

## 클로즈 아키텍처 기반의 비즈니스 프로세스 프레임워크

서채연<sup>1</sup>, 김동우<sup>1</sup>, 김영철<sup>1\*</sup>  
<sup>1</sup>홍익대학교 소프트웨어공학연구소

### Business Process Framework based on the Closed Architecture

Chae-Yun Seo<sup>1</sup>, Dong-Woo Kim<sup>1</sup> and R. Young-Chul Kim<sup>1\*</sup>

<sup>1</sup>Software Engineering Lab., Hongik University

**요약** 현대의 기업은 급변하는 비즈니스 환경에 맞는 응용 프로그램을 적시에 개발이 필요하다. 그러나 대부분의 기업들은 비즈니스 프로세스 개념 없이 애플리케이션을 개발하여 업무 변경 시, 필요한 응용프로그램을 적시에 개발하기 어렵다. 이를 해결하기 위해 클로즈 아키텍처 기반의 비즈니스 프로세스 프레임워크 제안한다.

이는 BPM(Business Process Model)과 SOA(Service Oriented Architecture), 그리고 CBD(Component Based Development) 접목을 통해 급변하는 비즈니스 프로세스 변경/개선이 용이하며, 이를 지원하는 응용프로그램을 재사용을 통해 적시에 개발하고 자 한다. 본 논문에서는 비즈니스 프로세스 프레임워크 내에 5-Layer 구조 및 그에 따른 BNF(Backus Naur Form)을 정의하였다.

**Abstract** Modern enterprise may be in need of developing the right application programs under rapidly changing business environment in fast time-to-market. But Most of enterprise without business process mechanism are very hard to develop the application needed in time when it will be changing their business. To solve this problem, we propose the business process framework based on the closed architecture. This approach is to develop the right application with reusing software modules in time and easily under rapid business process changing/improving through mapping BPM(Business Process Model), SOA(Service Oriented Architecture), and CBD(Component Based Development). we also define 5-Layers of BPF(Business Process Framework), and BNF(Backus Naur Form) for structuring BPF.

**Key Words** : Service, BPM, CBD, Layer Architecture, BPF(Business process framework), BNF

### 1. 서론

현재의 기업은 빠르게 변화하는 비즈니스 환경 속에서 최소한의 정보기술로 다양한 비즈니스 프로세스를 기업 내/외부의 요구에 쉽고 빠르게 전환이 필요한 실정이다. 그러나 많은 기업의 시스템들에는 복잡한 애플리케이션들로 혼재되어서, 기존의 비즈니스 프로세스를 변경하거나 새로이 개발/유지 보수하는 데 어려움이 있다. 이러한 환경은 급변하는 시장에서 기업의 신속한 비즈니스 규칙 수립과 의사 결정을 지연시켜, 시장 경쟁력을 저해시키는 요인이 되고 있다[1]. 이를 해결하려는 최근 이슈가 되고

있는 것은 비즈니스 인텔리전스 환경에서 서비스를 이용하는 것이다[2]. 민첩한 IT환경을 SOA를 통해 비즈니스 프로세스에 접근하여, 서비스를 추출해서 독립적으로 구현하고, 새로운 비즈니스 프로세스를 창출하고, 구현된 서비스를 어디에서든지 제공이 가능해야한다. 이를 통해, BPM에서 서비스를 적용하여 빠른 비즈니스 프로세스를 개선하고 재사용이 가능할 것이다. 컴포넌트와 서비스의 재사용은 현대의 소프트웨어 개발의 많은 핵심 쟁점을 해결할 수 있는 잠재력이 있다. 시장 변화에 대해 빠르게 반응하여 단기간 시장 진입이 가능하고, 개발과 관리 비용을 감소시킴으로써 향상된 품질을 만들 수 있다[2].

본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터육성지원사업의 연구결과로 수행되었음(C1090-0903-0004).

본 논문은 2007학년도 홍익대학교 학술연구진흥비에 의하여 지원되었음.

\*교신저자 : 김영철(bob@hongik.ac.kr)

접수일 09년 07월 08일

수정일 09년 08월 12일

계재확정일 09년 08월 19일

컴포넌트와 서비스 재사용의 궁극적인 목적은 다음과 같은 비즈니스 상의 많은 이점을 얻기 위한 것이다[3]. 첫째는 향상된 생산성과 짧아진 리드타임을 통해 일을 덜고, 둘째는 향상된 품질을 통해 에러의 수를 줄이고, 셋째는 향상된 생산성을 통해 유지보수 비용을 줄이기 위해, 넷째는 재사용 시 필요한 전체적인 요구분석을 통한 재작업을 줄이기 위한 것이다[4]. 비즈니스 프로세스 모델링을 통해 전체 업무 프로세스 모델링을 최적화하여 빠르게 변하는 비즈니스 프로세스 변화에 효율적으로 대처하려 한다.

소프트웨어 공학의 컴포넌트 기반 개발이란 컴포넌트를 재사용 하여 소프트웨어를 개발하기 위한 방법론이다 [11]. 개발된 컴포넌트는 컴포넌트 레파지토리에 저장하고, 저장된 컴포넌트는 필요한 사용자에게 의해 재배치/재사용 가능하다.

본 논문은 비즈니스 프로세스와 컴포넌트를 접목하는 3-Layer 구조[1]를 확장한다. 개발자 관점에서는 비즈니스 프로세스 모델링과 서비스를 접목하여 효율적인 개발을 하고, 관리자 관점에서 각 레이어를 클로즈 아키텍처로 정의하여 관리하기 편리하게 BPM의 5-Layer구조를 제안한다.

본 논문은 2장 비즈니스 프로세스 프레임워크, 3장 각 레이어 구성을 위한 BNF를 설명하고, 4장에서는 결론 및 향후 과제에 대해 기술한다.

## 2. 비즈니스 프로세스 프레임워크

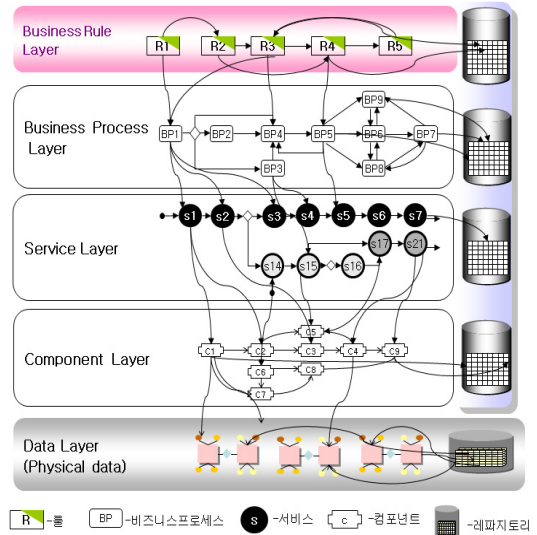
본 장에서는 새로운 비즈니스 프로세스의 창출 및 변경에 대해 효율적으로 그에 맞는 응용 프로그램을 적시에 개발하기 위해 BPM과 서비스, 서비스와 컴포넌트를 접목하고, 비즈니스 프로세스 프레임워크 5-Layer구조의 확장 및 개선을 제안한다[3].

### 2.1 비즈니스 프로세스 프레임워크

비즈니스 프로세스 프레임워크의 레이어 구조는 아래와 같은 특성을 갖는다[3]. 각 레이어는 독립성을 유지한다. 한부분의 변경이 다른 부분에 미치는 영향을 최소화하여 안정적인 환경을 제공한다. 유지보수가 용이하고, 시스템 모듈의 의존성을 최소화하여 안정적 구조를 지닌다.

각 레이어의 레파지토리는 테이블 화하여, 개발 할 BPSQL(Business Process Structured Query Language)을 통해 각 레이어별 데이터 쿼리를 생성하여 필요한 정보

를 추출하고자 한다. 레이어 구조는 비즈니스 요구 발생 시 기 존재하는 재사용 컴포넌트를 활용하여 신속하게 신규 서비스를 생성할 수 있으며, 생성된 서비스로 신규 비즈니스를 구성할 수 있다. 그림 1은 비즈니스 프로세스 프레임워크이다. 클로즈 아키텍처 기반으로 BPM과 서비스 그리고 CBD를 접목하여 5-Layer 구조로 확장한 것이다[3].



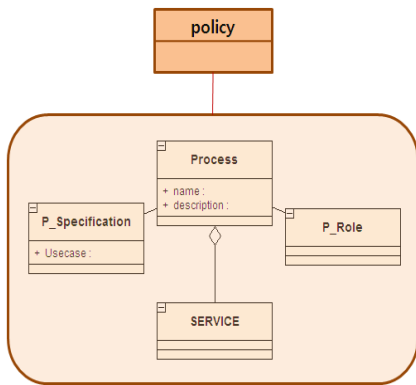
[그림 1] 클로즈 아키텍처 기반의 비즈니스 프로세스 프레임워크

최상위의 5-Layer는 비즈니스 규칙, 4-Layer는 비즈니스 프로세스, 3-Layer는 서비스, 2-Layer는 컴포넌트, 마지막 레이어는 실질적인 데이터가 존재하는 데이터베이스 즉, 데이터 모델링 레이어이다. 그리고 각 레이어에는 레파지토리가 존재한다. 이 프레임워크는 BPM의 5-Layer 구조는 각 레이어의 바로 다음 단계와 직접 연결된 구조인 클로즈 아키텍처로써, 각 레이어의 관리 및 유지 보수가 쉽다.

각각의 레이어는 다음과 같이 정의한다.

#### 2.1.1 비즈니스 규칙 레이어(Business Policy Layer)

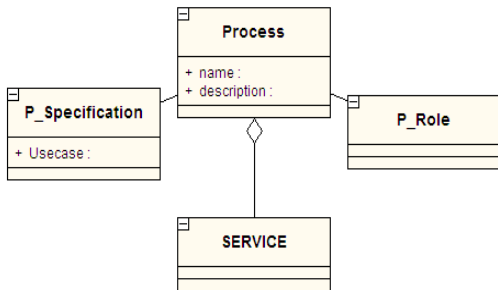
비즈니스 규칙은 ‘비즈니스와 관련해서 특정 조건의 충족 여부에 따라 적용되거나 적용되지 않는 규칙이나 방침’이라고 정의한다[5]. 그림 2는 policy를 클래스 다이어그램을 이용하여 UML로 표기하였다. 규칙은 각 프로세스에 영향을 주고, 규칙에 따라 프로세스를 수행한다.



[그림 2] UML의 클래스 다이어그램을 이용한 policy 정의

### 2.1.2 비즈니스 프로세스 레이어 (Business Process Layer)

특정한 목표나 목적을 달성하기 위한 활동, 작업 및 절차들의 집합을 프로세스라고 한다[6].



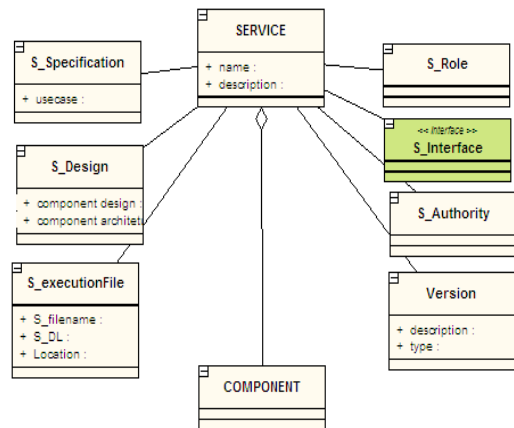
[그림 3] UML의 클래스 다이어그램을 이용한 프로세스 정의

프로세스는 주어진 목적을 위해 수행되는 일련의 단계들로, 어떤 목적 달성을 위해 무엇을 어떻게 해야 하는지에 대한 구체적인 활동 절차들로 구성된다[5]. 개발자는 비즈니스 프로세스를 디자인하는데, 비즈니스의 요구사항이 변경에 쉽게 대처하기 위해 서비스들의 조합으로 쉽게 시스템을 구축할 수 있다. 또한, 변화에 유연한 비즈니스를 구축하여 비즈니스의 변화의 속도에 따른 적절한 프로세스를 구축할 수 있다. 그림 3은 비즈니스 프로세스를 UML의 클래스 다이어그램으로 표현한 것이다. 비즈니스 프로세스는 스펙과 롤이 있으며, 하나의 서비스 또는 여러 개의 서비스로 구성된다. 각 프로세스는 고유 이름이 있고, 스펙은 유스케이스로 표현가능하다.

### 2.1.3 서비스 레이어(Service Layer)

서비스 레이어는 서비스의 유연성, 가시성, 유용성을

갖고 있다. 서비스의 유용성은 각 서비스 간 의존성을 최소화한 아키텍처이다. 가시성은 서비스의 흐름이 외부에 노출되어 이해가 쉽다. 유용성은 기 제작된 서비스 및 모듈의 재사용으로 개발자가 편리하고 쉽게 활용할 수 있다. 이러한 특징을 통해 급변하는 환경 변화에 능동적으로 대처할 수 있을 뿐 아니라 다양한 커스트 마이징 요건을 쉽게 수용할 수 있는 유연성을 보장한다[8]. 본 논문에서 서비스는 여러 개의 컴포넌트가 하나의 서비스를 만들고, 여러 서비스들은 하나의 비즈니스 프로세스를 만든다[9].

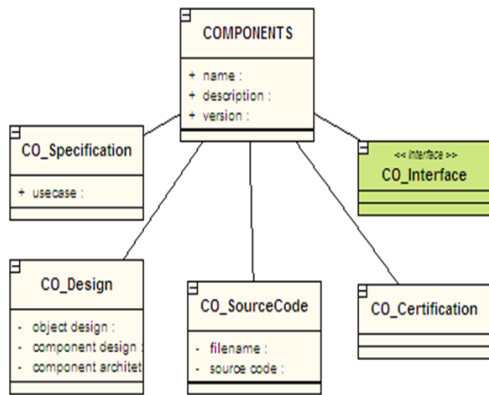


[그림 4] UML의 클래스 다이어그램을 이용한 서비스 정의

그림 4는 정의한 서비스의 스펙을 이용하여 UML의 클래스 다이어그램으로 도식화 한다[2]. 서비스는 스펙, 디자인, 실행 파일, 역할, 인터페이스, 인증, 버전을 갖는다. 이러한 서비스는 여러 개의 컴포넌트로 구성된다[2]. 서비스의 스펙은 유스케이스로 표현하고, 디자인에는 서비스를 구성하는 컴포넌트의 디자인과 구조를 포함한다. 실행 파일에는 파일 명과 묘사 언어 (description language) 가 있고, 로케이션을 통해 서비스를 구성하는 컴포넌트의 위치를 알 수 있다.

### 2.1.4 컴포넌트 레이어(Component Layer)

컴포넌트는 동적인 흐름을 갖는 워크플로우 형태이다. 컴포넌트는 여러 개의 컴포넌트, 컴포넌트 워크플로우로 구성된다. 컴포넌트는 내포된 하위컴포넌트를 가질 수 있다. 도메인 분석 방법론을 통해 추출한 컴포넌트(WODA)를 테이블화한 컴포넌트 레퍼지토리에 저장한다[2].



[그림 5] UML의 클래스 다이어그램을 이용한 컴포넌트 정의

그림 5는 컴포넌트를 UML의 클래스 다이어그램으로 표현하였다. 컴포넌트는 스펙, 디자인, 소스코드, 인증, 인터페이스가 존재한다. 컴포넌트는 버전을 추가하면서 업데이트 할 수 있다. 컴포넌트의 스펙은 유스케이스로 표현한다. 컴포넌트 디자인에는 객체디자인과 컴포넌트 디자인, 구조를 포함한다. 인증을 통해 유용한 컴포넌트를 식별 할 수 있다.

### 2.1.5 데이터 레이어(Data Layer)

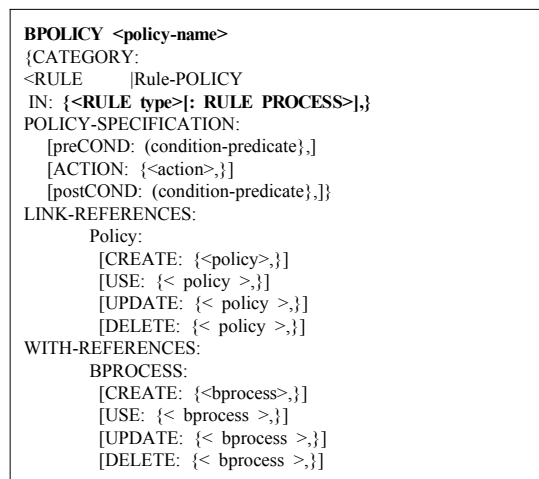
실질적으로 물리적인 공간에 데이터가 저장되는 곳이다. 상위 4-Layer가 수행되면서 이 레이어에 실제 데이터가 접근하여 사용한다. 데이터는 E-R 다이어그램으로 모델링되고, 레파지토리에 모든 데이터들은 테이블 화되어 저장된다. 그 이유는 기존의 질의 언어의 확장을 통해 쉬운 접근성과 빠른 정보 검색을 하기 위함이다[10]. 데이터 레이어는 E-R 다이어그램으로 모델링한다.

## 3. 5-Layer구조를 위한 BNF (Backus Naur Form)

비즈니스 프로세스 프레임워크 내 각 레이어는 BNF를 이용하여 설계 한다.

### 3.1 비즈니스 규칙 (Business Policy)

그림 6은 BPF(Business Process Framework)의 최상위 층인 비즈니스 규칙 모델링 레이어의 BNF이다.

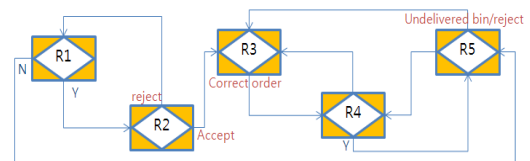


[그림 6] 비즈니스 규칙 레이어의 BNF

비즈니스 규칙에 맞게 비즈니스 프로세스 모델링에 적용한다. 룰 확인은 선 조건을 확인한 후, 액션을 취한다. 액션이 룰에 올바른지 검사하기 위해 후 조건을 체크한다. 표 2는 비즈니스 규칙에 대한 표현기호이다.

[표 2] 비즈니스 규칙에 대한 표현기호

표현기호	설 명
	규칙
	조건 규칙
	각 규칙을 연결하는 선

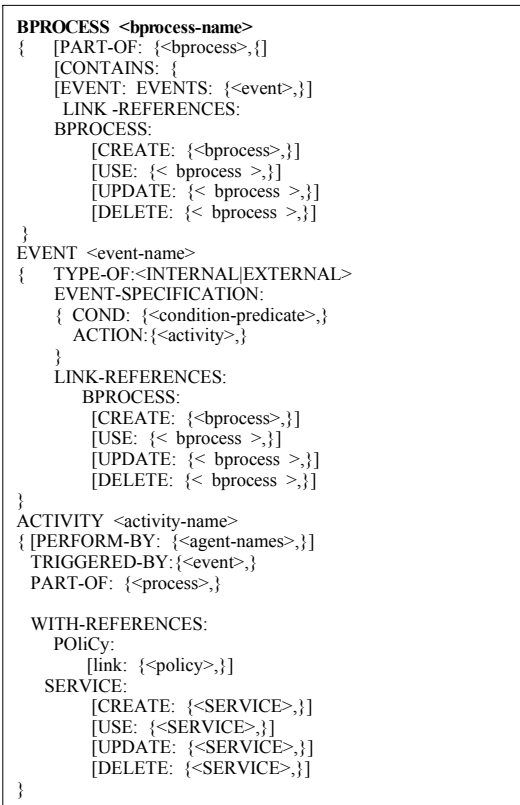


[그림 7] 비즈니스 규칙 모델링

그림 7은 비즈니스 규칙을 BNF로 모델링하였다. R1은 등급, R2는 인증, R3은 확인, R4는 신용, R5는 정보에 관한 규칙들이다.

### 3.2 비즈니스 프로세스 (Business Process)

그림 8은 비즈니스 프로세스 모델링 레이어의 BNF인 비즈니스 프로세스를 정의한 것이다.

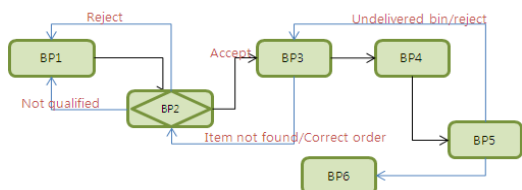


[그림 8] 비즈니스 프로세스 레이어의 BNF

표 4는 비즈니스 프로세스에 대한 표현기호이다. 그림 9은 물품을 구매하기 위한 비즈니스 프로세스이다.

[표 4] 비즈니스 프로세스 표현기호

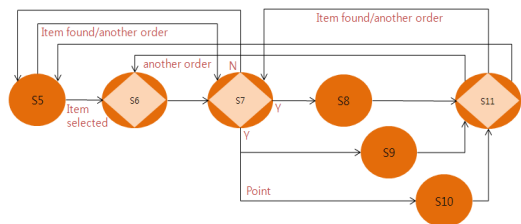
표현기호	설명
	비즈니스 프로세스
	분기가 있는 비즈니스 프로세스
	각 비즈니스 프로세스를 연결하는 선



[그림 9] 비즈니스 프로세스 모델링

### 3.3 서비스 (Service)

비즈니스 프로세스 하위 레이어에서 실행되는 서비스들이다. 본 논문에서 정의한 서비스 스펙을 간단히 살펴보면, 여러 개의 컴포넌트로 구성된 서비스에는 이름과 설명이 있다. 서비스 버전을 통해 새로운 서비스와 구분할 수 있고, 서비스 인터페이스를 통해 각 서비스와 커뮤니케이션 할 수 있다. 표 6은 서비스 표현 기호에 대한 설명이다. 표 7은 모델링된 서비스를 설명한다. 물품 구매 프로세스의 하위 레이어의 서비스들 중 모델링된 서비스를 설명한다. 그림 10은 서비스 모델링에 대한 시나리오이다.

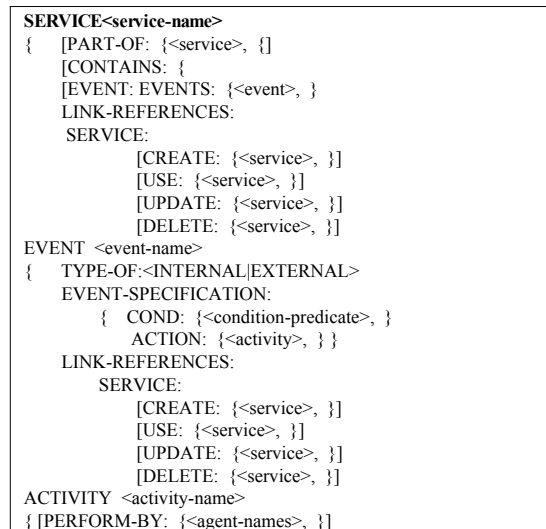


[그림 10] 서비스 모델링

[표 6] 서비스 표현기호

표현기호	설명
	서비스
	조건 서비스
	각 서비스를 연결하는 선

그림 11는 BPF의 서비스층 BNF를 정의한다.



```

TRIGGERED-BY: {<event>, }
PART-OF: {<process>, }
LINK-REFERENCES:
  SERVICE:
    [CREATE: {<SERVICE>,}]
    [USE: {<SERVICE>,}]
    [UPDATE: {<SERVICE>,}]
    [DELETE: {<SERVICE>,}]
  WITH-REFERENCES:
    BPROCESS:
      [link: {<bprocess>,}]
    COMPONENT:
      [CREATE: {<component>, } ]
      [USE: {<component>, } ]
      [UPDATE: {<component>, } ]
      [DELETE: {< component >, }]}
    
```

[그림 11] 서비스 레이어의 BNF

### 3.4 컴포넌트 (Component)

그림 12은 컴포넌트 모델링 레이어의 BNF를 정의한다. 컴포넌트는 WODA(Workflow Oriented Domain Analysis)로 추출되고[5], 정의한 컴포넌트 스펙을 이용하여 작성한다. 컴포넌트는 여러 개의 컴포넌트, 컴포넌트 워크플로우로 구성된다. 컴포넌트는 내포된 하위컴포넌트를 가질 수 있다.

[표 8] 컴포넌트 표현기호

표현기호	설명
	컴포넌트
	조건 컴포넌트
	각 컴포넌트를 연결하는 선

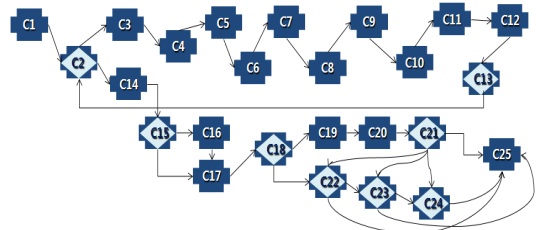
```

COMPONENT<component-name>
{ [PART-OF: {<component>, } ]
  [CONTAINS: {
    [EVENT: EVENTS: {<event>, } ]
    LINK-REFERENCES:
      COMPONENT:
        [CREATE: {<component>, } ]
        [USE: {<component>, } ]
        [UPDATE: {<component>, } ]
        [DELETE: {<component>, } ]
  ]
  EVENT <event-name>
  { TYPE-OF:<INTERNAL|EXTERNAL>
    EVENT-SPECIFICATION:
      { COND: {<condition-predicate>, } }
      ACTION: {<activity>, } }
    LINK-REFERENCES:
      COMPONENT:
        [CREATE: {<component>, } ]
        [USE: {<component>, } ]
        [UPDATE: {<component>, } ]
        [DELETE: {<component>, } ]
  ]
  ACTIVITY <activity-name>
  { [PERFORM-BY: {<agent-names>, } ]
    TRIGGERED-BY: {<event>, } ]
    PART-OF: {<process>, } ]
    LINK-REFERENCES:
  }
}
    
```

```

COMPONENT:
[CREATE: {<COMPONENT>,}]
[USE: {<COMPONENT>,}]
[UPDATE: {<COMPONENT>,}]
[DELETE: {<COMPONENT>,}]
WITH-REFERENCES:
  SERVICE:
    [link: {<service>,}]
  DATA:
    [CREATE: {<data>, } ]
    [USE: {<data>, } ]
    [UPDATE: {<data>, } ]
    [DELETE: {<data>, } ]
    
```

[그림 12] 컴포넌트 레이어의 BNF



[그림 13] 컴포넌트 모델링

표 8은 컴포넌트 표현기호이다. 그림 13에 모델링된 컴포넌트이다.

### 3.5 데이터 (Data)

데이터 모델링은 시스템을 구축하는데 필요한 각각의 데이터들을 E-R 다이어그램으로 표현한다. 구매프로세스에 대한 시스템을 구축하기 위해선 회원 관리 테이블, 판매테이블, 재고테이블, 배송테이블, 결제테이블, 물품테이블 등으로 구성된다. 그림 14는 BPF의 최하위층인 데이터 레이어의 BNF이다.

```

DATA <DATA-name>
{[PART-OF: {<data>, } ]
  [CONTAINS: {
    [EVENT:
      START-EVENT: {<event>, } ]
      EVENTS: {<event>, } ]
      END-EVENT: {<event>, } ]
    LINK-REFERENCES:
      DATA:
        [CREATE: {<data>, } ]
        [USE: {<data>, } ]
        [UPDATE: {<data>, } ]
        [DELETE: {<data>, } ]
    EVENT <event-name>
    { TYPE-OF:<INTERNAL|EXTERNAL|TIME>
      EVENT-SPECIFICATION:
        { COND: {<condition-predicate>, } }
        ACTION: {<activity>, } }
    LINK-REFERENCES:
      DATA:
        [CREATE: {<data>, } ]
    }
  }
}
    
```

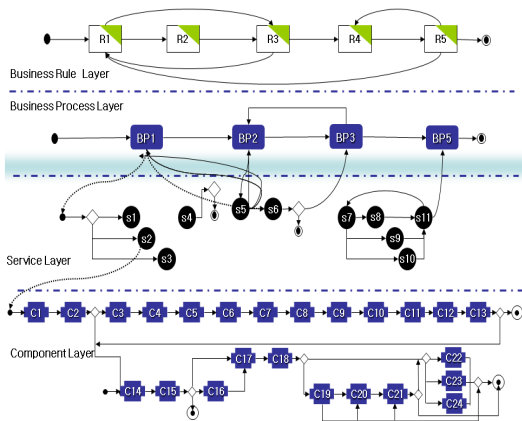
```

[USE: {<data>, }]
[UPDATE: {<data>, }]
[DELETE: {<data>, }]
ACTIVITY <activity-name>
{[PERFORM-BY: {<agent-names>, }]
START-TIME:<time>
END-TIME:<time>
TRIGGERED-BY: {<event>, }
PART-OF: {<process>, }
LINK-REFERENCES:
DATA:
[CREATE: {<data>, }]
[USE: {<data>, }]
[UPDATE: {<data>, }]
[DELETE: {<data>, }] }
WITH-REFERENCES:
COMPONENT:
[CREATE: {<COMPONENT>, }]
[USE: {<COMPONENT>, }]
[UPDATE: {<COMPONENT>, }]
[DELETE: {<COMPONENT>, }] }
ACTIVITY-METHODS:
{<method name>
IMPLEMENTED BY: <author-name>
LOCATION: <full path name of the source files>
} }
    
```

[그림 14] 데이터 레이어의 BNF

### 3.6 비즈니스 프로세스 프레임워크 내의 layer 들 간의 접목

다음은 온라인 쇼핑몰의 상품구매 프로세스에 대한 적용 사례이다. BPF로 구조를 정의하여 구매 프로세스의 표준화와 향후 업무 개선에 효율적으로 대처하기 위한 기반을 마련하고자 한다.



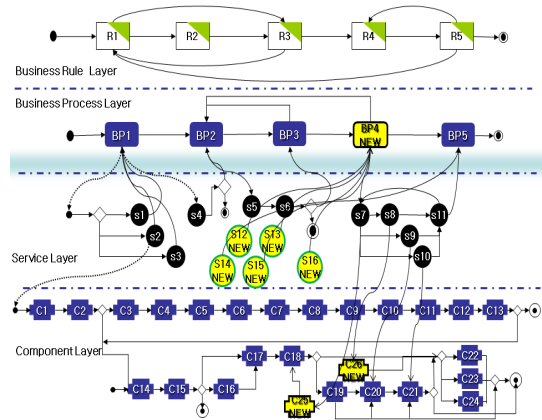
[그림 15] 구매 프로세스 중 결제 프로세스 5-Layer 구조

그림 15은 온라인 쇼핑몰의 구매 프로세스 모델링이다. 구매 시스템의 5-Layer는 룰과 프로세스, 서비스, 컴포넌트의 구조를 갖는다. 최상위는 비즈니스 프로세스 룰을 모델링한다. 각 룰을 수행하는 비즈니스 프로세스를

모델링하고, 각 비즈니스 프로세스에서 서비스를 추출하여 서비스를 모델링한다. 서비스는 하나의 컴포넌트 또는 여러 컴포넌트로 생성할 수 있다.

그림 15은 현재 사용 중인 구매 시스템이다. 구매 프로세스 중 결제 서비스는 오직 현금으로만 거래된다.

신용카드 결제를 생성하고, 각 레파지토리에 저장된 서비스와 컴포넌트를 추가하여 새로운 결제 프로세스를 개발한다.



[그림 16] 서비스와 컴포넌트를 추가하여 만든 결제 프로세스

그림 16은 기존 업무 프로세스의 변경으로 필요한 업무환경을 제공해주기 위해 소프트웨어 재사용을 통해 개발한 사례이다. 각 레이어의 굵은 테두리로 표시되고 “New”라고 표기된 것이 새로 추가된 비즈니스 프로세스, 서비스, 컴포넌트이다. 시스템을 레이어 구조로 정의하여 설계하였기 때문에 편리한 동적설계가 가능하고, 각 레이어 내 데이터 접근이 용이하다. 이는 비즈니스 환경 변화에 민첩하게 대응 할 수 있는 시스템 구축 또한 가능하다. 프레임워크 구축으로 각 레이어간 상호 작용으로 원활한 관리 및 정보 검색 효율화를 이루고 각 프로세스 간 접근 용이성이 좋다. 프레임워크 표준화에 의해 업무 능률이 향상되고 관리를 효율적으로 할 수 있다.

## 4. 결론

본 논문에서는 효율적인 비즈니스 프로세스 관리를 위해 비즈니스 프로세스 프레임워크 5-Layer 구조를 제안하였다. 전체 시스템을 레이어 아키텍처로 구성함으로써 각 레이어를 동적/정적인 구조를 설계할 수 있고, 전체 시

시스템 내 각 레이어간 효율적인 데이터 추적이 가능하다. 또한, 모든 stakeholder들의 관점에 따라 시스템 관리가 가능하다. 개발자 관점인 Bottom-up면에서는 시스템 구축 시 레이어 간 비즈니스 프로세스와 서비스의 접목을 통해 새로운 업무를 개발한다. 그리고 레파지토리에 저장된 컴포넌트와 서비스를 효율적으로 재사용하여 서비스와 비즈니스 프로세스를 구현 하면 개발 시간과 비용을 절약할 수 있다. 또한, 비즈니스 프로세스가 동적으로 변화는 업무에 잘 적응할 수 있도록 모델링 가능하다.

향후 시스템의 성능이 향상된 프레임워크인 오픈 아키텍처에 대한 연구가 필요하다. 모든 레이어를 연결 할 수 있는 오픈 아키텍처 상에서의 BNF와 SQL을 사용한다면 모든 레이어가 연결 가능하다. 또한, 최하위 레이어의 컴포넌트 레파지토리의 관계 테이블 화에 대한 연구도 같이 진행되고 있다.

### 참고문헌

[1] 서운숙, 김영철, “확장된 BPM과 컴포넌트 기반방법론 접목에 관한 연구”,홍익대학교, 2005.

[2] 서운숙, 김영철, “A Closed Architecture기반의 3Layer”,한국모바일학회, 11월, 2006.

[3] 서채연, 김동우, 김재수, 김영철,“효율적인 비즈니스 프로세스 모델링을 위한 5-Layer Architecture”, IWT Vol. 6, No. 1, 19-22, 5월, 2008.

[4] 김윤정, 워크플로우 메커니즘을 통한 소프트웨어 컴포넌트 식별 방법론에 관한 연구, 홍익대학교, 2004.

[6] Howard Smith. "Business Process Management". 시그마인사이트컴. 2004.

[10] P. Bichler, G. Preuner, M. Schrefl, "Workflow Transparency", Conference on Advanced Information Systems Engineering, 1997.

[11] Roger S. Pressman "Software Engineering A Practitioners' Approach" 3rd Ed. McGraw Hill, 2004.

[12] BMP FORUM, “BMP 2009 FORUM”, 2004~9.

[13] 이정권, “BEA 솔루션데이2007-HP”, 3월, 2007.

[14] par Hedley Apperly, Ralph Hoffman, et Steve Latchem “Service-And Component-Based Development

[15] Oracle SOA Architect Forum, 2009.1.8.

[16] Bea Aqualogic BPM2007(why BPM + SOA), 2007.06.19.

[17] Hedley Apperly,Ralph Hofman etc, Service- and Component-Based Development: Using the Select Perspective and UML,Addison-Wesley, 2003.

[18] Tomas Earl, Service-oriented architecture : concepts

,technology, and design, 2006.

[19] Tomas Earl,, Service-oriented architecture : a field guide to integrating XML and web service, 2006.

### 서 채 연(Chae-Yun Seo)

[정회원]



- 2005년 2월 : 홍익대학교 일반대학원 전자전산과 (공학석사)
- 2008년 2월 : 홍익대학교 일반대학원 전자전산과 (박사수료)
- 2009년 3월 ~ 현재 : 유한대학 경영정보 겸임교수

<관심분야>

CBD, 컴포넌트, BPM, SOA, 서비스

### 김 동 우(Dong-Woo Kim)

[정회원]



- 2005년 2월 : 홍익대학교 일반대학원 전자전산과 (공학석사)
- 2009년 2월 : 홍익대학교 일반대학원 전자전산과 (박사수료)
- 2005년 3월 ~ 현재 : 교보정보통신 시스템 통합 사업1팀

<관심분야>

EAI, BPM, SOA, 서비스, 컴포넌트

### 김 영 철(R. Youngchul Kim)

[정회원]



- 2000년 : 일리노이공대 전자계산학과 (이학박사)
- 2000년 3월 ~ 2001년 2월 : LG산전 중앙연구소 임베디드 시스템 부장
- 2001년 3월 ~ 현재 : 홍익대학교 컴퓨터정보통신 부교수

<관심분야>

CBD, 컴포넌트, BPM, TTM(Test Maturity Model), RBT