

# 대규모 공동 시뮬레이션을 위한 정보 공유 전략: 공유 메모리와 DDS 평가

정우성, 김정식, 강세훈, 유대승

한국전자통신연구원

woosung@etri.re.kr, j.s.kim@etri.re.kr, skang@etri.re.kr ooseyds@etri.re.kr

## Communication Strategies for Large-Scale Co-Simulation: Evaluation of Shared Memory and DDS

Woo-Sung Jung, Jeongsik Kim, Saehoon Kang, and Dae Seung Yoo\*

Electronics and Telecommunications Research Institute

### 요약

FMI 기반의 Co-Simulation은 복잡한 물리 시스템의 모델링과 검증을 위해 필수적인 도구로 활용되고 있으며, 이 과정에서 빈번한 데이터 교환은 성능과 실시간성에 큰 영향을 미친다. 본 연구는 대표적인 데이터 교환 기법인 공유 메모리 방식과 DDS(Data Distribution Service) 방식을 비교하여 그 특성과 적용 환경을 분석한다. 공유 메모리 방식은 단일 머신 내에서 초저 지연·고처리량을 제공하여 대용량 주기적 데이터 교환에 적합하다. 반면, DDS는 분산 환경에서 확장성과 상호 운용성을 보장하며, QoS 정책을 통해 다양한 실시간 요구사항을 충족할 수 있다. 본 연구는 두 접근 방식의 장단점을 비교하여, 환경 특성에 따른 최적의 데이터 교환 전략을 제시하고 향후 안정적이고 효율적인 정보 교환 기법의 필요성을 논의한다.

### I. 서론

복합 시뮬레이션은 구현하고자 하는 대상의 복잡성이 증가함에 따라, 여러 도메인에 걸친 시뮬레이터들을 연동하는 Co-Simulation의 활용이 급격히 확대되고 있다. 이러한 환경에서는 물리 모델, 제어 알고리즘, 네트워크 시뮬레이터 등이 상호 작용하며 전체 시스템의 모습을 재현한다. 특히, FMI(Funational Mock-up Interface) 표준[1]의 확산은 서로 다른 툴과 플랫폼 간 모델의 상호 운용성을 높이고 있다. 따라서, 다양한 분야에서 Co-Simulation은 단순한 연구 도구를 넘어 실제 시스템 설계와 검증을 위한 핵심 인프라로 자리매김하고 있다.[2]

그러나 Co-Simulation 환경에서는 시뮬레이터 간 빈번하고 대용량의 데이터 교환이 필수적이다. 다수의 시뮬레이터가 동시에 연동될 경우, 데이터 교환 과정에서 발생하는 지연(latency)과 데이터 처리량(throughput) 문제는 전체 시뮬레이션 성능을 크게 저하시킬 수 있다. 특히, 분산 환경에서는 네트워크 스택을 통한 데이터 직렬화와 전송으로 인한 오버헤드가 누적되어 실시간성이 위협받는다. 반대로 단일 머신 환경에서는 높은 빈도로 교환되는 데이터를 효율적으로 처리하지 못하면 성능 병목이 발생한다. 따라서 데이터 교환 방식의 선택은 Co-Simulation 성능을 결정짓는 중요한 요소가 된다.

이러한 문제를 해결하기 위해 다양한 데이터 교환 기법이 연구되고 있으며, 그중 대표적인 방식이 공유 메모리(Shared Memory)[3]와 DDS(Data Distribution Service)[4]이다. 공유 메모리 방식은 단일 머신 내부에서 초저지연·고처리량 통신을 가능하게 하며, DDS는 표준화된 Pub/Sub 미들웨어로서 분산 환경에서 확장성과 실시간성을 보장한다. 본 연구에서는 두 방식의 구조와 특징을 분석하고, 환경적 조건에 따라 어떤 방식이 유리한지를 비교한다. 이를 통해 Co-Simulation의 효율적 데이터 교환 전략을 제안하고자 한다.

### II. 본론

#### 1) 공유 메모리 기반의 접근 방법

공유 메모리 방식은 동일 머신 내에서 실행되는 여러 FMU 간 데이터 교환을 위해 운영체제가 제공하는 공유 메모리 영역(shared memory segment)을 활용한다. 각 FMU는 이 영역에 직접 접근하여 데이터를 기록하거나 읽으며, 접근 충돌을 방지하기 위해 뮤텍스(mutex), 세마포어(semaphore) 등의 동기화 메커니즘이 함께 사용된다. 그림 1은 공유 메모리 기반 구조의 개요를 보여주며, Producer FMU는 공유 메모리 버퍼에 데이터를 기록하고 Consumer FMU는 이를 읽어 가는 형태로 동작한다.

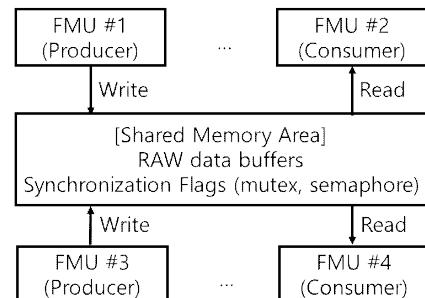


그림 1 공유 메모리 방식의 대표 구조

데이터 교환 과정은 일반적으로 쓰기(write) → 동기화 → 읽기(read) 단계로 구분된다. 먼저 Producer FMU는 버퍼에 데이터를 기록하고, 기록 완료 플래그를 통해 Consumer에게 준비 완료를 알린다. 이후 Consumer FMU는 해당 버퍼를 읽고, 읽기 완료 신호를 다시 Producer에게 전달한다. 이러한 방식은 메모리 복사를 최소화하고, Zero-Copy 기법을 적용할 경우 대용량 데이터도 초저지연으로 교환할 수 있다. 단일 머신 내부에서는 네트워크 프로토콜을 거치지 않기 때문에 매우 높은 처리량과 낮은 지연을 보장한다.

공유 메모리 기반 접근 방식의 가장 큰 장점은 절대적인 성능이다. 특히 영상, LiDAR 포인트 클라우드, 고속 센서 데이터 등 대량의 주기적 데이터 전송에 최적화되어 있다. 반면, 확장성 측면에서는 한계가 존재한다. 공유 메모리는 단일 머신 환경에 국한되므로, 확장성을 지원하는 네트워크 기반의 분산 시뮬레이션에는 적용이 어렵다. 따라서 공유 메모리는 HILS (Hardware-in-the-Loop Simulation), 로컬 환경의 병렬 시뮬레이션, 또는 단일 노드 기반 고성능 테스트베드에서 가장 효과적인 접근 방식이다.

## 2) DDS 기반의 접근 방식

DDS(Data Distribution Service)는 OMG(Object Management Group) 표준에 기반한 발행/구독(Publish/Subscribe) 방식의 미들웨어이다. DDS 도메인(Domain) 내의 참여자(Participant)는 데이터 작성자(Publisher)와 데이터 독자(Subscriber)를 통해 특정 토픽(Topic) 단위로 데이터를 발행하거나 구독한다. 그럼 2는 DDS 기반 구조를 나타내며, Publisher FMU는 데이터를 DDS에 발행하고 Subscriber FMU는 이를 구독 수신한다.

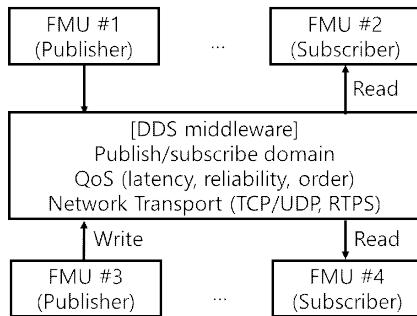


그림 2 DDS 방식의 대표 구조

데이터 교환은 다음 단계로 진행된다. Publisher FMU는 특정 Topic에 데이터를 발행(Publish)하고, DDS 미들웨어는 이를 RTPS(Real-Time Publish-Subscribe) 프로토콜을 통해 전송한다. Subscriber FMU는 동일 Topic을 구독(Subscribe)함으로써 데이터를 수신한다. 이 과정에서 DDS는 다양한 QoS(Quality of Service) 정책을 적용할 수 있으며, 지연 시간 제한, 데이터 신뢰성 보장, 전달 순서 보장 등을 설정 가능하다. 이를 통해 DDS는 단순한 데이터 전달을 넘어 실시간성과 예측 가능성을 보장한다.

DDS의 주요 장점은 분산 환경에서의 확장성과 유연한 QoS 지원이다. 따라서 이기종 노드가 네트워크로 연결되는 대규모 분산 Co-Simulation 환경에서 장점을 가진다. 특히 자율주행 시뮬레이션, 스마트 그리드 등 다양한 실시간 분산 응용 분야에서 DDS는 사실상의 표준으로 자리잡고 있다. 그러나 DDS는 데이터 직렬화와 복사에 따른 오버헤드가 발생하며, 단일 머신 내에서는 공유 메모리보다 성능이 떨어진다.

## 3) 두 방식의 비교와 환경별 적합성

앞서 설명한 두 접근 방식은 적용 환경과 요구 사항에 따라 상반된 장점을 보인다. 공유 메모리는 단일 머신 내 초저지연·고처리량 데이터 교환에서 최적의 성능을 제공하며, DDS는 분산 환경에서 실시간성과 상호 운용성을 보장한다. 따라서 두 방식은 대체 관계가 아니라, 환경에 따른 선택적 적용 또는 상호 보완적 활용이 필요하다.

단일 머신 기반의 시뮬레이션에서는 공유 메모리가 성능상 유리하며, 특히 고빈도 대용량 데이터가 교환되는 경우 필수적이다. 반대로 다수의 이기종 노드가 참여하는 분산 시뮬레이션에서는 DDS가 적합하다. 또한 최근 연구에서는 DDS 구현체에 공유 메모리 옵션을 결합하여 동일 머신 내 통신 성능을 극대화하는 시도가 이루어지고 있다. 이는 두 방식이 경쟁적이면서도 보완적인 관계임을 보여준다. 표 1은 공유 메모리 방식과 DDS 방식을 성능, 확장성 등 주요 지표별로 비교한 것이다.

표 1 공유 메모리와 DDS의 접근 방식 비교

구분	공유 메모리	DDS
통신 범위	동일 머신 내부	로컬/분산 네트워크
지연·처리량	매우 낮음/ 매우 높음	상대적 높음/제한적
구현 난이도	OS 종속, 동기화 필요	미들웨어 설치 및 QoS 설정 필요
확장성	낮은(단일 노드 한정)	높은(이기종, 멀티지원)
적합 사례	HILS, 대용량 데이터	분산 CPS, 자율주행 등
핵심 강점	Zero-copy 초저지연	QoS 기반 확장, 신뢰성

## III. 결론

본 연구에서는 Co-Simulation 환경에서의 대용량 데이터 교환을 위해 대표적으로 활용되는 공유 메모리 방식과 DDS 방식을 비교하였다. 공유 메모리는 단일 머신 내에서 초저지연·고처리량을 제공하는 반면, DDS는 분산 환경에서 확장성과 실시간성을 보장한다. 따라서 환경 특성에 따라 적합한 방식을 선택하는 것이 필수적이다. 이를 바탕으로 다음과 같은 향후 연구 방향을 제시할 수 있다.

첫째, 공유 메모리와 DDS를 결합한 하이브리드 프레임워크의 개발이 필요하다. 이는 로컬 환경에서는 초저지연을, 분산 환경에서는 확장성을 동시에 확보할 수 있는 유연한 접근 방식을 제공할 수 있다. 둘째, DDS QoS 정책을 활용한 실시간 보장 기법에 대한 심층 연구가 요구된다. 이를 통해 분산 Co-Simulation 환경에서도 지연 상한(latency bound)과 신뢰성을 예측 가능하게 설정할 수 있다. 셋째, AI 기반 데이터 교환 최적화 기법을 통하여 데이터 전송 빈도, 압축 방식, 전송 우선순위를 동적으로 조절함으로써 전체 성능을 극대화할 수 있다. 넷째, Zero-Copy 공유 메모리 기법의 성능 분석이 필요하다. 특히 고주파수·대용량 데이터 교환 시의 한계를 실험적으로 검증하고, 병목을 완화할 수 있는 개선 방안을 탐구해야 한다. 마지막으로, 실제 응용 분야로의 확장 연구가 요구된다. 예를 들어, 자율주행 HILS, 스마트 그리드 시뮬레이션, 디지털 트윈과 같은 응용 사례에 본 연구에서 논의한 기법을 적용하고 검증하는 작업이 필요하다. 이러한 후속 연구들은 Co-Simulation 환경에서 안정적이고 효율적인 데이터 교환을 실현하는 데 중요한 기반이 될 것이다.

## ACKNOWLEDGMENT

본 논문은 2025년도 해양수산부 재원으로 해양수산과학기술진흥원의 지원을 받아 수행된 연구임(20220531, 시뮬레이션 평가기술 개발).

## 참 고 문 헌

- [1] Functional Mock-up Interface, <https://fmi-standard.org/> (accessed at: Sep. 29th, 2025).
- [2] J. Kim and W.-S. Jung“ A Preliminary Study in FMI-based Approaches to Co-simulate Heterogeneous Components in Maritime Systems”, KICS 2024.
- [3] Thule, Casper, Kenneth Lausdahl, and Peter Gorm Larsen. “Overture FMU: export VDM-RT models as tool-wrapper FMUs.” The 16th Overture Workshop. Newcastle University, 2018.
- [4] Peeroo, Kaleem, Peter Popov, and Vladimir Stankovic. “A Survey on Experimental Performance Evaluation of Data Distribution Service (DDS) Implementations.” arXiv preprint arXiv:2310.16630 (2023).