

A String Comparison Approach to Process Logic Differences between Business Process Models

Yeh-Chun Juan¹

¹Department of Industrial Engineering and Management Information, Huaan University, Taipei (223), Taiwan
E-mail: ycjuan@cc.hfu.edu.tw

Abstract

Analyzing process logic differences between company's process and best practice process can assist project team in discovering process improvement opportunities. Process logic comparison has been mentioned in some literature, but research in this area still lacks effectively and efficiently analytical approach.

This research has applied string coding and comparison to analyze the process logic differences between business process models. There are three portions in this approach. First of all, all process paths embedded in each compared process model are identified and coded into alphabetical process path strings. Then, all process path pairs necessary for further processing difference analysis are decided. Finally, process path string comparison is applied to each process path pair to identify the differences in processing sequence and processing mode.

Keywords: String coding, String comparison, Process logic, Gap analysis, Process benchmarking

1. Introduction

Recently, process benchmarking is standing out among various process redesign methodologies. Process benchmarking intends to look for performance improvement opportunities by analyzing the process logic differences between company's processes and benchmarking partners' *best practices* processes.

Since a business process is a series of activities executed according to appropriate process logic to create a value for customers, the process logic of a process can be defined as a composite of alternative process paths and sequences designed in a process for various business scenarios [1], and what is more, a specific process path should be chosen and executed to respond to the encountered situation. Therefore, the process logic differences between two business processes indicate the differences in alternative process paths for various business scenarios, unique activities, processing mode, such as sequential and

parallel, and processing sequence between compared business process models.

To facilitate business process improvement projects, some business process engineering (BPE) systems integrating various specialized methodologies and techniques for business process modeling and analysis have been developed.

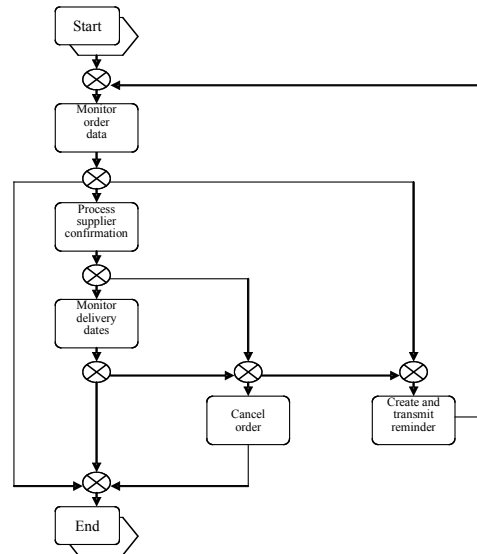


Fig. 1: An 'order monitoring' process model.

As shown in Fig. 1, most of the existing BPE systems have adopted a flowchart-like model to describe the process logic, the temporal and logical sequence, of a process. To illustrate alternative process paths and parallel paths within a process, two logical operators, 'exclusive OR' (XOR) and 'AND', are always introduced. In this research, the logic operators, XOR and AND, are denoted by \otimes and \cap , respectively.

Although BPE systems have successfully adopted flowchart-like models to describe the process logic of a process and Tenner and DeToro [2] have mentioned the concept of man-made side-by-side flowchart comparison, some complex process models still make project teams with little or no awareness of the alternative process paths and processing sequence.

As shown in Fig. 1, there are as many as seven alternative process paths embedded in this process model. Man-made analysis of process logic differences between flowcharts is absolutely a heavy and complicated task. Therefore, this research has developed a string coding and comparison approach to process logic differences between business process models. Besides, embedding this approach in BPE systems can facilitate project team to easily understand the process logic differences between their own process and benchmarking partners' process.

2. Problem Description and Literature Review

As mentioned above, several issues about process logic differences between business process models can be raised.

- How to search out alternative process paths embedded in each compared process model?
- How to identify the processing differences including the processing sequence and mode differences between process paths?

For the first issue, since a process model has mingled several alternative process paths designed for various business scenarios, the problem is how to search out these process paths for further processing difference analysis.

As for the second issue, since knowledge of the processing differences is conducive to process redesign, for two process paths designed for similar business scenario, the differences in unique activities, processing mode, such as sequential and parallel, and processing sequence between them should be analyzed.

Wang [3] used IDEF0 to model a business process. And the connections between activities in IDEF0 models are divided into five groups. The IDEF0 model is transformed into a process string on the basis of the five connection types. Besides, a method based on the process string has been proposed to measure the similarity degree of two IDEF0 models. However, the measure considers only the similarity degree of connections, i.e. the similarity of processing mode, in IDEF0 models, but the analysis of similarity degree and interrelationship between activities, such as processing sequence, in IDEF0 models is omitted.

Juan and Ou-Yang [4] developed an algorithm to search out the alternative process paths in a flowchart and the process segments in a process path. And then, the semantic similarity degree (SSD) is used to measure the similarity degree of activities contained in compared process paths or process segments. As regards the difference analysis of processing sequence and mode still need to draw support from project team.

In manufacturing engineering area, identical or similar parts and process plans can be searched out from a retrieval computer-aided process planning (CAPP) system by matching the group technology (GT) codes of parts [5]. Since the GT codes are predefined on the basis of the design/manufacturing attributes and usually appear in the form of a digital/alphabetical string, similarities among parts and their process plans can be identified by applying string comparison techniques to their GT codes. String coding and string comparison have played an important role in these applications. This research just intends to applying the string coding and comparison techniques to automate the difference analysis of processing sequence and processing mode, which is a semi-automated work in Juan and Ou-Yang [4].

3. The Proposed Approach

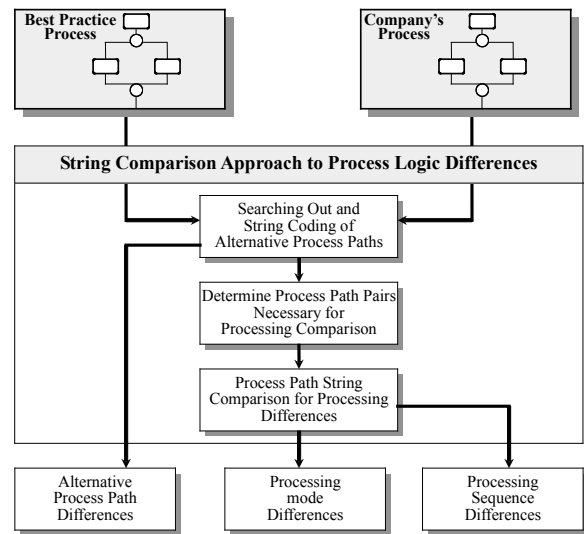


Fig. 2: The proposed string comparison approach to process logic differences.

As shown in Fig. 2, basically, there are three portions in the proposed approach. First of all, all the alternative process paths embedded in each process model are identified and coded into alphabetical process path strings. Then, process path pairs necessary for further processing difference analysis are decided. Finally, process path string comparison is applied to each process path pair to identify the processing sequence and mode differences between process paths of compared process models. Now let's describe the details of this approach.

3.1. Searching Out and String Coding of Process Paths

This step is to search out all the alternative process paths embedded in compared process models and coded them into alphabetical process path strings.

The string coding method is explained first. As mentioned before, a process model is composed of activity nodes, XOR operators, and AND operators. As Fig. 3(a) showed, for each activity node with different name in compared process models, an exclusive letter is given to it. The activity nodes with the same name should share the same letter. Then, the process string is derived according the following coding rules:

1. The series of connected activity nodes is coded as a string of their corresponding letters.
2. XOR operators will result in two or more parallel series of connected activity nodes, the letter strings used to describe these parallel series of connected activity nodes will be connected with ‘×’ and all these letter strings will be parenthesized with ‘(’ and ‘)’.
3. AND operators also can result in two or more parallel series of connected activity nodes, all the letter strings will be connected with ‘+’ and parenthesized with ‘(’ and ‘)’.

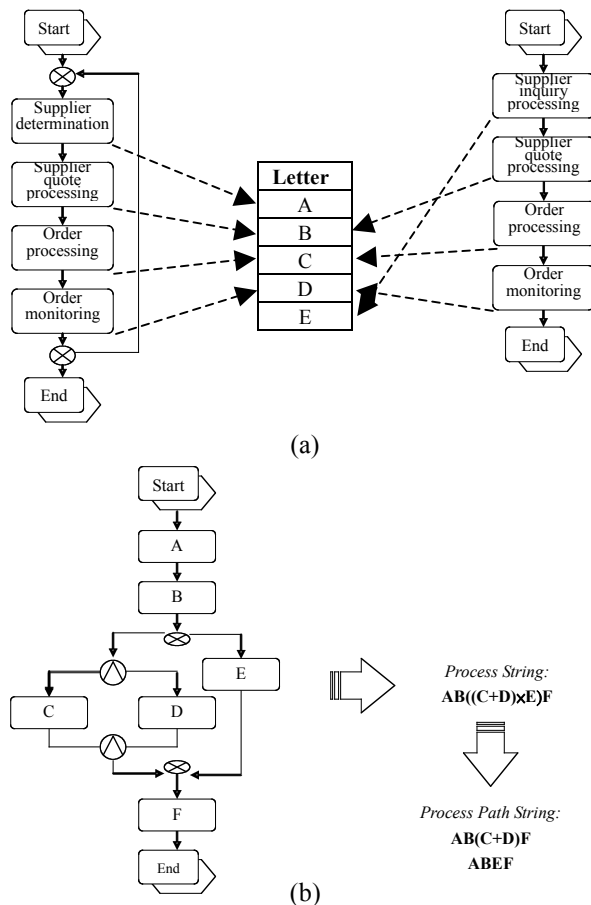


Fig. 3: Searching out and string coding of alternative process paths.

For example, the process model of Fig. 3(b) will be coded as ‘AB((C+D)×E)F’.

After the process string is derived, all the alternative process paths embedded this process should be searched out next. As mentioned above, the role of XOR operators in a process model, i.e. the ‘×’ signs in a process string, is to choose and activate a proper process path for a specific business situation. Therefore, the alternative process path strings can be extracted by expanding the process string using ‘×’ signs. As Fig. 3(b) showed, the process string ‘AB((C+D)×E)F’ can be expanded by the ‘×’ sign to form two process path strings, ‘AB(C+D)F’ and ‘ABEF’. So, each compared process model can be transformed into one or more process path strings.

3.2. Determine Process Path Pairs

Once all process path strings of compared process models have been derived, the next work is to determine the process path pairs necessary for further processing difference analysis.

The method used here is modified from Juan and Ou-Yang [4]. Basically, the process path pairs could be sieved out by the following two rules:

1. The ‘Linear’ process path, i.e. the process path string with no duplicate letters, represents a series of activities have been performed sequentially or concurrently to complete or cancel the process. On the other hand, the ‘Loop’ process path, i.e. the process path string with duplicate letters, means a series of activities must be reworked until business conditions changed. Since they are different semantically, making a processing comparison between them is unnecessary.
2. The two process paths to be chosen for further processing difference analysis should be designed for similar business situation. This can be judged by $SSD \geq \text{threshold}$ where the SSD is a measure proposed in Juan and Ou-Yang [4]. Here the SSD of a process path pair is modified as:

$$SSD(P_1, P_2) = \frac{2|CLet|}{|Let_{P_1}| + |Let_{P_2}|}$$

Symbols:

P_1, P_2 : Two process path strings.

$|Let_{P_1}|, |Let_{P_2}|$: Number of letters in P_1, P_2 .

$|CLet|$: Number of common letters in P_1 and P_2 .

3.3. Process Path String Comparison

After finding proper process path pairs, the next step is to analyze the processing differences for them. Here

the processing differences are analyzed by process path string comparison. The string comparison will focus on three subjects: unique activities, processing mode and processing sequence.

Unique activity analysis: By comparing two process path strings, PPS_1 and PPS_2 , three kinds of letters could be found. The first is those letters common to PPS_1 and PPS_2 , i.e. $CLet = Let_{PPS_1} \cap Let_{PPS_2}$. The second is those letters specific to PPS_1 , i.e. $SLet_{PPS_1} = Let_{PPS_1} - Let_{PPS_2}$. And the third is those letters specific to PPS_2 , i.e. $SLet_{PPS_2} = Let_{PPS_2} - Let_{PPS_1}$. Project team should discuss whether these specific activities are value-added or negligible tasks. Those negligible activities should be eliminated. Besides, project team should pay attention to the processing sequence and mode of common activities.

Processing mode analysis: First, $SLet_{PPS_1}$ and $SLet_{PPS_2}$ are deleted from PPS_1 and PPS_2 , respectively. So, the two deleted strings are composed of $CLet$. By comparing two deleted strings, two kinds of letters could be found. The first is those letters which appear outside the parentheses of PPS_1 and inside the parentheses of PPS_2 . The activities represented by these letters are executed sequentially in PPS_1 and concurrently in PPS_2 . The second is those letters which appear inside the parentheses of PPS_1 and outside the parentheses of PPS_2 . The activities represented by these letters are executed concurrently in PPS_1 and sequentially in PPS_2 .

Processing sequence analysis: Since the substrings in parentheses connected with '+' sign are executed concurrently, their sequence is interchangeable. In this step, therefore, the permutations of substrings in each parentheses of compared process path strings should be derived first. And the permutation will form a set of substitutes for each compared process path strings. Then, all parentheses and '+' sign are deleted from all substitute process path strings. Third, by comparing two sets of substitute process path strings letter by letter, the pair of substitute process path strings with a minimum difference is drawn for processing sequence analysis. For the drawn string pair, the same letters appearing in different position represent the corresponding activities are executed in different sequence.

As two process paths in Fig. 4, PP_1 is coded as 'ABC', PPS_1 , and PP_2 is coded as 'DB(A+E)C', PPS_2 . First, $SLet_{PPS_2} = PPS_2 - PPS_1 = \{D, E\}$ is derived. Secondly, comparing PPS_1 with $SLet_{PPS_2}$ -deleted PPS_2 , we can find activity A is executed sequentially in PP_1 while it is executed concurrently in PP_2 . Finally, by deleting the parenthesis and '+' sign from $SLet_{PPS_2}$ -deleted PPS_2 and comparing it with PPS_1 , we can find activities A and B are executed in different sequence in PP_1 and PP_2 because they are in different position.

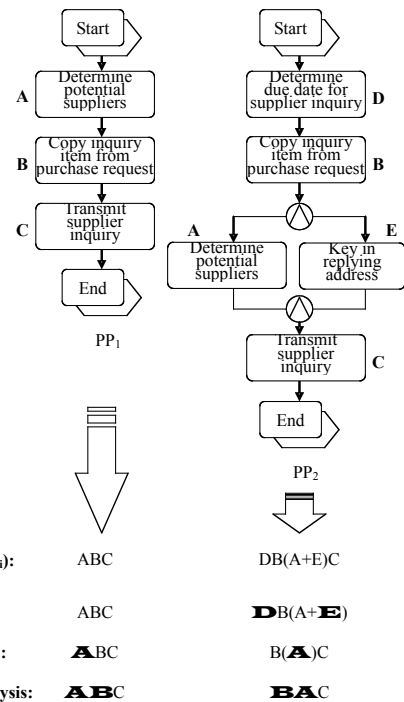


Fig. 4: Process path string comparison for processing difference analysis.

4. A Case Study

Here the proposed approach was applied to the analysis of process logic differences between two 'Purchase order processing' process models, as shown in Fig. 5(a), which are excerpted and modified from the SAP R/3 reference model 4.0B [6]. The process of process logic difference analysis would be as follows.

1. As Fig. 5(b) showed, two process paths have been searched out for PM_1 and PM_2 respectively and coded into process path strings.
2. From the SSD of process path pairs shown in Fig. 5(c), $(PM_1.PPS_1, PM_2.PPS_1)$ and $(PM_1.PPS_2, PM_2.PPS_2)$ are chosen for further processing difference analysis if the threshold is 0.8.
3. As Fig. 5(d) showed, the major processing differences of two chosen process path pairs are on 'unique activity' and 'executed mode'.

5. Conclusions

Process logic difference analysis has become the most important task in process benchmarking. This research has developed a string comparison approach to assist project team in analyzing process logic differences between process models.

6. Acknowledgements

The research was sponsored by the Republic of China (ROC) National Science Council under the project number NSC 93-2213-E-131-009.

7. References

[1] G. Keller, and T. Teufel, *SAP R/3 process oriented implementation*, Addison Wesley, 1998.
 [2] A.R. Tenner, and I.J. DeToro, *Process redesign: the implementation guide for managers*, Addison Wesley, 1997.
 [3] C.M. Wang, *BASS: BPR Analogy Support System*, Master Thesis, Department of

Information Mgmt., National Taiwan University of Science and Technology, ROC, 1996.
 [4] Y.C. Juan, and C. Ou-Yang, "A process logic comparison approach to support business process benchmarking," *International Journal of Advanced Manufacturing Technology*, Vol. 26, pp. 191-210, 2005.
 [5] S.S. Heragu, "Group technology and cellular manufacturing," *IEEE Transactions of Systems, Man, and Cybernetics*, Vol. 24, pp. 203-215, 1994.
 [6] IDS prof. Scheer GmbH, *ARIS connectivity for R/3: installation guide*, IDS Scheer AG., 1998.

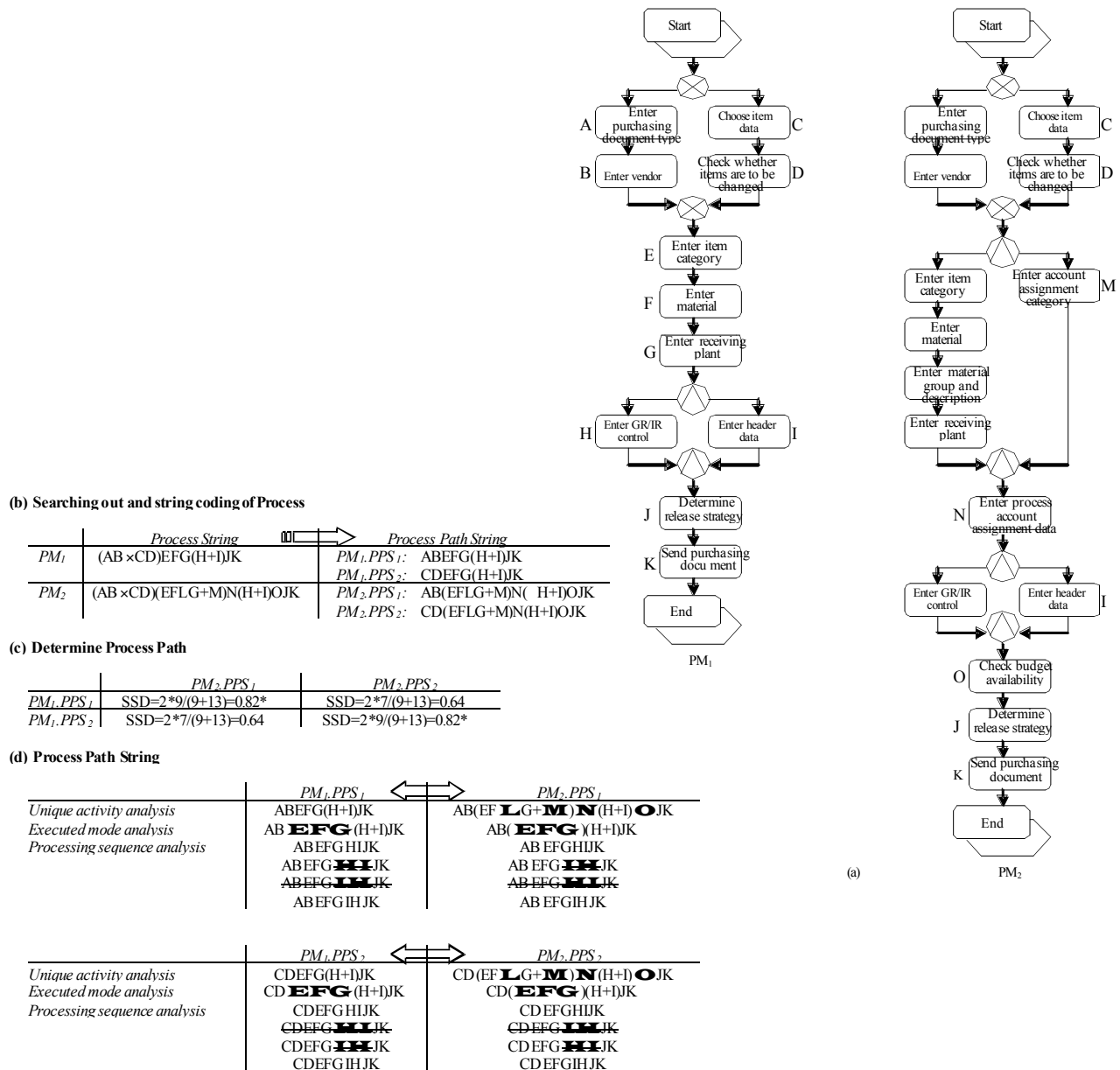


Fig. 5: Process logic difference analysis of two 'Purchase order processing' process models.