

모바일 클라우드 환경에서 비즈니스 프로세스를 위한 XMDR-DAI를 이용한 소셜 기반의 협업 시스템

이종섭¹ · 문석재^{2*}

Collaborative System based on Social using XMDR-DAI for Business Process in Mobile Cloud

Jong-Sub Lee¹ · Seok-Jae Moon^{2*}

¹Department of General Education, Semyeong University, 65, Semyeong-Ro, Jecheon, Chungbuk 27-136, Korea

^{2*}Department of Computer Science, Kwangwoon University, 20, Gwangun-ro, Nowon-gu, Seoul 139-701, Korea

요 약

기업들은 클라우드 협업 환경에서의 새로운 비즈니스 모델에 대한 비용과 시간을 투자하여 비즈니스 프로세스 처리에 대한 최적화 작업을 진행하는데, 소셜 비즈니스를 기반으로 협업 기술을 적용하고 있다. 이러한 협업 비즈니스 프로세스 처리는 전통적인 BPM(Business Process Management) 방식에서 클라우드, 모바일 클라우드 환경으로 변화하기 때문이다. 이에 본 논문에서는 모바일 클라우드 환경에서 비즈니스 프로세스 관리를 위한 소셜 기반의 협업 시스템을 제안한다. 본 시스템은 모바일 클라우드 기반의 협업 업무를 처리하기 때문에 데이터 상호운용에 적합한 XMDR-DAI를 적용하였다. 이 XMDR-DAI는 클라우드 환경에서 독립적으로 운영되는 로컬 시스템간의 데이터 공유 및 교환 서비스를 제공하는 것으로, 소셜 기반의 협업 비즈니스 프로세스 운영을 담당한다. 소셜 기반의 협업 비즈니스 프로세스 처리는 내부 쿼리로 인한 스키마 매핑상의 구조, 의미 충돌, 그리고 데이터간의 단위 변환등과 같은 충돌이 발생하게 되는데, 각 매핑 과정에서 일어나는 충돌문제를 XMDR-DAI로 해결하였다.

ABSTRACT

In this paper, we propose a social-based collaboration systems for business process management in the mobile cloud. This XMDR-DAI is to provide services for data sharing and exchange between local systems that operate independently in a cloud environment, take charge of a social-based collaborative business process management. Social-based collaborative business processes are handled in a collision among a structure such as unit conversion, meaning conflict, and the schema mapping data resulting from the inner query. The conflict was resolved by mapping process which takes in each XMDR-DAI.

키워드 : 모바일 클라우드, XMDR, 협업 프로세스, 메타데이터 스키마

Key word : Mobile Cloud, XMDR, Collaborative Businss, Metadata Schema

Received 26 August 2015, Revised 25 September 2015, Accepted 05 October 2015

* Corresponding Author Seok-Jae Moon(E-mail:msj8086@kw.ac.kr, Tel+82-02-940-5288)

Department of Computer Science, Kwangwoon University, #20 Gwangun-ro Nowon-gu, Seoul 139-701, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2015.19.10.2331>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

최근 기업들은 새로운 비즈니스 모델의 비용과 시간을 투자하여 변화를 모색하고 있다. 그에 대한 방법으로 클라우드, 모바일 컴퓨팅, 그리고 소셜 비즈니스를 기반으로 협업 기술을 적용하여 빠르게 변화하고 있다 [1]. 일반적으로 기업 내에 업무 처리 최적화를 위한 협업 환경은 BPM(Business Process Management)이다 [2,3]. BPM은 업무 담당자들 간의 원활한 업무 협업을 지원하는 것으로, 사용자 인터페이스를 기반으로 서비스를 한다. 이러한 BPM 기반으로 운영되는 시스템들은 Sharing, Expert, Mobile, Governance 등이 제공되어야 한다. 다시 말해, 협업 프로세스는 상호간의 공유를 위한 환경을 지원해야 하고, 협업 활성화를 위한 전문가를 지원할 수 있는 시스템이 요구된다. 그리고 언제 어디서나 업무 협업을 할 수 있도록 모바일 서비스가 지원되어야 하며, 협업 및 변경 정책 관리도 되어야 한다 [4,5]. 기존 BPM들은 접근 권한에 따른 프로세스 관리, 업무 실시간 처리 현황 제공, 업무 관련자들 간의 커뮤니케이션, 해당 업무에 대한 다양한 의견 교환, 포스트된 의견이 업무로 연결 등이 주기능이다. 이러한 기능들이 최근에는 소셜 네트워크를 이용한 비즈니스 업무 협업 처리 방식으로 변화하고 있다[6].

본 논문에서는 모바일 클라우드 환경에서 비즈니스 프로세스 관리를 위한 소셜 기반의 협업 시스템을 제안한다. 제안 시스템은 기존의 BPM 기반의 협업 업무 처리 방식을 소셜 협업을 통한 업무 처리 방식으로 변환한 것이다. 그리고 본 제안 시스템은 XMDR-DAI[7,8]를 이용한 소셜 비즈니스 프로세스 처리 방식을 적용한다. 이 XMDR-DAI는 클라우드 환경에서 독립적으로 운영되는 레거시 시스템간의 데이터 공유 및 교환을 비즈니스 프로세스를 통한 상호운용 서비스를 제공하는 역할을 한다. 기존의 비즈니스 프로세스들은 처리과정에서 발생하는 내부 쿼리로 인한 스키마와 데이터 충돌이 발생했던 문제점들이 있었고, 이러한 문제점들은 본 논문에서는 XMDR-DAI로 해결하려고 한다. 또한 비즈니스 프로세스 내부 질의 대한 처리는 본 논문에서 제시한 GQP, LQP 기법을 적용한다[7]. GQP(Global Query Process)는 비즈니스 프로세스 로직 안에 글로벌 스키마 기반으로 작성된 GQ(Global Query)가 포함된 형태로써, GQ는 DML(Data Manipulation Language)로

구성되며, 생성 규칙에 의해서 정의되고, ANSI 표준에서 정의된 규칙을 따른다. 그리고 LQP(Local Query Process)는 비즈니스 프로세스 로직 안에 GQ가 변환 규칙에 의해서 로컬 스키마 기반으로 GQP가 재정의된 LQ(Local Query)가 포함된 쿼리 형태이다. 이 LQ 또한 DML(Data Manipulation Language)로 구성된 것이며, 생성 규칙에 의해서 정의되고, ANSI 표준에서 정의된 규칙을 따른다. 그리고 LQP 역시 GQP에서 정의한 GQ 생성 규칙 1,2,3이 적용되었다.

본 논문에서 제안한 시스템은 다음과 같은 서비스들을 지원할 수 있다. 첫째, 비즈니스 프로세스의 가시화이다. 이는 모바일 지원을 통해 협업 업무 처리에 쉽게 접근하고 광범위하게 활용 가능해지며, 프로세스 확인이 가능하여 비전문가도 쉽고 빠르게 프로세스 처리 및 변경이 가능해진다. 둘째, 대시보드를 통한 프로세스 처리 실시간 현황 파악이 가능해진다. 즉, 프로세스 가시화를 위한 모델링 환경을 제공함으로써 프로세스 및 비즈니스 정책, 관련 문서의 연계 등이 가능해진다. 셋째, 소셜 환경을 통한 협업 프로세스 처리를 커뮤니티에서 검색하여 전문가에게 요청할 수 있게 된다. 현재 실시간으로 처리중인 업무에 대한 상세 정보를 제공하게 되어 해당 업무에 대한 상세 이력 및 코멘트 내역을 확인하여 처리할 수 있게 한다.

II. 관련 연구

2.1. BPM 모바일

Forrester에 의하면, 기업들이 비즈니스 프로세스 관리시스템(Business Process Management System)을 모바일 환경에 적합하도록 개편하는데 수조원에 달하는 투자가 이뤄질 것으로 전망하고 있다[9]. Mobile BPM 솔루션 벤더로는 한국의 HandySoft[10]를 위시하여, Appian[11], Pegasystems[12], BizAgi[13], TIBCO[14], PNMSOFT[15], OpenText[16] 등이 있다. Social BPM은 미리 설계된 프로세스 모델 기반의 정형적인 프로세스와 더불어, 실시간 자발적으로 형성되는 직원들 간 비정형 협업 활동이나 외부 고객과의 소셜 협업을 지원함으로써 기업의 경영성과를 개선시킨다[6]. 예컨대, 영업 직원이 영업프로세스 수행 중에, 고객이 트위터에서 회사 웹사이트로 접속하여 제품설명서를 다운로드하는

소셜액티비티를 실시간 통보받아 대응할 수 있다.

Appian, Pegasystems 등 대부분의 BPM Suite 제품들이 이미 소셜네트워크의 통합을 통해 실시간 프로세스 협업을 지원하고 있다. 한국의 uEngine사도 Process Codi라는 Social BPM 제품을 공개소프트웨어로 제공하고 있다. BPM에 BI(Buisness Intelligence)를 결합한 Intelligent BPM[17]은 기업 내 부서별로 산발적인 요구에 의해 구축되던 BI Application을 정형화된 Operational Process에 Embedded시킴으로써, 프로세스 수행을 위한 일상 업무에서 필요한 Analytics를 보편적으로 수행하게 하고, BI 활용의 확산을 촉진한다. Intelligent BPM은 프로세스 성과 모니터링 및 보고, 경영성과 예측, 경영활동 계획, 시나리오 모델링, 복잡한 의사결정, 실시간 상황인지, 실시간 Action 추천 등 보다 지능적인 프로세스 관리를 가능하게 한다.

BPM PaaS(Platform as a Service)[18]로도 불리는 Cloud BPM은 BPM Suite을 PaaS 서비스로 제공하고 BPM 관련 데이터도 IaaS에 저장함으로써, 기업에서 BPM Suite에 대한 투자없이 서비스 가입을 통해 BPMS를 활용할 수 있게 한다. Appian, Cordys, IBM 등이 BPM PaaS를 제공하고 있다. Cloud BPM은 아직 데이터 보안 및 레거시시스템과의 통합 문제 등의 우려로 급속히 확산되고 있지는 않다.

2.2. XMDR-DAI

XMDR-DAI(eXtended Metadata Registry-Data Access & Integration)[7]는 MDR보다 더 확장된 개념인 XMDR을 이용한 데이터 통합에 따른 메타데이터 스키마, 데이터 구조 및 의미 충돌을 해결하기 위한 에이전트로써, 관계형 데이터베이스 메타데이터를 객체지향 데이터베이스에 저장하는 기술이다. 다시 말해, 분산된 데이터들의 스키마 구조와 데이터 간의 충돌을 해결하고자 MDR과 온톨로지를 결합하여 데이터를 통합하는 저장소이다. XMDR-DAI이 적용된 협업 시스템은 단일 뷰어 형태의 어플리케이션을 이용하여 로컬 데이터베이스간 상호운용에 필요한 데이터 통합 서비스를 제공한다. 그리고 메타데이터 레지스트리와 온톨로지 시소러스 개념을 이용한 데이터간 연관 정보를 제공하여 데이터 무결성, 확장 검색 서비스를 제공한다. 또한 로컬 데이터베이스간 개체 스키마를 이용한 데이터 공유 및 교환 과정에서 글로벌 스키마와 로컬 스키마간 매핑-변

환 할 수 있는 기능과 협업에서의 쿼리 기반의 워크플로우를 이용하여 데이터 상호운용할 때 쿼리 구성에 필요한 기준을 제시하고, 변환 메커니즘 및 분산 쿼리처리 모듈을 제공하여 데이터 이주 상황에 발생하는 복제, 이동, 그리고 분할 기능을 제공한다.

2.3. GQP, LQP 생성 규칙

GQP(Global Query Process)는 본 제안 시스템에서 프로세스 상호운용시 발생하는 여러 가지 메타데이터 기반의 쿼리 충돌문제를 해결하기 위한 방법론으로 [표 1, 2]에서 정의한 생성 Rule을 적용한 것으로, 메타 글로벌 스키마 개체에 정의된 속성을 바탕으로 만들어지는 쿼리 프로세스이다[7]. GQP 생성 Rule은 선택(SELECT), 삽입(INSERT), 삭제(UPDATE)로 구분되며, ANSI 표준에 정의한 명세를 따른다.

표 1. GQP 문법

Table. 1 GQP(Global Query in Process) grammar

GQP TYPE	PHASE TYPE
SELECT TYPE	SELECT phase {GS_Field_List} FROM phase {GS_Table_Name} WHERE {GS_Field_List(op)Value_List}
INSERT TYPE	INSERT phase[INTO]GS_Table_Name (GS_Field_list){[VALUES]phase {GS_Field_List(op)Value_List}
UPDATE TYPE	UPDATE phase {GS_Table_Name} SET phase {GS_Field_List=Value_List} WHERE phase {GS_Field_List(op) Values_List

표 2. GQP 생성 규칙

Table. 2 GQP General Rule

TYPE	RULE FORMAT
Rule 1	GS_Field_List = { x x ∈GSR(GS _{field} values) }
Rule 2	GS_Tield_List = { x x ∈GSR(GS _{table} values) }
Rule 3	GS_Vield_List = {user input}

GQP 생성 시 Rule 1은 프로세스 쿼리에서 필드 요소 GS_Field_List, Rule 2는 테이블 요소, Rule 3은 쿼리 필드 조건들에 필요한 인스턴스 값들을 나타낸 것이다. 그리고, GQP 생성 규칙이 적용된 쿼리들은 FROM 절이 있는 경우와 없는 경우로 나눌 수 있다. 쿼리 내부에

FROM절이 포함되어 있는 경우는 비즈니스 프로세스가 XMDR 기반으로 BPR(Business Process Repository)에 등록되어 있어, 프로세스 내부의 쿼리가 협업에 의해서 미리 정의된 경우이다. 따라서, 협업에 의해서 발생하는 데이터 이주 상의 필요한 DB테이블이 Proxy DB에 미리 만들어져 있어서 FROM절이 GQBP에 포함되어 있다. 그리고 FROM절이 없는 경우는 비즈니스 프로세스의 내부 쿼리가 사용자에게 의해서 실시간으로 정의되어 수행되는 경우이다. 이는 내부 쿼리가 미리 정의되어 있지 않기 때문에 협업에 의해서 데이터 이주 상의 필요한 DB테이블을 쿼리가 수행되면서 실시간으로 Temporarily 테이블을 Proxy DB에 생성하여 FROM절이 GQP에 포함되어 있지 않는다.

III. 제안 시스템

3.1. 시스템 개요

본 절에서는 클라우드 환경에서 협업 프로세스를 실시간으로 쉽게 빠르게 처리 변경이 가능하고, 관련 문서의 연계 등 프로세스 가시화를 위한 모델링 환경을 제공하는 소셜 기반의 협업 시스템을 제안한다. [그림 1]은 본 논문에서 제안한 시스템 구성 개요로서, Social Applications, Collaboration System, Process Resources, Legacy Systems로 구성된다. 그리고 Collaboration System은 Process Provider, Social Provider, XMDR-DAI Provider로 구성 된다. 다음은 각 계층의 구성요소에 대한 설명이다.

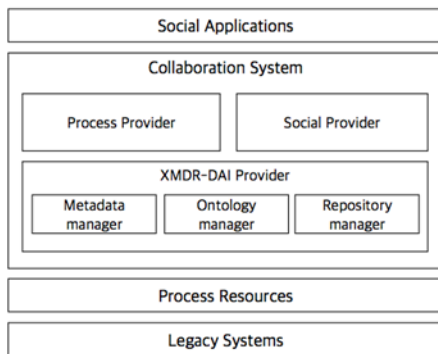


그림 1. 제안 시스템 구성
Fig. 1 Proposal system composition

● Collaboration System: 이 계층은 본 시스템의 핵심 부분으로, XMDR-DAI를 중심으로 클라우드 상의 분산되어 있는 비즈니스 프로세스들을 연계하여 매칭하고, 협업 프로세스 처리에서 발생하는 내부 쿼리상의 충돌 문제들에 대한 것을 해결해 준다. 또한 이 계층은 Process Provider, Social Provider, XMDR-DAI Provider로 구성된다.

■ Process Provider: Process Provider는 사용자가 프로세스 리스트를 요청할 때 프로세스 저장소로부터 정보를 제공하는 역할을 한다. 그리고 요청된 프로세스가 사용자의 권한, 또는 프로세스의 Rules 정보에 대한 확인 처리도 한다.

■ Social Provider: Social Provider는 사용자가 협업이 필요한 프로세스를 선정하여 프로세스를 처리할 수 있는 전문가를 찾고자 할 때 소셜 환경을 통해 전문가 리스트 정보를 제공해주는 역할을 한다.

■ XMDR-DAI Provider: XMDR-DAI Provider는 클라우드 상의 레거시들 간의 프로세스 협업 처리를 위해 상호운용(공유 및 교환)할 때, 발생하는 스키마 또는 데이터 충돌을 매핑 해결해주는 역할을 한다. 다시 이 계층은 Metadata Manager, Ontology Manager, Repository Manager로 구성된다.

- Metadata Manager: Metadata Manager는 협업 프로세스에서 필요한 스키마들의 의미적인 문제나 표현적인 충돌문제를 표준 스키마로 정의하여 매핑 척도가 되는 정보를 관리한다.

- Ontology Manager: Ontology Manager는 표준 스키마로 정의된 매핑과정의 로드맵 정보를 관리한다.

- Metadata Repository Manager: Metadata Repository Manager는 클라우드 상의 레거시들 간의 프로세스 협업에 사용되는 모든 메타데이터 스키마를 저장하고 관리한다.

● Process Resources: 이 계층은 각 레거시 시스템들마다 비즈니스 협업에 처리되는 프로세스를 모아놓은 곳이다.

● Legacy Systems: 이 계층은 클라우드 환경에 포함된 레거시 시스템들이다.

3.2. 시스템 동작

[그림 2]는 사용자가 처리해야할 비즈니스 프로세스에 대한 정보와 전문가 리스트 정보를 요청하는 과정을

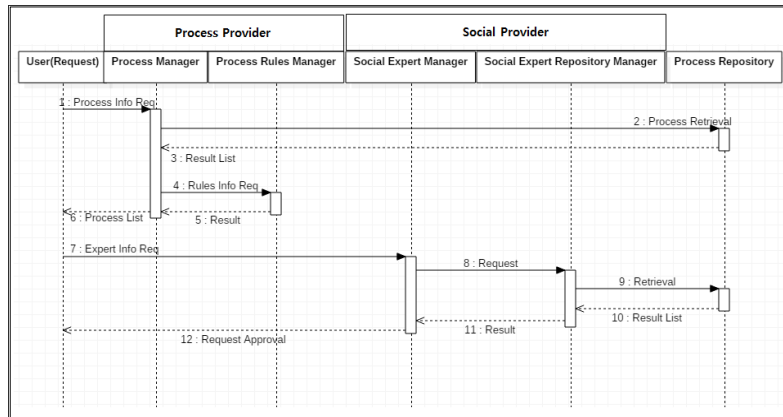


그림 2. 시스템 시퀀스 1
Fig. 2 System sequence 1

나타낸 것이다. [그림 2]의 User(Request)에서 Process Manager에게 프로세스 정보를 요청한다(1). Process Manager는 Process Repository를 통해 프로세스 정보를 조회하게 된다(2,3). 검색된 프로세스 정보들은 Process Manager가 Process Rules Manager에게 프로세스 적용 규칙 정보를 조회 요청한다(4). 이 조회 요청되는 규칙은 프로세스의 메타정보와 권한에 대한 정보이다. 1,2,3,4 처리 과정을 통해 얻어온 프로세스 리스트와 프로세스 규칙을 User(Request)에게 보여주게 된다.

이후, User(Request)는 프로세스 리스트를 확인한 후에 협업 요청이 필요한 프로세스에 대한 처리를 위해 전문가 정보에 대한 리스트 요청을 한다(5). 요청을 받은 Social Expert Manager는 Social Expert Manager를 통해 전문가 정보를 조회하게 된다(6,7,8,9). 다시 말해, 협업 요청해야 될 프로세스에 적합한 전문가 조회하는 것이다. Social Expert Manager에서는 조회된 전문가에게 프로세스 처리 승인을 요청하게 된다(10). 이후, 승인 요청 받은 사용자(전문가)는 요청된 프로세스 내역을 보고 승인 여부를 판단한다(11). [그림 4]는 [그림 2]에서 사용자(전문가)가 요청된 프로세스 내역을 보고, 승인 후에 프로세스가 위임되어 처리하는 과정이다. [그림 4]에서 User(Expert.)는 요청된 협업 프로세스 리스트를 Process Provider를 통해 전달받는다(1). 이렇게 전달 받은 사용자(전문가)는 프로세스 리스트를 기반으로 프로세스 처리에 적절한 메타데이터 스키마 정보를 Metadata Manager에게 요청한다(2). 시스템 시퀀스 [그림 3]과 같은 코드로 구현된다.

요청을 받은 Metadata Manager는 Ontology Manager에게 메타데이터 스키마 상에서 함께 처리해야 될 연관 스키마 정보를 요청하고, Metadata Repository Manager에게 스키마 프로파일 정보를 요청한다(3,4,5,6,7). 이후, 사용자(전문가)는 스키마 프로파일 정보와 연관 스키마 정보를 전달 받은 후, 프로세스 정의하여 각 협업 레거시 시스템으로 처리하게 된다(7,8).

```

Algorithms: System execute flow(expert request);
public class SysExecute extends Thread {
    private SocialExpertMger sem; private SocialExpertRepositoryMger serm;
    private ProcRepository pr; private ProcRules prs;

    public void Agent_Execute(String s) {
        ArrayList<proc> procList = new ArrayList<proc>;
        Map<expert> expertInfo = new HashMap<expert>;
        procList[] = procProvider();
        expertInfo[] = socialProvider(procList);

        public void run() { ...
            pr = new ProcRepository();
            prs = new ProcRules();
            sem = new SocialExpertMger();
            serm = new SocialExpertRepositoryMger();
            ...
        }

        public expertList[] socialProvider( procList pl ) { ...
            try {
                expertRep[] = sem.reqInfo(pl);
                expertList[] = sem.reqInfo(expertRep);
            } catch(Exception e) { e.printStackTrace(); }
            return expertList;
        }

        public procList[] procProvider() { ...
            try {
                ResultSet = pr.retrievalProc();
                procList[] = prs.reqInfo( ResultSet rs );
            } catch(Exception e) { e.printStackTrace(); }
            ...
            return procList[];
        }
    }
}
    
```

그림 3. 시스템 시퀀스 1 알고리즘
Fig. 3 System sequence 1 algorithms.

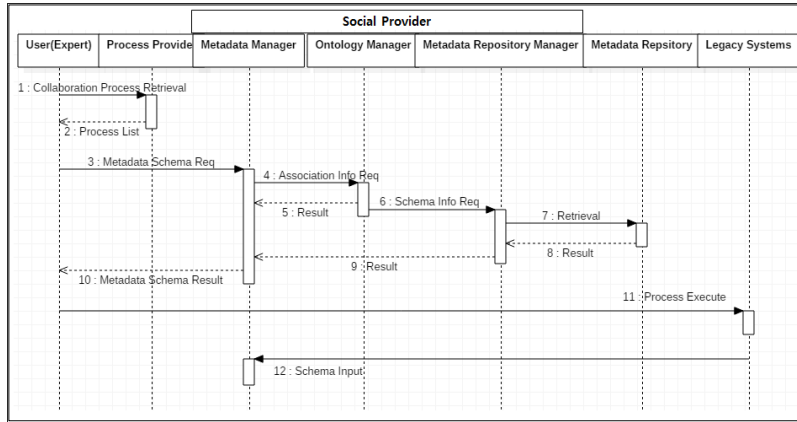


그림 4. 시스템 시퀀스 2
Fig. 4 System sequence 2

```

Algorithms: System execute flow(process control);
public class SysExecute extends Thread {
    private MetadataMger mdm; private OntologyMger onm;
    private MetadataRepositoryMger mrm; private CollaborationLegacySys;

    public void Agent_Executer(String s) {
        ArrayList<procList> procInquiry = new ArrayList<procList>;
        Map<schema> = new HashMap<schema>;
        procInquiry[] = procProvider();
        schemaInfo[] = retrievalSchema(procInquiry);

        cls.procExecute(new legacyQuery(schemaInfo));
    }

    public void run() { ...
        mdm = new MetadataMger();
        onm = new OntologyMger();
        mrm = new MetadataRepositoryMger();
        cls = new CollaborationLegacySys();
        ...
    }

    public schemaInfo[] retrievalSchema(procInquiry pi) { ...
        try {
            associationInfo[] = onm.reqInfo(pi);
            schemaInfo[] = mrm.reqInfo(new MetadataRepository(), associationInfo);
        } catch (Exception e) { e.printStackTrace(); }
        ...
        return schemaInfo;
    }

    public procList[] procProvider() { ...
        try {
            ResultSet = pr.retrievalProc(); //프로세스 리포지토리에서 데이터 받음
            procList[] = pra.reqInfo(ResultSet rs);
        } catch (Exception e) { e.printStackTrace(); }
        ...
        return procList;
    }
}
    
```

그림 5. 시스템 시퀀스 2 알고리즘
Fig. 5 System sequence 2 algorithms

IV. XMDR-DAI 메커니즘

4.1. XMDR-DAI 구성

본 절에서는 협업 프로세스 처리 방식으로 XMDR-DAI를 적용한다. 레거시 시스템 간의 프로세스 상호 운용에서 발생하는 충돌을 해결한 것으로 협업 기반의 유형자산관리 사례를 적용한 [그림 6]이다. [그림 6]는

MSO(Metadata Semantic Ontology), InSO(Instance Semantic Ontology)로 구성된 온톨로지 시소러스 영역과 메타 글로벌 스키마와 메타 로컬 스키마로 구성된다. 메타 글로벌 스키마(글로벌 스키마)는 협업에 필요한 테이블 구조를 나타낸 것으로 각 테이블명(APPLY, ITEMS)과 필드로 구성된다. 메타 로컬 스키마(로컬 스키마)는 본 적용 시스템에 참여된 레거시 데이터베이스 안에 테이블들과 필드의 구조를 나타낸 것이다. 예컨대, 글로벌 스키마에서 APPLY 테이블의 필드명 name과 매칭되는 로컬 스키마에서 정보를 찾기 위해, MSO를 통해 검색한다. MSO 맵에서는 name 노드가 Item-name, Asset-name으로 구분되는 것을 알 수 있고, 이중에 Item-name은 로컬 스키마의 APPL 테이블에서 Item_Name임을 알 수 있게 된다.

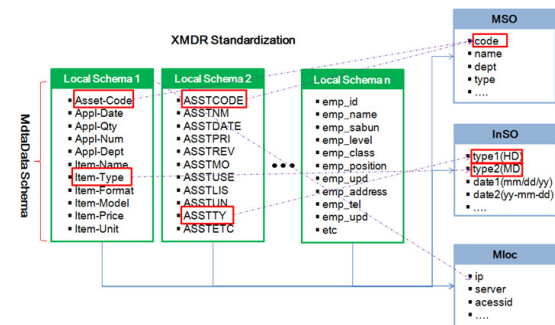


그림 6. 메타 정보 충돌에 대한 XMDR-DAI 구성 설계
Fig. 6 XMDR-DAI about meta information collision

이러한 매칭 방법을 통해 글로벌 스키마와 로컬 스키마간의 의미적으로 같은 것을 찾아 변환할 수 있다. InSO 맵에서는 데이터 단위와 표현에서 충돌되는 문제를 해결한다. 예를 들어, 글로벌 스키마에서 ITEMS에서 Type은 varchar 형식으로 InSO의 type 노드가 HD(mm), MD(mm)으로 구분되는 것을 알 수 있고, 이는 로컬 스키마의 ASSET 테이블에서 AssetType는 MD(mm)으로 매칭되는 것을 알 수 있다. 이런 매칭 방법을 통해 글로벌 스키마와 로컬 스키마간의 데이터 표현에서 코드, 단위, 일련번호 형식 등을 변환할 수 있게 된다.

4.2. 쿼리 파싱

본 논문에서 트리 그래프 기반으로 표현된 쿼리는 실제 로컬 데이터베이스에서 수행되는 쿼리로 변환해야 된다. [그림7](a)는 Single-Node Physical Plan을 이용한 파싱 기법이고, 제시된 쿼리를 예제로 파싱 방법을 설명한다. 쿼리 파싱은 [그림8]에서처럼 select, operation call, join, scan로 구분하여 처리한다. select는 쿼리문에서 필드절로 p.codeID, p.pirce, t.assetnm이다. operation call은 select에서 조인된 필드를 구별하는 것으로 t in apply가 p in asset로 조인되어 있는 것을 알 수 있고, 그 조인된 필드는 t.assetnm이다.

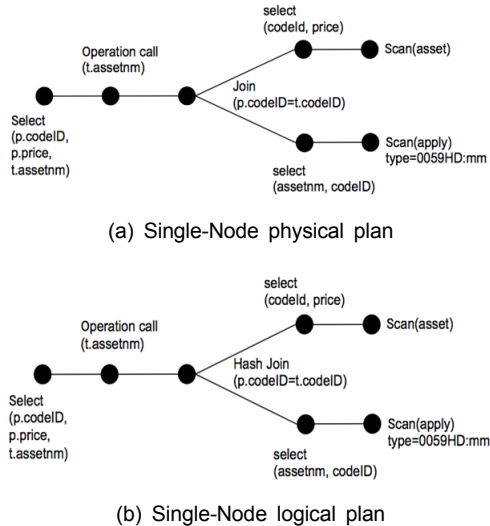


그림 7. 트리 그래프를 이용한 쿼리 분석1
Fig. 7 Query analysis plan using tree graph1

그리고 join은 두 개의 테이블이 조인키 필드인 p.codeID=t.codeID이다. 이 join을 기준으로 왼쪽 트리와 오른쪽 트리로 구성하여 select를 구별할 수 있게 된다. 즉, 왼쪽 select는 codeID, price이고 오른쪽 select는 assetnm, codeID이다. 이렇게 분석된 것을 왼쪽 트리부터 생성하고 스캔하여 로컬 쿼리로 변환된다. 그런 이후, [그림7]의 (b)처럼 Single-Node Logical Plan로 다시 표현하고, 실제 쿼리 파싱하여 변환하기 위해서 hash_join인 해시 테이블 기법을 이용하여 계획하여 분석한다. 이처럼 [그림7]과 같이 계획된 쿼리는 그림 [그림8]과 같이 parallel 방법으로 변환을 하게 된다. 이 방법은 exchange 연산자를 기준으로 하여 쿼리를 매칭 변환하게 되는데, heuristic 알고리즘 기법으로 왼쪽 노드부터 트리 순위 규칙에 따라 매핑 변환이 이루어진다.

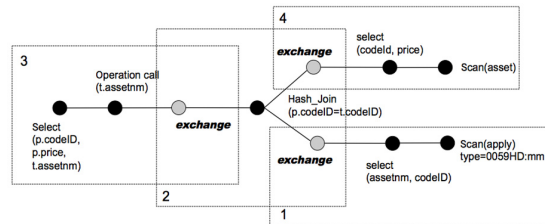


그림 8. 트리 그래프를 이용한 쿼리 분석2
Fig. 8 Query analysis plan using tree graph2

V. 적용 사례

5.1. 적용 사례

본 논문에서는 유형자산관리 업무를 적용하여 레거시 시스템간의 소셜 기반의 비즈니스 프로세스를 위한 협업 업무를 적용하였다. 이 적용 사례는 모바일 기반으로 자산신청-구매결의등록-발주서조회 그리고 대여 처리에 대한 소셜 비즈니스를 기반으로 협업 지원을 요청하는 과정을 보인다. [그림9]에서 신청자가 Monitor를 구매하기 위해 “Do the Process” 업무를 수행하는 중이다. (1)은 업무를 처리 하기 위해서 ID를 입력한다. (2)에서 전문가 리스트 확인한다. (3) DavidVan에게 업무 협업 요청을 한다. [그림10]에서 (1)은 DavidVan은 신청자로부터 업무 협업 요청 메시지를 확인한다. (2)는 업무 협업 요청의 기한이 남은 리스트이다.

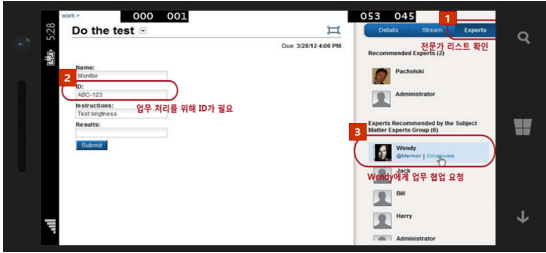


그림 9. 모바일 적용 사례1
Fig. 9 Mobile Apply Example1

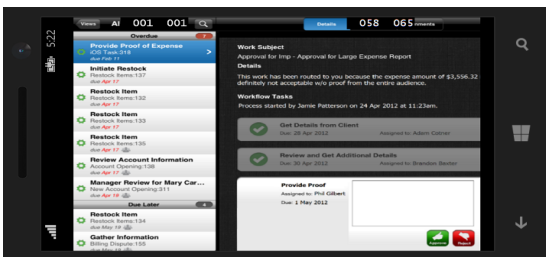


그림 10. 모바일 적용 사례2
Fig. 10 Mobile Apply Example2

5.2. 성능분석

XMDR-DAI를 적용한 협업 시스템에서 소셜 기반의 비즈니스 프로세스 수행 과정상의 데이터 접근 서비스와 수집 서비스 수행 결과를 바탕으로 성능을 평가한다. 시스템에서 비즈니스 프로세스의 절차 단계에 따라 실행되는 과정에서 프로세스가 정방향 처리와 역방향 처리가 있는데, 역방향 처리는 분석 대상에서 제외한다. 또한, XMDR-DAI에서의 온톨로지 시소러스 정보에 의한 연관 정보에 필터링은 쿼리에서 의미 연관성 조건이 20개를 넘지 않는다.

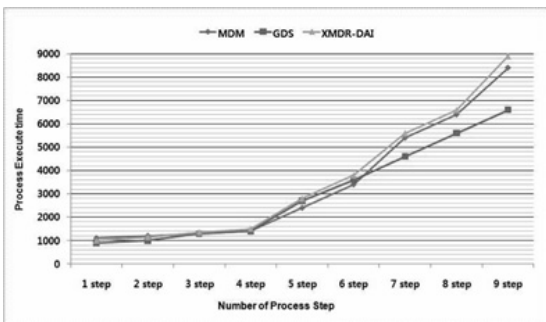


그림 11. 프로세스 내부 수행 절차 수에 대한 수행 시간
Fig. 11 Execution time for the process

본 제안 시스템에 적용된 유형자산통합 시스템에서 소셜 기반의 비즈니스 프로세스 내부의 쿼리 처리에 따른 데이터 접근 서비스와 수집 서비스를 수행할 때 프로세스 내부 절차 수에 따른 처리 시간을 [그림11]과 같이 성능을 평가한다. 성능 평가를 위한 테스트는 협업된 4개의 로컬 데이터베이스 서버를 이용하였고, 약 15,000건 이상의 수집 결과를 가지는 비즈니스 프로세스를 사용하였다. 또한 한 프로세스 내부 수행 절차 수는 최대 9단계를 넘지 않는다. [그림11]에서와 같이 MDM(Master Data Management), GDS(Grid Data Service) 방식 그리고 본 XMDR-DAI 방식에 따라 비즈니스 프로세스를 수행했을 때 각 수행 절차 수에 따른 성능을 비교한 것이다. 한 비즈니스 프로세스의 내부 절차가 1단계에서 4단계까지는 세 방식 모두 1100ms에서 1500ms 정도의 처리 시간이 나오지만, 5단계 이상부터는 2400ms에서 660ms 정도의 처리 시간으로 급격히 크게 증가하였다. 하지만 MDM과 XMDR-DAI 방식은 GDS방식보다 7단계에서부터 약 1000ms이상 차이가 났다. 이와 같은 처리 시간에 대한 결과에서 GDS는 메타데이터 스키마 기반의 쿼리 방식이고, MDM과 XMDR-DAI 방식은 마스터 데이터 기반의 쿼리 처리 방식이기 때문에 약간의 차이가 난다. 하지만 3가지 방식들은 4단계에서부터 처리 시간이 증가하는 것으로 보아 비즈니스 프로세스 내부 절차의 수는 4단계 수준이 효율적이다. 그렇기 때문에 제안된 XMDR-DAI 기반의 시스템도 협업에 의한 비즈니스 프로세스를 정의할 때에는 이를 고려해야 한다. 프로세스 절차의 수가 증가할 수록 협업 자체가 복잡해지므로 제안된 협업 시스템도 4단계 수준정도의 프로세스를 수행해야 되어야함을 살펴볼 수 있다.

VI. 결론

본 논문에서는 클라우드 환경에서 비즈니스 프로세스 관리를 위한 소셜 기반의 협업 시스템을 제안하였다. 제안 시스템은 소셜 기반으로 협업 업무 처리 방식으로 변환한 것으로 XMDR-DAI를 이용하여 소셜 비즈니스 프로세스 적용하였다. XMDR-DAI는 클라우드 환경에서 운영되는 레거시 시스템간의 데이터 상호운용 서비스를 제공한다. 일반적으로 데이터 상호운용 서비스는

협업 비즈니스 프로세스를 처리과정에서 쿼리 충돌과 데이터 충돌에 대한 문제점들이 있었다. 이러한 문제점들을 본 시스템은 XMDR-DAI로 해결하였고, 소셜 기반의 비즈니스 프로세스 질의 처리를 GQP, LQP 기법으로 적용하여 통합된 형태의 운영 서비스를 제공하게 된다. 그리고 사용자가 협업을 요청 및 처리하는 과정을 소셜 네트워크를 통한 통일된 뷰어를 제공함에 따라 비즈니스 협업 처리가 보다 효율적으로 서비스 된다.

본 논문에서 제안한 시스템에서 제공하는 서비스는 다음과 같다. 첫째, 비즈니스 프로세스의 가시화이다. 모바일 지원을 통해 협업 업무 처리에 쉽게 접근하고 광범위하게 활용 가능해지며, 프로세스 확인이 가능하여 비전문가도 쉽고 빠르게 프로세스 처리 및 변경이 가능해진다. 둘째, 대시보드를 통한 프로세스 처리 실시간 현황 파악이 가능하다. 프로세스 가시화를 위한 모델링 환경을 제공함으로써 프로세스 및 비즈니스 정책, 관련 문서의 연계 등이 가능해진다. 셋째, 소셜 환경을 통한 협업 프로세스 처리를 커뮤니티에서 검색하여 전문가에게 요청할 수 있다. 실시간으로 처리중인 업무에 대한 상세 정보를 제공하여, 해당 업무에 대한 상세 이력 및 Posing, Comment 내역을 확인하여 처리할 수 있게 한다. 향후 연구 과제로는 소셜 기반의 협업 비즈니스 프로세스 처리에 대한 데이터 사일로 정책이 빅데이터 기반으로 해결되어야 한다. 추후 빅데이터 기반인 프레임워크를 제시하고, 데이터 처리 과정을 noSQL 방식으로 변환 처리 되는 방법이 제시되어야 한다.

ACKNOWLEDGMENTS

This work reported in this paper was conducted during the sabbatical year of Semyeong University in 2015.

REFERENCES

[1] Weinhardt, Christof, et al. "Cloud computing - a classification, business models, and research directions." *Business & Information Systems Engineering* 1.5 (2009): 391-399.

[2] Weske, Mathias. *Business process management: concepts, languages, architectures*. Springer Science & Business Media, 2012.

[3] Jeston, John, and Johan Nelis. *Business process management*. Routledge, 2014.

[4] Lohmann, Patrick, and Michael Zur Muehlen. "Exploring the Implications of Top Management Team Characteristics on Business Process Innovativeness." Available at SSRN 2554118 (2015).

[5] Rosemann, Michael, and Jan vom Brocke. "The six core elements of business process management." *Handbook on Business Process Management* 1. Springer Berlin Heidelberg, 2015. 105-122.

[6] Kemsley, Sandy. "Business process management and the social enterprise." *Handbook on Business Process Management* 1. Springer Berlin Heidelberg, 2015. 463-474.

[7] Moon, SeokJae, GyeDong Jung, and YoungKeun Choi. "XMDR-DAI based on GQBP and LQBP for business process." *Advanced Computer Science and Information Technology*. Springer Berlin Heidelberg, 2010. 72-85.

[8] Moon, SeokJae, GyeDong Jung, and YoungKeun Choi. "A Study on Cooperation System design for Business Process based on XMDR in Grid." *International Journal of Grid and Distributed Computing* 3.3 (2010).

[9] Anisiewicz, Jędrzej, et al. "Configuration of Complex Interactive Environments." *New Research in Multimedia and Internet Systems*. Springer International Publishing, 2015. 239-249.

[10] HandySoft: <http://www.handysoft.com/>

[11] Appian: <http://www.appian.com/>

[12] Pegasystems: <http://www.pega.com/products/pega-7#tab-mobile>

[13] BizAgi: <http://www.bizagi.com/>

[14] TIBCO: www.tibco.com

[15] PNMSOft: <http://www.pnmssoft.com/>

[16] OpenText: <http://www.opentext.com/>

[17] I. Shapiro R, Khoshafian S (2013) iBPMS intelligent BPM systems: impact and opportunity. Future Strategies Inc, Canada 2. Öunapuu, Enn. "Teaching and Promoting Smart Internet of Things Solutions Using the Serious-Game Approach." *Fusion of Smart, Multimedia and Computer Gaming Technologies*. Springer International Publishing, 2015. 73-90.

[18] Bochon, Iryna, Volker Ivens, and Ralf Nagel. "Challenges of cloud business process management." *Cloud Computing for Logistics*. Springer International Publishing, 2015. 119-139.



이종섭(Jong-Sub Lee)

1990년 호서대학교 전자계산학 이학사
1994년 광운대학교 전자계산학 이학석사
1995~1997, 광운대학교 컴퓨터과학과 박사수료
2012 ~ 현재, 세명대학교 교양과정부 조교수
※관심분야 : 분산처리, 빅데이터, 클라우드, 모바일



문석재(Seok-Jae Moon)

2002년 광운대학교, 컴퓨터소프트웨어학과이학사
2004년 광운대학교, 컴퓨터과학과 공학석사
2010년 광운대학교, 컴퓨터과학과 공학박사
2014년 ~ 현재, (주)글로벌비트캠프연구소 수석연구원
2015년 ~ 현재, (주)이사야 전무이사
※관심분야 : 빅데이터, 클라우드 컴퓨팅, 데이터 상호운용, 시맨틱 공학