

# HDEVS 형식론에 기반한 통합 하이브리드 모델링 방법론 및 시물레이션 엔진 설계

권세중<sup>1</sup> · 성장호<sup>2</sup> · 송해상<sup>3†</sup> · 김탁곤<sup>1</sup>

## Integrated Hybrid Modeling Methodology and Simulation Engine Design Based on HDEVS Formalism

Se Jung Kwon · Changho Sung · Hae-Sang Song · Tag Gon Kim

### ABSTRACT

A hybrid system is a combination of sub systems which have different types of state and time: a typical example is a combination of discrete event and continuous systems. A HDEVS(Hybrid DEVS) formalism was proposed for modeling and analyzing a hybrid system. The HDEVS formalism allows modelers to construct a hierarchical and modular model based on the mathematical set theory. Because the HDEVS formalism was applied to the distributed and interoperated simulators, modelers should make several heterogenous models dividing a target system. Hence, this paper proposes an extended hybrid coupled model of HDEVS formalism and an integrated hybrid modeling methodology in contrast to the existing simulation framework on interoperable simulators. By applying the proposed modeling method, a target system can be translated to a hybrid model in a similar form as the target system. This paper also contains a simulation engine design for the proposed modeling methodology and a case study which simulates water tank control systems.

**Key words** : Hybrid Systems Modeling and Simulation, HDEVS formalism

### 요약

하이브리드 시스템은 서로 다른 상태와 시간을 가지는 부 시스템의 조합으로 이루어진다. 대표적인 예가 이산 사건 시스템과 연속 시간 시스템의 조합이다. HDEVS 형식론은 이러한 하이브리드 시스템을 모델링하고 분석하기 위해 제안되었는데, 이러한 형식론을 통해 모델러는 수학적 형식론에 기초한 계층적이고 모듈성이 있는 모델을 설계할 수 있었다. 그러나 HDEVS 형식론이 주로 분산된 연동 시물레이션에 적용되었기 때문에 모델러는 하이브리드 시스템을 연동에 참여할 시물레이터에 맞게 서로 다른 모델들로 구분하여 재구성해야 했다. 따라서 모델은 시스템을 그대로 표현하기보다 나누어진 모델들의 연동 구조로 표현되었다. 본 논문은 이러한 문제를 해결하고 통합된 하이브리드 모델을 만들 수 있는 모델링 방법론과 그에 대한 시물레이션 방법론을 제안한다. 기존에 연동형 구조에 적용되었던 것과 달리, 하이브리드 시스템은 그 시스템 본래의 형태 그대로 통합된 모델로 모델링 될 수 있다. 또한 이 논문은 제안하는 모델링 방법론에 따르는 시물레이션 엔진 구조에 대해서 논하고 물탱크 조절 예제를 통한 간단한 사례 연구도 포함한다.

**주요어** : 하이브리드 시스템 모델링 및 시물레이션, HDEVS 형식론

\*본 연구는 국방과학연구소 핵심연구사업의 지원으로 수행되었습니다.(계약번호 UD110037DD)  
접수일(2012년 12월 31일), 심사일(2013년 2월 25일),  
게재 확정일(2013년 3월 18일)

<sup>1)</sup> KAIST 전기및전자공학과

<sup>2)</sup> KIST 유럽연구소

<sup>3)</sup> 서원대학교 컴퓨터공학과

주 저 자 : 권세중

교신저자 : 송해상

E-mail; hssong@seowon.ac.kr

## 1. 서론

일반적으로 시스템의 행동을 분석하거나 모의하기 위해서는 각 시스템의 특성에 맞는 모델링 방법론을 적용하여 모델을 기술하고 이에 알맞은 시물레이션 알고리즘이나 수학 공식을 이용하여 모델을 실행해야 한다. 이산 사

건 시스템의 경우에는 주로 DEVS (Discrete EVent systems Specification)<sup>[1]</sup>와 같은 형식론을 사용하여 모델을 기술하고 DEVS 실행 알고리즘을 통해 모의 결과를 얻으며, 연속 시간 시스템의 경우에는 미분방정식 모델을 기술하여 수치해석 알고리즘을 통해 결과를 얻는다.

실세계의 시스템은 하나의 종류로만 이루어지기보다 두 종류 이상의 시스템이 혼재되어 있는 하이브리드 시스템인 경우가 많다. 예를 들어, 적 미사일을 타격하는 미사일 방어체계는 레이더, 미사일, C2체계, 피해 평가 분석 등의 다양한 부 시스템의 합으로 이루어진다. 이중 레이더의 탐지나 미사일의 기동은 연속 시간에 이루어지는 시스템이며, C2나 피해 평가 분석은 사건 단위의 불규칙한 상태 천이가 일어나는 이산 사건 시스템이다. 이러한 하이브리드 시스템은 부 시스템들의 시간과 데이터의 종류가 각각 다르기 때문에 한 가지 M&S(Modeling and Simulation) 방법론만으로는 효율적이고 정확한 결과를 얻기 어렵다. 따라서 이러한 시스템을 모의하기 위해서는 특수한 하이브리드 M&S 방법론이 필요하다.

이를 위하여 다양한 하이브리드 방법론이 제안되었는데, 그 중 대표적인 예로는 혼합형 모델링 형식론을 제안한 ADEVS<sup>[2]</sup>와 MATLAB에 이산 사건 모델을 추가한 MATLAB/Simulink<sup>[3]</sup>가 있었다. 하지만 ADEVS와 같은 혼합형 모델링 방법은 진입 장벽이 높고 기존 모델의 재사용이 어려웠으며, MATLAB/Simulink와 같은 연속 시간 시스템 해석 툴의 경우는 연속 시간 모델 해석에 초점이 맞추어져 있어 위계임과 같은 대단위 시스템 시뮬레이션에는 부적합한 문제가 있었다.

이러한 문제를 해결하고자 HDEVS 형식론에 기반한 연동형 M&S 방법론이 제안되었는데, 이는 서로 다른 시스템을 위한 시뮬레이터들을 연동하는 방식으로 형식론을 통한 시스템 이론과 함께 연동 이론을 적용하였다<sup>[4]</sup>. 때문에 구조적/계층적 모델링을 통한 대단위 시스템 시뮬레이션에 적합하며 각 단일 시뮬레이터 및 모델을 재사용할 수 있다는 장점이 있다.

하지만 연동형 시뮬레이터에 대한 연구는 연동 구조에 따른 재사용성에 초점을 맞추어, 각 부 시스템들을 분류하여 시스템 종류에 맞는 시뮬레이션 모델들을 분리 구현했기 때문에 통합된 모델 개발 환경을 제공할 수 없었다. 개발된 연동형 모델들은 단일 하이브리드 시스템을 그대로 표현하기보다 사용자에게 의해 나누어진 이기종 모델들의 연동 형태로 표현했기 때문에 실제 시스템 형태가 모델링 결과와 동일하지 않은 문제가 발생했으며 연동에 따른 부하가 발생했다.

본 논문에서는 이러한 연동형 구조에 따른 문제를 해결하면서도 재사용성이 높은 형식론 기반의 모델링이 가능하도록 통합 모델링 방법론을 제안한다. 또한 이를 위하여 기존의 HDEVS 형식론을 일부 수정, 확장한다. 이러한 통합 모델링 방법론을 통해 모델러는 하이브리드 시스템을 분석, 분리하여 각각의 구분된 모델을 설계하기보다 통합된 하이브리드 모델을 설계함으로써 좀 더 직관적인 모델 구현이 가능하다.

또한 본 논문은 제안된 방법론을 통해 설계된 모델을 실행할 수 있는 시뮬레이션 엔진을 설계한다. 구현된 모델은 결국 서로 다른 시간 종류와 데이터 형식을 가지는 시스템으로 이루어져있기 때문에 결국 각각의 시스템 분류에 맞는 시간을 처리하는 시뮬레이션 엔진에 의해 실행되어야 한다. 따라서 사용자의 관점에서는 통합된 엔진이지만 내부적으로는 모델을 분리하여 실행하고 연동하는 시뮬레이션 엔진의 구현이 필요하다. 이를 위해서는 전처리 알고리즘, 시간 관리 알고리즘, 데이터 변환 알고리즘 등이 필요한데, 기존의 HDEVS 형식론에서 제안되었던 내용을 바탕으로 통합형 시뮬레이션 엔진에 필요한 내용들을 본 논문에서 다룬다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 하이브리드 시스템과 기존의 하이브리드 M&S 방법론을 다룬다. 그리고 3장에서 수정된 HDEVS 형식론과 함께 통합 하이브리드 모델링 방법론을 제안하며 4장에서는 제안하는 방법론을 따라 구현된 모델을 위한 시뮬레이션 엔진의 구조에 대해 다룬다. 5장에서는 예제를 통해 결과를 확인하며 마지막으로 6장에서는 결론을 맺는다.

## 2. 하이브리드 M&S 방법론

### 2.1 하이브리드 시스템

일반적으로 시스템은 상태변화의 원인이 되는 입력과 상태변화의 결과로 야기되는 출력 및 시스템의 상태를 기록하는 상태변수로 구성된다<sup>[5]</sup>. 시스템은 시스템을 이루는 입력, 출력, 그리고 상태변수의 종류에 따라서 Fig. 1과 같이 분류할 수 있다.

하이브리드 시스템은 두 가지 이상의 다른 수준의 동적 시스템을 결합한 시스템으로서 Fig. 1의 시스템들 중에서 두 가지 이상의 개념이 혼재되어 있는 시스템을 가리킨다. 본 논문에서는 주로 연속 시간 시스템과 이산 사건 시스템의 하이브리드 시스템을 다룬다.

이러한 부 시스템들이 이루는 하이브리드 시스템 안에

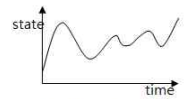
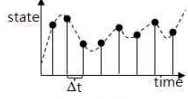
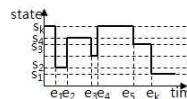
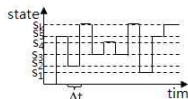
	Continuous Time	Discrete Time
Continuous State	 <ul style="list-style-type: none"> <li>Continuous System</li> <li>Differential Eqn.</li> <li>Analog Circuits</li> </ul>	 <ul style="list-style-type: none"> <li>Discrete Time System</li> <li>Difference Eqn.</li> <li>Sampled-data Systems</li> </ul>
Discrete State	 <ul style="list-style-type: none"> <li>Discrete Event System</li> <li>DEVS Formalism</li> <li>War game</li> </ul>	 <ul style="list-style-type: none"> <li>Digital System</li> <li>Finite State Machine</li> <li>Digital Circuits</li> </ul>

Fig. 1. Classification of Systems<sup>[6]</sup>

는 서로 다른 종류의 데이터가 존재한다. 예를 들어, 연속 시간 시스템은 연속시간에 이루어지는 신호를 입출력으로 사용하며 이산 사건 시스템은 불특정 시간 간격에 발생하는 메시지를 사용한다. 따라서 이들 사이의 데이터 교환이 이루어질 때면, 변환을 담당하는 인터페이스 모델이 필요하다. 이들 간의 연결에는 A/E(Analog-to-Event) 변환기와 E/A(Event-to-Analog) 변환기가 존재하여 서로의 시스템 간의 데이터 변환을 담당하게 된다<sup>[4]</sup>.

## 2.2 기존의 하이브리드 M&S 방법론

하이브리드 M&S 방법론은 크게 모델링과 시뮬레이션 관점에서 3가지로 분류할 수 있다. 첫 번째는, 모델링 형식론 통합 방법으로 하이브리드 시스템만을 위해 제안된 통합 모델링 형식론을 통해 하이브리드 모델을 기술하고 실행한다. 두 번째는, 위와 반대로 부 시스템을 위한 모델을 각각 기술하여 통합된 시뮬레이션 엔진을 사용하여 모델을 실행하는 방법이다. 특수한 방법으로 모델링하기보다 기존의 모델링 방법론을 세부 시스템에 알맞게 적용한다. 세 번째로는, 두 번째와 같이 각각의 알맞은 모델링을 하면서 시뮬레이션 엔진 또한 별도로 실행하는 방법이다. 이러한 경우에는 연동이 필요한데, RTI와 같은 연동 미들웨어를 통해 각 시뮬레이터를 연동한다.

2.2장에서는 이중에 모델링 형식론 통합과 시뮬레이션 엔진 통합만을 다룬다. 시뮬레이션 연동을 통한 하이브리드 방법론은 3장에서 HDEVS 형식론과 함께 다룬다.

### 2.2.1 모델링 형식론 통합

통합된 모델링 형식론을 통한 방법은 DEVS 연구진들이 진행한 연구들이 주를 이루며, DEVS 시뮬레이션 알고리즘으로 연속 시간 시스템과 이산 사건 시스템을 함께

실행한다. PowerDEVS<sup>[7]</sup>, GDEVS<sup>[8]</sup>의 경우에는 연속 시간 모델을 근사하여 DEVS 모델로 나타내는 방법을 제안한 경우이다. PowerDEVS는 Quantized State System(QSS)을 사용하여 연속 시스템의 상태를 양자화 하여 DEVS 시뮬레이션 엔진으로 실행한다. 또한 GDEVS의 경우에는 입출력을 구간적 선형 방정식(Piecewise Linear Equation)으로 근사하여 DEVS 시뮬레이션 엔진을 통해 실행한다. 즉, 이들은 DEVS 시뮬레이션 엔진으로 연속 시스템을 실행하기 위하여 근사된 DEVS 모델을 만든다. 이러한 접근은 DEVS 형식론을 통해 모든 시스템을 나타낼 수 있다는 장점이 있지만 수치 해석 알고리즘 이상의 정확도와 성능을 보장하기 어렵고 연속 시스템을 기술하기 위해 사용자가 DEVS 형식론을 이해해야한다는 문제점이 있다.

이러한 연구에서 가장 진보된 연구 결과가 ADEVS다<sup>[2]</sup>. ADEVS는 Split Hybrid System Modeling (SHSM) 형식론을 사용하여 이산 사건 시스템과 미분 방정식이 결합된 하이브리드 모델을 기술하고 DEVS 실행 알고리즘으로 실행한다. SHSM 형식론은 미분 방정식을 풀어내는 수치 해석 알고리즘을 통해 하이브리드 모델을 실행하며 DEVS 알고리즘은 시간 관리를 담당한다. 그러나 하나의 작은 모델 단위는 하이브리드 모델로 볼 수 있지만 하이브리드 모델 간의 연결은 사건 단위 연결밖에 불가능하기 때문에 모든 하이브리드 시스템을 표현하기에는 문제가 있다.

이러한 통합된 모델링 형태의 하이브리드 모델 기술은 모델링 하는 사용자가 새로운 모델 기술 방법을 이해해야 하며 기존의 많은 연속 모델들을 재사용할 수 없다는 점에서 문제점이 발생한다. 특히 국방 M&S의 이미 개발되어 있는 공학급 모델이나 이산 사건 모델들을 그대로 사용할 수 없고 새로운 형식론에 맞게 구성해야한다는 점과, 정밀한 시뮬레이션을 하기에 어렵다는 점에서 상당한 문제점을 지닌다.

### 2.2.2 시뮬레이션 엔진 통합

모델링은 분리되어 있으나 시뮬레이션은 통합되어 있는 형태는 HyVisual<sup>[9]</sup>, MATLAB/Simulink<sup>[3]</sup> 등과 같이 주로 연속 시간 시스템을 위한 툴에서 찾아 볼 수 있다. HyVisual의 경우는 회로 설계를 위한 연속 시간 시뮬레이터에 이산 사건 모델링을 위해서 유한 상태 머신(Finite State Machine)을 사용한다. 비슷하게 MATLAB/Simulink의 경우에는 연속 시간을 모델링하는 MATLAB에 이산 사건 처리를 위한 Simulink ToolSet을 추가하여 하이브리드 시뮬레이션을 한다.

이러한 구조는 모델링에 있어서 각각의 부 시스템에 맞는 방법을 통해 기술하고 통합 시뮬레이션을 할 수 있기 때문에 모델링 편의성이나 성능 면에서 뛰어나지만 이산 사건 모델링에 있어서 모델이 형식론에 기반을 두지 않거나 불안정한 형식론(FSM)에 기반을 두기 때문에 이산 사건 시스템의 모델링에 적합하지 않다. 특히, 개발 목적이 대단위의 전장 체계를 모델링하는 것과는 맞지 않아서 국방 분야의 워 게임이나 분석 모델을 개발함에 있어서 부족하다.

### 3. HDEVS 형식론에 기반한 통합 하이브리드 모델링

#### 3.1 HDEVS 형식론과 연동형 M&S 방법론

DEVS 형식론은 1976년 B. P. Zeigler에 의해 제안된 집합론에 근거한 형식론으로 이산 사건 시스템을 모듈 별로 나누고 이를 계층적인 연결로 모델링 할 수 있는 수학적 기반을 제공한다<sup>[1]</sup>. 시스템 구성 요소를 나타내는 원자모델과 여러 모델을 합쳐서 새로운 모델을 구성할 수 있는 결합모델로 구성되어 있으며, 이 두 가지 종류의 모델을 사용하여 시스템을 계층적이고 모듈러하게 표현할 수 있다. 이러한 객체 지향적 모델링은 모델의 재사용성, 유지보수성, 신뢰성 등을 높일 수 있다. 따라서 국방 시스템과 같이 대단위의 복잡한 모델링에 적합하다.

이러한 DEVS의 장점을 살리면서 하이브리드 시스템에 적용하고자 한 형식론이 HDEVS(Hybrid DEVS) 형식론이다. HDEVS 형식론은 DEVS 모델과 연속 시간 모델을 연동할 수 있는 이론적 배경이 된다<sup>[4]</sup>. HDEVS 형식론은 기존의 DEVS 결합, 원자 모델에 상미분방정식을 푸는 연속 원자 모델, 연속 결합 모델과 하이브리드 결합 모델을 추가하여 하이브리드 시스템을 계층적이고 모듈러한 형태로 모델링한다<sup>[4]</sup>.

연속 원자 모델과 연속 결합 모델은 연속 신호를 입력력으로 하며, 연속 원자 모델은 상미분방정식으로 기술되는 상태 천이 함수를 가진다. 그리고 하이브리드 결합 모델을 통해 이산 사건을 위한 결합 모델과 연속 시간을 위한 연속 모델을 연동한다. 따라서 HDEVS 형식론의 전체 모델의 구조는 Fig. 2와 같은 형태가 된다. 연속 결합 모델과 원자 모델은 연속 신호를 데이터로 사용하며 DEVS는 이벤트 메시지를 데이터로 사용한다. 이때, 서로 다른 데이터를 변환하기 위해서 하이브리드 결합 모델 안에 변환기 함수가 추가된다.

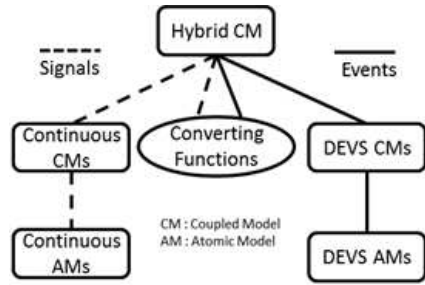


Fig. 2. Structure of Classical HDEVS Formalism

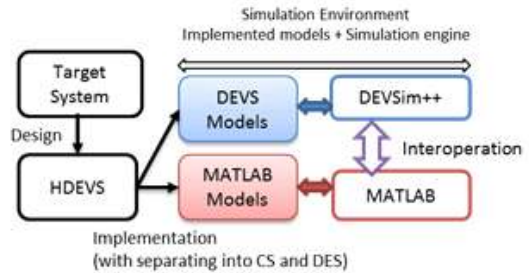


Fig. 3. M&S Process for interoperation<sup>[11]</sup>

Fig. 2와 같이 기술된 모델 명세는 Fig. 3과 같이 연동형 시뮬레이터로 구현된다. HDEVS로 기술된 각종 세부 모델을 분리하여 연속 모델 및 변환기를 MATLAB/Simulink를 통해 구현하고 DEVS 모델을 DEVSIM++을 통해 구현하였다<sup>[12]</sup>. 각각 모델을 정확히 실행할 수 있는 틀을 이용하여 모의하였다는 점에서 장점이 있다. 그러나 HLA/RTI 연동에 따른 연동부하가 증가하며 하이브리드 시스템을 그대로 모델링하기보다 전체 시스템을 해체하여 부 시스템의 종류별로 재구성한 형태로 모델링을 해야 한다는 문제점이 있다.

#### 3.2 제안하는 통합 하이브리드 모델링 방법론

제안하는 통합 하이브리드 모델링 방법론은 기존의 모델을 재사용할 수 있으면서도 통합된 하이브리드 모델을 구조적/계층적으로 구성할 수 있도록 한다. 연속 시간 시스템에 해당하는 컴포넌트는 미분방정식을 통해 연속 시간 모델로 기술하고 이산 사건 시스템에 해당하는 컴포넌트는 DEVS로 기술하여 재사용성을 높인다. 이러한 구조는 HDEVS 형식론과 비슷한 형태를 가지게 되지만, 모든 결합 모델을 확장된 하이브리드 결합 모델로 통합함으로써 최상위에 하이브리드 결합 모델이 단 하나만 존재하게 되

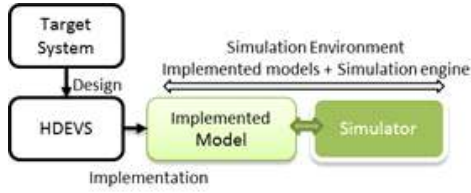


Fig. 4. Proposed M&S Process

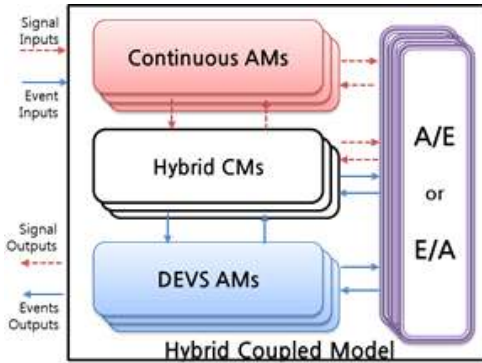


Fig. 5. Internal Structure of Hybrid Coupled Models

는 연동형 HDEVS 형식론과 달리 시스템의 부분 부분이 모두 하이브리드 결합 모델이 될 수 있다.

Fig. 4는 제안하는 모델링방법을 적용한 M&S 과정을 설명한다. Fig. 3과 대조적으로 하나의 모델만이 구현되고 실행되는 것을 확인할 수 있다.

이러한 하이브리드 모델 개발을 위해 필요한 하이브리드 결합 모델의 구조는 Fig. 5와 같다. 하이브리드 결합 모델 내부에는 다수의 이산 사건 모델, 연속 시간 모델, 하이브리드 결합 모델 등이 모두 혼재되어 존재할 수 있다. 다만, 서로 다른 두 종류의 입출력이 존재하기 때문에 입출력 포트 연결에 주의해야하는데, 연속 시간 모델은 신호만을 다루고 이산 사건 모델은 메시지만을 다룬다. 신호와 메시지를 모두 다룰 수 있는 하이브리드 결합 모델에는 신호와 메시지 입출력 포트가 모두 존재한다.

만약 신호가 이벤트로 변환되거나 이벤트가 신호로 변환되는 것을 원할 때는 A/E 또는 E/A 인터페이스 모델을 통해 변환되어야 한다. A/E 모델은 신호 입력을 받아서 메시지 출력을 내놓으며, E/A 모델은 메시지 입력을 받아서 신호 출력을 내놓는다. 이러한 A/E, E/A 모델들도 객체지향적인 모델로 구성함으로써 모듈성을 높인다.

위와 같은 통합 하이브리드 모델링 방법론을 위한

HDEVS 형식론은 다음과 같다. 기존의 HDEVS 형식론에서 이산 사건 결합 모델과 연속 시간 결합 모델을 삭제했으며 하이브리드 결합 모델을 수정했다.

정의 1. 하이브리드 결합 모델

$$HCM = \langle X, Y, M_i, IC, EIC, EOC, SELECT \rangle$$

$$X = X_{disc} \cup X_{cont}$$

- $X_{disc}$  : 이산사건 입력 집합
- $X_{cont}$  : 연속시간입력 집합

$$Y = Y_{disc} \cup Y_{cont}$$

- $Y_{disc}$  : 이산사건 출력 집합
- $Y_{cont}$  : 연속시간출력 집합

$$\{M_i\} \subseteq \{cM_i\} \cup \{dM_i\} \cup \{HCM_i\} \cup \{AEM_i\} \cup \{EAM_i\}$$

- $\{cM_i\}$ : 연속시간 원자 모델의 집합
- $\{dM_i\}$ : 이산 사건 원자 모델의 집합
- $\{HCM_i\}$ : 하이브리드 결합 모델의 집합
- $\{AEM_i\}$ : A/E 변환기 모델의 집합
- $\{EAM_i\}$ : E/A 변환기 모델의 집합

$$IC \subseteq \bigcup_i dY_i \times \bigcup_j dX_j \cup \bigcup_i cY_i \times \bigcup_j cX_j$$

: 내부 Coupling 관계

$$EIC \subseteq X_{disc} \times \bigcup_i dX_i \cup X_{cont} \times \bigcup_i cX_i$$

: 외부 입력 Coupling 관계

$$EOC \subseteq \bigcup_i dY_i \times Y_{disc} \cup \bigcup_i cY_i \times Y_{cont}$$

: 외부출력 Coupling 관계

- $\{dX_i\}$ : 하위 모델들의 이산사건 입력 집합
- $\{cX_i\}$ : 하위 모델들의 연속시간 입력 집합
- $\{dY_i\}$ : 하위 모델들의 이산사건 출력 집합
- $\{cY_i\}$ : 하위 모델들의 연속시간 출력 집합

$$SELECT = 2^M - \emptyset \rightarrow M_i : \text{모델들 간의 우선순위}$$

정의 1은 이러한 하이브리드 결합 모델에 대한 수학적 인 표현이다. 각 연속 모델, 이산 사건 모델, 하이브리드 모델, 변환기 모델들이 동등한 관계를 가지며 Coupling 관계를 가지고 있으며 이산 사건 입출력은 이산 사건 입출력끼리, 연속 시간 입출력은 연속 시간 입출력끼리 연결된 것을 확인할 수 있다. 이러한 하이브리드 모델의 하위 모델은 언급한 것처럼 하이브리드 모델, 이산 사건 모델, 연속 시간 원자 모델, A/E, E/A 변환기가 가능한데, 이산 사건 원자 모델은 DEVS의 원자 모델과 동일하며

나머지 모델들의 수학적 표현은 다음과 같다.

**정의 2. 연속 시간 원자 모델**

$$cM = \langle X_{cont}, S_{cont}, Y_{cont}, f, g \rangle$$

$X_{cont}$  : 연속 입력 집합  
 $S_{cont}$  : 연속 상태 집합  
 $Y_{cont}$  : 연속 출력 집합  
 $f : \frac{d}{dt} S_{cont}(t) = f(S_{cont}(t), X_{cont}(t), t)$   
 연속 상태 천이 함수  
 $g : Y_{cont} = g(S_{cont}(t), X_{cont}(t), t)$   
 연속 출력 함수

**정의 3. A/E 변환기 모델**

$$AEM = \langle X_{cont}, Y_{disc}, f_{AE} \rangle$$

$X_{cont}$  : 연속 시간 입력 집합  
 $Y_{disc}$  : 이산 사건 출력 집합  
 $f_{AE} : \Omega \rightarrow 2^\Sigma$   
 -  $\Omega : X_{cont}$ 의 연속 집합  
 -  $\Sigma : Y_{disc}$ 의 멱집합

**정의 4. E/A 변환기 모델**

$$EAM = \langle X_{disc}, Y_{cont}, g_{EA} \rangle$$

$X_{disc}$  : 이산 사건 입력 집합  
 $Y_{cont}$  : 연속 시간 출력 집합  
 $g_{EA} : \Sigma \rightarrow \Omega$   
 -  $\Sigma : X_{disc}$ 의 멱집합  
 -  $\Omega : Y_{cont}$ 의 연속 집합

기존의 형식론에서는 하이브리드 결합 모델에 속한 함수로만 표현되었던 변환기 모델들은 정의 3과 정의 4처럼 동등한 모델로 표현되었다. 서로 다른 형태의 입출력 집합을 가지고 있으며 각각의 변환 함수를 통해 데이터의 종류를 변환한다.

위와 같은 HDEVS 형식론에 기초한 하이브리드 모델을 사용하여 사용자는 하이브리드 모델의 계층 구조를 모델링할 수 있다. 이러한 모델링 방법은 객체 지향적이고 구조적/계층적인 모델링의 장점을 가진다. 동시에 기존의 연동형 모델링 방식과 달리 국방 시스템 등 다양하고 복잡한 하이브리드 시스템을 있는 형태 그대로 기술할 수 있기 때문에 더 폭넓고 직관적인 하이브리드 시스템 모델링이 가능하다.

**4. 제안하는 통합 모델링 방법론을 위한**

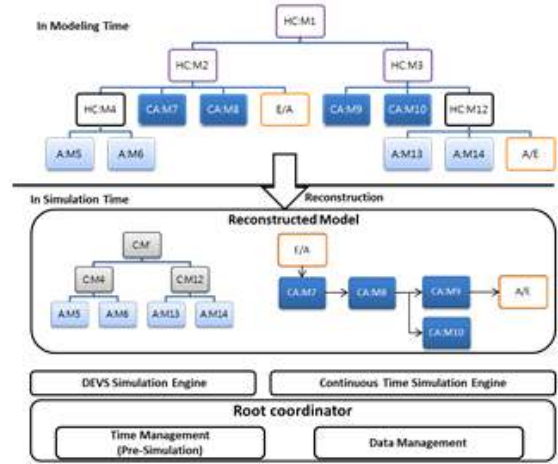


Fig. 6. Structure of Simulation Environment

**시뮬레이션 엔진 설계**

Fig. 6은 통합 하이브리드 모델링 방법론으로 설계된 통합 모델을 실행하기 위한 시뮬레이션 환경의 구조이다. 하나로 통합되어 있는 하이브리드 모델은 실행 시작 전에 시뮬레이션 엔진에 의해 재구성되어 DEVS 모델과 연속 시간 모델로 분리되고 인터페이스 모델들이 식별된다. 또한 통합 시뮬레이션 엔진은 DEVS 해석기와 상미분방정식 풀이를 위한 상미분 방정식(Ordinary Differential Equation, ODE) 해석기를 통해 각각의 모델을 실행하고 Root Coordinator를 통해 시간과 데이터 관리가 이루어진다.

DEVS 실행 엔진은 재사용성을 위해서 DEVSim++ [12] 엔진을 그대로 사용했기 때문에 생략한다. 이어지는 4.1~4.3장에서는 전처리(Pre-Process) 과정, Root Coordinator, 연속 시간 시뮬레이션 엔진에 대해 논한다.

**4.1 전처리 과정**

사용자에 의해서 하나의 결합 모델로 설계된 모델을 실행하는 방법은 하나의 시뮬레이션 방법으로 이루어질 수 없다. 연속 시간 시뮬레이션은 연속된 시간의 흐름을 따라 시간 진행이 일어나며, 이산 사건 시뮬레이션은 이벤트 발생에 따라 시뮬레이션이 진행되기 때문에 동일한 시간 흐름에 따라 통합 시뮬레이션을 진행하게 되면 연속 시간의 흐름을 기준으로 하여 실행을 해야 한다. 따라서 모델이 실행되지 않는 시간을 건너뛰게 되는 이산 사건 모델들은 상당히 비효율적으로 시간을 관리하게 된다. 때

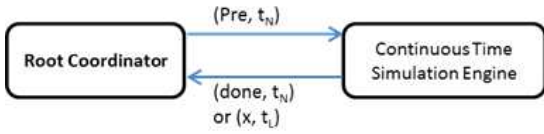


Fig. 7. Pre-Simulation

문에 시뮬레이션 엔진의 성능을 위해서 통합된 하나의 모델을 구현하더라도 시뮬레이션 엔진은 내부적으로 각각의 실행 엔진을 두고 모델을 분리 실행해야한다. 즉, 시뮬레이션 엔진의 전처리 기능으로 사용자가 구현한 모델을 연속시간 모델과 이산 사건 모델로 분리하는 작업이 필수적이다. 이러한 작업은 사용자의 관점에서는 보이지 않기 때문에 통합 모델링의 목적을 훼손하지는 않는다.

제안하는 방법론을 통해 구현된 모델은 하이브리드 결합 모델을 통해 구성되어 있기 때문에 이러한 하이브리드 결합 모델을 해체해서 DEVSim++이 실행할 수 있는 계층적인 DEVS 결합 모델과 연속 시간 시뮬레이션 엔진이 실행할 수 있는 수평적으로 구성된 연속 시간 모델의 집합으로 분리한다.

특히, 연속 시간 모델은 신호가 연속해서 흘러야 하기 때문에 계층적으로 구성될 경우 출력을 전달하는 과정에 있어서 상당한 오버헤드가 발생한다. 따라서 계층을 모두 없애고 수평적으로 구성하며, 각 모델의 출력과 입력을 직접 연결하여 바로 데이터가 전달될 수 있도록 구성한다.

#### 4.2 Root Coordinator

Root Coordinator의 역할은 서로 다른 시간과 데이터 형태를 가지는 엔진 사이의 시간 관리이다. 이때 서로 다른 두 시뮬레이터 사이의 시간 관리는 Pre-Simulation 알고리즘을 사용한다. Pre-Simulation은 DEVS 실행 엔진의  $t_N$ (전체 모델의 다음 실행 시간)까지 미리 연속 시간 엔진을 진행시킴으로, 메시지 발생 여부를 확인하는 알고리즘이다<sup>[11]</sup>. 시뮬레이션 시간 진행의 주도권은 이산 사건 시뮬레이션 엔진에게 있다. DEVS 모델 실행 엔진의 다음 실행시간  $t_N$ 까지 연속 시간 시뮬레이션을 실행시켜서 이벤트 발생 여부를 확인하여, 이벤트가 발생한다면 발생한 이벤트를 기준으로 스케줄링을 하고, 이벤트가 발생하지 않는다면  $t_N$ 시간에 발생한 이벤트를 기준으로 DEVS 모델을 실행한다.

Fig. 7은 Root Coordinator와 연속 시간 엔진 간의 Pre Simulation에 대해 간략히 표현한 그림이다.  $(Pre, t_N)$ 은  $t_N$ 까지 시간 진행을 의미하며  $t_N$ 까지 아무런 상태 이벤트

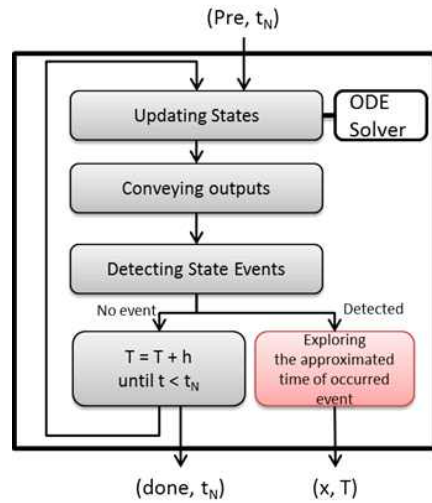


Fig. 8. Continuous Time Simulation Engine

가 발생하지 않는다면 연속 시간 시뮬레이션 엔진은  $(done, t_N)$ 을 반환하며, 시간  $t_i$ 에 메시지  $x$ 가 발생하면  $(x, t_i)$ 을 반환한다.

#### 4.3 연속 시간 시뮬레이션 엔진

연속 시간 해석을 위한 상미분 방정식 해석기에는 4차 룽게-쿠타 알고리즘<sup>[13]</sup>을 적용하였다. 또한 서로의 데이터 교환을 담당하는 상태 이벤트 검출기와 신호 생성기 관리기가 존재하게 된다. 상태 이벤트 검출기는 연속 신호가 A/E 모델에 기술된 조건을 만족시키는지 여부를 검사하며, 신호 생성기 관리기는 E/A 모델에 의해 발생한 연속 신호를 전달한다.

Fig. 8은 연속 시간 시뮬레이션 엔진의 프로세스를 단계별로 나타낸 그림이다.  $(Pre, t_N)$  메시지를 받아서  $t_N$ 까지 시뮬레이션을 진행하게 되면, 시뮬레이션 엔진은 상태 업데이트, 출력 갱신, 상태 이벤트 확인, 시간 업데이트의 순서를 거쳐서 일정한 시간 간격으로 모델들을 실행한다. 이때, 시간 간격은 수치 해석 알고리즘의 시간 간격  $h$ (또는  $\Delta t$ )에 해당한다.

상태 업데이트는 각 모델의 상태 변수를 업데이트 한다. 각 상태 변수에 대한 미분 방정식을 수치 해석 알고리즘을 풀어서  $T$ 로부터  $h$ 만큼 진행시키고 출력을 임시로 저장한다. 임시로 저장된  $T+h$  시간의 출력 값들은 출력 갱신 과정에서 한꺼번에 전달되는데, 만약 하나의 모델씩 상태 업데이트 후에 출력을 갱신 하게 되면 발생한 출력이 연결된 모델의 동일한 시간의 상태 업데이트에 영향을

미치기 때문이다.

연속 시간 모델들의 시간 진행이 끝난 후에는 A/E 변환기 모델들의 상태 이벤트 발생여부를 확인한다. 현재 시간 T로부터 새로 구해진 T+h까지 상태 이벤트 발생 여부를 확인하고 시간 간격 사이에 이벤트가 발생했다면 상태 이벤트의 정확한 발생 시간을 찾기 위한 재귀적 탐색을 통해 시간을 구하며 A/E 변환기의 출력 함수로부터 메시지를 받아서 반환한다.

상태 이벤트가 발생하지 않았다면 T+h까지의 시간 진행을 완료하고 현재 시간 T를 업데이트 하여 시뮬레이션을 계속 진행한다. 주어진 시간 t<sub>N</sub>까지 시간 진행을 하게 되면 Pre Simulation을 종료하고 (done, t<sub>N</sub>)을 반환한다. 현재 시간부터 t<sub>N</sub>까지는 연속 시간 시스템이 아닌 사건 시스템에 영향을 미치지 않는다는 뜻으로 그 시간동안 이산 사건 시스템의 정확한 시뮬레이션 결과를 보장한다.

E/A 변환기로부터 발생하는 시그널은 E/A 변환기에 대응되는 신호 생성기들을 통해 공급한다. 신호 생성기들의 출력은 출력 갱신시에 연결된 입력 변수에 저장된다.

### 5. 사례 연구

제안하는 모델 개발 방법론을 바탕으로 구현된 시뮬레이션 환경을 물탱크 조절 예제에 적용했다.

물탱크 조절 시스템은 Fig. 9와 같은 내용으로 일정 양의 물이 들어오고 나가는 물탱크의 높이가 일정 높이를 넘어가지 않도록 입력을 조절한다. 이때, 물탱크의 높이는 Fig. 9의 수식과 같이 미분방정식으로 표현되고, 물이 일정 높이를 넘어가지 않도록 하는 컨트롤러는 DEVS 모델로 표현되며 두 모델 사이에 데이터 변환은 A/E 변환기와

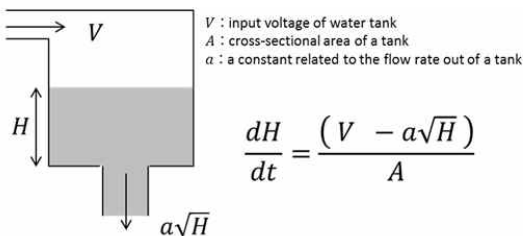


Fig. 9. Water Tank Control Systems

1) 시간 간격 h를 반씩 줄여가면서 재귀적으로 탐색한다. 시간 간격이 최소 시간 단위까지 줄어들거나 데이터의 범위가 최소 범위 내에 들어오면 이벤트 발생을 알리고 종료한다. 자세한 내용은 향후 논문에서 다룰 예정이다.

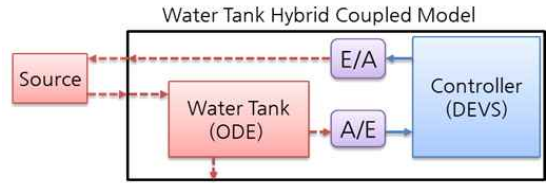


Fig. 10. Structure of Hybrid Coupled Model

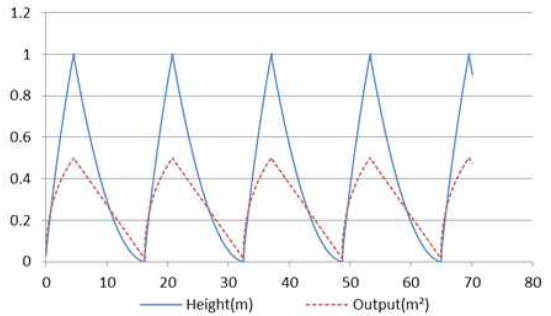


Fig. 11. Experimental Results

E/A 변환기에 의해서 이루어진다.

Fig. 10은 Fig. 9의 시스템을 통합 하이브리드 모델을 통해 나타낸 모델이다. 하나의 하이브리드 결합 모델 안에 연속 시간 모델과 DEVS 모델이 존재하며 그들 간의 데이터 교환을 E/A, A/E 변환기를 통해 수행한다.

물탱크의 높이를 시그널로 받고 있는 A/E 변환기는 물탱크의 높이가 일정 높이를 넘어가는 순간 컨트롤러에게 메시지를 보내고 컨트롤러는 E/A 변환기 모델을 통해 신호를 발생시켜 Source 모델의 출력 신호를 제어한다. Source 모델은 E/A 변환기의 신호에 따라 물의 양을 조절한다. Fig. 11은 위의 모델을 C++로 구현하여 시뮬레이션 엔진을 통해 실행된 결과를 그린 그래프이다. 물이 넘치는 순간(1m)과 높이가 0m가 되는 순간에 컨트롤러가 정상 동작한 것을 확인할 수 있었으며, 이 결과는 기존의 DEVSim++과 MATLAB/Simulink를 사용한 연동형 시뮬레이터로 구현된 결과와 일치함을 확인했다.

### 6. 결론

지금까지 HDEVS 형식론에 기반한 통합 모델 개발 방법론을 제안하고 이를 위해 HDEVS 형식론을 수정했다. 또한 시뮬레이션 엔진 설계에 대해 논하고 물탱크 조절 예제를 통해 동작 결과를 보였다.



제안된 하이브리드 모델링 방법론은 기존의 연동형 HDEVS 형식론을 보완함으로써 하이브리드 결합 모델을 통해 서로 다른 원자 모델들이 혼재된 형태의 모델을 구현할 수 있으며 이를 통해 하이브리드 시스템을 보다 정확하게 기술할 수 있다. 또한 사용자의 관점에서 통합된 모델을 실행할 수 있는 시뮬레이션 엔진을 개발함으로써, 서로 다른 종류의 모델을 실행하되, 사용자의 관점에서는 통합 모델을 그대로 실행할 수 있도록 하였다.

이를 통해 사용자는 모델링 이론에 기초한 하이브리드 모델링을 통해 재사용성을 높이고 대단위 시스템 분석에 적합한 시뮬레이션을 할 수 있다. 이러한 결과물은 세부 알고리즘의 개선을 거쳐 향후 국방 분야 또는 민간 분야의 다양한 시스템 분석과 모의에 사용될 수 있을 것이다.

향후에는 이러한 시뮬레이션 엔진의 수치 해석 알고리즘과 상태 이벤트 검출 알고리즘을 개선하여 성능을 개선할 것이며 복잡한 하이브리드 시스템 예제에 적용할 것이다.

## 참 고 문 헌

1. Zeigler B.P., Praehofer, H., and Kim, T.G., Theory of Modeling and Simulation, Second Edition, Academic Press, 2000.
2. J. Nutaro, A Discrete Event system Simulator, <http://www.oml.gov/~1qn/adevs/adevs-docs/manual.pdf>, September 7, 2011.
3. SimEvents: Model and simulate discrete-event systems, <http://www.mathworks.co.kr/products/simevents/index.html>.
4. Lim, S.Y. and Kim, T.G., "Hybrid Modeling and Simulation Methodology based on DEVS Formalism," SCSC'2001, Orlando, USA, pp. 188 - 193, July, 2001.
5. Kim, T.G., "Modeling and Simulation Engineering", Korea Information Processing Society Review, Vol. 14, No. 6, pp. 3 - 17, Nov., 2007.
6. Kim, T.G., EE612 Lecture Note, KAIST, 2011, <http://sim.kaist.ac.kr/>.
7. PowerDEVS, <http://www.fceia.unr.edu.ar/lsd/powerdevs/index.html>.
8. Giambiasi, N., Escude, B., and Ghosh, S. GDEVS: A generalized discrete event specification for accurate modeling of dynamic systems. Transactions of the Society for Computer Simulation International 2000; 17: 120-134.
9. Brooks, C., Cataldo, A., Lee, E.A., Liu, J., Liu, X., Neuendorffer, S., and Zheng, H., "HyVisual: A Hybrid System Visual Modeler", Technical Memorandum UCB/ERL M05/24, UCLA, CA, July 15, 2005.
10. Kim, T.G. and Kim, J.H., "DEVS Framework and Toolkits for Simulators Interoperation Using HLA/RTI", in Proceedings of ASIAsim '2005, Beijing, China, pp. 16 - 21, Oct., 2005.
11. Sung, C.H. and Kim, T.G., "Framework for Simulation of Hybrid Systems: Interoperation of Discrete Event and Continuous Simulators Using HLA/RTI", 25th ACM/IEEE/SCS Workshop on PADS 2011, Nice, France, June 14-17, 2011.
12. Kim, T.G. and Park, S.B., "The DEVS Formalism: Hierarchical Modular Systems Specification in C++", 1992 European Simulation Multiconference, York, United Kingdom, pp. 152 - 156, June, 1992.
13. Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. Numerical Recipes in C 3rd ed., Cambridge Univ. Press, 2007.



**권 세 중** (sjkwon@smslab.kaist.ac.kr)

2009 한국과학기술원 전산학과 학사  
2011 한국과학기술원 전기및전자공학과 석사  
2011~현재 한국과학기술원 전기및전자공학과 박사과정

관심분야 : 모델링&시뮬레이션, 시뮬레이션 엔진 구현, 하이브리드 시뮬레이션



**성 창 호** (changho.sung@gmail.com)

2003 부산대학교 전자전기통신공학부 학사  
2011 KAIST 전기및전자공학과 박사  
2011~2013 KAIST 정보전자연구소 연수연구원  
2013~현재 KIST 유럽연구소 연구원

관심분야 : 시스템 모델링, 하이브리드 시뮬레이션, 분산 시뮬레이션



**송 해 상** (hssong@seowon.ac.kr)

2000 한국과학기술원 전자전산학과 전기및전자공학전공 공학박사  
1999~2000 고등기술연구원  
2001~2002 (주)스페이스네트  
2002~현재 서원대학교 컴퓨터공학과 교수

관심분야 : 시스템 모델링 시뮬레이션 이론 및 응용, 국방시스템공학, 소프트웨어공학



**김 탁 곤** (tkim@ee.kaist.ac.kr)

1975 부산대학교 전자공학과 학사  
1980 경북대학교 전자공학과 석사  
1988 Univ. of Arizona, 전기및컴퓨터공학과 박사  
1980~1983 부경대학교, 통신공학과, 전임강사  
1987~1989 (미)아리조나 환경연구소, 연구엔지니어  
1989~1991 Univ. of Kansas, 전기및컴퓨터공학과, 조교수  
1991~현재 KAIST 전자전산학과, 교수

- 한국시뮬레이션 학회 회장 역임
- 국제시뮬레이션학회(SCS) 논문지(Simulation) Editor-In-Chief 역임
- SCS Fellow
- 모델링 시뮬레이션 기술사(미국)
- *Who's Who in the World* (Marguis 16th Edition, 1999) 등재
- 연합사, 국방부/합참, 기품원 자문위원 역임
- KIDA Fellow 역임
- ADD 자문위원(현)

관심분야 : 모델링/시뮬레이션 이론, 방법론 및 환경개발, 시뮬레이터 연동