 TPNSim++ 문장을 사용하여 프로그램을 구성하고, 한편으로는 SLAM II 문장을 사용하여 프로그램을 구성한 다음 그 결과를 비교하였다. 이를 위해서 다음을 가정하였다.

4.1 가정 사항

. Overhaul 정비를 위해 도착하는 장비의 도착

정비유형	정비 항목	세부항목	작업시간(일양분표)	
			최소치	최대치
Overhaul 정비	최초검사	검사	27	31
	장비분해	분해	14	16
	엔진 정비	분해	17	19
		정비	24	26
		조립	17	19
		시험	18	20
	차체 정비	분해/제청	45	49
		조립/도색	39	44
	밋션 정비	분해	22	24
		정비	31	34
		조립	23	25
	해수추진기 정비	분해/정비	23	25
		조립/도색	23	25
	전기 장비 정비	분해	11	15
		정비	23	27
		조립	11	13
		시험	12	16
	장비 조립	조립	8	3
	최종시험	시험	26	30
구성품 고장 정비	엔진 정비	분해	5	7
		정비	8	12
		조립	5	7
		시험	4	6
	포탑 정비	분해	2	4
		정비	5	7
		조립	2	4
		시험	4	6
	밋션 정비	분해	2	4
		정비	5	7
		조립	2	4
		시험	2	4
	해수추진기정비	분해	2	4
		정비	4	6
		조립	2	4
		시험	1	3
	전기 장비 정비	분해	1	3
		정비	3	5
		조립	3	5
		시험	1	3

<표 3 > 구성품 정비 소요시간

구분	정비수량 평균/분산		자원 가동율 평균					
	AAV	구성품	기관반	지원반	동력반	차체반	전기반	본부반
TPN Sim ++	14.64 / 1.02	204.6 / 11.3	83.8	87.56	80.84	86.24	83.04	71.04
SLA MII	14.84 / 0.74	207.84 / 14.01	84.12	88.04	81.12	85.64	84.28	71.76

<표 4> 시뮬레이션 결과

시간은 최소치 73, 최대치가 77인 일양분포를 따른다.

. 고장 정비를 위해 도착하는 장비의 도착시간은 평균이 6.1인 지수분포를 따른다. 그리고 고장 정비중에서 전자 장비의 고장일 확률은 39%, 엔진 고장일 확률은 24%, 해상 추진기 고장일 확률은 14%, 밋션 고장일 확률은 13%, 그리고 포탑 고장일 확률은 10%이다.

. 정비반은 각각 1개 반으로 구성하고 각 정비반별 overhaul정비와 고장 정비에 따른 각 작업 소요 시간은 <표3>과 같다.

4.2 시뮬레이션 결과 분석

시뮬레이션은 1년간 단위로 25회 실시하였으며 그 결과는 <표 4>와 같다.

이들에 대한 평균 검증과 분산 검증을 실시한 결과 5% 유의 수준에서 결과가 서로 같다고 할 수 있었다.

5. 결 론

본 연구에서는 군사 정비 체계 시뮬레이션 개발시에 TPNSim++와 SLAM II로 모델링하고 그 상호 결과를 비교 분석 해 봄으로써 TPNSim++의 검증과 활용성을 보았다. 이 결과 동일한 상황에서 TPNSim++와 SLAM II는 같은 결과를 주었다. 이를 통해서 볼 때 TPNSim++는 차후 군 정비 체계를 포함한 군사 시뮬레이션 개발시 유용하게 활용될 수 있을 것이다.

특히, TPNSim++에서 사용하는 TPNSim nets라는 그래픽 분석 도구는 시간 Petri nets에 기반을 두고 있기 때문에 Petri nets가 가지고 있는 표현의 간결성, 이해의 용이성이 높고, 시뮬레이션 대상 체계를 TPNSim nets로 표현하기만 하면 이를 TPNSim++ 문장으로 전환하여 시뮬레이션 프로그램을 작성할 수 있기 때문에 짧은 시간내에 TPNSim++의 사용 방법을

습득할 수 있을 것이다.

추가적으로 본 연구에서 TPNSim++는 한국과학재단의 지원으로 개발된 것임을 밝혀둔다.

참 고 문 헌

1. 이강림, 객체지향 TPN 그래픽 시뮬레이션과 군사적 응용 ,국방대학원 석사 논문,1995.
2. 이종호, TPN 시스템 시뮬레이션을 위한 기본클라스 구현,국방대학원 석사, 1994
3. 최상영, 객체지향 시뮬레이션에 관한 연구, 국방대학원 연구보고서, 1994.
4. 하금대, 페트리네트 이론을 이용한 객체지향형 정비모형 설계에 관한 연구, 국방대학원 석사 논문, 1992.
5. Al-Sammak A. J., A comparative study between Petri Net and SLAM, SIMULATION,1992.11.
6. Bong-Wan Choi, Department of Industrial and Manufacturing Systems Engineering, Iowa State University.
7. James T. Lim, Chia-Chu Lee, A Three-Phase Discrete Event Simulation with EPNSim Graphs, Simulation 60:9, June 1993, pp.382-392.
8. James W.Hooper, Strategy-related Characteristics of Distrete-Event Languages and Models, Simulation 46:4, April 1986, pp.153-159.
9. Robert M. OKeefe, The Three-phase Approach: A Comment on Strategy-elated Characteristics of Discrete-event Language and Models, Simulation 47:5, November 1986, pp. 208-211.
10. Peterson,James L., Petri net theory and the modeling of systems, Prentice-Hall,1981.
11. Peter J.HAAD and Gerald S.Shedler, Stochastic Petri Net Representation of Discrete Event Simulations, IEEE Transactions on software Engineering. VOL. 15, NO. 4, APRIL 1989.
12. Prisker,A. Alan B., Introduction to Simulation and SLAM-II,Systems Publishing Co.1986.
13. Richard Zurawski and MengChu Zhou, Petri Nets and Industrial Applications, IEEE Transactions on industrial electronics, VOL.41,NO.6,DECEMBER 1994.