

DEVS 형식론을 기반으로 한 Workflow 관리 시스템 개발

홍기정, 이종권, 김탁곤

한국과학기술원 전자전산학과
시스템 모델링 시뮬레이션 연구실

E-mail : {kjhong,jklee}@smslab.kaist.ac.kr, tkim@ee.kaist.ac.kr

Development of a DEVS-based Workflow Management System

Ki Jung Hong, Jong-Kwon Lee, and Tag Gon Kim

Systems Modeling Simulation Laboratory
Department of Electrical Engineering and Computer Science, KAIST

요 약

Workflow 관리 시스템은 비즈니스 프로세스의 자동화를 지원하는 것으로서, 비즈니스 프로세스의 각 업무 단계에 관련된 인적 자원 및 IT 자원을 적절히 활용하도록 구현되어야 한다. 이러한 workflow 관리 시스템은 workflow 프로세스에 대한 명세로부터 실행 모델이 구성되어 workflow 엔진을 통해 수행되는 과정을 거친다. 한편, 명세된 프로세스의 논리적 검증과 성능 예측을 위하여는 workflow의 수행 전에 시뮬레이션을 통한 분석이 선행되어야 한다. 본 연구에서는 기술된 workflow 프로세스 명세로부터 이산사건 시스템 형식론인 DEVS를 기반으로 통일된 실행모델을 생성하고, 이를 바탕으로 시뮬레이션 분석과 실시간 수행을 하는 시뮬레이션 엔진과 실시간 분산수행 엔진을 각각 개발하였다. 간단한 업무 프로세스의 예를 통하여 분산환경에서 실행함으로써 제안된 방법의 유용성을 검증할 수 있었다.

1. 서론

Workflow란 비즈니스 프로세스의 일부분 혹은 전체를 컴퓨터를 이용하여 자동화한 시스템을 말한다 [1][3][4][5]. Workflow 관리 시스템은 이러한 업무 흐름에 관계된 workflow 분석을 제공하며 이를 통하여 지금까지의 사무용 소프트웨어와는 달리 작업의 흐름을 시스템화할 수 있기 때

문에 업무처리의 방법 자체를 변화시켜 비즈니스 리엔지니어링을 가능하게 한다.

이러한 workflow 관리 시스템은 workflow 프로세스에 대한 명세로부터 실행모델이 구성되어 workflow 엔진을 통해 수행되는 과정을 거친다. 한편 명세된 프로세스의 논리적 검증과 성능 예측을 위하여 workflow의 수행 전에 시뮬레이션

을 통하여 분석을 하는 것이 필요하다.

본 논문에서는 workflow 관리 시스템을 이산 사건 시스템 형식론인 DEVS에 기반하여 개발하는 방법론을 제안하고 있다. Workflow 프로세스 명세로부터 DEVS로 표현되는 실행모델을 생성하고, 이를 바탕으로 시뮬레이션 분석과 실시간 수행을 하는 시뮬레이션 엔진과 실시간 분산수행 엔진을 각각 개발하였다. 아울러, 다른 프로세스 모델링 도구 및 workflow 시스템과의 상호 연동성을 제공하기 위하여, 프로세스 명세를 표준 프로세스 기술언어인 WPD(Workflow Process Definition Language) [2] 로 표현하고, WPD에서 시뮬레이션 분석과 실시간 수행에 필요한 DEVS 모델 변환기를 개발하였다. 이렇게 동일한 실행 모델을 DEVS에 기반하여 사용함으로써 프로세스의 분석과 실제 수행 사이에 생길 수 있는 모델링 과정의 오류를 최소화할 수 있는 장점이 있다. 그림 1은 본 연구에서 개발된 workflow 관리 시스템의 전체 framework을 보여 주고 있다.

본 논문의 2절에서는 workflow 시스템에 대한 간단한 설명을 하고, 3절에서는 DEVS 형식론을 기반으로 개발된 workflow 관리 시스템에 대해 세부적으로 기술한다. 4절에서는 간략한 예제를 통하여 workflow 프로세스를 DEVS형식론을 이용해 모델링하고 시뮬레이션하는 것을 소개한다. 마지막으로 5절에서 결론을 맺는다.

2. Workflow 시스템 개요

Workflow 관리 시스템 (Workflow Management System, WfMS)이란 Workflow를 정의, 관리, 실행하는 전체 시스템을 일컫는다 [1].

비즈니스 프로세스는 구성요소의 복잡성 및 존속 기간에 따라 짧게는 수분에서 길게는 수개월에 이르는 주기를 갖는다. 또 다양한 종류의 Information Technology (IT) 및 통신 환경을 통해 구현될 수 있다. 즉 WfMS는 매우 다양한 형

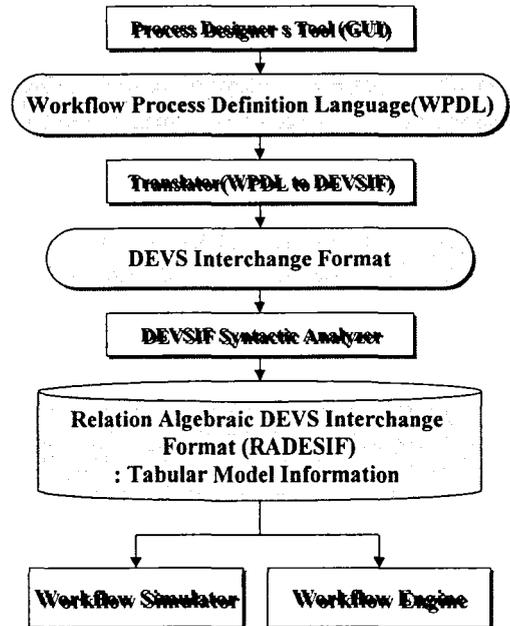


그림 1 DEVS 기반 Workflow 관리 시스템

태로 존재한다. 그러나 이러한 다양성에도 불구하고 WfMS는 몇몇 공통점을 가지고 있다. 그리고 그를 통하여 서로 다른 제품들 사이의 통합 및 협력이 가능하다.

최상위 관점에서 볼 때, WfMS는 다음의 세가지 기능을 제공한다 [1].

- 설계시 기능은 workflow 프로세스 및 그에 따른 activity들의 정의, 모델링에 관련된 기능이다.
- 실행시 제어 기능은 WfMS의 작업 환경에서 각 workflow 프로세스들과 그에 관련된 activity들을 수행하는데 관련된 기능이다.
- 실행시 상호 연동 기능이란 각 activity 단계에서 사용자 및 IT 자원과의 상호 작용을 담당하는 기능이다.

그림 2는 WfMS의 기본적인 특성 및 위 세 기능들 사이의 관계를 보여 준다 [1].

3. DEVS 기반 Workflow 시스템

3.1 DEVS 형식론

DEVS형식론은 계층적 모듈라한 형태로 모델링하는 방법을 제시한다. 복잡한 이산사건 시스템은 계층적으로 분리해서 표시할 수 있는 데 더 이상 나눌 수 없는 가장 작은 개체를 DEVS형식론에서는 atomic 모델이라고 한다. atomic 모델의 수학적 정의 다음과 같다.

$$\begin{aligned}
 AM &= \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle \\
 X &: \text{입력사건집합} \\
 S &: \text{상태집합} \\
 Y &: \text{출력사건집합} \\
 \delta_{int} &: S \rightarrow S: \text{내부상태전이함수} \\
 \delta_{ext} &: Q \times X \rightarrow S: \text{외부상태전이함수} \\
 \lambda &: S \rightarrow Y: \text{출력함수} \\
 ta &: S \rightarrow Real: \text{시간전진함수} \\
 Q &= \{(s, e) | s \in S, 0 \leq e \leq ta(s)\} \\
 &: \text{total state of } AM(e: \text{elapsed time})
 \end{aligned}$$

DEVS의 coupled모델은 계층적인 모델 구조를 기술한다. 이것은 새로운 모델을 구성하기 위해서 컴포넌트 모델을 coupling하여 더 작은 모델에서 더 크고 복잡한 모델을 만들 수 있도록 한다. Coupled모델의 수학적 정의는 다음과 같다.

$$\begin{aligned}
 CM &= \langle X, Y, \{M_i\}, EIC, EOC, IC, SELECT \rangle \\
 X &: \text{입력사건집합} \\
 Y &: \text{출력사건집합} \\
 \{M_i\} &: \text{DEVS컴포넌트집합} \\
 EIC &\subseteq X \times \bigcup X_i: \text{외부입력관계} \\
 EOC &\subseteq \bigcup Y_i \times Y: \text{외부출력관계} \\
 IC &\subseteq \bigcup Y_i \times \bigcup X_i: \text{내부입출력관계} \\
 SELECT &: 2^{\{M_i\}} - \emptyset \rightarrow \{M_i\}
 \end{aligned}$$

DEVS 형식론에 대한 자세한 논의와 모델링은 [6]에서 찾을수 있다.

3.2 DEVSIF소개

DEVSIF(DEVS intermediate format)는 이산사건 시스템의 형식적 모델을 지원하기 위해 개발되었다. DEVSIF는 전체 모델을 기술하기 위해 interface, atomic model과 coupled model이라는 3가지 요소를 가진다. Interface는 atomic model

과 coupled model의 공통적인 입/출력 사건을 기술한다. DEVSIF는 DEVS형식론과 같은 모델링 능력을 가지며 객체지향적 모델링을 지원한다. DEVSIF에서 atomic model을 기술하기 위한 BNF형태의 문법은 다음과 같다:

```

interface model_name [:parent_model_name]
    input : {...}
    output : {...}
end model_name;
atomic          model          model_name
[:parent_model_name]
state variables :{var_name in type_def;}*
initial condition : [expr]*;
internal transition: [(expr)=>(expr);]*
external transition :
    [(expr)*input_event => {[expr;]+}]*
output function : [(expr)=>expr;]*
time advance : [(expr)=>expr]*
end model_name;
    
```

DEVSIF에서 coupled model을 기술하기 위한 BNF형태의 문법은 다음과 같다:

```

interface model_name [:parent_model_name]
    input : {...}
    output : {...}
    
```

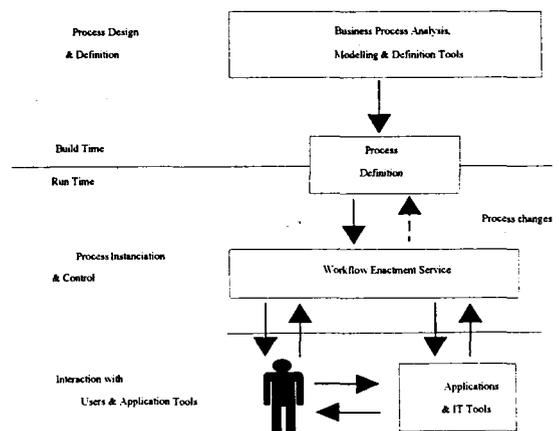


그림 2 Workflow System의 특성

```

end model_name;
coupled          model          model_name
[:parent_model_name]
component : {[child_name in model_name;]+}
    
```

```

external input coupling :
  {[model_name.input_event->
  child_name.child_input_event;]*}
external output coupling :
  {[child_name.child_output_event->
  model_name.output_event;]*}
internal coupling :
  {[src_child_name.src_output_event->
  dst_child_name.dst_input_event;]*}
[select : [{{child_name;}}]]
end model_name;
    
```

DEVSIF model과 DEVS형식론과 비교해 추가된 것은 atomic model에 초기조건을 제외하고 나머지는 일치한다. 초기조건은 실제 시물레이션을 하기 위해 사용된다.

3.3 WPDL to DEVSIF 변환기

WPDL에서 DEVSIF로의 변환은 다음과 같은 알고리즘에 의하여 이루어진다.

- (1) 하나의 activity는 하나의 DEVS atomic model로 변환된다. 단, activity가 subflow로 구현된 경우는 변환하지 않는다.
- (2) 하나의 transition은 하나의 DEVS atomic model로 변환된다.
- (3) 하나의 workflow는 하나의 DEVS coupled model로 변환된다.
- (4) 하나의 workflow Model은 하나의 DEVS

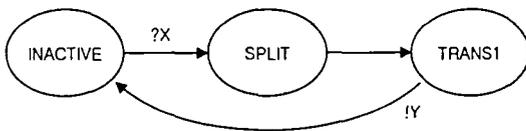


그림 3 State diagram of Route model

Coupled model로 변환된다. 이때, 이 coupled model은 DEVS의 model tree 중에서 root에 해당하는 최상위 model이 된다.

- (5) Relevant data는 DEVSIF에서 제공하는 shared 변수로 변환된다.

Workflow activity는 None, Application, Subflow, Loop, Route의 5가지 종류가 있다. 각

각의 종류에 따라 DEVS atomic model로 변환되

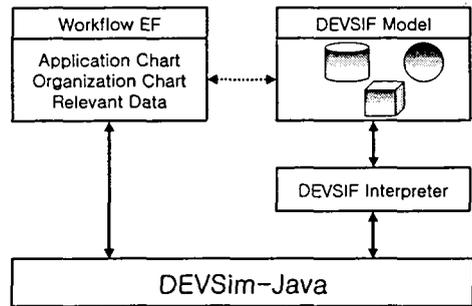


그림 4 Workflow 시물레이션 구조도

는 방식이 달라지게 된다. 아래의 그림은 Route의 변환에 대해서만 설명하고 있으며 Subflow는 DEVS coupled model로 변환한다.

Route의 각 state에 대해서 설명하면, INACTIVE state는 Initial state이다. 아무런 동작 없이 token이 들어오길 기다린다. SPLIT state는 token을 전달하기 위해 token에 path 정보를 수정한다. TRANSI state는 token을 출력한다. 실제로 Route의 state diagram은 입력으로 들어오는 path와 출력되는 path의 개수에 따라 무수히 많아질 수 있다.

3.4 Workflow 시물레이션과 실행

Workflow 시물레이션은 DEVSIM-Java라는 KAIST의 SMSLab에서 개발된 Java언어를 기반한 DEVS형식론 시물레이션도구를 이용하여 수행된다[7]. 그리고 실제 workflow의 실행은 시물레이션 엔진에 실시간 기능과 multi-thread를 사용함으로써 이루어진다. 즉, 같은 모델을 다시 compile하지 않고 시물레이션으로 검증이 끝난후 바로 실행을 할 수 있다는 점이다. Workflow의 application등을 지원하기 위한 workflow EF모델을 개발하였는데 시물레이션과 실제 실행시에 workflow EF모델은 다르게 동작하도록 되어있다. Workflow EF모델은 실제 실행시는 TCP/IP를 이용한 네트워크 연결을 통해 조직도에 의해 업무의 흐름에 맞게 업무관계자에게 작업을 할당하지만 시물레이션시는 업무관계자와의 상호연동이

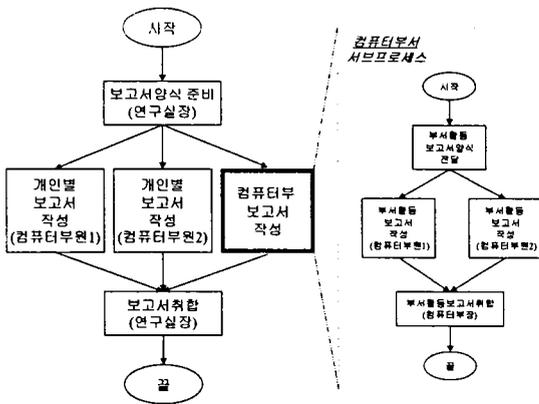


그림 5 연차 보고서 제출 프로세스

없기 때문에 단순한 작업완료 event를 어느 정도의 delay 후에 보낸다. delay는 WPDL에서 모델링할 때 기술할 수 있으며 기술하지 않으면 실제 performance는 시물레이션 시 측정하기 힘들고 단지 실행이전에 논리적 검증의 단계로써 시물레이션을 이용할 수 있다.

4. 예제: 연차보고서 제출 프로세스

어느 대학 연구실에 다음과 같은 제도가 있다. 연말이 되면 연구실 구성원은 모두 연차보고서를 작성하고 연구실장이 그 연차보고서를 모아서 평가하는 것이다. 구성원은 각 개인별 연차보고서도 써야 하지만, 어느 부서에 소속되어 있으면 그 부서에서의 활동에 대한 연차보고서도 써야 한다. 각 부서의 부서장이 부서원들이 작성한 연차보고서를 모아서 연구실장에게 보낸다.

연차보고서의 템플릿은 매년 변하기 때문에 연구실장이 작성하여 구성원들에게는 각각 보내고 부서에는 부서장에게 보내준다. 부서장은 부서원들에게 연차보고서 템플릿을 보내주고 각 구성원과 부서원들은 그 템플릿을 가지고 연차보고서를 작성하게 된다.

위의 그림에 서술된 연차보고서 준비를 위한 activity의 WPDL표현은 다음과 같이 기술된다.

```
ACTIVITY 'PREPARATION'
NAME "PREPARATION"
IMPLEMENTATION
APPLICATIONS TOOL_LIST
```

```
'MSG_LAB_CHIEF_PEOPLE'
'SENDFILE_PEOPLE'
'MSG_LAB_CHIEF_DEPARTMENT'
'SENDFILE_DEPARTMENT'
END_TOOL_LIST
PERFORMER 'LAB_CHIEF'
SPLIT AND END_ACTIVITY
```

기술된 WPDL은 앞에서 기술한 변환 알고리즘에 의해 DEVSIF로 자동 번역되어 다음과 같이 기술된다.

```
interface CPREPARATION
inputs: { X in TOKEN , Back in TOKEN ,
get in WORKITEM };
outputs: { Y in TOKEN , put in WORKITEM };
end CPREPARATION;
atomic model CPREPARATION
state variables:
phase in { INACTIVE, PREPROCESS, ...};
workitem in WORKITEM;
token in TOKEN;
initial condition:
phase := INACTIVE;
workitem := new WORKITEM;
token := new TOKEN;
internal transition:
(phase=PREPROCESS) => {
phase:=REQUEST1;
workitem.ToolName.append 'MESSAGE';
workitem.ToolName.append 'PREPARATION';
workitem.ToolName.append 'LAB_CHIEF';
...
}
...
external transition:
(phase=INACTIVE) * X => {
phase := PREPROCESS;
token := X;
}
...
output function:
...
time advance:
(phase=INACTIVE) => ta:=infinity;
(phase=PREPROCESS) => ta:=1;
...
end CPREPARATION;
```

이렇게 DEVSIF로 바뀐 모델은 RADESIF로 변환되어 DEVSIF Interpreter에 의해 시물레이션과 실시간 실행이 이루어진다[8].

5. 결론

본 논문은 workflow 프로세스를 기술하는 표준 WPDL로부터 이산사건 시스템 형식론인 DEVS를 기반으로 통일된 실행모델을 생성하고, 이를 바탕으로 시물레이션 분석과 실시간 수행을 하는 시물레이션 엔진과 실시간 분산수행 엔진을 각각 개발하였으며 이를 이용한 workflow시스템을 제안한다. 그리고, DEVSIF라는 형식을 제안하고 이산사건시스템 모델의 구축, 검증, 실행하는 프레임워크를 제안한다. 연차 보고서 제출프로세스의 예를 통해 workflow 프로세스를 DEVS형식론에서 제안하는 방법으로 검증할 수 있었다.

참고 문헌

[1] Workflow Managment Coalition Specification TC00-1003, *The Workflow Reference Model*, Nov. 1994.

[2] Workflow Managment Coalition Specification TC-1016-P, *Interface 1: Process Definition Interchange - Process Model*, Nov. 1998.

[3] Stefan Jablonski and Christoph Bussler, *Workflow Management: Modeling Concepts, Architecture and Implementation*, International Thomson Computer Press, London, U.K., 1996.

[4] Adnrzej Cichocki, Abdelsalam (Sumi) Helal, Marek Rusinkiewicz, and Darrell Woelk, *Workflow and Process Automation*, Kluwer Academic Publishers, Massachusetts U.S.A., 1998.

[5] Michael Jackson and Graham Twaddle, *Business Process Implementation: Building Workflow Systems*, Addison-Wesley, U.K., 1997.

[6] B.P Zeigler, *Multifaceted Modeling and Discrete Event Simulation*, Orlando, FL, Academic PRes, 1984

[7] Tag Gon Kim, *DEVSIM++ User's Manual*, SMS Lab KAIST, 1994
<http://sim.kaist.ac.kr/>

[8] 홍기정, 김탁곤, "관계대수를 이용한 이산사건 시스템 모델링", *한국 시물레이션 학회 춘계학술대회 논문집*, pp237-241,1999