Mining Workflow Event Log to Find Parallel Task Dispatching Rules

Liu Yingbo[1,2], Wang Jianmin[2] and Sun Jiaguang[2]

[1]Department of Computer Science, Tsinghua University 100084, Beijing, China

[1]School of Software, Tsinghua University 100084, Beijing, China

Abstract. In many workflow applications, actors are free to pick up work items in their work list. It is not unusual for an actor to start a work item before completing previously accepted one. Frequent occurrence of this behavior implies potential patterns of work parallelism, which is serviceable to a workflow scheduler to better dispatch ongoing tasks. In this paper, we apply association rule mining techniques to workflow event log to analyze various kinds of activity parallel execution patterns. When an actor accepts a new work item, the parallel execution rules mined from event log can help a workflow scheduler to find those work items that might be suitable to be undertaken by the same actor simultaneously. In the experiment on three vehicle manufacturing enterprises, we have found 32 strong rules of 40 different workflow activities. We describe our approach and report on the result of our experiment.

1 Introduction*

It is a common experience in our daily life that some tasks could be done in a parallel way, for instance, "In the way I go to buy milk, I might as well bring back some bread…", A smart use of this knowledge is, in the end, an aid to a our productivity and results in a great saving of time. In today's enterprises, employees have already learned to coordinate their activities in such a manner. Consider, for example, a vehicle manufacturing enterprise we investigated, within a period of 31 months there are 99765 work items of 922 different kinds of activity that has been performed by 147 actors, statistics of workflow event log shows that more than one third of completed work items (36541) are performed in parallel with one or more other ones by the same actor, which means potential parallel work patterns may exist among activities.

As a means of discovering these parallel activities, we present an approach to analyze the workflow event log. This approach first groups different workflow event entry into transactions and then association rules mining is applied to the transactions to find which activities are likely to be executed together. This information can help a

* This work is supported by the Project of National Natural Science Foundation of China (No. 60373011) , the 973 Project of China (No.2002CB312006) and Program for New Century Excellent Talents in University (NCET)

Yingbo L., Jianmin W. and Jiaguang S. (2007).
Mining Workflow Event Log to Find Parallel Task Dispatching Rules.
In *Proceedings of the 4th International Workshop on Computer Supported Activity Coordination*, pages 45-54
DOI: 10.5220/0002408300450054

workflow management system better coordinate the activities of ongoing processes and it also helps process designers gain insight into potential work parallelism, thereby, improve the process design accordingly.

With this approach, we have been able to find out 12 rules of 10 activities in a data set of 31-months and we have also obtained 18 rules of 26 activities in a 14-months data set. However, in a data set of less than 4 months, only 2 rules of 4 activities were found.

This paper makes two contributions: it presents an approach of identifying parallel activity execution rules and it evaluates the possibility of applying this approach on real data sets.

The rest of paper is organized as follow: We begin by introducing the background of parallel work dispatching mechanism in workflow and the enterprises we investigated (Section 2), then we introduce our approach of finding parallel execution rules, including data preprocessing and data mining method (Section 3). After introducing our approach, we keep on presenting the experiment result (Section 4) and we discuss on these results (Section 5). Related efforts on process mining and process scheduling are given in section 6. Finally, we summarize our work (Section 7).

2 Background

Understanding our approach requires a basic knowledge about work item dispatching mechanism in workflow[1]. In addition, we provide an overview on three enterprises we investigated.

2.1 Parallel Work Item Dispatching Mechanism in Workflow

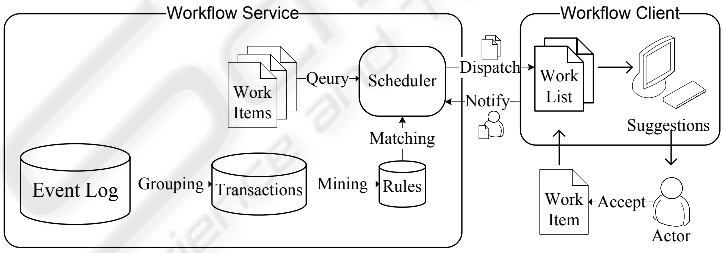

Fig. 1. Batch work dispatch mechanism in workflow management systems.

Figure 1 illustrates the architecture of parallel work dispatching mechanism in workflow, the workflow service reads the event log (far left), groups the event entries into transactions, mines the transactions for rules which describe implications between activities: "If an actor is going to buy some milk then he might be willing to bring back some bread too.", when an actor accepts a work item (say "Buy Milk"), the workflow client notifies the scheduler this fact and the scheduler match the rules with ongoing processes to find appropriate work items (say "Buy Bread") for parallel execution, then, it dispatch these work items to the work list as a suggestion to the actor.

In the discussion of this paper, we focus on finding parallel execution rules. Next, we provide information about three enterprises as a further introduction to the background.

2.2 Overview of Workflow Event Log in Enterprises

All these three enterprises are vehicle manufacturing enterprises. We investigate them because workflow is successfully used in many aspects of their business, like: configuration change, order processing, design review, technical notification, standard release, and new material classification etc.

Table 1. General overview of three enterprises' workflow event log.

Enterprise	**A**	**B**	**C**
Operation Time	117 days	421 days	949 days
Number of Actors	179	244	147
Workflow Activities	256	399	922
Completed Work Items	10808	42099	99765
Parallel Execution	3600	13675	36541

Table 1 is a general overview of workflow event log in these enterprises. In order to maintain confidential, we use A, B and C to represent them. As illustrated in the table, the workflow system in these enterprises has a different length of operation time. Besides, there are many actors who have completed lots of activities, which clearly reveals the fact that workflow has been heavily used. Moreover, in all these enterprises, nearly one third of completed work items are parallel executed by the assigned actor, which motivates us to investigate the parallel execution patterns. Next we present our approach.

3 Mining Parallel Activities

3.1 Grouping Event Entries to Transactions

We begin by introducing some definitions for event log, event entries, and transactions, adopting some notions used in [1] [2].

Definition 1 (Workflow Event Log). Let *A* be the set of activities defined in workflow models, *P* a set of actors and *T* a set of time points. $E=A\times P\times T^2$ is the set of possible event entries (e.g. *<a, p, s, e>* means the execution of *a* is performed by *p* started from *s* and ended at *e*). In the following discussion, we use $C\in 2^E$ to represent all the event entries in an enterprise's workflow event log.

Definition 2 (Parallel Executed Activities). For a given event entry c∈ *C (c = <a, p, s, e>)*, the parallel executed activities of this event entry are those activities that has already been accepted by actor *p* when the activity *a* starts, they can be defined by following function $T: C \rightarrow 2^A$:

$$\forall c = <a,p,s,e> \in C, T(c) = \{a' | < a', p', s', e' > \in C \wedge p' = p \wedge s \in [s', e']\} \tag{1}$$

Definition 3 (Transactions of Workflow Event Log). The transactions of a workflow event log are obtained by collecting all the parallel activity sets that contains two or more different activities, and it is denoted by $T^+[C]$.

$$T^+[C] = \{T(x) \mid x \in C \wedge \#T(x) > 1\} \tag{2}$$

3.2 Excluding Outlier Event Entries

In real situation, there always exist some work items which are executed for an extremely long period of time. For instance, in Enterprise C, the maximum activity execution time is 252 days. These long lasting work items are usually caused by business irrelevant reasons like unfinished testing or invalid operation etc. Their execution always overlaps with the execution of other work items, and inevitablly leads to a large number of unnecessary transactions. Therefore, it is important to preprocess the event log by excluding these outliers.

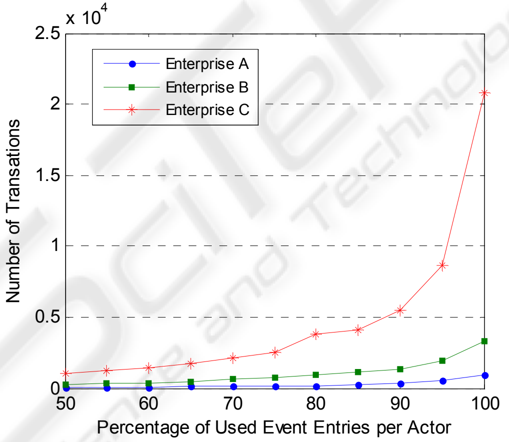

Fig. 2. Number of transactions for different percentage of event entries.

In our experiment, we use K percent shortest executed event entries for each actor as a way of data preprocessing. Figure 2 illustrates the trend of transaction number with K = 50, 55…90, 95, and 100. It clearly shows that K percentage filtering has caused a significant reduction of transaction number, especially in Enterprise C. However, the reduction tends to be small as K decreases. Therefore, we select the value of K by minimizing the reduction of transaction number while maintaining as many event entries as possible. Table 2 lists the selected K and the transaction number in three enterprises.

Table 2. Identified transactions in three enterprises.

Enterprise	A	B	C
Selected K	95	95	90
Identified Transactions	516	1878	5471
Included Activities	110	225	527

3.3 From Transactions to Rules

Given the transactions as described in the previous section, the next step is to find out parallel activity execution rules. Our approach of finding parallel activities is based on Association Rule mining technique[3]. This technique takes as input a set of transactions and generates a number of interesting rules.

Typically, association rules are considered interesting if they satisfy both a minimum support and minimum confidence. In the context of parallel activity association rule mining, the support and confidence can be interpreted as follow:

For any two activity sets *S* and *D*, $S \cap D = \Phi$, the rule *S=>D* holds in the transaction set $T^{+}[C]$ with **support** *s*, where *s* is the percentage of the transactions in $T^{+}[C]$ that contains $S \cup D$, this is taken to be the probability, $P(S \cup D)$. The rule *S=>D* has **confidence** *c* in the transaction set $T^{+}[C]$ if *c* is the percentage of transactions in $T^{+}[C]$ containing *S* that also contain *D*. This is taken to be the conditional probability, $P(S \mid D)$.

In practice, such thresholds are set by users or domain experts. Since we are not familiar with the business context, it is unlikely for us to correctly specify these parameters at the first beginning. Therefore, we determine such thresholds by enumerating different values of support and confidence, and then we decide which value is suitable for mining by observing the number of generated rules.

4 Experiment and Results

We use the Apriori Algorithm[4] to compute the association rules, the Apriori Algorithm takes a minimum support and minimum confidence and then computes the set of all associations rules in two phases: First, it iterates over the set of transactions and forms the frequent activity sets from the activities that satisfy minimum support. Then, it computes rules from the frequent activity sets. More precisely, from each of the frequent activity set $S \in A$, it creates those rules $S - X => X$ where X is a subset of *S*. (Note that all these rules have the same support, but different confidences.) Only rules that are above the minimum confidence are returned.

In order to determine a suitable value for each enterprise, we enumerate different support threshold from 0.2% to 7% with an increment of 0.2%, and for each support value, we further enumerate the confidence threshold from 50% to 100% with an increment of 5%.

4.1 Selecting Support and Confidence Threshold

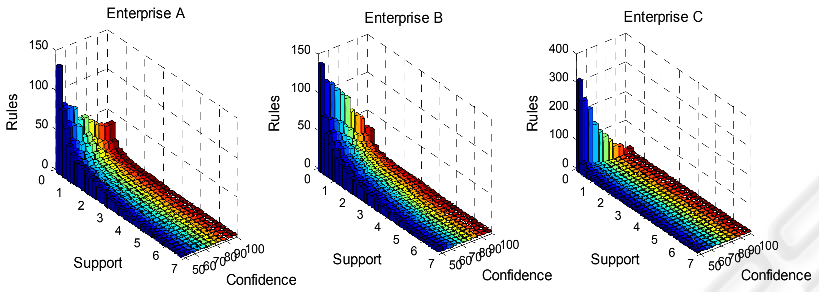

Fig. 3. Box plot of rule number with different support and confidence threshold.

Figure 3 illustrates the number of generated rules with different values of supports and confidences using 3D bar chart. In three enterprises, the number of rules reduces as support and confidence thresholds become higher. However, in Enterprise C, this number drops significantly as support threshold changing form 0.2% to 0.6%. In our opinion, it is related to domain specific reasons, since in enterprise C, there are 5471 transactions. Changing of support from 0.2% to 0.6% means the requirement of minimum occurrence for each rule changes from 11 to 33, which is a big difference in an enterprises' daily operation. However, in enterprise A and B, the difference of is much smaller under the same requirement of confidence (from 1 to 3 in A and from 4 to 11 in B). Therefore, it is reasonable for us to consider a rule to be interesting according to its occurrence number. Table-3 lists the selected support/confidence threshold and the number of generated rules.

Table 3. Generated rules of selected support and confidence threshold.

Enterprise	A	B	C
Selected Support	4%	1.6%	0.8%
Minimum Occurrence	21	30	44
Selected Confidence	85%	70%	65%
Conditional Occurrence	18	21	27
Number of Rules	2	18	12
Parallel Activities	4	26	10

4.2 Some Example Parallel Activity Execution Rules

According to Table 3, the number of interesting rules are not as many as we had expected, further investigation shows that, in all three enterprises, each rule contains no more than two different activities, most of these activities are similar kinds of work in different process definitions. Some rule examples are listed below (at the end of each rule, the support, occurrence number and confidence are provided):

- In Enterprise A, the only tow rules are all concerned with the review activity of cooling treatment operation. It is described as follow:

Cooling Treatment Operation in Second Level Component Confirmation Process of Part Design Drawing => Cooling Treatment Operation in Second Level Component Confirmation Process of Technological Document (5.8%, 30, 93.3%)

- In Enterprise B, the activities of assembling shop confirmation in different engineering change management processes are always occurred together:

Assembling Shop Confirmation in Technological Notification Process => Assembling Shop Confirmation in Quality Standards Notification Process (3.9%, 74, 100.0%)

- In Enterprise C, the activity of check payment and issue invoice is occurred in an extremely frequent way during product sales related processes:

Check Payment in Product Release Process => Check Payment in Product Shipment Process (16.1%, 883, 92.4%)

Issue Invoice in Product Release Process => Issue Invoice in Product Shipment Process (11.6%, 634, 91.8%)

5 Discussion

Until now we have been describing the results, in this section, we provide our discussion on the results and some possible issues.

5.1 Evaluation on Applicability

In our opinion, the only sure way to know whether our approach are helpful for improving enterprise's business process is to perform an empirical study in which we put the role of evaluation into the hands of domain experts in three enterprises. Because, a full utilization of these rules will depend to some degree on familiarity with enterprises not the workflow system, for this reason, we argue that such a study is absolutely necessary and we plan one as part of our future work.

5.2 Possible Extensions

We believe our approach warrant further study because of the context in which it can be applied, for activity parallel execution is just one form of pattern. As a matter of fact, there are many other kinds of patterns behind the parallel executed activities, for instance, association between design documents of different product families, and association between customer requirements etc. Therefore, other mining approaches can also be applied, such as multidimensional association rule mining and symbolic interval knowledge mining[5] etc. The mined result may provide more business oriented insights beyond our common understanding of enterprises.

5.3 Threats to Validity

Despite the result reported in this paper, there are some threats to the validity of our approach that need to be mentioned:

Firstly, we have studied the event log of three vehicle manufacturing enterprises. Although these enterprises themselves are very different, they just represent part of the characteristics of enterprises in vehicle manufacturing domain, thus, we cannot claim that their workflow execution histories would be representative in other enterprises and more importantly, in other domains. Secondly, we have made no attempt to assess the quality of event log, thus, the rules we mined may reflect good practice as well as bad ones. However, we believe that actors make more "good" decisions than "bad" ones, thus, there is more good than bad to learn from history.

6 Related Work

Our work applies a data mining approach to schedule work items among actors in workflow management systems. Hence, it is related to process mining and process scheduling.

6.1 Process Mining

Process mining is a vast area in the literature of workflow, it aims at improving the application of workflow by providing techniques and tools for discovering process, control data[6],[7],[8],[9] organizational[10], and social structures[11] from workflow event logs. Although, there is a considerable amount of work about process mining, most of them concerns discovering process model from event log, early effort on this topic is presented in [6] by Agrawal et al. and in [12] by Cook and Wolf, however, their approaches are limited to sequential processes. In [13], Aalst et al presented a new algorithm to extract a process model from event log and represent it in terms of Petri net. Later on, Wen Lijie et al improved Aalst's algorithm to find Non-Free-Choice constructs which represents the global dependencies among tasks[14]. In [15], Gianluigi Greco et al, proposed two algorithms to discover frequent patterns of workflow execution, their approach borrows the idea of frequent item set mining[3] to find sub components of workflow which are frequently appear in the successful executions and then constructs the workflow model by connecting these components. Other work about process mining focus on applying genetic data mining approaches, like data warehouse and machine learning, to workflow event log[16],[17],[18],[10].

6.2 Process Scheduling

Scheduling, however, despite its successful application in manufacturing fields, is not widely accepted in workflow. Grego'rio Baggio Tramontina et al discussed some of the problems that prevent existing scheduling techniques from being used in workflow [19], in addition, they proposed a "Gauss and Solve" scheduling approach to utilize traditional scheduling approaches. Carlo Combi et al focus on temporalities in the conceptual organizational model and task assignment policies. They proposed a temporal organizational model to describe different temporal constrains of resources, and they designed a scheduling algorithm, which evaluates the priority of tasks according to the expected deadline for completion and expected duration[20]. Eder et al's work focuses on personal scheduling. They changed their objective of scheduling from ordering cases in workflow system to assisting individual workflow participants. To meet this end, they provide workflow participants information about upcoming tasks so that they can proactively take measures to prepare for those tasks[21] . This idea is also represented in Liu JianXun et al's work, they observe that many tasks in the workflow can be prepared before they are actually dispatched, so they proposed an agent based framework to model such tasks in the workflow definition [22]. Other work about workflow scheduling concerns scheduling in a single workflow instance, Pinar Senkul and Ismail H. Toroslu proposed a architecture which provides a specification language that can model resource information and resource allocation constraints, particularly, they use constraint programming to schedule workflows with resource allocation constraints[23].

7 Summary

In this paper, we have presented an approach to improve the task dispatching mechanism of workflow management system. Our approach uses a Boolean association rule mining algorithm that is applied to workflow event log to discover frequent parallel executed activities. In the data sets of three vehicle manufacturing enterprises, we have been able to discover 32 strong rules of 40 workflow activities. In addition to presenting our approach and results, we have presented our discussion on possible extension of our approach.

We believe that our approach shows some promise for improving the current state of task assignment mechanism in workflow management systems. Our future plan includes an empirical study of usability of our mined rules. An investigation on other kind of association patterns in workflow related data and experiment of other analytical techniques on temporal characteristic of workflow.

References

1. N. Russell, A.H.M.t. Hofstede, D. Edmond, and W.M.P.v.d. Aalst, Workflow Resource Patterns. (2005), Eindhoven University of Technology: Eindhoven.
2. M. Fabian, Algorithms for Time Series Knowledge Mining, in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. (2006), ACM Press: Philadelphia, PA, USA.

3. H. Jiawei and M. Kamber, Data Mining : Concepts and Techniques (2001), San Francisco: Morgan Kaufmann. 550.
4. C. Borgelt. Efficient Implementations of Apriori and Eclat. in 1st Workshop of Frequent Item Set Mining Implementations (FIMI 2003). (2003). Melbourne, FL, USA.
5. F.A. James, Maintaining Knowledge About Temporal Intervals. Commun. ACM, (1983). 26(11): p. 832-843.
6. R. Agrawal, D. Gunopulos, and F. Leymann. Mining Process Models from Workflow Logs. in International Conference on Extending Database Technology(EDBT). (1998).
7. A. K. A. de Medeiros, W.M.P.v.d. Aalst, and A. Weijters, Workflow Mining: Current Status and Future Directions, in On the Move to Meaningful Internet Systems 2003: Coopis, Doa, and Odbase. (2003). p. 389-406.
8. W. M. P.v.d. Aalst, B.v. Dongen, J. Herbst, L.G.S. Maruster, and A. Weijters, Workflow Mining: A Survery of Issues and Approaches. Journal of Data and Knowledge Engineering, (2003). 47: p. 237-267.
9. W. M. P.v.d. Aalst and A. Weijters, Process Mining: A Research Agenda. Computers in Industry, (2004). 53(3): p. 231-244.
10. M.Z. Muehlen, Organizational Management in Workflow Applications – Issues and Perspectives. Information Technology and Management (2004). 5(3-4): p. 271 - 291.
11. W. M. P.v.d. Aalst, H.A. Reijers, and M. Song, Discovering Social Networks from Event Logs. Computer Supported Cooperative Work (CSCW) (2005). 14(6): p. 549-593.
12. J. Cook and A. Wolf, Discovering Models of Software Process from Event-Based Data. ACM Transactions on Software Engineering and Methodology, (1998). 7(3).
13. W. M. P.v.d. Aalst, T. Weijters, and L. Maruster, Workflow Mining: Discovering Process Models from Event Logs. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, (2004). 16(9): p. 1128-1142.
14. W. Lijie, W.M.P.v.d. Aalst, J. Wang, and J. Sun, Mining Process Models with Non-Free-Choice Constructs(to Be Published). Data & Knowledge Engineering, (2007).
15. G. Greco, A. Guzzo, G. Manco, and D. Sacca, Mining and Reasoning on Workflows. Ieee Transactions on Knowledge and Data Engineering, (2005). 17(4): p. 519-534.
16. G. Daniela, C. Fabio, C. Malu, D. Umeshwar, S. Mehmet, and S. Ming-Chien, Business Process Intelligence. Comput. Ind., (2004). 53(3): p. 321-343.
17. J. Eder, G.E. Olivotto, and W. Gruber, A Data Warehouse for Workflow Logs, in Engineering and Deployment of Cooperative Information Systems, Proceedings. (2002). p. 1-15.
18. M.z. Muehlen, Workflow-Based Process Monitoring and Controlling-Or: What Can You Measure You Can Control, in Workflow Handbook 2001 Workflow Management Coalition, L.Fischer, Editor. (2000), Lighthouse Point,. p. 61-67.
19. Greg, B. rio, W. Jacques, and E. Clarence, Applying Scheduling Techniques to Minimize the Number of Late Jobs in Workflow Systems, in Proceedings of the 2004 ACM symposium on Applied computing. (2004), ACM Press: Nicosia, Cyprus.
20. C. Combi and G. Pozzi. Task Scheduling for a Temporal workflow Management System. in Thirteenth International Symposium on Temporal Representation and Reasoning, Time'06. (2006).
21. J. Eder, P. Horst, G. Wolfgang, and N. Michael. Personal Schedules for Workflow Systems. in Proceedings on Business Process Management: International Conference, BPM 2003, Eindhoven, The Netherlands, June 26-27, 2003. (2003).
22. J. X. Liu, S.S. Zhang, J. Cao, and J.M. Hu, An Agent Enhanced Framework to Support Pre-Dispatching of Tasks in Workflow Management Systems, in Engineering and Deployment of Cooperative Information Systems, Proceedings. (2002). p. 80-89.
23. P. Senkul and I. H. Toroslu, An Architecture for Workflow Scheduling under Resource Allocation Constraints. Information Systems, (2005). 30(5): p. 399-422.