

## 비즈니스 프로세스 정형화 기법\*

이정화\*, 김진우\*, 김광복\* 손진현\*

\*한양대학교 컴퓨터공학과

jwlee, gwkim, gbkim@database.hanyang.ac.kr, jhson@hanyang.ac.kr

### Business Process Formalization Technique

Jeongwha Lee\*, Gunwoo Kim\*, Gwangbok Kim\*, Jin Hyun Son\*

\*Dept of Computer Science and Engineering, Hanyang University in Ansan

#### 요 약

비즈니스 프로세스란 기업의 목표 달성을 위하여 다양한 비즈니스 규칙에 의해 정의된 상호 연관성이 있는 비즈니스 기능들의 집합을 의미한다. 비즈니스 프로세스는 크게 모델링, 구현, 실행, 관리 이렇게 총 4가지의 단계로 구성이 되는데 모델링 단계에서 비즈니스 프로세스 다이어그램을 모델링 할 경우 예기치 못한 여러 가지 이상 현상을 포함 할 수 있다. 본 논문에서는 비즈니스 프로세스 정형화 기법을 이용하여 비즈니스 프로세스에서 모호성을 제거하고 이상 현상 검출에 불필요한 요소를 변환하여 기존의 워크플로우에 적용되었던 이상 현상 검출기법을 BPMN을 이용하여 모델링 되어진 비즈니스 프로세스에서도 사용할 수 있도록 하였다.

#### 1. 서론

BPMN(Business Process Modelling Notation)이란 업무 프로세스 플로우와 웹 서비스를 모델화한 표준 프로세스 모델링 표기법이다 [1]. 비즈니스 프로세스를 모델링 할 경우 일반적으로 이러한 표기법을 사용하여 모델링 하게 되는데 모델링 작업은 사람에게 의해 수행되어지기 때문에 프로세스의 규모가 큰 경우, 모델링 단계에서 예기치 못한 이상 현상이 발생할 수 있게 된다. 이러한 이상 현상이 미리 검출되지 않고 프로세스 엔진에 의해 실행된다면 막대한 손실 및 비용을 초래할 수 있기 때문에 모델링 단계에서 이상 현상을 검출하는 작업이 필요하다. 따라서 비즈니스 프로세스 이상 현상 검증 기법에 대한 연구가 활발하게 진행되어져 왔는데 이를 크게 4가지로 분류해 보면 Petri-net기법, Graph Reduction 기법, Matrix-based 기법 Logic-based 기법으로 나눌 수 있다 [2], [3], [4]. 하지만 위의 검증기법들 대부분이 OR 게이트웨이와 Complex 게이트웨이를 제외한 AND와 XOR 게이트웨이에 대해서만 다루고 있고 루프구조 및 중첩 구조와 같은 복잡한 프로세스에 대한 검증이 불확실하며 워크플로우 방식에 기반을 두고 있어서 최신의 BPMN(Business Process Modeling Notation) 스펙을 따르지 않는 등 몇 가지 문제점이 존재하였다. 따라서 본 논문에서는 기존의

검증 기법들에 대해서 살펴보고 비즈니스 프로세스 정형화 기법을 적용하여 비즈니스 프로세스에서 이상 현상 검출에 불필요한 요소들을 지원 가능한 요소들로 변환하여 위에서 언급된 검증기법들을 BPMN을 사용하여 모델링 된 비즈니스 프로세스 에서도 적용할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 비즈니스 프로세스를 모델링하기 위한 표기법인 BPMN 및 기존의 이상 현상 검증기법에 대한 연구를 살펴본다. 3장에서는 비즈니스 프로세스를 정형화 하는 기법을 설명하며 4장에서는 프로세스 정형화 기법의 타당성에 대해서 설명한다. 마지막으로 5장에서는 논문의 결론을 맺는다.

#### 2. 관련연구

이 장에서는 기존의 4가지 대표적 이상 현상 검증기법에 대한 연구를 살펴보기로 한다.

##### 2.1 Petri-net 기법

Petri-net 기법은 비즈니스 프로세스 모델을 표현하고 검증하기 위해 오래전부터 사용되어지던 방식이다. Petri-net 기법의 장점으로는 정형화된 이론에 기반을 두고 있고 토큰 기반의 워크플로우 상태를 표현할 수 있으며, 정밀한 분석 및 검증 틀 이라고 할 수 있다. 하지만 Petri-net 기반 워크플로우 표현 방식이 이해하기 어렵고 표기에 대한 한계 때문에 대부분의 WfMS(Workflow Management System)에서는 Petri-net 기법을 사용하지 않는다.

##### 2.2 Graph Reduction 기법

Graph Reduction 기법은 테드락과 동기화의 부족, 이

\* 이 논문은 2007년도 정부(과학기술부)의 재원으로 한국 과학재단의 지원을 받아 수행된 연구임(No.R01-2007-000-20135-0)

\* 본 연구는 지식경제부 및 정보통신 연구진흥원의 대학 IT연구센터 육성·지원사업 (IITA-2008-C1090-0801-0031)의 연구결과로 수행되었음

2개의 구조적 에러를 검출하기 위해 개발되었다. 이 기법은 터미널 축소, 순차 축소, 인접 축소, 종료 축소, 중첩 축소 이렇게 5가지의 축소 규칙을 반복적으로 적용하여 에러를 검출하는 기법이다. 복잡도도  $O(n^2)$ 으로 상당히 좋은 편이나, 중첩된 구조를 부분적으로 지원하고 루프구조를 지원하지 못하는 단점이 있다.

**2.3 Matrix-based 기법**

Matrix-based 기법은 워크플로우 모델을 표현하는데 인접 Matrix를 이용하고 계산 복잡도를 줄이기 위해 인라인 블록이라는 개념을 사용한다. 이 기법은 루프를 가진 워크플로우 모델에서 테드락과 동기화의 부족을 검출하는데 효과적이고 복잡한 구조를 가지는 프로세스도 검증이 가능하지만 OR 게이트웨이를 다룰 수 없는 문제점이 있고 계산 복잡도도 검증이 된 바가 없다.

**2.4 Logic-based 기법**

Logic-base 기법은 명제 논리학(propositional logic)을 이용하여 OR 게이트웨이를 포함한 워크플로우 모델에서 앞에서 언급한 5가지 구조적 문제들을 검출할 수 있고 계산 복잡도도  $O(n^2)$ 으로 좋은 편이다. 하지만 중첩된 구조를 부분적으로밖에 지원하지 못하는 단점이 있다.

**3. 비즈니스 프로세스 정형화 기법**

사용자에 의해 설계된 비즈니스 프로세스는 구조적인 측면에서 게이트웨이의 생략과 같은 모호성을 가지고 있기 때문에 이러한 모호성을 제거하기 위하여 정형화된 킷트를 플로우로 변환하는 과정이 우선되어야 한다 [5], [6]. 또한 기존의 검증 기법이 이벤트 및 특정 태스크는 고려하고 있지 않기 때문에 태스크와 게이트웨이 혹은 커넥팅 오브젝트 이외의 엘리먼트 들을 제거하거나 지원 가능한 엘리먼트로 치환하는 작업이 선행되어야 한다.

정형화 기법을 적용하기 전에 아래와 같은 5가지의 전제 조건이 미리 만족되어 있어야 한다.

가정 1 : 정형화 기법을 적용하기에 앞서 모델링된 비즈니스 프로세스는 구문론적 에러를 포함하지 않아야 한다. 구문론적 에러란 BPMN 스펙에 정의된 표기법을 따르지 않을 경우에 발생하는 이상 현상을 의미한다.

가정 2 : 모델링 된 비즈니스 프로세스는 협업 프로세스가 아닌 단일 프로세스로 구성되어야 한다. 이것은 만약 비즈니스 프로세스가 협업 프로세스로 구성이 되었을 경우 프로세스 간에 메시지 플로우가 존재하게 되는데 기존의 검증기법들은 메시지 플로우에 대해서 고려하고 있지 않기 때문에 여기에서는 단일 프로세스에 대해서 정형화 기법을 적용하기로 한다.

가정 3 : 태스크 타입은 고려하지 않는다. 태스크 타입이란 None, Service, Receive, Send, User, Script, Manual, Reference로 설정될 수 있는 태스크의 특성을 의미하는데 이러한 태스크 타입들은 메시지 플로우에 의해 영향을 받기 때문에 고려하지 않도록 한다.

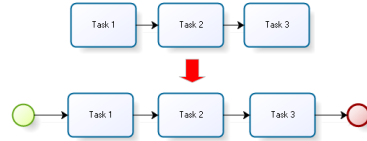
가정 4 : 서브 프로세스는 항상 내부에 어떠한 프로세스가 존재하는지 알려져야 한다. 서브 프로세스는 내부에 또 다른 프로세스를 가지는 복합 요소인데 내부에 어떠한 프로세스가 존재하는지 알아야만 독립적으로 정형화 기법을 적용할 수 있기 때문에 서브 프로세스는 항상 내부에 어떠한 프로세스가 존재하는지 알려져야 한다.

가정 5 : 다중 인스턴스 타입은 항상 Serial 인지 Parallel인지 결정되어야 한다. 다중 인스턴스의 경우 순서에 따라 2가지의 다른 타입의 구조가 존재하기 때문에 항상 Serial 인지 Parallel인지 결정되어 져야 한다.

비즈니스 프로세스 정형화 기법은 다음과 같은 3가지 단계로 구성된다. 3.1절에서는 모호성 제거에 관련된 정형화 기법을 기술하고 3.2절에서는 기존의 검증 기법에서 지원 불가능한 요소들에 대한 정형화 기법을 기술한다.

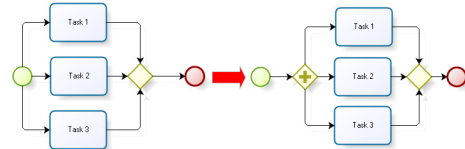
**3.1 모호성 제거**

BPMN 표준안에 의거하여 다음과 같이 시작 및 종료 이벤트 혹은 게이트웨이가 생략될 수 있는데 기존의 검증 기법은 시작 및 종료 이벤트, 태스크, 게이트웨이를 기반으로 하기 때문에 모든 시작 및 종료 이벤트가 표시되어야 한다.

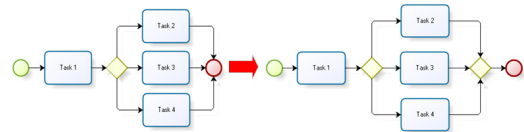


(그림 1) 시작 및 종료 이벤트 표시

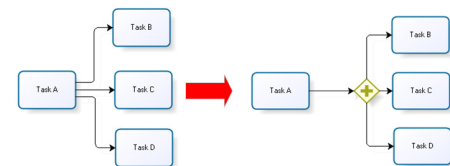
게이트웨이의 생략에 대해서도 아래 그림과 같이 모두 명시해 주어야만 한다.



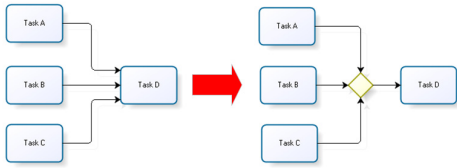
(그림 2) 게이트웨이 표시 1



(그림 3) 게이트웨이 표시 2



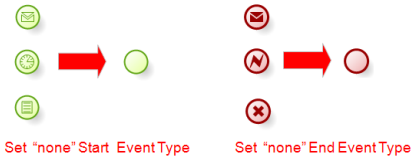
(그림 4) 게이트웨이 표시 3



(그림 5) 게이트웨이 표시 4

3.2 지원 불가능 요소 변환

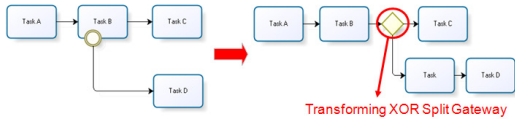
이벤트의 경우 그림 6과 같이 시작과 종료 이벤트에 올 수 있는 이벤트 타입을 None 이벤트 타입으로 변환하여야 하며, 중간이벤트(intermediate event) 및 링크 이벤트의 경우, 각각의 경우에 따라 다음그림 7, 8, 9와 같이 변환하여야 한다.



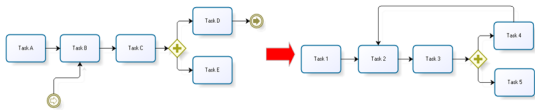
(그림 6) 이벤트 타입 변환



(그림 7) 독립적으로 사용된 중간 이벤트 변환

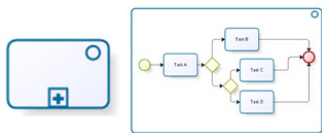


(그림 8) 종속적으로 사용된 중간 이벤트 변환

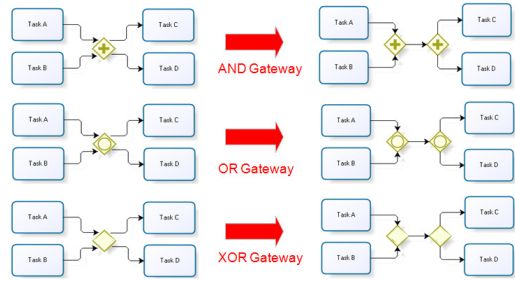


(그림 9) 링크 이벤트 변환

또한 그림 10과 같이 서브 프로세스가 비즈니스 프로세스 내에 존재할 경우 서브 프로세스 자체를 태스크로 치환하고 해당 서브 프로세스를 하나의 완전한 비즈니스 프로세스로 간주하여 정형화 기법을 적용하여야 한다.

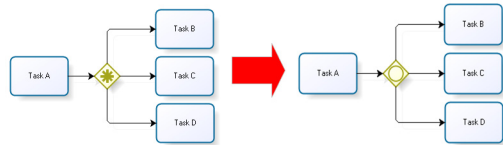


(그림 10) 서브 프로세스 변환



(그림 11) 게이트웨이 특성에 따른 변환

위 그림 11의 경우는 미리 정의된 게이트웨이의 특성에 위배하는 표기들을 특성에 맞게 변환시켜 주는 방법이고 그림 12의 경우는 지원 불가능한 요소(Complex Gateway)를 지원 가능한 요소로 변환하는 과정을 의미한다.



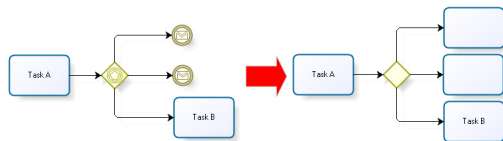
(그림 12) Complex 게이트웨이 변환

그림 12의 경우는 표1의 게이트웨이쌍 진리표를 통해서 게이트웨이 변환에 의미상의 차이가 없는 것을 확인할 수 있다.

<표 1> 게이트웨이쌍 진리표

Split \ Join	AND	OR	XOR	Complex
AND	True	True	False (Lack of Sync)	True
OR	False (Deadlock)	True	False (Lack of Sync)	False (Deadlock)
XOR	False (Deadlock)	True	True	False (Deadlock)
Complex	False (Deadlock)	True	False (Lack of Sync)	False (Deadlock)

그림 14의 경우도 지원 불가능한 게이트웨이(Event-Based XOR Gateway)를 지원 가능한 게이트웨이(Data-Based XOR Gateway)로 변환하는 과정을 의미한다.



(그림 14) Event-Based XOR 게이트웨이 변환

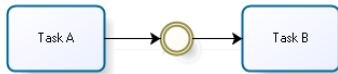
#### 4. 정형화 기법 검증

정형화 기법을 이용하여 비즈니스 프로세스를 변환하는데 있어서 주의해야 할 점은 기존의 검증기법이 컨트롤 플로우에 초점을 맞추고 있기 때문에 컨트롤 플로우 상의 의미적인 차이가 존재하지 않아야 한다는 것이다. 검증을 위해 기존에 소개된 로직 접근 방법의 표기법을 이용하였다 [7], [8]. 먼저 컨트롤 플로우 상의 의미가 일치하기 위해서는 아래와 같은 2가지 조건을 만족시켜야 한다.

Definition 1 : 요소들이 동일한 indegree와 outdegree를 가질 경우 컨트롤 플로우 상의 특성이 동일하다고 가정한다.

Definition 2 : 요소들이 동일한 preceding 요소와 succeeding 요소를 가질 경우 컨트롤 플로우 상의 특성이 동일하다고 가정한다.

위 2가지 정의를 기반으로 아래 그림 15의 예시를 통해 정형화 기법을 검증하기로 한다.



(그림 15) 독립적으로 사용된 중간 이벤트

위 그림을 수식으로 표현하면 Task1 → E<sub>i,n</sub> → Task2와 같이 표현할 수 있다. 수식은 아래와 같이 설명될 수 있다.

- E(이벤트)는 3가지의 하위Set으로 구분되어질 수 있다. 1) 시작이벤트(Es), 2) 중간이벤트(Ei), 3) 종료이벤트(Ee)

-  $\forall x \in E_i, in(x) = 1 \text{ and } out(x) = 1$  : 중간이벤트(Ei)에 속하는 x는 indegree가 1이고 outdegree가 1인 컨트롤 플로우 특성을 지닌다.

-  $\forall x \in T, in(x) = 1 \text{ and } out(x) = 1$  : 태스크(T)에 속하는 x는 indegree가 1이고 outdegree가 1인 컨트롤 플로우 특성을 지닌다.

따라서 중간이벤트와 태스크는 동일한 indegree와 outdegree를 가지므로 컨트롤 플로우 상의 의미가 일치한다고 볼 수 있다.

#### 5. 결론

본 논문에서는 기존의 워크플로우 형태에서 사용되어진 대표적인 4가지 이상 현상 검증기법을 BPMN을 이용하여 모델링한 비즈니스 프로세스에도 적용할 수 있게 하기 위해 컨트롤 플로우 상의 의미상의 차이가 없도록 모호성을 제거하고 불필요한 요소들을 제거하거나 다른 형태로 치환하여서 이상 현상 검증의 발판을 마련하였다. 이뿐만 아니라 BPMN의 중요 목적인 비즈니스 분석가부터 모든 비즈니스 사용자까지 즉시 이해할 수 있는 표기법을 규정하는 것 외에도 비즈니스 프로세스의 실행을 위한 XML언어의 발생을 가능하게 하는 내부 모델을 규정하는

것이 또 다른 BPMN의 목적이라 할 수 있는데 이러한 매핑 과정에서 Complex Gateway는 지원하지 못한다고 규정 되어있다.

이를 바탕으로 향후 과제에서 개발할 모델링 틀에서 이상 현상 검증기능을 지원하고 난 뒤 BPEL(Business Process Execution Language)로의 매핑기능을 지원하고자 할 경우 먼저 모델링된 비즈니스 프로세스에 정형화 기법을 적용하고 난 뒤 검증 알고리즘을 적용 한 후 실행 언어로의 매핑을 지원할 계획이다.

#### 참고문헌

- [1] Object Management Group, "Business Process Modeling Notation (BPMN) - Version 1.1" January 2008, BPMI.org, 2008
- [2] W.M.P. van der Aalst. Verification of Workflow Nets. In P. Azéma and G. Balbo, editors, Application and Theory of Petri Nets 1997, volume 1248 of Lecture Notes in Computer Science, pages 407 - 426. Springer-Verlag, Berlin, 1997
- [3] W. Sadiq, and M.E. Orłowska, "Analyzing Process Models Using Graph Reduction Techniques", Information Systems, 2000
- [4] A.H.M ter Hofstede et al. Verification problems in conceptual workflow specifications[J]. Data & Knowledge Engineering, 1998
- [5] Wohed, P., van der Aalst, W.M., Dumas, M., ter Hofstede, A., Russell, N., "On the Suitability of BPMN for Business Process Modeling" In Dustdar, S., Fiaderino, J.L., Sheth, A.P. (eds.) Proceeding 4th International Conference on Business Process Management 2006
- [6] P.Wohed, W.M.P van der Aalst, M. Dumas, and A.H.M. ter Hofstede, and N. Russell. "Pattern-based Analysis of BPMN - an extensive evaluation of the Control-Flow, the Data and the Resource Perspectives. BPM Center Report BPM-05-26, BPMcenter.org, 2005
- [7] Qianhui Althea Liang, J. Leon Zhao, "Verification of Unstructured Workflows via Propositional Logic", Seventh IEEE/ACIS International Conference on Computer and Information Science, 2008
- [8] HENRY H. BI, LEON ZHAO, "Applying Propositional Logic to Workflow Verification", Information Technology & Management, 2004