워크플로우 모델의 제어 의존성 분석

박 웅[0] 오종태 김광훈 백수기
경기대학교 일반대학원 전자계산학과 그룹웨어연구실
{wpark, jtoh, kwang, skpaik}@kuic.kyonggi.ac.kr

Control Dependency Analysis in Workflow

Woong Park[0] Jong-Tae Oh Kwang-Hoong Kim Su-Ki Paik
Collaboration Technology Research Lab, Dept. of Computer Science, Kyonggi University

요 약

본 논문에서는 워크플로우 모델링 도구인 ICN(Information Control Net) 모델을 기반으로 하는 워크플로우 제어 의존성 분석 메커니즘을 제안하였다. 즉, ICN 모델로 정의된 워크플로우의 각 액티버티들 간에 존재하는 제어 의존 관계를 표현하기 위한 제어 의존 넷(Control Dependency Net)을 정형적인 방법으로 정의하였고, ICN 모델로부터 제어 의존 넷을 생성하는 알고리즘을 정의하였다. 본 논문에서 제안한 워크플로우 제어 의존성 분석 메커니즘은 워크플로우 빌드타임(Build-time) 측면과 워크플로우 런타임(Run-time) 측면에서 중요한 의미를 갖는다. 전자의 측면에서는 워크플로우의 복잡성이 증가함에 따라 더욱 요구되고 있는 워크플로우의 시멘틱 에러 테스팅 기능에 효과적으로 적용될 수 있으며, 후자의 측면에서는 워크플로우의 성공적인 적용을 위해 필수적으로 요구되는 제어 흐름의 동적 변경(Dynamic Change) 지원 기능의 완결성을 향상시키는데 효과적으로 활용될 수 있다.

1. 서론

최근 정보통신 기술의 발달은 조직체에서의 사무업무 처리 프로시져(Business Procedure)에 깊은 영향을 미치고 있다. 최근 몇 년 동안에 많은 사무업무들이 사라지거나, 변형되거나, 완전히 새로운 형태의 업무가 생겨나고 있다. 이러한 사무업무 형태 및 환경의 변화에 대처하고, 보다 효율적인 조직의 운영을 위해 워크플로우와 사무업무 처리과정 리엔지니어링(BPR: Business Process Reengineering) 의 개념이 나타났다. 워크플로우 기술은 크게 일련의 단위업무들로 구성되는 워크플로우를 정의하고 분석하는 워크플로우 모델 분야와 워크플로우의 각 업무들의 실행과 그들 간의 흐름을 제어하는 워크플로우 관리 시스템 분야로 나뉘어진다. 특히, 워크플로우 관리 시스템은 어느 조직체내에서 또는 조직체 간에 구성되어 있는 사무업무 프로시져 또는 사무업무 프로세스(Business Process)를 기반으로 하는 업무의 흐름을 제어하고 자동화 시키기 위한 기술이다. 워크플로우 관리 시스템은 상품의 주문 및 판매 프로세스나 각종 양식(form)을 처리하고 관리하는 전자 결재 프로세스 등과 같은 규칙적인 일상 업무을 자동화할 수 있다는 것 외에도 현재 많은 조직체들이 막대한 투자를 하고 있는 네트웨크 시스템의 이용 및 효용성을 극대화 시킬 수 있는 몇 안 되는 개발 기술들 중 하나이다. 즉, 최근에 조직체 경영 분야에서 활발하게 대두되고 있는 사무업무 처리과정 리엔지니어링의 최종적인 기술지원 체계를 담당하는 것이 바로 워크플로우 기술이다. 워크플로우의 비즈니스 프로세스는 일련의 물리적 또는 논리적 단위의 액티버티(Activity)들과 이들의 수행을 담당하는 역할(Role) 및 참여자(Actor), 그리고 액티버티들 간에 전달되는 문서 또는 정보들로 구성된다. 보통의 비즈니스 프로세스는 많은 액티버티들로 구성되어 있다. 그리고, 비즈니스 프로세스의 진행을 위해 각각의 액티버티는 응용프로그램 및 사용자 그리고 규칙들과 연결된다. 이러한 비즈니스 프로세스 안에서 액티버티 간에 정의되는 실행 순서의 흐름을 제어 흐름이라 한다. 그리고, 각각의 액티버티들은 OR 또는 AND 노드를 통해 선택적 또는 병렬적 실행 제어 순서 관계를 갖을 수 있다. 본 논문에서는 이와 같은 액티버티들 간의 실행 순서에 대한 제어 흐름을 분석하여 각 액티버티들간에 존재하는 실행 제어 의존성 관계를 정의하기 위한 정형적 메커니즘을 제안한다. 즉, 그래프를 통한 워크플로우의 정형적 표현기법인 ICN 모델을 기반으로 정의된 워크플로우 프로세스로부터 액티버티들간의 제어 의존성 관계를 정의하는 제어 의존 넷을 생성하는 알고리즘을 제안하였다. 본 워크플로우 제어 의존성 분석 메커니즘은 워크플로우 관리 시스템의 주요 실행 모듈인 런타임과 빌드타임 모두에서 효과적으로 활용될 수 있다. 즉, 빌드타임 측면에서는 워크플로우 응용분야의 대형화와 복잡성의 증가에 따라 그 중요성이 더욱 증가하고 있는 워크플로우 모델의 테스트 기능에 대한 확장을 도모할 수 있다. 기존의 빌드타임 모듈에서는 정의된 워크플로우에 대해 구문(Syntax) 에러 테스팅 중심의 기능을 지원하고 있으며, 의미적(Semantic) 에러 테스팅을 위해서는 페트리 넷 등과 같은 별도의 분석 도구를

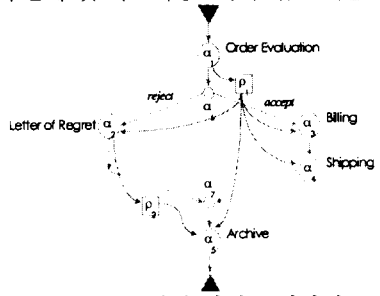
필요로 한다. 그렇지만, 본 논문의 제어 의존성 분석 메커니즘을 활용함으로서 별도의 분석 도구없이도 워크플로우에 대한 효과적인 의미적 에러 테스팅 기능을 지원할 수 있다.

런타임 측면에서는 워크플로우의 적용에 가장 저해가 되는 요소인 동적 변경 지원 기능의 완결성 부족 문제에 대한 해결책을 제공할 수 있다. 워크플로우의 제어 흐름에 대한 동적 변경 지원 기능은 현재 구현하기 가장 어려운 기능중의 하나일 뿐 만 아니라 그 만큼 워크플로우의 적용을 확장시키기 위해 가장 긴요하게 요구되는 기능이기도 하다. 하지만 기존의 워크플로우 관리 시스템에서 제공하고 있는 동적 변경 지원 기능은 제어 흐름에 대한 동적 변경 지원보다는 액티버티에 대한 참여자의 동적 할당 등을 중심으로 지원되고 있다. 그 주요 이유는 워크플로우의 제어 흐름에 대한 동적 변경 지원 기능의 완결성을 보장하기가 어렵기 때문이다. 본 논문에서 제안하는 제어 의존 넷 및 이의 생성 알고리즘을 통해서 워크플로우 제어 흐름의 동적 변경 지원 기능에 대한 완결성을 향상시킬 수 있다.

다음 장에서는 워크플로우 모델링 도구중의 하나인 ICN 모델에 대해 기술하며, 제3장에서는 워크플로우상의 제어 의존성 분석 메커니즘을 정의한다. 마지막으로, 본 메커니즘의 적용을 예를 들어 기술한다.

2. 정형적 ICN(Information Control Net) 모델

워크플로우 모델이란 조직체나 작업그룹에서의 업무 환경과 업무 프로세스을 적절히 표현한 것이다. 워크플로우 모델은 조직의 모습을 할당업무(tasks), 업무자(Actors), 역할, 액티버티(Activity)와 자료저장소의 모습으로 표현한다. 이와 같이 워크플로우 모델은 컴퓨터로 표현될 때 프로시져의 생성, 변화 및 시뮬레이션 작업을 할 수 있다. ICN은 사무실(Office)의 개념을 일련의 관련된 프로시져(Procedures)의 집합으로 정의하며 이러한 프로시져는 전후관계가 존재하는 액티비티들의 집합으로 표현된다. ICN 은 그림 형태로 프로시져, 액티비티, 저장소(Repositor -ies), 전후관계를 나타내는 제어흐름(Control Flow) 과 데이타흐름(Data Flow) 을 표현한다. ICN 제어흐름 그래프는 큰 원으로 표현되는 일련의 액티비티와 작고 빈 원으로 표현되는 OR 노드, 작고 채워진 원으로 표현되는 AND 노드, 그리고 이러한 노드들을 연결하는 선(edge)로 구성된다. 화살표(Arc)는 실선(Solid)과 점선(Dashed)으로 표현되는데 이들은 노드들 간의 선후관계 및 자료저장소와의 입/출력을 표현한다.



<그림 1> 주문처리 관계를 나타내는 워크플로우

기본 ICN 은 4 개의 구성요소인 프로시져, 액티비티, 선후관계(Precedence)와 자료저장소로 구성되며 A를 일련의 액티비티들의 집합이라고 하고, R을 일련의 자료 저장소의 집합이라 할 때 수식적인 표현과 정의는 다음과 같다.

$\Gamma = (\delta, \gamma, I, O)$

δ : precedence constraint among activities

γ : repository input/output requirement of activity

I : initial input repository

O : final output repository

① I 는 초기에 입력되는 자료 저장소들의 유한집합이며 ICN 의 실행 전에 어떠한 외부의 프로세스에 의해서 로드(load)되어져야 한다고 가정한다.

② O 는 마지막으로 출력되는 자료 저장소들의 유한집합이며 ICN 의 실행 후에 어떠한 외부의 프로세스에 의해서 이용되는 정보를 표함하고 있다고 가정한다.

③ $\delta = \delta_i \cup \delta_o$

여기서 $\delta_o : A \rightarrow \wp(A)$은 하나의 액티비티를 후행하는 액티비티들의 집합에 연결하는 관계를 나타내며 $\delta_i : A \rightarrow \wp(A)$은 하나의 액티비티를 선행하는 액티비티들의 집합에 연결하는 관계를 나타내는 관계이다.

④ $\gamma = \gamma_i \cup \gamma_o$

여기서 $\gamma_o : A \rightarrow \wp(R)$은 하나의 액티비티를 후행하는 액티비티 집합들을 출력 자료 저장소들의 집합과 연결하는 것 중 하나이며, $\gamma_i : A \rightarrow \wp(R)$은 하나의 액티비티를 선행하는 액티비티 집합들을 입력 자료 저장소들의 집합과 연결하는 관계를 나타내는 것 중 하나이다.

<그림 1>에 대한 정형적 표현은 아래와 같다.

$\Gamma = (\delta, \gamma, I, O)$

$A = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\}$

$R = \{\rho_1, \rho_2\}$

$I = \{\ \}$

$O = \{\ \}$

$\delta_i(\alpha_1)=((\lambda)),\ \delta_o(\alpha_1)=((\alpha_6));\ \gamma_i(\alpha_1)=(\lambda),\ \gamma_o(\alpha_1)=(\rho_1)$

$\delta_i(\alpha_2)=(\alpha_6),\ \delta_o(\alpha_2)=((\alpha_7));\ \gamma_i(\alpha_2)=(\rho_1),\ \gamma_o(\alpha_2)=(\rho_2)$

$\delta_i(\alpha_3)=(\alpha_6),\ \delta_o(\alpha_3)=((\alpha_4));\ \gamma_i(\alpha_3)=(\rho_1),\ \gamma_o(\alpha_3)=(\lambda)$

$\delta_i(\alpha_4)=(\alpha_3),\ \delta_o(\alpha_4)=((\alpha_7));\ \gamma_i(\alpha_4)=(\rho_1),\ \gamma_o(\alpha_4)=(\lambda)$

$\delta_i(\alpha_5)=(\alpha_7),\ \delta_o(\alpha_4)=(\lambda);\ \gamma_i(\alpha_5)=(\rho_1, \rho_2),\ \gamma_o(\alpha_5)=(\lambda)$

$\delta_i(\alpha_6)=(\alpha_1),\ \delta_o(\alpha_6)=((\alpha_2), (\alpha_3));\ \gamma_i(\alpha_6)=(\lambda),\ \gamma_o(\alpha_6)=(\lambda)$

$\delta_i(\alpha_7)=(\varnothing, \alpha_4),\ \delta_o(\alpha_7)=((\alpha_5));\ \gamma_i(\alpha_7)=(\rho_2),\ \gamma_o(\alpha_7)=(\lambda)$

3. 제어 의존성 분석 메커니즘

ICN 모델을 바탕으로 제어 흐름 의존성을 분석하기 위한 CDN(Control Dependent Nets)을 정의한다. 또한 ICN으로부터 CDN을 생성하는 알고리즘도 고안하였다. 워크플로우의 제어 의존성 분석은 각 액터버티들간의 존재하는 실행 의존 관계를 정의함으로서 구현될 수 있다. 즉, ICN 모델상의 OR 노드와 AND 노드 그리고 Loop 부분이 액티버티들간의 실행 의존 관계를 형성하는데 중요한 역할을 한다.

3.1 용어 정의

CDN 을 정의하기 위해서 ICN 모델의 추가되는 요소가 있다.

(1) ICN에서 W(Walk) 는 액티비트 간에 순서를 나타내며, W 의 길이는 $|W|$ 이다. $|W| = 0$ 이면 액티비티를 아무 일도 하지 않으면 empty 부른다. $|W|$ 가 nonempty 이면 시작 액티비티 u 와 마지막 액티비티 v 의 순서는 $u - v$ 표현한다.

(2)액티비트 간의 우월성

ICN에서는 다음에 오는 조건을 만족해야한다.

① Γ 는 초기의 액티비티 $á_I$ 와 마지막 액티비티인 $á_F$ 포함하면 $ä_I(á_I) = \{ô\}, ä_O(á_F) = \{ô\}$

② Γ 에 모든 액티비트는 $á_I$ - $á_F$ 순서를 실행한다.

- $u \in A$ forward dominates $v \square A$ iff every v - v_F $u \square \Gamma$; u properly forward dominates v iff $u \neq v$ and u forward dominates v
- $u \in A$ strongly forward dominates $v \square A$ iff u forward dominates v 이고 정수 $k \geq 1$ 일 때 $|W| \geq 1$
- $á \square (A-\{á_F\})$ immediate forward dominator $ifd(á)$ 일 때 $á$ - $á_F$ 는 $á$ 의 proper forward dominator

3.2 정형적 CDN(Control Dependent Nets) 모델

CDN은 프로세스의 각 단계에서의 OR노드 또는 AND노드를 설계하기 위해 사용된다. CDN 을 통해서 각각의 액티비티간의 제어 이행 조건(control-transition conditions)을 효율적으로 발생시킬 수 있다. 제어 이행 조건은 액비티비 간의 제어 흐름 및 실행 중에 제어 흐름 동적 변경을 하기 위해 사용된다. CDN은 $\Omega = (\varphi, \kappa^c, S, E)$ 정의할 수 있고, A는 액티비트 집합이고, T는 control-transition conditions 의 집합이다.

- $\varphi = \varphi_i \cup \varphi_o$

φ_o: $A \rightarrow \wp(A)$ 액티비티의 후행하는 액티비티 간의 의존성, $\varphi_i : A \rightarrow \wp(A)$ 는 액티비티 선행하는 액티비티 간의 의존성 관계이다.

- $\kappa^c = \kappa^c_i \cup \kappa^c_o$: control-transition conditions
- S 는 한정된 초기의 control-transition conditions
- E 는 한정된 종료된 control-transition conditions

3.3 CDN 생성 알고리즘

INPUT : An extended ICN
OUTPUT : An Control Denpendencey net
BEGIN
For all x ∈ A and y ∈ A in an extended ICN
IF x is strongly control dependent on y
//OR or Loop Constructs Get a dependent flow
ADD x To φ_o (y) ; ADD y To φ_i (x);
//Get control-transition conditions between x and y
ADD κ^c_i (x) To κ^c_o (y) ;ADD κ^c_i (x) To κ^c_i (x)
ELSE IF fd(y) ∈ x(fd(y)∉fd(ibd(y))OR idb(y) = ϕ)
//Get a dependent flow
ADD x To φ_o (y); ADD y To φ_i (x);
//Get control-transition conditions between y and x
Add κ^c_i (x) To κ^c_o (y) ;ADD κ^c_i (x) To κ^c_i (x)
FI
FI
ROF
End

확장 ICN 을 이용한 CDN 을 생성하는 알고니즘을 구현한다.

3.4 CDN 생성 예

위에 그림은 [그림 1] 주문처리 워크플로우 대한 엑티비트 간에 제어 의존성의 표현했다. 위의 그림을 기초로 정형화된 CDN 모델을 표현했다.

$\Omega = (\varphi, \kappa, S, E)$
$A = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_F\}$
$T = \{defaut, or(hire="reject"), or(hire="accept"), And(default)\}$
$S = \{\varnothing\}$ $E = \{\varnothing\}$

$\varphi_i(\alpha_1) = \{\varnothing\}$ $\varphi_o(\alpha_1) = \{\alpha_1, \alpha_5, \alpha_6, \alpha_7, \alpha_F\}$ $\kappa_i(\alpha_1) = \{\varnothing\}$ $\kappa_o(\alpha_1) = \{d\}$
$\varphi_i(\alpha_1) = \{\alpha_1\}$ $\varphi_o(\alpha_1) = \{\varnothing\}$ $\kappa_i(\alpha_1) = \{d\}$ $\kappa_o(\alpha_1) = \{\varnothing\}$
$\varphi_i(\alpha_2) = \{\alpha_6\}$ $\varphi_o(\alpha_2) = \{\varnothing\}$ $\kappa_i(\alpha_2) = \{or(hire="reject")\}$ $\kappa_o(\alpha_2) = \{\varnothing\}$
$\varphi_i(\alpha_3) = \{\alpha_6\}$ $\varphi_o(\alpha_3) = \{\varnothing\}$ $\kappa_i(\alpha_3) = \{or(hire="accept")\}$ $\kappa_o(\alpha_3) = \{\varnothing\}$
$\varphi_i(\alpha_4) = \{\alpha_6\}$ $\varphi_o(\alpha_4) = \{\varnothing\}$ $\kappa_i(\alpha_4) = \{d\}$ $\kappa_o(\alpha_4) = \{\varnothing\}$
$\varphi_i(\alpha_5) = \{\alpha_1\}$ $\varphi_o(\alpha_5) = \{\varnothing\}$ $\kappa_i(\alpha_5) = \{d\}$ $\kappa_o(\alpha_5) = \{\varnothing\}$
$\varphi_i(\alpha_6) = \{\alpha_1\}$ $\varphi_o(\alpha_6) = \{\alpha_2, \alpha_3, \alpha_4\}$ $\kappa_i(\alpha_6) = \{d\}$ $\kappa_o(\alpha_6) = \{or(hire="reject"), or(hire="apt")\}$
$\varphi_i(\alpha_7) = \{\alpha_1\}$ $\varphi_o(\alpha_7) = \{\varnothing\}$ $\kappa_i(\alpha_7) = \{d\}$ $\kappa_o(\alpha_7) = \{\varnothing\}$
$\varphi_i(\alpha_F) = \{\alpha_1\}$ $\varphi_o(\alpha_F) = \{\varnothing\}$ $\kappa_i(\alpha_F) = \{d\}$ $\kappa_o(\alpha_F) = \{\varnothing\}$

액티비티 1,5,6,7, 출력값은 입력값인 α_1 에 의해서 결정되므로 1,5,6,7, 출력값은 입력값에 제어 의존성을 가지고 있다. 액티비티 2,3,4 는 α_6 OR 분기 의해서 제어 흐름이 이동함으로 α_6 에 제어 의존한다.

4. 결론

본 논문에서는 워크플로우 모델 상에서의 제어 흐름을 분석하기 위해 정형적인 표현기법인 ICN 을 기반으로 정의된 워크플로우 프로세스로부터 액티비티간의 제어 의존성 관계를 정의하는 제어 의존 넷을 생성하는 알고리즘을 제안하였다. 제어 의존 넷 및 이의 생성 알고리즘을 통해서 워크플로우 제어 흐름의 동적 변경 지원 기능에 대한 완결성을 향상시킬 수 있다.

5. 참고 문헌

[1] Clarence A. Ellis and Gary J. Nutt, " Office Information Systems and Computer Science " , Computing Surveys, Vol. 12, No. 1, March 1980.

[2] Clarence A. Ellis, " Goal Based Models of Groupware" , University of Colarado, Boulder, Colorado, USA.

[3] Kwang-Hoon Kim and Su-Ki Paik, "Actor-Oriented Workflow Model ", The Second Cooperative Database Systems for Advanced Applications, Wo llongong Australia, March 1999

[4] Byung-Deuk Yoo , Byung -Ok Jang ,Kwang -Hoon Kimg, Su -Ki Paik , "Data Dependency Analysis in Transactional Work flow"

[5] Hoon Jin , Hak-seiong Kim , Kwang-hoon Kim , Su-ki Paik "A Java-Based ICN Modeling Tool"

[6]Workflow Management Clalition Specification Document, " The Workflow Reference Model ", Document Number : TC))-1003 Version 1.1 ,Jan 1195

[7]Kwang-Hoon Kim , Su -Ki Paik "Actor-Oriented Workflow Model"

[8]Kwang-Hoom Kim "Architectures for Very large Scale Workflow Management Systems"

[9]Hyeong-Seok Hong , Jak-Seong Kim , Kwang-Hoon Kim , Su -Gi Paik "A Transactional Workflow Monitring Tool"