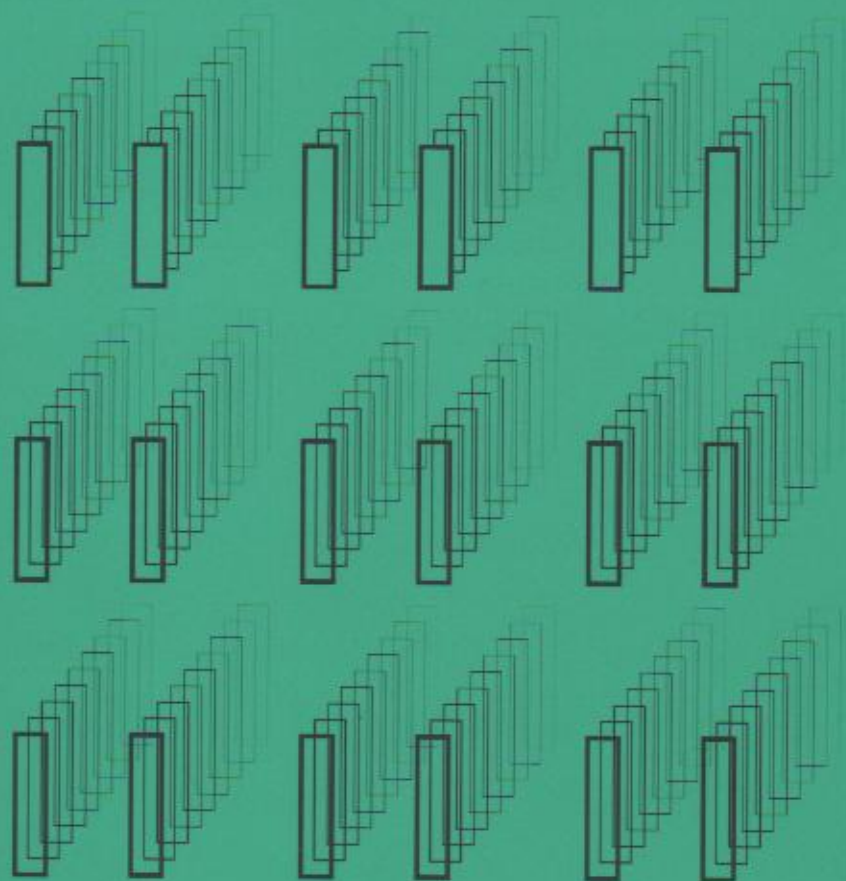


弘大論叢 XXXIX

# 産業技術

第17輯



弘益大學校 産業技術研究所

2007

## 目 次

RecurDyn을 이용한 Mold Ejector Sleeve Pin의 와이어 컷 방전가공 메카니즘 시뮬레이션	권 영 주	5
벤틸레이티드 디스크 브레이크의 열응력 해석	김 양 술 안 수 철	15
선조가공 방식의 Worm Roll Cylindrical Dies의 소재특성 및 절삭성에 관한 연구	최 창 섭	29
산화물 내에서의 산소의 화학 확산계수	백 현 덕	39
펄스 트랜스포머의 새로운 설계에 관한 연구	김 영 주 김 상 훈	51
퍼지제어기를 이용한 풍력발전용 이중여자 유도기의 출력제어	장 교 범	61
Mach band 현상의 해석을 위한 시각 필터 연구	김 장 한	77
Contourlet 변환을 사용한 효과적인 방향성의 다해상도 영상 표현에 관한 연구	지 인 호 장 선 봉	87
비즈니스 프로세스 질의 언어에 대한 연구	김 영 천 김 동 우 서 운 숙	105
직접에탄을 연료전지용 촉매에 관한 연구	이 병 철 김 영 목	111
전시공간의 자연채광에 관한 연구	김 동 진	123
건축공간에 나타난 오브제의 기능에 관한 사례 연구	송 병 하 소 은 탁	133



## 비즈니스 프로세스 질의 언어에 대한 연구

### A Study on Business Process Structured Query Language(BPSQL)

金 永 哲\*

Kim, R. Young-Chul\*

金 動 佑\*\*

Kim, Dong-Woo\*\*

徐 潤 淑\*\*

Seo, Yun-Suk\*\*

#### < ABSTRACT >

We suggested three layer architecture of BPM(Business Process Model) for easily managing and changing business process. But this paper mentions the six layer on the closed architecture for efficiently managing the business process and quickly developing business process changed, and also BPSQL(Business Process Structured Query Language) for controlling the information retrieval on each layer. Therefore, we will work more easily and efficiently to model new changing business process and to retrieve information than the previous business process management and BPQL (Business Process Query Language)

이 논문은 2007년도 홍익대학교 학술연구 조성비에 의하여 연구되었음.

\* 科學技術大學 副教授, Associate Professor, College of Science & Technology

\*\* 科學技術大學 大學生, Graduate Student, College of Science & Technology

# I. 서 론

과거에 비해 비즈니스 프로세스가 다양해 지고 복잡해짐에 따라 비즈니스 프로세스 통합에 관한 연구가 활발하게 진행되고 있다. 서로 다른 환경을 가진 기업들 간의 비즈니스 프로세스를 통합하기 위해서는 표준화가 진행이 되어야한다. 본 논문에서는 프로세스를 모델링할 수 있는 언어들을 소개하고 우리가 제안하고자 하는 BPSQL을 제시하고자 한다. 오늘날 비즈니스 프로세스 언어는 BPM(Business Process Management)에서 사용되는 언어이다. BPM은 확장된 기업의 프로세스를 자동화하고 통합하고 최적화하기 위해 설계된 새로운 세대의 기술을 일컫는다. 현재 BPM의 최대 목적은 비즈니스 프로세스를 표준에 맞춰 설계하고 설계된 프로세스를 자동으로 운영해 주는 것이다. BPMS(Business Process Management System)는 프로세스 운영을 위해 사용되는 BPM을 실현해주는 도구이다. BPMS는 BPQL(Business Process Query Language) 인터페이스를 지원하는 비즈니스 운영환경으로서, BPML로 설계된 비즈니스 프로세스들을 실제로 실행시키고 관리할 수 있는 시스템을 일컫는다. 이는 XML 기반 표준 사양인 BPML BPMN 그리고 BPQL BPEL에 의해 가능하게 된다. 이 언어를 사용하여 비즈니스 프로세스를 모델링하고 구현한다면 빠르고 효과적인 비즈니스 프로세스를 생성할 수 있다. 비즈니스 프로세스의 설계를 위해 BPML(Business Process Modeling Language)과 BPMN (Business Process Modeling Notation)이 사용되고, 실질적인 비즈니스 프로세스 운영을 위해 BPQL을 사용한다.

우리는 비즈니스 프로세스를 효율적으로 생성하고 관리하기 위해 BPM의 6-layer 구조를 제안하고, BPM의 프레임워크를 각 계층별로 관리할 수 있는 언어인 BPSQL(Business Process System Query Language)을 정의한다. BPSQL은 각 계층의 데이터를 테이블화하여 레퍼지토리에 저장하고, 저장된 데이터를 재사용하기 쉽다. 제안한 BPSQL은 데이터베이스의 쿼리문을 다룰 수 있는 이용시라면 쉽게 사용하다는 장점이 있다.

본 논문의 구조는 다음과 같다. 제 2장에서는 관련연구로서 기존의 비즈니스 프로세스 언어에 대해 설명한다. 제 3장에서는 우리가 제안한 BPSQL을 설명한다.

## II. 관련 연구

비즈니스 프로세스 언어는 국제적인 표준기구인 BPML, OMG, W3C, WfMC, OASIS 등에서 표준을 제정하고 있으며, 대표적인 비즈니스 프로세스 표준안으로는 BPML, BPEL, WSCI, WS-CDL, BPSS등이 있다.



BPML (비즈니스 프로세스 모델링 언어)은 Intalio가 도입하고, BPMI.org가 후원한 비즈니스 프로세스 모델링용 메타언어로 BPEL과 정면으로 경쟁하고 있다. BPML을 지원하는 벤더들이 있기는 하지만 BPEL로 인해 빛을 보지 못하고 있다.

BPMN (비즈니스 프로세스 관리 표기법)은 비즈니스 프로세스 모델링을 위한 표준 표기법(아이콘과 그래픽 세트 등) 표준인 BPMN은 비즈니스 프로세스 모델링 툴들과 BPM 제품내의 모델링 기능을 통틀어 공통의 모델링 그래픽을 사용하자는데 취지를 두고 있다. 따라서 BPMN은 BPEL 등 다른 BPM 표준들과 보완적인 관계에 있다. BPMN은 비즈니스 프로세스 모델링을 시각화하고자 하는 표준으로 크게 두 가지 목적을 가지고 있다. 첫 번째 목적은 비즈니스 프로세스 분석가로부터 개발자에 이르는 모든 사용자들이 쉽게 이해할 수 있는 표기법을 제공하고, 두 번째는, BPEL같은 XML기반의 비즈니스 프로세스 실행언어를 비즈니스 관점에서 시각화 하고자 한다.

BPEL(비즈니스 프로세스 실행 언어)은 웹서비스를 사용해, 비즈니스프로세스를 조직하는 실행언어이다. BPEL은 IBM, Microsoft 및 BEA에 의해 최초로 제안되었으며, 현재는 대다수의 BPM 벤더들에 의해 최고의 BPM 표준으로 인정받고 있다. BPEL은 향후 스케이 확장됨에 따라 더욱 광범위한 지원을 받게 될 것으로 예상된다.

Wf-XML은 OASIS ASAP(Asynchronous Service Access Protocol)을 기반으로 워크플로우 관리 연합(WfMC)에서 개발한 표준인 Wf-XML은 BPM 엔진간 상호운용성을 제공하여 다수의 엔진에 걸쳐 장기간 실행되는 비즈니스 프로세스의 실행을 가능하게 해 준다.

수년간 워크플로우 관리연합에서 활동해온 워크플로우 벤더들과 BPM 신규 벤더들 모두 Wf-XML이 BPEL에 의해 대체되어 왔고 따라서 향후 주도적인 표준으로는 부적합하다고 판단하고 있다.

XPDL은 전체 프로세스를 설명하는 비즈니스 프로세스 정의 언어로써, 모델링 툴들과 BPM 시스템들 간의 상호교환 표준으로 개발되었다. 제품 내의 프로세스 모델링, 실행 및 제어에 사용되는 BPM 컴포넌트들을 통합할 때 XPDL을 사용하는 벤더가 다수 있다.

XPDL은 현재 WfMC에서 지원하고 있으며, 몇몇 BPM 벤더들이 채택하고 있지만, 향후 BPEL로 대체될 것으로 보인다.

BPQL(비즈니스 프로세스 질의 언어)는 BPMI.org에서 표준화 작업을 하고 있다. BPQL은 프로세스의 실행을 맡는 프로세스 서버와 프로세스의 베저를 맡는 프로세스 저장소(Process Repository) 들 모두 포함하는 BPM 기반 구조에 대한 관리 인터페이스다. 이 인터페이스는 또한 SOAP에 기반을 둔 것으로 알려져 있다. 프로세스 저장소에 대한 BPQL 인터페이스는 비즈니스 분석가들이 프로세스 저장소에 의해 관리되는 프로세스 모델의 베저 및 최수를 관리할 수 있도록 해준다.



BPMLorg의 BPQL 표준화 작업은 연구가 진행된 지 여러 해가 지난 시점에서조차 여전히 공식적인 스펙 발표가 이루어지고 있으며, 드래프트 버전 또한 제시되고 있지 않은 상태이다. 따라서 어떠한 내용을 담을 것이라는 것에 대한 포괄적인 청사진만 제시되고 있을 뿐, 그 구체적인 내용과 사례를 제시하고 있지는 못하다. 현재까지 BPMLorg에서 밝히고 있는 BPQL 관련한 내용은 표준 BPM 스택 상에서 BPQL의 위치와 의미, 그리고 이러한 BPQL 인터페이스가 WSDL 웹 서비스 정의를 사용하여 제시될 것이라는 점 등이다.

### III. BPSQL(비즈니스 프로세스 시스템 질의 언어)

비즈니스 프로세스 프레임워크는 각 계층 간의 데이터 마이그레이션을 좀 더 확실하게 연구하기 위해 BPM의 6-Layer 구조로 확장하였다.

최상위계층인 6-Layer는 비즈니스 규칙, 5-Layer는 비즈니스 프로세스 모델링, 4-Layer는 서비스 계층, 3-Layer는 컴포넌트 프로세스 모델링, 2-Layer는 컴포넌트, 마지막계층인 실질적인 데이터가 존재하는 데이터베이스 즉, 데이터 모델링 계층이다.

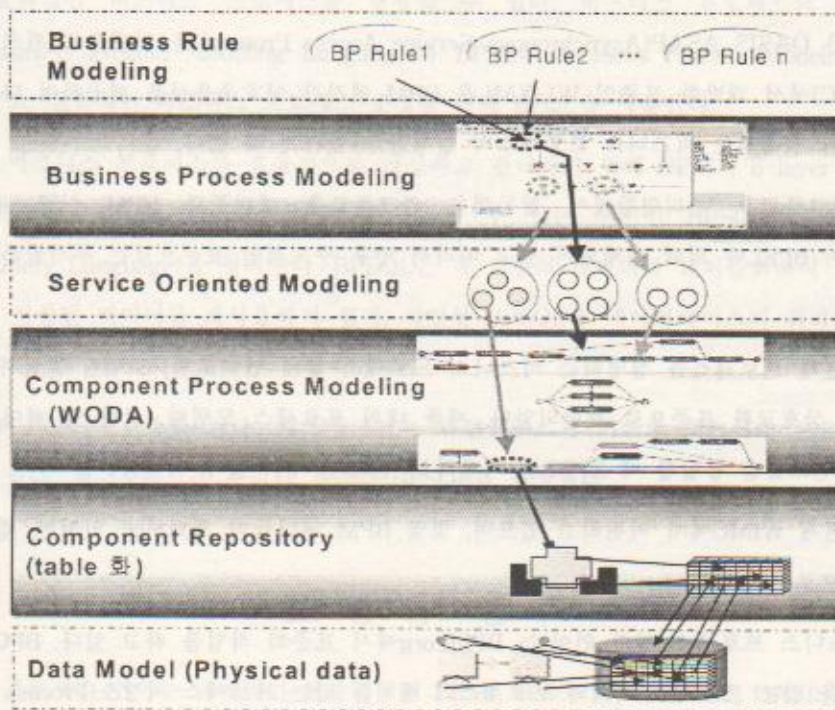


그림 2. A Closed Architecture 기반의 BPM FRAMEWORK

그림 1은 BPM과 CBD를 집속한 Closed Architecture 개념의 6-Layer 구조이다. 기존의 3-Layer에 최상위 계층에는 비즈니스 프로세스 흐름을 모델링하였고, 비즈니스 흐름에 의해 수행되는 비즈니스



스 프로세스가 존재한다. 비즈니스 프로세스는 여러 서비스들에 의해 생성되고, 서비스는 여러 개의 컴포넌트에 의해 생성된다. 컴포넌트 프로세스 모델링 계층 이하는 기존의 3-Layer와 동일한 컴포넌트 워크플로우들은 아래 계층인 컴포넌트 레퍼지토리와 매핑된다.

상위계층의 비즈니스 프로세스 계층에서는 여러 개의 비즈니스 프로세스가 존재하고, 상위계층에서 중복된 비즈니스 프로세스를 찾아 새로운 비즈니스 업무를 만들 때 비즈니스 프로세스를 제시 용하여 쉽게 만들 수 있다. 결국 상위 계층의 단위 프로세스는 중간 계층의 워크플로우로 매핑되고, 워크플로우내의 컴포넌트들은 하위 계층의 컴포넌트 레퍼지토리에 참조 또는 매핑 되어진다.

이렇듯, BPM의 6-Layer 구조는 각 계층의 바로 다음 단계와 직접 연결된 구조인 클로즈 아키텍처 이다. 우리는 BPM의 프레임워크를 관리가 쉽도록 하기 위해 유지보수관점에서 클로즈 아키텍처개념을 사용한다. 각 계층에 대한 자료를 검색하고 관리하기 위해 우리는 각 계층별 쿼리언어인 BPSQL을 정의하였다

```

BPROCESS (<process-name>)
{
  [PART-OF: {<process>}]
  [CONTAINS: {
    EVENT
    EVENTS: {<event>}
  }
  WITH-REFERENCES:
  SERVICE
  [CREATE: {<service>}]
  [USE: {<service>}]
  [UPDATE: {<service>}]
  [DELETE: {<service>}]
  POLICY: {<policy>}
}

EVENT (<event-name>)
{
  TYPE-OF: {INTERNAL|EXTERNAL|TIME}
  EVENT-SPECIFICATION
  [ COND: {<condition-predicate>}
  ACTION: {<activity>}
  ]
  WITH-REFERENCES:
  SERVICE
  [CREATE: {<service>}]
  [USE: {<service>}]
  [UPDATE: {<service>}]
  [DELETE: {<service>}]
  POLICY: {<event-policy>}
}

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

그림 2. 비즈니스 프로세스 모델링 계층의 BNF

```

COMP_PROC (<comp_proc-name>)
{
  [PART-OF: {<comp_proc>}]
  [CONTAINS: {
    EVENT
    START-EVENT: {<event>}
    EVENTS: {<event>}
    END-EVENT: {<event>}
  }
  WITH-REFERENCES:
  ENTITY
  [CREATE: {<component>}]
  [USE: {<component>}]
  [UPDATE: {<component>}]
  [DELETE: {<component>}]
  POLICY: {<policy>}
}

EVENT (<event-name>)
{
  TYPE-OF: {INTERNAL|EXTERNAL|TIME}
  EVENT-SPECIFICATION
  [ COND: {<condition-predicate>}
  ACTION: {<activity>}
  ]
  WITH-REFERENCES:
  COMPONENT
  [CREATE: {<component>}]
  [USE: {<component>}]
  [UPDATE: {<component>}]
  [DELETE: {<component>}]
  POLICY: {<event-policy>}
}

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

그림 3. 컴포넌트 프로세스 모델링 계층의 BNF

```

DATA (<DATA-name>)
{
  [PART-OF: {<data>}]
  [CONTAINS: {
    EVENT
    START-EVENT: {<event>}
    EVENTS: {<event>}
    END-EVENT: {<event>}
  }
  WITH-REFERENCES:
  DATA
  [CREATE: {<data>}]
  [USE: {<data>}]
  [UPDATE: {<data>}]
  [DELETE: {<data>}]
  POLICY: {<policy>}
}

EVENT (<event-name>)
{
  TYPE-OF: {INTERNAL|EXTERNAL|TIME}
  EVENT-SPECIFICATION
  [ COND: {<condition-predicate>}
  ACTION: {<activity>}
  ]
  WITH-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>}
  ACTIVITY-METHODS:
  {<method name>}
  IMPLEMENTED BY: {<author-name>}
  LOCATION: {full path name of the source files}
}
    
```

그림 4. 데이터 모델링 계층의 BNF



그림 2는 6-layer중에서 비즈니스 프로세스 모델링 계층의 BNF, 그림 3은 6-layer중에서 컴포넌트 프로세스 모델링 계층의 BNF, 그림 4는 데이터모델링 계층의 BNF인 쿼리 언어를 정의하였다. 컴포넌트 모델링 계층의 컴포넌트는 WODA(Workflow Oriented Domain Analysis)로 추출되었고, 우리가 정의한 컴포넌트 스펙을 이용하여 작성하였다.

#### IV. 결 론

BPM 6-layer의 비즈니스 프로세스 프레임워크는 각 계층의 바로 다음 단계의 직립 연결된 구조인 클로즈 아키텍처이다. BPM의 프레임워크를 관리가 쉽도록 하기위해 유지보수관점에서 클로즈 아키텍처개념을 사용한다. 각 계층 간의 정확한 데이터 마이그레이션을 위해 BPSQL을 정의한다. 각 계층별 쿼리언어인 BPSQL은 총 6개의 쿼리언어로 되어있다. 이 쿼리언어를 사용하여 데이터 트레이서도 가능하다. 본 논문에서는 컴포넌트 프로세스 모델링 계층의 쿼리언어만 보였다.

향후 과제는 효율성을 극대화할 수 있는 오픈 아키텍처 형태의 프레임워크에 대해 연구하고 오픈 아키텍처 상에서의 BPSQL에 대한 연구가 필요하다. 비즈니스 프레임워크의 3-Layer인 서비스를 연구하여 이 서비스를 웹 서비스와 연동하고 비즈니스 프로세스에 적용할 수 있도록 연구할 것이다.

#### < 참고 문헌 >

- 1) 김윤정,김영철, "워크플로우지향도메인분석", 한국컨텐츠학회논문지, vol. 6, no. 1, 2006. 1
- 2) Howard Smith. "Business Process Management", 시그마인사이드컴. 2004
- 3) 서운숙, "확장된 BPM과 컴포넌트 기반 방법론 매핑에 관한 연구", 2005. 2
- 4) 서운숙, 김영철, "A Closed Architecture기반의 3-Layer", 한국모바일학회, 2006.11
- 5) Matjaz Juric, "BPEL의 소개와 실습", 한국오라클, 2006. 6
- 6) 한국전산원, "비즈니스 프로세스 관리기술 표준의 BPMS적용에 관한 지원 연구", 2005. 12