A Machine Learning Approach to Semi-Automating Workflow Staff Assignment

Liu Yingbo
Department of Computer Science
Tsinghua University
Room 408, Building 12 Tsinghua University, Beijing, China 100084
86-10-62776738
lyb01@mails.tsinghua.edu.cn

Wang Jianmin
School of Software, Tsinghua University
Tsinghua University, Beijing, China 100084
86-10-62781776
jimwang@tsinghua.edu.cn

Sun Jiaguang
School of Software, Tsinghua University
Tsinghua University, Beijing, China 100084
86-10-62781776
sunjg@tsinghua.edu.cn

ABSTRACT*

Staff assignment is an important aspect of workflow resource management. In many current workflow applications, staff assignment is still performed manually by resource assigners like process initiator or process monitor. In this paper, we present a semi-automated approach intended to ease the burden of staff assigner. Our approach applies a machine learning algorithm to workflow event log to learn various kinds of activities each actor undertakes. When a new process is initiated, the classifiers generated by the machine learning technique suggest a suitable actor to undertake the specified activities. With this approach, we have achieved an average prediction accuracy of 85.8% and 80.1% on two car manufacturing enterprises respectively. We report on the result of our experiment and discuss issues and improvement of our approach.

Categories and Subject Descriptors

H.4.1 [**Office Automation**]: Workflow management

General Terms

Management, Experimentation, Human Factors

Keywords

Staff Assignment, Resource Management, Workflow, Machine Learning

1. INTRODUCTION

Workflow resources, especially human resources play an important role in successful application of workflow technology. However, whilst the activities are automated by modern workflow management systems, staff assignment design is still undeveloped in many workflow projects [11].

* This work is supported by the Project of National Natural Science Foundation of China (No. 60373011) and the 973 Project of China (No.2002CB312006).

SAC'07, March 11-15, 2007, Seoul, Korea.

Nowadays, the task of finding appropriate actor for each activity in workflow applications is still performed manually by staff assigners. The quality of assignment strongly depends on individual assigner's insights and experience. However, in order to prevent unnecessary expenses caused by unsuitable staff assignment, assigners, even those experienced assigners, are required to be very careful on making the decision of assignment. When business pressure becomes higher and higher, assigners will be burdened by the rate at which new processes are initiated.

Consider, an example of the enterprises we investigated, within a period of 31 months (from Oct-31-2003 to Jun-06-2006) there are 8,612 business processes completed, and 147 actors from 28 different organizations involved in the execution of these processes, averaging 96 activities per day that were performed by different actors. Therefore, quick and correct decision on staff assignment is critical to successful accomplishment of its business objectives.

As a means of reducing the time to execute the process, we present a semi-automated approach to help staff assigner like process initiator or monitor to find the right person for the target activity. Our approach uses a machine learning algorithm to recommend to an assigner one actor who may be appropriate for undertaking the activity. This information can help the assigner in two ways: it may allow the assigner to initiate a process more efficiently, and it may allow the assigner with less overall knowledge of the enterprise to perform assignments more correctly.

Our approach requires an enterprise's workflow system to have had an event log for some period of time and the workflow models from which the patterns of who executes what kinds of activities can be learned. We believe this information is generally available for most of current workflow management systems. Using our approach, in one car manufacturing enterprise, we have been able to correctly suggest appropriate actors to undertake the activities with an average prediction accuracy of 85.8%. We have also applied our approach on another car manufacturing enterprise. In their data set, we reached an average prediction accuracy of 80.1%

This paper makes two contributions: firstly, it presents an approach for helping to automate workflow staff assignment in workflow management systems; secondly, it evaluates the applicability of a machine learning approach for staff assignment using real world data sets from two enterprises.

Table 1 Overview of two enterprises' workflow history data

	Start Date	End Date	Workflow Models	Activities	Workflow Instances	Event Entries	Actors	Related Organizations
A	May-30-2005	Jul-26-2006	24	399	4,005	42,099	244	29
B	Oct-31-2003	Jun-06-2006	49	922	8,612	99,765	147	28

The paper is organized as follow: we begin by presenting background information about workflow structure and event log, general workflow history information from two enterprises (Section 2). Given this background, we describe our semi-automated approach to staff assignment (Section 3) and present the results of applying our approach on real data set (Section 4). We then discuss issues related to this approach and possible improvements (Section 5). In the following discussion, related efforts that attempt to automate part of the resource assignment in workflow management systems are presented (Section 6). Finally, we summarize the paper (Section 7).

2. BACKGROUND

Understanding our approach requires a basic knowledge of workflow. These concepts will be covered in this section. In addition, we provide an overview of current staff assignment in two enterprises and some information of their workflow history data.

2.1 Workflow Structure and Event Log

We first present a set of definitions that we will use throughout this paper.

A *workflow* or *workflow model* is a description of a business process in sufficient detail that it is able to be directly executed by a workflow management system. A workflow is composed of a number of *activities* or *tasks*, which are connected in the form of a directed graph. An executing instance of a workflow is called *workflow instance* or *case*. There may be multiple instance of a particular workflow running simultaneously, however each of these is assumed to have an independent existence and they typically execute without reference to each other[17].

In the discussion of this paper, we treat activities in a workflow as a single unit of work, which will be undertaken by some *actors*. Each invocation of an activity that executes is termed a *work item*. In general, a work item is directed to an *actor* for execution. Once the actor commits the work item, corresponding activity will be marked as completed and other activities will be invoked, mean while, an *event entry* is created to log the actor's operation, for example, work item's accept/commit time and actor's identity and workflow instance etc. These event entries form a workflow system's *event log*.

2.2 Current Staff Assignment in Enterprises

In previous section, we outlined basic concepts of workflow management. As a further introduction to the background, we provide information about two enterprises. Both of these two enterprises are car manufacturing enterprises, we investigate them because workflow is successfully applied in many aspects of their business, like: configuration change, order processing, design review, technical notification, standard release, and new material classification etc.

The staff assignment strategy are quite similar in these two enterprises: anyone who wants to initiate a business process may pick up a workflow model first, and manually assign a concrete actor for each activity. After the workflow instance has been created, the process monitor can modify the original assignment of those unexecuted activities. Since the staff assignment is performed manually, process initiator and process monitor, are not only required to be familiar with the businesses but also the organizational structure of enterprise.

Table 1 is a general overview of workflow history information in these two enterprises. In order to maintain confidential, we use A and B to represent them. As illustrated in the table, workflow has been heavily used in these two enterprises and there are many actors from different organization that is involved in the execution of these workflows. Next, we present our semi-automated approach in detail.

3. A Semi-Automated Approach to Workflow Staff Assignment

Our approach of semi-automatic staff assignment is based on machine learning, its rationale can be described as follows: for a given activity, each event entry of this activity can be viewed as a training sample (or instance) and the event entries of those completed activities in the same workflow instance can be viewed as this training sample's features. The training sample may have a label that indicates the actor who undertook the activity. A supervised machine learning algorithm takes as input a set of training samples with known labels and generates a classifier. The generated classifier can then be used to assign a label to an unknown sample, which, in the context of workflow, is an unassigned activity instance. The process of creating a classifier from a set of instances is known as training the classifier.

This approach is semi-automated because the assigner must decide whether the suggested actor is the actual actor to whom the activity will be assigned, and he or she may make this choice based on knowledge other than that available in the event log, such as the workloads of actors, or who is on vacation etc.

In order to train a classifier, we take following four steps:

Step 1: Selecting target activities
Step 2: Determining features from workflow model
Step 3: Constructing training set from event log
Step 4: Applying machine learning to obtain a classifier

As is typical in machine learning, we evaluate the performance of each classifier using k-fold cross-validation. For a given training set, we separate them into k mutually exclusive subset with approximately equal size, and iteratively test each subset using classifier trained by the remaining k-1 subsets. The accuracy is defined by the average ratio of appropriate assignment for k iterations. In real situation, k is selected according to the characteristics of training set. In general, 10-fold is recommended for estimating the accuracy due to its relatively low bias and variance. While, in this paper, we let k = 3, because, in our case, the size between training sets varies widely, some of them with more than 1000 samples while some of them with less than 100 samples, so let k = 3 is more suitable for our experiment.

3.1 Selecting Target Activities

For a set of workflow models in an enterprise, it is not necessary to predict the suitable actor for all activities. Firstly, if an activity can only be performed by very few actors (for example, less than 3 actors), the assignment rule would be so obvious that can be easily derived by assigners themselves, thus it is not necessary to train a classifier for such kind of activities. Besides, if an activity is executed for relatively very few times, like those activities which are designed to handle cases' special conditions, then the size of training set would be very small, and the accuracy of classifier would be unacceptable. Therefore, those activities, which are suitable for learning, need to be executed for many times and by many different actors.

In fact, according to our investigation, there is a very small portion of activities which is suitable for leaning. Table 2 is a statistics of activity count according to their historical actor number and log entry number, as is illustrated in the table that most of activity is undertaken by very few actors (1-3) and executed for relatively few times (1-300). After this investigation, we selected 22 activities from enterprise A and 35 activities from enterprise B.

Table 2 Statistics about activities

		Actor Count			Activity Log Entry Count		
Range		1-3	4-10	10+	1-300	301-1000	1000+
Activity Count	A	306	59	34	368	19	12
	B	857	50	15	857	54	11

3.2 Determining Features from Workflow Model

In order to train a classifier for a given activity, we need to find out which event entries are similar to each other, so that typical assignment patterns can be derived by the learning algorithm. The similarity is based on characteristics of completed activities' event entries in the same workflow instance. However, there is a substantial amount of information in those event entries that can be used, like actors' identity, data contained in the workflow and structure information of workflow instance etc., which part of information is selected fundamentally determines the performance of a classifier.

In the context of machine learning, the process of determining suitable information from which patterns can be learned is called feature selection, after the feature has been selected, the training instance can be described by a feature vector. In our approach, we use actor identity information of those completed activities' event entry as features. We make this selection because this part of information is generally available for most of workflow systems, another important reason for us to make this choice is that feedback from enterprise users convinced us that people tends to work together in a workflow instance hence it is more reasonable to base our learning on actors' identity. We formally define the features as follows:

Definition 1 (Workflow Instance's Event Log). For a given workflow, let A be the set of activities defined in this workflow and P a set of actors. $E=A\times P$ is the set of possible event entries (e.g. $<a, p>$ means the execution of a is performed by p). The event log of a workflow instance $c\in E^*$ is a sequence of event entry, that describes the execution of workflow. In the following discussion, we use C to represent all the instances' event log of the given workflow model.

Definition 2 (Feature Activities). For a given activity $a\in A$ we define a function *feature:* $A\rightarrow 2^A$ to represent those activities whose event log that will be used by classifier to determine staff assignment.

$$\forall a \in A,\ feature(a) = \{a_0, a_1, a_2 \ldots a_n\}$$

Note that, there are many ways to determine feature activities, in our experiment, we use those activities in the workflow model that have precedence relationship with the target activity. We find out these precedence activities by first excluding edges in a workflow model that might cause loop execution, then we make the relation on activities to be a directed acyclic graph, thus for any activity its precedence activities set can be determined.

3.3 Constructing Training Set from Event Log

After features have been selected for an activity, each of its event entry can be characterized by a feature vector, and the associated label for this event entry is represented by the actor who undertook this activity, hence the training set can be constructed.

Definition 3 (Actor Projection). for a given workflow instance event log $c\in C$ and activity $a\in A$, actor projection operation π produces a set of actors who has undertaken the activity a in c:

$$\forall a \in A, c \in C, \pi_a(c) = \{p \mid <a, p> \in c\}$$

If an activity is never executed in the workflow instance, we use $\{\phi\}$ as the result of actor projection. For convenience, we use $\pi_{a_0}(c)$ to be the creator of workflow instance.

Definition 4 (Training Sample). For a given activity $a\in A$ and a workflow instance event log $c\in C$, we define all the training sample of c for a as follow:

$$\forall a \in A, c \in C, sample(a,c) = \pi_a(c) \times \prod_{a_i \in feature(a)} \pi_{a_i}(c)$$

The notion $\prod_{a_i \in feature(a)} \pi_{a_i}(c)$ is the Cartesian product of activities' actor set obtained by performing actor projection on the workflow instance.

Finally, the training set of activity a can be constructed by collecting all the workflow instances' training sample.

3.4 Applying Machine Learning to Obtain a Classifier

We use three widely accepted machine learning algorithms to obtain a classifier, the first one is C4.5 decision tree [15], we choose this algorithm because it is proposed by previous research

[10] and it resembles the decision process of assigner. The second one we selected is naïve bayes [8], which is a probabilistic machine learning algorithm, we select this algorithm because it is quite simple to realize in the real world applications and it can provide a reference point for finding more appropriate algorithms. The last algorithm we used is Support Vector Machine (SVM) [14] because it is suitable for sparse training sets. For the purpose of this paper is to demonstrate the applicability of machine learning approach in staff assignment, we use existing tool WEKA [21] to train our classifiers and perform the test.

4. EXPERIMENT AND EVALUATION

To demonstrate how well our approach can be used in real world applications, and to determine which algorithm is more suitable for staff assignment problem, we applied our approach on two enterprises' data sets. Because of page limit, we cannot provide all the detail of each activity in this paper, but we report evaluation of our result.

4.1 Evaluation on Applicability

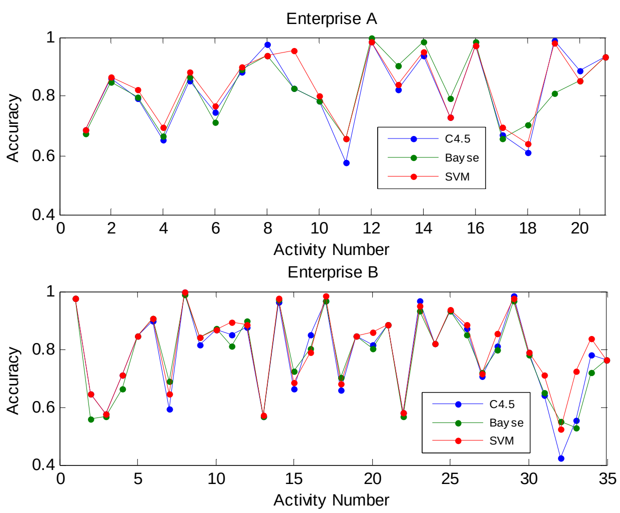

Figure 1 Prediction accuracy of in two enterprises

Figure 1 shows prediction accuracy for 57 selected activities of two enterprises, as is shown in the figure that, the prediction accuracy is not the same for all selected activities, some activities is well predicted by the generated classifier (>90%) but some of them are not good. While almost all classifiers have achieved a prediction accuracy that is greater than 50%, only one classifier trained by C4.5 in enterprise B whose prediction accuracy is less than 50% (42.5%). This classifier's target activity is purchase confirmation activity in enterprise B's car configuration change process, in real business of enterprise B, this activity is conditionally performed. Therefore, the training set of this activity is very small (with only 40 samples). On the other hand, the actors of feature activities are fixed to one or two actors, so they are not distinguishable enough for learning algorithms to train a good classifier.

Table 3 lists the average prediction accuracy of three algorithms. According to table 3, the best prediction accuracy is over 80%, However, we do not think this result can provide sufficient evidence to draw a conclusion that this approach would be acceptable in real situation. The only sure way to know whether our approach can help enterprise staff assigner is to perform an empirical study. Nevertheless, we believe the prediction accuracy we report in this paper is sufficient to warrant such an empirical study.

Table 3 Average prediction accuracy

	C4.5	Naïve Bayes	SVM
Enterprise A	85.41%	86.24%	85.82%
Enterprise B	77.04%	76.64%	80.06%

4.2 Which Algorithm Is More Suitable?

It is hard to decide which algorithm is more suitable for the problem of staff assignment, because the prediction accuracy achieved by three algorithms is quite similar. I our opinion, for the problem of staff assignment, it will not make much sense for algorithms with a prediction accuracy difference that is less than 5%, thus, the general prediction stability will be more important than average prediction accuracy.

Figure 2 is a box plot of classifier's prediction accuracy for three learning algorithms. As is demonstrated in the figure, the distribution density of C4.5 is less than Naïve bays and SVM. This situation means that C4.5 is more sensitive to the quality of training set, therefore its general prediction stability will be less better than the other two. Second, judging from the first quintile of the three algorithms, SVM is slightly higher than the other two, which means general prediction accuracy of SVM will be better than the other two algorithms. Therefore, in the sense of general stability and accuracy, SVM will be more suitable for the problem of staff assignment in these two enterprises.

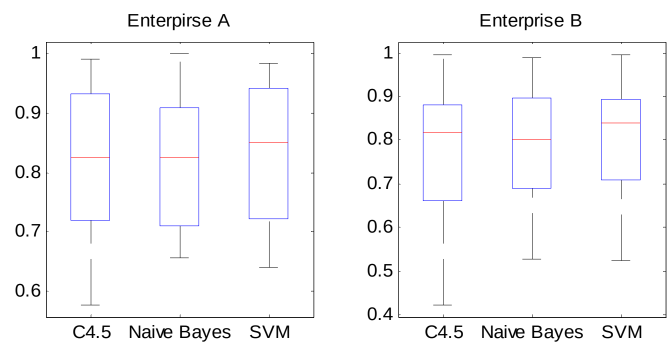

Figure 2 Box plot of accuracy in both enterprises

5. DISCUSSION ON IMPROVEMENT

Although, our approach achieved an average prediction accuracy that is greater than 80%, it is still reasonable for us to believe that there is plenty of space for improvement. In this section, we discuss some possible directions.

Firstly, machine learning algorithms generally produce better results the more data there is available from which to learn. Therefore, the prediction accuracy can be further improved by incorporating other data sources. In our opinion, there are two promising sources: the first one is actor's expertise information and their social relationship[3, 4], the approach presented in [10] is an example of using such kind of information. The other source of information is the data that are going to be processed by workflow, John, A et al's work [7] can be viewed as an example.

For the first source of information, its applicability depends on how well the expertise of actors and their relationship information can be expressed by system, which inevitably leads to the discussion of sophisticated capability model for person and organizations. Although, there are many kinds of these models that are presented in the literature, many of them are actually difficult to implement for current workflow systems, besides, the cost of practicing such kind of model is rather expensive, hence the availability of this part of information will be a big problem.

On the other hand, the data that processed by workflow are another promising source to be incorporated for learning, at least, its availability is much better than expertise information. The barrier that needs to be faced is that, data are usually application context dependent, which means different enterprise's data contains different set of information. Therefore, special approaches are needed in order to extract useful feature information. Sometimes, it might not worthwhile to spend such a cost. Nevertheless, feedback from enterprise user has also revealed the fact that assigners often make their decision based on the documents that is going to be processed by workflow. This fact motivates us to carry out further investigation.

In real situation, workflow event log is not available all at a time, however, the approach we present in this paper trains the classifier using a batched set of data. Alternatively an incremental algorithm could be used whereby instances are provided one at a time to the classifier and the classifier updates itself appropriately[19]. This incremental approach will be more suitable than batched approach especially for those enterprises whose workflow system is intensively used.

Moreover, the approach we presented in this paper only recommend one potential actor once a time, as a matter of fact, groups of actors often work on similar kinds of tasks, therefore, it might be more helpful to recommend a small list of potential actors than just recommend one actor. However, determining the best group of actor is not as easy as it seems to be, further experiment and development of learning approach are needed.

6. RELATED WORK

Our work is related to workflow resource management. Although resource management is very important in workflow applications, most of work in the field of workflow focuses on control and data flow, only a relatively small body of research into the resource aspect is presented [9, 13, 18].

According to our knowledge, the research that is most similar to ours is presented in [10], In their paper, Linh Thao Ly et al. have shown that the problem of deriving staff assignment rules using information from event log data and organizational information as input can be interpreted as an inductive learning problem. Therefore, machine learning techniques can be adapted in order to solve the problem. In particular, they have used decision tree learning on simulated data set to derive meaningful staff assignment rules. Our work is different from theirs in that, our learning is based on actors' identity information of completed activities, which makes our approach novel to automating staff assignment. In addition, we evaluate our approach on real data set using three different learning algorithms, which further proves the applicability of using machine learning in workflow staff assignment

Other research that is related to our work mainly concerns structural characteristics of workflow resource.

In [20, 22], Ming-Chien, Shan et al. presented a resource management facility which is implemented in HP workflow system Changegngine. Their approach consists of an independent resource manager that integrates existing local and site resource management systems under a common schema. The proposed system implements an SQL-like language for the querying and assignment of resources.

In [2], As a part of XRL/Woflan project which aims at the development of an exchangeable workflow modeling language, van der Aalst et al. proposed a set of UML diagrams to specify workflow resource models, this UML diagram can be expressed using XML tags that facilitate the information exchange.

In [12], Michael Zur Muehlen gives a comprehensive overview of organizational aspect about workflow technology in the context of the workflow life cycle, he first presented a generic resource meta model for the initial specification of the resource structure and its population. This meta model combines workflow-oriented with organization-oriented modeling concepts. Subsequently, he outlined variations in the assignment and synchronization policies for the run time allocation of tasks to performers, and presented different strategies for the actual activity assignment at run time. Later, he discussed the maintenance of resource information during the operation of a workflow applications at the macro and micro level. Finally, he presented strategies for the use of workflow event log data to improve resource utilization and development of new process strategies. In the end of the paper, he mentioned the idea of knowledge-based resource allocation algorithms, but he didn't provide any detailed information about the implementation.

Nick Russell et al's work of workflow resource patterns can be viewed as a more comprehensive study about resource management of contemporary workflow products[17, 18], they investigated the resource management mechanism of five workflow products: Staffware，WebSphere MQ Workflow，FLOWer， COSA and iPlanet, and proposed 43 types of workflow resource patterns. These patterns can be classified into six categories: Creation Pattern, Push Pattern, Pull Pattern, Detour Pattern, Auto-start Pattern and Visibility Pattern. All these patterns can be viewed as a guideline for workflow management system construction. Like workflow control pattern and data pattern, their work is also one part of workflow pattern research[1, 16].

In spite of the research about resource allocation in the field of workflow, the idea of automatic resource allocation can also be found in other areas, in [7], A. John et al. proposed a supervised machine learning approach for automating bug fixing assignment in open source development projects. They extracted the feature information and related manual assignment record from historical bug report, and applied three machine learning algorithms to derive the bug assignment rules. In the following experiment, they validated the applicability of their approach in Firefox and Eclipse projects and reached a precision level of 64% and 57%.

Another research about automatic resource allocation is proposed by Carala A. Lima Reis et al[5]. They focus on Process-Centered Software Engineering Environments (PSEEs). In their paper, they described a mechanism to automatically assist software process instantiation through user-defined policies. When the process is instantiated, the resource will be automatically suggested as an ordered list. This mechanism utilizes a meta-model to describe the attributes about resource and software process. The policy is then defined using a language based on this meta-model. After the policy has been defined, a planner component will interpret the policy and suggest an ordered list of resource for each activity. Although, they described the architecture and provided a prototype example, they did not describe how to generate the policies, according to their description, this work is mainly dependents on manual work.

The problem of finding appropriate person is also addressed by the researchers of security field, like RBAC[6] model, but their objective is focused on the authorization and the security related issues[18].

7. SUMMARY

In this paper, we discussed an approach to semi-automating staff assignment in workflow management system. Our approach uses a supervised machine learning algorithm that is applied to workflow event log. In the experiment on two enterprises' data sets, our approach achieved an average prediction accuracy that is greater than 80%. In addition to presenting our results, we have analyzed the performance between different learning algorithms and we have discussed the issues on our approach and possible improvement.

We believe that our approach shows some promise for improving the current state of workflow staff assignment. Our future plans include an investigation of additional sources of information, further experiment on other real data set with improved approach and an empirical study of the use of the approach by enterprise workflow assigners.

8. ACKNOWLEDGMENTS

We are grateful to Tsinghua InfoTech Company for providing the workflow event-log data of their TiPLM system. We also thank Professor Yun Yang's helpful comments on paper.

9. REFERENCE

[1] Aalst, W.M.P.v.d., Hofstede, A.H.M.t., Kiepuszewski, B. and Barros, A.P. Workflow Patterns. *Distributed and Parallel Databases, 14* (1). 5-51.

[2] Aalst, W.M.P.v.d., Kumar, A. and Verbeek, H.M.W., Organizational Modeling in UML and XML in the Context of Workflow Systems. in *18th Annual ACM Symposium on Applied Computing (SAC 2003)*, (Melbourne, Florida USA, 2003), ACM Press, 603-608.

[3] Aalst, W.M.P.v.d., Reijers, H.A. and Song, M. Discovering Social Networks from Event Logs. *Computer Supported Cooperative Work (CSCW) 14* (6). 549-593.

[4] Aalst, W.M.P.v.d. and Song, M., Mining Social Networks: Uncovering Interaction Patterns in Business Processes. in *Business Process Management: Second International Conference, BPM 2004*, (Potsdam, Germany, 2004), 244-260.

[5] Carla, A.L.R., Rodrigo Quites, R., Heribert, S. and Daltro, J.N. A Policy-based Resource Instantiation Mechanism to Automate Software Process Management *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, ACM Press, Ischia, Italy, 2002.

[6] Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R. and Chandramouli, R. Proposed NIST Standard for Role-Based Access Control. *ACM Transactions on Information and System Security, 4* (3). 224-274.

[7] John, A., Lyndon, H. and Gail, C.M. Who Should Fix This Bug? *Proceeding of the 28th international conference on Software engineering*, ACM Press, Shanghai, China, 2006.

[8] John, G.H. and Langley, P., Estimating Continuous Distributions in Bayesian Classifiers. in *Eleventh Conference on Uncertainty in Artificial Intelligence*, (Morgan Kaufmann, San Mateo, 1995), 338-345.

[9] Kumar, A., Aalst, W.M.P.v.d. and Verbeek, H.M.W. Dynamic Work Distribution in Workflow Management Systems: How to Balance Quality and Performance? . *Journal of Management Information Systems, 18* (3). 157-193.

[10] Ly, L., Rinderle, S., Dadam, P. and Reichert, M., Mining Staff Assignment Rules from Event-Based Data. in *Lecture Notes in Computer Science*, Vol.3812 (2006), 177-190.

[11] Moore, C. Common Mistakes in Workflow Implementations, Giga Information Group, Cambridge MA(2002), 2002, 2.

[12] Muehlen, M.Z. Organizational Management in Workflow Applications – Issues and Perspectives. *Information Technology and Management 5*(3-4). 271 - 291.

[13] Pesic, M. and Aalst, W.M.P.v.d., Modeling Work Distribution Mechanisms Using Colored Petri Nets. in *Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2005)*, (Aarhus, Denmark, 2005), 157-176.

[14] Platt, J. Fast Training of Support Vector Machines using Sequential Minimal Optimization. in *Advances in Kernel Methods - Support Vector Learning, B. Schoelkopf, C. Burges, and A. Smola, eds.*, , MIT Press, 1998.

[15] Quinlan, R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA., 1993.

[16] Russell, N., Hofstede, A.H.M.t., Edmond, D. and Aalst, W.M.P.v.d. Workflow Data Patterns, Queensland University of Technology, 2004.

[17] Russell, N., Hofstede, A.H.M.t., Edmond, D. and Aalst, W.M.P.v.d. Workflow Resource Patterns, Eindhoven University of Technology, Eindhoven, 2005.

[18] Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M. and Edmond, D., Workflow Resource Patterns: Identification, Representation and Tool Support. in *Lecture Notes in Computer Science*, (2005), 216-232.

[19] Segal, R. and Kephart., J., Incremental Learning in SwiftFile. in *the Seventh International Conference on Machine Learning*,, (2000), 863–870.

[20] Weimin, D. and Ming-Chien, S., Enterprise Workflow Resource Management. in *HP Labs Technical Reports* (1999), 108-115.

[21] Witten, I.H. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2005.

[22] Yan-Nong, H. and Ming-Chien, S., Policies In a Resource Manager of Workflow Systems: Modeling, Enforcement and Management. in *HP Labs Technical Reports* (1999), 103.