

워크케이스 마이닝을 위한 실행계열분석 알고리즘 설계 (E-Walk Series Analysis Algorithm for Workcase Mining)

백 수 기(Su-Ki Paik)¹⁾

요약

워크플로우 마이닝은 워크플로우 엔진의 실행 정보를 포함하고 있는 워크플로우 로그 파일을 이용하여 워크플로우 모델 즉, 비즈니스 프로세스를 재설계하기 위한 방법을 제공하는 워크플로우 분야에서 새로운 연구 분야이다. 본 논문에서 제안된 워크케이스 마이닝을 위한 실행계열분석 알고리즘은 빌드타임에서 정의된 워크플로우 모델의 액티비티 실행순서와 런 타임에서 실제 실행된 모델의 실행순서와 비교하여 빌드 타임의 워크플로우 모델과 실행 모델과의 차이를 최소화하여 비즈니스 프로세스의 생산성을 향상 및 새로운 비즈니스 모델을 정의할 수 있는 기준을 제시한다. 이와 같은 작업은 빌드타임에 정의된 워크플로우 프로세스를 액티비티 의존넷을 이용하여 최소의 액티비티 집합으로 작성하고, 작성된 최소의 액티비티 집합과 실제 워크플로우 엔진에서 실행된 프로세스의 실행결과를 제안된 실행계열 분석 알고리즘을 이용하여 비교함으로써 수행할 수 있다.

Abstract

Workflow mining is a newly emerging research issue for rediscovering and reengineering workflow models from workflow logs containing information about workflow being executed on the workflow engine. This paper newly defines a workflow process reduction mechanism that formally and automatically reduces an original workflow process to a minimal set of activities, which was used proposed "E-walk series analysis algorithm". Main purpose of this paper is to minimize discrepancies between the workflow process modeled and the enacted workflow process as it is actually being executed. That means, we compare a complete set of activity firing sequences on buildtime with workflow execution logs which was generate on runtime. For this purpose we proposed two algorithm, the one is "Activity-Dependent Net Algorithm" and the other is "E-Walk Series Analysis Algorithm"

논문접수 : 2005. 7. 15.

심사완료 : 2005. 8. 10.

1) 정회원 : 경기대학교 정부과학부 교수

본 논문은 2004학년도 경기대학교 학술연구비지원에 의하여 수행되었음

1. 서 론

어느 시대 또는 어느 형태의 사회라 할지라도 그 사회가 갖는 사람과 사람들 간의 상호 활동과 그들 간의 관계를 지원하기 위한 특징적인 수단과 방법을 제공하고 있다. 컴퓨터 기술과 전자통신 기술의 급진적인 발전과 이러한 기술들 간의 수렴은 바로 전자적인 작업환경이라는 새롭고도 매우 효율적인 상호 작용 지원 수단 및 방법을 잉태하게 되었다. 전자적인 작업환경이란 전(全) 조직체적 통합 시스템으로 정보 처리 활동과 정보 통신 활동의 통합을 통해 조직 내의 구성원들 간의 상호 활동 및 관계를 정의하고 지원하는 개선된 형태의 조직 활동 수단 및 방법이다.

워크플로우 기술이란 바로 이러한 전자적 작업 환경을 구현하기 위한 종합적인 연구 분야로서 컴퓨터 및 통신 분야 뿐 만 아니라 사회학 분야나 언어학 분야, 경영학 분야 등의 다각적인 협력 관계를 통해서만 성공적으로 완성될 수 있는 매우 다중 적인 연구 분야라고 할 수 있다[1,2].

워크플로우는 크게 워크플로우 모델 분야와 워크플로우 관리 시스템 분야로 나눌 수 있고 워크플로우 모델 분야는 일련의 비즈니스 관련 업무들의 흐름(Control Flow)뿐만 아니라 관련 데이터(Data Flow)와 조직 내의 행위자(Actor) 또는 참여자(Participant), 역할(Role) 등을 표현하고 정의하는 수단을 제공하고, 워크플로우 관리 시스템 분야는 워크플로우 모델 분야에서 정의된 업무 흐름을 실제 수행하고 그와 관련된 각종 자원들을 관리함으로써 업무 흐름 전반을 관리 및 제어하는 기능을 수행한다.

그러나 최근에 조직 내 또는 조직 간에 발생하는 비즈니스 프로세스가 복잡해지고, 규모가 커지고, 오랜 수행 시간을 요구하고 있어 비즈니스 프로세스를 디자인하는 작업은 매우 복잡하고, 많은 시간을 요구하는 작업이다. 또한 작성된 비즈니스 프로세스가 워크플로우 관리 시스템에서 정상적으로 수행되리라고 보장 할 수

없다. 즉, 워크플로우 관리 시스템에서 수행되기 위한 완벽한 비즈니스 모델을 정의하는 것이 쉽지 않다는 것을 의미하고 있다.

워크플로우 마이닝이란 "사용자에 의해 정의된 워크플로우 모델과 워크플로우 프로세스를 시스템을 통해 실행된 정보를 이용하여 현재 실행된 프로세스를 평가하고, 기존의 프로세스를 재구성 하거나 워크플로우 프로세스를 생성할 때 도움을 줄 수 있는 데이터를 추출하는 것"과 같이 정의할 수 있다. 워크플로우 마이닝은 실행정보의 마이닝 데이터 - 워크케이스의 실행 시간, 각 워크케이스에 참여한 행위자(actor), 워크플로우 프로시저로부터 발생된 인스턴스, 워크케이스에 할당된 역할 등에 따라 다양한 정보를 제공할 수 있다[3,4,5].

본 논문에서는 빌드타임에서 정의된 모델의 실행경로와 런 타임에서 실행된 워크케이스의 실행경로를 비교/분석하여 새로운 워크플로우 모델의 제시 또는 작성된 비즈니스 프로세스의 개선을 위한 작업으로서 액티비티 의존성을 이용한 워크케이스 마이닝을 위하여 각 액티비티의 의존성을 분석하여 중요경로(Essential Path)를 결정하는 액티비티 의존 넷 알고리즘(Activity-Dependent Net Algorithm)과 각 워크케이스 실행 정보를 이용하여 중요 경로와 일치여부를 결정하는 실행계열 분석 알고리즘(E-Walk Series Analysis Algorithm)을 제안한다.

논문의 구성은 다음과 같다. 2장에서는 이론적 배경으로 워크플로우 마이닝과 관련된 일반적인 사항에 대하여 기술하였고, 3장에서는 본 논문에서 제안된 알고리즘의 설계를 위하여 중요 경로 결정을 위한 액티비티 의존넷 알고리즘과 실행계열 분석 알고리즘을 설계한 후 그 과정을 예를 들어 설명한다. 마지막으로 4장에서 결론 및 향후 연구과제에 대하여 기술 하겠다

2. 이론적 배경

2.1 워크플로우 마이닝 개념

지난 10여년동안 많은 상용 워크플로우 관리시스템 - IBM MQSeries, COSA, Stafware 등등 -은 다양한 기업 정보기술 분야에 적용되어 사용되어 왔다. 그러나 대부분의 워크플로우 관리 시스템 제품들이 처리해야 할 비즈니스 프로세스의 복잡화, 대규모화, 매우 긴 실행시간 등과 같은 조건에 만족하기 위한 유연성과 동적 변경 등을 처리하는데 있어서 많은 문제점을 나타내고 있다[3,4].

현재 대부분의 워크플로우 관리 시스템은 빌드-타임에 정의된 비즈니스 프로세스를 이용하여 런-타임에서 실행시키는 과정이다. 그러나 어떠한 이유로든 두 단계의 정보의 모순 또는 불일치로 인하여 워크플로우의 실행이 더 이상 불가능할 때 발생하는 문제를 처리할 수 있는 유연성을 제공치 못하고 있다.

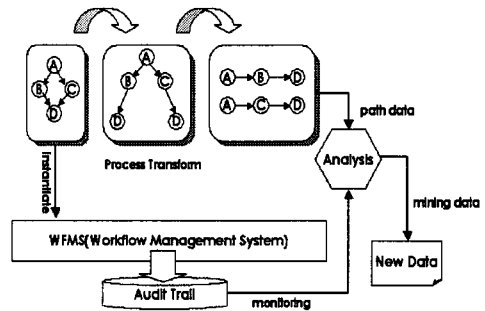
이와 같은 문제를 해결하기 위해서는 실행 프로세스의 흐름정의와 관련된 모든 정보들이 프로세스 모델과 처리 프로세스에서 별도로 관리되어야 한다. 유연성과 관련된 문제, 즉 빌드-타임과 런-타임 정보의 불일치로 발생할 수 있는 문제를 해결하기 위하여 프로세스의 실행된 정보를 이용하여 빌드-타임의 프로세스 정의에 이용하는 방법(reverse the process)이나, 워크플로우 시스템의 런-타임 시점에 빌드-타임에서 정의된 프로세스가 변경되기 위해서는 프로세스의 흐름정의 및 실행된 모든 정보를 알기 위한 방법으로 워크플로우 마이닝이라는 개념이 필요하게 되었다[3,4,5,6].

워크플로우 마이닝 개념은 기업 또는 조직의 프로세스 개선이라는 문제와도 직결될 수 있는 사항으로 워크플로우 마이닝의 개념은 다음 <표 1>과 같고 개념도는 [그림 1]과 같다.

사용자에 의해 정의된 워크플로우 프로세스를 시스템을 통해 실행된 정보를 바탕으로 현재 실행된 프로세스를 평가하고, 기존의 프로세스를 재구성하거나 새로운 워크플로우 프로세스를 생성할 때 도움을 줄 수 있는 데이터를 추출하는 과정

<표 1> 워크플로우 마이닝 정의^[3,4,5]

이러한 목적을 달성하기 위한 [그림 1]에서 보여 지듯이 워크플로우 마이닝 과정은 정의된 비즈니스 프로세스와 실제 실행된 인스턴스의 실행 경로를 비교 분석하는 과정이다. 즉, 빌드타임에서 정의된 비즈니스 프로세스를 분석하여 모든 가능한 경로를 구한다.



[그림 1] 실행결과를 이용한 워크플로우 마이닝 개념

한편 정의된 비즈니스 프로세스가 워크플로우 엔진에서 하나의 인스턴스로 발생되고 실행된 정보를 이용하여 구해진 실제 실행된 경로의 정보를 구한다. 이렇게 얻어진 두 정보를 비교, 분석하여 새로운 실행경로에 대한 유용한 정보를 구할 수 있다[10].

2.2 추상적 워크플로우 마이닝 모델

워크플로우 관리시스템에서 실행정보를 이용하여 일련의 과정을 거쳐 유용한 의미의 정보를 추출해내는 과정인 워크플로우 마이닝 개념은 [그림 2]와 같이 3단계의 추상화된 모델 - *Workflow Walk Driven Model*, *Dependency Driven Model*, *Procedure Driven Model* - 로 표현할 수 있다.

*Workflow Walk Driven Model*은 워크플로우 시스템의 엔진에서 발생된 인스턴스가 실행되면서 워크플로우 마이닝을 위한 데이터인 워크플로우 실행정보를 발생하는 단계이다. 이 때 발생하는 데이터는 뒤에서 설명할 것과 같이

워크플로우 마이닝의 목적에 따라 사용되어질 데이터가 상이할 것이다. 예를 들어 실행되고 있는 데이터의 사용빈도 또는 발생되고 실행되고 있는 인스턴스 내의 각 액티비티의 수행시간 또는 각 액티비티의 실행경로 등과 같은 정보가 실행정보에 저장될 수 있을 것이다.

이와 같이 저장된 정보는 워크플로우 마이닝을 위한 원시데이터가 될 것이다. 현재 실행된 인스턴스의 모든 정보를 가장 정확하게 표현한 실행 정보와 같은 원시데이터는 워크플로우 마이닝을 위한 가장 중요한 요소이므로 어떤 목적으로 마이닝을 하느냐에 따라 실행된 정보로부터 저장될 요소가 정의될 수 있다.

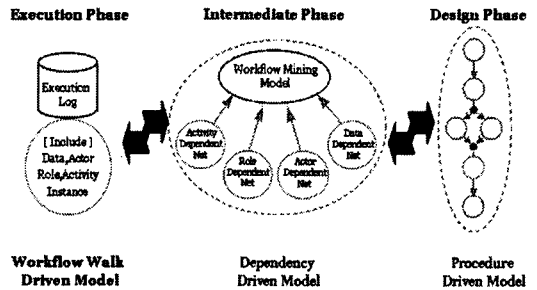
*Dependency Driven Model*은 워크플로우 마이닝을 위한 사전단계로서 *Workflow Walk Driven Model*에서 발생된 실행 정보인 원시데이터 워크플로우 마이닝 모델에 따라서 각 의존성을 분석하여 저장하는 단계이다

*Procedure Driven Model*은 *Dependency Driven Model*에서 저장된 데이터를 이용하여 워크플로우 마이닝 정보를 비즈니스 프로세스의 개선 또는 새로운 비즈니스 프로세스의 정의에 적용되는 단계이다. 즉, 워크플로우 마이닝을 통하여 얻어진 유용한 정보를 이용하여 자원의 재배치, 비즈니스 프로세스의 변경, CRM 구축 등의 작업을 통하여 워크플로우 마이닝 정보를 이용하는 단계이다.

3. 워크케이스 마이닝을 위한 실행 계열분석 알고리즘 설계

액티비티 의존성을 이용한 워크케이스 마이닝이란 빌드타임에 정의된 비즈니스 프로세스 모델의 정보로부터 각 액티비티 간의 의존성의 분석을 통하여 정적으로 정의된 중요 경로와 워크플로우 엔진에서 실행된 워크케이스의 실행 정보를 이용하여 빌드타임에 정의된 모든 경로와 비교함으로써 도달 가능한 모든 경로 중에서 작성된 비즈니스 모델이 어떤 경로로 진행되는지를 살펴봄으로써 정의된 모델의 개

선 여부와 새로운 모델의 작성 시 이용할 수 있는 정보를 작성하는 과정을 의미한다.



[그림 2] 추상화된 워크플로우 마이닝 모델

3.1 중요경로 결정을 위한 액티비티 의존넷 알고리즘

워크케이스 마이닝 예제의 첫 번째 단계로서 빌드타임에서 정의된 비즈니스 프로세스의 실행 가능한 모든 경로를 알아보기 위하여 비즈니스 프로세스에 정의된 액티비티의 의존성을 조사하여 각 액티비티간의 의존성 관계를 액티비티 의존 넷으로 표현한다.

이와 같이 구성된 액티비티 의존 넷은 정의된 비즈니스 프로세스의 제어경로 즉, 조건 분기 수행 또는 병렬수행, 순차실행과 같은 정의된 워크플로우 모델의 실행경로를 찾을 수 있는 기능을 제공한다. 또한 액티비티 의존 넷을 통하여 각 액티비티의 전이 조건을 표현할 수 있고 이와 같은 전이 조건은 각 액티비티의 수행 순서 뿐만 아니라 실행 중 작성된 모델의 동적 변경을 위한 타당성 자료로도 될 수 있다. 또한 액티비티 의존 넷의 최종의 결과로 나타난 그래프를 이용하여 정의된 비즈니스 모델의 중요경로를 찾아 낼 수 있다.

정의된 비즈니스 프로세스가 수행될 수 있는 모든 각각의 경로 또는 워크케이스를 알아보기 위하여 각 워크케이스에서 수행되는 모든 액티비티를 진행과정으로 표현할 수 있다. 즉, 정의

된 비즈니스 프로세스의 시작 노드부터 종료 노드까지 각각의 워크케이스에 참여하는 모든 액티비티를 이용하여 표현하는 방법이다. 그러나 현재와 같이 표현되어야 할 비즈니스 프로세스가 복잡하고, 많은 액티비티를 필요로 하는 상황에서 모든 액티비티를 이용하여 각 워크케이스를 표현하는 것 보다 각 워크케이스를 구분할 수 있는 액티비티를 이용하여 표현하는 것이 효과적이다. 이와 같은 이유에서 본 논문에서는 정의된 비즈니스 프로세스 모델을 이용하여 액티비티 의존 넷을 구성하고 그 결과로부터 중요경로를 구하고자 한다. 액티비티 의존 넷은 확장된 ICN Model에서 δ (제어 경로(Control Flow) 부분)로부터 생성된다[7,8,9].

▶ 정의

액티비티 의존 넷은 일반적으로 액티비티의 집합 A 와 전이조건 집합 T 에 대하여 $\Omega = (\Phi, \zeta, S, E)$ 로 정의된다.

▣ $\Phi = \Phi_i \cup \Phi_o$

$\Phi_o : A \rightarrow \Psi(A)$ 는 각 액티비티에 후행 제어 의존성이 있는 액티비티 집합을 의미하며 $\Phi_i : A \rightarrow \Psi(A)$ 는 하나의 액티비티에 선행 제어 의존성이 있는 액티비티 집합을 의미한다.

▣ $\zeta = \zeta_i \cup \zeta_o$

ζ_i 는 하나의 액티비티에 대하여 선행 제어전이 조건 집합을 의미하며, ζ_o 은 하나의 액티비티에 대하여 선행 제어전이조건 집합을 의미한다.

▣ S 는 초기 제어전이조건 집합으로, 워크플로우 실행 전 외부 프로세스에 의해 정보가 적재된다고 가정한다.

▣ E 는 최후의 제어전이조건 집합으로, 제어 정보는 외부의 프로세스에서 이용될 수 있다

ICN으로 표현된 비즈니스 프로세스 모델에서 액티비티 의존 넷의 정형화된 의미를 위하여 액티비티 노드로 들어오는 실선으로 표현된 화살표는 Φ_i, ζ_i 로 표현되고, 액티비티 노드로부

터 나가는 것과 같이 표현된 실선의 화살표는 Φ_o, ζ_o 로 표현 할 수 있다.

▶ 액티비티 의존 넷 알고리즘

액티비티 의존 넷은 확장 ICN(EICN : Extended Information Control Net) 모델로부터 액티비티 의존 넷 알고리즘을 통하여 구할 수 있다.

액티비티 의존 넷 알고리즘을 정의하기 위하여 "제어 강 의존성" 개념을 정의하여야 한다. 즉, 각 액티비티의 의존성 중에서 어떤 액티비티가 강한 의존성 관계를 갖고 있는지를 구분해야 이후에 중요경로를 구할 수 있는 기능을 제공한다.

Γ 는 확장 ICN이고 u, v 가 액티비티 집합 A 에 포함되었을 경우 즉, $u \in A$ 이고 $v \in A$ 과 같을 때, 다음과 같은 경우에 $v-u$ 경로가 존재하고, vWu 가 v 의 전방지배(Forward Domination)가 존재하지 않는 경우에 액티비티 u 는 액티비티 v 에 제어 강 의존성 관계가 있다고 할 수 있다. 다음 <표 2>는 액티비티 의존 넷 알고리즘을 나타내고 있다

```

INPUT : An Extended ICN
OUTPUT : An Activity-Dependent Net

BEGIN
  FOR all x∈A and y∈A in an Extended ICN
  //OR or LOOP activity
    IF x is strongly control dependent on y
      //get a dependent flow
      ADD x TO  $\Phi_o(y)$ ;
      ADD y TO  $\Phi_i(x)$ ;
      //get control transition conditions
      //between y and x
      ADD  $k_i(x)$  TO  $\zeta_o(y)$ ;
      ADD  $k_i(x)$  TO  $\zeta_i(x)$ ;
    ELSEIF  $fd(y) \in x$  AND ( $fd(y) \notin fd(ibd(y))$ )
      OR  $ibd(y) = \emptyset$ 

```

```

//get a dependent flow
ADD x TO  $\phi_o(y)$  ADD y TO  $\phi_i(x)$ ;
//get control transition conditions
//between y and x
ADD  $k_i(x)$  TO  $\xi_o(y)$ ;
ADD  $k_o(x)$  TO  $\xi_i(x)$ ;
ENDIF
NEXT
END
    
```

<표 2> 액티비티 의존 넷 알고리즘

위의 알고리즘에서 나타나듯이 각 액티비티의 중에서 분기된 조건 중에서 하나를 선택하는 OR 제어전이조건과 제어전이조건 자체로서 액티비티의 집합으로 이루어진 LOOP 제어전이 조건 이후에 나타나는 액티비티는 제어전이조건에 대하여 강의존성 관계가 성립됨을 알 수 있다.

3.2 액티비티 의존 넷 예제 - 주문처리 플로우

<표 2>에서 표현한 알고리즘을 이용하여 액티비티 의존 넷을 나타내기 위하여 다음 [그림 3]과 같은 예제 비즈니스 프로세스가 있을 경우 [그림 3]에 대한 액티비티 의존 넷의 그래픽 표현은 [그림 4]와 같이 나타낼 수 있다.

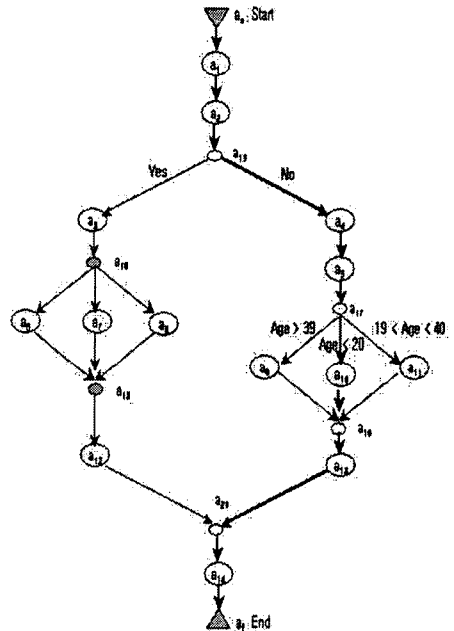
[그림 3]의 예제 비즈니스 프로세스는 20개의 액티비티를 갖는 프로세스로서 a_{15} 에서 OR 분기를 통하여 경로가 결정될 수 있다. 또한 a_{15} 분기에서 No라는 조건이 선택되어 분기를 하면 a_{17} 의 분기 조건에 실행 경로가 결정 될 수 있는 확장 ICN의 예를 나타내고 있다.

[그림 3]에 표현된 주문처리 워크플로우 모델을 액티비티 의존넷으로 나타내면 다음 [그림 4]와 같다.

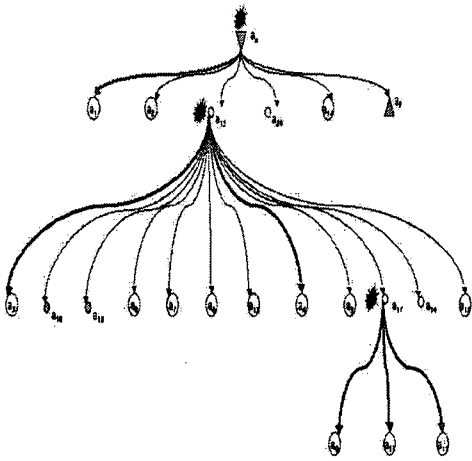
[그림 3]에서 나타나듯이 7가지 - *default*, *OR(yes)*, *OR(no)*, *AND(default)*,

OR(age<20), *OR(19<age<40)*, *OR(age>39)* - 의 제어 전이 조건이 나타난다. *default*란 의미는 각 액티비티 사이에 시간적 순서에 따라 진행되는 일반적인 경우를 의미한다. 그러므로 액티비티 $a_5, a_1, a_2, a_{15}, a_{20}, a_{14}, a_f$ 의 실행순서는 시작노드인 a_5 에 강 종속되어 있다.

OR(yes), *OR(no)*는 이후에 나타나는 액티비티에 대하여 실행경로를 결정할 수 있는 제어 전이 조건이다. 그러므로 $a_3, a_4, a_5, a_6, a_7, a_8, a_{12}, a_{13}, a_{16}, a_{17}, a_{18}, a_{19}$ 는 a_{15} 에서 *OR(yes)*, *OR(no)*의 조건에 강 종속되어 있다. 또한 *OR(age<20)*, *OR(19<age<40)*, *OR(age>39)*는 a_{15} 에서 *OR(no)*가 선택된 경우에 발생할 수 있는 제어 전이 조건이다. 즉, a_{15} 에서 *OR(no)*가 선택되었을 경우 a_{17} 의 *OR(age<20)*, *OR(19<age<40)*, *OR(age>39)* 조건에 따라 a_9, a_{10}, a_{11} 이 결정될 수 있으므로 a_9, a_{10}, a_{11} 은 a_{17} 에 강 종속되었음을 알 수 있다.



[그림 3] 주문처리 워크플로우 모델



[그림 4] 주문처리 프로세스의 액티비티 의존 넷 그래프

3.3 중요경로의 결정

작성된 액티비티 의존 넷을 이용하여 수행될 수 있는 모든 경로를 구할 수 있다. 이때 수행에 참여하는 모든 액티비티를 고려하여 기술하지 않고 가장 중요한 즉, 실행경로를 판단할 수 있는 액티비티만을 이용하여 수행 경로를 나타내는 것을 중요경로라고 한다. 이와 같이 중요경로를 찾아내는 이유는 처리해야 할 비즈니스 프로세스가 복잡해지고 규모가 커짐에 따라 모든 액티비티를 이용하여 런타임에서 발생된 실행경로와 비교하는 것은 많은 시간을 필요로 할 수 있다. 이와 같은 이유로 실행에 참여하는 모든 액티비티를 고려하는 것이 아니라 수행 경로를 표현하는 모든 액티비티 중에서 영향력 있는 액티비티(Very Efficient Activity)를 추출하여 는 것이 중요하다. 일반적으로 표현된 비즈니스 프로세스의 수행 경로에 영향을 주는 OR 노드와 LOOP 노드인 경우이다. 순차적인 경우는 시간적 순서에 따라 진행되고 AND 노드인 경우에는 모든 액티비티를 수행해야 함으로 수행 경로를 결정하는 요소가 될 수 없다. 이러한 이유로 OR 노드

또는 LOOP 노드 뒤에 수행되는 액티비티는 실행 경로를 결정할 수 있는 영향력 있는 액티비티이다.

[그림 4]에서 굵은 실선으로 연결된 액티비티를 영향력 있는 액티비티라 할 수 있고 이러한 영향력 있는 액티비티를 이용하여 중요경로를 찾을 수 있다. 즉, [그림 4]을 영향력 있는 액티비티를 이용하여 간략하게 표현하면 다음과 같이 4가지의 경로로 나타낼 수 있다.

- ▶ Path 1 : {a_s, a₁, a₁₅, a₃, a_f}
- ▶ Path 2 : {a_s, a₁, a₁₅, a₄, a₁₇, a₉, a_f}
- ▶ Path 3 : {a_s, a₁, a₁₅, a₄, a₁₇, a₁₀, a_f}
- ▶ Path 4 : {a_s, a₁, a₁₅, a₄, a₁₇, a₁₁, a_f}

3.4 실행계열 분석 알고리즘의 설계

워크케이스 마이닝을 위한 마지막 단계로서 하나의 워크케이스가 워크플로우 엔진에서 수행되고 그 결과와 빌드타임에서 정의된 수행 가능한 모든 경로와 비교하여 어떠한 경로를 통하여 수행 되었는지를 알아보기 위하여 새로운 알고리즘이 필요하다.

즉, 빌드타임에서 정의된 수행 가능한 경로들 중에서 각각의 워크케이스가 어떤 실행경로를 통하여 수행되었는지는 비즈니스 프로세스에 참여하고 있는 자원들을 효율적으로 이용하기 위한 필요한 정보를 제공한다. 이러한 기능을 수행하는 알고리즘을 실행계열분석 알고리즘이라 하고 다음 <표 3>와 같이 나타낼 수 있다.

```
#DEFINE 중요경로의 수 Number_of_Path
#DEFINE 액티비티의 수 Number_of_Activity

INPUT : Execution Path Log
        (EPL[Number_of_Activity]),
        Essential Path(EP[Number_of_Path]
                       [Number_of_Activity])
```

```

OUTPUT : Boolean

BEGIN
  FOR a_cnt = 1 TO Number_of_Activity
    IF EPL[a_cnt] = Execution Activity THEN
      masking 1
    ELSE
      masking 0
    ENDIF
  NEXT
  //Execution Path Log Assign into EPL[]
END

BEGIN
  FOR cnt = 1 TO Number_of_Path
    FOR a_cnt = 1 TO Number_of_Activity
      IF (EP[cnt][a_cnt] XOR (EPL[a_cnt]
        AND EP[cnt][a_cnt])) == 0 THEN
        Call Return_Process();
      ENDIF
    NEXT // Essential Path Assign into EP[][]
  NEXT // Number of Workcase
END
    
```

<표 3> 실행계열분석 알고리즘

"Execution Path Log"는 각 워크케이스가 워크플로우 엔진에서 실제 실행된 경로를 의미하고 "Essential Path"는 빌드타임에 정의된 워크플로우 모델의 액티비티의 의존성을 이용하여 표현된 수행가능한 모든 경로의 중요경로를 나타낸다.

3.5 워크케이스 마이닝 예제

<표 4>에서 제시된 알고리즘을 이용한 워크케이스 마이닝 실행과정의 예를 살펴보기 위해서는 워크플로우 엔진에서 실제 실행된 워크케이스의 경로가 필요한데 [그림 3]와 같이 정의된 워크플로우 모델을 실행하였을 경우

한 워크케이스에서 {a1, a2, a15, a4, a5, a17, a10, a19, a13, a20, a14}와 같이 실행되었다고 가정하고 상기에서 기술된 4개의 중요경로 - {a1, a15, a3}, {a1, a15, a4, a17, a9}, {a1, a15, a4, a17, a10}, {a1, a15, a4, a17, a11} - 와 비교하는 과정을 설명하겠다.

워크케이스의 실제 수행경로와 빌드타임에 정의된 모델을 이용하여 중요경로로 얻어진 것 중 하나인 {a1, a15, a3}의 마이닝 과정은 다음과 같다.

다음의 [그림 5]는 워크플로우 엔진에서 실제 수행된 실행정보를 나타내고 있고 즉, 20개의 액티비티에서 수행된 액티비티는 1로, 그렇지 않은 액티비티는 0으로 표현한 것이다. [그림 6]은 빌드타임에 정의된 모델을 이용하여 중요경로로 얻어진 것 중 하나인 {a1, a15, a3}을 표현하고 있다.

1	2	3	4	5	6	7	8	9	10
1	1	0	1	1	0	0	0	0	1
11	12	13	14	15	16	17	18	19	20
0	0	1	1	1	0	1	0	1	1

[그림 5] 엔진에서 수행된 워크케이스의 실제 수행경로

1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	0	0	0	0	0
11	12	13	14	15	16	17	18	19	20
0	0	0	0	1	0	0	0	0	0

[그림 6] 액티비티 의존넷을 이용하여 빌드타임에서 얻어진 첫 번째 중요경로

첫 번째 과정으로 두 실행경로에 대하여 AND 연산을 수행한다. 그 결과는 [그림 7]와 같고 다음으로 두 경로의 일치여부를 확인하여 위하여 AND 연산 결과 [그림 7]와 액티비티 의존넷을 이용하여 빌드타임에서 얻어진 첫 번째 중요경로 [그림 6]와 Exclusive OR 연산을 수행하여 그 결과가 0 값을 같으면 두

수행경로는 일치한다는 것을 알 수 있고 현재 워크플로우 엔진에서 수행된 워크케이스는 {a₁, a₁₅, a₃}의 경로로 수행되었음을 알 수 있다. Exclusive OR 연산의 수행결과는 [그림 8]과 같다.

1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0
11	12	13	14	15	16	17	18	19	20
0	0	0	0	1	0	0	0	0	0

[그림 7] AND 연산 수행결과

1	2	3	4	5	6	7	8	9	10
0	0	0	1	0	0	0	0	0	0
11	12	13	14	15	16	17	18	19	20
0	0	0	0	1	0	0	0	0	0

[그림 8] Exclusive OR 연산 수행결과

[그림 8]과 같이 최종 결과를 살펴보면 3번째 bit에서 1이 마스킹 되었으므로 엔진에서 현재 수행된 워크케이스의 실행경로와 빌드타임에서 얻어진 중요경로 중 {a₁, a₁₅, a₃}과는 다르게 수행되었음을 알 수 있다.

이와 같은 작업을 모든 중요경로에 각각 적용하여 수행하여 보면 최종적으로 현재 엔진에서 수행된 워크케이스는 {a₁, a₁₅, a₄, a₁₇, a₁₀}과 같은 경로로 수행되었음을 알 수 있다.

4. 결론

워크플로우 마이닝이 워크플로우 및 비즈니스 프로세스 연구 분야에서 중요한 연구 주제로 인식됨에 따라 본 논문에서는 워크플로우 마이닝의 개념과 마이닝 대상에 따른 워크플로우 마이닝의 구분 하였다. 또한 워크플로우 엔진에서 실행된 워크케이스의 실행경로를 마이닝을 위하여 빌드타임에 정의된 모델로부

터 발생할 수 있는 모든 실행경로와 현재 워크플로우 엔진에서 수행되어진 워크케이스의 수행경로를 비교하기 위하여 두 개의 알고리즘을 제안하였다.

이와 같은 워크케이스 마이닝 결과로 얻어진 실행경로는 워크플로우 시스템이 갖고 있는 한계점인 유연성 및 예외상황 처리, 동적변경 등에 대한 해결책을 제시 할 것이라 생각되며, 본 논문에서 제시하였듯이 워크플로우 마이닝 개념은 현재 수행되고 있는 비즈니스 프로세스의 객관적 분석을 통하여 더욱 개선된 비즈니스 프로세스를 작성할 수 있으며, 조직 및 기업의 자원 활용도를 높임으로써 기업 또는 조직의 내부적인 질적 향상과 고객과의 관계 개선을 위한 정보(CRM)를 이용하여 외부적인 발전을 할 수 있는 초석을 마련할 수 있을 것이다.

향후 연구과제로는 현재 제안된 알고리즘을 이용하여 시스템을 구현하는 것이다. 구현된 워크플로우 마이닝 시스템을 통하여 비즈니스 프로세스의 개선이 어떻게 이루어지는지 또는 새로운 비즈니스 프로세스에 어떻게 적용되는지를 나타내어야 한다.

또한 현재 마이닝의 정보를 실시간으로 모니터링 하는 기능인 워크플로우 마이닝 모니터링의 기능에 대해서도 설계를 하여야 할 것이다. 워크플로우 마이닝 모니터링이란 워크플로우 마이닝 시스템에서 생성된 정보를 사용자 - 워크플로우 마이닝을 이용하는 시스템 또는 사람 - 모두에게 어떻게 제공할 것인지와 같은 단순 단순 모니터링 기능 뿐 만 아니라 관리자의 기능도 수행할 수 있도록 기능 정의 및 아키텍처에 관한 부분도 변경이 필요할 것이다.

참고문헌

[1] Frank Leymann and Dieter Roller, "Production Workflow : Concepts and Techniques", 2000 published by Prentice-Hall, Inc

- [2] Kim, K. "Architectures for Very Large Scale Workflow Management Systems. PhD Thesis. Computer Science Department, University of Colorado. May 1998
- [3] A.J.M.M. Weijters, W.M.P van der Aalst, "Process Mining Discovering Workflow Model from Event-Based Data"
- [4] A.J.M.M. Weijters, W.M.P van der Aalst, "Rediscovering Workflow Model from Event-Based Data"
- [5] 김학성, "워크플로우 마이닝 프레임워크와 아키텍처에 관한 연구", 박사학위논문, 2003
- [6] 김학성, 문기동, 백수기, "워크플로우 마이닝을 위한 실행계열분석 알고리즘", 한국인터넷정보 학회 춘계학술발표논문집, 4권1호, pp419-422, 2003.05
- [7] Clarence A.Ellis and Gary J.Nutt, "Modeling and Enactment of Workflow Systems." In Application and Theory of Petri Nets 1993, Proceedings 14th International Conference, Chicago, Illinois, USA, pages 1-16. Springer-Verlag, 1993.
- [8] Clarence A. Ellis and Gary J.Nutt "Adding Actors and Roles to the ICN Model"
- [9] Kwang-Hoon Kim and Su-Ki Paik, Actor-Oriented Workflow Model, The Second Cooperative Database Systems for Advanced Applications, Wollongong Australia, March 1999.
- [10] R.Agrawal, D.Gunopulos, and F.Leymann. "Mining Process Models from Workflow Logs", in Proc. of the sixth International Conference on Extending Database Technology(EDBT), 1998.

백 수 기



1972년 연세대학교 토목공학과 학사

1979년 동국대학교 경영대학원 정보처리 석사(MBA)

1992년 동국대학교 대학원 전산통계학과 박사

1980~현재 경기대학교 정보과학부 교수

관심분야 : 통신알고리즘, RTE Business Process Monitoring, 정보통신