Jemal Abawajy
Dominik Slezak (Eds.)

Advanced Science and Technology Letters ASTL 13

Information Technology and Computer Science

Proceedings
International Conference, ITCS 2012
Porto, Portugal, July 2012

SERSC Science and Engineering Research Support soCiety

A Performance Model for Hypercube Based NoC with Fully Adaptive Routing Algorithm 68
Jin Liu, Xiaofeng Wang, Hongmin Ren, Jin Wang, Jeong-Uk Kim

Knowledge Modeling Guideline for Reentrant Features in SAGE 76
Jeong Ah Kim, InSook Cho, BinGu Shim, Min Hee Choi, SunTae Kim, Sun Ah Kim, Yongho Kim, Chang Hee Lee, Heeseong Yun and Dongkyu Seo

Test Case Extraction for Intelligent Power Switch of Heterogeneous Vehicles... 82
Dong Ho Kim, Hyun Seung Son, Woo Yeol Kim, and Robert Young Chul Kim

A Study on Requirement Extraction on Use Case Approach 89
Bo Kyung Park, and Robert Young Chul Kim

Query Language for Business Process Framework based on Closed Architecture 94
Chae Yun Seo, and Robert Young Chul Kim

Verification of Requirements Extraction and Prioritization using Use Case Points 100
So Young Moon, Bo Kyung Park, and Robert Young Chul Kim

Automatic Test Case Generation using Multiple Condition Control Flow Graph 105
Hyun Seung Son, Woo Yeol Kim, Jae Seung Kim, and Robert Young Chul Kim

A Multi-agent based Facility Maintenance Planning and Monitoring System: A Case Study 110
JaeHoon Lee , MyungSoo Lee, SangHoon Lee, SeGhok Oh, and JoongSoon Jang

The Archiving Method for Records of Public Sector's Facebook Page 117
Yun-Young Hwang, In-Ho Jang, Seung-Jun Cha and Kyu-Chul Lee

A Study on Architecture of a Cyber Checkpoint Model in the Cloud Computing Environment 123
Min-Woo Park, Jung-Ho Eom , Sung-Hwan Kim, Nam-Uk Kim, and Tai-Myoung Chung

Query Language for Business Process Framework based on Closed Architecture

Chae Yun Seo[1], and Robert Young Chul Kim[1]

[1] Dept. of CIC(Computer and Information Communication), Hongik University, Sejong Campus, 339-701, Korea
{jyun,bob}@selab.hongik.ac.kr

Abstract. Modern enterprises may be in need of developing the right application programs in a rapidly changing business environment for a fast time to market. However, most enterprises without a business process mechanism find it difficult to develop applications in time when they need to change their business. To solve this problem, we proposed a business process framework (BPF) based on closed architecture [4,5]. This approach easily develops the right application under rapid business process changes/improvements on time by reusing software modules through a mapping business process model (BPM), service oriented architecture (SOA), and component-based development (CBD). In addition, we defined five layers of a BPF. This paper enhances nested query language to access easily and quickly stored data of each layer in this BPF environment, and shows how to get information of each layer on closed architecture with this query language.

Keywords: Service, BPM, CBD, Layer Architecture, BPF(Business process framework), multiple query language

1 Introduction

Many business processes need to be able to change quickly and easily to cope with the internal and external demands of enterprises, using minimum information technology in a quickly changing business environment. However, it is difficult to change or maintain existing business processes or develop new business processes because complicated applications are integrated into diverse systems in enterprises. Such an environment delays quick business policy establishment and decision making in enterprises in a rapidly changing market, thereby hindering market competitiveness [1].

The framework of information engineering substantially reduces the system maintenance cost and difficulty of enterprises. The tools used in the framework enable perfectly structured designs and facilitate design modifications in most cases. The changed designs in turn generate new program codes. The framework tools eliminate problems by testing programs with incomplete documentation and modifying such programs [2].

We proposed a business process framework (BPF) that improves the information-engineering framework in order to develop an effective business system from existing research. This paper enhances Business Process SQL (BPSQL) based on nested Structured Query language (SQL) to search and access information in each layer on BPF[3,4,5,6]. This paper presents execution of BPSQL on the process between two and three neighboring layers. In particular, this paper focuses on the process for SQL to access the stored data table. The application of this process to an integrated information management system is also presented. Finally, the process to determine the data values of a layer in a BPF using BPSQL is verified.

This paper is described as follows: Section 2 describes the BPF. Sections 3 and 4 explain BPSQL for neighboring upper and lower layers and for neighboring upper, middle, and lower layers. Finally, Section 5 discusses the conclusions and further research.

2 Related Work

A BPF is comprised of five layers: business rules, business processes, services, components, and data modeling. A repository like DB tablization is present at each layer [5]. A BPF is a closed architecture where each layer is directly connected to the next layer.

The layered structure can quickly produce new services by reusing existing components when business demands exist. New businesses can be configured with this service [5]. The repository at each layer is tabulated, and a layer generates data query using BPSQL [6]. Finally, the required data are extracted.

3 BPSQL Processing in Each Layer

BPSQL enables access to and search of the data table mapped to the process layer. This section describes how to access the stored data table mapped in the process layer. Next, it discusses the verification of the access and search processes for the mapped data table in the neighboring upper and lower layers (two layers) and upper, medium, and lower layers (three layers) using BPSQL.

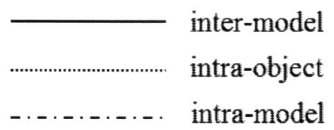

Fig. 1. Notation for BPSQL Processing

Figure 1 shows the notation used to explain the processing. The straight line is the connection to the internal model. The dotted line is the connection to sub-objects. The dash–dot line shows the connection to the upper/lower model.

3.1 BPSQL for migrating between Upper and Lower Layers

Figure 2 illustrates the BPSQL processing between neighboring upper and lower layers. UP_LAYER and DOWN_LAYER are the upper and lower layers, respectively. The neighboring upper and lower layers are represented by two layers. UP_LAYER can access DOWN_LAYER.

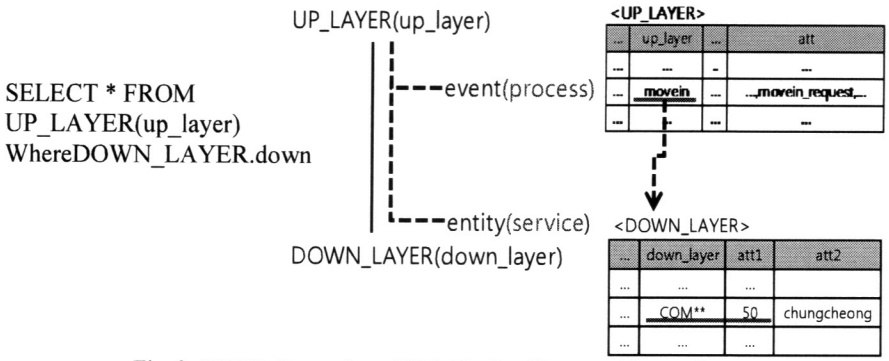

Fig. 2. BPSQL Processing of Neighboring Upper and Lower Layers

3.2 BPSQL for migrating between Upper, Middle, and Lower Layers

Figure 3 illustrates the BPSQL processing for neighboring upper, middle, and lower layers. The three-layered BPSQL is described by UP_LAYER, MIDDLE_LAYER, and DOWN_LAYER. As an example, for a three-layered BPF structure, UP_LAYER is the policy layer, MIDDLE_LAYER is the business process layer, and DOWN_LAYER is the service layer.

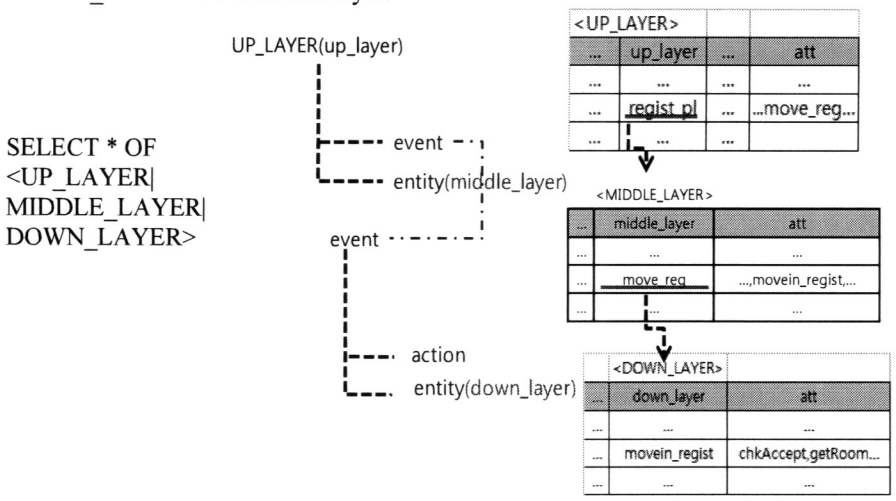

Fig. 3. BPSQL Processing of Neighboring Upper, Middle, and Lower Layers

4 Case Study

This section presents an application of the enhanced BPSQL processing to a BPF-based integrated information management system. The studied integrated information management system is developed by mapping the business process model and components in BPF. The top layer is the rule of the information management system. It contains simple rules for using the information management system. Furthermore, the business process, service, component, and data layers are configured for the information management system. Processing of the neighboring upper and lower layers (two layers) and upper, middle, and lower layers (three layers) for the application of BPSQL to the integrated information management system is explained herein.

The neighboring upper and lower layers (two layers) are described first. As an example, for the query "show the enterprise whose move-in request number is 50 in the move-in process," the BPSQL sentence is as follows.

```
SELECT * FROM movein(movein_request) Where movein_request.Num#=50
```

This is the information access using BPSQL to the lower layer, which is related to the move-in process layer. Figure 4 shows the BPSQL of the process–service layer. UP_LAYER is the business process layer, and DOWN_LAYER is the service layer. The "movein_request.Num#=50" event occurs in the move-in process. Next, the system obtains access to the lower layer mapped to the query.

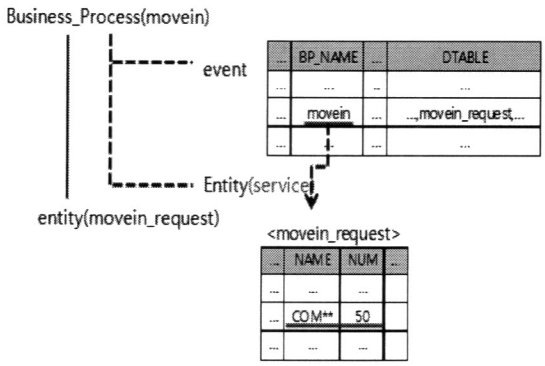

Fig. 4. BPSQL of accessing between Process and Service Layer

The neighboring upper, middle, and lower layers (three layers) are described next. As an example, for the query "show the enterprise moved in," BPSQL is as shown below.

```
SELECT  *  OF regist_pl(move_reg)
```

Figure 5 illustrates the BPSQL processing of the policy–process–service layer (three layers). In accordance with the defined BPSQL, "OF" can get access to all entities mapped to the table. The "regist_pl" event is entered to the policy and is enabled to get access to the lower layer (entity) mapped to "movein_ regis"

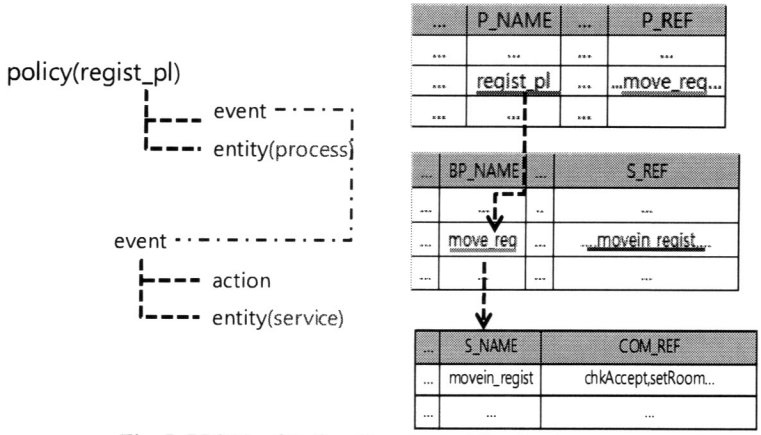

Fig. 5. BPSQL of Policy, Process, and Service layer

5 Conclusion

Our existing researches have proposed a closed structure for the business process framework; all information generated in a BPF is stored in the repository based table at each layer. This paper is enhanced BPSQL to easily and quickly search data saved in BPF. This BPSQL based on nested structured query language can easily gain access to layers and extract data from them.

This paper presents the access and search process of the data table mapped in neighboring upper and lower layers (two layers) and upper, middle, and lower layers (three layers) with our BPSQL. The process is applied to the development of an integrated information management system. Through such business process framework, we can also easily manage and integrate the separate data processing and decision-support systems built by different teams at different times in different places.

This framework mentioned in this paper easily and successfully should be implemented a business application using BPSQL.

Further research will focus on the application of open architecture-based BPSQL instead of closed architecture-based BPSQL for run time efficiency and management.

Acknowledgments. This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)(NIPA-2012-(H0301-12-3004)) and the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation.

Reference

1. Par Hedley Apperly, Ralph Hoffman, Et Steve Latchem: Service-And Component-Based Development. (2003)
2. James Martin: Information Engineering. Prentice-Hall International, Inc (1990)
3. Chae Yun Seo, So Young Moon, R. YoungChul Kim, Byoung-Ho Ahn, "A Study on Modeling Efficient Business Process Framework: Mapping Business process Layer and Data Layer". The 1st YEllow Sea International Conference on ubiquitous Computing, Shandong Univ, China. Vol. 1, p. 59 (2011)
4. Chae Yun Seo, Dong Woo Kim, R. Young Chul Kim, "Business Process Framework based on the Closed Architecture". Kais Vol. 10, No. 8, pp. 1939~1946 (2009)
5. Chae Yeon Seo, Dong Woo Kim, Jae Soo Kim, R. YoungChul Kim, "5-Layer Architecture for Efficient Business Process". IWIT Vol. 6, No. 1, pp. 19-22 (2008)
6. Chae Yun Seo, So Young Moon, R. YoungChul Kim, "A Study on Data Migration for BPSQL Query Language on Business Process Framework (BPF)". KIISE conference (2011)
7. Roger S. Pressman: Software Engineering A Practitioners' Approach. 3rd Ed, McGraw Hill (2004)