



Preference-Based Resource and Task Allocation in Business Process Automation

Reihaneh Bidar^(✉), Arthur ter Hofstede, Renuka Sindhgatta,
and Chun Ouyang

Queensland University of Technology, Brisbane, Australia
{r.bidar,a.terhofstede,renuka.sr,c.ouyang}@qut.edu.au

Abstract. Preference plays an important role in organisational and human decision making as it may be a manifestation of proven practices or of individual working styles. The significance of the notion of preference has been recognised in a number of different disciplines. Unfortunately its potential does not seem to have been fully unlocked in the field of Business Process Automation (BPA), even though resource and task allocations play a pivotal role in process performance and these allocations could be guided by explicit formulations of preferences. In this paper, we examine the state of the art with respect to preference in the field of BPA and use this as the basis for a conceptual model capturing recognised manifestations of preference in the literature. We investigate how preferences may exhibit themselves in process automation through the notion of well-established (workflow) resource patterns. We then show that manifestations of preference may occur in real-life process event logs and how these can be extracted through the application of machine learning techniques. The findings from this research contribute towards establishing a rich understanding of preferences in the context of business processes, ways of specifying and deriving these preferences, and their more explicit incorporation in work allocation mechanisms, which can lead to a step change for realising better process performance and more effective work collaboration in today's organisations.

Keywords: Preference · Business Process Automation · Resource patterns · Resource allocation · Process mining

1 Introduction

In the context of Business Process Management (BPM), a resource is an entity that can perform an activity, either alone or in collaboration with other resources, including humans, software applications and cyber-physical systems (such as robots). Resources are requested at run-time to perform a specific activity towards the objective of a particular process instance [6, 21]. Human resources can exhibit a variety of behaviours, depending on “their attentiveness in the task,

nature of the task and other personal preferences” [29]. This dynamic behaviour of human resources (or resources for short) can affect their performance and the process differently, while this does not hold for non-human resources [26].

From a future work perspective, the forces of technological advancement and resource empowerment are transforming the nature of work conducted and are compelling organisations to redesign their systems to consider resource autonomy and empowerment [12]. Resources are both more productive and more motivated when they have some degree of control over their work [24]. When designing (or redesigning) business processes, resource allocation is crucial for resource performance [20]. Current resource allocation strategies in BPM systems only consider general organisational information such as the role of a resource [6], and other resource attributes (e.g. experience or workload) are overlooked [30]. It is suggested that focusing on these other resource attributes will lead to the improvement of resource allocation and thus process performance [30].

An important resource attribute that has not received a lot of attention in the BPM literature is the notion of *preference*. The notion of preference has been captured in different areas such as task recommendation in crowdsourcing [31] and AI and decision making systems [4]. However, resource preferences have not been considered in a business process automation (BPA) context, an area which is uniquely positioned by the fact that resources are all identified, may play certain roles and be part of an organisational hierarchy, and participate through well-defined allocation rules in the performance of clearly delineated tasks. The focus of this paper is the forms that preferences of resources in a BPA context can take and how preferences can be derived from event logs, i.e. the process execution history. Preferences naturally present themselves in the conduct of work in the form of proven practices, established working relationships, or working styles suited to particular individuals. Resources have different preferences, which may affect their motivation [9] and overall process outcomes in the case of preference for certain activities [23]. Thus, understanding resource preference is important for managing task and resource allocation.

Preference has been defined in different ways in the BPM literature. Sohail, Dominic, and Shahzad [29] define preference as an inclination of human resources to use a non-human resource for executing the assigned task. Lee [19] on the other hand, only focuses on the preference of resources for tasks and defines resource preference as “the property that the resource likes to carry out some tasks more than others”. We take a broader view on preference and we define preference as the degree to which a resource has a tendency for choosing particular types of work or for involving particular resources in the conduct of work.

The objective of this paper is to advance the state of the art in the field of BPM by examining the notion of preference in more detail and to set the stage for unlocking the potential of this notion in business process automation. This is achieved by exploring the current literature and formally capturing notions of preference that are retrieved in the form of a conceptual data model. Patterns related to the resource perspective in business process automation are then examined for the (implicit or explicit) presence of preferences. Preferences can also

manifest themselves in execution logs and it is shown how some specific forms of preference can be discovered as this opens up the possibility of automatically detecting and updating preferences at runtime.

The key research questions we consider in this paper are RQ1) *What are current manifestations of preference in a BPA context?*, and (RQ2) *How can we derive certain forms of preference from event logs?*. To address the first research question, we develop a conceptual data model of preference which is informed by existing studies in the field of BPM. We then examine well-known resource patterns to determine to what degree they encapsulate various forms of preference. As these patterns have been used to assess workflow management systems, this also gives us an idea how preferences may be reflected in these systems. To address the second research question, we use machine learning applied to real-life publicly available event logs to show how certain forms of preference can be automatically derived and what factors in the logs may contribute to higher accuracy.

The main contributions of this paper are as follows: (i) synthesis of a rich notion of preference through the provision of a conceptual model of preference (preferences may be classified as resource-task, resource-resource, and task-resource), (ii) detection of implicit preferences by looking at what is encoded in resource patterns, which provides an indication of what workflow management systems offer, and (iii) an approach to derivation of preferences from event logs based on machine learning which can be used to guide resource allocation and task selection in business process automation.

The paper is organised as follows. A brief literature review (Sect. 2) is followed by a formal representation of preferences in the form of two ORM [13] models (Sect. 3). Manifestations of preferences in workflow resource patterns (Sect. 4) and real-life event logs (Sect. 5) are subsequently investigated. The paper concludes with a brief summary and avenues for future work.

2 Related Work

It has been stated that “[p]reference is inherently a multidisciplinary topic” [17] and “[t]he expression of preference by means of choice and decision making is the essence of intelligent, purposeful behavior” [27]. Preference has been found to be a fundamental attribute for decision making by agents and for supporting the decisions of users [10]. Preference has been used in an AI context to improve decision-making algorithms [4] and to improve planning [16]. In the context of recommender systems and crowdsourcing, preference is used to model and predict results for alternative options (c.f. Guo et al. [11] and Yuen et al. [31] resp.). The goal of preference-aware interactive systems is to help users perform tasks [22] by providing support for decision making by learning and reasoning over preferences [5]. While the aforementioned work is not targeted at resource allocation in the context of business process automation, the work of Yuen et al. [31] aims to support task selection based on preferences deduced from worker search history and thus illustrates that resource preferences can guide task allocation.

In the field of management, Shaw et al. [25] examined preference and its relationship between task interdependence, reward interdependence and preference for group work regarding performance and satisfaction of individuals. They found that the interaction of task interdependence and preference for group work was positively related to group-member performance. García et al. [8] proposed an ontological model for preference as a solution for discovery and ranking of scenarios in the context of user preferences and semantic web services.

Current research efforts related to the topic of preference in the field of BPM focus on determining the ontology of preference [3] and recognising preference as an underlying factor for resource allocation [26, 30, 32]. Preference is identified as an important criterion in Arias et al. [2] for the purpose of research allocation, but it is not further elaborated upon. Zhao et al. [32] proposed a method to support resource allocation by mining resource characteristics and task(-oriented) preference patterns. Sindhgatta et al. [26] suggested an approach to support resource allocation decisions by extracting information about process performance and process context that takes into account the aspect of preference. Huang et al. [14] proposed a resource allocation mechanism to measure resource behaviour from four perspectives consisting of preference, availability, competence and cooperation, and similarly Sohail et al. [29] found preference to be an important variable that should be measured for resource behaviour along with competencies and suitability.

In addition to the above, some studies considered preference an attribute of resource behaviour. For instance, Lee [19] proposed a resource scheduling method to model resource competence and preference in order to improve performance of workflow management systems and resource utilisation in workflows, while Pika et al. [23] developed a software framework that allows organisations to extract information about some resource characteristics including experience, preferences, and collaboration patterns from process event logs. Furthermore, compatibility of resources for task assignment has been considered [18], but it is different from preference as compatibility focuses on process and team outcome while preference focuses on choices resources make.

Although the existing literature recognises the importance of preference in understanding resource behaviour and, more specifically, the importance of preference in the context of resource allocation in business process automation, establishing a rich notion of preference in this context is largely unexplored in the BPM literature. Our research aims to contribute to removing this gap.

3 Conceptual Modelling

In this section, we design a conceptual model of preference which is informed by the existing studies on the topic of preference in the field of BPM. We use Object Role Modeling (ORM) to design our conceptual model. ORM is a fact-oriented method for modelling information systems at the conceptual level [13].

Findings from the literature review helped synthesise a current notion of preference in the context of business processes. Three typical scenarios can be found

in the existing studies. First, a resource prefers one task to other tasks; second, a resource has preference to work with one specific resource among several resources; and third, when a task is to be allocated a resource, preference is given to one resource among several resources to perform the task. Hence, preference can be expressed as a (directed) relationship between resource and task (referred to as *resource-task preference*), a relationship between two resources (*resource-resource preference*), and a relationship between task and resource (*task-resource preference*).

We present two ORM models to capture the above notions of preference. The model shown in Fig. 1 specifies resource-task preference and resource-resource preference, and the model depicted in Fig. 2 specifies task-resource preference.

In the model of Fig. 1, it can be seen that a resource has a tendency for performing a particular task among some offered alternatives, or for involving certain other resources in the execution of a task. A human resource may prefer a particular type of task to other types of task [19,26,30]. He/she may prefer to use a particular non-human resource (e.g., a tool or instrument) to another non-human resource for executing an assigned task [28,29]. Sohail, Dominic, and Shahzad [29] give the example of a nurse who is assigned to measure blood pressure of a patient and may have a preference for either a clinical mercury monometer or a digital sphygmomanometer as the instrument (both are instances of non-human resources). A resource may also prefer a certain person over another for collaboration purposes [14].

In the model of Fig. 2, it is shown that for the execution of a task human resources with particular characteristics are preferred. In order to offer a task to a particular resource instead of to another resource (e.g., during automated resource allocation), resources can be ranked according to their characteristics [3, 32]. A number of resource characteristics to be considered include skills and specific knowledge, experience, workload, and execution of a certain task in the past [3]. For example, a human resource who has more task completions and longer work experience with a company is preferred over other resources to do a certain task [3].

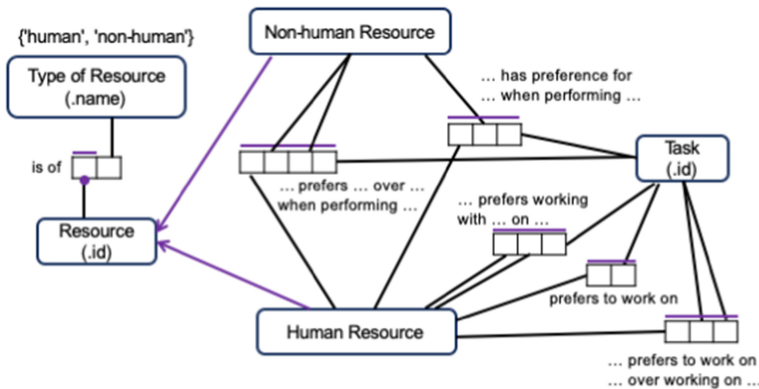


Fig. 1. An ORM model of resource-task preference and resource-resource preference

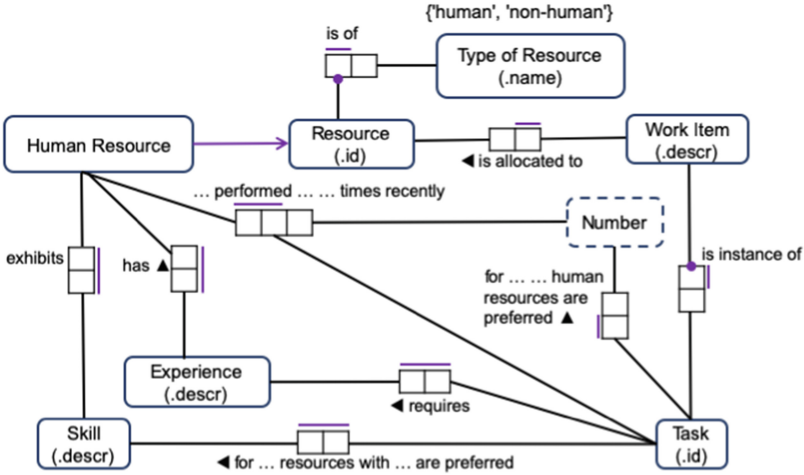


Fig. 2. An ORM model of task-resource preference (with resource characteristics)

4 Resource Patterns Analysis

In this section, we revisit the workflow resource patterns [24] from a preference perspective in order to understand to what extent preferences can be captured or facilitated through the mechanisms implied by these patterns. The workflow resource patterns (or resource patterns for short) were defined to provide a comparative insight into resource management capabilities of process-aware information systems. They have been used in the evaluation of various tools and languages (e.g. BPMN) and provided insights into their relative strengths and weaknesses and thus into opportunities for future improvements.

We use the conceptual model of preference proposed in Sect. 3 as the basis for the assessment of the resource patterns. Table 1 provides an overview of the assessment results between this notion of preference and the resource patterns. Preferences can be hard coded as part of the process definition at design time (DT), or they can manifest during the process execution at run-time (RT) through the application of a resource pattern. A resource pattern may involve some notion of preference but one that is outside the scope (OS) of our definition. Finally, a pattern is not applicable (NA) for assessment if it focuses on mechanism(s) irrelevant to preference. Below we discuss in more detail the assessment of the resource patterns that belong to the various groups.

Creation Patterns. A preference for a specific resource to perform a certain task can be expressed at design time through the use of the ‘Direct allocation’ pattern. Similarly, the ‘Role-based distribution’ pattern can be used to capture a preference for assigning a task to the resources playing a certain role at design time. We can also decide, at design time, that resource capability (e.g. demonstrated by possession of certain knowledge and skills) is used as a basis for the distribution of certain tasks to resources (i.e. ‘Capability-based distribution’), or that

Table 1. Evaluation of resource patterns from a preference perspective

Nr	Resource patterns	Rate	Nr	Resource patterns	Rate
Creation Patterns			Push Patterns		
1	Direct allocation	DT	12	Dist. by offer - single resource	NA
2	Role-based distribution	DT	13	Dist. by offer - multiple resources	NA
3	Deferred distribution	NA	14	Dist. by allocation - single resource	NA
4	Authorization	OS	15	Random allocation	NA
5	Separation of duties	NA	16	Round robin allocation	NA
6	Case handling	NA	17	Shortest queue	NA
7	Retain familiar	NA	18	Early distribution	OS
8	Capability-based distribution	DT	19	Distribution on enablement	NA
9	History-based distribution	DT	20	Late Distribution	OS
10	Organizational distribution	DT			
11	Automatic execution	NA			
Pull Patterns			Detour Patterns		
21	Resource-init. allocation	RT	27	Delegation	RT
22	Resource-init. exec. - allocated WI	RT	28	Escalation	NA
23	Resource-init. exec. - offered WI	RT	29	Deallocation	NA
24	System-determ. work queue cont	DT	30	Stateful reallocation	RT
25	Resource-determ. work queue cont	RT	31	Stateless reallocation	RT
26	Selection autonomy	RT	32	Suspension/Resumption	NA
			33	Skip	RT
			34	Redo	NA
			35	Pre-do	NA
Auto-start Patterns			Visibility Patterns		
36	Commencement on creation	NA	40	Config. unallocated WI visibility	OS
37	Commencement on allocation	NA	41	Config. allocated WI visibility	OS
38	Piled execution	RT			
39	Chained execution	NA			
Multiple-resource Patterns					
42	Simultaneous execution	OS			
43	Additional resources	RT			

task execution history is to be considered (i.e. ‘History-based Distribution’) for example because a resource has acquired a certain amount of experience with a task. In addition, tasks can be distributed to resources that hold a certain position or that have certain relationships with other resources, and this can be formalised at design time through the use of the ‘Organizational distribution’ pattern.

The ‘Authorization’ pattern is concerned with privileges a resource may hold in terms of what work-related actions the resource is allowed to perform (e.g. delegation or skipping of work). This may be seen as a form of preference in a broad sense, but such preference is outside the scope of our model.

The remaining patterns in the group are not applicable for assessment as they focus on the mechanisms that are irrelevant to preference. For example, the ‘Case handling’ pattern focuses on having the same resource to work through an entire instance of a process regardless of preference.

Push Patterns. All the patterns in this group focus on the existence of a distribution mechanism (e.g. ‘Random allocation’ or ‘Shortest queue’) in a system rather than the ability to select certain specific resources due to preference. However, the ‘Early Distribution’ and ‘Late Distribution’ patterns are also concerned with whether the (predetermined) distribution of a task to a resource is made available at an earlier or a later stage, which might be seen as a weak form of preference and which is not considered in our model.

Pull Patterns. The ‘Resource-initiated allocation’ and ‘Resource-initiated execution’ patterns are concerned with the presence of mechanisms for resources to choose which tasks to commit during process execution and thus can be used to capture resource-task preferences at run-time. Through application of the ‘System-determined work queue content’ pattern one can capture the presentation of worklists (e.g. the order in which tasks are listed), and as such, this pattern can be applied to encapsulating resource preferences at design time (e.g. through the use of certain data attributes). Using the ‘Resource-determined work queue content’ and ‘Selection autonomy’ patterns, resource preferences can be captured that manifest at run-time, specifically their preference for how work is presented to them (which may influence what they choose to work on next) and what tasks to work on next.

Detour Patterns. The ‘Delegation’ pattern can be used to capture that, at run-time, preference may play a role in determining to whom work is delegated. Similarly, the ‘Stateful reallocation’ and ‘Stateless reallocation’ patterns may be applied when work is reallocated. Skipping of tasks by the ‘Skip’ pattern may be a manifestation of resource preference of not wishing to perform certain work at run-time. The remaining patterns are not applicable for assessment (e.g. the ‘Suspension/Resumption’ pattern is to allow a resource to pause/continue with a task that has already started rather than to capture its preference for a task).

Auto-start Patterns. These patterns are not applicable for assessment except for the ‘Piled execution’ pattern which can be used to capture that instances of a certain task should all go to a specific resource upon request by that resource. Piled execution should be enabled at design time, but resource preferences come to the fore at run-time.

Visibility Patterns. Both patterns are out of scope as they are concerned with the means to make work items visible (‘Configurable allocated’ or ‘unallocated’), which may enable certain viewing preferences to be realised.

Multiple-resource Patterns. The ‘Simultaneous execution’ pattern is out of scope as it is concerned with the ability of a resource to work on multiple work items at the same time, which is not a preference in the model defined in this paper. Next, the ‘Additional resources’ pattern constitutes an important preference-related

pattern as it allows multiple resources to be involved in performing a task. The run-time involvement of additional resources can be guided by preference.

Discussion. Insights learned from the above pattern-based analysis are two-fold. First of all, it shows that preferences are pervasive in resource mechanisms in process-aware information systems though their presence is not necessarily explicit. It seems worthwhile to make the role of preference more visible and this could be achieved by considering them an aspect of resource allocation (in the sense of aspect-orientation [15]). Hence any preference-related change can be facilitated through its own aspect.

Secondly, our analysis also reveals that the full richness of preference does not manifest itself in the resource patterns, which seems a consequence of the time in which these patterns were conceived. For example, at the time of conception of the resource patterns, the idea of having multiple resources involved in carrying out a task was relatively novel and only limited support was offered by contemporary process-aware information systems. This explains the existence of only a single pattern with that focus and as our earlier analysis of preferences shows, additional and more sophisticated patterns (and associated mechanisms) can be envisaged. To this end, the emergence of research relating social media and BPM (e.g. [7]) could give rise to additional patterns as could mechanisms offered by modern process-aware information systems.

5 Detecting Preferences in Process Logs

In this section we look at real-life event data and how some forms of preference may manifest themselves and how they may be derived. The derivation of preferences, especially if performed on an ongoing basis as to make sure they are up to date, can produce useful information for work allocation in the context of a business process automation environment. We should note upfront that preferences as they manifest themselves in real-life settings may have different support than what can be found in the literature as a combination of factors could influence these preferences. Hence, we take a broader view on log-derived preferences than what is supported by the literature.

An event log contains a set of events. We assume each event has the following key attributes: *case_id*, *task*, *time_stamp*, *status*, *resource*. The *case_id* captures the case identifier of each unique case, *task* corresponds to the process' activity being performed by *resource* at a given *times_stamp*, and *status* \in {*schedule*, *start*, *complete*} reflects the states in the life cycle of the task where a task is scheduled, started, and completed by a resource (note that other statuses may be recorded in an event log). Each event may further have process-specific attributes such as a loan amount for a loan application process.

The preference of a resource in performing a task or working with another resource can be influenced by factors such as the experience of the resource in performing a task, the number of tasks that need to be completed, and the workload of other resources. Computing preference as a frequency of a resource performing a certain task [14], while indicative may not be sufficient. Figure 3(a)

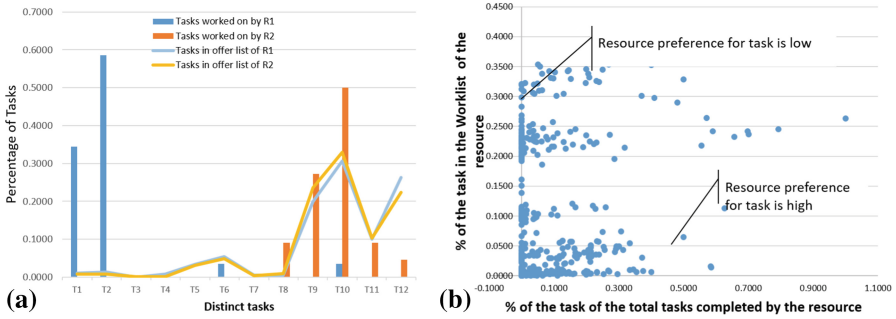


Fig. 3. (a) The percentage of tasks two resources work on and the percentage of tasks in their worklist. (b) Each point in the chart represents the percentage of a task available in the worklist for a resource vs. percentage of that task of the total tasks worked by that resource.

uses a real world event log to illustrate the dependence of resource preference on different factors. The frequency of tasks completed by two resources (R1 and R2) and frequency of tasks in the worklist of the resource is presented. Worklist of a resource contains all tasks that are ‘scheduled’ or not allocated at the time of the allocation of a task to the resource. From the figure, it can be deduced that Resource R1 has a high preference to work on task T2 which is infrequent on her worklist. Resource R2 prefers to work on tasks T9 and T10 which are the most frequent tasks in the worklist of R2. Figure 3(b) is a plot of the frequency of tasks completed by a resource and the frequency of tasks in the worklist of the resource. While one factor influencing resource preference is presented, there can be many such factors. Given the scenario of multiple factors impacting resource preference, our proposed approach (Fig. 4) uses supervised learning to examine manifestations of resource preferences. Supervised learning consists of arriving at a hypothesis by mapping the inputs (or features derived from the event log) to an output class (or label) that is interpretative of the preference of a resource. The assumption is that, if the learned hypothesis predicts the output values for unseen events (test data), then this hypothesis will be a good representation of the resource preference. Two classifiers, each using specific input features and corresponding output classes, are trained to learn two different resource characteristics that indicate a preference that resources may have. K-fold cross-validation is used to train and test the performance of these classifiers.

5.1 Predicting Next Selected Task in a Worklist

The preference of a resource for performing a certain task over other tasks available on its worklist, is explored by training a classifier with a set of input features comprising of the resource, and (work) list of work items (instances of tasks) available to the resource. The output label is the selected task the resource will work on next. As preference may be influenced by other resource characteristics, such as experience of the resource with the task, we add these additional features

when building the model. These input features are computed for all events that are in the log where the status is *start*. The following input features for event e are used:

- **Resource:** As the objective is to learn the preference of a resource, this is an input feature. Binary encoding is used to create a resource feature vector $\mathbf{r} = (r_1, r_2, \dots, r_n)$ corresponding to n resources in the log, where $r_i = 1$ for the resource of event e , and 0 otherwise.
- **Worklist:** There can be one or more work items available on the worklist at the time of event e for the resource for which we are interested in its choice of subsequent task. The worklist of a resource consists of all tasks that are ‘scheduled’ and not ‘started’ prior to the time of event e . The worklist feature vector $\mathbf{w}_{t_e} = (a_1, a_2, \dots, a_m)$, corresponds to m tasks in the log (where t_e refers to the timestamp of event e). Frequency encoding is used where each feature represents the task, and the feature value is the frequency of the number of work items of that task in the worklist at time t_e .
- **Previous owner:** In certain scenarios, a work item of a task is placed back onto the worklist after being worked on by a resource. A feature vector using binary encoding is used to represent the resource that worked on that work item prior to the event (*case.id, task* are used to identify the work item). For newly scheduled or created tasks, this is a zero value vector.
- **Experience:** The number of completions of work items of a task a by a resource r , during a time slot $[t_1, t_2)$ is used as the indicator of experience of r in performing a : $exp(r, a, t_1, t_2)$ [23]. Linear scaling (or min-max normalisation) is used to normalise the experience of resources with a task to the $[0,1]$ range. The experience vector for a resource r_i reflects its experience with all tasks considering a time slot $[t_1, t_2)$, and is given as:
 $ex_{r_i} = (exp(r_i, a_1, t_1, t_2), exp(r_i, a_2, t_1, t_2), \dots, exp(r_i, a_m, t_1, t_2))$.

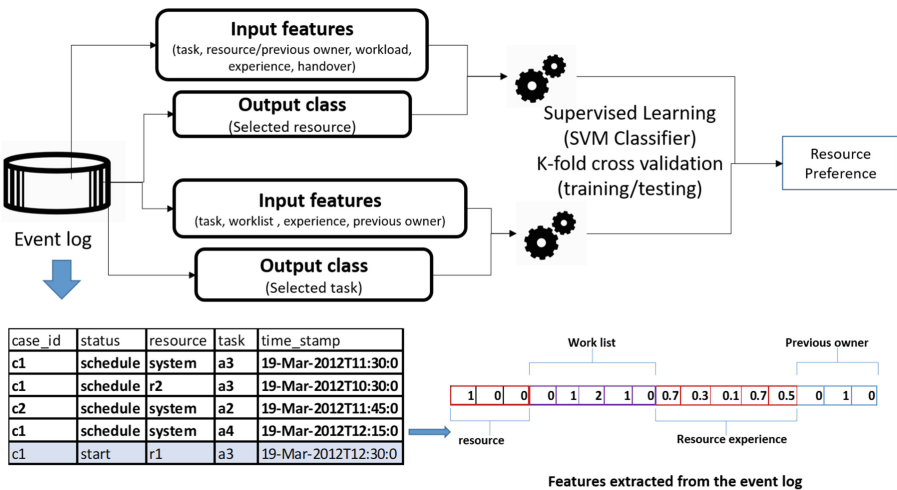


Fig. 4. Approach to learning preferences from event logs.

Figure 4 presents an overview of our approach and includes an example of a feature vector comprising of resource, worklist, resource experience and previous owner features.

5.2 Predicting the Resource for the Subsequent Task

Given a resource and a task that the resource has executed, in this section, we look at predicting which next resource will work on a subsequent task. We consider handover of task preference influenced by factors such as the workload of the resource the task is being handed over to, experience of the resources on the task being handed over, and frequency of handovers of tasks made by the resource to other resources. The preference of a resource for another resource to do upcoming work is explored by training a classifier on a set of input features for all events in the log where the status is *start*. The input features comprise the previous owner and the task itself. The output label is the (selected) resource of the event. Experience of resources with the task and an additional runtime factor of workload of resources are added when building the model. The following input features for event e concerning task a are used:

- **Previous Owner:** This feature represents the resource working on a work item of task a prior to the event. The objective is to learn the preference of this resource. Binary encoding is used to represent this feature as discussed in Sect. 5.1.
- **Task:** Binary encoding is used to create a task feature vector $\mathbf{a} = (a_1, a_2, \dots, a_m)$ corresponding to m tasks in the log, where $a_j = a = 1$ and 0 otherwise.
- **Experience:** The experience vector is computed during a time slot $[t_1, t_2)$ and the feature vector containing experience of n resources with task a is given as:

$$\mathbf{ex}_a = (\exp(r_1, a, t_1, t_2), \exp(r_2, a, t_1, t_2), \dots, \exp(r_n, a, t_1, t_2)).$$
- **Workload:** The workload $wl(r, t)$ is the number of tasks that are not yet completed by resource r at time t of event e . The feature vector for workload consists of the workload of n resources at the time of event e :

$$\mathbf{l}_{t_e} = (wl(r_1, t_e), wl(r_2, t_e), \dots, wl(r_n, t_e)).$$
- **Handover:** The frequency of tasks handed over by a resource r_1 of event e_t to resource r_2 of event e_{t+1} computed during a time slot $[t_1, t_2)$ is used as the resource handover experience $hover(r_1, r_2, t_1, t_2)$ [1]. The handover vector for a resource r_1 is given as:

$$\mathbf{hd}_{r_1} = (hover(r_1, r_1, t_1, t_2), hover(r_1, r_2, t_1, t_2), \dots, hover(r_1, r_n, t_1, t_2)).$$

5.3 Data Sets

The approach is evaluated on two Business Process Intelligence Challenge (BPIC) logs¹. These logs contain resource and task life cycle information required for computing the input features of the model (Table 2).

¹ <https://www.win.tue.nl/bpi/doku.php?id=2012:challenge>, <https://www.win.tue.nl/bpi/doku.php?id=2013:challenge>.

Table 2. Event log statistics

Event log	# events	# cases	# resources	#tasks	# events considered	Case attr. used
BPIC 2012 W	262200	13087	69	7	10704	Loan amount
BPIC 2013 (Incidents)	65533	7554	1472	1	7436	Product, impact

1. **BPIC 2012 log:** From the BPIC 2012 logs we chose the work items log (or BPIC-W 2012 log). The data set contains events for a period of 6 months. The first three months of data is used to compute experience and handover experience of resources. The remaining three months of data is used to train and test the classifier. Tasks are categorised into two bins using the loan amount: (≤ 10000 , > 10000). Two types of ownership changes were considered: (i) events corresponding to newly scheduled work items where the corresponding task never had previous work items in the same case associated with a resource or with events where the scheduled work item had a previous work item of the same task in the same case but it was associated with a different resource, (ii) events corresponding to handovers where the work item of the task is completed by a resource and followed by a work item of the subsequent different task which is performed by a different resource.
2. **BPIC 2013 log:** From the BPIC 2013 logs we chose the BPIC 2013 (Incidents) log (or BPIC-I 2013). The log contains 1472 resources. However, only 40 resources have worked on at least 1% of the cases. As the event log contains only one task, domain attributes are used to further characterise tasks. A high impact ‘PROD424’ incident is considered as a task distinct from a low impact ‘PROD660’ incident. Characterising tasks using the attributes results in 20 distinct tasks.

5.4 Evaluation

For the purpose of evaluation, we experimented with Random Forest Classifier, K-Nearest Neighbour Classifier, and Support Vector Machine (SVM) Classifiers. Based on our experimental results, we chose SVM as it provided better results over the others. Two SVM Classifiers are trained and evaluate using 5-fold cross-validation. Experiments are performed by building SVM models using the Python library Scikit-learn².

Three commonly used performance measures are reported - classification accuracy, macro-averaged F1 and weighted F1 score. Accuracy is the ratio of correctly classified data points to the total number of data points. The F1 score is the harmonic mean of the precision and recall of a classifier. The F1 score is measured for each class and then the average is taken. This measure is known as the macro-averaged F1. The weighted F1 score is computed where the F1 score

² <https://scikit-learn.org/stable/>.

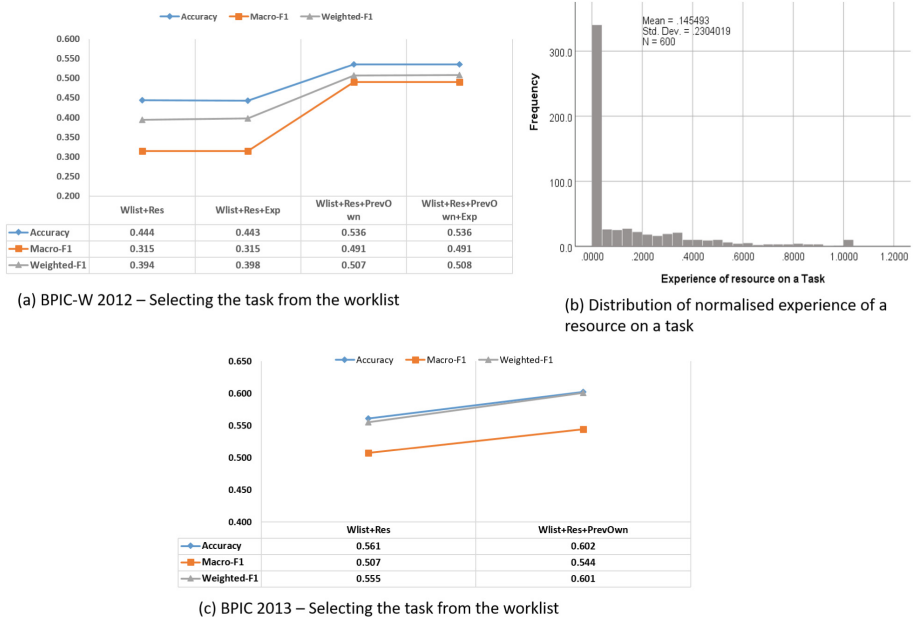


Fig. 5. (a) Model performance for predicting the task from the resource worklist using BPIC-W 2012 event log, (b) Distribution of resource experience computed from BPIC-W 2012 event log, and (c) Model performance using BPIC 2013 event log

is measured for each class and the average is weighted by the number of data points of the class. The number of resources as an output class is lower than the total number of resources in the event log. We trained the model considering classes that had at least 15 data points in the test data set and hence resources with lower numbers of data points were not used.

The performance of the model predicting the next task on a resource’s worklist is evaluated and presented in Fig. 5. In order to gain a detailed understanding of the proposed input features, we show experiments with a number of variants of the input features: (1) resource and its worklist, (2) resource, its worklist and resource experience on each of the tasks, (3) resource, its worklist and the previous owner, and (4) resource, its worklist, previous owner and its experience. For the BPIC-I 2013, the event logs cover a time period of one month. Hence experience of a resource was not included as a feature as the time interval of the log was too small to compute a measure of resource experience. Model performance is low with an accuracy of 0.444 for the BPIC-W 2012 event log when resource and worklist are used as input features (WList+Res in Fig. 5(a)). The model performance does not improve (for BPIC-W 2012) with the inclusion of resource experience (WList+Res+Exp). As resource experience would naturally be an important consideration in task selection, we further investigate the reason for its limited influence. It can be observed that in the BPIC-W 2012 data

set, a large number of resources have very low levels of experience with a limited few having high levels of experience. This skewness in the distribution of resource experience results in its limited influence on the model performance. The model performance improves significantly when the previous owner is considered. Hence, manifestations of preference could consider resource (co-workers) and task in conjunction (WList+Res+PrevOwn). Addition of resource experience yields no improvement in accuracy of Macro-F1. The accuracy is 0.5612 for BPIC-I 2013 and increases to 0.6020 when previous owner is considered, further indicative of resources considering tasks and co-workers together (Fig. 5(c)). Given the high number of output classes (10 tasks in BPIC-W 2012, and 20 tasks in BPIC 2013), the accuracy of 53–60% provides a good indication that the classifier is able to learn the preference function and predict the right task with favourable accuracy.

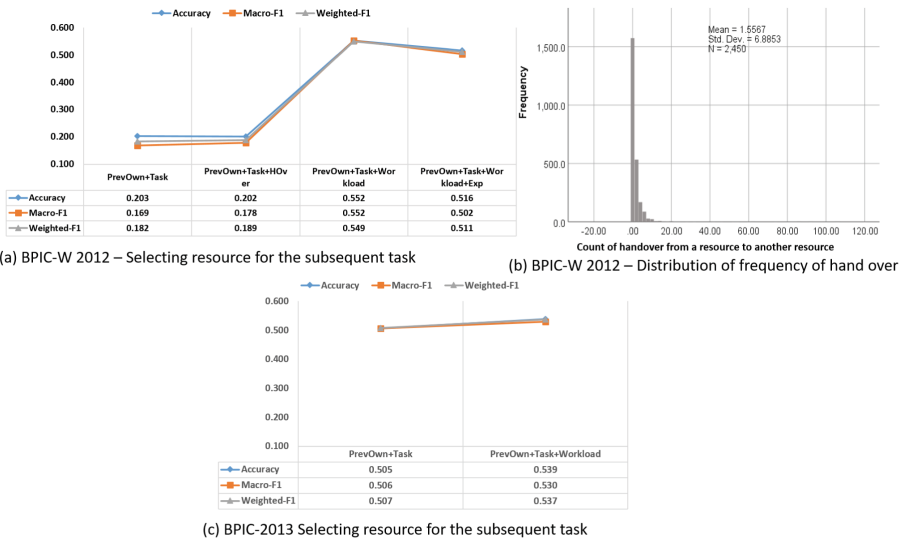


Fig. 6. (a) Model performance for predicting the next resource for BPIC-W 2012, (b) Distribution of past hand overs between resources (c) Model performance for predicting next resource for BPIC-2013

The second model is concerned with predicting who will perform instances of the next task. We show results with variants of the proposed features: (1) previous owner and task, (2) previous owner, task and past handovers made by the previous owner, (3) previous owner, task and workload of the available resources, and (3) previous owner, task, workload and experience of the resources with the task (BPIC-W 2012). Figure 6 presents the results for BPIC-W 2012 and BPIC-2013 for the variants of the features. Here again the accuracy for the BPIC-W 2012 event log is low when considering previous owner and task. Addition of past handovers made by a resource does not cause any improvements in

the model accuracy (PrevOwn+Task+HOver in Fig. 6(a)). The long-tail distribution of handovers made by resources indicates that resources often work with many other resources while a select few have a high value of handovers to specific resources (Fig. 6(b)). Addition of resource workload significantly improves model performance. Addition of experience of resources on the task does not improve the model accuracy, which can be attributed to the distribution of resource experience on the tasks. Hence, model results show that resources consider workload when handing over tasks to other resources. Model performance for the BPIC 2013 log, improves by 3 points (6%) by addition of workload (Fig. 6(c)), confirming the influence of workload on the handover preference of one resource for another.

The model accuracy ranges from 53% to 60% in predicting the subsequent task or the resource indicating that the preference function can be learned from the event logs when information of workload, worklist and handovers are captured. The use of different features provides insights into the factors influencing resource preference such as frequency of the item in the worklist and workload of resources. Using a preference model to predict resource preferences would lead to ‘preferred’ resource and task allocation.

6 Conclusions and Future Work

While it has been recognised that preference can play a prominent role in resource behaviour and that this may guide resource allocation, there is only a small body of work in the BPM literature that touches upon this topic. Preferences can’t take various forms and may influence resource allocation and choice of task in the context of business process automation. In this paper preference has been treated as an explicit concept in business process automation and various manifestations have been shown, some as mentioned in the literature and later formalised in the form of ORM models, some as (implicitly) present in workflow resource patterns and thus as mechanisms in process automation systems, and some as can be found in event logs. In the latter case it was shown how these could be discovered through the use of machine learning. This was illustrated using two real-life logs from the set of BPIC logs.

The present work provides a starting point for examining preference in greater detail in a business process automation context, in terms of the forms it can take, the way these forms can be used to support resource allocation and how they can automatically be derived from event logs in order to keep them up to date. All of these give rise to further work. Ideally, preferences are more explicitly represented in BPM systems so that their influence is more visible and they can be utilised better.

We would like to conclude by acknowledging some limitations. We recognise that one can think off-the-cuff of many forms of preference in the context of resource allocation for business process automation. However, we have consciously refrained from doing so and stuck to what we unearthed from the literature. Also, ideally, more publicly available logs will be made available in

the future containing rich resource information. These logs could expose other forms of preference, provide more insight into the accuracy of the methods we presented in Sect. 5, and even give rise to new automatic preference derivation mechanisms.

References

1. van der Aalst, W.M.P., Song, M.: Mining social networks: uncovering interaction patterns in business processes. In: Desel, J., Pernici, B., Weske, M. (eds.) BPM 2004. LNCS, vol. 3080, pp. 244–260. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-25970-1_16
2. Arias, M., Munoz-Gama, J., Sepúlveda, M.: Towards a taxonomy of human resource allocation criteria. In: Teniente, E., Weidlich, M. (eds.) BPM 2017. LNBIP, vol. 308, pp. 475–483. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74030-0_37
3. Cabanillas, C., García, J.M., Resinas, M., Ruiz, D., Mendling, J., Ruiz-Cortés, A.: Priority-based human resource allocation in business processes. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) ICSSOC 2013. LNCS, vol. 8274, pp. 374–388. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45005-1_26
4. Canal, G., Guillem A., Carme, T.: A taxonomy of preferences for physically assistive robots. In: 26th IEEE International Symposium on Robot and Human Interactive Communication, pp. 292–297. IEEE (2017)
5. Doyle, J.: Prospects for preferences. *Comput. Intell.* **20**(2), 111–136 (2004)
6. Erasmus, J., Vanderfeesten, I., Traganos, K., Jie-A-Looi, X., Kleingeld, A., Grefen, P.: A method to enable ability-based human resource allocation in business process management systems. In: Buchmann, R.A., Karagiannis, D., Kirikova, M. (eds.) PoEM 2018. LNBIP, vol. 335, pp. 37–52. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02302-7_3
7. Batista, M.F., Magdaleno, A., Kalinowski, M.: A survey on the use of social BPM in practice in brazilian organizations. In: Brazilian Symposium on IS, June 2017
8. García, J.M., Ruiz, D., Ruiz-Cortés, A.: A model of user preferences for semantic services discovery and ranking. In: Aroyo, L., et al. (eds.) ESWC 2010. LNCS, vol. 6089, pp. 1–14. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13489-0_1
9. Glasser, W.: *Choice Theory: A New Psychology of Personal Freedom*. Harper Collins, New York (1998)
10. Goldsmith, J., Junker, U.: Preference handling for artificial intelligence. *AI Mag.* **29**(4), 9–103 (2008)
11. Guo, Z., et al.: An enhanced group recommender system by exploiting preference relation. *IEEE Access* **7**, 24852–24864 (2019)
12. Hagel, J., Schwartz, J., Bersin, J.: Navigating the future of work: can we point business, workers, and social institutions in the same direction? In: *Deloitte Review*, Issue 21, pp. 27–45. Deloitte University Press (2017)
13. Halpin, T.: Temporal modeling and ORM. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2008. LNCS, vol. 5333, pp. 688–698. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88875-8_93
14. Huang, Z., Xudong, L., Duan, H.: Resource behavior measure and application in business process management. *Expert Syst. Appl.* **39**(7), 6458–6468 (2012)

15. Jalali, A., et al.: Supporting aspect orientation in business process management - from process modelling to process enactment. *Softw. Syst. Model.* **16**(3), 903–925 (2017)
16. Jorge, A.B., McIlraith, S.A.: Planning with preferences. *AI Mag.* **29**(4), 25 (2008)
17. Kaci, S.: *Working with Preferences: Less Is More*. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-3-642-17280-9>
18. Kumar, A., Dijkman, R., Song, M.: Optimal resource assignment in workflows for maximizing cooperation. In: Daniel, F., Wang, J., Weber, B. (eds.) *BPM 2013*. LNCS, vol. 8094, pp. 235–250. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_20
19. Lee, K.M.: Adaptive resource scheduling for workflows considering competence and preference. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) *KES 2004*. LNCS (LNAI), vol. 3214, pp. 723–730. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30133-2_95
20. Macris, A., Papadimitriou, E., Vassilacopoulos, G.: An ontology-based competency model for workflow activity assignment policies. *J. Knowl. Manag.* **12**(6), 72–88 (2008)
21. zur Muehlen, M.: Organizational management in workflow applications—issues and perspectives. *Inf. Technol. Manag.* **5**(3–4), 271–291 (2004)
22. Peintner, B., Viappiani, P., Yorke-Smith, N.: Preferences in interactive systems: technical challenges and case studies. *AI Mag.* **29**(4), 13–13 (2008)
23. Pika, A., Wynn, M.T., Fidge, C.J., ter Hofstede, A.H.M., Leyer, M., van der Aalst, W.M.P.: An extensible framework for analysing resource behaviour using event logs. In: Jarke, M., et al. (eds.) *CAiSE 2014*. LNCS, vol. 8484, pp. 564–579. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07881-6_38
24. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M.: *Work-Flow Patterns: The Definitive Guide*. MIT Press, Cambridge (2016)
25. Shaw, J.D., Duffy, M.K., Stark, E.M.: Interdependence and preference for group work: main and congruence effects on the satisfaction and performance of group members. *J. Manag.* **26**(2), 259–279 (2000)
26. Sindhgatta, R., Ghose, A., Dam, H.K.: Context-aware analysis of past process executions to aid resource allocation decisions. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) *CAiSE 2016*. LNCS, vol. 9694, pp. 575–589. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39696-5_35
27. Slovic, P.: The construction of preference. *Am. Psychol.* **50**(5), 364–371 (1995). ISSN: 0003–066X
28. Sohail, A., Dominic, P.D.D.: Business process improvement: a process warehouse based resource management method. In: *International Symposium on Technology Management and Emerging Technologies*, pp. 291–296. IEEE (2015)
29. Sohail, A., Dominic, P.D.D., Shahzad, K.: Business process analysis: a process warehouse-based resource preference evaluation method. *Int. J. Bus. Inf. Syst.* **21**(2), 137–161 (2016)
30. Vanderfeesten, I., Grefen, P.: Advanced dynamic role resolution in business processes. In: Persson, A., Stirna, J. (eds.) *CAiSE 2015*. LNBP, vol. 215, pp. 87–93. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19243-7_8
31. Yuen, M., King, I., Leung, K.: Task recommendation in crowdsourcing systems. In: *Proceedings of the First International Workshop on Crowd- Sourcing and Data Mining*, pp. 22–26. ACM (2012)
32. Zhao, W., et al.: An entropy-based clustering ensemble method to support resource allocation in business process management. *Knowl. Inf. Syst.* **48**(2), 305–330 (2016)