Available online at www.sciencedirect.com

ScienceDirect

Computers in Industry 59 (2008) 463–476

COMPUTERS IN INDUSTRY

www.elsevier.com/locate/compind

A semi-automatic approach for workflow staff assignment

Yingbo Liu [a,b,*], Jianmin Wang [b,c,d,1], Yun Yang [e,2], Jiaguang Sun [b,c,d,1]

[a] *Department of Computer Science, Tsinghua University, Beijing 100084, China*
[b] *School of Software, Tsinghua University, Beijing 100084, China*
[c] *Key Laboratory for Information System Security, Ministry of Education, China*
[d] *Tsinghua National Laboratory for Information Science and Technology, Beijing, China*
[e] *Swinburne CITR—Centre for Information Technology Research, Faculty of ICT, Swinburne University of Technology, Australia*

Received 8 April 2007; received in revised form 27 November 2007; accepted 17 December 2007
Available online 19 February 2008

Abstract

Staff assignment is of great importance for workflow management systems. In many workflow applications, staff assignment is still performed manually. In this paper, we present a semi-automatic approach intended to reduce the number of manual staff assignment. Our approach applies a machine learning algorithm to the workflow event log to learn various kinds of activities that each actor undertakes. When staff assignment is needed, the classifiers generated by the machine learning technique suggest a suitable actor to undertake the specified activities. With experiments on three enterprises, our approach achieved a fairly accurate recommendation.

Keywords: Staff assignment; Resource management; Workflow; Machine learning

1. Introduction

In the context of workflow, staff assignment serves for the purpose of specifying the relationship between activities and resources [1]. It ensures that the operation of a workflow conforms to its intended design principles and operates as efficiently and deterministically as possible. In most cases, staff assignment is performed at workflow build-time stage to restrict the range of resources that can undertake an activity, usually by means of the ‘‘role’’ concept. At run-time stage, workflow engine automatically assigns the work to all the resources with that role or to specific resource using some simple mechanisms such as queue lengths or round-robin, etc. [2]. However, in many real situations, such simple run-time work allocation mechanisms are not sufficient for organizations to correctly assign work to resources.

Consider, for example, an engineering design process in a car manufacturing enterprise we investigated, a typical part design activity is defined to be undertaken by the resources who belong to the designer role, but the actual designer who is responsible for the design can only be specified when a concrete requirement arrives. Because this task is directly related to the manufacturing and eventually determines the quality of the final product, it is unlikely to let designers arbitrarily accept the work items. Therefore, run-time staff assignment is needed. Usually it is performed manually by workflow initiators or monitors. To our knowledge, such manual staff assignment occurs frequently in manufacturing enterprises, especially for those important tasks in business processes.

Although there are various mechanisms of staff assignment in the literature of workflow [2–5], few of them focus on actively recommending resources at run-time, especially using the workflow history information. In Ref. [4], Muehlen envisioned that the workflow history information could be used to improve workflow run-time resource allocation. Russell et al. introduced a pattern (R-HBA), which offers or allocates work items to resources on the basis of their previous execution history, but none of the investigated workflow systems provide

* Corresponding author at: School of Software, Tsinghua University, Zijing, No. 15 Building, 1327A Room, Beijing 100084, China. Tel.: +86 10 51537184; fax: +86 10 62773417.

E-mail addresses: lyb01@mails.tsinghua.edu.cn (Y. Liu), jimwang@tsinghua.edu.cn (J. Wang), yyang@it.swin.edu.au (Y. Yang), sunjg@tsinghua.edu.c (J. Sun).

[1] Tel.: +86 10 62781776; fax: +86 10 62781776.
[2] Tel.: +61 3 92148752; fax: +61 3 98190823.

0166-3615/$ – see front matter
doi:10.1016/j.compind.2007.12.002

direct support for this pattern [2]. Ly and Rinderle et al. proposed a method to derive staff assignment rules from event logs that mainly aims at facilitating the assignment at build-time stage [5,6].

In this paper, we present an approach intended to reduce the amount of manual staff assignment performed at workflow run-time instantiation and execution stages. Our approach applies a machine learning algorithm to the workflow event log in order to learn various kinds of activities that each actor undertakes. When staff assignment is needed, the classifiers generated by the machine learning technique suggest a suitable actor to undertake the specified activities.

Our approach requires an enterprise's workflow system to have had an event log for some period of time and the corresponding workflow models, so that the patterns of who executes what kinds of activities can be learned. Using our approach, we have been able to correctly suggest appropriate actors to undertake the activities with overall prediction accuracies of 82.88%, 79.48% and 79.44%, respectively in three vehicle manufacturing enterprises.

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

This paper is organized as follows: we begin with presenting some background information about the workflow event log and general information about three enterprises (Section 2). Given this background, we describe our semi-automated approach for staff assignment (Section 3) and present the results of applying our approach on real world datasets (Section 4). We then discuss some possible improvements (Section 5). Related works about workflow resources allocation are discussed in Section 6. Finally, we summarize the paper (Section 7).

2. Background

Our work is based on a workflow management system called Tsinghua InfoTech Product Lifecycle Management (TiPLM) workflow. This workflow management system is a part of Tsinghua InfoTech Product Lifecycle Management solution [7], which is a strategic business solution in support of the collaborative creation, management, dissemination, and use of product definition information across the enterprise [8]. By the time this paper is written, TiPLM has been successfully deployed in 38 Chinese manufacturing enterprises.

In this section, we provide a brief introduction of TiPLM workflow as well as some basic concepts of workflow management. Besides, we give an overview on workflow applications and current staff assignment practice in three enterprises.

2.1. Workflow model and event log in TiPLM workflow

In TiPLM workflow, a *workflow* model is described by a directed graph that has four different kinds of nodes:

- *Activity nodes* represent tasks that are undertaken by some actors.
- *Route nodes* represent decision points that determine the execution flow.
- *Start node* denotes the start of a workflow.
- *End node* denotes the end of a workflow.

Nodes are connected by directed links to define different kinds of control flows. There are two kinds of links in TiPLM workflow: normal link and false link. Normal links represent normal execution flows, while false links represent those execution flows when route node conditions are evaluated to be false.

Fig. 1 shows a screenshot of the TiPLM workflow model editor, in which an electronic configuration change process of a car manufacturing enterprises is being edited.[3]

A workflow model can be instantiated multiple times, each instantiation corresponds to an actual execution of the workflow model, which is called a *workflow* instance. In practice, multiple workflow instances may be concurrently active and they execute without referencing to each other.

When an activity node is instantiated at the execution time of a workflow instance, the TiPLM workflow engine reads the staff assignment of this activity, and puts the *work items* into the assigned actors' worklist. Actors then periodically connect to the worklist via a desktop application, pull the work item assigned and execute it. Once the actor commits the work item, the corresponding activity will be marked as completed and its succeeding activities will be instantiated. Meanwhile, an *event entry* is generated to log the actor's operation, e.g. work item's timestamp, actor's identity, workflow instance information, etc. TiPLM workflow's *event log* stores all these event entries.

Fig. 2 shows a worklist handler of TiPLM workflow, through which an actor can monitor workflow instances, execute/commit work items using pop-up menus and watch the future work items assigned in the same workflow instance.

2.2. Workflow applications in three enterprises

In previous section, we have outlined the functionalities of the TiPLM workflow management system. In order to test the validity of our approach, we collected workflow history data from three manufacturing enterprises. The first enterprise is Xiamen KingLong United Automotive Co. Ltd. [9]; the second one is Hebei Zhongxing Automobile Co. Ltd. [10] and the third one is Datong Electronic Locomotive Co. Ltd. [11]. We investigate them because workflow management systems have been successfully used in many aspects of their business, e.g. engineering design and review, release management, notification dissemination, purchase management, customer service, etc.

[3] In order to make it discernable, we repainted the false links using dotted line, however, in real workflow model false links are represented by solid red line.