Research Journal of Applied Sciences, Engineering and Technology 4(24): 5561-5564, 2012
ISSN: 2040-7467

Submitted: April 25, 2012 Accepted: May 13, 2012 Published: December 15, 2012

Rationality Validation Algorithm Based on Petri Net

HaiLan Pan, AnBao Wang and CuiHong Wu
Department of Computer and Information, Shanghai Second Polytechnic University, Shanghai, China

Abstract: In order to inspect the correctness of workflow during process design and avoid exceptions after processes run, this study provides rationality Validation Algorithm based on Petri Net. Firstly the study points out the necessity of rationality validation in workflow model, then analyzes all kinds of existing validation algorithm of workflow model rationality and emphatically describes a validation algorithm based on Petri net. At last this algorithm is improved and the correctness of the algorithm is verified by an instance.

Keywords: Petri net, rationality, state reachability, validation algorithm

INTRODUCTION

Workflow management technology has been successfully used in many enterprises, which is used as an effective tool for supporting process modeling and business process automation. The rationality of workflow model directly influences scheduling and management of activities operating in time and it is also very important to execution and monitoring of enterprise business process automation (Zhou *et al*., 2005a). At present the main methods of workflow modeling mainly include method based on activity network, dialogue, transaction and formalization etc and the technique of Petri net is a relatively successful application in method based on formalization.

Petri net is a graphical and mathematical modeling tool and it has standard model semantics, because all its elements are strictly defined (Yuan, 2004). In order to inspect the correctness of workflow during process design and avoid exceptions after processes run, this study provides rationality Validation Algorithm based on Petri Net. This study is mainly to analyze existing validation algorithms of workflow model rationality and a validation algorithm based on Petri net is described and improved, then the correctness of this improved algorithm is verified by an instance of handling complaint.

THE DEFINITIONS OF PETRI NET AND WF-NET

Petri net is expressed with formula PN = (P, T, F) and the letter P stands for Place and the letter T stands for Transition and the letter F stands for collection of Connections.

Definition 1: WF-net (Wang and Wen, 2002). If Petri net PN = (P, T, F) can be defined as WF-net, it must satisfy the following three conditions:

- Existing a source Place *i* and it satisfies $i = \phi$
- Existing a output Place *o* and it satisfies $o = \phi$
- Adding a new Place *t* to PN and it is used to connect Place *i* and Place *o* and satisfies $t^* = \{O\}$ $t^* = \{i\}$, then this PN is strongly connected.

Definition 2: Rationality of WF-net (Wang and Wen, 2002)

The original state M_0 of workflow is defined as follow: source Place *i* has one Token and the Token quantity in the other Place is zero. A WF-net (PN, M_0) is rational when it satisfies the following three conditions:

- For state M which can be reached from each original state M_0, there is an implemented sequence realizing from state M to end state M_e
- State M_e is the only end state that can be reached from state M_0 and this state contains a Token only in output Place o
- There is no endless Transition in (PN, M_0)

THE VALIDATION ALGORITHM OF WORKFLOW MODEL RATIONALITY

This section emphatically analyses a validation algorithm described in study (Zhou *et al*., 2005; Aalst, 1998) and this algorithm is perfected and improved and the improved algorithm will be verified in the following section.

The idea of algorithm implementation: The idea of algorithm implementation is divided into five steps:

Step 1: Establish matrix based on WF-net and express anteroposterior sequences of each Transition by

Corresponding Author: HaiLan Pan, Department of Computer and Information, Shanghai Second Polytechnic University, Shanghai, China

matrix and add additional sequence to show each Transition's former and subsequent quantity and logical relationship.

Step 2: According to anteroposterior Places of each Transition expressed by WF-net, determine rules of vector's representation when states are changed.

Step 3: Start off from initial state, according to the preorder Place of Transition and Transition triggered after judging by matrix and combine the subsequent Place of Transition, the new status can be determined.

Step 4: Repeat step 3 until no changes occur.

Step 5: If the final states are all shaped like state of (0, 0… 1) and all Transitions are all triggered and all Places are passed, then workflow expressed by Petri net is correct, otherwise this workflow is error.

The specific description of algorithm: The following matrix describes the sequence between each Transition:

$$\begin{array}{l} \textit{column additional sequence} \\ ABC\ldots \\ U = \begin{bmatrix} \textit{elements in matrix uses 1} \\ \textit{or 0 to show whether there} \\ \textit{is reachable path form} \\ \textit{Transition represented by} \\ \textit{line to Transition represented} \\ \textit{by columns} \end{bmatrix} \begin{array}{ll} A & \textit{row} \\ B & \textit{additional} \\ C & \textit{sequence} \\ & \\ & \ldots \end{array} \end{array}$$

The collection of input Places of Transition t is represented by •t and the collection of output Places of Transition t is represented by t•, so the algorithm can be realized by linked list. The symbols of •t and t• are respectively used to determine vector representation of state producing latest.

All states are described by vector of n dimension and n is the numbers of Places, so M_0 is defined as the initial state." History" used to store workflow states that have happened is the collection of states, the original "History" only exists the initial state M_0. "Now" is also the collection of states, which used to store workflow states that can be performed immediately and the original "Now" also only exists the initial state M_0. "Transitions" used to store Transition that have been triggered is the collection of Transition and the original "Transitions" is null. " Places" used to store Place that have been reached is vector of n dimension and Places [*i*] shows component *i* and Places [*i*] = O indicates that Place *i* has not reaches and Places [*i*] = 1 indicates that Place *i* has reaches and the original "Places" is (1, 0… 0).

If vectors X, A, B are known and X = A+B, then:

$$Xi = \begin{cases} 1 & others \\ 0 & Ai = 0 \wedge Bi = 0 \end{cases}$$

The improved validation algorithm of workflow model rationality is as follows:

- Build matrix according to WF-net and the element of "1" or "0" in matrix separately means whether reachable paths exist between Transition represented by row and Transition represented by column.
- Add additional sequence to describe number of before and subsequent Transitions and logical relationship including and-split and-join, or-split and or-join.
- Find out collections of input and output Places of each Transition according WF-net and the symbols of •t means collections of input Places and the symbols of t means collections of output Places.
- Initialize variables: M_0 = (1, 0… 0), History = {M_0}, Now = {M_0}, Transitions = M_0, Places = (1, 0…0) (vector dimension decides by numbers of Places in WF-net).
- There are k Transitions (t_1, t_2…t_k) can be triggered in this state and the sequence is judged by matrix. After Transitions are respectively triggered state of M_1, M_2…M_k are produced and input and output Places of Transitions judge vector description of states. If $M_i \notin History$ ($i \in [1, k]$), then Places = Places+M_i, History = History+{M_i} and Transitions = Transitions+ {t_1, t_2…t_k}.
- There is no Transitions can be triggered in this state. If M_i = (0, 0…1), return and get next element; If $M_i \neq$ (0, 0…1), report error and exit program; and do until elements of Now are all reached.
- Repeat (5) until "Now" becomes null and Transitions = T and Places = P.
- The algorithm finishes.

The algorithm validated by instance of handling complaint: Here the new algorithm will be verified by an actual business process of handling complaint.

The description of instance: WF-net of process of handling complaint is shown in Fig. 1.

In the process of handling complaint, complaint just received is firstly recorded and the customers who complain and related departments are contacted. According to the information collected from department and customers, all data must be sorted to decide handling result that department pays the compensation or system sends rejection letters to customers. Finally complaints will be filed and the whole process is over.

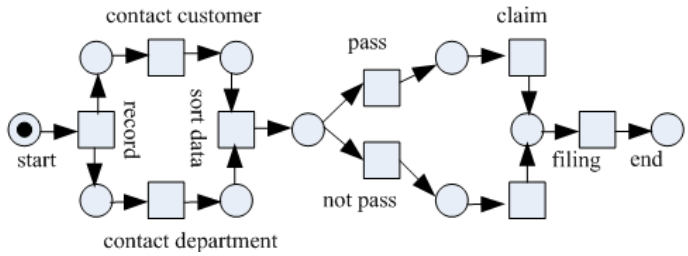

Fig. 1: The process of handling complaint

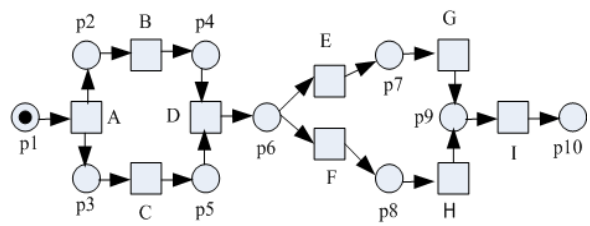

Fig. 2: WF-net numbered according to sequence of breadth first

The prerequisite of algorithm validated: In order to facilitate verification using algorithm, Transitions and Places of this process are represented as Fig. 2.

In Fig. 2 Transitions and Places of Fig. 1 are numbered according to sequence of breadth first and Places are shown by p1, p2 pn (n is sum of Places), Transitions are shown by A, B, C.

The process of algorithm validated: According to algorithm defined as above, the process of validation is as follows:

- Build matrix according to Fig. 2 and add additional sequence to describe number of before and subsequent Transitions and logical relationship including and-split and-join, or-split and or-join:

$$U = \begin{array}{c} \text{And-join} \quad \text{Or-join} \\ \begin{matrix} 0 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 2 \\ A & B & C & D & E & F & G & H & I \end{matrix} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{ll} A & 2 \; And-split \\ B & 1 \\ C & 1 \\ D & 2 \; Or-split \\ E & 1 \\ F & 1 \\ G & 1 \\ H & 1 \\ I & 0 \end{array}$$

- Input and output Places of each Transition are as follows: •A = {p1}, A• = {p2, p3}, •B = {p2}, B• = {p4}, •C = {p3}, C• = {p5}, •D = {p4, p5}, D• = {p6}, •E = {p6}, E• = {p7}, •F = {p6}, F• = {p8}, •G = {p7}, G• = {p9}, •H = {p8}, H• = {p9}, •I = {p9}, I• = {p10}.
- Because numbers of Places of WF-net in Fig. 2 are ten, dimension of vector is also ten, then below variables is initialized: M_0 = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0), History = {M0}, Now = {M_0}, Transition = Φ, Places = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0).
- Six steps are as follows:

Step 1: Because •A = {p1} judges that Transition A is triggered, State M_0 is switched to State M_1. Because A• = {p2, p3}, M_1 = (0, 1, 1, 0, 0, 0, 0, 0, 0, 0), Places = (1, 1, 1, 0, 0, 0, 0, 0, 0, 0), History = { M_0, M_1}, Now = {M_0} and Transition = {A}.

Step 2: State M2 is produced when Transition B and Transition C (and-split) are triggered and these are judged by matrix U, •B = {p2} and •C = {p3}. Because B• = {p4} and C• = {p5}, M_2 = (0, 0, 0, 1, 1, 0, 0, 0, 0, 0), Places = (1, 1, 1, 1, 1, 0, 0, 0, 0, 0), History = {M_0, M_1, M_2}, Now = {M_2} and Transition = {A, B, C}.

Step 3: State M3 is produced when Transition D (and-join) is triggered and these are judged by matrix U and •D = {p4, p5}. Because D• = {p6}, M_3 = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0), Now = {M_3}, Places = (1, 1, 1, 1, 1, 1, 0, 0, 0, 0), History = { M_0, M_1, M_2, M_3} and Transition = {A, B, C, D}.

Step 4: State M4 or State M5 is produced when Transition E or Transition F (or-split) is triggered and these are judged by matrix U, •E = {p6} and •F = {p6}. Because E• = {p7} and F• = {p8}, M_4 = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0), M_5 = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0), Places = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0), History = {M_0, M_1, M_2, M_3, M_4, M_5}, Now = { M_4, M_5} and Transition = {A, B, C, D, E, F}.

Step 5: State M6 is produced when Transition G or Transition H (or-join) is triggered and these are judged by matrix U, •G = {p7} and •H = {p8}. Because G• = {p9} and H• = {p9}, M_6 = (0, 0, 0, 0, 0, 0, 0, 0, 1, 0), Places = (1, 1, 1, 1, 1, 1, 1, 1, 0), History = {M_0, M_1, M_2, M_3, M_4, M_5, M_6}, Now = { M_6} and Transition = {A, B, C, D, E, F, G, H}.

Step 6: State M7 is produced when Transition I is triggered and these are judged by matrix U and •I = {p9}. Because I• = {p10}, M_7 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1), Places = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1), History = {M_0, M_1, M_2, M_3, M_4, M_5, M_6, M_7}, Now = { M_7} and Transition = {A, B, C, D, E, F, G, H, I}.

Now = Φ, Transitions = T and Places = P.

This WF-net satisfies correctness according above validation process.

CONCLUSION

The method of workflow modeling based on Petri net capitalizes the strict definition of semantic and graphic language in Petri net, so this approach realizes clear and strict definition of workflow process and also facilitates to analyze and verify model (Aalst and Basten, 2002; Li *et al*., 2010). This study analyzes all kinds of existing validation algorithm of workflow model rationality and emphatically describes a validation algorithm based on Petri nets, then this algorithm is improved and the correctness of the algorithm is verified by an instance. Of course, there are far from enough to verify the reasonability of the model, control and optimization of model need be realized and all these will be carried out research in the future study.

REFERENCES

Aalst, W., 1998. The application of petri nets to workflow management [J]. J. Circuit. Syst. Comput., 8(1): 22-66.

Aalst, W. and T. Basten, 2002. Inheritance of workflows: An approach to tackling problems related to change [J]. Theor. Comput. Sci., 270(1-2): 125-203.

Li, X., T. Shen, M.H. Yao and X.J. Tang, 2010. The research and application of flexible technology about workflow management system [J]. Comput. Technol. Dev., 20(1): 120-122.

Wang, J.M. and L.J. Wen, 2002. Workflow management-model, method and system [M], Tsinghua University Press, Beijing.

Yuan, C.Y., 2004. The Principle and Application Based on Petri Net [M]. Electronic Industry Press, Beijing.

Zhou, J.B., H. Lin and Z.C. Yu, 2005a. Workflow Optimization Analysis Based on Petri Net [J]. retrieved from: http://159.226.25.137:81/Jweb_zgglkx/EN/.

Zhou, F.M., B. Wu and Q. Gu, 2005b. Workflow modeling and correctness analysis based on Petri net [J]. Comp. Sci., 32(2): 121-124.