

# 비즈니스 프로세스 모델에서의 설계 오류 유형 분석\*

김진우\*, 이정화\*, 김재형\*, 손진현\*

\*한양대학교 컴퓨터공학과

e-mail: gwkim, jwlee, jhkim {@database.hanyang.ac.kr}, jhson@hanyang.ac.kr

## Analysis of Design Error Types in Business Process Modeling

Gun-Woo Kim\*, Jeong Hwa Lee\*, Jae Hyung Kim\*, Jin Hyun Son\*

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

### 요 약

비즈니스 프로세스란 기업의 모든 고객에게 적절한 서비스를 제공하기 위해 순차적 또는 동시적으로 발생할 수 있는 모든 연관된 업무의 집합을 의미한다. 이러한 비즈니스 프로세스는 프로세스 설계 작업을 통하여 시스템 차원에서 관리함으로써 프로세스의 생산성 및 효율성 등을 확대하게 된다. 하지만 프로세스 설계 작업은 주로 사람에 의해 수행되기 때문에 예상하지 못한 오류들이 발생할 수 있게 된다. 이러한 설계 오류들은 기업이 원하는 과정으로 프로세스가 진행하지 않을 수 있으며, 이는 기업의 막대한 인적 손실 및 비용 손실을 초래하게 된다. 그러므로 비즈니스 프로세스 설계 단계에서 오류가 없도록 설계 하거나 또는 미리 정의된 오류 유형들을 설계 도구를 통하여 검출할 수 있는 작업이 필요하다. 본 논문에서는 비즈니스 프로세스 설계 단계에서의 오류 검출 작업에 활용될 수 있도록 유형 별로 설계 오류들을 분석 하고 정의하였다.

### 1. 서론

비즈니스 프로세스는 기업의 모든 고객에게 적절한 서비스를 제공하기 위해 순차적 또는 동시적으로 발생할 수 있는 기업의 업무들을 적절하게 수행하기 위하여 연결된 하나 이상의 절차 또는 액티비티의 집합을 말한다.

비즈니스 프로세스는 일반적으로 프로세스 설계 - 구현 - 실행 - 관리 이렇게 총 4가지의 단계로 구성된다. 총 4 단계 중 가장 먼저 수행이 되는 프로세스 설계 작업이 이 중 가장 중요한 부분을 차지하게 되는데 BPMN (Business Process Modeling Notation)을 이용하여 시스템 내/외부 객체들의 상호 관계들을 정의하여 비즈니스 프로세스 다이어그램을 설계 하여야 한다. 프로세스 설계 단계에서는 항상 일관성과 완벽성을 제공하여 비즈니스 분석가들과 개발자들이 설계단계에서 정의된 프로세스 모델의 비즈니스 요구 사항들을 이해할 수 있도록 해야 한다. 또한 설계 작업을 하는 동안에 표준 절차에 대한 대안과 예외들도 파악되어야 하는데, 설계 작업은 사람에 의해 수행되기 때문에 예기치 못한 이상 현상이 발생할 수 있게 된다. 이러한 설계 오류들이 미리 검출되지 않고 실행 단계에서 프로세스 엔진에 의해 실행된다면 기업의 막대

한 손실 및 비용을 초래할 수 있기 때문에 설계 단계에서 미리 오류가 없도록 모델링을 하거나 혹은 설계 도구 (Process Modeling Tool) 자체에서 설계 오류들을 검출하는 작업이 필요하다.[1],[3],[4] 따라서 본 논문에서는 비즈니스 프로세스 이상 현상 예방 및 검증을 위해 어떠한 이상 현상이 존재하는지 미리 정의하고 올바른 비즈니스 프로세스 모델링을 위해 몇 가지 제약사항 들을 나열하였다.

본 논문의 구성은 다음과 같다. 2장에서는 비즈니스 프로세스를 모델링하기 위한 표기법인 BPMN 및 구성요소들에 대해서 살펴본다. 3장에서는 각각의 타입에 따른 비즈니스 프로세스 이상 현상들을 설명하며 마지막으로 4장에서는 논문의 결론을 맺는다.

### 2. 배경지식

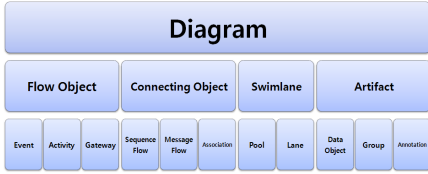
이 장에서는 비즈니스 프로세스 모델링 표기법인 BPMN 및 구성 요소들에 대해서 살펴본다.

#### 2.1. BPMN 개요

BPMN은 비즈니스 프로세스를 가시화 하기위한 모델링 표기법으로써 미국 비영리단체 BPMI에서 비즈니스 프로세스 모델에 관해 업계 표준을 만들고자 생겨났다. BPMN은 모든 사용자가 사용하고 이해하기 쉬운 Notation 개발을 목표로, BPEL4WS와 BPML과 같이 비즈니스 프로세스의 실행을 위해 설계된 XML언어에 시각적으로 표현 가능한 공통된 Notation을 제공하려는 목적으로 개발되었다. BPMN은 가독성(readability)을 위해 흐름도 표기법의 전통을 따르면서, 한편으로는 실행 가능한

\*이 논문은 2007년도 정부(과학기술부)의 재원으로 한국과학기술재단의 지원을 받아 수행된 연구임(No.R01-2007-000-20135-0). 이 논문은 2007년도 정부재원(교육인적자원부 학술연구조성사업비)으로 학술진흥재단의 지원을 받아 연구되었음 (KRF-2007-313-D00757)

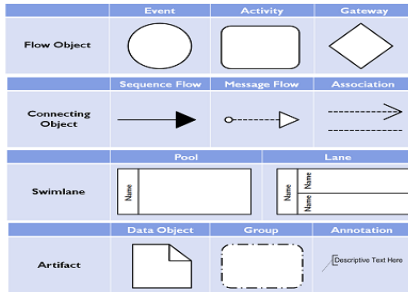
언어 구성체(constructs)로의 변환(mapping)을 제공한다  
아래의 그림은 현재 BPMN에서 제공하고 있는 BPMN 표준 구조이다.[2],[5]



(그림 1) BPMN 구조

## 2.2. BPMN의 구성요소

BPMN의 기본 구성 요소는 흐름객체(Flow Object), 연결객체(Connecting Object), 스윙레인(Swimlane), 부가객체(Artifact)로 구분할 수 있다.



(그림 2) BPMN 구성요소

흐름객체(Flow Object)는 프로세스의 실행 중에 발생하여 해당 프로세스의 흐름을 변경시키거나 다른 이벤트의 발생을 촉발시키는 이벤트(Event)와 실제 작업을 의미하는 액티비티(Activity), 그리고 프로세스의 흐름을 결정하는 게이트웨이(Gateway)로 구성된다.

연결객체는 풀(Pool)내의 액티비티들 사이의 연결을 정의하는 제어흐름객체(Sequence flow), 다른 풀들에 각각 존재하는 액티비티들 사이의 상호 통신을 정의하는 메시지흐름객체(Message flow), 프로세스내의 액티비티들과 관련 데이터나 설명문, 액티비티들의 그룹화 등으로 대표되는 부가객체(Artifact)들 사이의 연관 상태를 표현하는데 이용하는 연관성객체(Association)로 구성되어 있다.

스윙레인(Swimlane)은 풀(Pool)과 레인(Lane)으로 구성되어 있는데 BPMN에서는 풀과 풀 안의 레인이라는 그래픽 표기를 이용하여 프로세스와 그 프로세스에 연관된 액티비티들의 역할이나 조직의 구성단위를 가지적으로 정의한다.

마지막으로 부가객체(Artifact)는 비즈니스 프로세스의 제어흐름객체나 메시지흐름객체와 직접적으로 관련되지 않은 프로세스에 대한 추가적인 정보를 표시하기 위해 사용된다. 여기에는 문서, 데이터, 다른 오브젝트들이 프로세스 내에서 어떻게 사용되고 업데이트 되는지를 나타내는 데이터객체(Data Object), 프로세스 내의 요소들을 묶

어주는 그룹(Group), 다이어그램의 해석을 위해 추가적인 정보를 제공하는 주석(Annotation)으로 구성되어 있다.

## 3. 비즈니스 프로세스 모델에서의 이상 현상

비즈니스 프로세스를 모델링 할 때 발생할 수 있는 설계 오류들은 크게 3가지로 존재하는데 구문론적 오류(Syntactical Error), 구조적 오류(Structural Error), 그리고 의미론적 오류(Semantic Error)들이 있다.

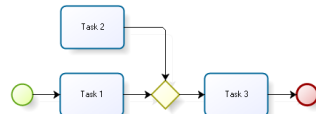
### 3.1. 비즈니스 프로세스 모델에서의 구문론적 오류

비즈니스 프로세스 구문론적 이상 현상은 실제 작업을 수행하는 액티비티 내부의 데이터 정보 및 토큰들을 고려하지 않으며 단지 모델링 요소들의 잘못된 사용만을 고려한다. 대표적인 구문론적 이상 현상은 흐름객체에서의 잘못된 사용, 연결객체에서의 잘못된 사용 그리고 스윙레인에서의 잘못된 사용 이와 같이 크게 3가지로 구분된다.

#### 3.1.1. 흐름객체에서의 잘못된 사용

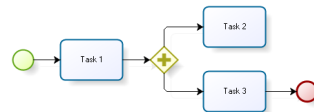
흐름 객체에서의 잘못된 사용은 이벤트, 액티비티 또는 게이트웨이의 잘못된 위치 선언으로 인해 생기는 이상 현상이다.

가) 시작이 없는 액티비티: 시작이 없는 액티비티는 시작 이벤트와 종료 이벤트가 모두 사용되었지만 도입 제어 흐름을 가지지 않는 흐름객체들이 존재하는 경우를 의미한다.



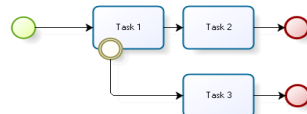
(그림 3) 시작이 없는 액티비티

나) 종료가 없는 액티비티: 종료가 없는 액티비티는 시작 이벤트와 종료 이벤트가 모두 사용되었지만 방출 순서 흐름을 가지지 않는 흐름 객체들이 존재하여 프로세스 종료가 수행되지 않는 경우를 의미한다.



(그림 4) 종료가 없는 액티비티

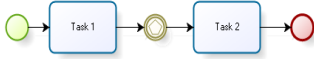
다) 액티비티 경계에 부착된 중간 이벤트의 잘못된 트리거 유형 사용: 액티비티 경계에 부착된 중간 이벤트의 잘못된 트리거 유형 사용은 트리거 유형이 None 또는 Link로 설정되어 액티비티 경계에 부착된 후 사용될 경우를 의미한다.



(그림 5) 액티비티 경계에 부착된 중간 이벤트의 잘못된

트리거 유형 사용 - None Trigger Type

라) 정상적인 순서 흐름에서 중간 이벤트의 잘못된 트리거 유형 사용: 정상적인 순서 흐름에서 중간 이벤트의 잘못된 트리거 유형 사용은 트리거 유형이 Cancel, Error, Multiple로 설정되어 정상적이 순서 흐름에서 사용될 경우를 의미한다.



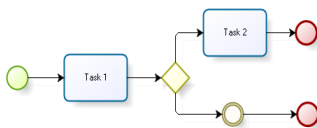
(그림 6) 정상적인 순서 흐름에서 중간 이벤트의 잘못된 트리거 유형 사용 - Multiple Trigger Type

마) 수신 태스크의 잘못된 사용: 수신 태스크를 사용하는 비즈니스 프로세스에서는 다음의 2가지 조건중 하나가 충족되어야 한다. 첫째 수신 태스크가 사용될 프로세스에서 시작이벤트와 도입순서 흐름을 가지지 않는다. 둘째 수신 태스크를 위한 도입순서 흐름을 가질 경우 도입순서 흐름은 항상 시작이벤트와 연결되어야 한다. 수신 태스크의 잘못된 사용은 위에 제시한 2가지 조건을 충족시키지 않고 사용될 경우를 의미한다.



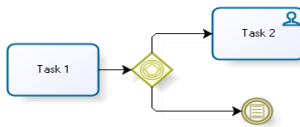
(그림7) 수신 태스크의 잘못된 사용

바) 데이터 기반의 XOR 게이트웨이의 잘못된 사용: 데이터 기반의 XOR 게이트웨이의 잘못된 사용은 일반 태스크를 목표로 한 방출 순서 흐름이 아닌 중간 이벤트를 목표로 한 방출 순서 흐름에서 사용될 경우를 의미한다.



(그림8) 데이터 기반의 XOR 게이트웨이의 잘못된 사용

사) 이벤트 기반의 XOR 게이트웨이의 잘못된 사용: 이벤트 기반의 XOR 게이트웨이의 잘못된 사용은 태스크 유형이 수신 태스크가 아닌 다른 유형의 태스크가 사용될 때 또는 중간 이벤트의 트리거 유형이 Message, Timer, Rule 또는 Link 가 아닌 다른 유형으로 사용될 경우를 의미한다.



(그림9) 이벤트 기반의 XOR 게이트웨이의 잘못된 사용

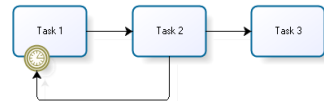
3.1.2. 연결 객체에서의 잘못된 사용

연결 객체에서의 잘못된 사용은 제어 흐름 객체, 메시지 흐름 객체 또는 연관성 객체의 잘못된 위치 선언으로 인

해 생기는 이상 현상이다.

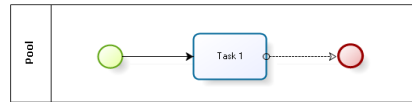
가) 시작 이벤트 및 종료 이벤트와 연결된 제어 흐름 객체의 잘못된 사용 : 시작 이벤트는 제어 흐름 객체 중 항상 나가는 방출 흐름 객체만을 가져야 한다. 또한 종료 이벤트는 제어 흐름 객체 중 항상 들어오는 도입 흐름 객체만을 가져야 한다.

나) 액티비티 경계에 부착된 중간 이벤트에서 제어 흐름 객체의 잘못된 사용: 액티비티 경계에 부착된 중간 이벤트는 절대로 순서의 목표가 될 수 없기 때문에 도입 흐름 객체를 가질 수 없다.



(그림10) 액티비티 경계에 부착된 중간 이벤트에서 제어 흐름 객체의 잘못된 사용

다) 메시지 흐름 객체의 잘못된 사용: 비즈니스 프로세스 모델링 단계에서 사용되는 메시지 흐름 객체는 시작 이벤트 또는 게이트웨이와 연결하여 사용될 수 없다.

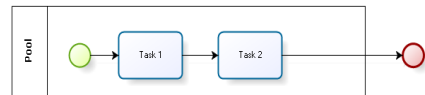


(그림11) 메시지 흐름 객체의 잘못된 사용

3.1.3. 스웸레인에서의 잘못된 사용

스웸레인에서의 잘못된 사용은 풀 또는 레인의 잘못된 위치 선언으로 인해 생기는 이상 현상이다.

가) 풀의 잘못된 사용: 풀 내부에 정의된 흐름 객체들 사이의 순서 흐름은 풀의 경계를 교차해서 사용될 수 없다. (단 메시지 흐름 객체는 풀의 경계를 교차하여 사용될 수 있다)



(그림 12) 풀 경계를 교차하여 표현된 이상 현상

나) 레인의 잘못된 사용: 비즈니스 프로세스 모델링 단계에서 레인은 풀 내부의 하부 파티 파티션으로 사용되기 때문에 풀 보다 먼저 사용될 수 없다.

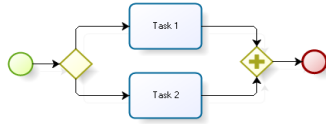
3.2. 비즈니스 프로세스 모델에서의 구조적 오류

비즈니스 프로세스 구조적 이상 현상은 잘못된 게이트웨이 사용으로 인하여 토큰의 병합 또는 분산되는 상호작용을 제어하지 못할 때 주로 발생한다. 대표적인 비즈니스 프로세스 구조적 이상 현상으로 데드락, 무한루프, 동기화의 부족, 수행되지 않는 액티비티가 존재한다.

3.2.1. 데드락

프로세스의 특정 단계에서 더 이상의 액티비티가 수행

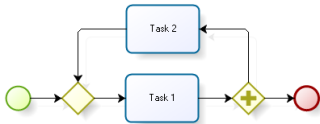
되지 않는 경우를 의미한다.



(그림 13) 데드락을 포함하는 프로세스

3.2.2. 무한루프

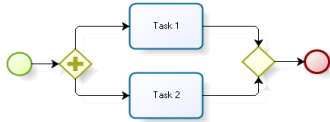
프로세스의 특정 단계에서 어떠한 액티비티가 영원히 반복적으로 수행되는 경우를 의미한다. 일반적으로 무한루프는 스플릿(Split) 게이트웨이에서 이전 액티비티 혹은 게이트웨이의 흐름객체가 존재할 때 발생한다.



(그림 14) 무한루프를 포함하는 프로세스

3.2.3. 동기화의 부족

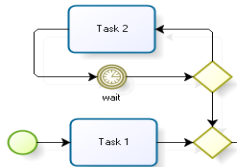
프로세스의 특정 단계에서 의도하지 않게 해당 액티비티가 여러 번 수행되는 경우를 의미한다. 일반적으로 동기화의 부족은 XOR-Join 게이트웨이에서 발생한다.



(그림 15) 동기화 부족을 포함하는 프로세스

3.2.4. 수행되지 않는 액티비티

프로세스 흐름에 따라 수행되지 않는 액티비티가 존재하는 경우를 의미한다.

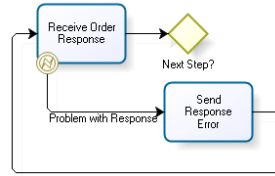


(그림 16) 수행되지 않는 액티비티를 포함하는 프로세스

3.3. 비즈니스 프로세스 모델에서의 의미론적 오류

비즈니스 프로세스에 있어서 의미론적 이상 현상은 비즈니스 프로세스의 최종 목표와 일치하지 않는 프로세스를 의미론적 이상 현상이 존재하는 프로세스라고 한다. 비즈니스 모델링 단계가 수행된 후 구현 단계에서 모델링한 비즈니스 프로세스를 BPEL로 변환하는 작업을 수행하게 되는데 이때 의미론적 오류를 포함하는 프로세스는 변환 과정 시에 오류를 포함하게 되거나 혹은 전체적인 비즈니스 프로세스가 완벽하게 매핑이 되지 않는 경우가 생길 수 있다. 아래 그림과 같이 예외 이벤트로 인해 발생한

흐름이 이전 액티비티가 수행을 마치기 전에 루프를 따라 이전 흐름으로 되돌아갔을 경우 비즈니스 프로세스가 완벽하게 매핑이 되지 않을 수 있다.



(그림 17) BPEL로 매핑되지 않는 의미론적 오류를 포함한 프로세스

위와 같은 경우에 태스크에서 과생된 또 다른 프로세스를 아래와 같이 연결 한 후 기존의 액티비티가 정상적으로 수행을 마칠 수 있을 때 까지 과생된 액티비티가 생성되어 있어야 한다.

4. 결론

본 논문에서는 비즈니스 프로세스를 모델링 하는데 있어서 어떠한 이상 현상들이 존재하는지 타입별로 구분하고 정의하여서 모델링 시에 혹은 모델링 도구에서 검증기능을 지원하고자 할 때 이러한 이상 현상들을 표본으로 삼아 잘 정의된 비즈니스 프로세스 모델을 생성하는데 도움을 주고자 하였다.

5. 참고문헌

- [1] 김학수, 박찬희, 설주영, 손진현, “컨트롤 흐름 경로 기반의 비즈니스 프로세스 타당성 검증 기법”, 정보처리학회 논문지D, August 3, 2007
- [2] 한국 전산원, “비즈니스 프로세스 관리를 위한 질의 언어 및 모델링 표기법 표준화 연구”, September 2005
- [3] Wasim Sadiq, Maria E. Orlowska, “On Correctness Issue in Conceptual Modeling of Workflows”, In Proceedings of the 5 th European Conference on Information Systems (ECIS '97), Cork, Ireland, June 19-21, 1997
- [4] Qianhui Althea Liang, J. Leon Zhao, “Verification of Unstructured Workflow via Propositional Logic”, Seventh IEEE/ACIS International Conference on Computer and Information Science, 2008
- [5] “BPMN Specification Release: BPMN v.1.1”, January 17, 2008