

제약조건이 있는 시뮬레이션을 위한 계층적 모델링 방법론*

Hierarchical Modeling Methodology for Constraint Simulations

이강선**

Kangsun Lee

Abstract

We have many simulation constraints to meet as a modeled system becomes large and complex. Real-time simulations are the examples in that they are constrained by certain non-function constraints (e.g., timing constraints). In this paper, an enhanced hierarchical modeling methodology is proposed to efficiently deal with constraint-simulations. The proposed modeling method enhances hierarchical modeling methods to provide multi-resolution model. A simulation model is composed by determining the optimal level of abstraction that is guaranteed to meet the given simulation constraints. Four modeling activities are defined in the proposed method: 1) Perform the logical architectural design activity to produce a multi-resolution model, 2) Organize abstraction information of the multi-resolution model with AT (Abstraction Tree) structure, 3) Formulate the given constraints based on IP (Integer Programming) approach and embrace the constraints to AT, and 4) Compose a model based on the determined level of abstraction with which the multi-resolution model can satisfy all given simulation constraints. By systematically handling simulation constraints while minimizing the modeler's interventions, we provide an efficient modeling environment for constraint-simulations.

* 이 논문은 2000년도 명지대학교 신입교수 연구비 지원에 의하여 연구되었음.

** 명지대학교 컴퓨터학부 조교수

1. 서론

시스템을 모델링 및 시물레이션 할 때 주어지는 제약조건에는 크게 기능적 요구사항(functional requirements)과 비기능적 요구사항(nonfunctional requirements)이 있다. 이에 따라 시스템의 모델링 과정도 기능적 요구사항을 구현하는 논리 구조 설계(logical architecture design)와 비기능적 요구사항을 구현하는 물리 구조 설계(physical architecture design) 단계로 나누어진다[1]. 논리 구조 설계 단계에서는 시스템을 구성하는 요소와 요소간의 관계 등을 표현하여 시스템의 기능적인 요구사항이 해결되도록 한다. 물리 구조 설계 단계에서는 논리 설계 후 작성된 모델에 비기능적 요구사항을 덧붙여 시스템에 대한 모델을 완성하게 된다. 비기능적 요구사항은 반응시간, 메모리 및 시스템 자원 사용정도, 신뢰성, 유지보수성 등이 포함된다. 시물레이션 수행시 가해지는 제약조건은 비기능적 요구사항이 대부분으로, 실시간 시스템에서의 시간 제약 등이 대표적인 예이다 [1-3].

기존의 모델링 및 시물레이션 CASE 도구들은 여러 라이브러리 함수들을 제공함에 의해 주어진 제약 조건들을 해결하였다. 즉, 모델링 단계에서 그림 형태의 모델을 자동적으로 프로그램화시키고, 시물레이션시의 디버깅·모니터링 및 결과 분석을 도와주는 기능이 있다. 그러나 이러한 CASE 도구들은 시스템에 대한 논리 구조 설계에만 초점을 두어 진행되기 때문에 비기능적 제약조건들을 표현하고 해결하는데 문제점을 지니고 있음이 지적되어 왔다.[1-3] 예를 들어, 시간 제약과 같은 제약조건이 있는 실시간 시스템의 경우 비기능적 제약조건의 만족 여부가 모델이 완성된 후 테스트 단계 혹은 실제 플랫폼에 이식되고 나서야 검토되어진다. 따라서 작성된 모델이 주어진 비기능적 제약조건을 만족시킬 수 없는 경우 다시 모델을 구축해야 하므로, 잘못된 모델을 개발하고 테스트하는 데 소요되었던 시간 및 개발 비용을 낭비하는 문제점이 있다[2-3, 9,15].

실시간 시스템을 위한 스케줄링 분석 방법(Scheduling Feasibility)은 주어진 모델이 제한된 시간이라는 특정 제약조건을 만족하는지를 실제 시물레이션 수행 전에 검토할 수 있는 효과적인 방법이다[6-7]. 실시간 인공 지능은 주어진 시간 안에 문제에 대한 적절한 답을 생성할 수 있는 문제 풀이 과정(problem solving procedure)을 어떻게 생성할 수 있을 가를 연구하는 분야로 주로 탐색(search) 기법을 사용하여 시스템에 주어진 시간 제약을 해결하고 있다[8-9]. 그러나, 이러한 분야에서 제시하는 방법은 시간 제약이 아닌 일반적인 제약 조건을 모델링 하기 힘든 구조임이 지적되어 왔다. 즉, 제시된 방법이 적용될 수 있는 문제의 범위가 극히 제한적이며, 모델의 각 태스크를 어떻게 구성할 것인가에 대한 구체적이고 효과적인 모델링 방법이 없기 때문에 실제 모델링 및 시물레이션 제작 도구로 사용되기에 어려움이 있다[1-3].

본 논문에서는 임의의 제약조건이 있는 시물레이션 수행을 위한 모델링 방법론을 제시한다. 제시된 방법론에서는 1) 논리 구조 설계, 2) 물리 구조 설계 단계를 거쳐 시스템에 주어진 기능적 및 비기능적 제약조건을 해결할 수 있는 모델을 생성하여 시물레이션에 사용한다. 논리 구조 설계 단계에서는 시스템의 기능적인 측면만을 고려하여 다중 추상화 모델(multi-resolution model)을 생성한다. 물리 구조 설계는 비기능적 제약사항을 표현하고 검토하는 단계로, 생성된 다중 추상화 모델을 어떠한 추상화 수준에서 시물레이션 할 때 주어진 모든 제약조건들이 해결될 수 있는지를 검사한다. 최종 모델은 결정된 추상화 수준에서 시물레이션 할 경우 주어진 제약조건의 해결을 보장하므로 제약조건이 효과적으로 해결됨과 동시에 모델 개발비용의 절감을 가져올 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에서 제안한 모델링 방법론의 개괄적 구조에 대해 설명한다. 3장에서는 제약조건을 해결할 수 있는 물리적 설계 방법을 설명한다. 4장에서는 본 연구에서 제안한 방법론을 시간 제약을 갖는

시스템 예를 통해 살펴본 후, 5장에서 결론을 맺는다.

2. 모델링 방법론

시뮬레이션을 위한 모델은 다중추상화를 제공하는 구조가 효과적임이 지적되어 왔다. 즉, 시뮬레이션 목적에 따라 어느 경우에 있어서는 구체적인 시뮬레이션 결과를 줄 수 있어야 하며, 또 다른 경우에 있어서는 개괄적인 의사 결정을 내릴 수 있는 수준을 동시에 제공하여야 한다. 이는 특별히 모델 제작 및 사용이 다수의 사용자를 포함하고 있는 협동적 모델링 상황에서는 필수적인 요건이다[10-11,15].

본 연구에서는 이러한 다중 추상화를 제공하는 모델을 구축하기 위해 1) 수평적 모델링, 2) 수직적 모델링의 절차를 반복적으로 수행한다. 수평적 모델링은, 동일한 추상화 수준에서 모델을 작성하는 과정으로, 현재 추상화 수준에 가장 적합한 모델 형태를 결정하고 이에 따라 모델링 작업을 수행하는 과정이다. 수직적 모델링은 수평적 모델링에서 표현된 모델 요소중에서 임의의 부분을 선택적으로 선별하여 다시 세분화 작업을 하는 과정이다. 이때 세분화 작업에 사용되는 모델 형태는 상위 수준의 모델 형태와 같을 필요가 없다. 수평적/수직적 모델링의 반복은 <그림 1>과 같은 계층적 구조 (hierarchical structure)를 갖는 모델을 생성시킨다.

2.1 및 2.2 절에서는 제안된 모델링 방법론이 어떻게 다중추상화를 제공하는가를 1) 수평적 모델링: 표현 형태가 다른 모델들로의 추상화, 및 2) 수직적 모델링: 선택적 세분화 (selective refinement) 작업에 초점을 두어 설명한다.

2.1. 수평적 모델링

수평적 모델링은 주어진 추상화 수준을 가장 적절히 표현할 수 있는 모델 표현 형태를 선정하고, 이에 따라 시스템을 모델링하는 과정이다. 예를 들어 시스템이 상태(states)와 상태간의 전이

(state transitions)로서 가장 잘 표현되어 지는 경우는 유한 오토마타 (Finite State Machine)에 따라 시스템을 표현한다. 만약 임의의 추상화 수준에서, 시스템이 분리된 기능 단위들의 집합과 그 기능단위들의 제어 흐름을 통해 표현되어 질 경우는 기능 모델링(Functional Modeling)을 이용한다. 본 연구에서는 수평적 모델링을 위해 다음과 같은 모델 표현 방법을 이용한다.

1. 유한 오토마타 (Finite State Machine)
2. 기능도 (Functional Diagram)
3. 시스템운동도(System Dynamics Graph)
4. Equational Modeling
5. Rule-Based Modeling

각 방법에 대한 자세한 모델링 방법은 참고문헌[10]을 참조한다.

2.2 수직적 모델링

수직적 모델링은 수평적 모델링에서 구축된 모델의 요소들을 필요에 따라 세분화하는 과정이다. 수직적 모델링은 Multimodeling 기법[9]을 바탕으로 서로 다른 형태의 모델을 연결하여 결국 전체 시스템의 dynamics가 서로 다른 형태의 모델로 하나의 시간 축 상에서 관찰 할 수 있도록 한다.

수직적 모델링이 적용될 수 있는 요소는 수평적 모델링이 어떠한 모델 형태로 이루어 졌는가에 따라 결정된다. 예를 들어 수평적 모델링에서 유한 오토마타를 사용하였을 경우 수직적 모델링을 통해 세분화 되어질 수 있는 모델 요소는 1) 상태(state)와 2) 상태전이(state transition)이다. 또한 수식 모델링의 수직적 모델링 과정에서 세분화될 수 있는 부분은 상태변수를 제외한 나머지 수식항이 해당된다. 각 모델 종류별 세분화 가능한 모델 요소는 <표 1>과 같이 정리될 수 있다. 수직적 모델링에서는 일단 세분화 가능한 요소들이 파악된 후, 앞서 수평적 모델링 과정을 반복적으로 수행한다. 수직적 모델링은 입력/출

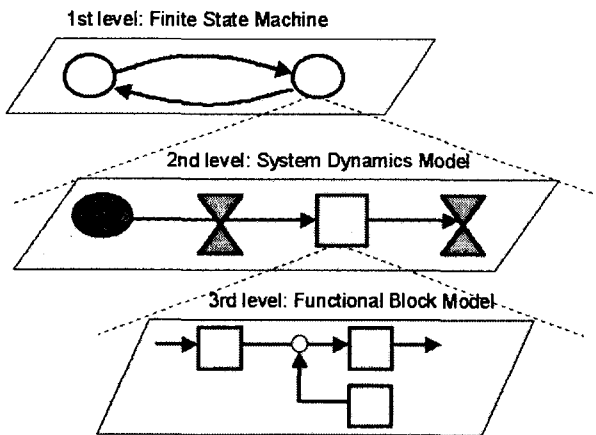
력 간의 맵핑을 통해 서로 다른 표현 형태의 모델들이 하나의 모델화 될 수 있도록 하는 Multimodeling 메카니즘을 제공한다.

<그림 1>은 모델링 과정의 예를 도식화 한 것이다. 첫 번째 단계에서는 시스템을 2가지 상태를 가지는 유한 오토마타로 묘사한다. 작성된 모델을 다시 세분화할 필요가 있을 때, 세분화 모델 요소를 파악하고, 각 요소를 효과적으로 표현할 수 있는 같은 형태 혹은 다른 형태의 모델을 작성한다. <그림 1>은 두 번째 단계에서 시스템 운동도를 이용하였고, 3번째 단계에서는 시스템 운동도의 일부 요소를 다시 기능도를 이용하여 세분화하였다.

제안된 모델링 방법론에 대한 자세한 내용은 참고문헌[5,10,15]에 나와 있다.

<표 1> 모델 형태에 따른 선택적 세분화 가능 요소

모델 형태	선택적 세분화 가능 요소
Finite State Machine	State, State Transitions
Functional Block Model	Block, Functional Directionality
System Dynamics	Rate, Level, Flow
Equation	non-state variables
Rule-Based Modeling	Preconditions, Consequences



<그림 1> 다중 모델링의 예

3. 제약조건이 있는 시뮬레이션을 위한 모델링 방법론

2장에서 살펴본 다중추상화 모델링 방법의 특징을 고려할 때, 제약조건이 있는 시뮬레이션이란 주어진 제약조건을 해결하기 위한 최상의 추상화 수준을 결정한 후 결정된 추상화 수준에 따라 모델을 재구성하는 문제로 재정의되어 질 수 있다.

3.1 절에서는 먼저 제약조건 해결을 위한 추상화 수준 결정 방법을 Integer Programming 기법에 기초한 수학적 모델로 정형화시키는 과정을 보인다.

Integer Programming은 OR (Operations Research)에서 최적화 문제를 위해 사용되는 방법중의 하나로, 목적함수에 관련된 변수들이 정수값을 갖는 부류이다[12-13]. 3.2 절은 결정된 추상화 수준에서 시뮬레이션을 수행하기 위해 어떻게 모델을 재구성하는가를 살펴본다.

3.1 제약조건 해결을 위한 추상화 수준 결정 방법

시스템이 가지고 있는 제약조건을 C_1, C_2, \dots, C_n 이라 하자. 본 연구에서는 주어진 제약조건을 만족시키는 추상화 수준을 선택할 때 다음과 같은 사항을 고려한다.

고려사항 1. 수평적, 수직적 모델링을 통해 만들어진 모델 요소들을 최대한으로 보전한다. 즉, 주어진 제약 조건을 만족시키는 한도에서 모델링 과정에서 생성된 정보가 최소로 손실되도록 한다.

고려사항 2. C_1, C_2, \dots, C_n 의 모든 제약 조건이 만족되어야 한다

추상트리(Abstraction Tree)는 모델링 과정에서 획득한 추상화 정보를 표현하는 데이터 구조이다. AT의 node는 모델을 이루는 각 메소드로 구성된다. AT의 edge는 부모 노드와 자식 노드

사이에 추상화 관계가 성립할 때 부여된다. 즉, AT상의 임의의 노드 P_i 가 Q_1, Q_2, \dots, Q_m 으로 수직적 모델링 되었을 때 P_i 는 parent node로, Q_i 는 child node로 구성된다.

또한 AT에서는 AND와 OR 관계를 포함한다. 즉, P_i 가 Q_1, Q_2, \dots, Q_m 모두를 실행해야 종료되는 경우는 AND 관계로, P_i 가 Q_1, Q_2, \dots, Q_m 중 어느 하나의 메소드를 실행함에 의해 종료되는 경우는 OR 관계로 정의된다. 따라서 임의의 추상화 수준이란, OR 관계에 있는 node 중 어떠한 node를 시뮬레이션 모델에 포함할 것인가를 결정함에 의해 구성된다.

<고려사항1>은 제약조건을 만족시키기 위해 구축된 모델의 추상화 수준을 높여야 한다면, AT상에서 child수가 가장 적은 노드를 선택하여 손실되는 정보가 최소화 되도록 한다. <고려사항2>는 <고려사항1>을 해결할 때 동시에 만족해야 하는 사항들을 정의한다.

Algorithm 1은 이상과 같은 내용을 구체화한 것이다. E_{ij} 변수는 AT 상에서 OR로 구성되며 대응되는 node가 주어진 제약조건을 만족하기 위해 실행될 수 없는 경우는 1의 값을, 그 외의 경우는 0의 값을 갖는 이진변수이다. (Eq. 1)은 추상화 수준 결정을 위한 목적함수로, 제약조건을 만족하는 추상화 수준을 정할 때 AT상에서 손실되는 노드의 수를 최소화해야 한다는 <고려사항 1>을 반영한다. (Eq. 2)는 주어진 모든 시뮬레이션 제약조건을 설정한다. 즉 (Eq. 2)의 좌변은 임의의 추상화 수준을 택했을 경우 주어진 제약조건 해결을 위해 예상되는 손실분이며 우변은 손실분의 한계치를 정의한다. 각 제약 조건의 성격에 따라 부등식이나 등식의 형태로 정의되어질 수 있다. (Eq. 3)은 parent에서 추상화가 이루어질 경우, parent의 child가 중복적으로 추상화 이루어지는 것을 방지하기 위한 수식이다. 즉, (Eq. 3)의 우변은 제약조건이 있는 시뮬레이션 수행을 위해 부모노드가 선택되었는지의 여부를 나타낸다. 부모노드가 선택되었는 경우는 ($E_{ik} = 1$), 우변값이 0이 되므로, 부등식을 성립하기 위해 좌변의 값이 0이 되어야 한다. 좌변의 값을 0

Algorithm 1. 제약조건 만족을 위한 추상화 수준 결정 알고리즘

$$\text{Minimize } \sum_{i=1}^l \sum_{j=1}^{m_i} E_{ij} * D_{ij} \quad (\text{Eq.1})$$

subject to

$$\sum_{i=1}^l \sum_{j=1}^{m_i} C1_{ij} * E_{ij} \leq C1 \quad (\text{Eq. 2})$$

$$\sum_{i=1}^l \sum_{j=1}^{m_i} C2_{ij} * E_{ij} \geq C2$$

:

$$\sum_{i=1}^l \sum_{j=1}^{m_i} Cn_{ij} * E_{ij} \geq Cn$$

$$\sum_{k=1}^{m_i} E_{i+1,k} \leq m_i(1 - E_{ik}) \text{ for each } i \quad (\text{Eq. 3})$$

where

E_{ij} : AT에서 i번째 level의 j번째 node

D_{ij} : AT에서 E_{ij} 에 종속된 child node의 개수

Ck_{ij} : AT에서 E_{ij} 에 Ck의 조건에 대한 손실/손익분

Ci : i번째 제약조건의 임계치

m_i : i node의 자식수

으로 하기 위해서는 모든 자식노드에 대응하는 이진변수 E_{ij} 의 값이 0이어야 하므로, 부모 메소드가 시뮬레이션에서 선택되어졌다면 자식 메소드는 고려될 수 없다는 사실을 수학적으로 구현한다. 제약조건을 해결하는 추상화수준이란 결국 (Eq. 1)의 목적함수를 (Eq. 2)의 집합과 (Eq. 3)를 동시에 만족하여 풀 때 결정되어진다.

Integer Programming 기법에 의해 1의 값을 갖는 E_{ij} 를 가려내면 AT상에서 주어진 제약 조건을 해결하기 위해 생략되어야 할 모델 요소를 가려내게 된다. 추상화 수준은 E_{ik} 값이 0인 모든 메소드들의 집합으로 이루어진다.

3.2 추상화 수준에 따른 모델의 동적 구성

3.1의 단계를 통해, 주어진 제약조건을 해결하는 추상화 수준 결정 과정을 보았다. 그러나 결정된 추상화 수준으로부터 시뮬레이션 수행을 위

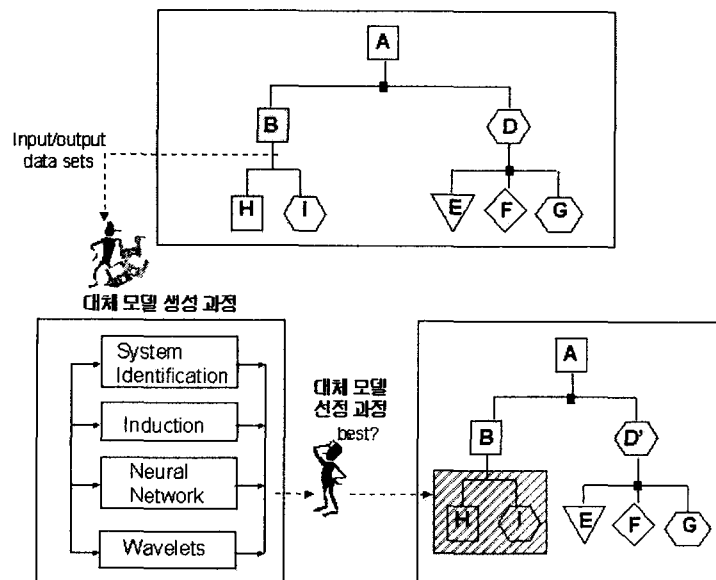
한 모델은 바로 생성될 수 없다. 이것은 앞서 2장에서 살펴본 바와 같이, 제안된 모델링 방법론이 갖는 계층적 구조 때문이다. 계층적 모델링 방법론은 임의의 추상화 수준에서 시뮬레이션이 수행될 때 바로 하위 추상화 수준의 모델 요소에 의해 실행에 필요한 정보가 제공되는 구조이므로, 어떠한 추상화 수준으로 우리가 시스템을 바라보건 실제적인 시뮬레이션은 항상 최하위 leaf 노드에서 일어나는 문제점이 있다. 따라서 결정된 추상화 수준에서 바로 시뮬레이션이 수행되려면 생략된 모델 요소에 대한 실행 정보를 시뮬레이션 수행시 제공해 줄 수 있는 방법이 요구된다.

System Identification, Neural Network, Wavelets는 학습능력을 이용하여 주어진 시스템의 행동양식을 모방 및 예측할 수 있다[13]. 본 연구에서는 System Identification, Neural Network, Wavelets을 이용하여 결정된 추상화 수준의 하위 추상화 수준이 갖는 행동(behavior)을 학습시킨다. 학습을 통해 생성된 system identification 메소드, neural network 메소드,

wavelets 메소드는 제약조건을 만족하는 추상화 수준이 결정되었을 때, 결정된 추상화 수준의 하위 메소드를 대체할 수 있는 black box 형태의 메소드로 활용된다. 특히 이러한 형태의 black box 메소드들은 일단 학습이 끝난 후에는 주어진 입력에 대한 예상되는 출력을 결정하는 시간이 매우 짧기 때문에 실제 시뮬레이션 수행시간에는 거의 영향을 미치지 않는다. 따라서, 3.1절에서 결정된 임의의 추상화 수준에서 시뮬레이션이 즉각 수행될 수 있는 효과적인 수단을 제공한다. 그림 2는 결정된 추상화 수준을 제공하는 모델의 동적 구성 과정을 보여준다.

4. 실험

시뮬레이션 수행시 가해지는 제약조건에는 반응시간, 메모리 사용정도 및 자원 사용도 등이 있을 수 있다. 즉, 구축된 모델을 일정한 한도의 메모리를 사용하면서 반응시간은 주어진 범위 안에 있도록 하는 경우 등 대부분의 복잡한 시스템에 대한 시뮬레이션에는 제약조건이 있는 경우가



<그림 2> 결정된 추상화 수준의 모델 생성 과정

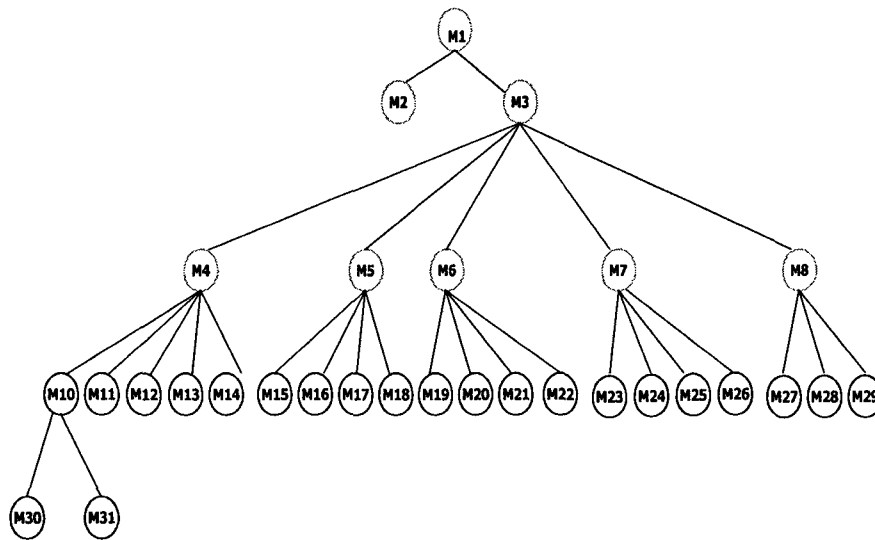
많다. 실시간 시스템은 제약 조건이 있는 시스템의 대표적인 예로, 실시간 시스템에 대한 시뮬레이션은 반드시 주어진 시간 안에 그 결과를 생성해야 한다. 실시간 시스템의 경우는 정확한 결과값을 주어진 시간을 초과하여 생성하는 것보다, 주어진 시간 안에 비교적 적당한 결과값을 생성하는 것이 더 바람직한 경우가 있다. 예를 들어 실시간 로봇 제어를 위한 시뮬레이션의 경우, 주어진 시간 안에 로봇의 진행경로상에 있는 방해물의 윤곽정도를 파악하는 것이 정확한 방해물의 실체를 주어진 시간을 초과하여 생성하는 것보다 훨씬 효율적이다. 실시간 로봇 제어와 같은 시스템에 추가되는 시뮬레이션 제약 조건은 크게 1) 반응시간, 2) 정확도의 두 가지가 있다. 즉 주어진 반응 시간을 만족시키는 범위에서 최상의 정확한 결과를 생성해 내야 한다[16-17].

실시간 로봇 제어를 위한 다중 추상화 모델이 <그림 3>과 같은 AT 구조를 생성하였다 하자. 또

한, <그림 3>에서 M4, M5, M6, M7, M8에서 추상화가 일어날 수 있다고 하자. 이때 M4 - M8에서 추상화가 일어날 경우, 시간의 단축 정도 및 정확도의 손실이 <표 2>와 같이 알려져 있다 하자.

M1 - M31까지는 2장에서 제시한 모델링 방법에 따라 구축되었기 때문에 각기 다른 형태를 취하고 있다. 이때 650만쯤의 speedup을 얻는 경우 중 가장 작은 정확도 손실을 갖는 추상화 수준을 결정한다고 하자. (Eq 1) - (Eq. 3)은 Algorithm 1에 따라 주어진 제약조건 시뮬레이션을 정형화한 것이다.

$$\begin{aligned} & \text{Minimize}(7*M_4 + 4*M_5 + 4*M_6 + 4*M_7 + 3*M_8) \text{ (Eq 1)} \\ & \text{subject to} \\ & t_4*M_4 + t_5*M_5 + t_6*M_6 + t_7*M_7 + t_8*M_8 \geq T_{\max} \text{ (Eq 2)} \\ & a_4*M_4 + a_5*M_5 + a_6*M_6 + a_7*M_7 + a_8*M_8 \leq A_{\max} \text{ (Eq 3)} \\ & t_4 = 592, t_5 = 100, t_6 = 19, t_7 = 31, t_8 = 22 \\ & a_4 = 0.59, a_5 = 0.45, a_6 = 0.43, a_7 = 0.47, a_8 = 0.07 \\ & T_{\max} = 650, A_{\max} = 5.0 \end{aligned}$$



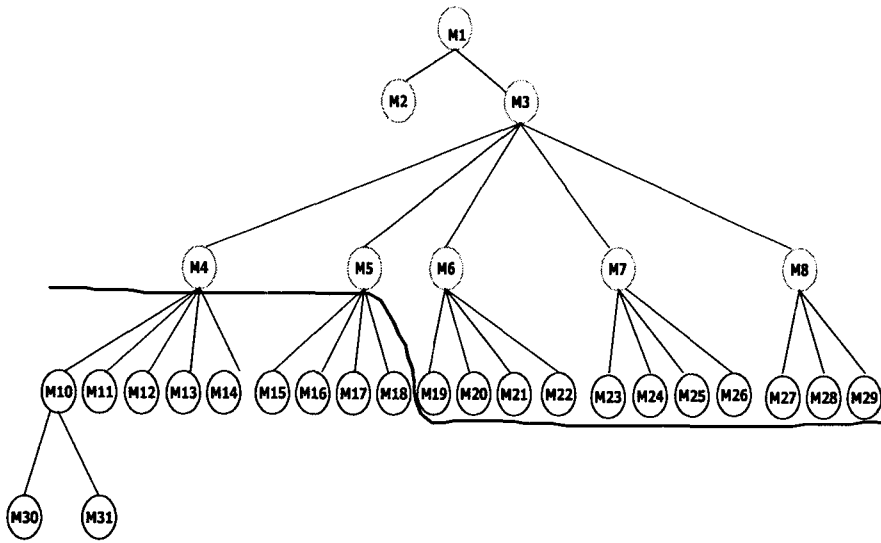
<그림 3> AT의 예

<표 2> Speedup/ Error의 예

	M4	M5	M6	M7	M8
speedup	592	100	19	31	22
Error	0.59	0.55	0.43	0.47	0.07

위에서 정의된 Integer Programming 모델을 서로 다른 speedup 값에 따라 풀면 <표 3>과 같은 결과를 얻을 수 있다. 예를 들어 본래 모델의 시뮬레이션 시간보다 650만큼 시간 단축이 필요한 제약조건이 있는 시뮬레이션의 경우에는, M₄와 M₅에 대응되는 모델요소를 추상화시킬 경우 가장 정확한 시뮬레이션 결과를 제시할 수 있음을 <표 3>에서 보여 주고 있다. <그림4>은 650의 speedup이 필요한 경우 결정된 추상화 수준을 보여주고 있다. 즉, M₄와 M₅는 본래의 모델이 아닌 그 행동만을 예측하여 제공할 수 있는 neural network, system identification, wavelets

형태의 모델로 바뀌어 제약조건이 있는 시뮬레이션에 쓰이게 된다. 결정된 추상화 수준은 주어진 제약조건을 만족시키기 위해 생략되어야 할 모델 표현 방식을 제시하는 효과도 있다. 예를 들어 생략된 M₄, M₅가 유한오토마타 메소드였다면, 주어진 제약조건 하에서는 상태와 상태 전이에 대한 정보는 적합지 않음을 제시한다. 만약 특정 모델 형태에 대해서만 추상화를 시킬 필요가 있는 경우, 혹은 특정 모델 형태를 보존시키는 것이 필요한 경우는 이를 (Eq. 2)와 같은 형태로 추가하면 되므로, 제시된 방법은 임의의 제약조건을 유연하게 표현할 수 있는 구조임을 알 수 있다.



<그림 4> 650 speedup을 위한 추상화 수준

<표 3> 예에 대한 추상화 수준 결정 결과
(정확도 손실치: SSE(Sum of the Squared Error))

deadline	50	90	200	650	700
결과					
생략되는 메소드	M5	M5	M4	M4,M5	M4,M5,M6
정확도 손실도	0.55	0.55	0.59	1.14	1.59

5. 결론

본 연구에서는 1) 계층적 구조를 띤 다중 추상화 모델을 구축하는 방법과, 2) 주어진 제약조건을 만족시키기 위한 임의의 추상화 수준 선정 방법을 살펴보았다. 결정된 추상화 수준은 제약조건이 있는 시뮬레이션을 위해 모델을 재구성할 때 이용됨을 보였다. 제시된 방법론은 모델링 및 시뮬레이션 CASE 도구인 OOPM(Object-Oriented Physical Modeler)에 구현되어 활용되고 있으며 [15], 시스템의 요구사항 분석에 활용될 수 있다. 즉, 시스템에 대한 모델을 트리 형태로 계층적으로 표현하고, 주어진 요구사항을 제시된 수식 모델을 통해 정형화한 후, 구축된 모델에서 각 요구사항을 만족시키는 추상화 수준이 존재하는가를 검사함으로써 요구사항 분석을 수행할 수 있다. 또한 제시된 방법을 모델 설계 단계가 아니라 실제 수행 중에 적용하여, 외부의 변화 조건을 반영할 수 있는 모델을 동적으로 재구성하는데 사용할 수 있다.

실시간 물체 추적 시스템은 대표적인 예로, 시간과 정확도의 제약조건을 동시에 만족시키면서 주어진 시간 안에 추적하고자 하는 물체의 위치와 방향을 동적으로 알아내야 한다[16-17]. 그러나, 모델링 단계가 아닌 실행단계에서 동적으로 추상화 수준을 결정하고 이에 따라 모델의 구조를 동적으로 바꾸는 것은 제시된 수식 모델을 실제 푸는데 걸리는 시간이 길기 때문에 효과적이지 못하다. 휴리스틱 검색이나 확률에 근거한 검색 기법은 수행에 필요한 시간이 짧기 때문에, 실행단계에서 모델 구조를 동적으로 재구성하는데 쓰일 수 있다[9,15]. 또한 임의의 추상화 수준에서 시뮬레이션이 일어나도록 하기 위해 본 논문에서는 하위 수준의 모델 요소들을 하나의 neural network 등의 모델로 대체시키는데, 이때 학습에 필요한 모든 변수들이 개발자의 직관 및 trial-and-error에 의해 얻어지게 된다. 이러한 단점은 본 논문에서 제시된 모델링 방법론을 자동화하는데 장애가 된다. 따라서 하위 수준의 모델 요소들을 학습하는 데 필요한 최상의 학습 파라

미터들을 자동적으로 산출하는 방법이 부가적으로 연구되어야 한다.

참고 문헌

- [1] Burns, A. and Wellings, A.J, "HRT: HOOD: A structured design method for hard real-time systems", Real-Time Systems, vol 6, pp 73-114, 1994
- [2] Alan J. Garvey and Victor R. Lesser, "Design-to-time real-time scheduling", IEEE Transactions on Systems, Man and Cybernetics, vol. 23, no. 5, 1491-1502, 1993
- [3] Kopetz H., Zainlinger R., Fohler G., "The design of real-time systems: from specification to implementation and verification", Software Engineering, vol. 6, 72-82, 1991
- [4] J.S Lark, L.D. S. Forrest, K.P.Gostelow, "Concepts, methods, and languages for building timely intelligent systems", Real-Time Systems, vol.2, 1990, pp. 127-148
- [5] Kangsun Lee and Paul A. Fishwick, "A methodology for dynamic model abstraction", Transactions of the society for computer simulation international, vol 13., no 4, 217 - 229, 1997
- [6] K. Ramamritham, J.A. Stankovic, "Dynamic task scheduling in distributed hard real-time systems", IEEE Software, vol. 1, no. 3, 1984, 65 - 75
- [7] J.A Stankovic, K. Ramamritham, S. Cheng, "Evaluation of a flexible task scheduling algorithm for distributed hard real-time systems", IEEE Trans. Comput., vol 34, no. 12, 1985, pp 1130 - 1143
- [8] R. E. Korf, "Depth-limited search for real-time problem solving", Real-Time Systems, vol. 2, 1990, 7- 24
- [9] Rutledge G. W., "Dynamic selection of models", Ph.D Dissertation, Department of Medical Information Sciences, Stanford

- University, 1995
- [10] Paul A. Fishwick, "Simulation model design and execution: Building digital worlds", Prentice Hall, 1995
- [11] Paul K. Davis and James Bigelow, "Introduction to multiresolution modeling (MRM) with an example involving precision fires", Proceedings of SPIE: Enabling Technology for Simulation Science II, 14-27, 1998
- [12] Don Phillips Ravindran and James J. Solberg, "Operations Research", John Wiley and Sons, 1987
- [13] "CPLEX: Using the CPLEX Callable Library", CPLEX Optimization Inc., 1995
- [14] Timothy Masters, "Neural, novel and hybrid algorithms time series prediction", John Wiley and Sons, Inc., 1995
- [15] Kangsun Lee and Paul A. Fishwick, "OOPM/RT: A multimodeling methodology for real-time simulation", ACM Transactions on Modeling and Computer Simulation, vol. 9, no. 2, April, 1999, pp. 141 - 170
- [16] Alan J. Garvey and Vistor R. Lesser, "Design-to-time scheduling with uncertainty", UMASS Computer Science Technical Report, 1995, 95-03
- [17] Jane W. S. Liu, Kwei-Jay Lin, et. al, "Algorithms for scheduling imprecise computations", IEEE Computer, 1991, 58-68

● 저자소개 ●



이강선

1992 이화여자대학교 전자계산학과, 이학사
 1994 이화여자대학교 전자계산학과, 이학석사
 1998 Computer Information Science and Engineering,
 University of Florida, Ph.D
 1999.1~2000.2 삼성전자 중앙연구소 소프트웨어센터 선임연구원
 2000.3~현재 명지대학교 컴퓨터공학과 조교수
 관심분야 : Modeling Methodology, Web-Based Simulation,
 Component-Based Modeling and Simulation