J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

Disjunctive Process Patterns Refinement and Probability Extraction from Workflow Logs☆

Kyoungsook Kim[1] Seonghun Ham[2] Hyun Ahn[2] Kwanghoon Pio Kim[2]

ABSTRACT

In this paper, we extract the quantitative relation data of activities from the workflow event log file recorded in the XES standard format and connect them to rediscover the workflow process model. Extract the workflow process patterns and proportions with the rediscovered model. There are four types of control-flow elements that should be used to extract workflow process patterns and portions with log files: linear (sequential) routing, disjunctive (selective) routing, conjunctive (parallel) routing, and iterative routing patterns. In this paper, we focus on four of the factors, disjunctive routing, and conjunctive path. A framework implemented by the authors' research group extracts and arranges the activity data from the log and converts the iteration of duplicate relationships into a quantitative value. Also, for accurate analysis, a parallel process is recorded in the log file based on execution time, and algorithms for finding and eliminating information distortion are designed and implemented. With these refined data, we rediscover the workflow process model following the relationship between the activities. This series of experiments are conducted using the Large Bank Transaction Process Model provided by 4TU and visualizes the experiment process and results.

☞ keyword : Workflow Process; Process Patterns, Proportional Information Control Nets, Workflow Logs, Temporal Work cases, Fidelity of Workflow Process, Model-Log Comparison, Workflow Intelligence. and Analytics

1. Introduction

In this paper, we try to realize a conceptual approach to disjunctive patterns in the business process model introduced in [1]. For realization, we refer to two algorithms. An algorithm for finding a parallel workflow model from the event log[2,3], and an algorithm for controlling path-based process knowledge analysis [4,5]. The critical problem with the control path oriented process knowledge analysis is that it can not be predicted because the runtime dynamically determines it. The workflow model has a parallel structure with many control paths rather than a simple serial structure. Besides, the execution frequency is different for each path because it is executed several times rather than stopping only once and is recorded in the log. Such information cannot be predicted because the workflow is added in real time while it is operating. Therefore, it is necessary to redesign and to engineer the workflow model using the log file containing the workflow's execution history. The authors' collaborative group has successfully invented amining framework that can discover disjunctive process patterns in workflow enforcement event logs, calculate quantitative values, and find out the proportions. Our experiments use the BPI challenges data [6] provided in 4TU. This data is provided for research purposes and is the actual workflow enforcement event log file.

This paper is described in the following order. The second sections summarize the scope of the problem of literature review and workflow process mining and knowledge discovery. The third section of the series describes a formal representation of the relationship between the type, format, and activity of the workflow event log. The fourth section presents some of the problems and solutions that can be encountered when re-engineering the workflow model. The fifth section explains the detailed experimental results by

1 Dept. of Computer Engineering, Kyunghee University, 1732 Deogyeong-daero Giheung-gu Yongin-si Gyeonggi-do, 17104, Republic of Korea
2 Div. of Computer Science and Engineering, Kyonggi University, 154-42 Kwangkyosan-ro Youngtong-gu Suwon-si Gyeonggi-do, 16227, Republic of Korea
* Corresponding author: (kwang@kgu.ac.kr)
[Received 30 December 2018, Reviewed 23 January 2019(R2 5 March 2019), Accepted 20 March 2019]
☆ Apreliminary version of this paper was presented at APIC-IST 2018.
☆ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (Grant No. 2017R1A2B2010697). This work also was partially supported by Kyonggi University's Graduate Research Assistantship 2018.

showing a series of disjunctive process patterns and proportions of using in the workflow process event log control path. The dataset used in the experiment is a subprocess of the large bank transaction process model [6]. The large bank transaction process model consists of ten thousand instances and consists of eight sub-processes. Finally section, we provide explanations for future research, conclusions, and conclusions.

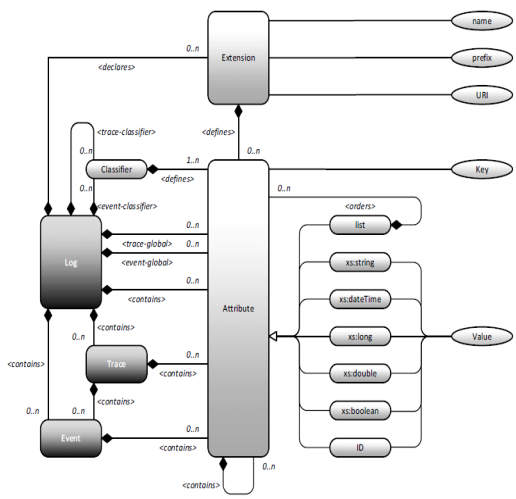
2. Related Work and Scope

The relevant topics of this paper are closely associated with the following issues. First, the concept of proportional workflow process models. This model can be formally represented by the Proportional Information Control Nets [1]. At this time, you should use the information about the workflow model, but you can get this information from the log file. Secondly, it is a way to rediscover the workflow model. This technique of rediscovering models from log files is done because the workflow model does not contain all the information. Due to the nature of the business process, the model is designed and then executed several times. At this point, the process chooses the path according to the status of the runtime in various control paths, and the usage proportion differs. This is what happens in the runtime, which is unknown when designing the workflow model. Due to the generation of additional information of this type of runtime, research [7] using Virtual Edges to check the control flow of the process dynamically was also conducted. However, due to the nature of business processes, dynamic processing is not appropriate. Therefore, additional information generated during runtime is obtained through the rediscovery of workflow model using a log file. Finally, it is a way to compare the workflow model and log. Two theories are presented in this method. One is the stochastic information control nets[8], the other is the proportional information control nets [1]. Stochastic information control netshave the ability to place probability distributions on OR control paths and AND control paths. Accurately, the estimated value is assigned to out-going arcs in the control path. Proportional information control nets can also place probability distributions in the control path. The difference with the previous method is that it is more accurate because it uses the observations found from the event log of the workflow process model.

In this paper, you can use the proportional information control nets to determine the overfitting or underfitting of the workflow process model and to calculate the fidelity of the model. Thus, the scope of this paper is to find the enactment proportions of disjunctive process patterns.

3. Workflow Event Log

As workflow process models are implemented, workflow event logs are recorded in specific data centers. Workflow event logs have various formats such as Common Workflow Audit Data (CWAD), Business Process Analytics Format (BPAF), and Mining Extensible Markup Language (MXML). In this paper, we use the XES[9](eXtensible Event Stream) format proposed by IEEE. XES consists of three layers: log, trace, and event. Events gather to form a trace, and traces assemble to create a log. The log layer contains metadata, and the trace and event layers have attribute values nd status values, including id. Figure 1 below shows the details of the XES log format. Attribute values re required to represent hierarchical elements: string, dataTime, long, double, boolean, ID. These elements are used in the workflow design to generate attribute values hat are purposeful.



(Figure 1) XES log data structure[9]

The workflow event log instantiates the process model in the workflow engine and logs the execution history of all activities by the workflow engine each time it is run. Therefore, the event log file is sequenced based on the execution time of the activity. Here, the expression representing the sequential relation between activities is as follows.

- f : $\delta = \delta_i \cup \delta_o$

Delta(δ) represents the relationship between activities. For activity α, δ_i represents an activity that passes an input value to α, and δ_o represents an activity that accepts a value from α as an input value. The relationship between these activities is mined from the event log and used for workflow reconfiguration and analysis.

4. Remove a fake parallel arcs

We can use the δintroduced in Section 3 above to express the relationship of all activities in the log. There is a redundant relationship becausea workflow model is a log file that is created by running multiple times. By combining the duplicate relationships and storing the number of times, the number of execution of each relationship can be known. This is defined as the quantitative relationship data of the activity. We found that there was a problem when using the data to reconfigure the workflow model.Compared to the original workflow model, part of the reconstructed workflow model has a different relationship. This is because the log file is recorded starting from the execution time of the event. This is because parallel processing is performed through an AND control path rather than a simple serial relation. The parallel processing from the AND control path is executed individually until it is merged in the JOIN control path. However, the parallel process is written in a log that is recorded based on the execution time. This means that the relationship to the parallel process is not properly represented in the activity's quantitative relationship data. In practice, there is no relation between activities, but the relationship arising from the above problem is defined as a fake parallel arc. The fake parallel arc must be removed for accurate mining. We have found that the quantitative value of the fake parallel arc in the quantitative relationship data of the activity defined above becomes larger than the quantitative value of the input (δ_i). Therefore, we use this data to find and delete the fake parallel arc. The following shows and describes the functions and algorithms required for fake parallel arcs.

- f: getDesCount(α) : Returns the number of successor activities (δ_o) for the input activity α.
- f: getCount(α, β) : activity Returns the quantitative value of the relationship from α to β. The relationship between α and β is symbolized as δ (α, β).
- f: compareAct(α, β) : It compares the contents of α and β of two activities and returns the same contents.
- f: setCount(α, β, num) : The quantitative value for the relation of δ(α,β) is changed to "num".
- f: remove(α, β) : Delete the relationship of δ(α,β)

Algorithm: Remove Fake Parallel Arcs

Input: Quantitative relationship data Set.

Output: Quantitative relationship data Set.

Begin

```
1.  For( α ∈ Activity Set )
2.      If( getDesCount( α) <= 1 )
3.          continue;
4.      If( getCount( δ( α), α) == getCount( α, δ( α)) )
5.          continue;
6.      arrFake.add( δ( α));
7.      For( β ∈ arrFake )
8.          arrRm.add(compareAct( δ( β),arrFake));
9.      Endfor
10.     For( γ ∈ arrRm)
11.         remove( β,γ)
12.     Endfor
13.     For( σ ∈ arrFake )
14.         setCount( α, σgetCount( δ( α), α));
15.     Endfor
16. Endfor
17. return modified data set
```

End

The above algorithm works as follows. The Quantitative relationship data Set created from the ancestor and successor activity set extraction algorithm [4] is received as an input

value. We take one from a set of activities through a for statement and call it α. First, we get back how many successor activities of α through getDesCount (α) function. If the value is less than or equal to 1, it is determined that there is no chance of a fake parallel arc because it is a general path, not a control path. If it is more than 2, it is a control path. The getCount function compares the quantitative values f α and δi (α), α and δo (α), respectively. If it is the same, it is OR, so go to the next activity with the continue function. If it is different, go ahead because it is AND. If the above two conditions are not satisfied, then the AND control path from α proceeds and a fake parallel arc is likely to occur. So we put an activity belonging to δo (α) in array arrFake. The value of arrFake is given to β through the for the statement. If the compareAct function finds any successor to βand any activity in arrFake, it returns. This process is judged as a fake parallel arc if there is a relation between α's successors. Since each successor in the AND control path operates independently. The returned activity is stored in the array arrRm to be removed, and the relationship between the activities stored in α and arrRm is cleared through the for the statement. Finally, we modify the quantitative value of the relationship between αand the activity in the array arrFake. This is because the AND control path requires that all quantitative values e equal to the quantitative values nitially received by α.

5. The Experimental Results

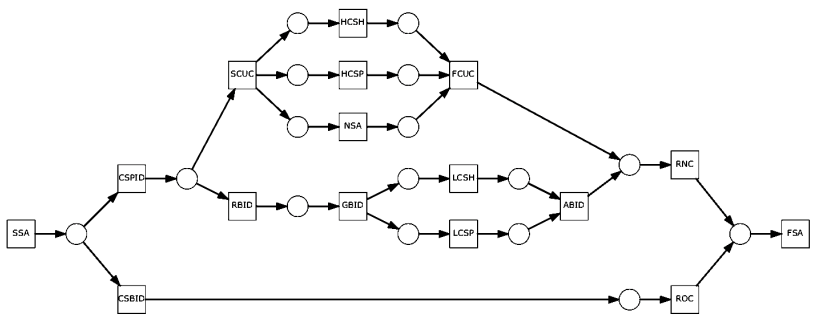
This chapter describes the process and results of the experiment. We did this using the process mining experimental framework developed by our cooperative research group and the Remove Fake Parallel Arcs algorithm presented in Chapter 4. The input dataset uses the Large Bank Transaction Process Model, one of the workflow event history logs published on the 4TU of the BPI Challenges website[6].

(Table 1) Activities on the Sender Authentification Sub-Workflow Process Model

The Formal ID	The Abbreviated	The Full Name of Activity
α_{6}	SSA	Start Sender Authentification
α_{85}	CSBID	Check Sender Bank ID
α_{7}	CSPID	Check Sender Personal ID
α_{63}	RBID	Request Bank ID
α_{64}	GBID	Generate Bank ID
α_{65}	LCSH	Low Check Sender Historial
α_{66}	LCSP	LowCheck Sender Profile
α_{67}	ABID	Activate Bank ID
α_{8}	SCUC	Start Check Unknown Client
α_{9}	HCSH	High Check Sender Historial
α_{10}	HCSP	High Check Sender Profile
α_{11}	NSA	Notify Sender to Authority
α_{12}	FCUC	Finish Check Unknown Client
α_{13}	RNC	Register New Client
α_{86}	ROC	Register Old Client
α_{14}	FSA	Finish Sender Authentification

5.1 The Log File for the Experiment

To experimentwe use the Large Bank Transaction Process Model. The model also provides an event log file and a petrinet-based workflow process modeling structure to check the accuracy of the test results. The model is made up of instances in full and has 125 jobs. Workflows are also



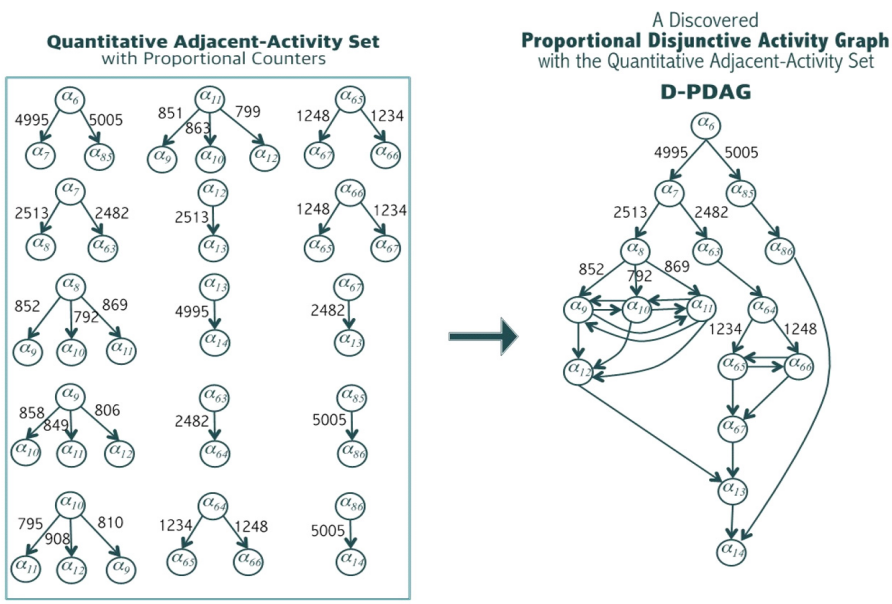
(Figure 2) The Petrinet-Based Sender Authentication Subprocess Model of the Experimental Workflow Model [6]

subdivided into eight sub-processes. We confirmed that the result of applying the event tracking mining algorithm developed by the research group is the same as the above information. For the sake of convenience, we will select one of the eight sub-processes to illustrate the experimental results of this paper. This subprocess model is named Sender Authentication and consists of 16 tasks. Fig. 2 and Table 1 show the Sender Authentication Subprocess Model. The figure is based on patrinet, and you can see two AND control paths and two OR control paths.

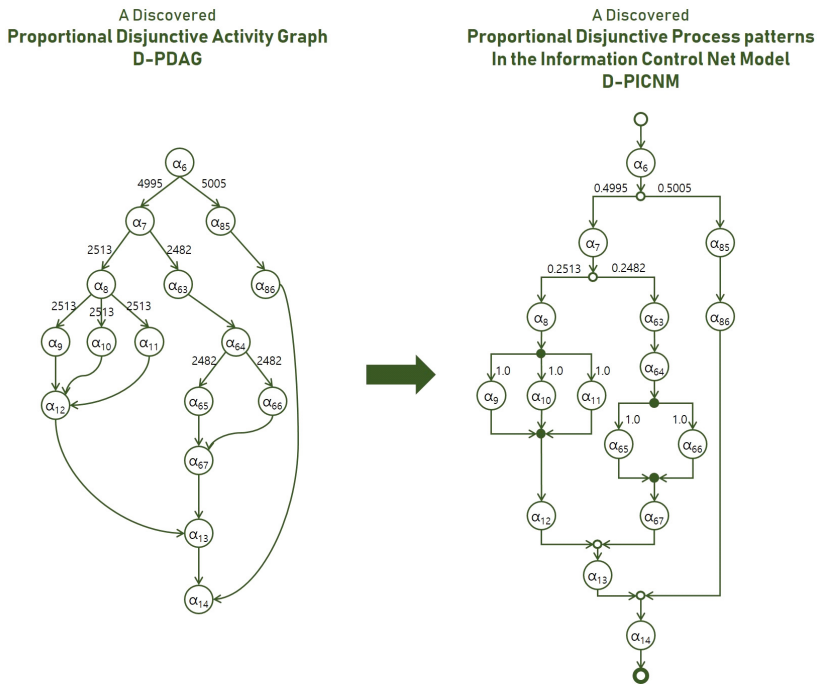
5.2 Mining a Proportional Disjunctive Activity Graph

The ultimate goal of this experiment is to combine the quantitative relationship data of the mined activities in the previous step to create a proportion disjunctive activity graph. We can use the ID of the activity to identify the sources and destinations of the relationships in the data. You can link these two relationships if the IDs of the destinations activity in one relationship are the same as the sources activity IDs in the other relationship. We have devised and implemented an algorithm to link activities in this way.

Fig.3 graphically shows a series of processes of extracting, sorting, and connecting the IDs of the activities from the log. The left side of the figure shows the quantitative relationship data of the activity. Two activities have a relationship, one activity is the source activity and the other activity is the destination activity. You can see which activity is the source activity from the starting point of the arrows that are linked to the activity and youcan get the quantitative value through the numbers. The proportional information control net is shown on the right. It was created by linking quantitative relationship data from the previously sorted activities. This proportional information control net was verified against the patrinet model in Fig. We found an activity relationship that did not exist in the original in the AND control path section. α9, α10, α11 The relationship between the three activities and the relationship between α65 and α66 activities. As mentioned earlier, run-time-based logging of log files creates a virtual relationship between the parallel processes. This incorrect information will affect the



(Figure 3) The Experimental Analytics intermediate Result from the Workflow Enacted Event History of the Sender Authentification Subprocess Model in the Large-Scale Bank Transaction Process Model [6]



(Figure 4) The Experimental Analytics Result of virtual parallel relationship removal processing from the Workflow Enacted Event History of the Sender Authentification Subprocess Model in the Large Bank Transaction Process Model (3)

results of the mining and should be removed. You must remove it using the Remove a fake parallel arcs algorithm described in Chapter 4 of this paper.

Fig. 4 shows the result of applying the Remove a fake parallel arcs algorithm to the previously extracted proportional information control net. In the left figure, we can see that there is no relation between three activities, α9, α10, and α11. α65, α66 are also the same. In the right figure, the quantitative value is converted to the proportion value. You cansee how much of the total number of workflows performed is in the path. The usage rate for each path can be used to determine the utilization and additionally information on the efficient allocation of resource values ssigned to that path. In certain situations, when the order of execution among parallel processes in the AND control path is meaningful information, the information can be preserved by omitting the quantitative input value conversion part in the Remove a fake parallel arcs algorithm.

6. Conclusions

This paper focuses on disjunctive process patterns and remove fake parallel arcs. It also aims to extract accurate enactment proportions from the workflow model. We deployed and experimented log files generated from the Large Bank Transaction Process Model in a framework developed by our collaborative research group, and created the correct dataset through the remove fake parallel arcs algorithm presented in this paper. The diagram and table show all the results and artifacts generated from all the experimental steps. These experimental results and outputs are concentrated in the disjunctive process pattern. In conclusion, through the experiment and the results, the conceptual approach proposed by the author's research group was implemented and supplemented, and the validity was verified to show the meaning of the workflow process model. We also aim to include loop process patterns as well as

disjunctive process patterns in the future. Loop process patterns are also frequently encountered in the workflow process model, which increases the number of executions in a given section, thus affecting workflow mining. The next goal is to create a workflow process mining framework covering both disjunctive process patterns and loop process patterns.

Acknowledgment.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (Grant No. 2017R1A2B2010697).

References

[1] K. im, M. Yeon, B. Jeong, and K. P. Kim, "A Conceptual Approach for Discovering Proportions of Disjunctive Routing Patterns in a Business Process Model," KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS, Vol. 11, No. 2, pp. 1148–1161, 2017.
https://doi.org/10.3837/tiis.2017.02.030

[2] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow Mining: Discovering Process Models from Event Logs," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, pp. 1128-1142, 2004
https://doi.org/10.1109/tkde.2004.47

[3] Lekic, Julijana, and Dragan Milicev. "Discovering models of parallel workflow processes from incomplete event logs." Model-Driven Engineering and Software Development (MODELSWARD), 2015 3rd International Conference on. IEEE, pp. 477-482, 2015
https://doi.org/10.5220/0005242704770482

[4] Minjae Parc and Kwanghoon Kim, "Control-path Oriented Workflow Intelligence Analyses," JOURNAL OF INFORMATION SCIENCE AND ENGINEERING, Vol. 24, pp. 343–359, 2008.
https://www.researchgate.net/publication/220587485

[5] Kwanghoon Kim, "Control-path Oriented Workflow Intelligence Analysis on EnterprizeWorkflow Grids," 2005 First International Conference on Semantics, Knowledge and Grid, pp.32-32, 2005
https://doi.org/10.1109/skg.2005.58tlf

[6] BPI Challenge 2012, 2013, 2014, 2015, 2016, 2017, 2018, 4TU.Centre for Research Data,
https://data.4tu.nl/repository/collection:event-logs-real.

[7] LiPing Liu, LinLin Ci, Wei Liu, and Hui Yang, "Control Flow Checking at Virtual Edges," KSII Transactions on Internet and Information Systems, vol. 11, no. 1, pp. 396-413, 2017
10.3837/tiis.2017.01.021

[8] Clarence A. Ellis, Kwanghoon Kim, Aubrey Rembert, and Jaques Wainer, "Investigations on Stochastic Information Control Nets," INFORMATION SCIENCES, Vol. 194, pp. 120–137, 2012.
https://doi.org/10.1016/j.ins.2011.07.031

[9] Günther, Christian W., and Eric Verbeek., "Xes standard definition," *Fluxicon Process Laboratories*, Vol 13, No. 14, 2009.
https://pure.tue.nl/ws/portalfiles/portal/3981980/692728941269079.pdf

◐ 저 자 소 개 ◑

Kyoungsook Kim

1984 B.S. in Computer Science, Kyonggi University
2001 M.S. in Computer engineering, Kyunghee University
2019 Ph.D. in Computer engineering, Kyunghee University
2019 ~ Present, Adjunct Professor, Department of Computer Science and Engineering, Kyonggi University
Research Interests : Large-scale database systems, applications, enterprise information systems, temporal databases.
E-mail : khmjmc@kgu.ac.kr

Seonghun Ham

2019 B.S. in Computer Science, Kyonggi University
2019~Present, M.S. Student in Computer Science, Kyonggi University
Research Interests : Workflow systems, discovery control flow, process mining, large-scale log analysis.
E-mail : shham9@kgu.ac.kr

Hyun Ahn

2011 B.S. in Computer Science, Kyonggi University
2013 M.S. in Computer Science, Kyonggi University
2017 Ph.D. in Computer Science, Kyonggi University
2018~Present, Assistant Professor of the Dept. of Computer Science and Engineering at Kyonggi University
Research Interests : Business Process Management, business process intelligence, process mining.
E-mail : hahn@kgu.ac.kr

Kwanghoon Pio Kim

1984 B.S in Computer Science, Kyonggi University
1986 M.S. in Computer Science, Chungang University
1994 M.S. in Computer Science, University of Colorado at Boulder
1998 Ph.D. in Computer Science, University of Colorado at Boulder
1998~Present, Professor of the Dept. of Computer Science and Engineering at Kyonggi University
Research Interests : CSCW, workflow systems, Business Process Management, process mining, enterprise social network analysis.
E-mail : kwang@kgu.ac.kr