한국시뮬레이션학회 '00춘계학술대회 논문집 2000.4.15 영남대학교

제약조건이 있는 시뮬레이션을 위한 모델링 방법론

이강선
명지대학교 공과대학 컴퓨터학부

A Modeling Methodology for Constraint Simulation

Kang Sun Lee
Division of Computer Science and Engineering, MyongJi University

요 약

실시간 시뮬레이션은 제약조건이 있는 시뮬레이션의 대표적인 경우로, 반드시 주어진 시간 안에 최상의 시뮬레이션 결과를 생성해내야 한다. 본 논문에서는 시간제약과 같은 시스템에 주어진 제약조건을 모델링 및 시뮬레이션하기 위한 방법론을 제시한다. 시스템의 모델링 과정에서는 계층적 방법론에 기초하여 다중 추상화(multi-resolution)를 제공하는 모델을 만든다. 또한 제약조건이 있는 시뮬레이션을 위해서는 구축된 모델의 추상화 수준을 조절한다. 즉, 1) 구축된 모델을 추상화 수준에 따라 정리하여 AT(Abstraction Tree)를 생성한 후, 2) 주어진 제약조건을 integer programming 기법을 이용하여 정형화하고, 3) 제약조건을 만족시키는 최상의 추상화 수준을 AT상에서 결정한다. 결정된 추상화 수준에 따라 시뮬레이션에 필요한 모델을 동적으로 재구성하여 임의의 제약조건이 있는 시뮬레이션을 효과적으로 수행함으로써, 모델의 재사용 및 경제성을 증진시킬 수 있다.

1. 서 론

시스템에 주어지는 요구사항은 크게 기능적 요구사항(functional requirements)와 비기능적 요구사항(nonfunctional requirements)로 나누어진다. 시스템의 모델링은 기능적 요구사항을 구현하는 논리 구조 설계(logical architecutre design)와 비기능적 요구사항을 구현하는 물리 구조 설계(physical architecture design)단계로 나누어진다.[1] 논리 구조 설계 단계에서는 시스템을 구성하는 컴포넌트와 이들간의 관계등을 표현하여 시스템의 기능적인 요구사항이 해결하도록 한다. 물리 구조 설계 단계에서는 논리 설계 후 작성된 모델에 비기능적 요구사항을 덧 붙여 시스템에 대한 모델을 완성하게 된다. 비기능적 요구사항은 시스템의 실제 수행에 관련된 요구사항으로 성능, 신뢰성 등이 포함된다. 시뮬레이션 수행시 가해지는 제한조건은 비기능적 요구사항이 대부분으로, 실시간 시스템에서의 시간제한등이 대표적인 예이다. [1-3]

본 논문에서는 임의의 갯수의 제약조건이 있는 시뮬레이션을 수행하기 위한 방법론을 제시한다. 제시된 방법론은 모델링 단계에서 주어진 제약조건을 해결할 수 있는 모델을 생성하여 시뮬레이션에 사용하게 된다. 즉, 논리 구조 설계 단계를 거쳐 생성된 다중 추상화 모델

(multi-resolution model)에서 시스템에 부여된 제약조건들을 해결할 수 있는 최상의 추상화 수준을 결정하고 이에 따라 모델을 동적으로 재구성하게 된다. 시뮬레이션은 결정된 추상화 수준을 갖는 모델로 이루어지므로, 제약조건이 효과적으로 해결됨과 동시에, 모델의 재사용을 높이고 개발비용의 절감을 가져올 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에서 제안한 방법론의 개괄적 구조에 대해 설명한다. 3장에서는 제약조건을 해결할 수 있는 방법을 설명한다. 4장에서는 본 연구에서 제안한 방법론을 예를 통해 살펴본 후, 5장에서 결론을 맺는다.

2. 모델링 방법론

시뮬레이션을 위한 모델은 다중추상화를 제공하는 구조가 효과적임이 지적되어 왔다 [4-6]. 즉, 시뮬레이션 목적에 따라 어느 경우에 있어서는 구체적인 시뮬레이션 결과를 줄 수 있어야 하며, 또 다른 경우에 있어서는 개괄적인 의사 결정을 내릴 수 있는 수준에서 단순한 구조의 모델이 적합함이 지적되어 왔다. 이러한 다중 추상화 수준을 제공할 때, 각 추상화 수준별로 별개의 모델을 만드는 것이 아니라, 하나의 모델을 구축한 후, 그 모델에서 추상화 수준을 조절하여 주어진 시뮬레이션 목적에 맞는 모델을 동적으로 재구성하는 것이 효과적이다. 본 연구에서는 이러한 다중 추상화를 제공하는 모델을 구축하기 위해 1) 수평적 모델링, 2) 수직적 모델링의 절차를 반복적으로 수행한다. 즉, 시스템에 대한 가장 상위의 모델을 정의한 후 이를 효과적으로 표현할 수 있는 모델링 표현 도구를 선택한다.(수평적 모델링) 이때 표현된 모델에서 임의의 부분을 다시 세분화할 필요가 있을 때 수직적 모델링을 수행한다. 수평적/수직적 모델링의 반복은 <그림 2>와 같은 계층적 구조 (hierarchical structure)를 갖는 모델을 생성시킨다.

2.1. 수평적 모델링

수평적 모델링은 주어진 추상화 수준을 가장 적절히 표현할 수 있는 모델링 방법론을 선정하고, 이에 따라 시스템을 표현하는 과정이다. 예를 들어 시스템이 상태와 상태간의 전이로서 가장 잘 표현되어 지는 경우는 유한 오토마타 (Finite State Machine)에 따라 시스템을 표현한다. 또한, 시스템이 구분되는 기능단위로 나누어지고, 구분되는 기능단위의 흐름을 통해 표현되어 질 수 있을 경우 기능 모델링 (Functional Modeling)을 통해 표현되어 질 수 있다. 본 연구에서는 수평적 모델링을 위해 다음과 같은 모델링 방법을 이용한다.

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

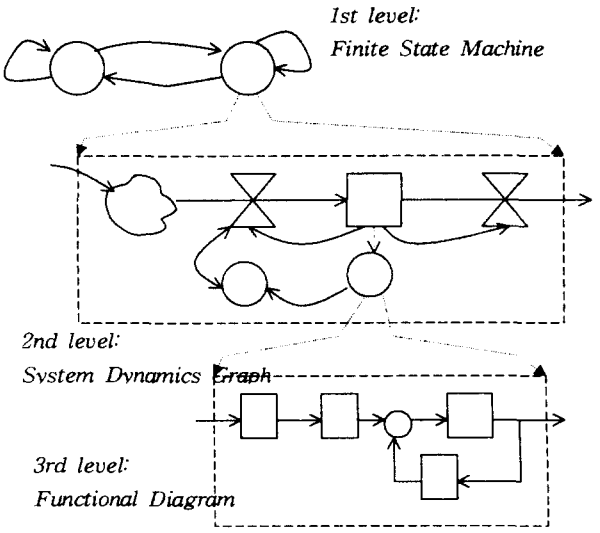

그림 1. 다중 모델링의 예

각 방법론에 대한 자세한 내용은 참고문헌 [5]을 참조한다.

2.2 수직적 모델링

수직적 모델링은 수평적 모델링에서 구축된 모델의 각 요소들을 필요에 따라 세분화 하는 과정이다. 이때 수평적 모델링 과정에서 어떤 표현방법을 택했는가에 따라 수직적 모델링의 적용 대상이 정해 지게 된다. 예를 들어 수평적 모델링에서 유한 오토마타를 사용하였을 경우, 수직적 모

델링에서 세분화 되어질 수 있는 부분은 1) 상태(state)와 2) 상태전이 (state transition)이다. 또한 수식 모델링에 의해 수직적 모델링에서 세분화 되어 질 수 있는 부분은 상태변수를 제외한 나머지 수식항이 해당된다. 일단 세분화 가능한 요소들이 파악된 후, 앞서 수평적 모델링 과정을 반복적으로 수행한다. 수직적 모델링은 입력/출력 간의 맵핑을 통해 서로다른 표현 형태의 모델들이 하나의 모델화 될 수 있도록 하는 메카니즘을 제공한다. <그림 1>은 모델링 과정을 도식화 한 것이다. 첫 번째 단계에서는 시스템을 2가지 상태를 가지는 유한 오토마타로 묘사 한다. 이때 한 상태를 세분화할 필요가 있을 때, 세분화된 사항을 효과적으로 표현할 수 있는 같은 형태 혹은 다른 형태의 모델을 결정한다. <그림 1>은 두 번째 단계에서 시스템 운동도를 이용하였고, 3번째 단계에서는 시스템 운동도의 일부 요소를 다시 기능도를 이용하여 세분화하였다. 이 과정에서 서로 다른 단계에 위치한 모델간에는 입력/출력 관계를 통하여 연결시킨다. 본 연구에서 설명한 모델링 방법론에 대한 자세한 내용은 참고문헌 [4-5]에서 찾아볼 수 있다.

3. 제한조건이 있는 시뮬레이션을 위한 방법론

제한조건을 해결하기 위한 시뮬레이션은 2장에서 살펴본 다중추상화 모델링 방법의 특징을 고려할 때 다음과 같이 재정의 되어질 수 있다. 즉, 제한조건을 만족하는 시뮬레이션이란, 주어진 제한조건을 해결하기 위한 최상의 추상화 수준을 결정한 후, 결정된 추상화 수준에 따라 모델을 재구성하는 문제이다.

3.1 제한조건 해결을 위한 추상화 수준 결정 방법

시스템이 가지고 있는 제한조건을 C1, C2, ..., Cn이라 하자. 본연구에서는 주어진 제한조건을 만족시키는 추상화 수준을 선택할 때 다음과 같은 사항을 고려한다.

고려사항1. 수평적, 수직적 모델링을 통해 만들어진 모델 요소들을 최대로 보전한다. 즉, 주어진 제한 조건을 만족시키는 한도에서 모델링 과정에서 생성된 정보가 최소로 손실되도록 한다.

고려사항2. C1, C2. ..., Cn의 모든 제한 조건이 만족되어야 한다

추상트리(Abstraction Tree)는 모델링 정보를 표현하기 위한 데이터 구조이다. AT에서 Pi가 Q1, Q2, ..., Qm으로 수직적 모델링 되었을 때 Pi는 parent node로, Qi는 child node로 구성된다.
<고려사항1>은 제한사항을 만족시키기 위해 구축된 모델에서 추상화 수준을 높여야 한다면, AT상에서 임의의 노드의 child수가 가장 적은 것을 선택하여 손실되는 정보가 최소화 되도록 한다. <고려사항2>는 <고려사항1>을 해결할 때 동시에 만족해야 하는 사항들을 정의한다. 이상과 같은 내용을 구체화하면 다음과 같다.

$$Minimize \sum_{i=1}^{l} \sum_{j=1}^{mi} Eij * Dij \quad \text{(Eq.1)}$$

subject to

$$\sum_{i=1}^{l} \sum_{j=1}^{mi} C_1 ij * Eij \le C1 \quad \text{(Eq. 2)}$$

$$\sum_{i=1}^{l} \sum_{j=1}^{mi} C_2 ij * Eij \ge C2$$

$$\vdots$$

$$\sum_{i=1}^{l} \sum_{j=1}^{mi} C_n ij * Eij \ge Cn$$

$$\sum_{k=1}^{mi} Ei+1,k \le mi(1-Eik) \text{ for each i} \quad \text{(Eq. 3)}$$

where

Eij: AT에서 i번째 level의 j번째 node
Dij: AT에서 Eij에 종속된 child node의 개수
Cₖij: AT에서 Eij에 Cₖ의 조건에 대한 손실/손익 분Ci: i번째 제약조건의 임계치
mi: i node의 자식수

Eij 변수는 이진변수로 AT상에서 대응되는 node가 주어진 제약조건을 만족하기 위해 추상화 시켜져야 하는 경우(즉, 버려져야 하는 경우)에는 1

의 값을 갖고, 그 외의 경우는 0의 값을 갖는 이진변수이다. (Eq. 1)은 목적함수로, 제약조건을 만족하는 추상화 수준을 정할 때는 AT상에서 손실되는 노드의 수를 최소화해야 한다는 사실을 반영한다. (Eq. 2)에서는 주어진 각각의 제약조건을 설정한다. 즉 좌변은 임의의 추상화 수준을 택했을 경우 예상되는 손실분이며 우변은 손실분의 한계치를 정의한다. 각 제약 조건의 성격에 따라 부등식이나 등식의 형태로 정의되어 진다. (Eq. 3)은 parent에서 추상화가 이루어 질 경우, parent의 child가 중복적으로 추상화 이루어지는 것을 방지하기 위한 수식이다.

제약조건을 해결하는 추상화수준은 결국 (Eq. 1)의 목적함수를 (Eq. 2)의 집합과 (Eq. 3)를 동시에 고려하여 풀 때 결정되어 진다. 정의된 수식은 Integer Programming 기법으로 풀이될 수 있다. Integer Programming은 OR(Operations Research)에서 최적화 문제를 위해 쓰는 방법중의 하나로, 변수들이 정수값을 갖는 것으로 목적함수가 정의된 부류이다.[7,8] Integer Programming 기법에 의해 1의 값을 갖는 Eij를 가려내면 AT상에서 주어진 제한 조건을 해결하기 위해 생략되어져야 할 모델 요소를 가려내게 된다.

3.2 추상화 수준에 따른 모델의 동적 구성

3.1의 단계를 통해 얻어진 추상화 수준은, 시뮬레이션을 위한 모델을 바로 생성하지 못한다. 이것은 앞서 2장에서 살펴본 바와 같이, 제안된 모델링 방법론은 계층적인 성격을 띄기 때문이다. 즉, 임의의 한 level에서 시뮬레이션이 수행될 때는 바로 하위 level에 의해 실행에 필요한 정보가 제공되는 구조이기 때문에, 결국 어떤 추상화 수준으로 우리가 시스템을 취하던 실제 적인 수행은 최하위 leaf 노드에서

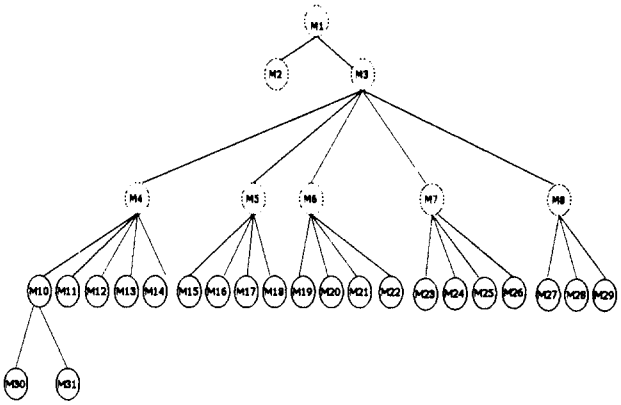

그림 2. AT의 예

일어난다. 따라서 결정된 추상화 수준에서 바로 시뮬레이션이 수행될 수 있도록 하기 위해서는 결정된 수준의 하위 모델 요소를 대치할 수 있는 간단한 모델이 필요하다.

System Identification, Neural Network, Wavelets는 학습능력을 이용하여 주어진 시스템의 행동양식을 모방, 예측할 수 있다.[9] 본 연구에서는 System Identification, Neural Network, Wavelets을 이용하여 결정된 추상화 수준의 하위 level이 갖는 행동(behavior)을 생성해 내는 black box 모델을 만든다. 생성된 System Identification, Neural Network, Wavelets 모델은 시뮬레이션 수행 시 주어진 추상화 수준 이외의 부분에 대한 정보를 제공한다. 이와 같은 모델들은 일단 학습이 끝난 후에는 주어진 입력에 대한 해답을 결정하는 시간이 매우 짧기 때문에 실제 시뮬레이션 수행시간에는 거의 영향을 미치지 않는다. 따라서, 3.1절에서 결정된 임의의 추상화 수준에서 시뮬레이션이 즉각 수행될 수 있는 효과적 인 수단을 제공한다.

4. 실험

실시간 시스템은 제한 조건이 있는 시스템의 대표적인 예로, 실시간 시스템에 대한 시뮬레이션은 반드시 주어진 시간 안에 그 결과를 생성해야 한다. 실시간 시스템에 부가되는 제약 조건은 크게 1) 반응시간, 2) 정확도의 두 개가 있다. 즉 주어진 반응 시간을 만족시키는 범위에서 최상의 정확한 결과를 생성해 내야 한다.

<그림 2>는 실험을 위해 임의로 구성한 AT를 보여 준다. <그림 2>에서 M4, M5, M6, M7, M8에서 추상화가 일어날 수 있다고 하자. 이때 M4 - M8에서 추상화가 일어날 경우 일어나는 시간의 단축 정도 및 정확도의 손실이 <표 1>과 같이 알려져 있다 하자.

표 1. 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

표 2. 예에 대한 추상화 수준 결정 결과

	50	90	200	650	700
Speedup	M5	M5	M4	M4,M5	M4,M5,M6
손실도	0.55	0.55	0.59	1.14	1.59

M1 - M31까지는 2장에서 제시한 모델링 방법에 따라 구축되었기 때문에 각기 다른 형태를 취하고 있다. 이때 650만큼의 speedup을 얻는 경우 중 가장 작은 error를 갖는 추상화 수준을 결정한다고 하면 (Eq 1) - (Eq. 3)에 의해 다음과 같이 정형화된다.

Minimize(7*M4 + 4*M5 + 4*M6 + 4*M7 + 3*M8)

subject to

t4*M4 + t5*M5 + t6*M6 + t7*M7+ t8*M8 >= Tmax

a4*M4 + a5*M5 + a6*M6 + a7*M7+ a8*M8 <= Amax

t4 = 592, t5 = 100, t6 = 19, t7 = 31, t8 = 22

a4 = 0.59, a5 = 0.45, a6 = 0.43, a7 = 0.47, a8 = 0.07

Tmax = 650, Amax = 5.0

위에서 정의된 Integer Programming 모델을 서로 다른 speedup 값에 따라 풀면 <표 2>와 같은 결과를 얻을 수 있다. 예를 들어 650의 speedup이 필요한 경우, 즉 본래 모델의 시뮬레이션 시간보다 650만큼 시간 단축이 필요한 제약조건이 있는 시뮬레이션의 경우에서는, M4와 M5에 대응되는 모델요소를 추상화시킬 경우 가장 정확한 시뮬레이션 결과를 제시할 수 있음을 <표 2>에서 보여 주고 있다. <그림3>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.

5. 결론

본 연구에서는 계층적 구조를 띈 다중 추상화 모델을 구축하는 방법과, 주어진 제약조건을 만족시키는 임의의 추상화 수준을 선정하는 방법을 살펴보았다

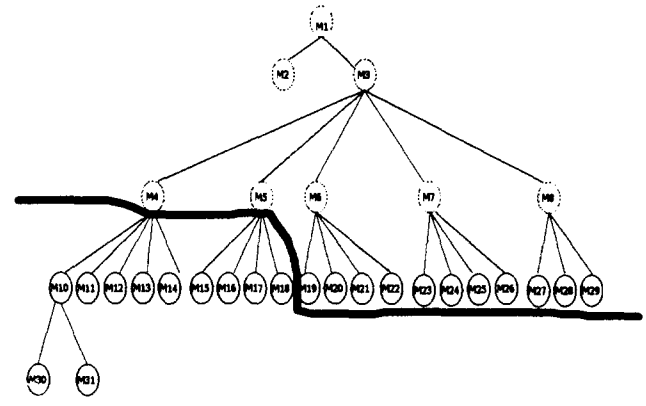

그림 3. 650 speedup을 위한 추상화 수준

결정된 추상화 수준은 제약조건이 있는 시뮬레이션을 위해 모델을 재구성할 때 이용됨을 보였다. 본 논문에서 제시된 방법은 또한 시스템의 요구사항 분석에 응용될 수 있다. 즉, 시스템에 대한 모델을 트리 형태로 계층적으로 표현하고, 주어진 요구사항을 제시된 수식 모델을 통해 정형화한 후, 구축된 모델에서 각 요구 사항을 만족시키는 추상화 수준이 존재하는 가를 검사함으로써 요구사항 분석을 수행할 수 있다. 또한 제시된 방법을 모델 설계단계가 아니라 실제 수행 중에 적용하여, 외부의 변화 조건을 반영할 수 있는 모델을 동적으로 재구성 하는데 사용할 수 있다. 실시간 물체 추적 시스템은 대표적인 응용 예이다. 그러나, 모델링 단계가 아닌 실행단계에서 동적으로 추상화 수준을 결정하고 이에 따라 모델

의 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.[10,11] 또한 임의의 추상화 수준에서 시뮬레이션이 일어나도록 하기 위해 본 논문에서는 하위 수준의 모델 요소들을 하나의neural network등의 모델로 대치시키는데, 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 학습하는데 필요한 최상의 학습 파라미터들을 자동적으로 산출하는 방법이 부가적으로 연구되어져야 한다.

참고문헌

[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] 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

[5] Paul A. Fishwick, "Simulation model design and execution: Building digital worlds", Prentice Hall, 1995

[6] 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

[7] Don Phillips Ravindran and James J. Solberg, "Operations Research", John Wiley and Sons, 1987

[8] "CPLEX: Using the CPLEX Callable Library", CPLEX Optimization Inc., 1995

[9] Timothy Masters, "Neural, novel and hybrid algorithms time series prediction", John Wiley and Sons, Inc., 1995

[10] Kangsun Lee and Paul A. Fishwick, "A multimodeling methodology for real-time simulation", Transactions on Modeling and Simulation, 2000

[11] Geoffrey W. Rutledge, "Dynamic selection of models", Ph.D dissertation, Department of Medical Information Sciences, Stanford University, 1995