

A Test Case Generation Method for Data Distribution System of Submarine

Suik Son[†] · Dongsu Kang^{††}

ABSTRACT

Testing maturity is critical to the system under development with lack of experience and skills in the acquisition of the weapon systems. Defects have a huge impact on important system operations. Sharing real-time information will lead to rapid command and mission capability in submarine. DDS(Data Distribution System) is a very important information sharing system and interface between various manufacturers or data formats. In this paper, we analyze data distribution characteristics of distributed data system to group data-specific systems and proposes a test case-generation method using path search of postorder and preorder which is a tree traversal in path testing method. The proposed method reduces 73.7% testing resource compare to existing methods.

Keywords : Data Distribution System, Model-based testing, Path Testing, Tree Traversal, Test Case Generation

잠수함 데이터 분산 시스템을 위한 테스트 케이스 생성 기법

손 수 익[†] · 강 동 수^{††}

요 약

무기체계 기술의 국산화 과정에서 경험과 기술이 부족한 상태로 개발 중인 체계에 대한 테스트 기술의 성숙도는 매우 중요하다. 테스트에서 발견하지 못한 결함은 시스템 운용 중 막대한 영향을 미친다. 잠수함에서 실시간 정보공유 체계는 신속한 지휘결심과 임무능력으로 이어진다. 데이터 분산 시스템은 매우 중요한 정보 공유체계이며 제조사나 데이터 형태가 다른 체계들 간에 이루어진다. 본 논문에서는 데이터 분산 시스템의 데이터 분산 특성을 분석하여 데이터별 체계를 그룹핑하고, 경로 테스트 기법에 트리순회 방법인 후위트리순회, 전위트리순회를 이용하여 테스트케이스 생성기법을 제시한다. 제시한 기법은 기존기법들에 비해 73.7% 테스트 리소스를 감소시킬 수 있다.

키워드 : 데이터 분산 시스템, 모델 기반 테스트, 경로 테스트, 트리순회, 테스트 케이스 생성

1. 서 론

미국 NIST에 따르면 소프트웨어 배포 후의 코드 수정 비용이 설계 단계의 코드 수정 비용보다 30배가 더 소요된다고 한다.[1] 소프트웨어 테스트는 고품질 소프트웨어를 달성하기 위해 소프트웨어 고장의 원인인 결함의 존재를 감지하기 위한 것이다.[2] 단순히 비용의 문제뿐 아니라 소프트웨어의 오류는 사람의 생명을 위협하기도 한다. 무기체계 기술의 국산화 과정에서 경험과 기술이 부족한 상태로 개발 중인 체계에 대한 테스트 기술의 성숙도는 매우 중요하고 테스트를 효과적으로 수행하는 것은 중요한 요소 중 하나이다.[3] 효과적인 테스트 기법이란 각종 체계의 안전한 구현과 개발 전체 비용

의 감소를 위해 조기에 결함을 발견하고 효율적으로 테스트를 수행하는 것이다.[4]

특히 잠수함 데이터 분산 시스템(Data Distribution System) 처럼 제조사나 데이터 형태가 다른 체계들간의 연동을 위한 연구는 매우 중요하고 결함발견을 위해 노력을 집중해야 한다. 그러므로 이와 관련된 연구는 지속되어야 하며, 중요한 의의를 가진다.

본 논문은 잠수함의 운용에 핵심적인 요소로 사용되고 있는 데이터 분산 시스템을 기반으로 데이터 분산 특성을 분석하고 DDS와 연동되는 체계간의 효율적인 테스트 기법을 제시한다. 결함을 발견할 수 있는 테스트 설계와 체계특성에 적합한 테스트 케이스를 생성하여 테스트 리소스를 감소시키고자 한다.

본 논문의 구성은 2장에서 관련연구를 통해 본문에서 다루고자 하는 분야에 대해 알아보고 3장에서 데이터 분산 특성 및 트리순회를 이용한 테스트 케이스 생성 기법을 제시한다. 4장에서 사례 연구를 통해 비교 분석하고, 5장에서 결론을 맺는다.

[†] 준 회 원 : 국방대학교 컴퓨터공학전공 석사

^{††} 종신회원 : 국방대학교 컴퓨터공학전공/사이버전과정 교수

Manuscript Received : October 18, 2018

Accepted : November 26, 2018

* Corresponding Author : Dongsu Kang(greatkoko@kndu.ac.kr)

2. 관련 연구

2.1 잠수함과 데이터 분산 시스템

재래식 디젤잠수함은 DDS(Data Distribution System)를 이용하며 주요 구성은 각 체계의 신호를 연동하는 인터페이스 케이블과 PCB (Printed Circuit Board)들로 구성되어 있다. Fig. 1은 데이터 분산 관계도로 체계의 안정성과 신뢰성을 위해 2중으로 구성되어 있으며 데이터 분산 체계의 샘플로 ISUS 83과 DDS Cabinet 1간에는 6개의 Interface와 DU(Distribution Unit)을 통해 데이터가 공유되어진다. 경로 테스트에서 이러한 Interface는 경로가 되며 WCS, NAD, DU와 같은 단위(체계)는 노드로 나타내어 진다.

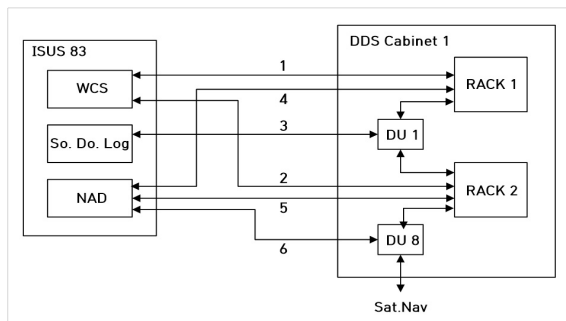


Fig. 1. ISUS 83 Data Interface

다음으로 OMG에서 정의한 DDS(Data Distribution Service)가 있다. Fig. 2는 OMG DDS의 주개념 중 하나로 데이터 형태(Data Type)인 Topic을 기준하여 미들웨어에서 데이터의 공유 대상 간에 식별(Grouping)을 사전에 해주는 것이다.[5-6]

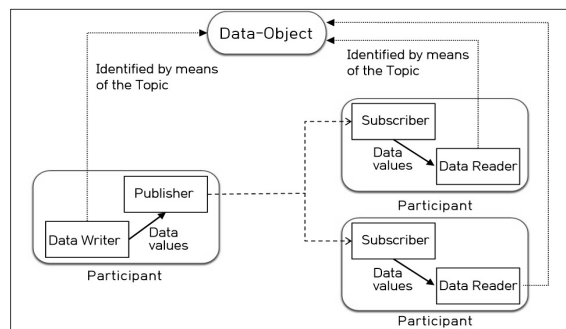


Fig. 2. Concepts of Data Distribution

규모가 작은 무기체계에서는 DB(Distribution Box)에 의해 한가지의 데이터만 일부 체계 간 공유가 이루어지고 데이터의 개수에 따라 여러 개의 DB가 존재한다. 하지만 이지스함이나 잠수함과 같은 플랫폼에서는 여러 체계들이 복잡하게 이루어져 정밀한 기능을 발휘하게 되는데 통상 작전, 항해, 기관운용 관련 데이터가 별도의 DDS 체계에 의해 통합적으로 분산된다.

실제 무기체계 내 이러한 데이터들의 Interface는 복잡하게 구성되어 있으며 별도의 분산체계가 없이는 상호 온전히 연동되지 않는 체계들이다. 이와 같이 데이터 분산 시스템의 연동 테스트를 수행할 때 고려해야 하는 요소가 다양하고 복잡

하므로 성공적인 테스트를 위해서는 효율을 향상 시킬 수 있는 테스트 방법이 필요하다.[7]

이렇게 데이터를 분산하는 시스템들의 공통적인 특성은 구성된 모든 체계들이 모든 데이터를 생성/송신하고 수신/사용하지 않는다는 것이다. 같은 데이터의 공유 대상 간에는 별도로 구분이 되며, 규모가 큰 시스템일수록 그 수가 많아지게 된다. 즉, 테스트 할 때 분산되는 데이터를 기준으로 구분하여 테스트를 효율적으로 실시할 수도 있다는 것이다.

2.2 모델 기반 테스트

모델 기반 테스트는 복잡하게 돌아가는 체계를 이해하기 위한 방법이다. 보잉사에서는 새로운 제트기를 설계할 때 수백만 번 이상의 시뮬레이션을 통해 최적의 양력과 효율을 얻을 수 있는 제트기를 설계한다. 모델은 체계를 추상화시키는 방법이고 다양한 상황에 대한 질문에 답을 줄 수 있을 뿐만 아니라 설계를 이끌어내기도 한다.

모델링 작업을 통해 복잡한 이슈를 이해하기 쉽고 잘 정의된 단위 조각들로 나누고 결국에는 체계 전체를 한층 자세히 이해할 수 있게 된다.[8] 모델 기반 테스트에서는 테스트 모델을 별도로 구축해야 한다는 점이 단점이라고 볼 수가 있지만 테스트 케이스 생성의 유연성, 테스트케이스의 재사용성과 요구사항과의 추적성 측면에서 상당한 이점을 제공한다.[9]

본 논문에서는 모델링된 체계를 통해 데이터 분산 특성을 분석하고 모델을 FG(Flow Graph)로 변환하여 경로 테스트 기법을 연구한다. 모델의 한 예인 UML을 사용한 모델을 바탕으로 그래프로 변환하여 테스트 케이스를 생성한 기존연구로는 유즈케이스와 시퀀스 다이어그램을 기반한 기법[10, 11] 액티비티 다이어그램을 기반한 기법[12, 13]이 있다.

2.3 트리순회

트리순회는 트리의 모든 노드를 각각 한 번씩 방문하는 일련의 과정으로 트리의 모든 노드를 일렬로 정렬하는 역할을 하며 방문하는 순서에 따라 분류한다.

순회방법에는 두 가지가 있다. 너비우선 순회는 각 노드의 방문을 가장 낮은(혹은 높은) 레벨부터 아래(혹은 위)방향으로 수행하며 각 레벨은 왼쪽에서 오른쪽(혹은 오른쪽에서 왼쪽) 방향으로 수행한다. 깊이 우선 순회는 트리의 왼쪽(혹은 오른쪽)으로 가능한 깊이 순회하는데 끝에 다다르면 한 단계 위로 올라와서 오른쪽(혹은 왼쪽)으로 간 다음 다시 왼쪽(혹은 오른쪽)으로 가능한 깊이 순회하는 방법이다. 깊이 우선 순회의 3가지 동작은 V(트리의 노드 방문), L(왼쪽 서브트리 탐색), R(오른쪽 서브트리 탐색)이다. 깊이 우선 순회 방식 3가지는 VLR[전

Table 1. Tree Traversal

LRV(postorder)	VLR(preorder)
postorder(node)	preorder(node)
if node.left ≠ null then	print node.value
postorder(node.left)	if node.left ≠ null then
if node.right ≠ null then	preorder(node.left)
postorder(node.right)	if node.right ≠ null then
print node.value	preorder(node.right)

위(preorder) 트리 순회], LVR[중위(inorder) 트리 순회], LRV[후위(postorder) 트리 순회]가 있으며 Table 1과 같다.[14]

2.4 경로 테스트 및 기존기법 분석

경로 테스트는 프로그램의 가능한 모든 경로를 테스트하는 기법이다. 반복 구조에서 각 반복을 하나의 독립적인 경로로 생각한다면 반복할 때마다 다른 경로로 인식하므로 경로 테스트 기법으로 테스트하기 쉽지 않으며, 현실적으로 모든 경로 테스트는 불가능 하게 된다.

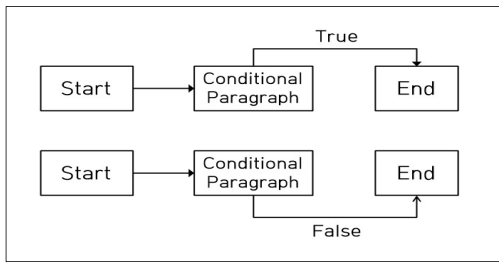


Fig. 3. Linear Independent Control Flow Graph

기본 경로는 함수에서 선형적으로 독립된 경로로 정의된다. 선형적으로 독립된 기본 경로는 함수의 독립된 경로의 유한 집합이 된다. 예를 들어 Fig. 3과 같이 하나의 조건문을 가진 함수라면 두 개의 선형적으로 독립된 경로를 갖는다. 선형적으로 독립된 기본 경로 조합은 기본 경로 셋의 슈퍼셋과 다른 경로를 표현함으로써 함수의 가능한 모든 경로를 커버한다. 따라서 기본 경로 테스트라는 구조적 테스트 기법은 완벽한 경로 테스트가 불가능한 문제에 대해 가능한 모든 의미 있는 경로를 커버함으로써 효과적인 해결책을 제시한다.[8]

경로 탐색 시에는 사전처리를 통해 경로 탐색의 효율을 향상시키는 연구가 이루어져 왔다. 노드들의 묶음이나 요약 그래프를 만들어 탐색 노드를 줄어든게 하는 기법[15, 16], 탐색 영역을 분할하거나 열등하거나 불필요한 노드 또는 간선을 구분하여 처리 하거나 선호하는 노드나 간선을 우선 탐색하는 기법[17, 18]들이 있으며, 탐색에 필요한 정보 테이블을 만들고 노드들의 정보를 활용하는 기법[19], 2가지 이상 경로 탐색 알고리즘을 하이브리드 한 기법[20] 등이 있다. 사전처리 방법으로 FG(Flow Graph)를 생성하여 경로를 찾는 기존 기법의 강약점 분석하여 정리하면 Table 2와 같다.

먼저 DFS(Depth First Search)로 경로는 찾는 기법[21]은

모델을 Flow Graph로 변환하여 테스트 케이스 생성 방법을 단순화한 장점이 있지만, 경로 생성에 대한 구체화가 부족하고 루프 및 중복 수행에 대한 제한이 있으며 단순 DFS의 적용으로 테스트 케이스가 많아진다.

모델 단축을 통해 테스트 케이스를 생성하는 기법[15]은 다양한 시스템에 전반적으로 적용이 가능하고 단순화한 경로로 테스트 케이스 감소가 가능하였다. 하지만 순환경로를 단순화 하여 DAG(Directed Acyclic Graph)를 생성하는 과정에서 여러 개의 중복되는 순환 대응에 제한된다.

차수대칭(Degree Symmetry) 그래프 변환을 통해 오일러 경로를 찾고 중복수행을 제거한 기법[22]은 다양한 시스템에 전반적으로 적용가능할 뿐만 아니라 테스트 케이스를 감소시켰고, 테스트 효율을 향상 시켰다. 하지만 변환규칙이 단순하여 특정 경로를 탐색하는 것과 자동화 수행이 제한되는 특징이 있다.

이러한 기법들은 연구범위 내에서 경로탐색의 효율을 달성하였지만, 모든 탐색 대상에 부합될 순 없다는 공통적인 한계가 존재한다. 따라서 잠수함 데이터 분산 시스템의 특성이 반영되어 기존연구들을 차용한 방법보다 효율을 향상시킬 수 있는 테스트 케이스 생성 기법이 필요하다.

3. 테스트 케이스 생성 기법

본 장에서는 기존의 테스트 케이스 생성기법이 가진 제한 사항을 극복하기 위한 기법을 제시한다. 제안 기법은 Fig. 4와 같이 5단계의 IDEF-0표기법으로 나타낼 수 있다.

체계모델링은 체계도를 바탕으로 대상 체계를 모델링하는 첫번째 단계이다. 데이터별 체계 그룹핑 단계에서는 2.1절의 데이터 분산 특성 분석결과를 바탕으로 테스트할 체계를 데이터별로 그룹핑하는 것이다. DFG(Data Flow Graph) 생성은 모델링된 다이어그램을 그래프로 변환하는 규칙을 적용하여 DFG를 생성한다. 이후 트리순회 방법인 LRV, VLR을 하이브리드하여 탐색하고 불필요 노드를 제거하여 테스트 경로를 생성한 후 테스트 케이스를 생성하게 된다.

3.1 데이터별 체계 그룹핑

데이터를 분산하는 시스템을 분석한 결과는 구성된 모든 체계들이 모든 데이터를 생성/송신하고 수신/사용하진 않으며, 같은 데이터의 공유 대상 간에는 별도로 구분이 되어 진다는 것이다. 이러한 특성을 활용하여 테스트 할 때 분산되어 지는 데이

Table 2. Comparison of Test Case Generation Methods with Pre-processing

Method	pre-processing	Strength	Weakness
DFS[21] (Depth First Search)	DFG	· Simplified method of creating test cases	· Lack of detail for path generation · Restrictions on loop and overlap execution · Too many test cases
DAG[15] (Directed Acyclic Graph)	DFG, DAG	· Can be applied wide range of systems · Reduced test case with simplified path	· Overlap with cycles and execution · Restrict automation due to complex conversion rules
Euler Path Search[22] (with Deduplication)	DFG, Degree Symmetry	· Can be applied wide range of systems · Reduced test case with D-S · Achieve efficiency with deduplication	· Simple conversion rules restrict some path search · Restrict automation

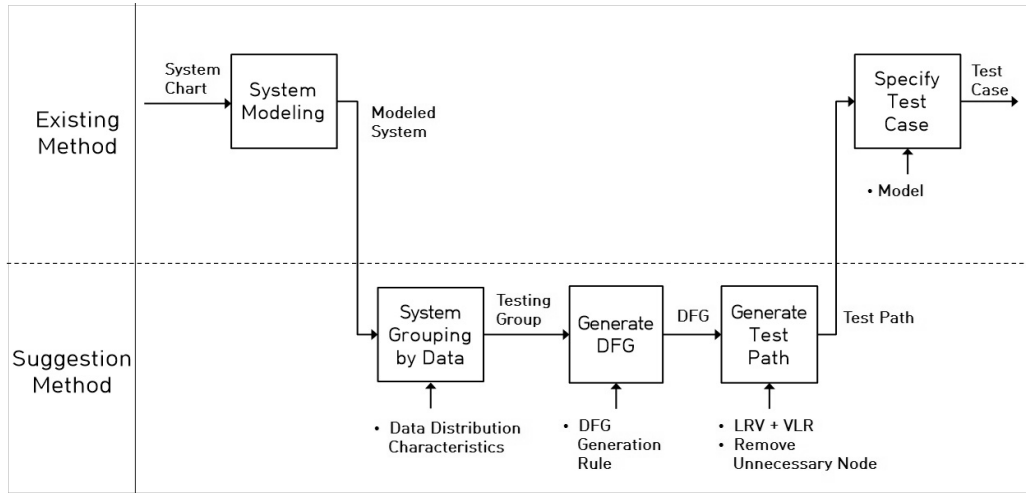


Fig. 4. Testcase Generation Process

Table 3. Mean of Modeling Abbreviation

	OMG DDS	Submarine DDS
Participant(Px)	Participants in System	Interlinked System
Topic(Tx)	Data Type	Data Type
Data Publish(DP)	Publish(Writer)	Transmit
Data Subscribe(DS)	Subscribe(Reader)	Receive

터를 기준으로 체계를 그룹핑하여 테스트를 실시할 수 있다.

기법의 예시를 보이기 위한 모델링에 사용된 약어는 OMG에서 정의한 DDS의 용어를 가져와 표현하였으며, 약어의 의미를 정리하면 Table 3과 같다.

기법 제시를 위해 잠수함 데이터 분산 시스템을 기본으로 하여 유사하게 샘플 모델링한 결과는 Fig. 5와 같다. Px는 연동된 체계, DDS의 Tx는 데이터 종류를 의미한다. 연동체계에 Interface Port로 모델링한 TxDP / TxDS 는 데이터(Tx)가 송신(Transmit) 또는 수신(Receive)되는 것을 나타내기 위해 표기하였으며 데이터가 연동체계 간 상호 전송(송신 또는 수신)되는 과정을 테스트하는 것을 의미한다. 이때, 테스트의 유효성 고려 송신 및 수신 체계가 1개 이상 존재할 때 테스트하는 것을 기본으로 한다.

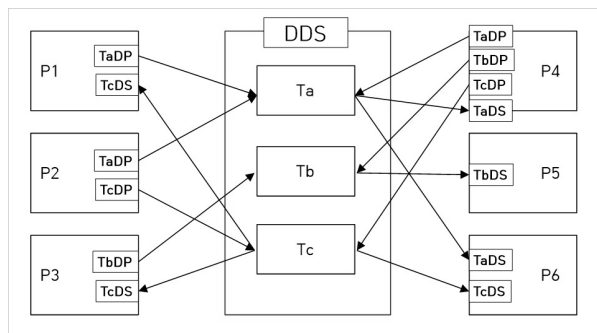


Fig. 5. Submarine Distributed Data System Modeling(example)

기존의 방법은 DDS로 연동되는 모든 체계들이 테스트할 준비가 되면 테스트를 했기 때문에 여러 문제점이 발생될 수 있다. 이를 사전에 해결하기 위한 한 가지 방안으로 테스트를 조기에 실시하는 방법이 있다. 부분적인 테스트를 앞당겨 실시할 수 있는 방안[1] 중 중복수행이 없이 '같은 데이터(Topic)를 공유하는 체계(Participant)가 준비 되었을 때 테스트' 하는 데이터 중심 테스트가 적절하다. 그에 따른 테스트 그룹은 아래와 같다.

- TG 1(Ta) : P1.TaDP, P2.TaDP, P4.TaDP, P4.TaDS, P6.TaDS
- TG 2(Tb) : P3.TbDP, P4.TbDP, P5.TbDS
- TG 3(Tc) : P2.TcDP, P4.TcDP, P1.TcDS, P3.TcDS, P6.TcDS

3.2 DFG 생성

데이터 분산 시스템과 연동되는 체계들을 모델링한 결과 분석된 특성은 첫 번째로 체계간 데이터 흐름을 살펴본 결과 분기/결합 경로는 존재하지만 반복 경로는 일반적으로 존재하지 않았다. 두 번째, DDS를 중심으로 하여 체계들이 트리의 형태를 보인다는 것이다. 이러한 특징을 바탕으로 데이터에 대한 체계도를 통해 생성된 모델을 DDS를 루트노드로 하여 트리순회가 가능한 DFG로 변환한다. 이후 트리순회를 사용하여 경로 탐색을 할 수 있게 된다.

체계를 모델링한 후 데이터 별로 그룹핑한 결과 순차, 분기, 결합의 3가지 종류로 데이터가 이동 되는 것으로 분석되었다. 모델링된 체계와 DFG는 동일한 데이터의 흐름을 나타낼 수 있도록 해야 하기 때문에 변환할 때에 기본적인 규칙을 준수하며 변환한다. 기본규칙은 모델링된 다이어그램을 DFG로 변환할 때에 모델의 다음 다이어그램이 이전 것보다 먼저 나열되지 않도록 변환되어야 한다는 것이다.

기본규칙을 준수하여 고안한 DFG 변환 방법은 Fig. 6과 같다. 변환 방법은 DDS를 기준으로 DDS 이전과 DDS 이후로 나누어지는데 순차적 흐름은 공통적으로 적용된다. DDS

이전에 데이터의 흐름이 분기될 때에는 별도의 경로를 가지게 하며 DDS 이전에 데이터가 결합하는 것과 DDS 이후에 데이터가 분기될 때에는 트리순회의 방법에 맞춰 기본규칙의 준수 하에 트리순회가 가능한 형태로 변환하게 된다. ㉞/㉟는 DFG가 이진 트리형태를 기준으로 변환하기 때문에 발생하는 다음/임시 노드이며 모델링된 세부 체계들은 데이터의 송신측(Transmit)은 T, 수신측(Receive)은 R로 표시한다.

	Model	DFG
in order		
Diverge before DDS		
Join before DDS		
Diverge after DDS		

Fig. 6. Model to DFG Conversion Method

3.3 테스트 경로 생성

동일한 테스트 결과를 가진다면 테스트에 사용된 리소스가 적을수록 테스트가 효율적이라고 할 수 있다. 이는 일반적인 방법으로 테스트하거나 다른 대상에 효율적인 기존기법을 차용하는 것이 아니라 대상체계인 DDS에 적합한 방법을 특성에 맞게 적용해야 가능하다.

경로탐색 기법에서 효율적인 경로 탐색을 위해 많은 연구가 이루어지고 있으며 해결 과제 중 하나는 각각의 중복되는 동일한 경로의 탐색이며 이는 불필요한 중복테스트로 효율이 떨어지게 된다. 트리순회는 트리의 모든 노드를 각각 한 번씩 방문하는 일련의 과정으로 트리의 모든 노드를 일렬로 정렬하므로 불필요한 중복을 최소화 시킬 수 있는 특성이 있다.

변환된 DFG는 결국 DDS를 루트노드로 하는 이진트리의 형태를 보이며 DFG에서 테스트 경로를 탐색 할 때 트리순회 방법 중 깊이 우선 순회(DFT) 방법인 LRV(후위 트리 순회)와 VLR(전위 트리 순회)를 사용할 수 있다. LRV, VLR 순회 순서는 Fig. 7과 같다. 이때, 변환된 그래프에서 DDS 이전에는 LRV, 이후에는 VLR을 사용하여 탐색하는 하이브리드 방식으로 경로를 탐색한다. 데이터의 흐름이 역행되지 않아야 하는 기본규칙과 더불어 트리순회 방법을 고려하여 모델을 DFG로 변환하는 규칙을 생성하였으므로 모든 노드들이 데이터의 흐름이 역행되지 않으면서 일렬로 정렬되어진다. 이후 DFG 생성을 위해 사용된 불필요한 임시노드(⊗)를 제거하여 최종 테스트 경로를 생성한다.

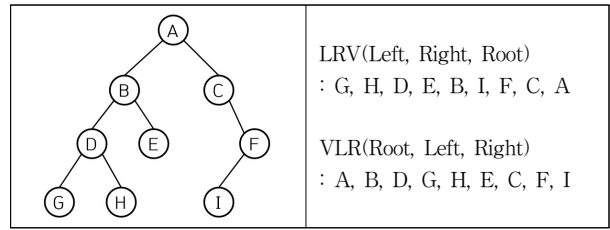


Fig. 7. Method of LRV, VLR

4. 사례 연구 및 비교 평가

본 장에서는 잠수함 데이터 분산 시스템(DDS)에 적용하여 사례 연구와 분석을 진행하였다.

4.1 기법 적용

잠수함에서 사용하는 DDS는 수십명의 생명과 직결되어 있는 중요한 체계 중 하나이다. 잠수함에서는 데이터 분산 특성을 가진 DDS를 운용하며 24시간 작동하는 DDS는 결함이 발생하지 않도록 매우 높은 신뢰성을 유지해야 한다. 따라서 한정된 테스트 리소스를 이용하여 효과적인 테스트를 통한 신뢰성 향상의 필요성은 갈수록 높아 질 것이다. 다음과 같은 순서로 테스트 케이스를 생성한다.

1) 데이터 분산 시스템(DDS) 체계 모델링

DDS와 각 체계를 모델링한 결과는 Fig. 8과 같다. Px는 DDS 본체와 직접적으로 연결되는 데이터 송/수신 체계를 기준으로 모델링 하였으며, 본 단계에서는 그 전/후의 세부체계와 송수신 관계는 불필요하며 DFG를 생성하는 3단계에서 세부 체계까지 모델링된다.

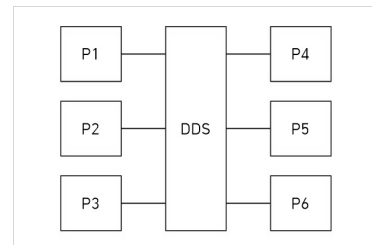


Fig. 8. DDS System Modeling

2) 데이터별 체계 그룹핑

모델링된 체계를 데이터를 포함하여 모델링한 결과는 Fig. 9A와 같으며, 데이터 별로 체계를 그룹핑한 결과는 Fig. 9B, 9C, 9D와 같다.

데이터별 체계 그룹핑 결과를 나열하면 아래와 같다.

- Ta : P1, P2, P3, P4, P5, P6
- Tb : P2, P4, P5
- Tc : P3, P6

3) DFG 생성

Ta의 다이어그램(Fig. 9B)을 DFG(Data Flow Graph)로 변환하기 위해 데이터 흐름(Data Flow)방향까지 자세히 모델링

하면 4개 개소에서 데이터를 송신하고 2개 개소에서 수신되어 진다. 이를 데이터의 흐름을 나타내기 위해 증폭기, 변환기, 처리기 등과 같은 세부 체계까지 모델링한 결과는 Fig. 10과 같다. Px의 구성요소 중 데이터의 송신측(Transmit)은 T, 수신측(Receive)은 R을 사용하여 나타내었다. 또한 Ta의 체계그룹의 세부 구성을 실제 체계에서 있을 수 있는 모든 경우의 수를 반영하기 위해 순차, 분기, 결합이 모두 포함되는 사례를 활용하여 모델링한다.

이후 상세 모델링된 다이어그램(Fig. 10)을 DFG로 변환하기 위해 Fig. 6의 Model to DFG 변환 방법을 사용하여 변환한 결과는 Fig. 11과 같다.

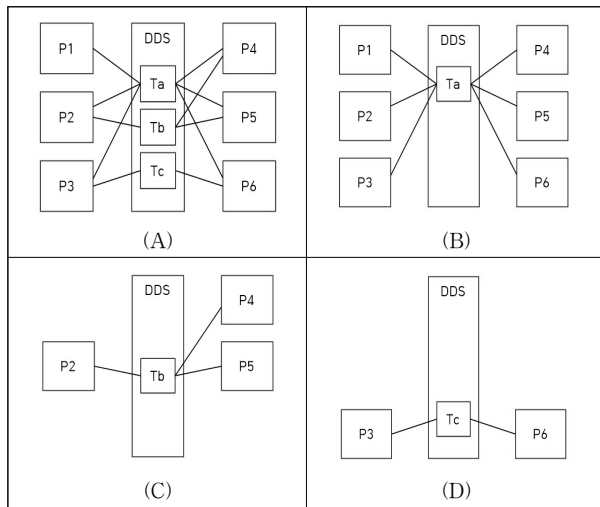


Fig. 9. System Grouping Results by Data

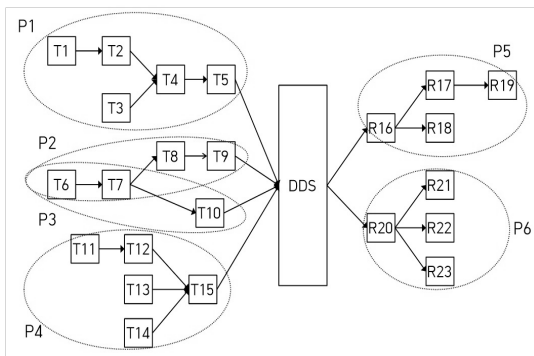


Fig. 10. Detailed System Modeling of Ta

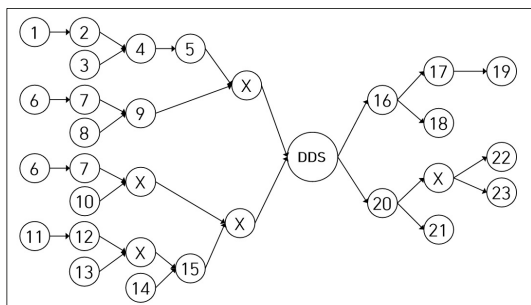


Fig. 11. DFG Generation Result

4) 테스트 경로 생성

DDS 이전에는 LRV, 이후에는 VLR을 사용하여 변환된 그래프를 탐색하는 하이브리드 방식으로 경로를 생성한다. 그리고 불필요 노드(⊗) 제거하여 최종 테스트 경로를 생성하였고, 임의 모델링한 Tb와 Tc도 같은 방법으로 테스트 경로를 생성하였다. 생성 결과 한 번에 탐색하는 경로의 길이(노드 수)는 증가하지만 분산데이터의 개수와 같은 개수의 테스트 경로가 생성되는 특징이 있다. 그 결과는 Table 4와 같다.

Table 4. Test Path of Modeled DDS

		Test Path
Ta	Search Path	1,2,3,4,5,6,7,8,9,X,6,7,10,X,11,12,13,X,14,15,X,DDS,16,17,19,18,20,21,X,22,23
	Test Path	1,2,3,4,5,6,7,8,9,6,7,10,11,12,13,14,15,DDS,16,17,19,18,20,21,22,23
Tb	Search Path	1,2,3,4,DDS,5,6,7,8,10,9
	Test Path	1,2,3,4,DDS,5,6,7,8,10,9
Tc	Search Path	1,2,3,X,4,5,DDS,6,7,8,9,10
	Test Path	1,2,3,4,5,DDS,6,7,8,9,10

5) 테스트 케이스 명세

테스트 경로를 통해 테스트 시나리오를 생성하여 테스트 케이스를 명세할 수 있다.[22] 생성된 Ta 에 대하여 테스트 케이스를 명세하면 Table 5와 같다.

Table 5. Test Case Specification

Test Case ID	TC-01
Value	Ta
Test Path	1,2,3,4,5,6,7,8,9,6,7,10,11,12,13,14,15,DDS,16,17,19,18,20,21,22,23
Test Scenario	START→T1→T2→T3→T4→T5→T6→T7→T8→T9→T6→T7→T10→T11→T12→T13→T14→T15→DDS→R16→R17→R19→R18→R20→R21→R22→R23→END
Expected Result	Send/Recevice Result (Success, Fail)

4.2 분석 및 평가

제안 기법과 기존기법을 비교하였다. 데이터의 생성이 시작되는 체계부터 최종 수신이 되어지는 마지막 체계까지 기본적인 데이터 흐름을 DFS[21]를 이용하여 순차적으로 테스트 하는 기법을 첫 번째 비교 대상, 모델단축(Summary)[15]을 이용하여 가능한 노드를 모음으로써 축약그래프를 활용한 기법을 두 번째 비교 대상, 모델단축 기법과 더불어 중복수행을 제거(Deduplication)[22]하는 기법까지 추가한 기법을 세 번째 비교 대상으로 하였다.

모델단축은 DDS를 기준으로 동일하게 발생되는 송신(Transmit) 경로에 적용하였으며, 중복수행 제거는 DDS 전/후로 동일한 축약그래프가 반복될 때 적용하였다.

본 논문에서는 기본 경로 테스트에서 기본 커버리지가 할 수 있는 모든 노드와 엣지를 한번이상 지나 수행되면 완료된 것으

로 하였다. 또한 특정 노드나 엣지에 결함이 있다고 가정시에 해당 노드나 엣지를 지나게 되면 테스트를 통해 결함이 발견되었다고 볼 수 있다. 특정 노드나 엣지에 결함이 있다고 가정하여 기법 별 테스트케이스 산출 결과 기존 기법과 제안 기법에서 결함이 존재하는 노드나 엣지를 모두 테스트하였으며 결함을 발견할 수 있었다. 이때, 기법별로 테스트 케이스의 수나 수행횟수가 적을수록 효율적인 테스트를 수행했다고 볼 수 있다.

먼저, 테스트 케이스 수에 의한 비교 분석이다. 대상 체계는 Ta~Tg의 연동체계를 임의의 모델링하여 누적시켜 규모가 점차 커지는 체계로 볼 수 있다. 데이터 개수가 1~7개까지 존재하는 데이터 분산 시스템이다. 그 결과는 Fig. 12와 같다.

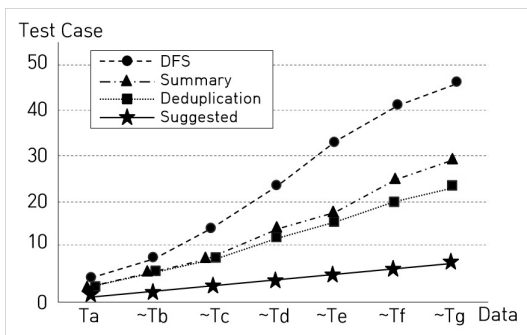


Fig. 12. Comparison of Testcase Count

두 번째, 경로 테스트에서 각 노드를 지나가는 것은 해당 체계를 수행한 것을 의미하며 같은 테스트 결과를 가져온다면 테스트 수행횟수(노드의 개수)가 적은 것이 테스트 비용과 시간이 적다고 할 수 있다. 테스트 수행횟수 비교결과는 Fig. 13과 같다.

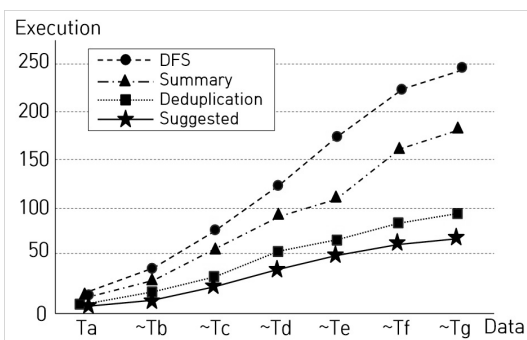


Fig. 13. Comparison of Test Execution Count

제안기법을 모델단축 후 중복제거까지 실시한 기존기법과 비교하면 기존기법의 테스트 케이스 수는 23개, 제안기법은 7개로 30.4%로 감소하였다. 기존기법의 수행횟수는 95회이며 제안기법은 70회로 73.7%로 상대적으로 감소하여 효율적이라 할 수 있다. 또한 Ta부터 Tg까지 데이터의 개수와 더불어 체계의 규모가 커질수록 테스트 케이스 및 수행횟수의 효율이 증가하고 있다. 테스트는 개발 리소스의 소모가 많은 과정이다. 체계의 신뢰성이나 안전성 향상을 위한 테스트의 비용은 높은 수준을 달성할수록 비용이 증가하기 때문에 테스트의 준비부터 수행까지 효율을 최대로 높여야 한다. 비교 분석

에 사용한 임의의 모델링 체계를 통해 테스트 케이스 수 및 수행횟수의 감소를 통해 테스트 효율을 향상 시켰으며, 테스트에 필요한 리소스인 비용과 시간을 줄일 수 있게 된다.

예를 들면 Ta의 경우는 T1, T2 2개소에서 데이터를 최초 생성하여 각기 송신을 하며 R7, R8 2개소에서 데이터를 최종 수신하여 사용하는 테스트 그룹이다. ① 기법은 송신 2개소 수신 2개소에 의한 테스트 케이스 4개와 DFS에 의한 기본탐색으로 테스트 케이스별 5회의 수행횟수로 총 20회의 수행횟수를 가지며 ② 기법은 단축될것이 없는 기본그룹으로 동일한 값을 가진다. ③ 기법은 DDS를 기준으로 T2- DDS-R8의 중복경로가 제거 되어 테스트 케이스 3개, T1-DDS의 중복경로가 추가제거되어 수행횟수는 13회로 감소하였다. 제안 기법(④)은 트리순회를 통한 테스트 경로 생성으로 수행횟수 9회를 가지는 1개의 테스트케이스가 생성된다. 비교분석 결과는 Table 6과 같다.

Table 6. Results of Comparative Analysis

		Data Count	①	②	③	④	Comparison
Test Case Count	Ta	1	4	4	3	1	33.4%
	~Tb	2	8	5	5	2	40%
	~Tc	3	14	8	8	3	37.5%
	~Td	4	23	14	12	4	33.4%
	~Te	5	32	17	15	5	33.4%
	~Tf	6	41	25	20	6	30%
	~Tg	7	47	29	23	7	30.4%
Execution Count	Ta	1	20	20	11	9	81.8%
	~Tb	2	40	32	20	16	80%
	~Tc	3	75	54	34	27	79.4%
	~Td	4	126	93	51	40	78.4%
	~Te	5	176	116	66	50	75.8%
	~Tf	6	224	164	85	62	72.9%
	~Tg	7	249	182	95	70	73.7%

① DFS ② Summary ③ Deduplication ④ Suggested

5. 결론 및 향후 연구

무기체계 기술의 국산화 과정에서 경험과 기술이 부족한 상태로 개발 중인 체계의 테스트 기술의 성숙도는 매우 중요하며 데이터 분산 시스템처럼 제조사나 데이터 형태가 다른 체계들간의 연동관계가 무기체계의 발전에 맞춰 지속 확장되는 시스템은 효과적으로 결함을 발견하기 위한 노력이 지속되어야 한다.

제시한 기법은 그룹핑된 체계들을 모델링하여 데이터 흐름의 특성을 분석하고 트리순회기법 적용을 위한 Model to DFG 변환방법을 고안하였으며 이진트리의 형태를 가지는 DFG로 변환하였다. 이후 DDS 이전/이후로 나누어 트리순회 방법인 LRV(후위 트리 순회), VLR(전위 트리 순회)를 이용하여 테스트 경로를 생성하고 테스트 케이스를 생성하였다. 이는 테스트 케이스 수와 수행횟수를 감소시킬 수 있다.

향후에는 제한한 기법을 바탕으로 테스트 커버리지와 테스트 케이스 명세에 대한 연구를 실시할 예정이다.

References

[1] S. I. Son and D. S. Kang, "A Method of Interface Test for Submarine Data Distribution System," In *Proceedings of KIPS*, Vol.25, No.1, pp.219-221, 2018.

[2] S. Sabharwal, R. Sibal, and C. Sharma, "Applying Genetic Algorithm for Prioritization of Test Case Scenarios Derived from UML Diagrams," *Journal of IJCSI*, Vol.8, Issue 3, No.2, pp.433-444, 2011.

[3] D. S. Kang, "The Defense use of High Performance Computing (HPC) Technology," *Journal of KIISE*, Vol.34, pp.85-91, 2016.

[4] S. S. Kim and D. S. Kang, "Fuzzing-based Test Case Generation Technique for Multimedia File Vulnerability Analysis," *Journal of SE*, Vol.14, No.6, pp.441-458, 2017.

[5] OMG, "Data Distribution Service for Real-time Systems Version 1.2," 2007.

[6] OMG, "The Real-time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol Specification version 2.1," 2010.

[7] D. S. Kang, "Ways to Improve the Weapon System Software," KNDU Thesis Collection, Vol.24, pp.71-90, 2016.

[8] A. Page, K. Johnston, and B. Rollison, "How we test software at Microsoft," Acorn Publisher, 2009.

[9] H. S. Choi, "Model-based Test - Concepts and Issues," *Journal of KISS*, pp.59-71, 2014.

[10] S. K. Swain, D. P. Mohapatra, and R. Mall, "Test case generation based on use case and sequence diagram," *Journal of IJSE*, Vol.3, No.2, pp.21-52, 2010.

[11] M. Sarma, D. Kundu, R. Mall "Automatic test case generation from uml sequence diagram," In *Proceedings of 15th International Conference on Advanced Computing and Communications*, pp.60-65, 2008.

[12] C. Mingsong, Q. Xiaokang, and L. Xuandong, "Automatic test case generation for uml activity diagrams," In *Proceedings of AST*, pp.2-8, 2006. 5.

[13] D. Kundu, and D. Samanta, "A novel approach to generate test cases from uml activity diagrams," *Journal of Object Technology*, Vol.8, No.3 pp.65-83, 2009. 5.

[14] Tree traversal [Internet], https://en.wikipedia.org/wiki/Tree_traversal

[15] S. H. Lee, D. S. Kang, C. Y. Song, and D. K. Baik, "A Method of Test Case Generation using BPMN-based Model Reduction for Service System," *Journal of KIPS*, Vol.16-D, No.4, pp.595-612, 2009.

[16] H. W. Kim, H. J. Seo, and Y. K. Lee, "Efficient Shortest Path

Techniques on a Summarized Graph based on the Relationships," *Journal of KIISE*, Vol.44, No.7, pp.710-718, 2017.

[17] Y. S. Myung, "An algorithm for the preprocessing shortest path problem," *Journal of KMSR*, Vol.19, No.1, pp.55-66, 2002.

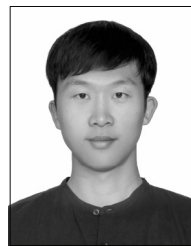
[18] S. H. Ok, J. H. Ahn, S. H. Kang, and B. G. Moon, "A Combined Heuristic Algorithm for Preference-based Shortest Path Search," *The Institute of Electronics Engineers of Korea - Telecommunications*, Vol.47, No.8, pp.74-84, 2010.

[19] I. H. Jang and K. B. Sim, "Optimal Routing Path Selection Algorithm in Ad-hoc Wireless Sensor Network," *Journal of KIIS*, Vol.15, No.6, pp.736-741, 2005.

[20] Y. H. Lee and S. W. Kim, "A Hybrid Search Method of A* and Dijkstra Algorithms to Find Minimal Path Lengths for Navigation Route Planning," *Journal of IEIE*, Vol.51, No.10, pp.109-117, 2014.

[21] P. Yotyawilai and T. Suwannasart, "Design of a Tool for Generating Test Cases from BPMN," in *Data and Software Engineering(ICODSE), IEEE 2014 International Conference on*, pp.1-6, 2014.

[22] J. J. Park and D. S. Kang, "A Method of Test Case Generation Using BPMN-based Path Search," *Journal of KIPS Software and Data Engineering*, Vol.6, pp.125-134, 2017.



손 수 익

<https://orcid.org/0000-0001-9121-4992>
 e-mail : s020436@naver.com
 2007년 해군사관학교 전산과학(학사)
 2018년 국방대학교 컴퓨터공학전공(석사)
 관심분야 : SW Engineering, SW Testing
 DDS, MBT, Path Testing



강 동 수

<https://orcid.org/0000-0001-6481-5071>
 e-mail : greatkoko@kndu.ac.kr
 2011년 고려대학교 컴퓨터공학과(박사)
 2015년~현 재 국방대학교 컴퓨터공학전공/
 사이버전과정 부교수
 관심분야 : Weapon System Software,
 Software Security Testing,
 Defense Acquisition