

Study on the Modeling Method of Software Process based on Timing and Parallel Automata

Sun Fengwei^{1, a}, Yuan Lijuan^{2, b}

¹ Institute of Computer Science, Chagnchun Information Technology Cologe, Changchun China

² Information Technology Department, Baoding University, Heibei of Baoding, China

^akingmesh@gmail.com, ^bylj0312@sina.com

Keywords: Software Process Technology; Process Modeling Method; Process Model; Software Process Definition Reuse

Abstract. The finite automata theory extended and then the timing parallel automata theory is got and applied in the software process modeling. The establishment of group software process model is on the basis of timing parallel automata which realize the activity planning, resource allocation and progress control of process. The process model has been checked rationality and the rationality definition and check rules have been given. The process modeling method in this paper is intuitive, easy to understand and could describe the dynamic change of process, and also present the concurrent activity and provide the effective support to parallel work and cooperative work.

Introduction

Process modeling is the reprocessing of software process activities, which refers to the development and maintenance of software products, project management, process management and improvement, it involves to many aspects of the formal methods that used in modeling, couples with its inherent complexity, the modeling methods of software process in construct the process model are varied. The automata theory is based on the set theory and the formal modeling tools of state logic, it has a strict semantics, graphical language expression methods and it can describe the status of system implementation and the sequence of activities.

Construction of group software process model based on the timing parallel automata

The establishment of software process model that based on the timing parallel automata is the establishment of mapping from software process concept to timing parallel automata concept, and represents the execution order among the various activities with the timing parallel automata. Activity states and their conversion relations are shown in Fig. 1.

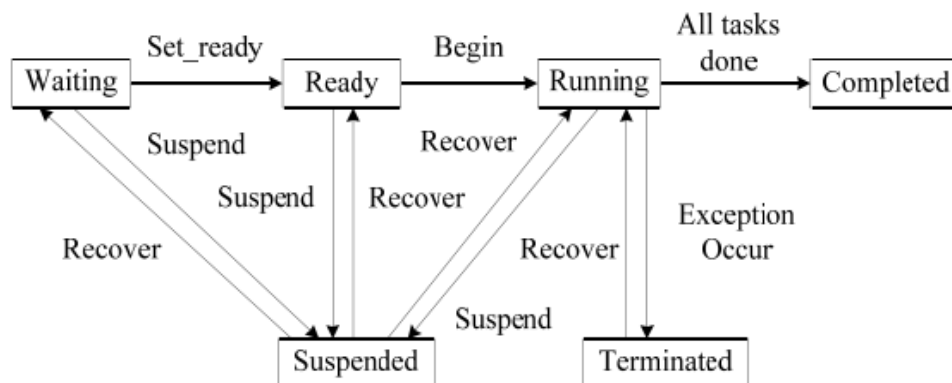


Fig. 1.State Transition Diagram of Activitis

Mapping of concept: Activities corresponding to the node set N of the timing parallel automata, the node set presents the collection of the whole activities of software process. Each state in S presents the activity collection which could appear in all the dynamic status at a certain moment.

Each activity in the software process is executed in order that is in accordance with the corresponding control structure. The activities could transform that based on the triggering events and the corresponding conditions.

Automata occurs state transition while the characters are read-in in the timing parallel automata; events and conditions triggered the changes of the activity in the software process. Therefore, events and conditions can be mapped to the input character of timing parallel automata. If the activated event collection is Events, the condition collection is Conditions, so the input character set of timing parallel automata could express as $\Sigma = \text{Events} \times \text{Conditions}$.

Mapping of the process control structure: The paper divides the control structures of team software process into sequential structure, the AND-join structure, the AND-split structure, the OR-join structure, the OR-split structure and the loop structure. These six control structures can be used as the basic structure of software process execution, all the implementation structures of the process are constructed by these six basic control structures.

- Sequential structure

As the most basic structure in the software process, sequential structure indicates that the order relationships that exist between the various activities. While the shift condition is satisfied, the next activity could begin after the end of the previous activity. The mapping of sequential structure to the timing parallel automata structure is that: the node represents activity, the connection between them using arc, the receiving characters are marked on the arc. As Fig. 2 (a) shows, when activity A has been completed, and the character a1 is read-in in the time t1 and then activate the activity B.

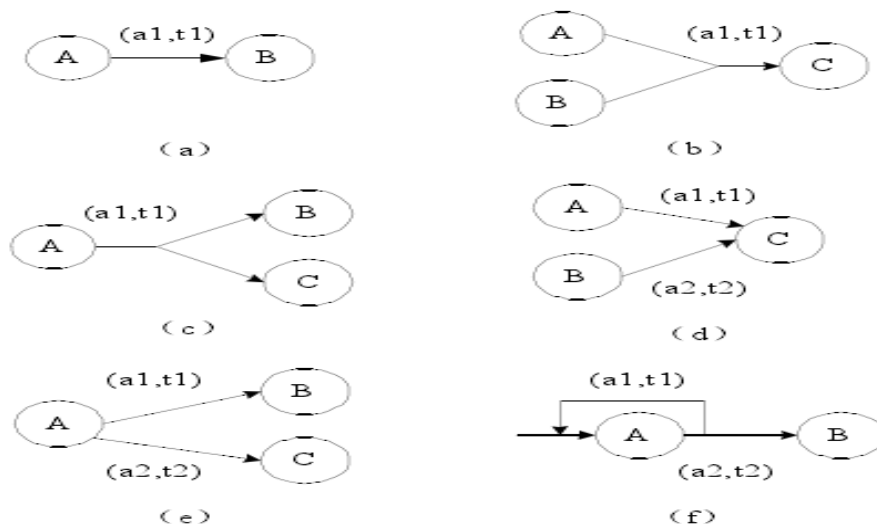


Fig. 2. The Mapping of Process Control Architecture

- AND-join structure

The AND-join structure describes the synchronization and relationship of different paths branch activities. The transfer of junction must ensure that the transfers on each branch have been completed. The mapping of AND-join structure to timing parallel automata structure is that: when the parallel branch activities have been completed and characters are read-in, the following activities are activated. As the Fig. 2 (b) shows that, while activity A and B have been completed, and the character a1 has been read-in in the time t1 and then the activity C is activated.

- AND-split structure

The AND-split structure describes the concurrent processing relationships of each branch activities after the transfer. The mapping methods of AND-split structure to timing parallel automate structure is that: when the precursor activities of various branches have been completed and the characters are read-in, the various branches activities are activated.

- OR-join structure

Join the several paths together which indicate that any one path can lead the OR-join structure to trigger. The mapping method of OR-join structure to timing parallel automata is that: when any precursor activities have been completed and the characters have been read-in, the activities can be activated.

- OR-split structure

The OR-split structure describes the selectable approaches of software process operation; it would choose one of the subsequent branches and only the one. The mapping method of OR-split structure to timing parallel automata structure is that: every event and its precursor are connected by arc, when precursor activity has been completed and activate one of subsequent activity according to the read-in character.

- Loop structure

Loop structure represents the repeated executions of activities until a certain condition is met. The mapping of loop structure to timing parallel automate is that appear a loop while corresponding to the node chart.

TSP which achieves low cost and short period in term of time and cost constraint is a multi-objective optimization problem and as a kind of directed stochastic intelligence optimization method evolutionary algorithm has some characteristics suitable for solving multi-objective optimization problem, therefore process modeling method based on multi-objective evolutionary algorithm is put forward. Then the modeling process of using the process modeling method above is described in detail. And a instance is given to prove the validity of model.

The problem of comprehensive evaluation of process modeling methods is researched in order to help developers choose the most appropriate modeling method based on specific modeling environment and requirement for achieving the best modeling effect. A fuzzy-grey comprehensive evaluation method for process modeling methods is presented. An evaluation system for software process modeling methods is given. A comprehensive evaluation method which combines fuzzy evaluation and grey theory is proposed. The evaluation method can make full use of fuzziness and grayness of evaluation information by experts to make the evaluation more objective and accurate.

To solve current problems, such as low definition efficiency, lack of relations between definitions and difficult definition maintenance, referencing to class inheritance in Object-Oriented Design, the definition of process inheritance is given with application of process reuse mechanism applied in process definition. Then software process definition reuse method base on process inheritance and realization way of process inheritance are given. The definition method defines a new process by inheriting existing process definitions which can implement process reuse, hence avoid "from scratch" process definition, increase definition efficiency and reduce definition cost. In addition, by inheritance changes of process definitions can be reflected more effectively and the transition of process definition instances are more convenient, thereby system flexibility and system maintenance efficiency can be increased significantly.

The reasonable inspection of process model

1. The correctness of the semantics process model must be ensured so as to establish the right process model. The model is reasonable while process model meets the semantic constraint. If the model is discovered unreasonable after its operation, the process model and the running process instance must be modified, which may cause part of the modified of running process instances, and also may lead to the re-running of the whole process. If the high cost of implementation of an activity, then the modification would be very costly.

2. Comparison of related process modeling method.

3. As a formal process modeling methods, the process modeling method of timing parallel automate in this paper is compared with the others, as shown in table 1.

Table 1. Comparison of Related Method

Comparison	representation	understandability	support for parallel
The process of program design method	Programming language	weak	Can not support a wide range of concurrent activities
Function decomposition method	process elements set,	weak	support
Modeling method based on rules	logic rules	weak	disable to provide effective support

Modeling method based on plan	process steps set and constraint set	weak	support
Modeling method based on knowledge,	process knowledge	weak	nonsupport
The method proposed in this paper	chart	strong	intuition, understandability, support

Summary

On the basis of the finite automata, the concept of timing parallel automata and the software process modeling method based on the timing parallel automata have been raised. The concept and process control structure of the team software process have been mapped in the timing parallel automata so that the team software process model that based on the timing parallel automata has been established. The model has a rigorous semantics and it is represented by graphical language so that would easy to understand. It can help developers to analyze the resource allocation, development progress, process control and other aspects of TSP.

References

- [1] LIU Shi-lan, REN Hao, "The Comparison Between Total Quality Management(TQM) and Traditional Management(TM)", Commercial Research, vol.337, pp.122-124, 2006.
- [2] HE Zhen, ZHOU Shan-zhong, "Process Management for Continuous Quality Improvement", Industrial Engineering Journal, vol.5(8), pp.38-41, 2005.
- [3] HE Zhen, LU Jin, LIU Xiao-liang, HE Shu-guang, "Design And Development of the Integrated Quality Management System IQMS2.0", Industrial Engineering Journal, vol.10(6), pp.54-58, 2007.
- [4] YUAN Yi-jun; LIU Hao, "Service outsourcing and promotion of technical innovative efficiency in manufacturing industry", Journal of Dalian University of Technology(Social Sciences), vol.28(4), pp.1-6, 2007.
- [5] CHEN Guo quan, MA Meng, "Studies on the process model of organizational learning", JOURNAL OF MANEGEMENT SCIENCES IN CHINA, vol.3(3), pp.15-23, 2000.
- [6] Edmondson, Mo ingeon, "Organizational learning and competitive advantage", New York: Mc Graw-Hill, pp.1-10, 1997.
- [7] DAI Wan-wen, ZHAO Shu-ming, Steve F Foster, "Study on a dynamic model of organizational learning processes under complex system", Studies in Science of Science, vol.24, pp.217-224, 2006.
- [8] Lenstra J K, Rinnooy K, 1981, "Complexity of vehicle routing and scheduling problem," Networks, Vol.11, pp. 221-227.
- [9] Davis L, 1991, "Handbook of genetic algorithm," New York: Van Nostrand Reinhold.
- [10] Goldberg DE, 1989, "Genetic algorithms in search, optimization, and machine learning," Reading: Addison-Wesley.
- [11] Holland J.H., 1975, "Adaptation in Natural and Artificial Systems," Ann Arbor:University of Michigan Press.
- [12] C. Sangit, C. Cecilia, A.L. Lucy, 1996, "Genetic algorithms and traveling salesman problems," European Journal of Operational Research, Vol.93, pp. 490–510.
- [13] Zhao He, Du Duanfu, 1998, "The Operators Choice and Design of Genetic Algorithm for TSP," Systems Engineering: Theory & Practice, Vol.18, pp. 62–65.