Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers in Industry 53 (2004) 245–264
COMPUTERS IN INDUSTRY
www.elsevier.com/locate/compind

Workflow mining with **InWoLvE**

Joachim Herbst[a,*], Dimitris Karagiannis[b]

[a]*DaimlerChrysler AG, Postfach 2360, 89013 Ulm, Germany*
[b]*University of Vienna, Vienna, Austria*

Abstract

State of the art information systems are based on explicit process models called workflow models. Experience from industrial practice shows that the definition of workflow models is a very time consuming and error prone task. Recently, there has been an increasing interest in applying techniques from data mining and machine learning to support this task. This approach has also been termed as process or workflow mining. In this paper, we give an overview of the algorithms that were implemented within the **InWoLvE** workflow mining system, we summarize the most important results of their experimental evaluation and we present the experiences that were made in the first industrial application of **InWoLvE**.

Keywords: Workflow mining; Machine learning; Workflow management

1. Introduction

State of the art information systems are based on explicit process models called workflow models. These models are interpreted by one or more workflow engines to drive the execution of business processes within or across several enterprises. Experience from industrial practice shows that the definition of workflow models is a very time consuming and error prone task. In depth knowledge of the business process and the ability to represent this knowledge using a formal workflow modelling language are needed for this task. Recently, there has been an increasing interest in applying techniques from data mining and machine learning to support this task [1–9]. This approach has also been termed as process or workflow mining. The basic idea of the workflow mining approach is to collect traces of workflow executions and to derive a workflow model from these observations. This is useful for example if some information system supporting the process, that logs all relevant events, is already in place before the workflow model is defined. Furthermore workflow mining techniques and advanced workflow technology, which is moving towards more operational flexibility [10–13], enable an evolutionary approach to the development of workflow applications, where an initially roughly defined and informal or semi-formal workflow model is iteratively refined and formalized.

In this paper, we give an overview of the algorithms that were implemented within the **InWoLvE** workflow mining system, we summarize the most important results of their experimental evaluation and we present the experiences that were made in the first industrial application of **InWoLvE**. The remainder of this paper is organized as follows. Section 2 defines the most important terms used throughout this paper. Section 3 formalizes the workflow mining problem, it defines problem classes and it gives an overview of the

* Corresponding author.
E-mail address: joachim.j.herbst@daimlerchrysler.com (J. Herbst).

0166-3615/$ – see front matter
doi:10.1016/j.compind.2003.10.002

induction and transformation algorithms used within the workflow mining system **InWoLvE**. In Section 4, we describe the **InWoLvE** prototype and we summarize the most important results of its experimental evaluation. The experiences we have made in the first industrial application of **InWoLvE** are covered by Section 5. In Section 6, we discuss related work and finally in Section 7 we summarize the main conclusions and give an outlook on our future work.

2. Definitions

2.1. Activities

The basic element of workflow instances, workflow models and stochastic activity graphs are activities. In the following, we use $A = \{a_\triangle, a_0, a_1, \ldots, a_n\}$ for the set of all activities. The activities $a_\triangle$ and a_0 denote special invisible activities that mark the beginning and the end of a model, an instance or a stochastic activity graph (SAG).

2.2. Workflow instance

Definition 2.1 (Workflow instance). A workflow instance is a tuple $e = (K_e, <_e, f_e, k_\triangle, k_0)$, where

- $K_e = \{k_\triangle, k_0, k_1, \ldots k_{n_e}\}$ is a set of nodes (activity instances),
- $<_e \subseteq (K_e \times K_e)$ is a partial order on K_e,
- $f_e : K_e \to A$ is the activity assignment function, and
- $k_\triangle$ and k_0 with $f_e(k_\triangle) = a_\triangle$ and $f_e(k_0) = a_0$ are the starting and end node of e.

We say that an activity a occurs in e if there exists a $k \in K_e$ with $f_e(k) = a$. In the following, we use E to denote a set of workflow instances also called the examples.

2.3. Workflow model

For the description of workflow models we use the ADONIS[1] definition language (ADL). As we focus on the functional and behavioral aspects of workflows we use only a subset of this language. A definition of the complete language can be found in [14].

Definition 2.2 (Workflow model). A workflow model is a tuple $W = (V_W, t_W, f_W, R_W, g_W, P_W)$, where

- $V_W = \{v_1, \ldots v_n\}$ is a set of nodes,
- $t_W(v_i) \in \{\text{START}, \text{ACTIVITY}, \text{DECISION}, SPLIT, JOIN, END\}$ indicates the type of a node,
- V_X for $X \in \{\text{START}, \text{ACTIVITY}, \text{DECISION}, SPLIT, JOIN, END\}$ denotes the subset $V_X \subseteq V_W$ of all nodes of type X,
- $f_W : V_{\text{ACTIVITY}} \to A$ is the activity assignment function,
- $R_W \subseteq (V_W \times V_W)$ is a set of edges (the successor relation),
- $g_W : R_W \to \text{COND}$ are transition conditions, and
- $P_W : R_W \to [0, 1]$ are transition probabilities.

Fig. 1 explains the different node types and shows their graphical representation. By stating an equivalent fragment of a petri net for each node type, it also provides a rough explanation of the semantics of ADL. A complete mapping from a workflow model in ADL to a petri net and thus a complete definition of the semantics is given in [7]. To guarantee a well-defined semantics a workflow model in ADL must obey a number of syntactical rules (e.g. unique start and end, properly nested splits and joins, ...). A complete description of the syntactical rules is also provided in [7].

2.4. Stochastic activity graph

Our workflow mining algorithm uses an intermediate representation for workflow models, which we call a stochastic activity graph.

Definition 2.3 (Stochastic activity graph (SAG)). A stochastic activity graph is a tuple $G = (V_G, R_G, A, f_G, P_G, v_\triangle, v_0)$, where V_G is a set of nodes, $R_G \subseteq V_G \times V_G$ is the set of edges, $f_G : V_G \to A$ is an activity assignment function, $P_G = \{p_v : \mathcal{P}(\text{out}_G(v)) \to [0, 1] | v \in V_G \backslash \{v_0\}\}$ is a set of transition probabilities and $v_\triangle \in V_G$ ($v_0 \in V_G$) with $f_G(v_\triangle) = a_\triangle$ ($f_G(v_0) = a_0$) represents the start (end) node. The set of nodes having an incoming edge from v is denoted by $\text{out}_G(v) = \{u | (v, u) \in R_G\}$ and for every $p_v \in P_G$ the following must hold: $\sum_{\text{out} \subseteq \mathcal{P}(\text{out}_G(v))} p_v(\text{out}) = 1$. Additionally each node must be reachable from the

[1] ADONIS is a registered trademark by BOC GmbH, Vienna, Austria.