

이산사건 시물레이션에 기반한 전자상거래 모델링/해석 소프트웨어 개발

윤근수*, 안명수**, 이준원***

*안동대학교 컴퓨터공학과

**(주)넷커스터마이즈

***안동대학교 전자정보산업학부

E-mail : newradio@netcus.com

최근들어 B2C, B2B로 불리는 인터넷 기반의 전자상거래 시스템의 성공적인 구축이 기업의 비즈니스 성공의 열쇠로 간주될 만큼 그 중요성이 커지고 있다. 본 논문에서는 상거래 행위에 개입하는 다자간의 비즈니스 플로우를 효과적으로 모델링하여 분석할 수 있는 DEVS(Discrete Event Modeling System) 이론에 기반한 새로운 비즈니스 모델링/해석 방법론을 제안한다. 제안하는 방법은 비즈니스 플로우를 기술할 수 있는 언어를 정의하고, 정의된 언어를 이용하여 기술되는 전자상거래에 개입하는 개체간 거래 행위를 DEVS 모델로 자동 변환하는 방법을 제시한다. 변환된 모델은 이산사건 시물레이션 환경인 rDEVSim++ 시물레이션 소프트웨어를 이용하여 실행된 후에 그래픽 인터페이스 환경에서 비주얼화되어 시스템의 동작 및 성능을 분석 가능하도록 지원한다. 위의 전과정을 포함하도록 개발된 소프트웨어는 기업의 비즈니스 모델 구축용으로 활용이 가능할 뿐만 아니라 전자상거래 관련 인력 양성을 위한 교육용 소프트웨어로도 활용이 가능하다.

1. 서론

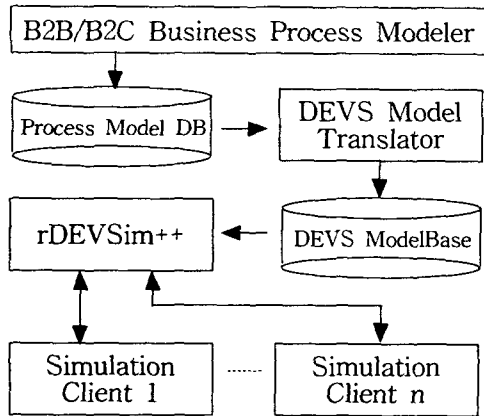
인터넷을 이용한 비즈니스 모델의 근간을 이루고 있는 B2B, B2C 등의 전자상거래 시스템은 거래 행위에 다수의 개체를 개입시키는 것이 특징이다. 다수의 물품 공급업자와 판매자, 그리고 결제자, 배송자들이 거래에 개입하게 되며 일반 이용자나 기업 구매자가 물품을 구매하게 된다. 따라서, 각 개입자 사이의 적절한 거래 업무의 흐름이 전자상거래 시스템의 성능향상은 물론 구매자의 편리성에 크게 기여할 수 있게 된다.

현재까지 국내에서는 이러한 시스템을 체계적인 기법을 적용하여 설계하거나 최적화하기 위한 방

법론에 대한 연구가 미약하며 개발 중이거나 상품화된 제품도 거의 없는 실정이다. 외국의 경우 비즈니스 모델을 평가/분석하기 위한 소프트웨어가 개발되어 활용되고 있다.

본 논문에서는 이러한 점을 고려하여 체계적인 시스템 모델링 및 시물레이션 이론에 근간을 둔 전자상거래 모델링 및 시물레이션 방법론을 제안하고, 제안하는 방법론을 소프트웨어로 구현한다.

제안하는 기법은 그래픽 환경으로 비즈니스 프로세스를 모델링하여 기술할 수 있는 프로세스 기술 언어를 적용한다. 기술된 프로세스를 이산사건 시물레이션 기법을 적용하여 컴퓨터상에서 수행할



<그림 1> B2B/B2C 모델링 해석 S/W 구조

수 있도록 시물레이션 환경하의 시물레이션 모델로 자동 변환할 수 있는 모델 변환기를 개발하였다. 모델 실행은 DEVS 형식론에 기반한 모델링 시물레이션 환경을 실시간으로 확장한 rDEVSIM++ 프로그램을 이용하여 실행하였으며 실행 결과는 다시 그래픽 화면에 모사 함으로써 거래 개체간의 거래 행위 및 거래 시스템의 성능을 평가 할 수 있도록 지원한다. 제안하는 방법론 및 소프트웨어는 전자상거래 시스템 구축을 위한 비즈니스 프로세스 개발은 물론 교육용 소프트웨어로 활용이 가능하다. 그림 1은 본 논문에서 제안하는 방법론을 보여 주고 있다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 모델링 환경 및 프로세스 기술 언어에 대하여 설명한다. 3절에서는 시물레이션 모델로 자동 변환하는 변환 알고리즘에 대하여 설명한다. 시물레이션 방법론 및 시물레이터 환경에 대한 설명이 4절에서 기술되며 5절에서 본 논문을 결론 짓는다.

2. B2B/B2C 비즈니스 프로세스 모델링

전자상거래 시스템을 모델링하기 위한 국제적인 표준 언어는 아직 제정된 사례가 없다. 본 논문에서는 상거래 시스템을 모델링하기 위하여 개체간 interaction을 그래픽하게 표현할 수 있고 거래에

필요한 정보는 거래자 정보를 전송하기 위한 ECML(Electronic Commerce Modeling Language)[1]과 비즈니스 프로세스를 기술하기 위하여 사용되는 WPDL (Workflow Process Definition Language)[4]를 사용하도록 한다.

현재, ECML은 단순하게 거래자 정보를 전송하는 규약을 정하고 있다. ECML에서 정하고 있는 기술은 선적지 주소, 도착지 주소, 대금 지급지 주소, 지급 카드 명세에 관한 정보만을 포함하고 있으므로 전자상거래를 위한 비즈니스 프로세스를 충분히 표현하지 못한다. 따라서, WPDL을 이용하여 프로세스를 기술하고 프로세스간에 요구되는 데이터 전송을 위해 ECML을 응용하도록 한다.

WPDL은 비즈니스 워크플로우 프로세스를 정의 위해 다음과 같은 인터페이스를 정의하고 있으며 본 논문에서 개발된 모델링 환경에서는 Block Diagram 편집기를 이용하여 상거래 프로세스를 모델링 할 수 있다.

WORKFLOW <process id>

<Workflow Process Definition Header>

[<extended library declaration>]

[<formal parameters>]

[<Access restriction part>]

<Activity List>

<Transition Information List>

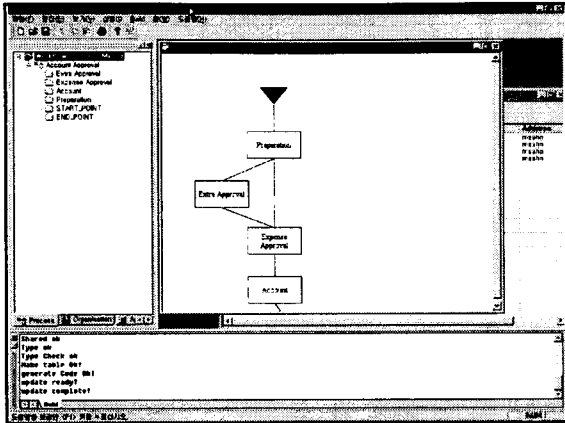
[<Workflow Participant Specification>]

[<Workflow Application List>]

[<Workflow Relevant Data List>]

END_WORKFLOW

그림 2은 Block Editor 형태의 모델링 환경을 보여 주고 있으며 개발된 프로세스 DB에 저장된다. 프로세스 모델을 DEVS 모델로 변환하는 알고리즘은 다음절에서 기술된다.



<그림 2> 프로세스 모델링 환경

3. DEVS 모델 변환 알고리즘

3.1 이산사건 모델링을 위한 DEVS 형식론

Zeigler에 의해 소개된 DEVS 형식론은 이산사건 시스템을 모듈로 나누어서 이를 계층적으로 모델링할 수 있는 수학적 기반을 제공한다. DEVS 형식론으로 가장 작은 모델인 atomic 모델을 기술하고, 이러한 기본 모델을 모듈화, 계층적인 방법으로 연결하여 복잡한 coupled 모델을 기술하게 된다.

Atomic 모델은 가장 기본적인 모듈로서 입력 사건 집합, 출력 사건 집합, 상태 집합, 모델의 다음 상태를 기술하는 전이함수들과 출력을 내기 위한 출력함수 및 시간 함수로 구성되며, 수학적으로 표현하면 다음과 같다.

Atomic 모델은 입출력 단자를 통하여 다른 모델들과 상호 통신하며 동작한다. 상태변수 집합(S)은 모델이 가질 수 있는 모든 가능한 상태를 포함한다. 모델의 상태는 외부에서 입력 사건을 받았을 때나 현 상태에서 내부적으로 정해진 시간이 경과했을 때 변화되는데 각각 외부전이 함수(δ_{ext})와 내부전이 함수(δ_{int})에 의해 상태전이 규칙이 정해진다. 시간전진 함수(ta)란 외부 입력 사건없이 한 상태에서 머물 수 있는 시간으로 정의되며 이 시간이 경과하면 내부전이(internal transition)를 겪게 된다. 출력 함수(Y)에서는 각 상태에서 어떠한

출력 사건을 발생시키는가를 정한다.

계층적으로 분해되어 모델링된 시스템의 구성요소 모델들을 서로 결합하는 기능은 두번째 모델 형식인 coupled 모델에 의해서 제공된다. Coupled 모델은 시스템의 각 구성요소 모델들이 서로 어떻게 연결되어 신호를 교환하는지를 기술한다. Coupled 모델은 atomic 모델들 또는 coupled 모델들을 그의 구성요소로 가지며 각 구성요소들과 요소들간의 연결상태를 표현한다. DEVS 이론에 대한 자세한 내용은 [5]를 참조하면 된다.

3.2 모델 변환 알고리즘

GUI 모델링 환경을 이용하여 모델링된 전자상거래 시스템은 시뮬레이션을 위하여 DEVS 모델로 변환된 후에 실제 시뮬레이션 프로그램에서 사용 가능한 실행 파일로 천이 된다.

모델 변환은 다음과 같은 규칙을 통해 이루어진다.

- (1) 하나의 Activity는 하나의 DEVS Atomic model로 변환된다.
- (2) 하나의 Transition은 하나의 DEVS Atomic model로 변환된다.
- (3) 하나의 Workflow는 하나의 DEVS Coupled model로 변환된다.
- (4) 하나의 Workflow Model은 하나의 DEVS Coupled model로 변환된다. 이때, 이 Coupled model은 DEVS의 model tree 중에서 root에 해당하는 최상위 model이 된다.
- (5) Relevant Data는 DEVSIF에서 제공하는 shared 변수로 변환된다.

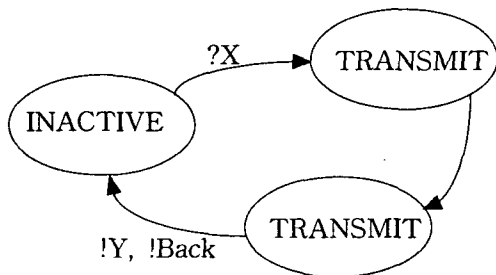
위의 규칙을 적용하여 DEVS 모델로 변환되는 과정에서 WPD의 Application 영역에서 정의된 함수는 외부 함수를 호출하여 실행되도록 변환된다. 이와 관련된 방법은 4장의 시뮬레이터 환경에서 기술된다. 모델 변환 과정을 기술하기 위하여

거래 개체간 프로세스 천이(Transition) 과정을 DEVS 모델로 변환하는 과정을 소개한다. 하나의 Transition은 그림3과 같은 3가지 상태 천이 모델을 갖는 Atomic 모델로 변환된다.

INACTIVE상태에서는 입력되는 이벤트의 처리할 지 여부를 결정한다. 처리 여부는 이벤트 목적지 이름이 자신의 이름과 같거나 'all' 이라면 ACTIVE 상태로 진행한다. ACTIVE 상태에서는 TRANSMIT 상태에서 출력할 변수를 설정하고 바로 TRANSMIT 상태로 천이한다. TRANSMIT 상태에서는 다음에 활성화될 Activity에 해당하는 Atomic 모델에 외부 사건을 전달하게 된다. 다음은 그림 2에서 모델링된 거래 모형중 첫 번째 Transition을 Atomic 모델로 변환한 결과이다.

```
atomic model CYEAR_REPORT_t3
state variables:
  "phase in { INACTIVE, ACTIVE, TRANSMIT };"
  return in TOKEN;
initial condition:
  phase := INACTIVE;
  return := new TOKEN;
internal transition:
  (phase=ACTIVE) => {
    return.value := 'TRUE';
    phase := TRANSMIT;
  }
  (phase=TRANSMIT) => { phase := INACTIVE; }
external transition:
  ((phase=INACTIVE)&&((X.value='t3')||(X.value='all'))
  * X => { phase := ACTIVE; }
output function:
  (phase=TRANSMIT) => { Y := X; }
  (phase=TRANSMIT) => { Back := return; }
time advance:
  (phase=INACTIVE) => ta:=infinity;
  (phase=ACTIVE) => ta:=1;
  (phase=TRANSMIT) => ta:=1;
end CYEAR_REPORT_t3;
```

위와 같이 변환된 DEVS 모델은 모델베이스에 저장 관리되게 되며 시뮬레이터에게 특정 프로세스에 해당하는 모델 실행을 요청하면 DEVS 시뮬



<그림 3> Transition 모델의 상태 천이도

레이션 엔진에 의해 시뮬레이션이 수행되게 된다. 시뮬레이션 엔진에 대한 자세한 설명은 다음 장에서 이루어 진다.

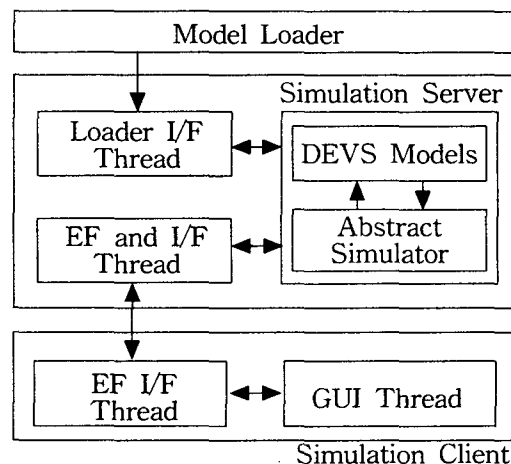
4. 전자상거래 프로세스 시뮬레이터

DEVS 모델을 실행하는 시뮬레이터는 다음과 같은 3가지 프로그램으로 구성되어 있다.

- 모델 Loader : 실행하고 하는 프로세스 모델을 모델 Executor에게 전달하는 기능을 수행한다. 사용자는 원하는 프로세스 모델을 모델베이스로부터 선택하여 통신 프로토콜을 이용하여 Executor에게 DEVS 모델 description을 전달하고 실행을 요청한다.
- 시뮬레이션 서버 : 실시간 DEVS 시뮬레이션 알고리즘을 구현하는 프로그램이다. Loader가 요청하는 프로그램을 시뮬레이션 Client와의 통신을 통해 원하는 상태천이를 하도록 한다.
- 시뮬레이션 Client : 전자상거래 모델에서 각 거래 참여자가 수행하는 activity를 입력하기 위하여 구동되는 프로그램이다.

그림 4는 위의 3 가지 구성 요소간의 인터랙션을 보여준다.

모델베이스에 저장되어 있는 DEVS 모델을



<그림 4> 실시간 시뮬레이터 구조

executor에서 실행하기 위해서는 loader와 executor 사이의 통신 프로토콜에 의해 사용자가 선택한 모델에 관한 데이터를 executor로 upload 시켜 주어야 한다. 모델 loader는 시물레이션 엔진 소프트웨어와 독립적으로 동작하면서 사용자의 선택을 처리하게 된다.

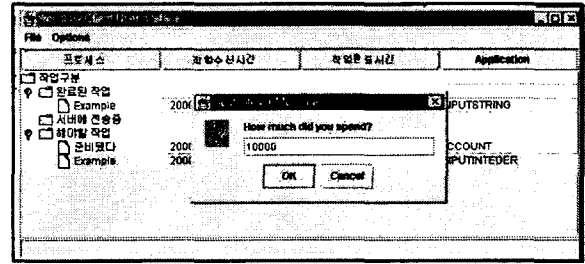
시물레이션 서버는 DEVS 모델을 실행하기 위한 abstract simulator algorithm[4]을 실시간 상에서 구현한 것이다. 프로세스를 기술하는 activity에 대응되는 application을 사용자가 직접 수행하는 방식으로 시물레이션이 진행되므로 application을 구동하기 위한 EF(Experimental Frame)thread와 시물레이션을 제어하기 위한 abstract simulator thread가 독립적으로 수행되는 multi-thread 방식을 적용하여 구현되었다.

모델 Agent는 모델 executor가 요청하는 application을 사용자가 처리할 수 있도록 요구하는 기능을 수행한다. 그림 5는 시물레이션 Client의 동작 상태를 보여 주고 있다.

5. 결론

본 논문에서는 전자상거래 시스템을 이산사건 시물레이션 기법을 적용하여 모델링하여 시물레이션할 수 있는 방법론을 제안하였다. 제안하는 방법론은 다음과 같은 특징과 활용도를 갖는다.

- 거래 개체간 전자상거래 거래 모형을 정의할 수 있는 프로세스 기술 언어인 WPDL을 적용하였다.
- 전자상거래를 모형화하기 위한 ECML과 개체간 interaction을 분리하여 모델링할 수 있는 프레임워크를 구축하였다.
- 언어로 기술된 비즈니스 프로세스를 이산사건 모델로 변환하는 모델 변환 알고리즘을 개발하였다.



<그림 5> 시물레이션 Client

- 변환된 모델을 실시간 환경으로 실행하기 위해 이산사건 시물레이터를 활용하였다. 따라서, 프로세스의 진행을 거래 참여자들이 실제 실행할 수 있는 환경을 개발하였다.
- 프로세스 정의, 실행, 분석을 모두 지원하는 통합 소프트웨어를 개발하였다.

본 과제에서 제안하는 방법론과 소프트웨어는 다양한 비즈니스 프로세스를 모델링하여 시물레이션할 수 있는 환경으로 확장중에 있으며 전자상거래 관련 인력 양성을 위한 교육용 소프트웨어로 활용하기 위해 추가 개발을 진행하고 있다.

6. 참고문헌

- [1] ECML, Electronic Commerce Modeling Language, 1999
- [2] Myung S. Ahn and Tag G. Kim, DEVSim++ User's Manual, Technical Report, TR-CORE94-1, EE, KAIST, 1994
- [3] Roger S. Pressman. Software Engineering, A Practitioner's Approach, 3rd Ed. McGraw Hill, 1997
- [4] WfMC, Workflow Standards : Interface 1 - Process Definition Interchange V 1.1, 1999
- [5] B.P. Ziegler, Multifaceted Modeling and Discrete Event Simulation, Academic Press, Orlando, FL, 1984