

TFM 기반 무기체계 소프트웨어의 시스템 테스트 모델 설계

김재환⁰ 윤희병

국방대학교 전산정보학과

kimch092@yahoo.co.kr⁰, hbyoon@kndu.ac.kr

Design of the TFM Based System Test Model for embedded Software of Weapon Systems

Jaehwan Kim⁰ Heebyung Yoon

Dept of Computer & Information Science, Korea National Defense University

요 약

본 연구에서는 시간 요소를 고려한 무기체계 소프트웨어의 시스템 테스트 모델을 제시한다. 이를 위해 컴포넌트 기반의 UML 표기법과 무기체계의 시간 제약 특징을 고려한 무기체계 소프트웨어의 시스템 테스트 모델인 TFM(Time Factor Method) 모델을 제안한다. TFM 모델은 세 가지 요소인 (X,Y,Z)로 구성되어 있는데, "X"에서는 최대시간경로를 선정하는 알고리즘을 통해 테스트 케이스가 도출되고, "Y"에서는 Sequence Diagram과 관련된 컴포넌트(오브젝트)를 식별하고, "Z"에서는 Timing Diagram을 통하여 식별된 각각의 컴포넌트의 실행시간을 측정한다. 또한 제안된 TFM 모델의 적용사례를 들어 테스트 케이스를 도출한다.

1. 서 론

시스템 테스트는 기본 요구사항 설계서를 기초로 하여 설계서에 표현된 기능이 제대로 실현되는가를 검증한다. 또한 시스템 테스트는 기본 요구사항 단계의 개발 초기에 설정된 정보를 사용하여 테스트 데이터를 생성한다. 이러한 시스템 테스트의 목적은 결함을 발견하기 보다는 성능관점에서 검사를 실시한다 [1]. 소프트웨어의 신뢰성을 보장하기 위한 테스트 기법으로는 Equivalence partitioning 테스트 기법이나 Boundary 테스트 기법 혹은 기능을 그룹별로 구분하여 테스트를 하는 등 많은 연구가 진행 되었다[2][3].

무기체계 소프트웨어 테스트를 위해서는 일반 소프트웨어와는 달리 기능이나 로직(Logic)은 물론 시간제약 조건을 반드시 준수하여야 한다. 그러므로 무기체계 소프트웨어의 특징을 고려한 테스트 기법이 요구되고 있다. 또한 무기체계 소프트웨어는 H/W 중심에서 S/W 중심으로 기술이 발전하고 있으며 최신에 기종인 F-22는 80%를 소프트웨어로 처리 하는 등 소프트웨어의 복잡도가 증가하고 개발비용 및 개발시간이 급격한 증가를 보이고 있다[4]. 이러한 문제를 해결하기 위해 재사용성을 위한 컴포넌트 기반의 개발 추세에 있으며, "UML 2.0"에서는 컴포넌트의 시간 관계를 표현하는 Timing Diagram등 다양한 표기법을 제안하였다[5].

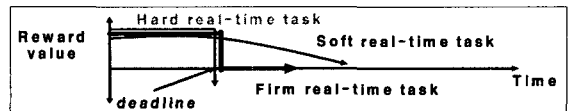
이에 본 논문에서는 컴포넌트 기반의 UML 표기법과 무기체계의 시간제약 특징을 고려한 무기체계 소프트웨어의 시스템 테스트 모델인 TFM(Time Factor Method) 모델을 (X,Y,Z)의 세 가지 요소를 이용하여 제안한다. 그리고 TFM 모델을 "다 기능 무기체계 시스템"에 적용한 사례연구를 통하여 시스템 테스트 케이스를 추출하는 방법을 제시하였다.

2. 관련 연구

2.1 항공 무기체계 소프트웨어의 특징

항공 무기체계 소프트웨어는 Critical System으로 분류되고,

대부분 임베디드 소프트웨어로 구성되어 있으며 시간제약 조건이 준수되어야 한다[6]. 임베디드 소프트웨어의 특징 중 하나인 시간제약 조건은 무기체계 소프트웨어에서 중요한 의미를 가지며, 종료시간(Deadline)의 준수 여부 또한 매우 중요한 의미를 갖는다. [그림 1]은 무기체계 소프트웨어의 시간제약 조건을 나타내고 있다.



[그림 1] 무기체계 소프트웨어의 시간제약 조건

[그림 1]의 시간제약 조건 중에서 무기체계 소프트웨어에 가장 적합한 것이 경성 종료시간(Hard Real-time task)으로 실시간 긴급을 요하고, 실패 시 막대한 인명이나 재산의 피해를 초래할 수 있기 때문에 소프트웨어 개발 시 중요한 요건으로 간주된다. 그러므로 무기체계 소프트웨어 테스트 분야에서는 시간요소가 중요한 정경 항목으로 간주된다.

2.2 소프트웨어 시스템 테스트

시스템 테스트에는 ASF(Atomic System Function) 개념이 있는데, ASF는 입력 이벤트에서 시작해서 출력 이벤트까지의 특정 단위를 의미하며, 클래스 테스트 케이스 추출기법, 통합 테스트 케이스 추출 기법과 시스템 테스트 케이스 추출 기법에서 널리 이용되고 있다[1]. 시스템 테스트에서 ASF는 발생할 수 있는 하나의 기본 단위이며, 하나의 동작이 시작하여 마무리 되는 하나의 Action의 결과라 할 수 있다.

3. 소프트웨어 시스템 테스트 모델 설계

3.1 시간 요소를 고려한 테스트 방법 (TFM)

하나의 소프트웨어는 각각 다른 컴포넌트와 컴포넌트의 결합

이라고 할 수 있으며, 시스템 테스트는 이들 컴포넌트들 간의 결합이 제대로 연결되는가를 검증하는 절차이다. UML에서는 이러한 연결이 컴포넌트 간 Message의 흐름을 통해서 이루어진다[7]. 또한 각각의 컴포넌트는 인터페이스를 통해 고유한 성능을 나타내며 “0”이 아닌 고유의 실행시간(Execution Time)을 갖고 있다. 이러한 실행시간은 프로그램에 대한 CPU Clock Cycle과 Clock Cycle Time의 곱이 되며 CPU Clock Cycle은 각각의 클래스에 대한 Instruction 수의 합이 된다[8]. 즉, 하나의 ASF는 시스템 측면에서 보면 고유한 특성을 갖는 여러 컴포넌트들의 집합이라 할 수 있으며, ASF의 실행시간은 모든 관계된 컴포넌트의 시간 합이 된다.

컴포넌트의 실행시간은 처음부터 일정한 시간으로 정해지는 것도 있지만 입력을 기다리는 경우도 있고, 또한 데이터 처리에 따라 시간이 유동적일 경우도 있다. 그러한 경우에는 Stress Test 방법에 따라 최대시간을 기준으로 측정된다[9]. 그것은 최악의 경우에도 제한시간을 준수해야 하는 무기체계 시스템의 특징을 만족시키기 위함인데, 같은 경로를 진행한다고 해도 처리시간이 유동적일 경우는 최대한 오래 진행될 경우의 수를 측정해야 모든 상황을 포함할 수 있기 때문이다. 이러한 시간측정은 실제 작동될 프로세서에서 실행되어야 하는데 그것은 작동할 프로세서의 처리속도에 따라 처리시간(능력)이 달라질 수 있기 때문이다.

3.2 테스트 케이스 선정 알고리즘

무기체계 시스템은 성능뿐만 아니라 제한된 시간에 성능을 발휘해야 그 의미가 있다. 아무리 좋은 성능을 가지고 있다 하여도 시간제약 사항은 반드시 지켜져야 하는 것이다. 최대시간 측정 알고리즘을 위하여 최장경로문제 알고리즘[10]의 기본 프로세스를 활용한 최대시간 측정 알고리즘이 [그림 2]에 제시하였다.

- ① 모든 $C_{ij} \geq 0$ 이면 모든 $V_i = 0$ ($i = 1, \dots, n$)로 둔다.
- ② 부모노드에서 시작하여 모든 호 (i, j) 를 점검하여 $V_j - V_i < C_{ij}$ 이면 $V_j = V_i + C_{ij}$ 로 둔다. 이러한 절차를 반복하여 모든 호 (i, j) 가 $V_j - V_i \geq C_{ij}$ 가 되도록 한다. 양의 값으로 순환(cycle)현상이 발생하면 해는 무한대이며, 그렇지 않으면 다음 단계로 간다.
- ③ $V_j - V_i = C_{ij}$ 가 되는 호를 연결하여 전체 트리경로를 만든다. Source node 1에서 sink node n까지 연결되는 경로를 발견하면 이 경로가 최장 경로이다. 그리고 최장경로의 시간 값은 $V_n - V_1$ 이다.
- ④ 잔여 경로에서 차 순위 경로를 선택한다.
- ⑤ 잔여 경로가 없으면 검색을 완료한다.

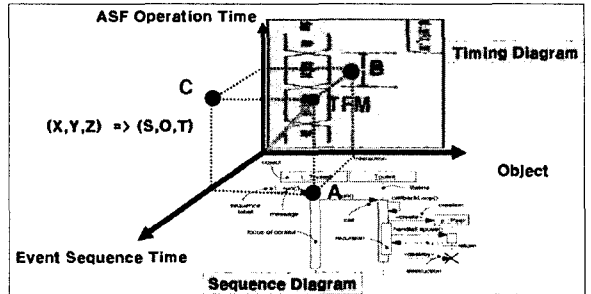
[그림 2] 최대시간 측정 알고리즘

위의 알고리즘을 통하여 최장시간 경로를 도출해 낼 수 있으며 모든 경우의 수를 위해서 잔여 경로에 대한 경로 검색을 하면 그 경로 수만큼의 테스트 케이스가 식별된다.

3.3 TFM(Time Factor Method) 모델 설계

테스트 케이스 선정 알고리즘을 UML에 적용하고 Diagram 간의 관계를 파악하기 위해 TFM(Time Factor Method) 모델을

설계하면 다음과 같다. 우선 TFM은 세 가지의 요소를 가지고 있다. 즉, (X,Y,Z)으로 구성되어 있는데, “X”에서는 최대시간경로를 선정하여 테스트 케이스가 도출되고, “Y”에서는 Sequence Diagram에서 관련된 컴포넌트(오브젝트)를 식별하고, “Z”에서는 식별된 각각의 컴포넌트의 실행시간을 측정한다. 이와 같은 TFM 모델이 [그림 3]과 같이 도시되어 있다.

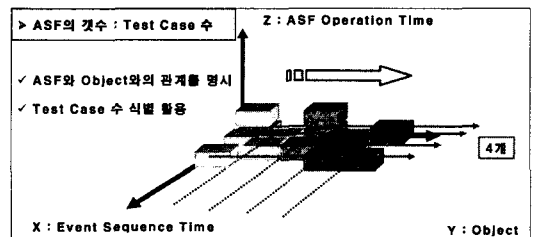


[그림 3] TFM 모델 설계

TFM 모델 설계를 위해 시스템 테스트 관점 접근과 무기체계 소프트웨어의 특징, 그리고 UML Diagram 접근방법을 이용한다. 시스템 테스트 측면에서는 시간 요소를 고려한 ASF를 활용하며, UML에서는 Interaction에 관계된 Sequence Diagram과 Timing Diagram을 활용한다.

[그림 3]을 보면 Sequence Diagram과 Timing Diagram의 객체(Object)를 공통분모로 하여 배치하며 X축과 Y축의 평면에는 Sequence Diagram을 배치하여 X축에는 Event Sequence time을 통한 메시지 흐름을 파악하고, Y축에는 Sequence Diagram의 관계된 Object를 배치한다. Y축과 Z축에는 Timing Diagram을 배치하여 Y축에는 Sequence Diagram의 공통분모를 Timing Diagram의 Object와 일치시키고, Z축에는 ASF의 Operation Time(Object 실행시간)을 배치한다.

이 모델의 공통은 Y축의 Object로서 UML에서 가장 기본적인 중요한 요소를 활용하였다. 이 모델은 재사용과 중복 사용되는 각각의 컴포넌트를 UML로 표기할 때 [그림 4]와 같이 객체간의 관계를 파악할 수 있다.



[그림 4] TFM 모델에서 테스트 케이스 식별

[그림 4]를 살펴보면 Object가 A부터 E까지 종류별로 사용되어있다. A Object는 4개로 구성 되어있고, B Object는 2개, C Object는 3개, D Object는 2개, E Object는 2개로 각각 구성되어있다. Object가 Z축으로 층을 쌓으면서 올라가 있는 경

우가 있는데 그 경우는 똑 같은 Object가 재사용되어서 구성되어 있음을 알 수 있다. 따라서 이 소프트웨어에서 사용된 총 Object는 13개임을 알 수 있다. 예시된 소프트웨어는 총 4개의 ASF로 구성되어 있고, 각각의 ASF와 Object와의 관계와 테스트 케이스를 파악할 수 있다.

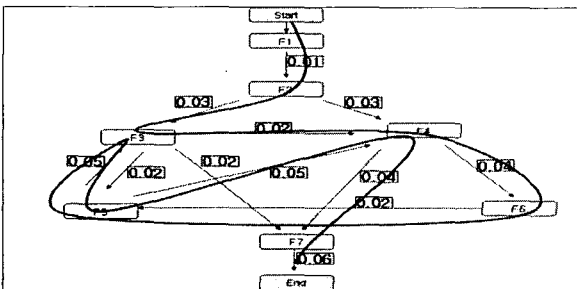
4. 사례 연구

TFM 모델의 사례연구로 “Multi-Function System Testing: Composition of Test Sets”[3]에서 사용한 시스템을 이용하였다. 컴포넌트 실행시간은 “0”으로 존재하는 경우는 없으므로 [표 1]과 같이 가상하여 부여하였다.

[표 1] 미사일 방어시스템의 컴포넌트 실행시간

컴포넌트	설 명	시간(s)
F1	식별 지역에 대한 레이더 파 주사	0.01
F2	목표물에 대한 추적	0.03
F3	목표물 식별(Identification)	0.02
F4	적/아 식별	0.04
F5	다량 목표물에 대한 추적 할당	0.05
F6	발사 통제(위험한 목표물)	0.02
F7	비 위험 목표물 식별	0.06

[그림 5]는 식별된 컴포넌트의 최장시간 경로를 나타내고 있다. 이들은 서로 매우 복잡한 관계를 갖고 있으며 경로를 통한 테스트 케이스는 무수히 많다. 이 시스템을 3.3절에서 언급한 TFM 모델에 적용하여 테스트 케이스를 추출할 수 있다.



[그림 5] 식별된 컴포넌트 최장시간 경로

TFM 모델의 X와 Y축의 Sequence Diagram과 최대시간 알고리즘을 통한 결과는 [표 2]와 같이 3가지의 테스트 케이스가 식별되었다.

[표 2] 식별된 테스트 케이스

테스트 케이스	경 로	제한시간
1	F1->F2->F3->F4->F6->F5->F3->F5->F4->F7	0.34
2	F3->F7	0.08
3	F2->F4	0.07

사례 연구에서 제시된 무기체계 시스템의 테스트는 [표 2]의 3가지 테스트 케이스를 확인하면 된다. 예를 들어, 첫 번째

경로의 테스트 케이스는 F1->F2->F3->F4->F6->F5->F3->F5->F4->F7 순으로 테스트를 하고 제한시간은 0.34(s)가 된다. 제시된 방법의 장점은 무기체계 특성 중의 하나인 시간제약 사항을 만족하면서 로직을 동시에 점검할 수 있으며, 컴포넌트 실행시간을 통한 테스트 케이스의 수를 줄일 수 있다. 또한 컴포넌트의 정상 작동 여부와 컴포넌트 간의 인터페이스도 확인이 가능하다. 개별적으로 잘 작동하던 컴포넌트가 연결 상태에서 문제가 발생하면 로직 혹은 인터페이스 문제일 수 있으며 시간은 무한대로 측정됨으로 결함을 확인할 수 있다. 또한 작동시간이 너무 빠르거나 늦으면 객체의 누락이나 인터페이스상의 문제가 있을 수 있다.

5. 결론

본 논문에서는 무기체계 시스템 테스트 케이스 선정을 위하여 시스템 테스트의 특징과 시간을 고려한 응답성 그리고 UML 표기의 Diagram을 분석하여 시스템 테스트 모델인 TFM 모델을 설계하였다. 또한 TFM 모델을 활용하여 미사일 방어시스템에 대한 시스템 테스트 케이스를 도출하였다. 본 연구에서 제안한 시스템 테스트 케이스 추출 방법은 컴포넌트 시간요소를 활용하는 관점으로 무기체계 소프트웨어의 신뢰성 향상에 기여할 것이다.

참고문헌

[1]Paul C.Jorgensen, *Software Testing, A Craftsman's Approach, part-IV*, CRC Press, pp.159-212, 1995.
 [2]Unni Sankar, "Software System Testing: A Statistical Approach Overview," IEEE Workshop, 2003.11.
 [3]Mark Sh. Levin and Mark Last, "Multi-Function System Testing: Composition of Test Sets," 8'th IEEE International Symposium on High Assurance Systems Engineering (HASE'04), pp.99-108, 2004.
 [4]http://www.globalsecurity.org/military/systems/aircraft
 [5]OMG, *UML 2.0 Superstructure*, pp.502-505, 2005.7.
 [6]A.En-Nouaary and F.Khenddek, R.Dssouli, "Testing Embedded Real-Time Systems," 7'th International Conference on Real-Time Computing Systems and Applications (RTCSA'00), pp.417-424, 2000.
 [7]Rumbaugh, Jacobson, and Booch, *The Unified Modeling Language Reference Manual*, Addison-Wesley, Reading, 1999.
 [8]David A. Patterson, John L. Hennessy, *Computer Organization And Design*, ELSEVIER, pp.240-278, 2005.
 [9]http://en.wikipedia.org/wiki/Stress_testing
 [10]Sartaj Sahni, Kun Suk Kim, "An O(log n) Dynamic Router-Table Design," IEEE Transactions on Computers, vol. 53, no. 3, pp. 351-363, Mar, 2004.