

---

## Java/RTI를 위한 Test Suite 개발

이정욱\* · 김용주\* · 김영찬\*

\*한밭대학교

Development for Java/RTI Test Suite

Jung-uk Lee\* · Yong-joo Kim\*\* · Young-chan Kim\*\*\*

\*Hanbat National University

E-mail : toodul1@freechal.com

### 요 약

HLA(High Level Architecture)는 미 국방성에서 제안한 분산 시뮬레이션을 위한 실시간 분산 데이터 공유를 위한 객체, 시간, 인터페이스 규격을 말한다. 이 HLA는 (1) Rules, (2) Interface Specification, (3) Object Model Template 등의 3가지 구성요소로 정의되며, 인터페이스 명세를 구현한 것이 RTI(Run-Time Infrastructure)이다. RTI는 분산 운영체계가 응용 프로그램에게 서비스를 제공하는 것과 유사한 방법으로 Federation을 구성하고 있는 Federate들에게 상호연동에 필요한 서비스를 제공한다. 모든 서비스들이 표준에 적합하게 구현되었는지 테스트하기 위한 방법은 DMSO에서 제안한 RTI를 검증하기 위한 2단계 과정을 통해 테스트된다. 본 논문에서는 1단계 테스트를 위한 테스트 절차와 그 방법에 대해 논한다. 테스트 결과를 토대로 각 영역에서 DMSO에서 제안한 Level One Test Procedures에 따라 RTI가 Interface Specification에 적합하게 구현되었는지 확인하며, 각 단계마다 올바른 명령과 예상된 결과가 나오는지 테스트하기 위한 Test Suite를 개발한다.

### ABSTRACT

The HLA is defined by three components: (1) Rules, (2) the HLA Interface Specification, and (3) the Object Model Template(OMT). The RTI(Run-Time Infrastructure) software implements the interface specification. It provides services in a manner that is comparable to the way a distributed operating system provides services to applications. A way to test whether is suitable for a standard, and all service was implemented is tested through two phases of processes to verify the RTI which proposed in DMSO. In this paper, we discuss Level One Test Procedures and a method. Confirms whether RTI was implemented in Interface Specification according to Level One Test Procedures appropriately through a test. Develops Test Suite for every each step to test whether a correct command and the expected results are occur.

### 키워드

HLA, RTI, Interface Specification, Level One Test Procedures

### 1. 서 론

HLA(High Level Architecture)는 미 국방성에서 제안한 분산 시뮬레이션을 위한 실시간 분산 데이터 공유를 위한 객체, 시간, 인터페이스 규격을 말한다. 이 HLA는 (1) Rules, (2) Interface Specification, (3) Object Model Template 등의 3가지 구성요소로 정의되며, 인터페이스 명세를 구현한 것이 RTI(Run-Time Infrastructure)이다. RTI는 분산 운영체계가 응용 프로그램에게 서비스를

제공하는 것과 유사한 방법으로 Federation을 구성하고 있는 Federate들에게 상호연동에 필요한 서비스를 제공한다.[1] 모든 서비스들이 표준에 적합하게 구현되었는지 테스트하기 위한 방법은 DMSO(Defense Modeling and Simulation Office)에서 제안한 RTI 검증방법을 통해 테스트된다.[2] RTI 검증은 표준과의 비교처리 과정으로서 일반 HLA 사용자 커뮤니티로 DMSO에 의해 제공

되는 서비스다. DMSO에 의해 검증된 RTIs를 선택한 사용자는 그 제품이 HLA 표준을 따르기 위해 엄격히 테스트되었다고 신뢰할 수 있다. 모든 서비스들이 표준에 적합하게 구현되었는지 테스트하기 위한 방법은 DMSO에서 제안한 RTI를 검증하기 위한 2단계 과정을 통해 테스트된다. 1단계는 RTI 개발자에 의해 그들 제품이 2단계 시험에 대한 준비가 되어 있는지 결정하기 위한 자가 진단을 할 수 있으며, 이 테스트는 HLA 인터페이스 명세서의 모든 중요한 영역에서 단순히 RTI의 능력을 보여 주고, 각 테스트는 단계별 명령과 예상되는 결과로 구성된다.[3][4] 2단계 시험은 RTI 검증 기능에 의해 제공될 수 있는데, 그것은 후보 RTI가 HLA 인터페이스 명세서에서 지정 받은 모든 요구 사항을 고수하는지 확인한다.

현재 DMSO에서는 1단계 시험에 대한 특별한 소프트웨어 없이 단지 테스트 절차에 대해서만 언급하고 있는데, 본 논문에서는 DMSO에서 제안한 1단계 시험을 구현하는데 목적이 있다. 이를 위하여 각 단계별로 테스트 절차를 작성하여 그 절차에 알맞게 서비스가 불러지는지 확인함으로써 테스트를 수행한다.

## II. 본 론

### 1. Level One Test Procedures

RTI가 Interface Specification에 적합하게 구현되었는지 테스트하기 위해서는 각 영역에서 서비스 호출에 대한 예측된 결과가 정확히 나오는지 확인함으로써 살펴볼 수 있다. 현재 DMSO는 1단계 시험에 대한 테스트 프로시저와 .FED 파일을 제공하고 있다. DMSO의 Level One Test Procedure는 각 영역을 8개로 나누고 각 호출 서비스에 대한 예측결과로써 구성 되어있다.[2]

1단계 테스트는 그들의 제품이 2단계 테스트에 대한 준비가 완료되었을 때 RTI 개발자에 의해 수행될 수 있다. RTI 1.3을 사용하는 모든 테스트들은 Federation Execution Data file(FED)을 사용하고, RTI 1516을 사용하는 모든 테스트들은 Federation Object Model Document Data(FDD)를 사용한다. 이 테스트를 위해 각 Federate은 하나의 문자로 지정된다: "A", "B". 두 종류의 문자가 테스트에서 발견된다: "행동", "관찰". "행동"은 각 Federate에서 서비스의 이름을 호출함으로써 행해지고, "관찰"은 서비스 호출로 발생할 수 있는 예상 결과를 관측하는 것이다. 서비스 호출에서 명백한 상황을 제외하고는 예외처리가 나와서는 안된다.

### 2. Joining Federations

하나의 Federation은 Federate이 Federation에 조인하기 전에 존재해야 하고 모든 Federate들은

Federation이 소멸되기 전에 하나의 Federation으로부터 탈퇴되어야 한다. Federate은 탈퇴하기 전에 하나의 Federation에 조인되어야 한다.[5][6]

#### Federate Step

1	A	Invoke Join Federation Execution("type1", "Verification")
2	A	Observe that the service invocation was unsuccessful giving exception "The specified federation execution does not exist"
3	B	Create Federation Execution("Verification", "LevelOne.fed")
4	A	Join Federation Execution("type1", "Verification")
5	B	Invoke Destroy Federation Execution("Verification")
6	B	Observe that the service invocation was unsuccessful giving exception "Federates are joined to the federation execution"
7	A	Invoke Resign Federation Execution(Delete Objects and Release Attributes)
8	B	Destroy Federation Execution("Verification")
9	B	Invoke Resign Federation Execution(Delete Objects and Release Attributes)
10	B	Observe that the service invocation was unsuccessful giving exception "The federate is not a federation execution member"

그림 1. Joining Federations

그림 1은 DMSO의 Level One Test Procedures의 Joining Federation에 관련된 내용이다. Federation Management는 Federation 생성, Federate 조인, Federation으로부터의 Federate의 탈퇴 및 Federation의 소멸 등에 관한 RTIambassador 서비스 호출에 대한 내용이다.

```

C:\WINDOWS\System32\cmd.exe
C:\Program Files\RTI_Lib\sample\Chat>java Join
***ERR: federation does not exist
***FED_HU: created federation execution
***FED_HU: joined federation execution
***ERR: federates still joined -- not destroying fedex
***FED_HU: resigned from federation
***FED_HU: destroyed federation execution
***ERR: Federate Not Execution Member

C:\Program Files\RTI_Lib\sample\Chat>
    
```

그림 2. Joining Federations 실행 결과

그림 2에서 보는 바와 같이 그림 1에서의 테스트 절차에 따라 올바른 실행 결과가 나온다는 것을 알 수 있다. 처음의 에러는 Federation이 생성되기 전에 조인을 시도함으로써 이미 예상된 결과가 나온다는 것을 볼 수 있다. 이처럼 모든 테스트에서 서비스 호출에 대한 예상된 결과를 관찰함으로써 테스트를 수행하게 된다.

### 3. Synchronization Points

동기화 라벨이 등록되면 모든 조인된 Federate들에게 알려지게 된다. 라벨에 대한 Federation 동기화는 라벨을 획득한 Federate이 마지막 조인된 이후에 발생하게 된다.

Federate	Step
1 A	Invoke Create Federation Execution("Verification", "LevelOne fed")
2 A,B	Invoke Join Federation Execution("type1", "Verification")
3 A	Invoke Register Federation Synchronization Point("lunch", "theTag")
4 A	Observe a successful Confirm Synchronization Point Registration("lunch")
5 A,B	Observe Announce Synchronization Point("lunch", "theTag")
6 A	Invoke Synchronization Point Achieved("lunch")
7 B	Invoke Synchronization Point Achieved("lunch")
8 A,B	Observe Federation Synchronized("lunch")
9 A,B	Invoke Resign Federation Execution(Delete Objects and Release Attributes)
10 A	Invoke Destroy Federation Execution("Verification")

그림 3. Synchronization Points

그림 1에서의 호출은 FederateAmbassador의 콜백(callback)이 없는 RTIambassador 서비스 호출에 대한 것이었다. 그러나 그림 3의 호출은 콜백 함수를 호출하는 RTIambassador 서비스 호출이다.

```
C:\WINDOWS\system32\cmd.exe
C:\Program Files\hpRTI_LEB\anples\Chat>java Syn
***FED_HU: created federation execution
***FED_HU: A joined federation execution
***FED_HU: B joined federation execution
****FED_HU****: synchronization Point Registration success
****FED_HU****: announce Synchronization Point =>lunch, theTag
****FED_HU****: announce Synchronization Point =>lunch, theTag
****FED_HU****: Federation Synchronized =>lunch
***FED_HU: A resigned from federation
***FED_HU: B resigned from federation
***FED_HU: destroyed federation execution
C:\Program Files\hpRTI_LEB\anples\Chat>
```

그림 4. Synchronization Points 실행 결과

#### 4. Save and Restore

Federation save 라벨은 즉시 모든 조인된 Federate들에게 제공되어야 한다. Federation save는 각자의 save가 성공적으로 완성됐다고 양쪽의 Federate에게 알려줄 때 선언된다. Federation save가 일단 성공적으로 완료되면, Federation 복귀에서도 사용될 수 있다.

Federate	Step
1 A	Invoke Create Federation Execution("Verification", "LevelOne fed")
2 A,B	Invoke Join Federation Execution("type1", "Verification")
3 A	Invoke Request Federation Save("theSave")
4 A,B	Observe Initiate Federate Save("theSave")
5 A,B	Invoke Federate Save Begun()
6 A,B	Invoke Federate Save Complete(Success)
7 A,B	Observe Federation Saved(Success)
8 B	Invoke Request Federation Restore("theSave")
9 B	Observe Confirm Federation Restoration Request("theSave", Success)
10 A,B	Observe Federation Restore Begun() and then Initiate Federate Restore("theSave", federateHandle)
11 A,B	Invoke Federate Restore Complete(Success)
12 A,B	Observe Federation Restored(Success)
13 A,B	Invoke Resign Federation Execution(Delete Objects and Release Attributes)
14 A	Invoke Destroy Federation Execution("Verification")

그림 5. Save and Restore

위의 그림 5는 Federation의 저장과 복귀에 대한 테스트 절차를 기술한 내용이다. 위의 절차에 맞게 모든 서비스가 불러지고 예측된 결과가 나온다면 테스트는 성공이다.

```
C:\WINDOWS\system32\cmd.exe
C:\Program Files\hpRTI_LEB\anples\Chat>java RandS
***FED_HU: created federation execution
***FED_HU: A joined federation execution
***FED_HU: B joined federation execution
****FED_HU****: initiate Federate Save =>theSave
****FED_HU****: initiate Federate Save =>theSave
****FED_HU****: Federation Save Success
****FED_HU****: request Federation Restore Succeeded theSave
****FED_HU****: Federation Restore Begun
****FED_HU****: Federation Restore Begun
****FED_HU****: initiate Federate Restore
****FED_HU****: initiate Federate Restore
****FED_HU****: Federation Restore Success
****FED_HU****: Federation Restore Success
***FED_HU: A resigned from federation
***FED_HU: B resigned from federation
***FED_HU: destroyed federation execution
```

그림 6. Save and Restore 실행 결과

#### 5. Object Discovery

Federate	Step
1 A	Invoke Create Federation Execution("Verification", "LevelOne fed")
2 A,B	Invoke Join Federation Execution("type1", "Verification")
3 A	Invoke Get Object Class Handle(objectRootA,B) Keep the returned object class handle and refer to it as objectClassB
4 A	Invoke Get Attribute Handle("X", objectClassB) Keep the returned class attribute handle and refer to it as attributeX
5 A	Invoke Get Attribute Handle("Y", objectClassB) Keep the returned class attribute handle and refer to it as attributeY
6 A	Invoke Publish Object Class(objectClassB, (attributeX, attributeY))
7 B	Invoke Get Object Class Handle(objectRootA) Keep the returned object class handle and refer to it as objectClassA
8 B	Invoke Subscribe Object Class Attribute(objectClassA, (attributeX))
9 A	Observe Start Registration for Object Class(objectClassB)
10 B	Invoke Register Object Instance(objectClassA)
11 B	Observe that the service invocation was unsuccessful giving exception "The federate is not publishing the specified object class"
12 A	Invoke Register Object Instance(objectClassB)
13 B	Observe Discover Object Instance(objectClassA)
14 A,B	Invoke Resign Federation Execution(Delete Objects and Release Attributes)
15 A	Invoke Destroy Federation Execution("Verification")

그림 7. Object Discovery

Federate은 published된 클래스를 다른 Federate이 subscribes할 때 객체 인스턴스의 등록을 시작하라고 요청한다. 그림 7의 실행 결과는 아래와 같다.

```
C:\WINDOWS\system32\cmd.exe
C:\Program Files\hpRTI_LEB\anples\Chat>java ObjectDis
***FED_HU: created federation execution
***FED_HU: A joined federation execution
***FED_HU: B joined federation execution
****FED_HU****: start Registration For Object Class =>6
***ERR: => hla.rti.ObjectClassNotPublished: RegisterObject
s not published serial:909087001
****FED_HU****: discover Object Instances
***FED_HU: A resigned from federation
***FED_HU: B resigned from federation
***FED_HU: destroyed federation execution
```

그림 8. Object Discovery 실행 결과

## 6. Time Advancement

regulating이나 constrained를 갖는 Federate들은 그들의 로컬 타임 진행을 허가할 수 있다.

Federate	Step
1 A	Invoke Create Federation Execution("Verification", "LevelOne.fed")
2 A, B	Invoke Join Federation Execution("Type1", "Verification")
3 A, B	Invoke Enable Time Regulation (0, 2)
4 A, B	Observe Time Regulation Enabled
5 A, B	Invoke Enable Time Constrained
6 A, B	Observe Time Constrained Enabled
7 A	Invoke Time Advance Request(5)
8 B	Invoke Time Advance Request(1)
9 B	Observe Time Advance Grant(1)
10 B	Invoke Time Advance Request(8)
11 A	Observe Time Advance Grant(5)
12 A	Invoke Time Advance Request(9)
13 B	Observe Time Advance Grant(8)
14 A	Observe Time Advance Grant(9)
15 A, B	Invoke Resign Federation Execution(Delete Objects and Release Attributes)
16 A	Invoke Destroy Federation Execution("Verification")

그림 9. Time Advancement

```

C:\Program Files\RTI_LEW5\examples\Chat>java TimeAdv
***FED_HU: created federation execution
***FED_HU: A joined federation execution
***FED_HU: B joined federation execution
***FED_HU****: time Regulation Enabled
***FED_HU****: time Regulation Enabled
***FED_HU****: time Constrained Enabled
***FED_HU****: time Constrained Enabled
***FED_HU****: time Advance Grant
***FED_HU****: time Advance Grant
***FED_HU****: time Advance Grant
***FED_HU****: time Advance Grant
***FED_HU: A resigned from federation
***FED_HU: B resigned from federation
***FED_HU: destroyed federation execution
    
```

그림 10. Time Advancement 실행 결과

## 7. 그 밖의 Test Procedures

이 밖에도 에트리뷰트의 소유권에 관한 Attribute Divestiture and Acquisition, 지역에 관한 Region Intersection과 Management Object Model등에 관한 테스트 절차도 있다.

### III. 결 론

본 논문에서는 RTI의 모든 서비스들이 표준에 적합하게 구현되었는지 테스트하기 위한 방법으로 DMSO에서 제안한 RTI를 검증하기 위한 2단계 과정 중 1단계 테스트를 구현해 봤다. 1단계 테스트는 2단계 테스트로 가기 위한 RTI 개발자들에게 의해 수행되는 중간 단계라 말할 수 있다. 현재 1단계 테스트에 대한 어떠한 소프트웨어도 나와있지 않지만 본 논문에서 기술된 프로그램을 통해 적합한 수행 결과를 얻는다면 1단계 테스트를 성공적으로 통과했다고 말할 수 있다. 그렇지만 HLA Compliant로서 인증 받기 위해서는 2단계 테스트를 통과해야만 인증을 받을 수 있다. 그

렇게 하기 위해 차후 2단계 테스트에 대한 테스트 절차와 Test Suite개발에 대한 연구가 더 필요하다.

### 참고문헌

- [1] 장상철외 3명. 국방모의분석체계 구축을 위한 상위체계구조[HLA] 기술 연구, 한국국방연구원, 1999, 10
- [2] Level One Test Procedure Version 1.0(<http://dmsomil/public/transition/hla/levelonetesting/procedures>)
- [3] Test Procedures for High Level Architecture Interface Specification, Version 1.3, May 1998
- [4] Defense Modeling and Simulation Office, "HLA Compliance Checklist, Version 1.3," April 1998
- [5] Defense Modeling and Simulation Office, "High Level Architecture Interface Specification, Version 1.3," 20 April 1998
- [6] Defense Modeling and Simulation Office, "RTI 1.3-Next Generation Programmer's Guide Version 3.2" 7 September 2000