

# 에이전트 기반 IEEE1516 페더레이션 연동 환경에서의 소유권 관리

\*유민욱, 김탁곤

KAIST 전기 및 전자공학과

e-mail : *mwyu@smslab.kaist.ac.kr, tkim@ee.kaist.ac.kr*

## Ownership Management for Agent-based Interoperation of IEEE1516 Federations

\*Min Wook Yoo, Tag Gon Kim

Department of Electrical Engineering

KAIST

### Abstract

HLA(High Level Architecture) is IEEE 1516 standard for interoperation of distributed heterogeneous simulators. Agent-based interoperation of federations has been studied to extend HLA to federation level without modifying of RTI(Run-Time Infrastructure). This paper proposes ownership management protocol for interoperation of two IEEE1516 federations. Racing condition problem is solved by proposed protocols and ownership management can be applied to interoperation of federations without violating HLA.

### I. 서론

HLA(High Level Architecture)는 다양한 환경에서 개발되는 시뮬레이터들의 연동을 위한 구조로 IEEE1516 표준으로 지정되어있다[1]. RTI(Run-Time Infrastructure)는 HLA에 명시되어 있는 서비스를 시뮬레이터들에 제공하는 미들웨어이다. HLA는 연동 시뮬레이션에 참여하는 개별 시뮬레이터들을 페더레이트로 정의하고 통합 시뮬레이션을 수행하기 위해서는 페더레이트들이 RTI 상에 하나의 페더레이션을 구성하

도록 명시하고 있다.

페더레이션의 연동은 보안 등의 이유로 새로운 페더레이션을 구성하지 않고 기존 페더레이션들을 유지하면서 통합 시뮬레이션을 수행하기 위해 제안되어 연구되어왔다. 에이전트 기반 페더레이션 연동 구조는 페더레이션의 동작을 대표하는 대리자를 사용하는 방법으로 기존의 RTI를 수정 없이 사용할 수 있도록 제안된 구조이다[2][3]. 페더레이션 연동 구조에서 각 페더레이션에는 서로 정보를 공유하기 위하여 동일한 객체가 존재하게 된다. 페더레이션 연동을 위해서는 두 페더레이션에 존재하는 동일한 객체의 소유상태가 HLA 규칙에 어긋나지 않도록 유지해야만 한다.

본 논문에서는 에이전트 기반 페더레이션 연동 환경에서 소유권 관리 프로토콜을 제안한다. HLA의 소유권 관리 프로토콜이 페더레이션 연동 환경으로 확장되어 적용되면서 발생하는 문제점을 확인하고 이를 해결할 수 있는 방안을 연구하였다. 이를 통해 두 개의 페더레이션 연동 환경에서도 하나의 페더레이션과 마찬가지로 소유권 관리가 될 수 있도록 하였다.

### II. Background

#### 2.1 에이전트 기반 페더레이션 연동 구조

그림 1은 에이전트 기반 페더레이션 연동 구조와 연동 과정을 보여준다. 두 페더레이션에는 각각 다른 페

더레이션을 대표하는 대리자가 참여한다. 대리자는 참여한 페더레이션의 정보를 모니터링하고 페더레이션의 상태가 변하여 동작 대행이 필요한 경우 다른 대리자로 필요 정보를 전달하여 적합한 서비스를 호출하도록 한다.

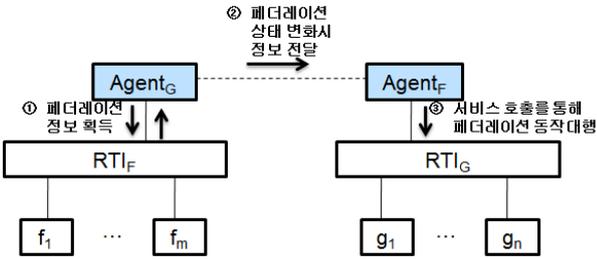


그림 1. 에이전트 기반 페더레이션 연동 구조

2.2 IEEE1516의 소유권 관리 서비스

IEEE1516의 소유권 관리 서비스는 페더레이션의 공유 객체들의 소유권 방출/이전/획득 등에 필요한 서비스를 정의하고 있다. 소유권 관리 서비스는 크게 3가지 프로토콜로 구분할 수 있다.

무조건 방출 프로토콜은 소유자가 무조건적으로 소유권을 방출하고 다른 페더레이트가 소유권을 원하는 시점에 획득하는 프로토콜이다. 소유자가 무조건 방출하는 시점부터 해당 객체의 소유권은 무소유 상태가 되고 다른 페더레이트로 그 정보를 알려주게 된다. 무소유 상태에서 소유권 획득 요청이 있을 경우 RTI는 요청 페더레이트로 소유권을 넘겨주게 된다.

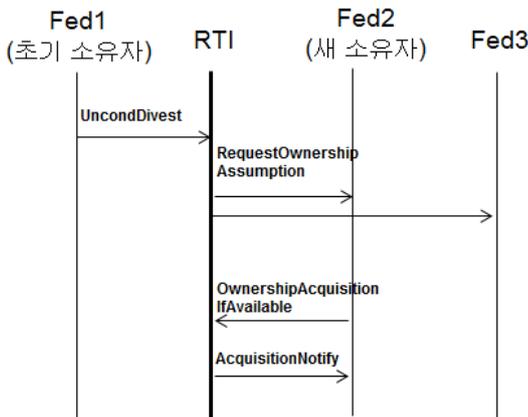


그림 2. 무조건 방출 프로토콜

협상이전 프로토콜은 먼저 소유자가 다른 페더레이트들에 소유권 이전 의사를 밝히고 새로운 소유자가 나타날 경우, 해당 소유권을 넘겨주는 프로토콜이다. 소유자가 소유권 이전 의사를 전달하면 RTI는 다른 페더레이트들에 소유권 획득의사를 묻게 된다. 획득하고자 하는 페더레이트가 있을 경우 소유자에게 방출 확인을 받은 이후 소유권을 넘겨주게 된다. 아무도 확

득하고자 하지 않거나 소유자가 방출을 취소하게 되면 소유자가 계속 해당 소유권을 갖고 있게 된다.

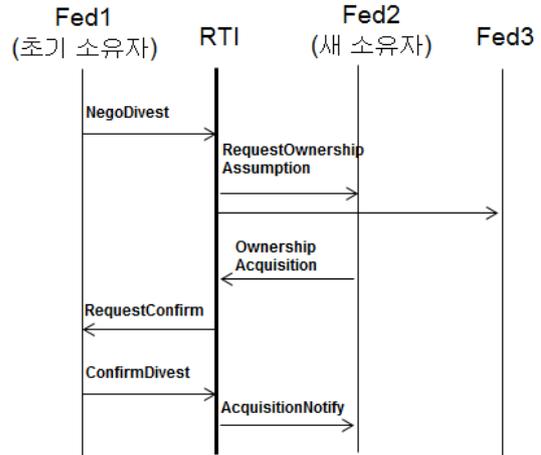


그림 3. 협상 이전 프로토콜

요청 이전 프로토콜은 페더레이트가 소유자에게 소유권 방출을 요청하여 소유권을 이전하는 프로토콜이 있다. 소유권을 갖고 있지 않은 페더레이트는 객체의 소유권을 요청할 수 있다. RTI를 통해 요청을 받은 소유자가 소유권 방출 의사를 밝히게 되면 요청한 페더레이트로 소유권이 넘어가게 된다.

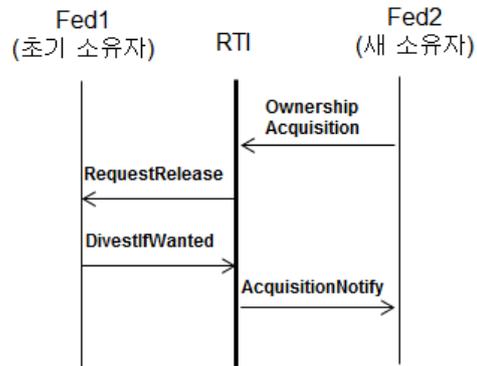


그림 4. 요청 이전 프로토콜

III. 소유권 관리 프로토콜

3.1 페더레이션 연동 프로토콜

에이전트 기반 페더레이션 연동 환경에서는 대리자가 페더레이션의 동작을 대표하기 위해 원 객체의 복제 객체를 만들어서 사용하게 된다. 페더레이션 연동 환경에서는 대리자의 소유권과 대리자가 대표하는 페더레이션의 사용자 페더레이트들의 소유권의 합집합이 동일해야만 한다. 초기 상태에서는 객체를 등록한 페더레이트가 소유권을 갖기 때문에 해당 조건을 만족하게 된다. 따라서 페더레이션 연동에서는 소유권이 이

전이 발생할 경우에도 해당 조건을 만족하도록 대리자의 소유권을 관리해야 한다. 다른 페더레이션으로 소유권이 넘어가거나 무소유 상태가 되는 경우에는 대리자의 소유권 변화가 필요하게 되고 페더레이션 내에서 소유권 이전이 일어나는 경우에는 대리자의 소유권은 그대로 유지하게 된다.

무조건 방출 프로토콜에서는 소유권을 방출한 시점에서는 해당 객체가 무소유 상태가 되어야 한다. 따라서 대리자는 다른 페더레이션에서도 해당 객체가 무소유 상태가 되도록 방출한다. 방출 이후에는 각 페더레이트는 언제든지 소유권을 획득할 수 있는 상태가 되는데, RTI는 페더레이트 소유권을 획득하더라도 다른 페더레이트에 알려주지 않는다. 따라서 대리자는 주기적으로 무소유 객체의 소유권을 조회하고 사용자 페더레이트가 소유권을 획득하게 되면 다른 페더레이션에서 대리자가 객체의 소유권을 획득하여 양 쪽의 상태를 동일하도록 맞춰준다.

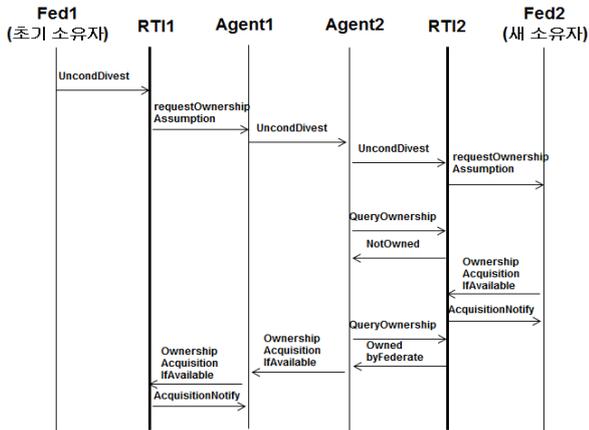


그림 5. 페더레이션 연동에서의 무조건 방출 프로토콜

협상 이전 프로토콜에서는 소유자가 소유권 방출의사를 밝히게 되면 두 페더레이션의 모든 페더레이트들에게 소유권 획득 의사가 있는지를 확인해야 한다. 대리자는 해당 정보를 RTI로 받게 되면 다른 페더레이션에서도 협상 방출을 요청하여 그 곳에 속한 페더레이트들에게도 소유권 획득 의사를 확인하도록 한다. 사용자 페더레이트가 소유권 획득을 요청해서 대리자에게 이전 확인을 요청하게 되면 다른 페더레이션에서도 대리자가 해당 객체의 소유권을 요청한다. 대리자가 소유권을 얻게 되면 이전 확인을 승인하고 얻지 못하면 이전 확인을 취소하여 양쪽의 상태를 동일하게 맞춰준다.

요청 이전 프로토콜에서는 사용자 페더레이트가 다른 페더레이션이 소유하고 있는 소유권 획득을 요청하게 되면 대리자로 소유권 방출 요청이 전달된다. 대리자는 반대편 페더레이션에서도 해당 소유권을 요청하

여 실 소유하고 있는 사용자 페더레이트에게 방출 요청을 하게 된다. 소유자가 이전을 승인하면 대리자가 소유권을 얻게 되고 대리자도 이전을 승인하여 다른 페더레이션에서 요청한 페더레이트로 소유권이 이전된다.

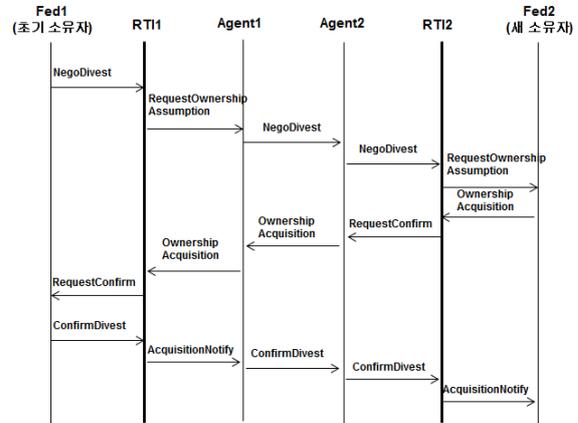


그림 6. 페더레이션 연동에서의 협상 이전 프로토콜

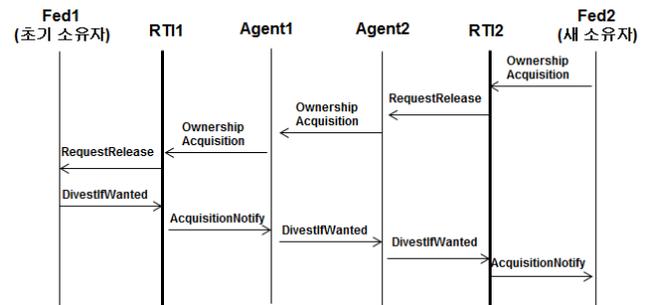


그림 7. 페더레이션 연동에서의 요청 이전 프로토콜

### 3.2 페더레이션 연동 시의 문제점

페더레이션 연동에서는 같은 객체가 두 페더레이션에 동시에 존재하고 서로 다른 RTI가 각 객체를 관리하기 때문에 객체 소유권에 대한 소유권 경합 문제가 발생할 수 있다[2]. 한 페더레이션 내에서는 RTI가 해당 문제를 처리하지만 다른 페더레이션간의 경합 문제는 각 객체를 다른 RTI가 관리하기 때문에 처리가 불가능하다. 무조건 방출 프로토콜과 협상 방출 프로토콜에서 소유권 경합 문제가 발생할 수 있다.

무조건 방출 프로토콜에서는 소유자가 소유권을 방출한 이후에는 해당 객체와 복제 객체는 무소유 상태가 된다. 이 상태에서 다른 페더레이션에 속한 두 개 이상의 사용자 페더레이트가 무소유 객체의 소유권을 얻고자 할 때, 문제가 발생할 수 있다. 협상 이전 프로토콜에서는 소유자가 협상 방출을 요청하게 되면 다른 페더레이트들에게 소유권 획득 의사를 묻게 된다. 이 경우에도 다른 페더레이션에 속한 두 개 이상의 사용자 페더레이션이 요청 객체의 소유권을 얻고자할 경우

경합 문제가 발생하게 된다.

### 3.3 페더레이션 연동 문제 해결 방안

무조건 방출 프로토콜에서의 두 사용자 페더레이트가 동시에 같은 객체의 소유권을 획득하였을 때, 한 페더레이트의 소유권을 강제로 방출시킴으로써 문제를 해결할 수 있다. HLA에 명시되어 있는 MOM 서비스를 이용하게 되면 다른 페더레이트가 특정 서비스를 호출하도록 제어할 수 있다. 그림 8은 연동 환경에서의 소유권 경합 문제 해결 프로토콜을 보여준다. 초기 상태에서 두 사용자 페더레이트가 동시에 해당 객체의 소유권을 얻음으로써 문제가 발생한다. 대리자가 페더레이션의 동작을 대표하기 위해 소유권을 얻어야만 하지만 다른 페더레이트로 소유권이 넘어가서 소유권을 얻지 못한다. 이 시점에서 대리자는 문제를 식별하게 되고 MOM 서비스를 이용하여 해당 소유권을 강제로 방출시켜 무소유 상태로 만들고 해당 소유권을 얻도록 하여 문제를 해결하도록 한다.

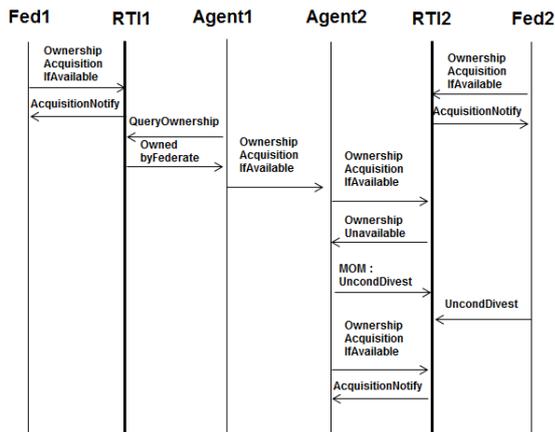


그림 8. 무조건 방출 프로토콜의 경합문제 해결

협상 방출 프로토콜에서는 대리자가 마지막으로 방출을 확인하는 시점에서 문제를 확인할 수 있기 때문에 방출을 취소함으로써 문제를 해결할 수 있다. 그림 9는 협상 방출 프로토콜에서의 경합 문제 해결 과정을 보여준다. 소유권 방출 의사를 모든 사용자 페더레이트에게 물었을 때 서로 다른 페더레이션에 속한 두 개 이상의 사용자 페더레이트(Fed2, Fed3)가 소유권을 얻고자 하고 있다. 좌측 페더레이션에서 RT1은 사용자 페더레이트와 대리자로부터 소유권 요청을 받게 된다. 대리자가 선택 받게 되면 그림 6과 같이 기존 프로토콜대로 진행되면서 Fed2로 소유권 획득 실패를 알려주게 되고 대리자가 선택 받지 못하게 되면 반대편의 대리자가 소유권 방출을 취소함으로써 두 사용자 페더레이트가 동시에 소유권을 갖지 못하도록 한다.

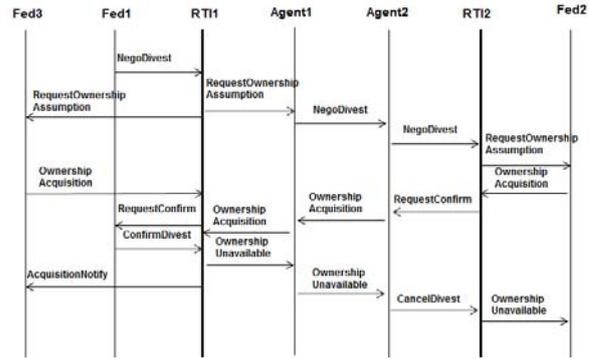


그림 9. 협상 방출 프로토콜의 경합 문제 해결

## IV. 결론

본 논문에서는 에이전트 기반 페더레이션 연동 환경에서의 소유권 관리 프로토콜에 대하여 제안하였다. 페더레이션 연동 환경에서 발생할 수 있는 소유권 관리시 발생할 수 있는 문제를 확인하고 이를 해결할 수 있는 방안을 제시하였다. 제안된 프로토콜에서 대리자는 스스로 문제를 식별하여 해결하여 사용자 페더레이트들은 추가적인 구현 없이 사용될 수 있도록 하였기 때문에 사용자 페더레이트는 페더레이션 연동 환경에 적용하는데 도움을 줄 수 있을 것이다.

## Acknowledgment

본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다. (UD11006MD)

## 참고문헌

- [1] IEEE Std. 1516-2010, IEEE Standard for Modeling and Simulation(M&S) High Level Architecture(HLA) - Framework and Rules, IEEE Computer Society, 2010.
- [2] J. Dingel, D. Garlan, and C. Damon, "A feasibility study of the HLA bridge," *Technical Report*, Department of Computer Science, Carnegie Mellon University, CMU-CS-01-103, 2001.
- [3] Min Wook Yoo and Tag Gon Kim, "Design and Implementations of Surrogates for Interoperation of HLA Federations," *European Simulation Interoperability Workshop*, Istanbul, Turkey, pp. 122 - 130, Jul. 2009.