BPMN 기반의 비즈니스 프로세스 모델에서 그래프 축소 기법을 활용한 구조적 이상 현상 검출*

김건우*, 이승훈*, 이정화**, 손진현*
*한양대학교 컴퓨터공학과
**네이버 서비스 기술개발센터
e-mail: gwkim, shlee{@database.hanyang.ac.kr}, ljk2491@naver.com, jhson@hanyang.ac.kr

Detecting Structural Anomalies in a BPMN-based Business Process Model using Graph Reduction Techniques

Gun-Woo Kim*, Seung Hoon Lee*, Jeong Hwa Lee**, Jin Hyun Son*
*Dept of Computer Science and Engineering, Hanyang University in Ansan
**NAVER: Developer Center in Korea

요 약

급변하는 경영 환경에서 기업의 목표나 정책목적을 실현하기 위해 많은 기업들이 비즈니스 프로세스의 중요성을 인식하게 되면서 비즈니스 프로세스 관리에 대한 관심이 높아졌다. 이러한 비즈니스 프로세스를 관리하기 위해 일반적으로 모델링 작업을 수행하게 되는데 모델링 작업 시 예기치 못한 여러 가지 이상 현상을 포함 될 수 있다. 본 논문에서는 본 연구자가 발표한 선행 논문을 바탕으로 기존의 이상 현상 검출 기법 중 그래프 축소 기법을 확장하여 모델링 단계에서 정의된 비즈니스 프로세스 모델의 구조적 이상 현상을 검출하고 명제 논리학을 이용하여 이의 타당성을 증명하였다.

1. 서론

비즈니스 프로세스는 빠르게 변화하는 기업 환경에서 기업의 목표나 정책 목적을 실현하기 위해 연결된 하나이상의 절차 또는 액티비티를 의미한다.[7] 이러한 비즈니스 프로세스를 관리하기 위해 BPMN(Business Process Modeling Notation)이라는 표준 모델링 표기법으로 모델링 작업을 수행하게 되는데 모델링 작업은 사람에 의해 수행되어 지기 때문에 프로세스의 규모가 큰 경우 예기치 못한 이상 현상들이 발생할 수 있다.[4],[5] 이러한 이상 현상들이 미리 검출되지 않고 프로세스 엔진에 의해 실행된다면 기업에 막대한 비용 손실을 초래하게 된다. 따라서 비즈니스 프로세스 모델에서의 이상 현상 검출 기법에 대한 연구가 활발히 진행 되어왔다. 하지만 기존에 연구된 대부분의 검출기법들은 BPMN 기반의 비즈니스 프로세스가 아닌 워크플로우 기반의 검증기법들로써 OR 게이트웨이 및 Complex 게이트웨이를 제외한 AND와 XOR 게이트웨이에 대해서만 다루고 있으며 프로세스 흐름이 역행하는 루프(Loop) 구조 및 중첩된(Overlapped) 구조와 같은 복잡한 프로세스 모델에서는 이상 현상 검출이 불확실 하였다.[3],[6]

본 논문은 본 연구자가 발표한 선행 논문[1]과 선행 논문[2]의 후속논문으로써 선행 논문[1]에서 정의한 비즈니스 프로세스 모델에서의 구조적 이상 현상과 선행 논문[2]에서 제시한 비즈니스 프로세스 모델 정형화 기법을 바탕으로 기존의 워크플로우 기반의 구조적 이상 현상 검출 기법 중 그래프 축소 기법을 확장하여 루프 구조와 중첩 구조가 포함된 비즈니스 프로세스 모델에서도 구조적 이상 현상을 검출하고 명제 논리학을 사용하여 이의 타당성을 증명할 수 있는 방법을 제시한 논문이다.

본 논문의 구성은 다음과 같다. 2장에서는 BPMN 기반의 비즈니스 프로세스 이상 현상과 기존의 이상 현상 검출기법에 대한 연구를 살펴본다. 3장에서는 그래프 축소 기법과 명제 논리학을 사용한 구조적 이상 현상 검출 방법을 설명하고, 마지막으로 4장에서는 논문의 결론을 맺는다.

2. 관련연구

이 장에서는 비즈니스 프로세스 이상 현상 및 기존의 이상 현상 검출 기법에 대해서 살펴본다.

2.1. 비즈니스 프로세스에서의 이상 현상

비즈니스 프로세스를 모델링 할 때 발생하는 이상 현상은 크게 3가지로 존재하는데 구문론적 이상 현상(Syntactic Anomalies), 구조적 이상 현상(Structural Anomalies), 그리고 의미론적 이상 현상(Semantic Anomalies)들이 있다.[1] 본 논문에서는 비즈니스 프로세스 모델에서 구조적 이상 현상 검출에 초점을 맞추고 있으므로 구조적 이상 현상에 대해서만 소개하겠다.

구조적 이상 현상은 비즈니스 프로세스 모델에서 잘못된 게이트웨이의 사용으로 토큰의 병합이나 분산되는 상호작용을 정상적으로 수행 할 수 없을 때 발생하는 이상 현상이다. 대표적인 구조적 이상 현상은 다음과 같다.[1]

* 본 연구는 지식경제부 및 정보통신 연구 진흥원의 대학 IT연구센터 육성·지원사업 (IITA-2009-c1090-0902-0031)의 연구결과로 수행되었음

* 이 논문은 2007년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음 (KRF-2007-313-D00747)

- 데드락: 프로세스의 특정 단계에서 더 이상의 액티비티가 수행되지 않을 경우를 의미한다.
- 동기화의 부족: 프로세스의 특정 단계에서 의도하지 않게 게이트웨이를 통과한 프로세스 토큰이 해당 액티비티에서 여러 번 수행될 경우를 의미한다.
- 수행되지 않는 액티비티: 프로세스 흐름에서 수행되지 않는 액티비티가 존재 하는 경우를 의미한다.
- 무한루프: 역행하는 흐름을 가진 프로세스의 특정 액티비티가 영원히 반복적으로 수행될 경우를 의미한다.

<표 1> Gateway Truth Table

Split \ Join	AND	OR	XOR	Complex (N out of M)
AND	True	True	False (Lack of Sync)	True
OR	False (Deadlock)	True	False (Lack of Sync)	True for N =1 False for N > 1 (Deadlock) False for N =M (Deadlock)
XOR	False (Deadlock)	True	True	True for N = 1 False for N > 1 (Deadlock) False for N = M(Deadlock)
Complex	False (Deadlock)	True	False (Lack of Sync)	True for N =1 False for N > 1(Deadlock) False for N =M (Deadlock)

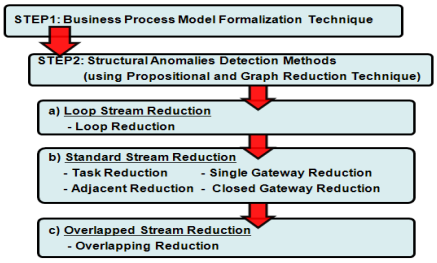
2.2. 비즈니스 프로세스 이상 현상 검출 기법

기존에 연구된 비즈니스 프로세스 이상 현상 검출 기법은 BPMN 기반이 아닌 워크플로우 기반의 검출 기법으로써 크게 다음과 같이 4가지 기법으로 나눌 수 있다.

- Petri-net: Petri-net 기법은 정형화된 이론에 기반을 두어 수학적인 표현 기법을 사용하였기 때문에 정밀한 분석이 가능하다. 하지만 Petri-net 기반의 워크플로우 모델은 표현방식이 매우 복잡하며 이해하기 어렵다는 단점이 있다.[4],[6]
- Graph Reduction 기법: Graph Reduction 기법은 터미널 축소, 순차 축소, 인접 축소, 종료 축소, 중첩 축소 5가지의 축소 규칙을 적용하여 이상 현상을 검출하는 기법이다. 하지만 OR 게이트웨이를 포함하지 않으며 데드락과 동기화의 부족만이 검출 가능하다. 또한 중첩된 구조를 부분적으로 지원하고 루프구조를 지원하지 못하는 단점이 있다.[8]
- Matrix-Based 기법: Matrix-based 기법은 워크플로우 모델을 표현하는데 인접 Matrix를 이용하고 계산 복잡도를 줄이기 위해 인라인 블록이라는 개념을 사용한다. 하지만 데드락과 동기화의 부족만이 검출가능하고 OR 게이트웨이를 포함하지 않으며 계산 복잡도도 검증이 된 바가 없다.[6]
- Logic-Based 기법: Logic-Based 기법은 명제 논리학을 기반으로 하여 OR 게이트웨이를 포함한 워크플로우 모델에서 데드락, 동기화의 부족, 무한루프를 검출할 수 있고 계산 복잡도 역시 $O(n^2)$으로 좋은 편이다. 하지만 루프 구조와 중첩된 구조를 부분적으로밖에 지원하지 못하는 단점이 있다.[6]

3. 비즈니스 프로세스 모델에서의 구조적 이상 현상 검출 방법

본 논문에서 제시된 비즈니스 프로세스 모델에서의 구조적 이상 현상 검출 방법은 그림 1과 같이 먼저 선행논문[2]에 제시된 정형화된 비즈니스 프로세스 모델로 변환한 후, 프로세스의 Flow Stream에 따라 루프 스트림(Loop Stream), 표준 스트림(Standard Stream) 그리고 중첩 스트림 (Overlapped Stream)으로 구분된 그래프 축소 기법을 순서대로 적용한 후 빈 프로세스(Empty Process)가 된다면 해당 프로세스에 구조적 이상 현상이 존재하지 않는 것이다. 하지만 그래프 축소 기법을 순서대로 적용한 후에도 프로세스가 남아 있게 된다면 해당 프로세스는 구조적 이상 현상을 포함하는 프로세스를 의미한다.



(그림 1) 비즈니스 프로세스 구조적 이상 현상 검출 단계

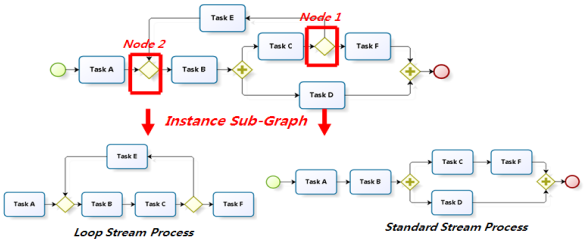
3.1. 비즈니스 프로세스 모델 정형화 기법

BPMN 기반의 비즈니스 프로세스 모델은 구조적인 측면에서 이벤트 및 게이트웨이 생략과 같은 모호성과 중간 이벤트 같은 이상 현상 검출 방법에 적용 불가능 한 요소들을 가지고 있기 때문에 이러한 요소들을 제거하기 위하여 정형화된 프로세스 모델로 변환하는 과정이 우선시되어야 한다.[2]

정형화 기법을 적용하기 전, 본 연구자가 선행논문[2]에서 제시한 5가지의 전제 조건을 미리 만족 하고 모호성 제거와 지원 불가능 요소 변환 룰들을 통해 정형화된 프로세스 모델로 변환되어야 한다.

3.2. 루프 스트림에서의 그래프 축소 기법

루프 스트림이란 비즈니스 프로세스 모델에서 역행하는 흐름이 존재할 경우를 의미한다. 루프 스트림에서는 데드락과 무한루프, 2가지의 이상 현상이 나타나게 된다.[1] 루프 스트림에서는 Loop Reduction 그래프 축소 방법이 사용되게 되며 해당 프로세스의 루프 내에 데드락과 무한 루프를 포함하지 않을 경우 역행하는 흐름인 루프를 삭제하여 표준 스트림으로 변환한다.



(그림 2) Loop Reduction을 적용한 모델

그림 2 에서와 같이 Loop Reduction 그래프 축소 기법을 적용하게 되면 루프를 포함하는 프로세스와 포함하지 않는 프로세스, 2개의 Sub-Graph를 생성하게 된다. 2개의 Sub-Graph 중 루프 스트림을 포함하는 프로세스가 이상 현상을 포함하고 있는지 확인한 후에 역행하는 흐름을 삭제하여 표준 스트림으로 변환 하여야한다. 그림 2 에서와

같이 루프의 시작 노드인 Node 1이 XOR Spilt 게이트웨이이고 루프의 종료 노드인 Node 2가 XOR Join 게이트웨이인 경우에 대해서 명제 논리학을 사용하여 살펴보도록 하자. 명제 논리학 기반의 프로세스 흐름 표현 방식에서 S는 Start 이벤트, E는 End 이벤트, ∧는 And 게이트웨이, ∨는 OR게이트웨이, ⊕는 XOR 게이트웨이, SGW는 Split 게이트웨이, JGW는 Join 게이트웨이를 의미한다.[6]

<표 2> Loop Reduction 검증

1. JGW(Task A ⊕ Task E) -> Task B 2. Task B -> Task C 3. Task C -> SGW(Task E ⊕ Task F) ∴ JGW(Task A ⊕ Task E) -> SGW(Task E ⊕ Task F)					
Task B	Task C	Task E	Task F	JGW (Task A ⊕ Task E)	SGW (Task E ⊕ Task F)
1	1	0	1	1	1
1	1	1	0	1	1

명제 논리학 기반의 토큰 표현 방식에서는 토큰이 태스크에 들어오지 않았을 때를 0으로 표기하고, 토큰이 태스크에 들어왔을 때를 1로 표기한다.[6] 첫 번째 행인 JGW(Task A ⊕ Task E) -> Task B와 두 번째 행인 Task B -> Task C는 Start 이벤트와 연결된 Task A의 상태가 항상 1이기 때문에 Task B와 Task C의 상태도 항상 1이 된다. 하지만 세 번째 행에서는 XOR Split 게이트웨이에 의해서 프로세스 흐름이 분기되기 때문에 Task E와 Task F 중 한 개의 Task만이 상태가 1이 될 수 있다.즉 Task E가 1일 경우와 Task F가 1일 경우, 2가지의 가능성을 가지게 되는데 Task E가 1일 경우에는 프로세스에서 루프가 실행되고 반대로 Task F가 1일 경우에는 루프 구조를 빠져나오게 된다. 이러한 루프 구조를 포함 한 프로세스에서는 LHS인 JGW (Task A ⊕ Task E)와 RHS인 SGW(Task E ⊕ Task F)의 상태가 2가지 가능성에서 모두 1로 동일하여야 한다. 표 2에서 보는 바와 같이 XOR Split과 Xor Join으로 이루어진 루프 구조에서는 2가지 가능성에서 모두 LHS와 RHS의 상태가 1로 동일함으로 이는 이상 현상을 포함하지 않는 프로세스로 간주하여 삭제하게 된다. 반대로 LHS와 RHS의 상태가 다를 경우 이상 현상을 포함하는 프로세스이기 때문에 삭제할 수 없다.

3.3. 표준 스트림에서의 그래프 축소 기법

표준 스트림이란 비즈니스 프로세스 모델에서 정방향의 프로세스 흐름만이 존재할 경우를 의미한다. 표준 스트림에서는 데드락, 동기화의 부족, 수행되지 않는 액티비티, 3가지의 이상 현상이 존재한다. 표준 스트림인 경우 Task Reduction, Single Gateway Reduction, Adjacent Gateway Reduction, Closed Gateway Reduction, 총 4가지의 그래프 축소 기법이 반복적으로 적용된다.

3.3.1. Task Reduction 그래프 축소 기법

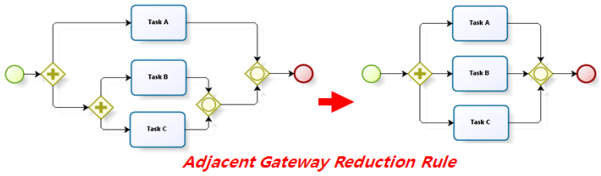
Task Reduction 그래프 축소 기법은 프로세스 모델에서 모든 액티비티 태스크를 삭제시킨다. 정형화된 비즈니스 프로세스 모델에서 액티비티 태스크는 한 개의 입력과 한 개의 출력만을 가지기 때문에 컨트롤 플로우 상의 오류를 포함하지 않게 된다. 그러므로 모든 액티비티 태스크를 삭제하여도 프로세스 모델의 구조적 이상 현상 검출 방법에는 영향을 미치지 않는다.[8]

3.3.2. Single Gateway Reduction 그래프 축소 기법

Single Gateway Reduction 그래프 축소기법은 게이트웨이 중 토큰의 분기를 결정하는 Split 게이트웨이에만 적용된다. BPMN기반의 프로세스 모델에서는 Split 게이트웨이와 Join 게이트웨이가 한 쌍으로 쓰이지 않고 Split 게이트웨이에 의해 2개 이상의 종료 이벤트로 흐름이 진행되는 Implicit Termination 패턴을 지원하기 때문에 이를 삭제하는 기능을 필요로 한다.[2],[5]

3.3.3. Adjacent Gateway Reduction 그래프 축소 기법

Adjacent Gateway Reduction 그래프 축소기법은 Split 게이트웨이 또는 Join 게이트웨이가 선행 또는 후행되는 게이트웨이와 동일한 타입일 경우 게이트웨이의 Incoming Path 또는 Outgoing Path를 선행 또는 후행되는 게이트웨이와 연결한 후 게이트웨이를 삭제하는 역할을 한다.



(그림 3) Adjacent Gateway Reduction을 적용한 모델

그림 3의 과정을 명제 논리학을 이용하여 검증하여 보면 표 2와 같다.

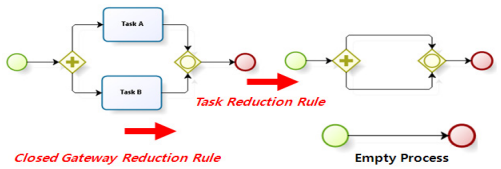
<표 3> Adjacent Gateway Reduction 검증

Adjacent Gateway Reduction 적용 전
1. S-> SGW(Task A ∧ SGW(Task B ∧ Task C)) 2. SGW(Task A ∧ SGW(Task B ∧ Task C)) -> JGW(JGW(Task B ∨ Task C) ∨ Task A) 3. JGW(JGW(Task B ∨ Task C) ∨ Task A) -> E ∴ S-> E
Adjacent Gateway Reduction 적용 후
1. S-> SGW(Task A ∧ Task B ∧ Task C) 2. SGW(Task A ∧ Task B ∧ Task C)) -> JGW(Task B ∨ Task C) ∨ Task A) 3. JGW(Task B ∨ Task C ∨ Task A) -> E ∴ S-> E

표 3 에서 알 수 있듯이 선행 또는 후행되는 게이트웨이가 동일한 타입일 경우 Task A, Task B, Task C중 어떠한 액티비티 태스크가 먼저 수행되더라도 프로세스 흐름에는 영향을 미치지 않는다. 즉 프로세스 흐름이 Adjacent Gateway Reduction 그래프 축소 기법으로 인하여 변경되지 않는 것을 의미한다.

3.3.4. Closed Gateway Reduction 그래프 축소 기법

Closed Gateway Reduction 그래프 축소기법은 한 쌍의 Split 게이트웨이와 Join 게이트웨이가 동일한 타입이거나 표 1에서 정의된 오류가 없는 타입일 경우 해당하는 게이트웨이 쌍을 삭제하는 역할을 한다.



(그림 4) Closed Gateway Reduction을 적용한 모델

그림 4의 과정을 명제 논리학을 이용하여 검증하여 보면 표 3과 같다.

<표 3> Closed Gateway Reduction 검증

1. S-> SGW(Task A ∧ Task B)
2. SGW(Task A ∧ Task B) -> JGW(Task A ∨ Task B)
3. JGW(Task A ∨ Task B) -> E

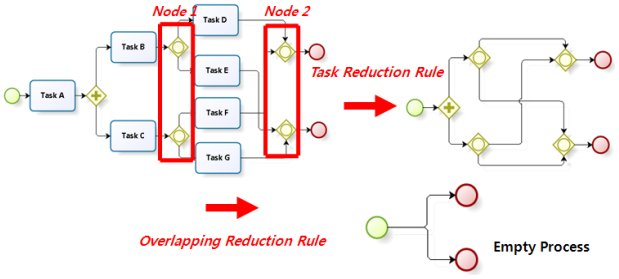
∴ S-> E

Task A	Task B	SGW (Task A ∧ Task B)	JGW (Task A ∨ Task B)
0	*0*	*0*	*0*
0	*1*	*0*	*1*
1	*0*	*0*	*1*
1	**1**	**1**	**1**

그림 4 에서 And Split 게이트웨이를 통과한 토큰들은 항상 Task A와 Task B로 향하여야한다. 즉 Task A와 Task B의 토큰은 항상 1이여만 하기 때문에 LHS인 Split 게이트웨이로 이루어진 SGW(Task A ∧ Task B)와 RHS인 Join 게이트웨이로 이루어진 JGW(Task A ∨ Task B)의 상태는 항상 1이 된다. 그 결과 표 3의 결과와 같이 LHS와 RHS의 상태가 1로 동일함으로 이는 이상 현상을 포함하지 않는 프로세스로 간주하여 삭제하게 된다.

3.4. 중첩 스트림에서의 그래프 축소 기법

중첩 스트림이란 비즈니스 프로세스 모델에서 교차방향의 프로세스 흐름이 존재할 경우를 의미한다.[4] 본 논문에서는 중첩 스트림의 패턴을 Split 게이트웨이로부터 나온 각각의 경로가 서로 다른 Join 게이트웨이와 연결된 교차방향의 프로세스 흐름이라고 정의하였다. 중첩 스트림에서는 데드락과 동기화의 부족, 수행되지 않는 액티비티 등의 이상 현상이 존재하며 Overlapping Reduction 그래프 축소 기법을 사용하여 이상 현상을 검출하게 된다.



(그림 5) Overlapping Reduction을 적용한 모델

<표 4> Overlapping Reduction 검증

1. S-> Task A
1. Task A -> SGW(Task B ∧ Task C)
2.a Task B -> SGW(Task D ∨ Task E)
- **Case 1: SGW(Task D ∧ Task E)**
- **Case 2: SGW(Task D ⊕ Task E)**
2.b Task C -> SGW(Task F ∨ Task G)
3. JGW(Task D ∨ Task F) -> E
4. JGW(Task E ∨ Task G) -> E

∴ S-> E

그림 5의 과정을 명제 논리학을 이용하여 검증하면, 시작 이벤트 와 연결된 Task A에서 AND Split Gateway로 분기된 Task B와 Task C의 토큰 상태는 항상 1임을 알 수 있다. 하지만 Task B와 Task C에서 OR Split Gateway로 분기된 Task D와 Task E, Task F와 Task G의 상태는 항상 1일수도 있고 둘 중에 1개만 1일수도 있다.

중첩 스트림에서는 이러한 2가지의 가능성을 치환 방법으로 해결할 수 있다. 즉 Task B에서 분기된 Task D와 Task E의 토큰 상태가 항상 1일 경우에는 AND 게이트웨이로 분기된 SGW(Task D ∧ Task E)로 치환 이 가능하며, 둘 중에 1개만 1일 경우에는 XOR게이트웨이로 분기된 SGW(Task D ⊕ Task E)로 치환이 가능하다. Task F와 Task G의 경우에도 동일하다. 중첩 스트림에서는 Split 게이트웨이와 Join 게이트웨이가 쌍을 이루지는 않지만 본 논문에서 제시된 표 1에 의하여 OR Join 게이트웨이로 이루어진 JGW(Task D ∨ Task F)와 JGW(Task E ∨ Task G)가 각각의 Task가 AND Split 게이트웨이로 치환된 경우와 XOR Split 게이트웨이로 치환된 경우를 모두 만족하기 때문에 이는 이상 현상을 포함하지 않는 프로세스로 간주하여 삭제하게 된다.

4. 결론

본 논문에서는 기존에 연구된 위크플로우 기반의 이상 현상 검증 방법 중 그래프 감소 기법을 확장 하여 BPMN 기반의 비즈니스 프로세스 모델에도 적용 할 수 있게 개선하였으며 루프(Loop) 구조 및 중첩된(Overlapped) 구조와 같은 복잡한 프로세스 모델에서도 이상 현상을 검출 할 수 있도록 하였다. 이를 바탕으로 향후 과제에서는 그래프 축소 기법에 대한 알고리즘 적용한 프로세스 모델링 툴을 개발할 예정이다.

참고문헌

[1] 김건우, 이정화, 손진현, “비즈니스 프로세스 모델에서의 이상 현상” 정보과학회 논문지: 제 14권 9호, December, 2008.

[2] 이정화, 김건우, 손진현, “비즈니스 프로세스 정형화 기법” 2008 정보처리학회 추계 학술대회, 14, November, 2008

[3] 김학수, 박찬희, 설주영, 손진현, “컨트롤 흐름 경로 기반의 비즈니스 프로세스 타당성 검증 기법”, 정보처리학회 논문지D, August 3, 2007

[4] A.H.M ter Hosfstede et al. Verification problems in conceptual workflow specificaions[J]. Data & Knowledge Engineering, 1998

[5] "BPMN Specification Release: BPMN v.1.1", January 17, 2008

[6] HENRY H. BI, LEON ZHAO, “Applying Propositional Logic to Workflow Verification”, Information Technology & Management, 2004

[7] Nainani B., "Closed Loop BPM using Standard based tools, Oracle Corporation 2004

[8] W. Sadiq, and M.E. Orlowska, "Analyzing Process Models Using Graph Reduction Techniques", Information Systems, 2000