Data-Driven and Context-Aware Process Provisioning

Renuka Sindhgatta Rajan
University of Wollongong

Data-Driven and Context-Aware Process Provisioning

Renuka Sindhgatta Rajan

Supervisor:
Professor Aditya Ghose
Co-supervisor:
Dr. Hoa Khanh Dam

This thesis is presented as part of the requirements for the conferral of the degree:

Doctor of Philosophy

The University of Wollongong
School of Computer Science and Software Engineering

Nov 15, 2018

Declaration

I, Renuka Sindhgatta Rajan, declare that this thesis submitted in partial fulfilment of the requirements for the conferral of the degree Doctor of Philosophy, from the University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Renuka Sindhgatta Rajan
Nov 15, 2018

Abstract

Business process provisioning involves the allocation of resources (people, technology, or information) to process tasks in order to optimally realize the goals of the process. Resource allocation or task allocation refers to matching the right resource(s) to a task. The allocation of resources to process tasks can have a significant impact on the performance (such as cost, time) of those tasks, and hence of the overall process. While the problem of optimal process provisioning is hard, process execution logs or event logs contain rich information about the task, resource and process outcome. Past resource allocation decisions, when correlated with process execution histories annotated with quality of service (or performance) measures, can be a rich source of knowledge about the best resource allocation decisions. This dissertation offers a number of different approaches to support data-driven business process provisioning.

In complex and knowledge intensive processes and services, human process participants (resources) often play a critical role. Process execution data from a range of sources suggest that human workers with the same organizational role and capabilities can have heterogeneous efficiencies based on their operational context. This dissertation investigates the variation in resource efficiencies with varying case attributes (or process instance attributes), using a log of past execution histories as the evidence base, also demonstrating how data-driven techniques can serve as the basis for methodological guidelines for effective dispatching and staffing policies required to meet the contractual service levels (quality) of the service system and the business process.

This evidence bases also suggests that the optimality of resource allocation decisions is not determined by the process instance alone, but also by the context in which these instances are executed. Current approaches on resource allocation have not considered process context, case attributes and resource efficiency together. In this dissertation, a context model that considers resource behaviors is defined to support process provisioning. A range of approaches are proposed to support different dispatching scenarios such as pull-based dispatching and push-based dispatching. These methods use the process context, resource context as well as the functional goals and Quality of Service (QoS) requirements of past process executions to derive

resource allocation policies. The proposed methods are evaluated on real-life event logs.

In addition, this dissertation also proposes a method that leverages unstructured text associated with process instance logs to discover context or situations impacting process outcomes. Approaches of extracting information from process instance logs, and identifying common patterns are evaluated. It is observed that unstructured text can potentially provide insights into external factors impacting process outcomes.

The methods proposed in this dissertation are of considerable practical value. Conventionally, the decisions taken on resource allocation is based on human judgment, experience and implicit understanding of the context. Consequently, resource allocation activity is subjective, and relies on the experience of managers or resources themselves. Automated, data-driven support can be used to reduce human errors and aid human judgement, leading to improved process provisioning decisions and hence, improved process performance.

Acknowledgments

I have had many enriching and valuable experiences throughout my Ph.D. studies. I would like to thank my supervisor, Professor Aditya Ghose for trusting, guiding and providing generous encouragement throughout the journey. His expert advice, ideas and helpful discussions shaped all of my research work. I also want to express my gratitude for his flexibility and willingness to discuss and take calls any time.

I would like to thank my co-supervisor Dr. Hoa Khanh Dam for all the detailed reviews, useful comments and guidance in refining the work to ensure it led to better finished papers.

A special thanks goes to my colleagues at IBM Research-India, Gargi Dasgupta and Karthikeyan Ponnalagu for their discussions, support and friendly advice. I would like to thank my fellow members at Decision systems lab for their insightful discussions. I am very thankful to the management of University of Wollongong and IBM Research-India, that made this research possible.

Finally, my immense gratitude to my family who have been magnanimous in their support. I would like to thank my husband Ravi, and daughter Divya for always believing in me and encouraging me. I am very thankful to my mother Leela for her help and support. I am ever grateful to my late father Rajan, whose memory will always be with me.

List of Publications

Over the course of my PhD studies, I (co-)authored the following publications guided by my supervisor Prof. Aditya Ghose, and my co-supervisor Dr. Hoa Khanh Dam. Large parts of this dissertation has been published in multi-author papers.

1. Renuka Sindhgatta, Aditya Ghose, and Hoa Khanh Dam. Leveraging Unstructured Data to Analyze Implicit Process Context. In *Business Process Management Forum - BPM Forum 2018*.

2. Renuka Sindhgatta, Aditya Ghose, and Hoa Khanh Dam. Context-Aware Recommendation of Task Allocations in Service Systems. In *Service-Oriented Computing - 14th International Conference, ICSOC 2016*.

3. Renuka Sindhgatta, Aditya Ghose, and Hoa Khanh Dam. Context-Aware analysis of past process executions to aid resource allocation decisions. In *Advanced Information Systems Engineering - 28th International Conference, CAiSE 2016*.

4. Renuka Sindhgatta, Aditya Ghose, and Gaargi Banerjee Dasgupta. Analyzing Resource Behavior to Aid Task Assignment in Service Systems. In *Service-Oriented Computing - 13th International Conference, ICSOC 2015*.

5. Renuka Sindhgatta, Gaargi Banerjee Dasgupta, and Aditya Ghose. Analysis of operational data for expertise aware staffing. In *Business Process Management - 12th International Conference, BPM 2014*.

6. Renuka Sindhgatta, Aditya Ghose, and Gaargi Banerjee Dasgupta. Learning 'good' quality allocations from historical data. In *Resource Management in Service Oriented Computing, ICSOC Workshop, 2014*

7. Mohammadreza Mohagheghian, Renuka Sindhgatta, and Aditya Ghose. Combining agent based modeling with distributed constraint optimization for service delivery optimization. In *18th IEEE International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, EDOC Workshops 2014*.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Area

A business process is “the way an organization arranges work and resources, for instance the order in which tasks are performed and which group of people are allowed to perform specific tasks” [1]. Business process provisioning deals with the execution of a process by providing sufficient resources (people, technology, and information), to realize its goals. To position process provisioning, the BPM life-cycle is introduced (Figure 1.1). The life-cycle details multiple phases of managing a business process [2]. During the *design* phase, a business process is specified and modeled. The designed process is transformed to an executing process in the *implementation* phase. This is followed by a *monitoring* phase, where the executing process is maintained and data from the executing process is collected. Monitoring phase is followed by an *analyis (or diagnosis)* phase. In the analysis phase, the process model and corresponding process data are analyzed and possible improvements are identified, triggering a new cycle of design, implementation, monitoring, and analysis phase.

To ensure effective implementation, it is necessary to monitor and analyze process data. Process mining is the area of research enabling discovery, verification and improvement of an executing process by extracting information from process logs [3]. The main source of data for process mining are the process execution logs or event logs. Event logs capture information of each process instance such as the task, the time the task started and ended, the resource performing the task and other relevant information about the process instance (commonly referred to as case).

Much of the past research has used event logs for three types of analysis [3]: *Discovery*, where the process model of the executing process is extracted using event logs [4]. The roles and responsibilities of the resources [a] involved, are discovered from the event logs [5]. *Conformance*, where the process model derived from event log

[a] the thesis only focuses on human resources

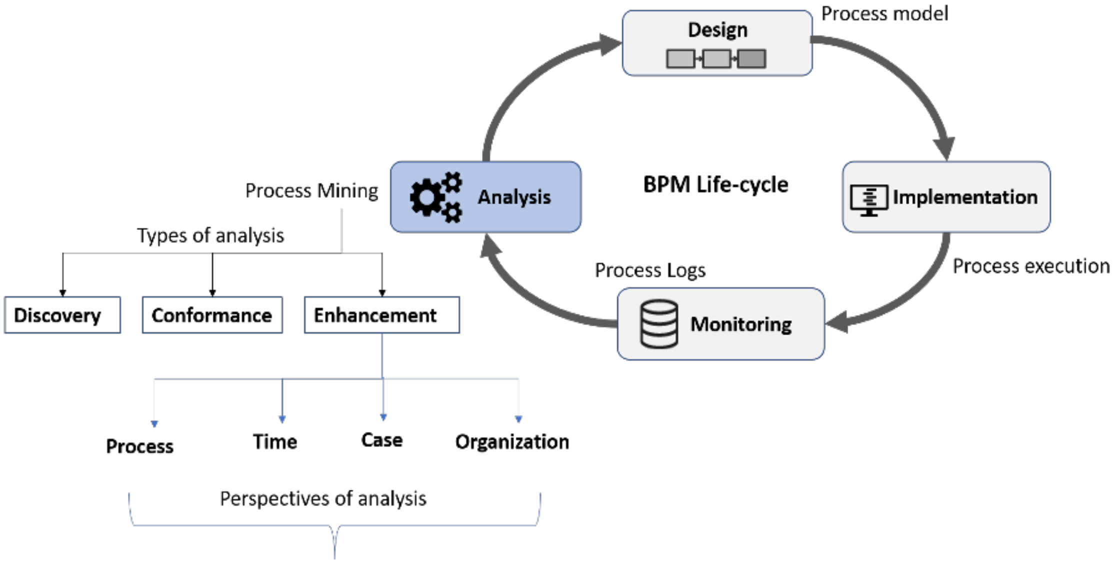


Figure 1.1: Business process life-cycle, types of analysis and perspectives

is compared with the designed process model [6]. *Enhancement*, where the existing process is improved or extended using the information recorded in the event logs.

Furthermore, enhancement of process using event logs can be viewed from four different perspectives [3]:

- **Process perspective**, focusing on the control-flow and the ordering of activities [1].
- **Time perspective,** analyzing different optimization methods to allocate tasks and meet the time constraints of the process [7], [8], [9], [10].
- **Case perspective,** analyzing different paths a process instance could take, and the impact of paths on the performance [11].
- **Organizational perspective,** analyzing resources and their efficiencies [5], [12], [13].

These four perspectives have been studied in isolation. However, resources efficiencies (organizational perspective) could vary for different cases (case perspective), and hence impact the process performance (time perspective). Not much attention has been paid in analyzing the three perspectives together. In this dissertation, I study all the three perspectives and further consider the variance in the efficiencies of individual resources in different situations.

The objective of the analysis from time, case and organizational perspective, is to improve resource allocation. Resource allocation or task allocation[b], is the

[b]task allocation and resource allocation are used as synonyms in this work

selection of a specific resource from a set of suitable resources for a task, during the execution of a business process. Previous work on task allocation considers all resources of a certain role to be the same [10], or with constant resource efficiency [8]. In practice, task allocation requires knowledge of variance in efficiency of resources. Often experienced managers use knowledge of resources and their efficiency, when allocating tasks to meet the required process performance.

1.1.1 Context-awareness in Resource Allocation

Context-awareness is the ability of a process to identify the context and make relevant adaptation. The importance of context has been recognized by many researchers in disciplines such as mobile computing, ubiquitous computing, information retrieval [14] [15], as well as business process management [16], [17]. There has been a collection of over 150 definitions of context from different disciplines [18]. Dey [19] defines context as "any information that can be used to characterize the situation of entities that are considered relevant to the interaction between the user and an application, including the user and the application themselves". Existing research on context-aware business process management has largely focused on the design phase of the process life cycle. There have been very few approaches that consider context during the analysis phase of the process life cycle [20], [21], [22]. In these studies, process attributes, context and process performance have been evaluated. The organizational (and resource) perspective has not been the focus of previous work. Nevertheless, context can be important when allocating task to resources. Thus, for handling an insurance claim from a high priority customer, we might allocate an experienced employee as a resource (the experience or other attributes of resource do not form part of the process data - they are neither generated, impacted or consumed by the process - but have a bearing on the execution of the process). Considering context of resource is important as it impacts their efficiency and hence the task allocation decision.

The main problem that motivates the research presented in this dissertation is:

How to model, extract and analyze contextual information of a business process in order to improve task allocation and realize process outcomes or goals?

The key contributions of this thesis is to propose some possible directions towards modeling, extracting and analyzing contextual information with the goal of improving task allocation:

- An approach to verify the influence of process attributes and context on resource efficiency. The data-driven approach is used to illustrate the impact of context on the cost of the executing process (staffing or providing required number of resources) in (Chapter 4).
- A context model using resource behavior indicators such as experience, cooperation, preference, computed from event logs (Chapter 5).
- A recommender system that uses the context model in addition to task and resource attributes, to recommend task allocation and improve allocation decisions (Chapter 5).
- A machine learning based method, used to derive resource allocation policies, taking into consideration, the influence of context on performance outcome. The policies can serve as input for future task allocation decisions (Chapter 6).
- A method to explore and discover contextual information that impacts performance outcome, from unstructured textual data available in communication and message logs of business processes (Chapter 7).

1.2 Thesis Outline

Most of the contributions presented in this thesis are based on published material. The thesis is organized as follows:

- Chapter 2 provides the reader with necessary preliminaries and is split into two parts. The first part summarizes the related work on business process management, process mining covering the three perspectives of organization, case and time. A summary of the work on modeling and analyzing context in business processes is presented. The chapter also provides a brief introduction to various machine learning methods used in the dissertation.
- Chapter 3 presents the overall research methodology. The research questions, the data used for answering the questions and the limitations of the data, are discussed.
- Chapter 4 highlights the variance in the resource efficiencies based on task attributes in a IT process [23]. This is followed by considering contextual attributes such as resource expertise to analyze variance in resource efficiencies. The chapter highlights the use of data-driven approach to identify variances

in resource efficiency for task allocation. The work presented here is based on [24].

- Chapter 5 considers the use of a context-aware recommender system for task allocation. A context model is built by considering resource behaviors. The task, resource, process outcome, and context is used as input to the recommender system. The recommender system is trained and an experimental evaluation is carried out to predict resource performance on data extracted from two real-life event logs. The material presented here is based on [25].

- Chapter 6 uses machine learning models to predict the performance outcome, given process context and process attributes. Experimental evaluation is carried out to identify predictors on a synthetically generated event log and a real-life event log. This work is based on [26].

- Chapter 7 uses unstructured text recorded by resources when executing tasks to identify patterns of common context that could lead to difference in the performance outcome of the process. Short phrases are extracted from textual logs of the process. The phrases are clustered and the performance outcome of the cases, the phrases are mapped to, is evaluated and compared with other cases. The material presented here is based on [27].

- Chapter 8 concludes the thesis and highlights future directions.

Chapter 2

Background

This chapter gives an introduction to the field of process modeling and process mining. It continues with a detailed description of the state-of-the-art in mining process execution data, focussing on resource assignment and allocations. Further, recent research on modeling and designing process context, followed by existing work on use of context in analyzing process outcomes is presented. Analysis of business process uses several data mining and machine learning techniques. A brief introduction to different methods used in this dissertation is presented.

2.1 Business Process Management

Business Process Management (BPM) "is the discipline that combines knowledge from information technology and knowledge from management sciences and applies this to operational business processes." [28] Another definition of business process management by Weske [2], highlights different phases of a business process: "Business process management includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes." Here, business process, "consists of a set of activities that are performed in coordination, in an organizational and technical environment. These activities jointly realize a business goal." Davenport [29] considers the relationship between the activities of a business process and defines as "a specific ordering of work activities, across time and place, with a beginning, an end, and clearly defined inputs and outputs."

BPM covers four phases in the life cycle of a business process. In the *design phase*, the process is designed and specified. A process model is used to specify the activities, their sequence or order, inputs and outputs. It is followed by the *implementation phase*, where the model is transformed into an executing process. In this phase, the process is realized using multiple software systems or process aware information systems. The *monitoring phase*, deals with verifying if the process is running as expected and if any adjustments are required. The process instance execution data becomes a critical need for the monitoring phase and the next phase of the process life cycle. In the *diagnosis phase*, the process instance logs are evaluated and analyzed for process improvement or re-design, triggering new iteration of the life cycle, if process changes are made.

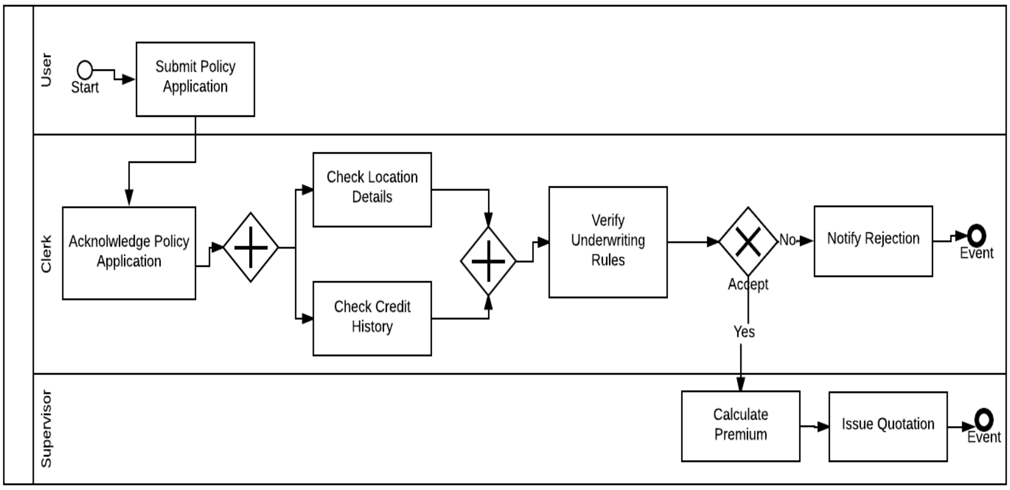


Figure 2.1: Business process of insurance policy application

2.1.1 Business Process Model

An important aspect of managing a business process is to design and document it, for it to be followed in a standard manner. A business process model is a representation of all aspects of a business process - activities, people performing the activities and data exchanged between activities.

Several formal notations exist to help specify a process such as Petri-nets [30], Event driven process chains (EPC) [31], Unified modeling language (UML) [32] and Business Process Modeling Notation (BPMN)[a]. BPMN is a widely used notation, with a goal of documenting processes that are understandable by all business users: i) designers creating the process, ii) technical developers building the process aware information systems that will execute or run the processes, and finally, iii) the business team who use and manage those processes. There are four categories of different elements of BPMN:

- **Events** depicting something that happens during a process and affects the flow of the process. Examples include start or end of a process, arrival of a message and so on.

[a]http://www.omg.org/spec/BPMN/2.0/

- **Activity** representing the work performed during the process.
- **Gateways** that control the flow of the process and either allow or disallow sequence of activities.
- **Sequence flows** showing the order in which activities are performed.

An example business process depicting issue of an insurance policy, represented using BPMN is provided in Figure 2.1. The model represents the sequence of activities such as 'Check Credit History', 'Verify Underwriting rules'. It further specifies the role or responsibility of the resource or worker performing an activity: 'Verify Underwriting Rules', 'Issue Quotation' is performed by resources having the role of a clerk and supervisor respectively. The model represents the sequence or ordering of activities, the divergence and convergences of the process flow using gateways. In the business process, the flow diverges depending on the outcome of the 'Verify underwriting rules' activity, to either notifying rejection of the application or calculating premium.

2.1.2 Resource Model in Business Process

Aalst et al. [1] introduced performers of activities in their definition of business process: "by process we mean the way an organization arranges their work and resources, for instance the order in which tasks are performed and which group of people are allowed to perform specific tasks."

Resources are entities that perform or are responsible for activities of a business process at runtime. BPMN 2.0 meta-model (Figure 2.2), defines a *HumanPerformer* element to help specify human roles. HumanPerformer inherits from *Performer*, and *ResourceRoles*. The earlier versions of BPMN had only *Performer*. *Potential owners* of the activity, are persons who can work on it.

The specification of resources and the surrounding organizational structure is kept separate from the process model, and is called the organizational model. Organization model is not supported by BPMN. Muehlen [33] considers organizational aspects of resources and defines an organizational meta-model. Cabbanilas [34] presents a conceptual map of all elements in the organizational meta-model. The organizational meta-model has relationships between various entities. The key entities or elements of the model that are relevant to my work are:

Person: A person or a resource (or knowledge worker), performs the activities of the business process. In this work person, resource and worker are used interchangeably.

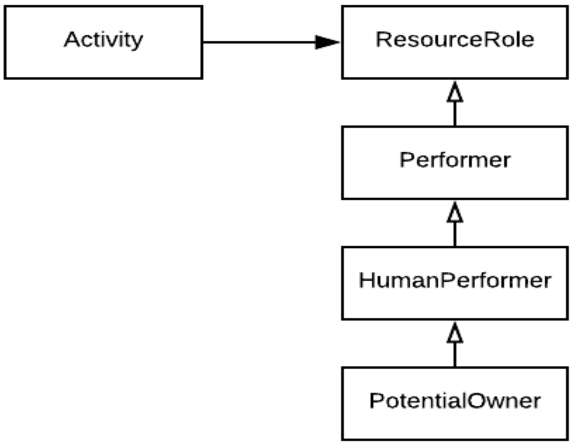


Figure 2.2: BPMN 2.0 class diagram for human resources

Capability: It is the ability of a person to perform a task. It can be referred to as a skill. Examples of capability include 'knows DB2 administration", "programming in Java".

Role: Role is the privilege a person has to perform tasks. A person can have one or more roles. Roles can be organized into a hierarchy where a role implicitly has all privileges of the role below in the hierarchy.

Team: In collaborative development, resources can be organized into teams where resources from multiple teams would work on different activities or tasks of the process.

Multiple relationships can be derived from the organization meta-model. A **person** has a **capability** and is privileged with a named **role**. A role has **permissions** to perform **activities**. The person is a **team member** in a **team** and occupies a **team position** and belongs to a **team type** (Figure 2.3).

2.1.3 Resource Assignment Models

Resource assignment, at design time, specifies resources that are capable and have the permission (or privilege) to perform specific activities in the business process, based on their roles. Given the specification, an actual resource or worker is allocated a task at run time (resource allocation). Russel et al. [35] define *workflow resource patterns* that capture different ways of resource assignment and allocation. The patterns are clustered into various categories: creation patterns, push patterns, pull patterns, detour patterns, auto-start patterns, and visibility patterns. The purpose of these patterns is briefly presented:

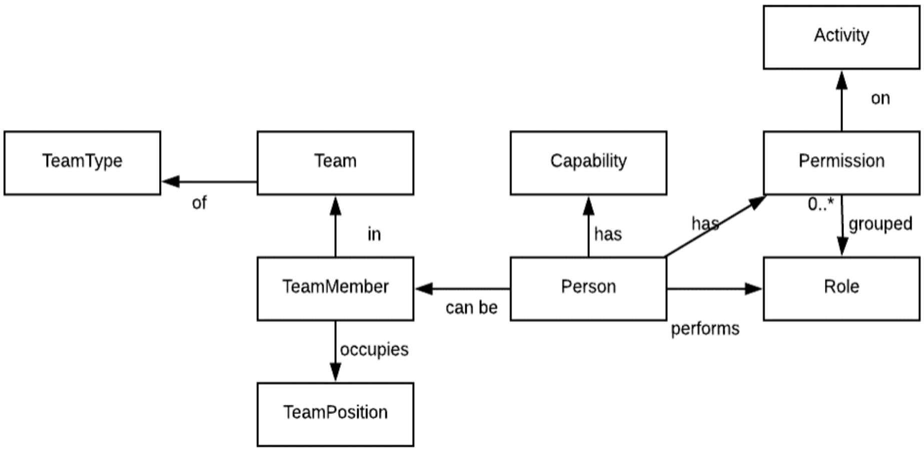


Figure 2.3: Relationships in the organization meta-model

Creation Patterns: These are assignment patterns specified at design time, and specify the range of resources that can work on the tasks related to an activity. They also influence the manner in which a task can be matched with a resource, capable of performing the task. These patterns provide clarity on how a task should be allocated to a resource before it is executed. There are eleven creation patterns: direct distribution, role based distribution, authorization, deferred distribution, separation of duties, case handling, retain familiar, case based distribution, history based distribution, organizational distribution, and automatic execution.

Push Patterns: These are allocation patterns where tasks (or work items) are allocated to resources by the system. Here, a system or a central dispatcher takes the initiative and owns the distribution of tasks. Nine push patterns are defined and divided into three groups. The first group of three patterns are of the system offering the task to a single resource, to multiple resources or allocates to single resource (on a binding basis). The second group of patterns relate to the system selecting a resource from multiple possible resources identified. Three possible strategies are: random allocation, round robin allocation and shortest queue. The final group of three patterns identify the timing of the task allocation to a resource and the time at which the task commences execution. Three possible patterns are: tasks allocated before they have commenced (early distribution), after they have commence (late distribution) or the two events are simultaneous (distribution on enablement).

Pull Patterns: These are allocation patterns, where the resources are made aware of specific tasks that require execution, and the commitment to undertake a specific task is initiated by the resource itself rather than the system. Generally this results in the task being placed on a common queue or the queue of a resource, where the resources may elect to commence execution on the task immediately or at a later point in time. Six pull patterns are divided into two distinct groups. The first group of patterns specify the different states of the task when the 'pull' request is made. The second group of patterns focus on the sequence in which the tasks are presented to the resource.

Cabannilas et. al [36] define a Resource assignment language (RAL), that uses description logic to specify such assignment patterns. It supports specification of all creation patterns, as RAL is defined at design time. An example of using RAL to assign task to supervisor is shown in Figure 2.4.

Role based allocation - The issue of quotation can only be done by a person who is a supervisor

```
Issue Quotation Authorization : HAS ROLE Supervisor
```

Case handling - The activity 'Notify Rejection' is done by the same person who performed the activity 'Acknowledge Policy Application'

```
Notify Rejection Authorization : (HAS ROLE Clerk) AND
(IS PERSON WHO DID ACTIVITY) Acknowledge Policy Application
```

Figure 2.4: RAL expression [36] for activities of business process in Figure 2.1.

Designing authorization constraints in BPMN has been discussed by Wolter et al. [37]. They present a formalization and modeling of task-based authorization constraints in BPMN, like separation of duty, case handling, and history based allocation. The task authorization constraint c for a set of conflicting tasks $T_c \in T$, where T is a set of all tasks of the process, is defined as:

$$c = (T_c, n_u, m_{th}), \quad \text{where} \quad n_u, m_{th} \in \mathbb{N} \tag{2.1}$$

The value n_u is the minimal number of users or resources that should be assigned to task $t_k \in T_c$. m_{th} is the threshold value of the sum of task instances a resource is allocated. A constraint of two tasks t_1, t_2 to be performed by different resources will be defined as $c_1 = (\{t_1, t_2\}, 2, 1)$. Similarly, a constraint of two tasks to be done by same resource will be defined as $c_2 = (\{t_1, t_2\}, 1, 2)$.

2.2 Process Mining

Process mining deals with discovering, checking compliance, and analyzing performance of a process by extracting data collected from process execution logs (also known as event logs) [3]. It is an activity that plays a dominant role during the monitoring and diagnosis phase of the process life cycle. Process aware information systems, record events (event log), where an each event refers to an activity in the process and the event is of a process instance or case. Consider, the event log for the process modeled in Figure 2.1. The process instances or cases are individual policy application requests and there is a trace of all events, recorded for each case. An example of a possible trace is shown in Table 2.1. Typically, events can be stored or recorded by multiple systems or applications involved in the process. However, the necessary information that should be captured by each application, is the following:

- Process instance information or a unique way of representing the case or the instance.
- Activity or the step in the process, the event refers to.
- Timestamp or the start time of the event. In addition, the end timestamp of the event may be recorded but is not mandatory.
- Resource responsible for executing the task of the process instance.

Additional domain or process specific information can be captured and analyzed. For example, in the insurance policy application process, the event log can capture additional information about type of insurance, the policy amount and so on.

There are three core areas where process mining has been used: i) control flow *discovery* that aims to construct the sequence of activities executing in the deployed process. ii) *conformance* verification to compare the executing process with the documented process. iii) *evaluating* the performance of executing process and their paths. Various algorithms have been developed to discover the process control-flow [3]. The α-algorithm is a very well known, simple and basic discovery algorithm. Several algorithms have been developed as detailed by Aalst et al. [3], that are capable of overcoming the limitations and strict assumptions of the α-algorithm such as incompleteness of the event log, ability to better handle loops, splits and noise in the event logs.

2.2.1 Organizational Mining

There are four distinct perspectives of process mining [5]: i) the process perspective, ii) the case perspective, iii) the time perspective, and iv) the organizational perspective . The process perspective focuses on the control flow or the ordering of activities. The case perspective focuses on the attributes of a process instance. The time perspective is concerned with timing and bottleneck analysis. The organizational perspective focuses on the performers of the activities, their roles and capabilities. The goal is to identify the structure and relationship between the resources. Song et al. [5], use similarity metrics based on the assumption that resources doing similar activities are more closely linked than resources doing completely different activities. They build a resource by activity matrix and use similarity metrics - Hamming distance, Pearson correlation coefficient to identify related resources. Resource by

Case	Activity Name	Timestamp	Resource	Type	..
1	Submit Policy Application	13-Mar-2017 9:10	Alex	Vehicle	..
1	Acknowledge Policy Application	13-Mar-2017 9:20	Barbara	Vehicle	
1	Check Location Details	13-Mar-2017 9:35	Carter	Vehicle	
1	Check Credit History	13-Mar-2017 9:50	Carter	Vehicle	
1	Verify Underwriting Rules	14-Mar-2017 17:10	Carter	Vehicle	
1	Calculate Premium	14-Mar-2017 17:10	Dan	Vehicle	
1	Issue Quotation	14-Mar-2017 17:10	Dan	Vehicle	
2	Submit Policy Application	13-Mar-2017 9:10	Joe	Home	..
2	Acknowledge Policy Application	13-Mar-2017 9:45	Barbara	Home	
2	Check Location Details	13-Mar-2017 9:35	Carter	Home	
2	Check Credit History	13-Mar-2017 9:40	Frey	Home	
2	Verify Underwriting Rules	14-Mar-2017 15:10	Frey	Home	
2	Notify Rejection	14-Mar-2017 17:10	Barbara	Home	
3	Submit Policy Application	13-Mar-2017 10:10	Min	Travel	..
3	Acknowledge Policy Application	13-Mar-2017 10:20	Barbara	Travel	
3	Check Location Details	13-Mar-2017 :35	Carter	Travel	
3	Check Credit History	13-Mar-2017 9:50	Frey	Travel	
3	Verify Underwriting Rules	14-Mar-2017 17:10	Carter	Travel	
3	Calculate Premium	15-Mar-2017 17:10	Dan	Travel	
3	Issue Quotation	15-Mar-2017 17:10	Dan	Travel	
4	Submit Policy Application	15-Mar-2017 9:10	Kala	Home	..
4	Acknowledge Policy Application	15-Mar-2017 9:45	Frey	Home	
4	Check Location Details	15-Mar-2017 9:35	Carter	Home	
4	Check Credit History	15-Mar-2017 9:50	Barbara	Home	
4	Verify Underwriting Rules	15-Mar-2017 17:10	Barbara	Home	
4	Notify Rejection	15-Mar-2017 17:10	Frey	Home	
5	...	...	...	...	..

Table 2.1: Event log [3] of the example business process in Figure 2.1

activity matrix consists all the resources (people) and the activities of the process. Each cell has the frequency of a resource performing the activity. Table 2.2 shows the resource by activity matrix for the event log detailed in Table 2.1. The resulting organizational model mined by using distance functions such as euclidean distance, and clustering techniques (detailed in Section 2.8), is shown in Figure 2.5.

Staff assignment mining [38], extracts complex assignment rules based on the capabilities of a resource and organizational hierarchy using decision tree learning. Positive and negative samples of data are created as triples:

$(x, a, \text{performer}(x, a))$, where, $x \in X$ is an instance of activity, $a \in A$ is the agent, and $\text{performer}(x, a)$ is $true$ if agent a has performed activity x, and $false$ otherwise.

Mining resource patterns using additional information from organizational model has been presented in [39]. The approach is capable of discovering most of the creation workflow resource patterns using declarative rule templates. Declarative templates are defined for resource allocation patterns. Support and confidence metrics, proposed by association rule mining methods used for declarative process model discovery [40], are applied to identify the relevant allocation rules.

Resource	Submit Policy	Ack. Policy	Check Location	Check Credit	Verify Underwriting	Notify Reject.	Calc. Prem.	Issue Quote
Alex	1	0	0	0	0	0	0	0
Barbara	0	3	0	1	1	1	0	0
Carter	0	0	4	1	2	0	0	0
Dan	0	0	0	0	0	0	2	2
Frey	0	1	0	2	1	1	0	0
Min	1	0	0	0	0	0	0	0
Kala	1	0	0	0	0	0	0	0
Joe	1	0	0	0	0	0	0	0

Table 2.2: Resource activity matrix [5] based on the event log in Table 2.1

2.2.2 Mining Resource Allocation

Given the process event logs, various methods have been used to identify patterns of allocation of tasks to specific resources. Much of the earlier work on resource allocation has focused on identifying role based access control (RBAC) models. RBAC model extraction, deals with identifying the privileges or authorization of resources on tasks from event logs [41], [42], [43]. Baumgrass et al. [42] parse event logs following XES, MXML format. They identify specific tags and derive activities performed by specific resources. Determining correctness and completeness of roles based on the RBAC specification, also known as *role mining*, has been presented in [44], [45].

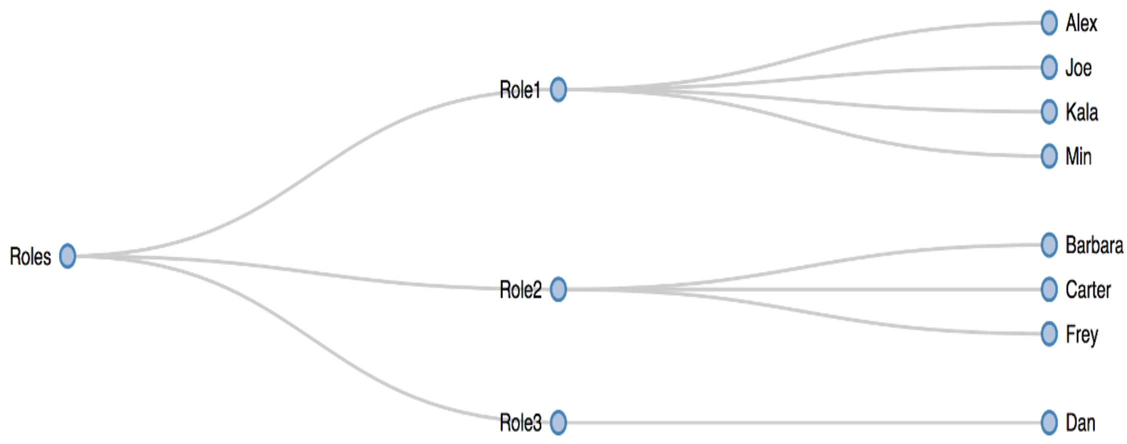


Figure 2.5: Roles identified based on resource activity matrix in Table 2.2.

Kumar et al. [46] introduce the notion of dynamically allocating a task to a resource. The authors define a work allocation metric that can be used to allocate task to resources based on suitability, availability, conformance and urgency. In this work the authors do not use event logs but perform simulation based experiments to emphasize the need for such allocation metrics.

Huang et al. [47] present a reinforcement-learning based resource allocation mechanism, which considers allocation of task in a process as an interactive problem. They apply a Q-learning algorithm, which is a reinforcement learning based method [48]. The method uses the workload and a cost function to provide a set of suitable work-items for a resource. Given a new work item, the algorithm lists the resources that are most suited for the work-item. However, this approach is computationally intensive and may be infeasible for large number of cases being executed.

Cabanillas et al. [49] integrate the problem of resource prioritization into assignment and allocation. First, each activity of the process has a resource assignment specification that defines its set of potential performers (in this work Resource Assignment Language is used). Second, preferences for resource prioritization is formulated using Semantic Ontology of User Preferences (SOUP) [50]. Preferences are specified for each activity of the process. Examples of preferences are skills or cost. These could be derived from the history of past executions. A ranking mechanism is defined that takes a preference and the set of resources to be ranked, as input. Based on the preference, a partially ordered set of the ranked resources is generated.

Optimal allocation of tasks to resources using constraint programming [51], linear programming [52], petri-nets [53] and other scheduling methods [9], [54], [55], [56], have been proposed. While they do not use event logs for analyzing historical information, they address a common goal of improving task allocation.

2.2.3 Mining Resource Behavior

Aalst et al. [7] list some of the main problems when modeling resources and simulating a business process:

- People are involved in multiple tasks. Most tools assume that resources work on a single task or process.
- People do not work at a constant speed. There is a relationship between the workload and the performance of a person.
- People may work in batches. Most process simulations assume that a person is always ready to start working on a task.

This work alluded to the need for analyzing resource behavior. Nakatumba et al. [12] analyzed the influence of **workload** on service time. The authors define workload as "the number of activities that have been executed over a particular period", which defines 'how busy' the resource has been. The service time is the time taken to process a given task. The authors use event logs to extract the service times and workload on the resource. A linear regression model is built. The regression model uses workload as a single predictor of service time. While this model is useful to compare specific resources and their efficiencies, it is limited as there are several factors of the process and resource influencing service times. However, this work was one of early studies that used event logs to characterize resource behavior.

Huang et al. [57] present resource behavior measures for competence, preference, availability and collaboration. Event logs are used to measure resource behavior:

- **Resource preference** at a time interval t is effectively, the ratio of the number of bids by a resource r on an activity a, to the number of bids by all the resources on the same activity. The preference of a resource may change with time and hence the measure combines the preference degree computed at time $(t-1)$ with the preference degree computed at time t.
- **Resource availability** is computed by considering the arrival rate of tasks for a given resource r and the number of completed activities in a given time interval. Availability of a resource is a boolean function.
- **Resource competence** is the capability of a resource to complete a task using lower cost. The cost could be any metric: time or quality (e.g in software development, defects would be an indication of quality).
- **Resource cooperation** between two resources $Coop(r_i, r_j)$ is measured by considering the conditional probability of a resource r_i working on an activity a_1, given resource r_j working on another activity a_2.

Kabicher-Fuchs et al.[58] discuss the need for measuring and focusing on **work experience** in process aware information systems. Further, they define an **experience** breeding meta-model [59]. Their work extends the resource model comprising of user, roles and tasks with additional concepts: experience, goals and levels. Experience is gained by performing tasks. There are various levels of experience that can be gained and help in achieving a goal. The premise of the work is based on the assumption that, allowing users to define experience breeding goals would motivate resources and increase their satisfaction. Five patterns of goals are described.

An example of a pattern of experience breeding goal is given as: "*BECOME SPECIALIST at CHECK CREDIT HISTORY until May 2017*".

Experience is measured by considering i) count of how often the experience has been captured ii) duration of experience (how long the experience been captured) iii) importance of experience (how important was the task) and iv) quality of experience (what was the quality of the task). The simulation experiments show that resource allocation using experience breeding measurements improved the quality, duration of task execution, and the goals of the resources as compared to a round robin approach of resource allocation.

Kumar et al. [60] highlight the use of **cooperation** among the resources involved in the process, and develop an allocation algorithm that maximizes team cooperation. The following metrics for measuring compatibility of a team or a process is defined:

$$Total \quad Compatibility = \sum_{\forall u_1,u_2,t_1,t_2} fit_{u_1,u_2,t_1,t_2} * coop_{t_1,t_2} * cweight_{u_1,u_2}$$

$$fit_{u_1,u_2,t_1,t_2} = \begin{cases} 1, & \text{if resource } u_1 \text{ and } u_2 \text{ perform task } t_1 \text{ and } t_2 \text{ respectively} \\ 0, & \text{otherwise} \end{cases}$$

$$coop_{t_1,t_2} = \begin{cases} 1, & \text{if cooperation is required between task } t_1 \text{ and } t_2 \\ 0, & \text{otherwise} \end{cases}$$

$$cweight_{u_1,u_2} \text{is compatibility of resources } u_1 \text{ and } u_2$$

A technique for computing $cweight_{u_1,u_2}$ from the logs is described. The metric is based on the assumption that, the throughput times of the tasks would be lower than average if resource u_1 and u_2 are compatible. The throughput time of the tasks would be higher than average if the resources are not compatible. The optimal work allocation that maximizes cooperation is found to perform 20% better that the heuristic greedy algorithm.

Resource behavior indicators [13], have been defined by Pika et al. In [61], they present a framework for analyzing and evaluating resource behavior indicators (RBI) from event logs. The framework consists of three modules. The first module computes information about the resources along the five categories - skills, utilization, productivity and collaboration. The behavior indicator measures captured by Pika et al. are presented:

- **Skills**: The metrics associated are: i) the distinct activities (indicating different types of tasks) performed by the resource, ii) distinct types of cases handled, and iii) the number of activities performed in a given time period
- **Utilization**: i) The number of completed tasks by resource in a given time period, ii) the number of completed cases in a given time period, iii) the ratio of the completed cases involving a resource to the total number of completed cases in the given time period, and iv) the workload of the resource (the number of tasks in progress, at a given time).
- **Preferences**: i) The fraction of time, the resource is multitasking, ii) the number of times in a given time period, the resource worked on a task with attributes the resource had never worked before, and iii) the number of times the task done by the resource was completed by another resource.
- **Productivity:** The productivity indicators include: i) The ratio of the number of completed tasks by a resource with a given outcome to the total number of completed tasks by the resource in a given time period, ii) the average task duration where the resource was involved, iii) the average case duration where the resource was involved, and iv) the average customer feedback for the cases completed in a given time period where the resource was involved.
- **Collaboration:** the indicators are i) the number of completed cases during a time period involving two resources, ii) the ratio of the number of distinct resources involved in the cases involving a resource to the total number of active resources during the given time period, and iii) the number of times the resource delegated a task to another resource.

The RBI time series is extracted from the event logs and their trend is tracked over a period of time. These can be used to identify outliers or points where the RBI values are significantly different from the typical values.

The second module of the framework *quantifies the relationship between RBI and outcomes*. The outcomes could be either customer feedback, cost or task duration. Regression analysis is used to determine the quantitative relationship between the RBI and the outcome (similar to [12]). The third module of the framework, *evaluates* *resource productivity*. The framework allows users to define inputs (some of the relevant RBI), and outputs, for a given resource during a given time slot. An efficient frontier is identified using the inputs and outputs from an event log, i.e. evaluate best practice for high productivity of a resource.

The work related to resource behavior is largely related to mining information from event logs and identifying outlier behaviors. However, using the behavior of resources to make allocation decisions to improve the efficiency of a process, would be valuable. In my work, the allocation of task by considering resource behaviors to maximize the process outcome, has been addressed.

2.3 Predictive Analytics Using Event logs

Predictive analytics based on event logs has primarily focused on two key areas: 1) predicting the next activity, 2) predicting completion time of the case or a related measure, i.e. if the case is overtime or not. The former focuses on enabling prediction of control-flow and the latter focuses on an outcome of the case in terms of the duration. This section discusses some of the recent work and progress made:

2.3.1 Completion Time Prediction

Early work on cycle time prediction [62], uses non-parametric regression method to predict remaining cycle time. The independent variables or inputs to the regression model are, the duration of all activities, the occurrence of activities and case related data. This approach hence, largely considers the case information. The resources working on the activity are not considered in this prediction model.

Similar work by, Aalst et al. [63] defines an approach, where the current state of the process instance is compared with other historical instances by applying various abstractions on the task sequences in a transition system - i) Maximal Horizon: in this case, instead of taking the entire prefix of activity sequence $\langle A, D, C, B, C, C, E \rangle$ only the last four events $\langle B, C, C, E \rangle$, can be considered as input for the next state calculation, ii) Filter: certain events or activities are filtered while considering the current state, iii) Sequence: bag or set where the set of activities is considered without considerations to the frequency or order (e.g set for the sequence is $\{A, B, C, D, E\}$). To predict the completion time, the partial sequence of events executed so far is used as a state in the transition system to arrive at the entire sequence. Then the information collected from earlier process instances that visited the same state is used to predict the completion time. Average completion time of earlier process instances in a similar state is completion time of the new process instance.

Suriadi et al. [64] in their work, enhance or enrich the event logs by converting or aggregating it to a case log. The case log has the relevant attributes of the case, such as the activities executed in the case. Additional information is extracted from the event log such as resource workload. The approach enriches and transforms event log into a form that allows a root cause analysis to be evaluated as a supervised classification problem. In their evaluation, the authors learn to classify or predict process instances that took longer duration than expected.

2.3.2 Next Activity prediction

Schonenberg et al. [65] use historical process logs to recommend activities in flexible business process. In this early work, the authors propose the ability to define target functions such as duration of the case, business value of the case and use these functions to identify similar cases from the history and recommend next best activity.

Lakshmanan et al. [66] use Markov Chains to build a probabilistic process model (PPM) for each process instance, where the transition probabilities are based on the semi-structured business process instance it represents. They use Markov techniques to predict the likelihood of executing next tasks. They compare the process instance-specific PPM with methods such as conditional probability and show that instance-specific PPM results in more accurate predictions.

Tax et al. [67] use Long Short-term Memory (LSTM) neural networks to predict the next activity of a running case and its completion time. Their experiments show that the LSTM based model outperforms existing baselines on real-life data sets. They further show that predicting the next activity and its timestamp using a single model results in higher accuracy than predicting each of these target variables using separate models. This approach is unable to deal with cases with multiple occurrences of the same activity, and the model predicts long sequences of the same activity.

2.3.3 General Predictive Analytics Framework

Maggi et al. [68] propose a framework to predict the outcome of a case (normal vs. deviant) based on the sequence of activities executed in a given case and the values of data attributes of the last executed activity in a case. A classifier is trained on historical cases and predicts the outcome based on cases similar to the current trace of a running case.

Teinemaa et al. [69] present a framework to predict process outcomes (normal or deviant) using unstructured textual information present in the communication logs, or process systems. They use various text processing methods to encode as features in addition to the case attributes and train a classifier based on historical event logs. The model is used to predict process outcomes of new cases.

Leoni et al. [70] present a generic framework for deriving and correlating business characteristics. The generic framework consists of three key steps: defining the use case, enriching event logs with relevant information, and making the relevant analysis. An event log for the process is used and enriched with additional case related information such as elapsed time of the process, workload of the resource or any other independent characteristics. A decision tree based approach is used to predict the dependent variable that can be defined based on the use case. The authors analyze five examples to evaluate the generic framework: 1) predicting violations by predicting next activity, 2) predicting outcome which is a quantitative value, 3) predicting the next activity, 4) predicting faults in process executions, and 5) predicting the performer or resource of an activity. This approach highlights the key steps for predictive analysis of a business process.

In all the above approaches to predicting completion time or performance outcomes, the resource characteristics is considered homogeneous i.e. the resource as a feature has been added in some of the works, but the impact of including the resource and the characteristics of the resource, on the prediction accuracies of performance outcomes or completion times has not been evaluated.

2.4 Analyzing Service Systems

Service System as defined by Maglio et al. [71], is an important unit of analysis in understanding operations of an organization. A Service System (SS) comprises of resources (that include people, organizations, shared information, technology) and their interactions that are driven by a business process to create a suitable outcome to the customer. Hence, much of the work done in the context of service systems is applicable to any knowledge intensive business process. A formal model of a service system has been defined by Ramaswamy et al. [72]. Some of the studies related to analyzing skills of service workers and team organizations for improved service delivery is discussed in this section.

2.4.1 Staffing and Routing in Service Systems

Initial work on arriving at an optimal staffing of a service system, considers requests arriving to the system, the associated customer, priority (severity), and required skills [23]. The combination of customer, priority, and request type determine the target service time and associated service level percentage attainment. The authors use a simulation model to optimize the number of agents required in each service delivery center, such that the service levels percentages of all customers. The average service time required to complete a customer request given its complexity and the skill of the service worker is considered as input. In addition to providing staffing recommendation, the optimization model can be used to perform what-if analysis. Another study compares different dispatching policies and their impact on staffing of teams with varying service system parameters such as service levels and availability of service workers [73].

Routing work to the relevant teams or service workers has been studied earlier [74], [75]. Shao et al. [74], evaluate routing of tickets by mining ticket resolution sequences. A Markov model is developed to statistically capture transfers between resolver groups or teams, toward efficient ticket resolution. The approach does not access ticket content. The authors extend their work by using textual information in the tickets and resolution sequences to capture multiple resolver groups [76]. Agarwal et al. [75] use the textual information present in the problem descriptions of IT incident tickets to identify relevant teams. The authors use a combination of classifiers to improve the accuracy of the model predicting the relevant team that should resolve the ticket. They do not consider routing of tickets to multiple teams.

2.4.2 Team Organization in Service Systems

Given that a service system has knowledge intensive work, the skills of knowledge workers plays a critical role. Team organization is important as it impacts the routing of service requests or tasks, and hence the completion time. As described in their work, Agarwal et al. [77], compare different team organizations to support requests from different customers (Figure 2.6). There can be three types of teams: (a) Customer focused (b) Business function focused and (c) Technology focused. Figure 2.6 shows a relationship among business functions, technologies and teams for each of the three models. The legend for technology, business and customer in the figure is as follows: technologies are denoted by colors, the business functions are denoted by the shape of the boxes and the customers are denoted by the different patters in the boxes. A customer has systems based on different technologies (Unix, Windows, Transaction Server, etc.) catering to different business functions (Payroll, Billing, Marketing, etc.).

- In the Customer focused (CF) model, all service interactions of a customer, across all business functions are served from single customer dedicated team.
- In the Business focused (BF) model, business functions of multiple customers are served from the common pool. The resources in a team have the desired domain knowledge in addition to the required technical skills required to carry out the tasks.

- In Technology-focused (TF) model, multiple customers using similar technologies are grouped into a team which is served by highly skilled people in the relevant technologies.

The authors model different types of requests and compare three distinct models in terms of the time it takes to complete a request. They conclude that nature of work arrivals and skill requirements of customers determine the suitability of team organization.

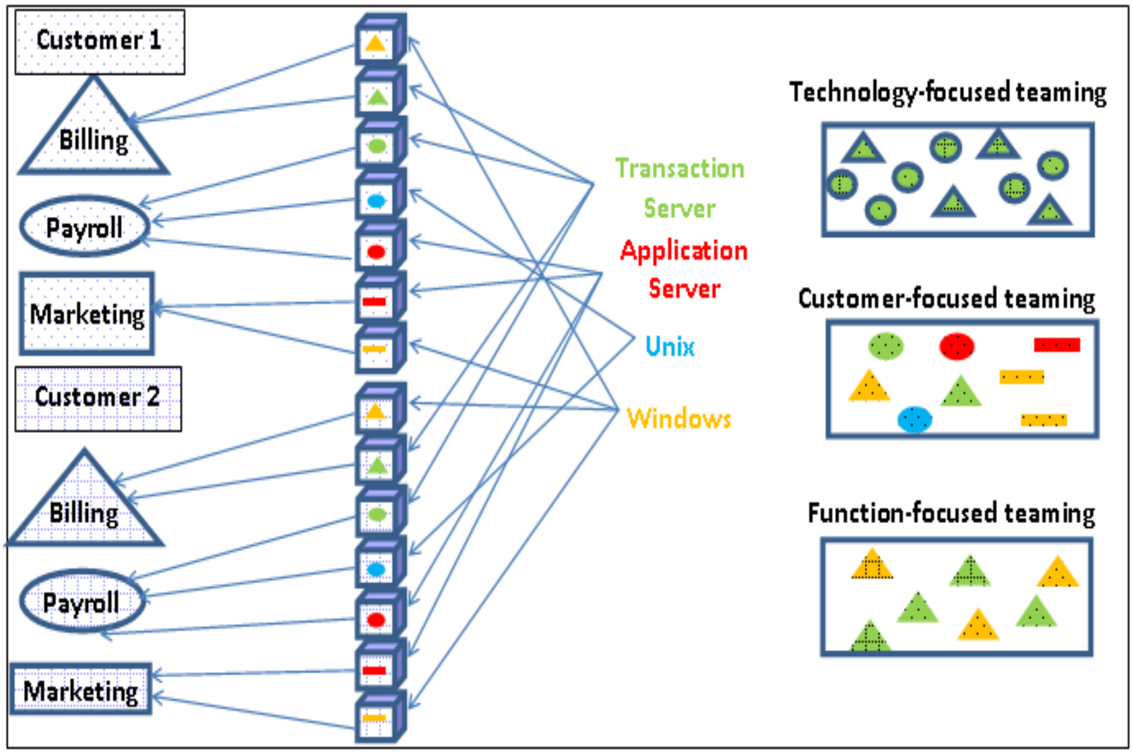


Figure 2.6: Organizing teams in a service system to support customers in [77].

Similar study on team organization compares the navigation of a service request (SR) or a work item through various teams [78]. The authors motivate the problem by providing a complex SR in IT service systems that requires multiple skills to resolve the problem. Here, the organization of the teams with different skills would impose different workflow on the resolution of the SR:

- Decoupled Workflow: When multiple teams work independently on a complex customer SR, with each team only responsible for partial resolution of the issue, it imposes a decoupled structure on the SR resolution flow.

- Collaborative Workflow: When the complex SR is handled by experts from multiple teams, working on the SR simultaneously, it imposes a collaborative structure on the SR resolution flow (as discussed in [79], [80]).

- Integrated Workflow: In cases where a team is composed of multiple skill specializations, the SR may be handled by multiple skills within the same team. Here one team owns the SR and one or more multi-skilled people work towards its resolution.

The authors compare the staffing required to support different customers for each of these workflows when customer request have different service levels. They conclude that the suitability of the workflow is based on service system parameters such as arrivals of service requests, service level agreements and the skill requirement of the service request.

The models for staffing and team organization consider homogeneous efficiencies of resources having similar skill or experience. They do not consider resource behaviors other than availability of a resource. The impact of service time based on resource behavior is not accounted for during model simulation and evaluation.

2.5 Context-Aware Business Process

One of the core premise of the work presented in this dissertation, is that human resources are **heterogeneous** and hence their efficiencies are impacted by specific *resource behaviors* that manifest as **resource context**. This section explores the state-of-the-art in modeling and specifying business process context. It is followed by the relevant studies on using context and analyzing performance of process or process outcomes.

2.5.1 Context-Awareness

Awareness of context has been widely discussed in areas such as mobile computing and e-commerce. Dey [19] defines context as "any information that can be used to characterize the situation of entities that are considered relevant to the interaction between the user and an application, including the user and the application themselves". Bazire et al. [18], create a database of more than 150 definitions of context picked up from various disciplines such as computer science, philosophy, economy, business and analyze the definitions using clustering techniques. The authors conclude that "context acts like a set of constraints that influence the behavior of a system (a user or a computer) embedded in a given task". Kiseleva et al. [81] introduce the notion of implicit and explicit context for predicting user behavior in e-commerce applications. The web user's age, gender and other known attributes are considered as explicit context, while information such as the purchase intent of the user is not known and is considered to be hidden context.

Dourish [82] presents two different views of context: representational view and interactional view. The representational makes the following assumptions:

- Context is information. It is something that can be known (and hence encoded)
- What counts as the context of activities can be defined in advance
- Context is stable. Contextual information does not change from instance to instance.
- Context and activity are separable. The situation within which the activity takes place, can be separated from the activity itself.

An alternate interactional view makes the following assumptions:

- Context is a property of the information and is a relational property. Something may not be contextually relevant to some particular activity.
- Contextual features are defined dynamically.
- Context is property that is relevant to particular settings.
- Context arises out of an activity. It is not there but produced by the activity.

The thesis largely considers the representational view of context. There are some cases where an interactional view can be relevant. The situation(s) that arises while performing an activity or task of a business process instance can be considered from an interactional view point.

2.5.2 Modeling Business Process Context

In BPM, contextual information has been categorized by Rosemann et al. [17]. The authors propose different layers of context: i) immediate context related to the control flow of the process, ii) internal context that captures information about the organization iii) external context capturing information beyond the organization, and iv) environment context that is beyond the organization but effects the business process.

Context modeling for business process has been introduced and discussed in [83], [84]. Saidani et al. [83] present the need for context related knowledge (CRK) at various elements of the meta-model of business process. The notion of context for a business process is considered to be any information reflecting the changing circumstances during the execution of the process. They define context as "the collection of implicit assumptions that is required to activate accurate assignment in the BP model at the process instance level." A taxonomy of common contextual information for a process is defined. The important kinds of context are:

- Location related context: representing location information. The location of the resource would impact the ability of the resource to execute a process instance.
- Time related context: representing features related to time such as hour of the day, month of the year, and so on. Process instances created at different times of the day would be assigned to different instances (depending on work shifts).
- Resource related context: representing all human resource properties. These are the age, gender, quality of communication, and any resource specific information that can be useful for assignment of tasks.
- Organization related context: represents the organizational hierarchy such as position, role of the resource.

Context is defined using $\langle ASPECTS, FACETS, ATTRIBUTES \rangle$. Aspects are the different elements of the taxonomy (location, time, resource, organization). Aspects have facets and further attributes.

The authors extend their work by specifying a context meta-model for business process [16]. The core concepts of the meta-model are:

- Context entity: Context entities are elements of the process such as actor, task, resource, organizational unit and so on
- Context attribute: Context entity has context attributes which are measurable and atomic.
- Context relationship: Context relationship connects two context entities
- Context element: Context relationship and context attribute inherit from context element. Context elements are of two types - i) static element is fixed and does not change with time (gender of the resource, age), while ii) dynamic element changes with time (such availability of the resource).
- Method of capture: It specifies how the context element is determined or computed
- Contextual situation: Context situation is determined by the contextual element and its associated value.

Further, there are two types of contextual information: contextual information which is independent of the business domain and the process and contextual information that is dependent on the business domain or the process. Figure 2.7 shows a partial domain specific context model that focuses on the resources of the example business process of Figure 2.1. The contextual attributes of the resources would include experience, location, certification and so on. For the customer, the location would be an important contextual attribute. Bessai et al. [85] motivate the need to dynamically orchestrate task allocation based on resource specific criteria that includes context: roles of resources, real workloads and resource availabilities. A framework consisting of a resource repository with all information about resources, and a centralized resource manager that allocates task based on the information contained in the repository is proposed.

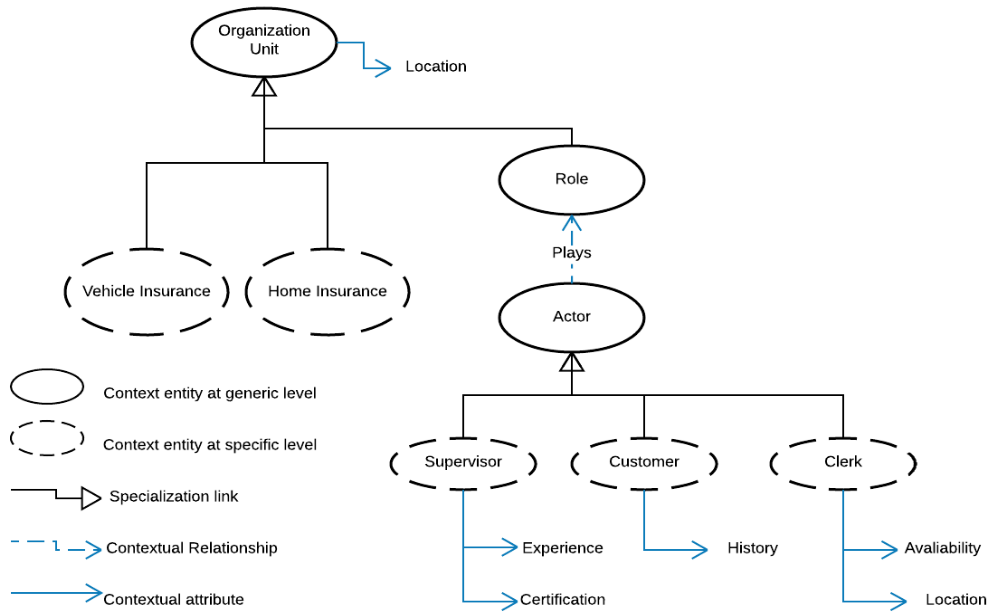


Figure 2.7: Partial definition of domain specific context model focusing on resources for example business process in Figure 2.1 based on context model defined in [16]

2.5.3 Learning from Context and Performance Outcome

Given a context, it would be useful to identify its path of execution. Ghattas et al. [20] evaluate the impact of specific properties of a process that impact its execution. Context is termed as information "addressing both the events and conditions in the environment and the specific properties of cases handled by the process". Context $C = \langle I, X \rangle$, and I as defined by authors, "is the variable values at the initial state of the process and X is the set of external events that affect the process instance at run time". In our example of insurance policy application process (Figure 2.1), I is the type of policy (vehicle, home, etc.) or policy amount. X would be the time the customer submitted the policy application. The process instances are grouped as process groups based on process behavioral similarity and context property similarity. A five step algorithm identifies the different process context variables that impact process execution paths:

- Group the process instances into N clusters based on domain knowledge
- Identify the behavioral similarity of the process instances based on process behavioral similarity. i.e. process path and termination state. Group process instances having similar process instance behavior.
- Determine the process contextual properties. Build a decision tree, using the context data as inputs and the process instance groups as dependent variable
- Form context groups by considering all paths of the decision tree. Eliminate or prune the paths that have process instances with different termination states.
- Merge context groups having the same process instances.

The authors use this approach in a clinical process and evaluate on 297 cases of patients [22], in order to automatically identify context groups. First similar process instances are clustered. Then decision tree is applied to predict the cluster labels. This enables to identify context groups. An example of the context group is *55 < age < 65 AND (General_state = Medium or General_state = Good) AND Beta Blockers= Y.*

In another study on using historical event logs and context, the paths and the process context that lead to specific process outcomes are identified [21]. The approach consists of the following steps:

- Define the goals associated to the process. This is the objective of the process which can be a weighted combination of soft goals.
- Select past process instances that executed with a similar objective from the repository or knowledge base of historical executions.
- Cluster process instances with similar paths and similar termination state into context groups.
- Use a decision tree algorithm, where path and context of a process instance are the independent variables, while the achieved weighted soft goal scores are the dependent ones.
- Determine the best performing path and context variable that lead to the performance outcome.

- Derive the decision rules from the decision tree and evaluate with the help of a domain expert .

The approach is evaluated by applying it on the data of 50,000 simulated process instances of a bottle manufacturing process. The key limitation of the approach, as discussed by the authors, is the inability to learn from exceptional situations or new lines of action, as the method relies on similar executions for the decision tree algorithm to learn from the past.

In this dissertation, historical execution logs are used to extract process outcomes and context with a focus on resource allocation, as human resources are crucial drivers of the performance of knowledge intensive business processes.

Reference	Analysis Type	Output	Info. extracted from logs				Allocation (Algorithm)
			R	*Task*	*Ctx.*	*O*	
[5]	Discovery	Roles	✓	✓	✗	✗	✗
[38], [39]	Discovery	Resource assignment rules	✓	✓	✗	✗	✗
[41], [42], [43]	Discovery	Role based access controls (RBAC)	✓	✓	✗	✗	✗
[46]	Enhancement	Resource metrics	✓	✓	✗	✗	✓ (allocation based on resource metrics)
[52], [53] [9], [54], [55], [56] , [47]	Analysis	Task scheduling	✗	✓	✗	✓	✓ (scheduling and optimization)
[49]	Enhancement	Resource metrics	✓	✓	✗	✗	✓ (allocation based on resource metrics)
[12], [57], [58], [13], [61]	Enhancement	Resource behavior metrics	✓	✓	✓	✓	✗
[60]	Enhancement	Metric based allocation	✓	✓	✓	✓	✓ (Optimization model based on single contextual attribute)
[62],[64], [65]	Enhancement	Completion time and next activty prediction	✗/✓	✓	✗	✓	✗
[74], [75]	Enhancement	Team allocation	✗	✓	✗	✓	✓ (predictive model for team identifcation)
[78],[77]	Enhancement	Team organization	✗	✓	✗	✓	✓ (Simulation based analysis)
[20], [21], [22]	Enhancement	Process performance patterns	✗	✓	✓	✓	✗

Table 2.3: Review of existing solutions for resource allocation based on information extracted from event logs where, R (resource attributes), Task (task attributes), Ctx (contextual attributes), and O (task or process outcome) are used for analysis

2.6 Review of process mining for resource allocation

Table 2.3 presents a review of the approaches studied in section 2.2 to section 2.5. For each of the methods addressing the analysis phase of the business process lifecycle, the table indicates (i) the type of analysis that was done, (ii) output of the analysis, (iii) different information or inputs extracted from the event logs, and (iv) if purpose of the analysis was resource allocation. The symbol ✓ is used to indicate the type of information or attributes extracted from the event log: R, indicates if information or attributes of resources was considered in the study, $Task$, if attributes of task were used, $Ctx.$; if contextual information was used in the study, and O; if the task or process performance was considered in the study. The table further indicates if the purpose of the approach was task allocation (with a symbol ✓).

As shown in Table 2.3, most of the approaches dealing with resource allocation do not consider contextual information. Further, many resource allocation approaches do not consider resource specific attributes (and assume all resources of a given role have similar behavior). There have been limited studies that consider resource characteristics when allocating task ([60]). The dissertation addresses this gap by building predictive models that use resource behavior and other contextual information for task allocation.

2.7 Leveraging Textual Data in BPM

This section presents existing work on using textual or unstructure data from event logs for process analysis. Existing studies have focused on i) discovering process models from textual artefacts, and ii) using textual information for predicting process performance:

Most organizations maintain documents that detail standard operating procedures describing a business process. During the execution of a process, process aware information systems provide the ability to document and capture important information about the execution (such as communication logs, email exchanges). These form a rich source of knowledge for the modeling and analyzing the process. With the recent advances in natural language processsing, there have been multiple studies leveraging textual data in modeling and analysis of business processes.

2.7.1 Textual Data for Process Modeling

Extracting and generating business process models using textual artefacts of an organization, has been explored. Ghose et al. [86] propose a Rapid Business Process

Discovery (R-BPD) framework and toolkit that employs text-to-model translation. The framework uses two types of text-to-model translation:

- *Template-based extraction*: Templates of commonly occurring textual patterns are identified by scanning documents in the document repository. An example of a common pattern is *if $<$ condition/event $>$, then action*. In our example process, the text documentation such as *if the credit history is poor then the application is rejected by the clerk*, can be used to extract activity.

- *Information extraction based*: Natural language processing technique is used to extact the verb (*vp*) phrases, noun phrases (*np*), recognize entities depitcing roles, people, and locations. Activities, resources and their roles are identified. A sentence in the process documentation, 'The customer fills relevant details and submits the application' contains two verb phrases : 'fills relevant details', 'submits the application' and one noun phrase with a role 'customer' .

Recent work by Friedrich et al. presents an automated approach of generating BPMN models from natural language text [87]. A sentence is broken down into individual constituent phrases and actions are extracted. In each sentence, grammar relations are analyzed to extract actors, actions and resources. This is followed by a text level analysis to identify the relationship between sentences. Specific text markers such as 'if', 'meanwhile', 'otherwise' are detected as they represent the gateways (conditional, parallel). Actions that are split across sentences are identified. Textual references are used to detect links between actions. As a last step, the flow of actions is determined. The procedure tends to produce models that are 9-15% larger than what is produced by humans. Sinha et al. [88] use multiple NLP techniques (discussed in section 2.9), to transform text to use case description and further to BPMN process model.

2.7.2 Textual Data for Process Analysis

Teinemaa et al. [69], use both unstructured text and structured attributes of cases for predictive business process monitoring. The framework consists of text models and classifiers. For each possible prefix length of the process, one text model to encode features and one classifier is trained. Four different methods of encoding text and extracting textual features is presented. In the reported evaluation, using textual models, enhances the predictive performance of identifying deviant cases.

There have been several efforts on using unstructured textual information available in problem tickets raised during IT application or service maintenance (instance of a service system). There are approaches that use supervised learning to identify the right team or service agents for efficient ticket assignment [75], [74], [89]. Automatic recommendation of resolution for problem ticket based on similar nearest neighbors has been studied [90]. The underlying approach evaluates semantically similar (or meaning similar) past problem tickets to recommend appropriate resolution. Automatically analyzing natural language text in network trouble tickets has been studied by Potharaju et al. [91]. The authors present Netseive, a tool that infers problem symptoms, troubleshooting activities and resolution actions. A framework named ReAct has been presented by Aggarwal et al. [92], that helps IT service agents identify set of actions based on the problem description. The framework uses unstructured text analysis on historical data of incident tickets and guides the agents to find the next best action. Mani et al. [93] use clustering techniques and assign salient labels to group similar problem tickets. They use a combination of latent semantic analysis (LSA, described in 2.9.3) and N-gram extraction to identify phrases or cluster labels.

In this dissertation, textual information from process logs in used to identify contextual information that could impact process performance. This is one of the early works that analyzes textual information, correlates the information with process performance to identify situations that impact process performance during process execution. To date there is no work that uses textual data for discovering context.

2.8 Machine Learning Models

This section details some of the machine learning methods used in my work. There are several techniques that are available and relevant based on the size and the characteristics of the data. Only a small subset of the techniques are detailed in the following sections.

2.8.1 Supervised Learning

Supervised learning problem consists of using the labels or output of a function from a sample data (training data) and arrive at a hypothesis mapping the inputs (or features) to the output labels. The assumption is that, if the learnt hypothesis predicts the values for unseen data (test data), then this hypothesis will be a good representation of the function [94]. There are several supervised learning methods such as support vector machine, linear regression, logistic regression, neural networks and decision trees. Methods such as decision trees are suitable to data containing heterogeneous inputs, i.e. data containing continuous values, discrete or binary values.

Decision trees

Decision trees are commonly used in multi-class classification. While the performance of decision trees is often lower than some of the other widely used supervised learning methods such as support vector machines and neural network based classifiers, decision trees are typically fast to train and easy to interpret. Decision tree partitions the space at every node based on a conditional check of the form $||X - a_0|| < a$, where X is a feature vector, a_0 a fixed vector, and a is a fixed positive real number. Decision trees can also be generalized to branching factors greater than two, but binary trees are most commonly used. To predict the label of any point $x \in X$, the tree is traversed, starting at the root node and going down the tree until a leaf is reached, by evaluating the condition at every node and moving to the right child of a node when the condition is true, and to the left child otherwise. Once the leaf node is reached, the label at the leaf node is the predicted value.

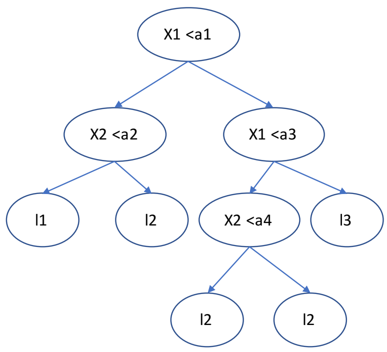


Figure 2.8: Binary decision tree with numerical conditions at each node as described in [94]

Classification and regression trees

Classification and Regression Tree (CART) can be used for both regression (predicting a continuous value) and classification (class labels). CART is a binary decision tree constructed by partitioning the data set at each node, using all predictor variables ($x_i \in X$) and creating two child nodes repeatedly. Different impurity measures are used to decide the predictor variable at each node (misclassification, entropy, Gini index), such that the node impurity is maximally decreased [95]. For any node n, class $l \in [1, k]$, and $p_l(n)$ denoting the number of data points at node n having the class label l, the Gini index is given as: $\sum_{l=1}^{k} p_l(n)(1 - p_l(n))$

Chi-square automatic interaction detection

Chi-square automatic interaction detection (CHAID) is based on the chi-square test association and adjusted significance testing [96]. CHAID tree is built by partitioning the data into two or more child nodes. For any node n, class $l \in [1, k]$, and the pairs of predictor or feature values ($x_i = \{a_1, a_2\} | x_i \in X$), are merged if Bonferroni test (a test suitable when multiple comparisons are required), fails to reject the null hypothesis with a high p-value. CHAID uses multiple splits at each node. CHAID decision tree classifier only accepts nominal or ordinal categorical predictors. When predictors are continuous, they are transformed into ordinal predictors before using the method.

C4.5 Tree

In C4.5 algorithm, at each node of the tree, the splitting criterion is the normalized information gain (difference in entropy). The predictor or feature with the highest normalized information gain is used to make the decision. The tree is pruned by decreasing the size and reducing the estimated error rate [97]. Unlike CART, which is a binary decision tree and the split at each node is binary, C4.5 can have two or more splits at each node. CART uses the Gini index for the splits at each node, while C4.5 uses information-based criteria.

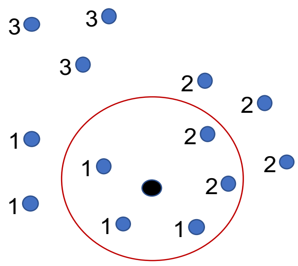


Figure 2.9: k-nearest neighbor based classification with k=5 as described in [98]

K-nearest neighbor classification

The nearest neighbor methods are called memory based methods or lazy learning methods. Given a training set of m labeled data points, a nearest-neighbor method decides that a data point in X, belongs to the same class as its closest neighbors in the training set. A k-nearest-neighbor [98] method assigns data point X, to that class to which the plurality (or majority vote) of its k closest neighbors in the training set belong. Relatively large values of k reduce a noisy classification. But large values of k also reduce the boundary between different classes. The distance metric used in nearest-neighbor methods can be simple Euclidean distance for numerical values of X. Euclidean distance between two values $(x_{11}, x_{12}, \ldots, x_{1n})$ and $(x_{21}, x_{22}, \ldots, x_{2n})$ is $\sqrt{\sum_{j=1}^{n} (x_{1,j} - x_{2,j})^2}$ For discrete variables, (e.g. text classification), other metrics such as the Hamming distance is used. An example of a nearest-neighbor decision problem is shown in Figure 2.9. The class of a training data point is indicated by the number next to it. In this case, the class label of 1 is assigned to the test data point due to the majority of the neighbors.

2.8.2 Unsupervised Learning

In unsupervised learning, the training data does not contain the function values or labels. The problem typically, is to partition the training set in an appropriate manner and make predictions of all unseen data.

Clustering

Clustering partitions or groups similar or homogeneous items. Clustering is performed to analyze very large data sets and is used to identify intrinsic grouping in an unlabeled dataset. There are different types of clustering algorithms: i) Hard or exclusive clustering ii) Overlapping clustering iii) Hierarchical clustering and iv) Probabilistic clustering

K-means

K-means is the simplest and one of the widely used hard clustering algorithms [99]. In this method, a certain (k) number of clusters are predefined. Each cluster has a centroid. First, the centroids are chosen for each cluster. The second step is to find the nearest center for each point and assign it to that cluster. When all the points have been assigned clusters, the position of the k centroids is re-calculated as center of all points in the cluster. After the k new centroids are chosen, second step of assigning clusters to all data points restarts the clustering process. The process continues till the centroids do not move. K-means uses Euclidean distance measure and makes a hard allocation of each point to one cluster. This can often lead to poor solutions. Another key requirement of k-means is the need to specify the number of clusters (k).

Hierarchical clustering

Hierarchical clustering creates a hierarchy of clusters and does not require the number of clusters as input [100]. There are two approaches to hierarchical clustering: 1) Agglomerative clustering, ii) Divisive clustering. In agglomerative clustering, each element is a single cluster at the beginning. At every iteration, the approach merges nearest clusters. The iterations end when all clusters are merged into a single cluster. The resulting tree is called a dendrogram. The tree can be cut at any level to produce different clusters as shown in Figure 2.10. The cut is the maximum distance allowed to merge clusters. Data points with a distance lower than the cut distance are considered as grouped together. In the figure, the cut at distance d_1 results in the clusters $\{1, 2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8, 9, 10\}$, while a cut at distance d_2 results in clusters $\{1, 2, 3, 4, 5\}, \{6, 7\}, \{8, 9, 10\}$. Divisive clustering adopts an opposite approach: initially, there is one single cluster and every iteration splits the cluster till each point becomes a singleton cluster. The resulting tree is again a dendrogram.

There are two types of clustering methods. The *Single Linkage* approach, merges two clusters by considering the minimum distance between the points in clusters to be merged. The *Complete Linkage* approach, two clusters are merged by considering the maximum distance between the points in the clusters. Complete linkage clustering results in more compact clusters as the merge criterion considers all points in the cluster.

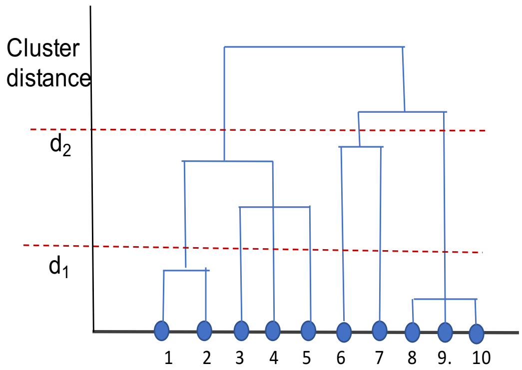


Figure 2.10: Dendrogram of hierarchical clustering

Affinity propagation

Affinity propagation identifies a set of 'exemplars' and forms clusters around these exemplars [101]. An exemplar is a data point that represents itself and some other data points. The input to the algorithm is pair-wise similarities of data points. Given the similarity matrix, affinity propagation starts by considering all data points as exemplars and runs through multiple iterations to maximize the similarity between the exemplar and their member data points. In each iteration, two kinds of messages are passed between the data points. These two messages, as defined in [101] are:

- Responsibility $r(i, j)$, is the accumulated evidence for how well-suited point j is to serve as the exemplar for point i, taking into account other potential exemplars for point i. Hence, it is a message sent from cluster members to candidate exemplars, indicating how well-suited the data point would be as a member of the candidate exemplar's cluster.

- Availability $a(i, j)$ is the accumulated evidence for how well-suited it would be for point i to choose point j as its exemplar. Availability messages are sent from candidate exemplars to potential cluster members, indicating how appropriate that point would be as an exemplar.

The iterations are performed until either the cluster boundaries remain same, or after some predetermined number of iterations. The exemplars and their members are the final clusters.

2.8.3 Recommender systems

Recommender systems (RSs) are methods that provide users suggestions on suitability of items and are widely used in web based applications [102], [103]. Items are a generalization of products, news, music and so on. Recommenders are used to suggest users who lack experience in making choices from a wide variety of alternatives. RS are based on a simple premise that similar users make similar choices and rely on the recommendation of their peers.

Content based recommenders

Content based methods analyze the content or a set of documents and descriptions rated by the user to build the profile of the user. The information in the content is parsed and used to recommend similar items.

Collaborative filtering based recommender system

Collaborative Filtering (CF) is widely used in e-commerce applications to produce personalized recommendations for users [103]. Functionally, CF builds a database of preferences or ratings done by distinct users on specific items. As indicated by Sarwar et al. [104], given a list of m users $U = \{u_1, u_2 \ldots u_m\}$ and a list of n items $I = \{i_1, i_2, \ldots, i_n\}$, each user u_i has a list of items I_{ui}, which are already rated. Here, *rating* is a specific range of real numbers (or a totally ordered set). A CF algorithm predicts rating or preference of an item for a user. This predicted value is within the same scale as the rating values provided by user. There are multiple approaches to predicting ratings using collaborative filtering: i) neighborhood based methods ii) model based methods.

User based neighborhood method, locates other users with similar profiles to that of the user for which the recommendations need to be provided (or the active user). These similar users are commonly referred to as ‘neighbors’. Two users are similar if they have rated items in a similar manner. The rating r_{ui} of user u on item i considers the k nearest neighbors $\mathcal{N}_i(u)$ of user u, who have rated the item i The similarity weight w_{uv} between the active user u and neighbor v, is defined by a similarity measure (e.g., Pearson correlation coefficient). The predicted rating as detailed in [105] is given as:

$$\hat{r_{ui}} = \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} r_{vi}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|}$$

Item based neighborhood method, predicts the rating of an active user u for an item i, based on the ratings of u on items similar to i or $\mathcal{N}_u(i)$. Two items are similar if several users of the system have rated the items in a similar manner [105].

$$\hat{r_{ui}} = \frac{\sum_{j \in \mathcal{N}_u(i)} w_{ij} r_{uj}}{\sum_{v \in \mathcal{N}_u(j)} |w_{ij}|}$$

Model based methods: Here, a predictive model is trained based on the data of the users, items and ratings. The model is used to predict the ratings of the users on new items. There are multiple methods such as use of Support Vector Machine (SVM) classifier [106], Singular Value Decomposition that reduces the dimentionality of the user-item matrix [107].

2.8.4 Evaluation measures

The performance of the classifier such as decision tree is based on an error matrix or a confusion matrix as shown in Figure 2.11. The following are the entries in the confusion matrix:

- True positive (tp): Data points that have been correctly classified as true labels
- False positive (fp): Data points that have been classified true but are false (also known as type-1 error).
- False negative (fn): Data points that have been classified as false and are true (also known as type-2 error).
- True negative (tn): Data points that have been correctly classified as false labels

Commonly used evaluation measures are detailed [108]:
Precision: is the fraction of predicted true classes that are correct.

$$\frac{tp}{tp + fp}$$

Recall: is the fraction of actual or condition true classes that were successfully classified as true.

$$\frac{tp}{tp + fn}$$

Accuracy: is the fraction of correctly classified instances.

$$\frac{tp + tn}{tp + fp + tn + fn}$$

F-measure: combines precision and recall.

$$\beta * \frac{precision * recall}{percision + recall}$$

		Gold Label (Ground Truth)	
		Condition True	Condition False
Predicted Class	Predicted True	True Positive	False Positive (Type 1 error)
	Predicted False	False negative (Type 2 error)	True negative

Figure 2.11: Confusion matrix of a binary classifier

Cross validation

The evaluation measures of a classifier needs to be verified on a set of *unseen* data points. Cross-validation [109], partitions the data into subsets. The enables training of the model on one subset (called the training set), and the validation of the model on the other subset (called testing set). To address variability in the data, the partitioning and evaluation is done multiple times. The two types of cross validation used in this work are:

- Holdout method: the data set is randomly partitioned into two sets d_0 and d_1. The model is trained using one set, d_0 and tested on the other subset d_1. Multiple such partitions are made and the evaluation metrics are averaged.
- k-fold cross validation: the data set is first partitioned into k subsamples of equal size. The training and testing is performed k times. In each iteration,

$k-1$ subsamples are used to train the model (training set) and the remaining subsample is used to validate the model. All subsamples are used for testing over k iterations. The k evaluation metrics are averaged.

2.9 Natural Language Processing

This section briefly describes some relevant natural language processing (NLP) methods that are commonly used and have been applied in this work when mining contextual dimensions from textual data.

2.9.1 Text analysis tasks

A text analysis pipeline consists of a set of tasks that are carried out on textual documents [110]. Some of the tasks, have become standard procedures with several NLP libraries supporting these tasks[b], [c]:

- Sentence segmentation: It is a preliminary step to detect the sentence boundaries and divide the document or text into sentences. This becomes a non-trivial operation due to the presence of punctuation marks that can be used to indicate a period or abbreviation.
- Tokenization: A sentence is broken down into a set of tokens as unique words. Identifying tokens is challenging when there are hyphenation and compound words. Tokenization is language specific.
- Parts of Speech Tagging (POS Tagging): The tokens of a sentence are marked with their relevant parts of the speech (POS), based on the word and its relationship with other words in the sentence. The tagging links words to relevant POS - noun, verb, adverb, adjective, conjunctions and punctuations.
- Stemming and Lemmatization: The objective of stemming and lemmatization is to derive the word's base form as documents use different different forms of a word (e.g - allocate, allocation, allocating). Stemming is a rule based method that truncates the ends of the words. Lemmatization analyzes the words and derives the base dictionary form. (For the word 'see' and 'saw', stemming may return 's' while lemmatization would return the *lemma* 'see' in both cases).
- Stop word removal: Very frequent words can be of little value when processing text as they are likely to appear in all the documents being processed and

[b] http://www.nltk.org/
[c] https://stanfordnlp.github.io/CoreNLP/

contain very little information. These are called stop words. They are filtered from the documents

- Anaphora resolution: This step resolves references of pronouns such as ('it', 'she', 'they') to the relevant items in the document. The items are usually, nouns mentioned in the earlier sentences.

2.9.2 Vector Space Model

"The representation of a set of documents as vectors in a common vector space is known as the vector space model" [100]. The model does not consider the ordering of words. Hence, each document $d_i = \{w_{i1}, w_{i,2}, \ldots, w_{i,n}\}$, where w_{ij} is the weight of the term j in the document i. The weight of the term is computed using a commonly defined *tf-idf* weight. *Term frequency (tf)* is the number of times the term occurs in the document. When the weights in the vector are represented by the term frequency, it is usually referred to as *bag of words* model. *Inverse document frequency (idf)* captures the discriminative power of a term with $idf_t = log\frac{N}{df_t}$ where, N is the total number of documents and df_t is the number of documents containing the term t. A high idf_t would indicate a rare term. In the vector space model, the weight of a term is computed as $tf * idf$. The document converted to vectors, are used for information retrieval, clustering and classification tasks.

2.9.3 Latent Semantic Analysis (LSA)

LSA is a technique for extracting and inferring relations by considering co-occurrence of words in a document or passage [111], [112]. As a first step, the text is represented as a matrix in which each row stands for a term and each column stands for document (term-document matrix). The cell contains the frequency of the term (row), in the document (column). Weighing mechanism such as *tf-idf* are applied. As a next step LSA applies singular value decomposition (SVD) to the matrix [107]. In SVD, a rectangular matrix is decomposed into the product of three matrices.Hence, $X = U\Sigma V^T$, where X is the term-document matrix. Σ is an diagonal matrix. The dimensionality can be reduced by deleting the coefficients of the diagonal matrix. Techniques such as clustering can be applied on the reduced dimensionality matrix.

2.9.4 Latent Dirichlet Allocation (LDA)

In LDA, each document is viewed to cover various topics [113], [114]. LDA assigns a set of topics to each document. The underlying assumption of LDA is that the topic distribution is assumed to have a sparse Dirichlet prior. The sparse Dirichlet prior causes the documents to cover only a small set of topics and topics containing only a small set of frequently used words. LDA is widely used to identify the topics covered by documents and group documents based on the topics.

2.10 Chapter Summary

The intention of this chapter was to introduce the work related to modeling and analysis of resources in business process, at various phases of the process lifecycle. Section 2.1 introduced modeling of resources and specification of resource assignment rules. These activities are carried out during the **design** phase of the process life cycle. Once the process is implemented and executed during the **implementation** phase, information of the process is collected and stored in the event logs during the **monitoring** phase. Next, section 2.2, focuses on the **diagnostics or analysis** phase of the process life cycle, where the event logs are mined. Specific aspects of the process are discovered, such as the organizational structure of the executing process, the task allocation rules implemented in the process, and the behavior of the resources executing the tasks. The next three sections focus on the predictive analytics. Frameworks and approaches for predicting completion time and next activity is discussed (section 2.3). The information available in the logs in used to evaluate bottlenecks and predict process outcomes. Here, some of the work on analyzing service systems to improve routing, staffing and team organization decisions is presented (section 2.4). In section 2.5, we introduce the work done in designing and analyzing business process context and its impact on the process outcome, which is inline with some of the core work presented in this dissertation. A review of existing solutions for resource allocation is presented in section 2.6. Methods that analyze both the structured and unstructured (text) data extracted from the execution logs are covered in section 2.7. The last two sections aim to present relevant machine learning models and natural language processing used in the dissertation (section 2.8, section 2.9).

Chapter 3

Research Methodology

This chapter summarizes the research methodology by discussing the research questions and describing the research approach. Data collected from multiple process execution logs used in this dissertation is presented. The analysis done on the data to process or filter and use it in the experiments is discussed. Detailed description of the problem, evaluation of the data and experimentation is presented in individual chapters.

3.1 Research Questions

The goal of the dissertation is to use data from existing process event logs and improve task allocation decisions:

The main research question *"How to model, extract and analyze contextual information of a business process in order to improve task allocation and realize process outcomes or goals?"*, is broken down into four specific research questions:

Research Question 1: *How does resource efficiency vary with case attributes, resource behavior (that manifests as context), further impacting task allocation?*
It is important to understand variance in resource efficiency with resource behavior such as competence, preference, skills and workload. The variance in resource efficiency may cause a failure in meeting the process outcome. For example, when a *less* important task is given to a *highly experienced* resource, the resource efficiency may not improve as one would expect because the resource may not *prefer* to work on less important task. Here, importance of task is a case (or process instance) attribute, resource preference is a resource behavior. Another example of variance in resource efficiency could be with the time of the day. Resources working at different times and working shifts may have different efficiency. The interplay of resource behavior, case attributes, contextual factors (time of the day), and their impact on efficiency is important for an effective task allocation. As a first step, event logs are

used to extract resource efficiency, case attributes and resource behavior. Statistical tests are used to validate significance of variance in the efficiency of resources based on case attributes and resource behavior. Data from three teams is collected and variance in the efficiency based on multiple case attributes and context is validated. Different resource efficiencies are used as input to a simulation model representative of the business process. The simulation model is used to evaluate the impact of context and resource behavior on resource efficiency and performance outcomes. This research question and the solution are detailed in Chapter 4.

Research Question 2: *How do we support task allocation based on process context, for pull based dispatching scenarios?*
The objective of answering this question, is to use the knowledge of variance in resource eciency when recommending tasks to resources. In a pull based dispatching system [35] (Figure 3.1), resources are made aware of the tasks in the system that are placed on a priority queue, prioritized by case attributes such as time of creation of task, urgency or importance. Each individual resource selects a set of tasks from the queue. Choosing the right set of tasks to execute and meet the performance goal, is a knowledge that resources gain through experience. A wrong judgement, could lead to re-work, delays or a risk of not meeting the performance goal. By evaluating attributes of the case, resource and the context based on past execution logs, an eective recommendation can be made, to enable resources select the right task and execute. Event logs are used to identify resource behavior (context), resource and task. The case attributes and outcomes are extracted from the logs. All the information extracted from the event logs is used to build a system that recommends and ranks tasks suitable to each resource. The research question and the solution are detailed in Chapter 5.

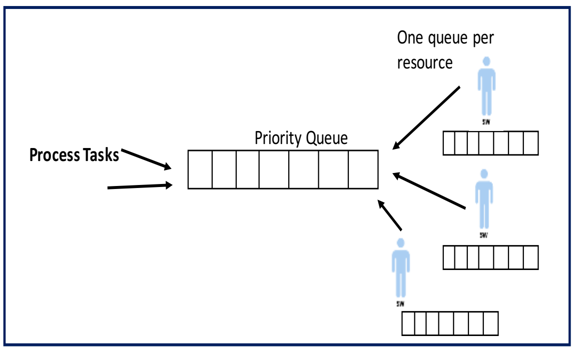


Figure 3.1: Pull based dispatching for task allocation

Research Question 3: *How do we learn task allocation rules based on process context, for push based dispatching scenarios?*

The third goal is to derive dispatching policies and assist the dispatcher of tasks (human or system), in a push based dispatching system (Figure 3.2). In a push based dispatching system, a central dispatcher takes the decision of identifying the right resource for the task [35]. The objective is to effectively allocate task considering context, case, and resource attributes. By using event logs, case attributes, resource attributes, and resource context are extracted and a machine learning model is built. The approach explored in this dissertation, is useful in scenarios where the historical data for each individual resource is sparse or unavailable (personalized recommendations is not possible), and resources change often. In such scenario, resources can be characterized by certain resource attributes and policies can be (machine) learnt for those resources. This research question and the solution are detailed in Chapter 6.

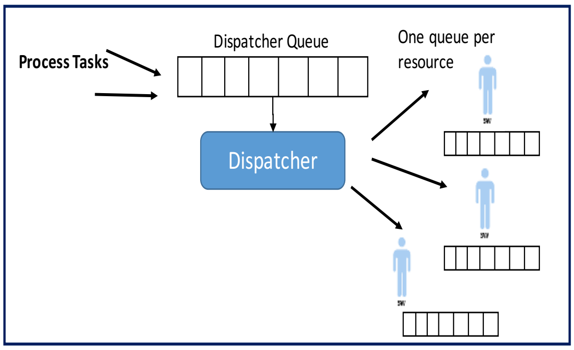


Figure 3.2: Push based dispatching for task allocation

Research Question 4: *How do we identify useful contextual information from process data?*

The final goal is to explore the possibility of extracting contextual information from process data. The notion of implicit context is introduced. Implicit concept refers to external situations that are not specified by domain experts, and need to be discovered. Textual logs, captured by resources during the execution of task can be a source of information, containing various situations that occur during process execution. Some of the information recorded may be standard procedures that are specific to the task execution, while others may represent external factors influencing the process or task outcomes. One such example could be that, the completion of a task requires more details from the customer and not having sufficient detail could lead to waiting for inputs or poor quality of task. Identifying these situations or process context, impacting performance outcome, would be useful in triggering a process enhancement. In the example stated, providing well defined templates to describe the task prior to the execution of task could be a possible improvement. An approach to identify context and its impact on process outcomes would be useful. This research problem is explored and the solution is described in Chapter 7.

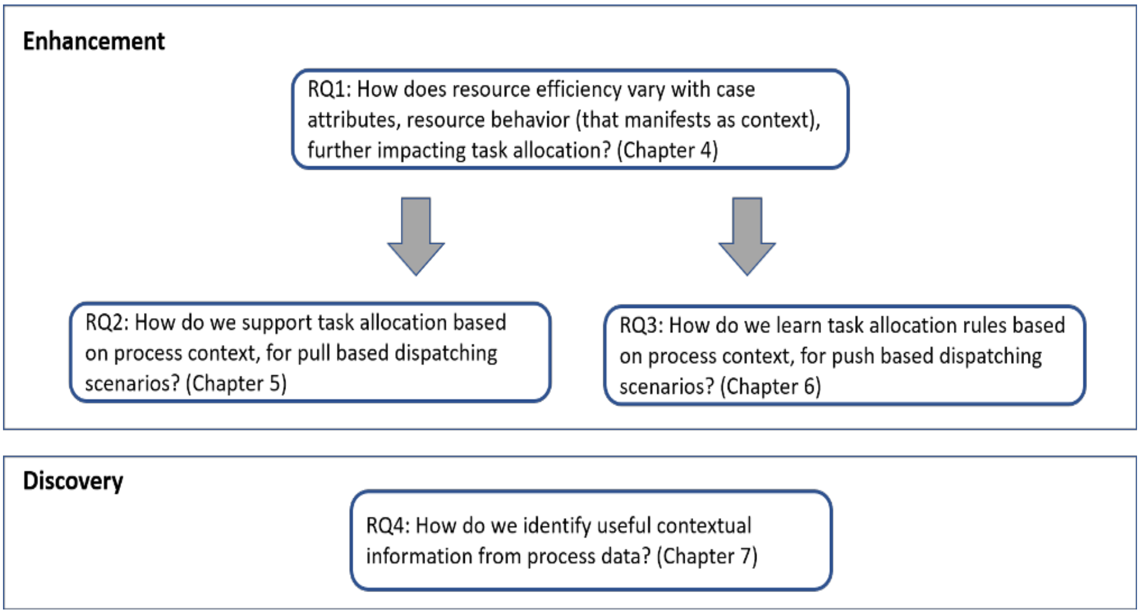


Figure 3.3: Conceptual connection of research questions

Figure 3.3 conceptual connections between the research questions. Each box represents the research question, and the corresponding chapter that addresses the research question. The purpose of RQ1, RQ2 and RQ3 is to analyze events logs and improve existing resource allocation decisions. Hence these reflect *enhancement* of the existing process. RQ1, validates the hypothesis of variance in resource efficiency with case and contextual factors. RQ2 and RQ3 answer the questions on using contextual factors to improve resource allocation decisions. RQ4 results in a *discovery* method with an objective of using an event log to extract knowledge having no a-priori information.

3.2 Research Approach

The dissertation makes use of a quantitative research approach [115]. Quantitative research approach is best suited as research questions being addressed call for (a) evaluating factors that influence resource efficiency, (b) understanding if contextual factors improve prediction of resource allocation decisions, and (c) analyzing the improvement gained by using contextual factors or resource behavior.

To address the first research question, data from three teams involved in supporting problems that occur in operating systems, is collected for a period of three weeks. Case specific factors, resource context (or behavior such as expertise), and resource efficiency are extracted from the logs. Statistical tests, to evaluate the variance in efficiency of resources under the influence of different contextual and case related factors, are conducted.

The second question is addressed by modeling contextual factors and building a prediction model suitable for pull based dispatching. A context-aware recommender system is used as a prediction model, as it considers context in addition to resources and tasks. Resource efficiency is defined as the target variable or the dependent variable. Contextual and task specific factors are used as independent variables. The approach and model predictions are evaluated on two real-life process event logs by extracting relevant contextual factors and resource efficiency. Resource efficiency is predicted with and without context and the improvements gained by using contextual factors are evaluated.

To address the third research question, two prediction models are evaluated: i) decision tree and ii) K-nearest neighbor. Both the models are trained with contextual factors and case attributes as input features. The process completion time is considered as the dependent or target variable. Evaluation is done on a synthetic data set to identify contextual factors impacting the completion time. Measures of importance of contextual variables provided by decision tree model, is used to identify the influence of context on the process completion time (or the process performance).

The final research question is tackled by using natural language processing on unstructured logs that are recorded by resources when working on tasks. In this work, unsupervised machine learning method is used to group textual messages. Groups or clusters of textual data are correlated to the process performance or process completion time. Clusters with significant variance in completion time indicate underlying common characteristics leading to better or poor completion time. The textual information in the clusters can be manually verified to identify contextual situation. This is a first step in exploring use of textual logs of real-life process, to extract contextual factors.

3.3 Data Collection

The dissertation uses multiple process event logs extracted from real-life processes to address the research questions. The use of multiple event logs further justifies the influence of context in different processes, and confirms the generalizability of answers to the research questions. In Chapter 4, the first research question (RQ1), is addressed by using data collected from 60 users working in three teams resolving IT service incidents. The data collection for a period of 3 weeks, used the organization's proprietary tool. The tool enabled a user to log the time stamp when starting and completing each task. In addition, the expertise of the resource and the type of task was recorded. Each type of task was associated to a specific complexity and priority (or urgency). For example 'updating antivirus on a server' is a complex task, while 'creating user account' is a simple task (see Table 3.1). The IT service incident priority, complexity is a case attribute, the expertise is the resource behavior or context of the resource solving the incident. The mapping of a task to the complexity of a task was provided by the IT service experts.

User	Task Description	Complexity	Priority	User Expertise	Start time	End Time
UserX	Install Printer Driver	HIGH	LOW	LOW	1/04/2013 23:57	2/04/2013 2:46
UserY	Manage Anti-virus	HIGH	LOW	HIGH	12/04/2013 4:36	12/04/2013 6:16
UserX	File reconfiguration	LOW	LOW	LOW	1/04/2013 4:03	1/04/2013 4:30

Table 3.1: Data collected from users to evaluate variance in resource efficiency with resource behavior and task attributes

Data from two real-life processes [116], [117], were processed and evaluated to address research questions RQ2 and RQ3 in Chapter 5 and Chapter 6.

A sample of the process event log of the financial institute is shown in Table 3.2. The log contains activities related to a loan application process. For each loan application or case, the loan amount, the activity, the resource performing the activity, status of the activity and the time stamp is recorded. Hence, the data contains information about the case (loan amount), the resource working on a task, and the time spent by the resource on the task. Previous work has analyzed the log to discover the control-flow and the time spent by the resource on tasks (resource efficiency) [118]. The data enables extraction of case context, resource behavior and resource efficiency. Hence, in this dissertation, the event log was used to extract contextual information, resource efficiency to address RQ2 and RQ3.

Table 3.3 refers to the sample of another real-life process event log capturing IT management process of software system[a]. Each case represents a service request raised for a product. The resource, is the owner responsible for working on the service request. Each owner belongs to a team and organization. The status and the sub-sratus represent the states of the service request. The request can be i) 'Queued', when waiting for a owner to start working, ii) 'Accepted', when an owner is working on the request and iii) 'Completed', when the service request has been resolved. Hence, the log contains information about resources working on the tasks, their organizational roles and the time spent by the resources on the task (difference in the time stamp between the 'Complete' and 'Queued' status of a service request).

[a] `http://www.win.tue.nl/bpi/lib/exe/fetch.php?media=2013:vinst_data_set.pdf`

The event log can be used to extract case, context and resource efficiency. Hence it is suitable for answering RQ2 and RQ3.

Case Id	Loan Amount	Activity Name	Resource	Status	Time stamp
183175	15000	Nabellen incomplete dossiers	10138	SCHEDULE	2011-12-14 09:07:37
183175	15000	Nabellen incomplete dossiers	10899	START	2011-12-14 09:19:00
183175	15000	Nabellen incomplete dossiers	10899	COMPLETE	2011-12-14 09:20:51

Table 3.2: Financial institute process log containing case, task and resource information

SRNumber	Time	Status	Sub Status	Organization	Team	Impact	Product	Owner
1-364285768	2010-03-31 16:00:56	Accepted	In Progress	Org line A2	V30	Medium	PROD582	Frederic
1-364285768	2010-03-31 16:45:48	Queued	Awaiting Assignment	Org line A2	V5 3rd	Medium	PROD582	Frederic
1-364285768	2010-04-06 15:44:07	Accepted	In Progress	Org line A2	V5 3rd	Medium	PROD582	Anne Claire
1-364285768	2010-04-06 15:44:38	Queued	Awaiting Assignment	Org line A2	V30	Medium	PROD582	Anne Claire
1-364285768	2010-04-06 15:44:47	Accepted	In Progress	Org line A2	V13 2nd 3rd	Medium	PROD582	Anne Claire
1-364285768	2010-04-06 15:44:51	Completed	Resolved	Org line A2	V13 2nd 3rd	Medium	PROD582	Anne Claire
1-364285768	2010-04-06 15:45:07	Queued	Awaiting Assignment	Org line A2	V30	Medium	PROD582	Anne Claire
1-364285768	2010-04-08 11:52:23	Accepted	In Progress	Org line A2	V30	Medium	PROD582	Eric
1-364285768	2010-04-08 11:53:35	Queued	Awaiting Assignment	Org line A2	V5 3rd	Medium	PROD582	Eric
1-364285768	2010-04-20 10:07:11	Accepted	In Progress	Org line A2	V5 3rd	Medium	PROD582	Anne Claire
1-740847897	2012-05-04 22:10:26	Queued	Awaiting Assignment	Org line C	G76	Medium	PROD383	Siebel
1-740847897	2012-05-04 22:13:09	Accepted	In Progress	Org line C	G76	Medium	PROD383	Michael
1-740847897	2012-05-04 22:15:22	Completed	Resolved	Org line C	G76	Medium	PROD383	Michael
1-740847897	2012-05-12 00:12:38	Completed	Closed	Org line C	G76	Medium	PROD383	Siebel

Table 3.3: IT incident log containing case, resource organization and resource information

Finally, to answer RQ4, a real life textual log of an IT application maintenance process was used where users logged comments. The data for a period of three months was used. For each case, textual logs were extracted and collated from all the tasks of the case. Table 3.4 illustrates few example of comments made by resources. Details of the process, the characteristics of the textual logs are described in Chapter 7.

3.4 Data Analysis

This section covers the processing of data extracted and analyzed from the event logs. The goal of data analysis was to remove incomplete information and noise that could lead to erroneous inputs to the prediction models:

No.	Communication log of the problem tickets recorded by knowledge workers
1	emailed user. *waiting for user to get back to me.* emailed user. looking for response. User confirmed that the issue is not replicated. Hence closing the incident.
2	Left a voicemail for customer at the number provided in this ticket. Requested he call option (one) for further assistance. Validated userid in the portal, made in Synch . Manually made in SYNC with that of GUI. Call made both on office phone and cell. *Voice sent on cell and office phone is not reachable.* 2nd call made to the customer. No response.. *3rd call made to the customer.* No response. Call closed due to no prior response from the customer.
3	Peformed netmeeting with user and there are no authorization issues. user is able to run the reports. *Training issue.*
4	called, Attributes corrected & mail send to user
5	Received confirmation from user, closing the incident.

Table 3.4: Unstructured textual information captured during IT application maintenance process

- *Outliers in task completion time:* The time taken by a resource to complete the task can be computed from the event log by considering the time stamp and status of the task. In each of these logs, certain log entries have very low or very high completion times. First, a logarithmic transformation was applied on the completion time. As the completion times follow a lognormal distribution, this transformation helps in visualizing the completion time using box plot. Outliers were identified using the graphical technique of inspecting the box plot. The mean (μ) and standard deviation (σ) was computed. The completion times (ct) considered for analysis are: $min(\mu - 3\sigma, 2) \leq ct \leq (\mu + 3\sigma)$. Very low completion times (2 minutes), were filtered as tasks with such low completion times would not be representative of a knowledge based task. Similarly, tasks with high completion time were filtered.

- *Missing information:* Event log can be incomplete. The financial institute loan application event log contained several events with missing resource information. Such events were filtered. Cases or tasks that were incomplete i.e. did not have a status indicating completion were filtered. As existing logs were used, missing information could not be rectified and hence, were filtered.

- *Multi-user tasks:* In the IT incident management log [117], certain service requests were handled by multiple users as show in first ten rows in Table 3.3. As these tasks have no additional information, it is not possible to identify number of resources required to work on multi-user tasks. Further, the computation of time spent by each resource on the task would not be accurate as the status updates in these tasks with multiple users has variations. Predicting number of resources required or the time spent by the resource is a challenge. Hence, in the prediction models, only tasks accepted and completed by a single user was considered. Data related to multi-user tasks was used to compute resource behavior metrics such as preference and utilization. However, the models were built to predict completion time of tasks completed by a single user.

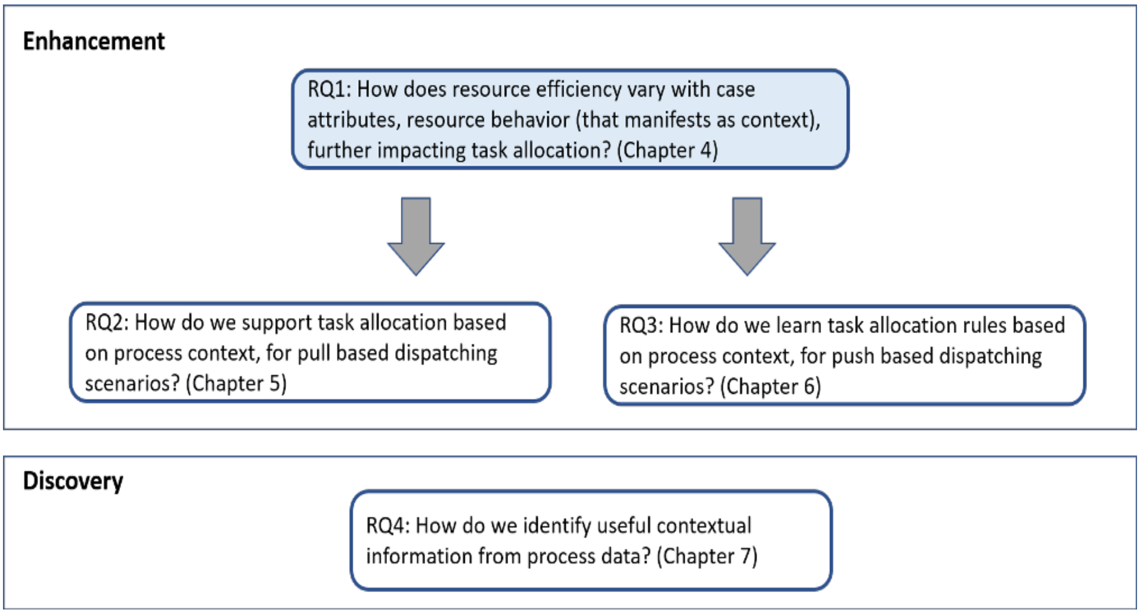
- *Automated tasks:* In all the event logs, certain tasks are performed by the system. For example Table 3.3 has 'Siebel' as a resource representing a software application. Similarly, the financial institute loan application event log has several tasks performed by the banking system. These events were filtered during the data analysis and processing phase. The prediction models were trained and tested by considering tasks done by human resources.

3.5 Limitations of the Method

In this dissertation, existing event logs were used. These logs were extracted from process aware information systems that had been implemented and were executing the process. Hence, missing or additional information could not be corrected or collected. In addition, there was no access to experts or process owners. The domain information was limited to available documentation of the processes. Additional domain information or expert feedback could not be used in the study. Generic factors and available domain factors were used in modeling context and other case attributes. Hence, the predictive model results present a comparative study of improvement gained in resource allocation with or without using contextual factors. Addition of domain knowledge and features could lead to further improvements in the performance of the models.

Chapter 4

Data-driven Task Allocation and Staffing



Existing body of work has explored the influence of resource efficiency on the cost and performance of a process [8], [23],[47]. Further, there have been studies analyzing resource behavior and their influence on resource efficiency [12], [61], [60]. While the need to consider resource behavior for task allocation has been recognized [60], limited attention has been paid to analyze the behavior of resources and its impact on task allocation. This chapter presents an approach of analyzing the variance in efficiency of resources based on multiple factors that include case attributes and resource behavior. The variance in efficiency of resources is used as input to a simulation model to determine the staffing required to meet the process performance. The output staffing solution without considering resource behavior is compared with the staffing solution that considers resource behavior, to answer the research question.

4.1 Introduction

A key characteristic of Knowledge Intensive Business Services (KIBS) [119] is its reliance on knowledge of workers, for delivering services to customers. "KIBS serve as service providers for knowledge intensive business processes (KIBP)" [120]. Hence, KIBS serve KIBP [121] of multiple customers. The quality and cost of the service delivered depends on the expertise of the workers involved. In IT infrastructure management services (a specific class of KIBS), there are several processes defined to ensure smooth operation and management of the customer's infrastructure. For example, the incident management process consists of activities to quickly restore normal service operations in the event of failure. Apart from being process intensive, the operations tend to be resource intensive as well. Hence, it is important to evaluate the efficiency of resources, allocate tasks to relevant resources and optimally staff the teams delivering services.

The focus, in this dissertation is on an IT Incident Management Process where the failures or events are reported by customers as Service Requests (SR). The service organization managing the processes is the service provider, and has a team of service workers (SW) who deliver the services. The time taken (completion time) to restore the service or resolve a SR is a critical performance metric, and hence is closely monitored within the IT management process. Typically the contracts specify a minimal percentage of SR (i.e X%) in a month that must be resolved within a target completion time (i.e. Y hrs). On a breach of the terms in the contract, the provider is liable to pay penalties. Hence keeping completion times within contractual target times is the most critical performance metric of this incident management process. Several factors affect completion times in an IT incident management system. The completion time of a SR depends on the (a) queue waiting time in the system and (b) the service time of the worker (time required to complete a single unit of work). The queue waiting time in turn depends on the amount of work that exists in the system and the resources available for doing that work. In case of an under-staffed system, all workers are busy and the queue waiting times are higher. This leads to overall higher completion times. The service time of the worker on the other hand is independent of the amount of work in the system, and depends on factors such as the worker expertise and the type of request. The focus in this study, is on the factors impacting the service time of the worker and their impact on the optimal staffing of the service system.

The service time of a worker is known to depend on the expertise of a worker gained through experience [122], [123]. Prior studies also indicate that the service times vary with work complexity. Complex work requires more time than simple work [124]. The priority of the work plays a key role as the target completion times varies with the priority of work. A high priority SR has lower target completion times. In this dissertation, additionally the service time of the workers, is analyzed in the context of the three factors: i.e., on (a) complexity of work (b) the minimum expertise level of the worker required for a work and (c) importance or priority of the work. It is observed that, *while experts have lower service time than novices for complex work and important work, they tend to have the same efficiency as novices for less important work.* The insights gained, are used to make informed skill-based staffing decisions within the incident management process. A simulation model is built to account for the behavior of experts and novices for varying work complexity and priority. A search based optimizer uses the simulation model to arrive at an optimal staffing.

This dissertation demonstrates that data-driven techniques similar to the work presented here, can be useful in identifying policies governing the optimal matching of service worker to service requests. It further illustrates that the efficiencies of service workers or human resources in any process, depends on multiple factors that go beyond the role or availability of the workers.

The intent here is not to suggest that the specific findings about the correlation between service worker and request profiles should work in all organizational settings and in all instances. Indeed, the validity of these specific findings is restricted to the specific organizational context. These might potentially not hold even in other parts of the same organization. However, the results presented serve as the basis for methodological guidelines on how data-driven analysis can lead to more effective allocations of workers to tasks.

4.2 IT Incident Management Process

This section provides an overview of the IT incident management process of the service system under study. Commonly used concepts of a service system supporting the incident management process are defined.

Figure 4.1 illustrates an incident management process. A problem or issue faced by a customer or a business user is **reported as an incident** into an incident management system. The dispatcher reviews the incident and **evaluates the complexity and priority of the incident**. The dispatcher further **identifies a service worker** suitable for resolving the incident. This task is based on specific rules and policies and hence is a rule based activity. The dispatching rules are described in Table 4.1. In the IT service system under study, workers are broadly categorized into two distinct classes: experts or experienced service workers and novices or less experienced service workers. If an incident is complex, an expert service worker is assigned the incident and if the incident is simple, a novice service worker is given the incident. An alternate dispatching policy applies when none of the novice workers are free i.e. all are busy resolving other incidents. In such a scenario, a simple ticket is assigned to a free expert worker. The worker assigned to the incident, **resolves the incident**. Once an incident is resolved, the business user validates and confirms the service provided by the worker and **closes incident**.

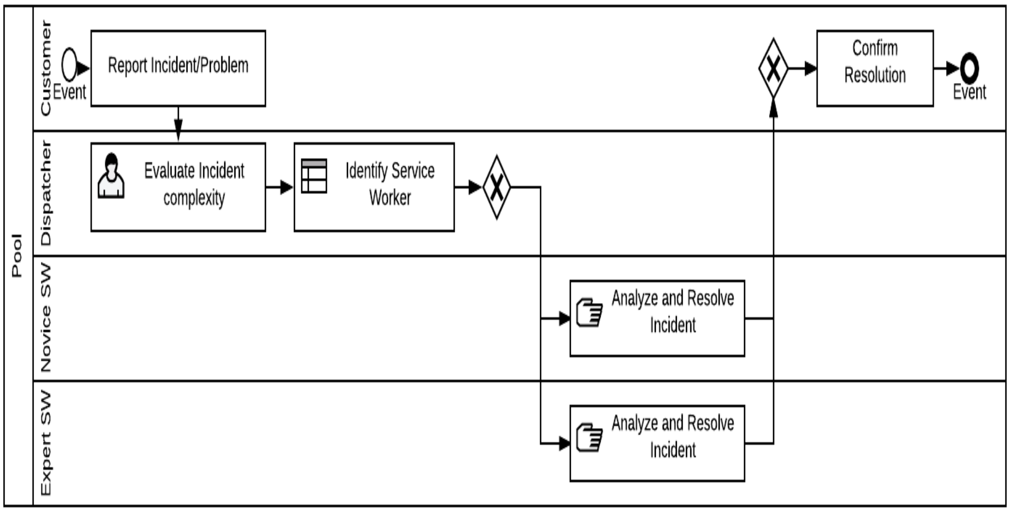


Figure 4.1: IT Incident management process

Dispatching Policy in teams
if (complexity *isLow*) and if (novice *isAvailable*) → assign to novice
if (complexity *isLow*) and if (**not** novice *isAvailable*) and if (expert *isAvailable*) → assign to expert
if (complexity *isLow*) and if (**not** novice *isAvailable*) and if (**not** expert *isAvailable*) → wait in queue
if (complexity *isHigh*) and if (expert *isAvailable*) → assign to expert
if (complexity *isHigh*) and if (**not** expert *isAvailable*) → wait in queue

Table 4.1: Dispatching policies

Table 4.1 contains the rules that dispatcher uses to identify a suitable service worker. Typically, the staffing of the teams supporting the incident management process described in Figure 4.1, is based on the complexity of the incident. If a large percentage of work is simple and can be done by less experienced workers, then a large percentage of the team will be staffed with less experienced workers. Similarly, a large percentage of complex work requires higher number of experts. Figure 4.2 shows the distribution of experts as compared to the distribution of high complexity work in ten teams within an organization supporting the incident management process. There is a positive correlation between the percentage of complex work and the percentage of experts in the team. The objective of the following sections in the chapter is to show, that the staffing of teams need to be based on other factors, in addition to complexity of the incident.

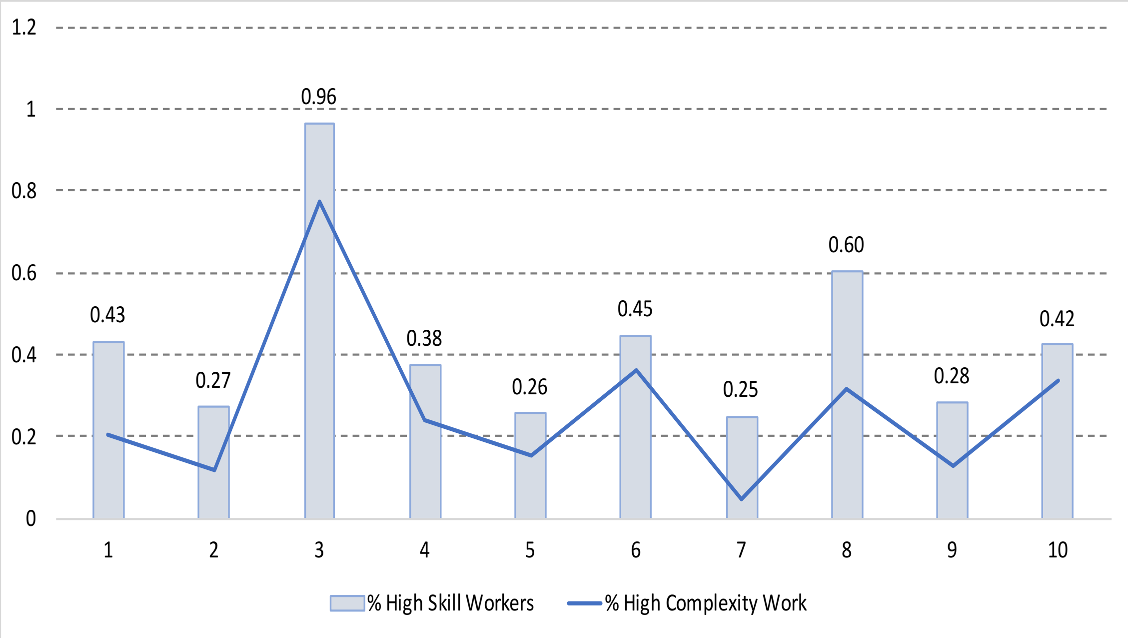


Figure 4.2: Percentage distribution of novice workers and low complexity work

4.2.1 Concepts in the Service System

The key concepts underpinning the service system are defined below:

Incident or Service Request Incidents or service requests constitute inputs to the service system and are handled by service workers. Each incident is characterized by its complexity and priority.

Complexity The complexity of an incident is indicative of the "level of difficulty". A finite set of complexities levels X are defined. A complexity level is associated with each incident.

Priority The priority of an incident indicates the urgency and impact of an incident. A finite set of priorities levels P are defined. A priority level is associated with each incident. A higher value of priority indicates that the incident is important and needs faster resolution.

Work Arrivals The arrival pattern of service requests is captured for finite set of time intervals T (e.g. hours of a week). That is, the arrival rate distribution is estimated for each of the time intervals in T, where the arrival rate is assumed to follow a stationary Poisson arrival process within these time intervals (one hour time periods) [125], [73].

Service Time Service time refers to the time taken by the service worker to handle the incident. This refers to the time interval between the time a service worker picks up the incident and the time the service worker resolves the incident. In the Figure 4.1, the service time is the time spent in the activity “Resolve Incident”.

Completion Time Completion time of an incident refers to the time elapsed between the generation of the incident by the customer and the completion of the process of handling the incident. The completion time includes the time an incident waits in the queue for it to be dispatched by the dispatcher to a service worker.

Expertise Expertise of a service worker is based on skill gained through experience. Service workers are categorized into a finite set of expertise levels L.

A mapping $\beta : X \rightarrow L$ is a map from the complexity of work to the minimum expertise of service worker required to support an incident. This mapping is used by the dispatcher to evaluate the complexity and decided the expertise of the SW capable of working on the incident. An expert is capable of resolving service request or incidents of all complexities.

Service Level Agreements (SLA) SLA measure of outcome of service. SLA are given for each customer and priority pair as $\gamma_{ip} = (\alpha_{ip}, r_{ip}), \alpha_{ip}, r_{ip} \subset \mathbb{R}$ is a map from each customer-priority pair to a pair of real numbers representing the SR target completion time and the percentage of all the SRs in a given time period (such as a month), that must be completed within this target time. For example, $\gamma_{customer_1, P_1} =< 4, 95 >$, denotes that 95% of all SRs from $customer_1$ with priority P_1 in a month be completed within 4 hours.i.e. completion time of 95% of the requests of the $customer_1 \leq 4$ hours.

4.2.2 Service System Model for Staffing

There are several complexities involved in modeling a service system as described by the authors in [23]. First, the incidents or service requests are differentiated by their complexities and priorities with request arrival rates varying over hours and days of the week. Second, the service levels vary for each customer and priority of the incident. Finally, the service times of the workers is dependent on multiple factors that are evaluated through the empirical study in this dissertation. Due to these inherent complexities, a simulation based modeling and optimization framework is used, to determine optimal staffing levels. For simplicity, in the optimization model, a service system supporting one customer is considered. It can be easily extended to support multiple customers by considering different service levels and different volume of requests per customer. The optimization model defined in [23] has been adopted for arriving at the number of workers at each expertise level to meet the service level agreements at minimal costs. The optimization model is described in brief:

- p, the set of priorities of a service request $p := \{1, 2, \ldots, P\}$
- x, the set of complexities $x := \{1, 2, \ldots, X\}$
- l, the set of expertise levels; $l := \{1, 2, \ldots, L\}$
- n_l, the set of workers with expertise level l
- $\overline{n_l}$, the upper bound on the number of workers with expertise level l
- $\underline{n_l}$, the lower bound on the number of workers with expertise level l
- c_l is the cost of a service worker with expertise level l
- v^t_{px} is the volume of requests in the period t with priority p and complexity x
- s_{pxl} is the service time for a request with priority p, complexity x and assigned to worker of expertise l
- β_{xl} is valued 1 if request of complexity x can be addressed by an expertise level l and 0 otherwise
- α_p is the target attainment for priority p during a measurement time
- r_p is the target resolution time for a request of priority p.

Objective Function and Constraints

The objective of the staffing solution is to minimize the cost of the service system as defined:

$$minimize \sum_{l \in L} n_l c_l \tag{4.1}$$

such that,

$$f_p(v^t_{px}, s_{pxl}, \beta_{xl}, n_l, r_p) \leq \alpha_p \tag{4.2}$$

$$\underline{n_l} \leq n_l \leq \overline{n_l} \tag{4.3}$$

Equation (3.1) is the staffing cost of the solution. Equation (3.2) is the constraint indicating service level agreements must be satisfied. The function f_p is computed by the simulation model which indicates if the attainment level α_p is met. Equation (3.3) is the restrictions set on the minimum and maximum staffing levels set for the solution.

The simulation model uses discrete event simulation to generate service request of defined priorities and complexities. The service time of the workers are based on their expertise levels, priority and complexity of the work. The outcome of the simulation model is the service level attainment considering all the factors described in function f_p.

4.3 Data Setting and Parameters

In this section various factors that impact the service time of a worker are presented. Further, the service time parameters are used in the simulation model to evaluate the impact of these factors on the task allocation and staffing.

4.3.1 Setting

Data collected from three teams within the organization is used for the study. All the three teams are involved in managing incidents of the operating systems (OS) domain, i.e manage OS of servers of customers. The data on service time (worker productivity) is collected for a period of three weeks by recording only the time on the task of resolving the incident. There are a total of 60 workers across the three teams. Service time data from approximately 4000 incidents is analyzed. For each incident, the complexity, priority, expertise of the assigned worker and the service time is extracted.

Dependent Variable

The service time is examined as the dependent variable. *Service Time* is used to evaluate productivity of a worker. As indicated in earlier studies [124], service time follows a log normal distribution as seen in Figure 4.3. The mean service time is 40.33 minutes and the standard deviation is 37.29.

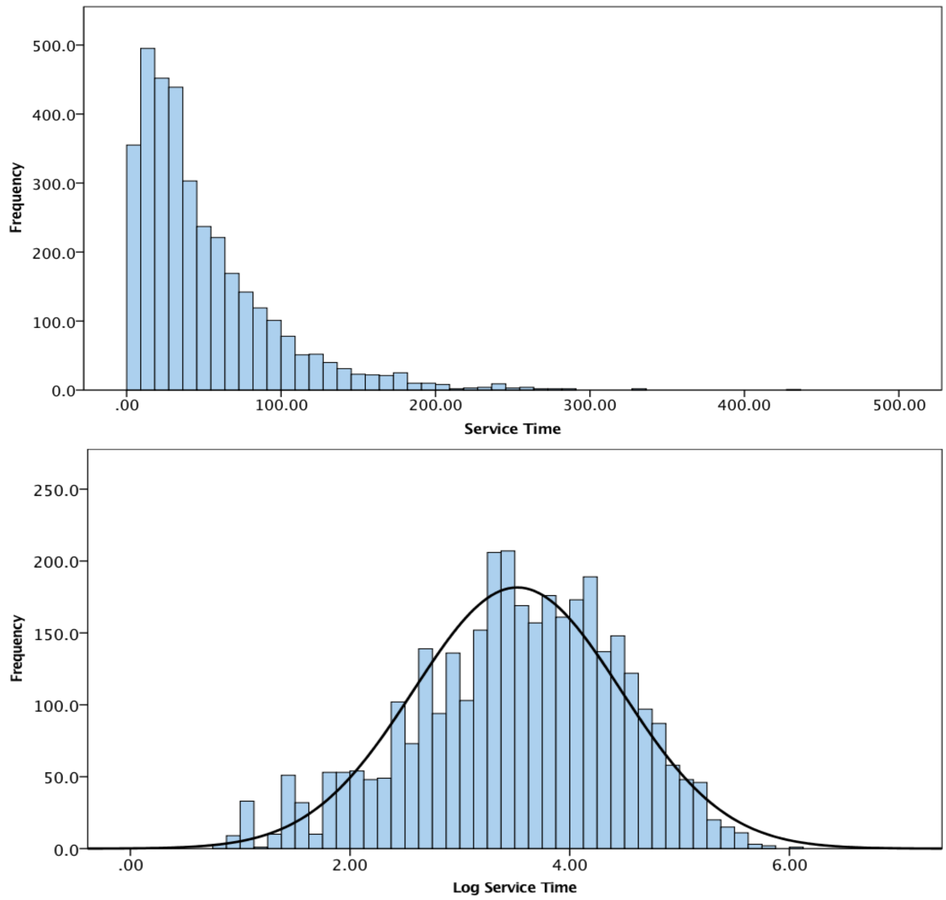


Figure 4.3: Service time distribution

Independent Variables

Complexity of incident, priority of incident and expertise of the worker are chosen as the independent variables.

Expertise The expertise of the workers in the team is based on the experience of the workers - novice with < 2 years experience, experts with > 2 years and < 7 years experience. Of the 60 workers, 20 are novices and 40 are experts. An expert is referred to as having a 'High' expertise and novice having 'Low' expertise.

Complexity The complexity is determined by the dispatcher. Incidents range from handling password reset requests (simple) to verifying security compliance of a server (complex). Two levels of complexity are considered: Simple and Complex. Simple work can be assigned to novices or experts. It is observed that 50% of the simple incidents get resolved by experts. While it is not preferable to assign complex work to novices, in the data collected across teams, it is observed that 10% of the complex incidents are assigned to novices.

Priority Priority of an incident determines its urgency and importance. There are 4 levels of priority - Very High(VH), High(H), Medium (M) and Low (L). VH priority incidents are rare and are always treated as exceptions. The low priority incidents also form a small percentage and since their service levels are relaxed, these incidents rarely need to get assigned to a higher skilled worker. i.e. a simple work is assigned to a novices even if they are busy as they have relaxed target time. Hence, in this study, the focus is on High and Medium priority tickets.

4.3.2 Model Parameters

The work arrivals, complexities, priorities, service time, cost, and expertise of service workers in the dataset are used as input parameters in the simulation model:

- The finite set of time intervals for arriving work, denoted by T, contains one element for each hour of week. Hence, $|T| = 168$. Each time interval is one hour long.
- Priority Levels P: Two levels of priority are considered $P = \{High, Medium\}$, where $High > Medium$.
- Expertise Levels L: Two different levels of expertise simulated $L = \{Low, High\}$, where, $High > Low$.
- Complexity Levels X: Two different levels of complexity are considered $X = \{Complex, Simple\}$ where, $Complex > Simple$
- Cost: The cost of a worker depends on the expertise. The cost of an expert is considered to be 50% higher than the cost of a novice.
- Service Time: The service time of the request s_{pxl}, is used as input to the simulation model (based on analysis done in section 4.4)

The model parameter values of priority levels, expertise levels, complexity levels, cost, service time and arrival of service requests used in the simulation model are computed from the data.

Table 4.2 shows the distribution of requests based on the priority, the service level target times and the percentage target levels that are used in the model.

Priority of Incident	Percentage Distribution	Service Level Target Times (minutes)	% Meeting Target Time
VeryHigh	2	240	95
High	20	480	95
Medium	75	720	90
Low	3	1440	90

Table 4.2: Work distribution and service level target times and percentages

4.3.3 Implementation

The implementation of the IT incident Management process model is built using the AnyLogic simulation software[a] [126], which supports discrete event simulation technique. It also provides optimization package that uses intelligent search procedures in scatter search combined with Tabu search metaheuristics [127], [128]. Forty weeks of simulation runs are performed. Measurements are taken at end of each week. No measurements are recorded during the warm up period of first four weeks. In steady state the parameters that are measured include:

- SLA measurements at each priority level
- Completion times of work in minutes (includes queue waiting times and service times)
- Resource utilization (captures the busy-time of a resource)
- Number of resources that is an indication of cost

For all the above parameters the observation mean and confidence intervals are reported. Whenever confidence intervals are wider, the number of weeks in simulation is increased and reported values in the dissertation are within confidence intervals. The simulation model further, does not dispatch a high complexity work to a novice. All the results consider scenario where an expert can do a simple work but a novice doesn't do a complex work in line with the real-life dispatching policy.

[a] http://www.xjtek.com/

4.4 Service Time Analysis and Staffing Solution

4.4.1 Impact of Work Complexity on Service Time

A commonly used approach in practice is to profile the service time of workers based on the complexity of requests assigned [124]. Table 4.3 shows the difference in means of service time and their confidence intervals, with complexity of the request. Statistical techniques such as ANOVA [129], can be used to analyze the variance in the mean of a dependent variable (service time) due to one or more independent variables (here the complexity). However, ANOVA assumes that the data follows Gaussian distribution and has equal variances across means. The homogeneity of variance is verified through Levene's test. The verification of Levene's test for homogeneity of variances fails. Hence, a non-parametric counterpart test (Kruskal-Wallis test), is used to compare variance of means across complexities. The Kruskal-Wallis test [129] for analysis of variance by ranks across the two levels of complexity yields a statistically significant difference (K=44.1, $p < 0.001$). The results of the Kruskal-Wallis test indicate a significant impact of work complexity on service time. The dispatching policy also indicates that complex work requires an expert to work on it while simple work can be resolved by novices or experts. Figure 4.4 shows the box plots depicting the log of service time for simple and complex work. It is observed that, in this setting: *Complex work takes more time to resolve as compared to simple work. Percentage distribution of simple and complex work forms an important input for arriving at the distribution of experts and less experienced workers.*

The service time variance with work complexity is used by the model and the staffing of experts and novices is determined for varying distribution of work complexities. Table 4.4 shows the results obtained. As the distribution of complex work increases from 20% to 40% of the workload, the number of workers increases from 4 experts (of total 9 workers) and 7 experts (of the total 11 workers).

Complexity	Mean Service Time	Std. Error	95 % Confidence Interval	
			Lower Bound	Upper Bound
Complex	55.28	0.94	53.43	57.13
Simple	39.01	1.15	36.75	41.27

Table 4.3: Summary statistics of service time variance with work complexity

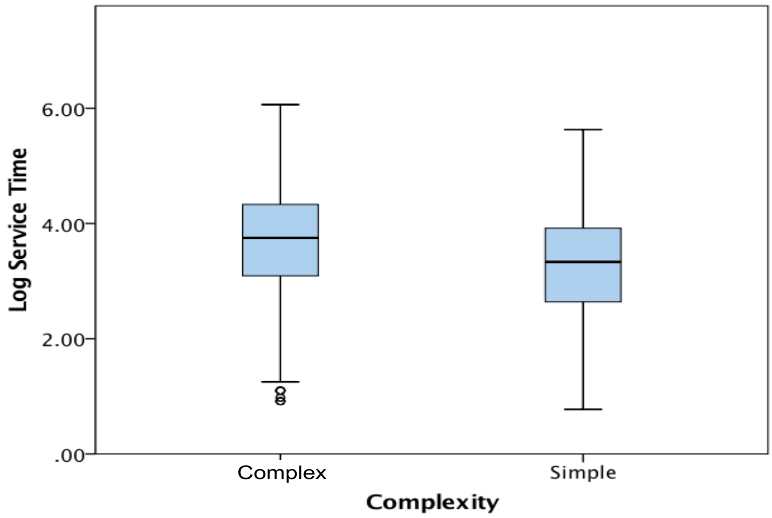


Figure 4.4: Box plot of log service time categorized by complexity

Complexity	% Distribution	Mean Service Time	Number of Workers		% Utilization	
			Expert	Novice	Expert	Novice
Complex	20	55.28	4	5	59.53	86.72
Simple	80	39.03				
Complex	40	55.28	7	4	60.83	89.91
Simple	60	39.03				

Table 4.4: Staffing of experts and novices considering service time variance with work complexity

4.4.2 Impact of Work Complexity and Expertise of Worker on Service Time

Expertise has a significant impact on the efficiency or productivity of a worker [123]. In this study, the variance in service time is evaluated along the dimensions of the expertise of the worker resolving the request. The Kruskal-Wallis test statistics for variance in means of service time across the levels of expertise fails to show statistical significance ($p = 0.403$). This anomaly is attributed to the fact, that less experienced workers do not work on complex incidents (only experts are assigned complex incidents). As complex incidents having higher service time, the overall impact of expertise on service time is not evident. Further, the variance in service time is evaluated considering expertise for low complexity work. The variance in service time means for varying expertise yields a statistically significant difference (K=33.2 , $p < 0.001$).

Table 4.5 shows the variance in service time considering both expertise and complexity of work. Service workers with low expertise level rarely work on complex tickets (as indicated by N=151 of 1964 incidents). However, a significant variance in service time means is observed for low complexity work (Means of 43.7 and 34.1 for Low and High expertise of worker respectively).

Complexity	Expertise	Mean Service Time	Std. Deviation	N
Complex	High	53.85	43.81	1813
	Low	72.41	59.67	*151*
Simple	High	34.12	34.63	646
	Low	43.72	36.18	670

Table 4.5: Summary statistics of service time variance with work complexity and service worker expertise

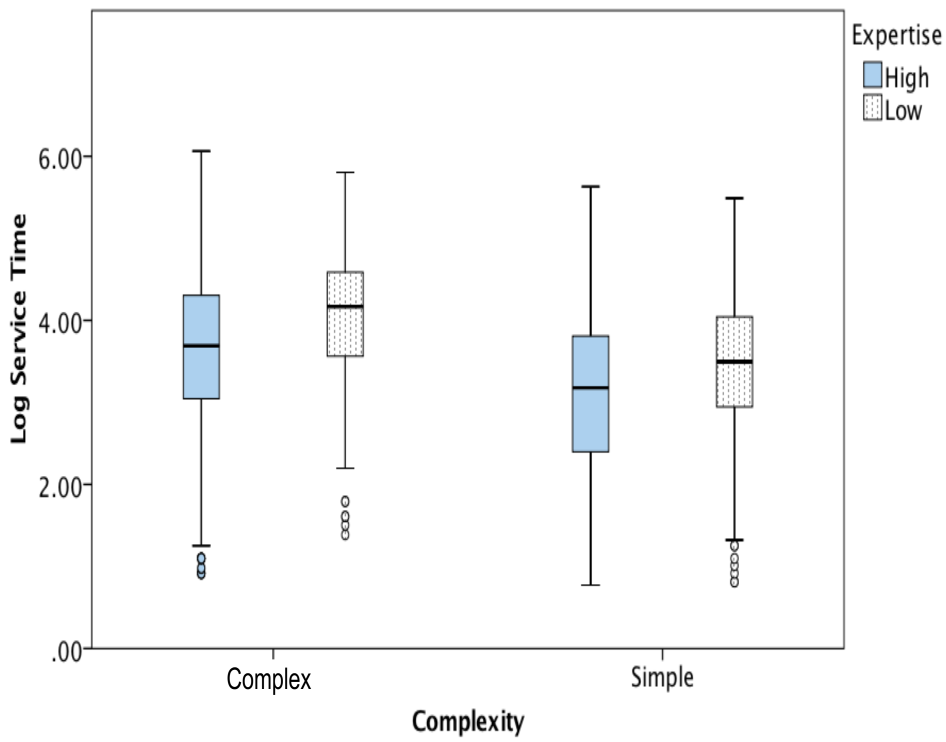


Figure 4.5: Box plot of log service time varying with work complexity and service worker expertise

When the service times derived by analysis of the dimensions of expertise and complexity are used into the simulation model with the staffing results obtained in section 3.4.1, only 85% of low priority incidents meet the service level required. Hence, the staffing derived in section 4.4.1 (service time variance with only complexity as a dimension) is lower than what is required for meeting the target service

Complexity	% Distribution	Expertise	Mean Service Time	Number of Workers		% Utilization	
				Expert	Novice	Expert	Novice
Complex	20	High	53.85	5	5	69.9	90.1
		Low	Not Assigned				
Simple	80	High	34.12				
		Low	43.72				
Complex	40	High	53.85	8	5	60.83	89.92
		Low	Not Assigned				
Simple	60	High	34.12				
		Low	43.72				

Table 4.6: Staffing of experts and novices considering service time variance with work complexity and worker expertise

levels. The variance in service time is modeled accounting for expertise and complexity of work to derive an optimal staffing. Table 4.6 indicates the staffing numbers for novice and experts when using the dimensions of complexity and expertise for service time variance. The staffing solution indicates a higher number of novices. This is because, in this setting, *analysis of service time considering expertise only indicates that, the service time of low complexity work is low when experts work on it. Novices take sufficiently longer time to work on low complexity work.* Hence, more number of novices are required to meet the service levels.

4.4.3 Impact of Work Complexity, Priority and Expertise of Worker on Service Time

Prior work on staffing considers priority of work as an important factor for modeling service time variance [23]. The impact of all the three factors on service time (worker expertise and work priority for simple and complex incidents), is evaluated. Table 4.7 shows the mean service times and the results of Kruskal-Wallis test for different complexities, expertise and priority of the workers. The first four rows show the service times for low complexity requests. The box plot of service times for low complexity work with different worker expertise and priority is shown in Figure 4.6. Here, less experienced workers have the same service time irrespective of the priority. Experienced workers, tend to have better efficiencies only for high priority tickets. It is observed that in the study setting, *the operational efficiency of experts for simple work varies with the importance of work (indicated by priority).* It can also been seen that for less important work, experts take as much time as less experienced workers. This could be attributed to several factors e.g. expert's attention on high priority work, mentoring novices, lower motivation to do less important work, etc. An in-depth analysis of these factors and evaluation through a survey would be needed to understand the variance in expert's efficiency.

Complexity	Expertise	Priority	Mean Service time	Std Deviation	Kruskal-Wallis Test
Simple	Low	Medium	47.77	39.39	$\rho > 0.05(0.4)$
		High	42.98	36.32	
	High	Medium	43.25	34.46	k=36.6 ($\rho < 0.001$)
		High	32.22	34.64	
Complex	Low	Medium	74.4	60.9	not given to workers with low expertise
		High	-	-	
	High	Medium	54.35	38.01	$\rho > 0.05(0.33)$
		High	53.85	45.33	

Table 4.7: Summary statistics of service time variance with work complexity, priority and service worker expertise

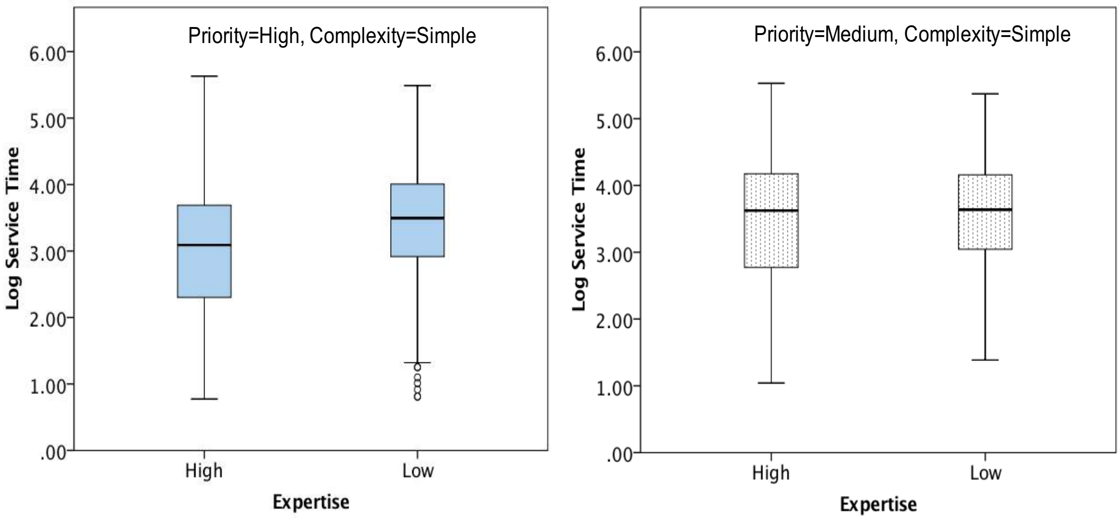


Figure 4.6: Box plot of log service time varying with priority and service worker expertise for low complexity work

The last four rows in Table 4.7 depict the service times for high complexity work. Here, the less experienced workers take longer time when lower priority work is given to them. The operational efficiency of experts does not change with the importance of work. The study data indicates that: *when the complexity of work matches the minimum skill of the worker, there is no improvement in the operational efficiency irrespective of the importance of work.* The staffing obtained in section 4.4.2 when

Complx.	% Dist.	Expertise	Priority	Mean Service Time	Num. Workers		% Util.	
					Expert	Novice	Expert	Novice
Complex	20	High	High	53.3	4	6	61.3	89.3
		High	Medium	54.5				
Simple	80	*High*	*High*	*32.2*				
		High	*Medium*	*43.2*				
		Low	High	42.98				
		Low	Medium	47.77				
Complex	40	High	High	53.3	7	6	63.2	87.2
		High	Medium	54.5				
Simple	60	*High*	*High*	*32.2*				
		High	*Medium*	*43.2*				
		Low	High	42.98				
		Low	Medium	47.77				

Table 4.8: Staffing of experts and novices considering service time variance with work complexity, worker expertise and priority

used in the simulation model accounting for service time mean variances with work complexity, worker expertise and work priority results in a target service level attainment 86% for low severity work. Hence, the staffing solution in section 4.4.2 under estimates the number of workers required to meet the service levels.

The results of the analysis are used to determine the staffing of experts and novices. It is observed that the number of experts reduces as the staffing solution converges at a larger number of novices in this model.

4.4.4 Observations and Dispatching Recommendations

The efficiency of service workers influences the optimal staffing in terms of cost and quality (adherence to service levels). By evaluating the service time of the worker across various dimensions of expertise, complexity and priority, the simulation and optimization framework reflects the behavior of experts and novices and provides the staffing in the face of these three factors. In section 4.4.1 when the service time is only based on complexity of work, the model arrives at a specific number of experts (4 and 7 experts as compared to 5 and 4 novices with varying work complexity distribution respectively) as low complexity work indicates lower service time. When the service time is analyzed in the context of the expertise and complexity (section 4.4.2), the number of novices increases as they take longer time to complete simple requests. The number of experts also increase (5 and 8 experts as compared to 5 and 5 novices respectively) as the experts are found to have better efficiency for simple work. When the experts efficiency is evaluated in the context of priority (section 4.4.3), the model further converges with a solution of having lower number of experts (4 and 7) as they perform better than novices for specific case of higher priority work. The number of novices increases in the final solution as they are preferred for all simple and low priority work.

These observations can be used to improve the dispatching policies or rules that are evaluated by a dispatcher when assigning tickets to service workers. As the complex work can only be assigned to experts and the behavior of the experts does not change for complex work, there is no change in the dispatching rule for assigning complex work. However, simple work can have new dispatching rules as indicated in Table 4.9. Existing dispatching policies in teams primarily evaluate the availability of a service worker. Hence, the dispatching rules in Table 4.1 first check for the availability of a novice and then dispatch to either a novice or an expert. It is recommend that the priority of the incident is also evaluated. If the priority of the incident is high, then an expert can work on it faster and work towards meeting the service levels. If the priority of the ticket is lower, then it should largely be handled by a novice to reduce the cost of the service system as novices and experts have similar service times. These dispatching rules are indicated in Table 4.9.

Recommended Policy in Teams
if (incident priority *isHigh*) and if(expert *isAvailable*) $\rightarrow$ assign to expert
if (incident priority *isHigh*) and if(**not** expert *isAvailable*) and if (novice *isAvailable*) $\rightarrow$ assign to novice
if (incident priority *isLow*) and if (novice *isAvailable*) $\rightarrow$ assign to novice
if (incident priority *isLow*) and if(**not** novice *isAvailable*) and if (expert *isAvailable*) $\rightarrow$ wait in queue
if (incident priority *isLow*) and if(**not** novice *isAvailable*) and if (**not** expert *isAvailable*) $\rightarrow$ wait in queue
if (incident priority *isHigh*) and if(**not** expert *isAvailable*) and if (**not** novice *isAvailable*) $\rightarrow$ wait in queue

Table 4.9: Dispatching policies for simple or low complexity work

4.5 Threats to Validity

In this section, the limitation of the study with respect of *construct validity, internal validity* and *external validity*, is identified .

Construct validity

Construct validity denotes that the variables are measured correctly. The dependent and the independent variables used in this study have been evaluated by earlier studies described in the section 2.4.1. However, the independent variables - expertise levels and work complexity measures can vary across studies. Expertise levels is based on the organization's categorization of its resources. Similarly, categorization of work complexity is relative to type of work being handled and the domain. In this study, this threat is mitigated by considering data from one organization and evaluating teams doing the same type of work i.e. IT service management for operating systems.

Internal validity

Internal validity is established for a study if it is free from systematic errors and biases. The development is accessed data from three teams for a period of 3 weeks. During this measurement interval, issues that can affect internal validity such as mortality (that is, subjects withdrawing from a study during data collection) and maturation (that is, subjects changing their characteristics during the study outside the parameters of the study) did not arise. Thus, the extent of this threat to validity is limited.

External Validity

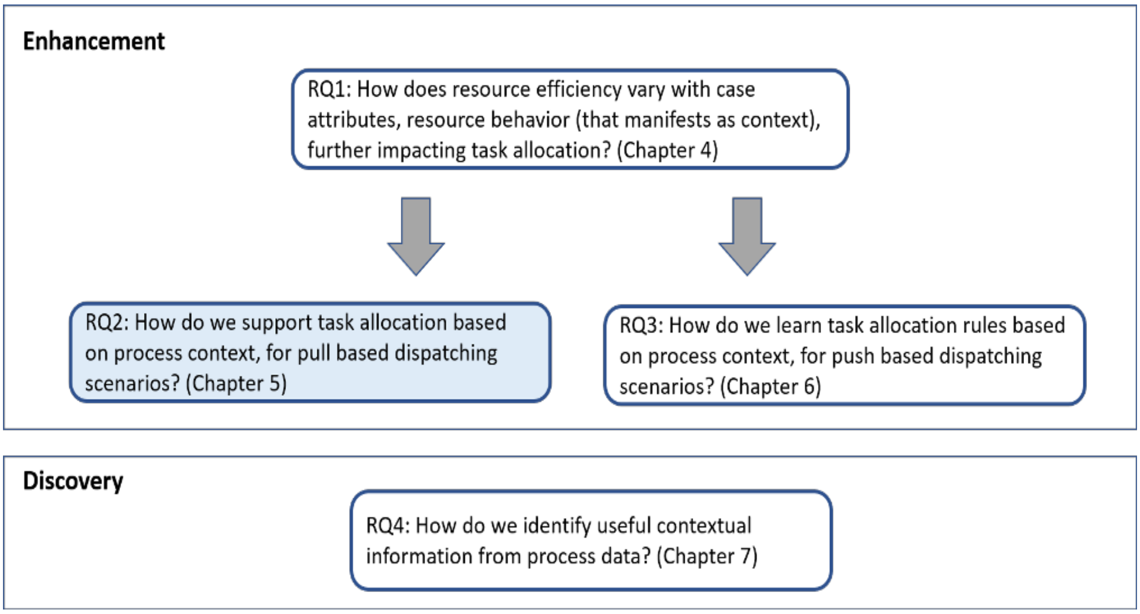
External validity concerns the generalization of the results from this study. The impact of various factors on the operational efficiency of workers is studied based on data collected from approximately 4000 incidents. While insights can be drawn from the study, I do not claim that these results can be generalized in all instances. These results might not hold even in other parts of the same organization. However, the results serve as the basis of using data driven approach for evaluating worker productivity leading to more effective allocation of service workers to service requests.

4.6 Chapter Summary

In this chapter, the variance in efficiency of service workers was evaluated on multiple factors such as complexity of work, priority or importance of work and expertise of the worker. The analysis of service times was further used to evaluate the staffing solution needed to meet the cost and quality requirements of the service system. It was observed that, in the operational study settings, the behavior of experts varies with the importance of work. The insights gained from this study offer implications for dispatching or ticket assignment policies that consider behavior of experts and novices. The study demonstrates that data-driven techniques similar to the study presented in this chapter, can serve as the basis for methodological guidelines and provide effective dispatching and staffing policies required to meet the contractual service levels (quality) of the service system and the business process. This study further alludes to the notion that resource efficiencies are dependent on several factors such as their preference (and other resource behaviors), which has largely been ignored while allocating tasks and staffing teams.

Chapter 5

Context-Aware Task Allocation



In a process, where tasks allocation follows a pull based dispatching policy, the ownership of selecting the right task to work on, lies with the resource. This chapter presents a context-aware recommender system that provides guidance on suitable tasks to resources. The recommender system considers context, resource, task, and resource efficiency. Hence, this allocation method considers resource, case, and time perspective together. The research question (RQ2), is addressed by defining a context model comprising of resource behavior and other task related context. Resource behavior, task attributes, and outcome (or efficiency) is extracted from event logs. The recommender system is evaluated with and without considering context. In addition, the influence of multiple contextual factors, is analyzed.

5.1 Introduction

In knowledge intensive business processes, the most critical resources arguably, are the human resources or knowledge workers. There are various methods of allocating tasks to resources. One of the common allocation methods is a pull-based dispatch policy. In such a scenario, workers or resources commit to tasks as compared to push-based dispatch where tasks are assigned to workers dynamically by the system or manually by a team lead. Pull-based dispatch is preferred when resources tend to multi-task and completion times of these tasks are not a priori known. A resource evaluates the task based on information available with the task (description, urgency, customer), and decides her suitability to commit to the task. The decision making is non-trivial and often knowledge workers, especially novice workers, find it hard to identify their suitability for a task. An added challenge is the fact that operational efficiencies of workers do not depend on the task alone, but also depend on the context or situation when executing a task. For example, a worker may be very efficient when processing a single task but may do poorly when catering to multiple tasks. There are several such situations that could impact the efficiency of the worker (type of task, team member involved, customer involved). Hence, the notion of *context* plays a key role in the decision making.

Dourish [82], presents key assumptions about representational view of context as discussed in Chapter 2: it is a form of information and is separable from the activity. Context is information that can be described using a set of attributes that can be observed and collected. These attributes do not change and are clearly distinguishable from features describing the underlying activity of the user within the context. Satisfying the assumptions of representational view of context, we define process context to be that body of *exogenous knowledge potentially relevant to the execution of the task that is available at the start of the execution of the task, and that is not impacted/modified via the execution of the task*. In this chapter, context is defined at a finer granularity of a task rather than a process.

The proposed approach involves recommendation of tasks to resources taking into consideration the context of the resource and the task. To this end, we build a context-aware recommender system (CARS) [14]. The input to building such a system is data from historical executions of tasks by resources with contextual information annotated in them (some of which are inferred) and the outcome of the execution. The outcome is a goal or performance indicator defined for the task. The recommender predicts the suitability of a task for a resource, by providing a rating. Prediction is based on the assumption that resources who have similar ratings on tasks are likely to have similar ratings towards other tasks. Hence, the rating of a task is predicted, by identifying resources who have had similar ratings on other tasks under similar context. The approach proposed is of considerable practical value. Conventionally, the decision taken by a resource (in many practical business process settings) is based on human judgment, experience and on her implicit understanding of the context. Consequently, task allocation activity is subjective and relies on the experience of a resource. Automated, data-driven support can potentially serve as a game-changer in these settings by providing a personalized recommendation to knowledge worker.

5.2 Motivation

As seen in Chapter 4, operational efficiency of the resources is dependent on many factors specific to the task and the resource. The efficiency or performance of human resources, involved in completion of tasks in a business process, is not homogeneous even if the resources have the same capability or skill. The performance of a resource too, varies depending on the situation. Using a real-life process execution log [116], we analyze the completion time of a task in a loan application process, by two resources at different times during the day. A Kruskal-Wallis H test [129] showed that there was a statistically significant difference in completion time of the task of *'Resource 11180'* at various time periods of the day, ($\chi^2(2) = 7.15, \rho = 0.05$), with a mean rank completion time score of 45.39 during 9 AM - 12 PM, 65.79 between 12 PM - 5 PM and 60.48 after 5 PM. However, *'Resource 10931'* does not have a statistically significant variance in the mean completion time, for the same task at different time periods. Figure 5.1 shows the mean completion times of the resources at various time periods of the day. Considering, time of the day, as the contextual dimension, the task allocation between 12 PM - 5 PM is more suited to *'Resource 10931'*. Hence, context-awareness would help in task allocation decision. The results further indicate that same contextual dimension may impact performance of one resource but not another, highlighting the heterogeneous behavior of human resources.

When a pull-based dispatching is adopted for task allocation, a work request or task instantiated in the system enters a common queue or a shared work list and remains there till a knowledge worker or resource commits to the task. Every knowledge worker is able to peek into the common queue and view the tasks they are authorized to work on, based on their roles and organizational positions. Workers evaluate the type of task, their suitability to execute the task and other factors to decide if they should commit to a task or not. Once a task is committed or selected, performance measures associated to the task need to be met (target completion time, degree of customer satisfaction and so on). While experienced workers in the system learn to identify tasks that they are best suited for, novice workers

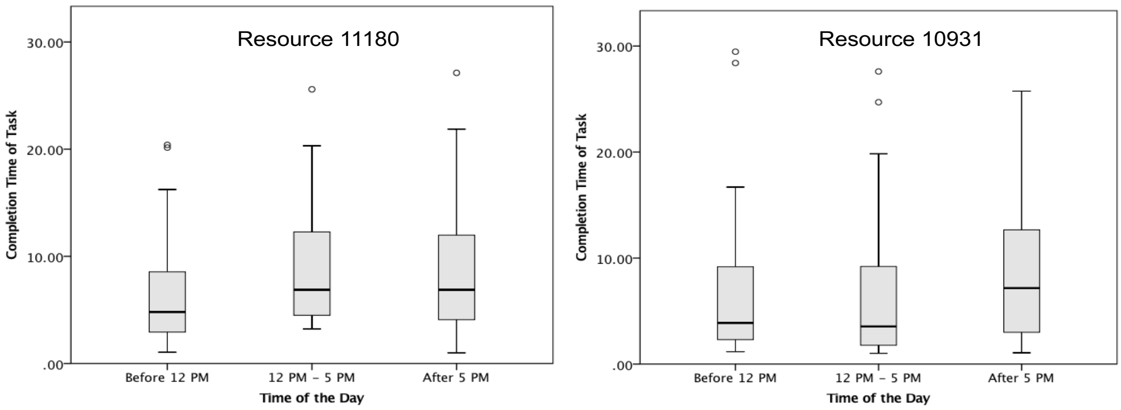


Figure 5.1: Completion time of two resources on same task at different time periods of the day

need help in identifying suitable tasks. Incorrect decision making could result in a resource placing task back into the queue, taking longer time to complete or having poor degree of customer satisfaction. Here, recommending the right task to the resource would lead to better process execution efficiency. Considering context while recommending the task (context-aware recommendation), would provide a resource specific (personalized), task allocation recommendation.

5.3 Approach

The approach consists of three phases: the modeling phase, the data extraction phase and the recommendation phase (see Fig. 5.2). The *modeling phase* involves identifying the contextual dimensions of the task, resource and the domain. The dimensions can be generic or domain specific. Here, domain experts would identify the relevant dimensions. The *data extraction phase*, involves using historical process execution logs to extract the contextual dimensions of the process, task and the resource. Relevant performance outcome measures such as completion time of the task or quality of the task are extracted or derived from the event logs. These form the inputs in building a context-aware recommender system. In the *recommendation phase*, for each resource, the relevant contextual information is computed and the the suitability of resource on the new and ongoing tasks is predicted.

To apply machine learning techniques, we need to engineer contextual dimensions for a resource, task and the process instance. A resource has several contextual dimensions (e.g. preference, current workload, etc.) as would the task and a process (e.g time of the day, time zone of customer, etc.). The performance outcomes for the relevant resource and task specific contextual dimensions are extracted from historical process execution logs. These form inputs to the recommender system. For a new task, the resource and their contextual dimensions are given as input. The rating of the task for the resource is predicted. Hence, the approach i) identifies the relevant contextual dimensions of resource(s) that impacts performance outcome or rating, (ii) determines the context-aware recommendation models for resource(s), and (iii) predicts the rating of the resource on a task in the task list. Before describing the details of the approach, the underlying topic of context-aware recommender system is introduced.

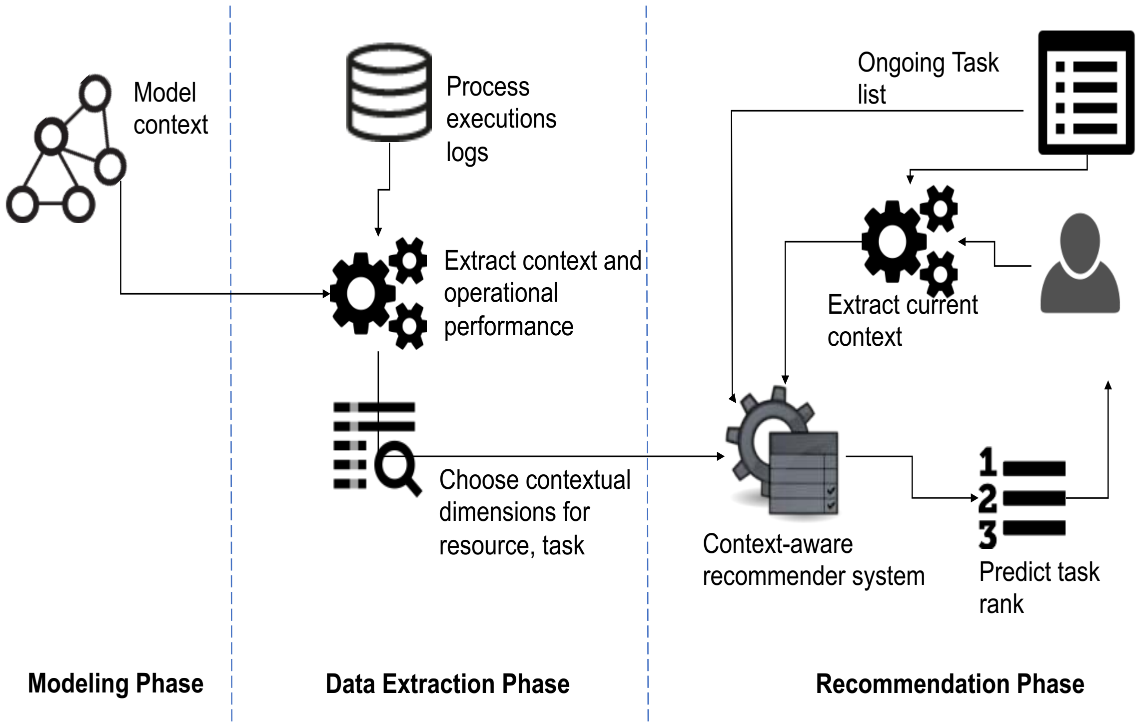


Figure 5.2: Overview of context-aware task allocation

5.4 Context-Aware recommendation system

A recommender system predicts the rating of a user for an item, which is reflective of the preference of the user for that item. The system defines a rating function:

$R : User \times Item \rightarrow Rating$

Each user and item pair is mapped to a rating value. This is considered to be prediction problem where ratings of all user and item pairs is not known but must be predicted. Such recommender systems are called *2D or two dimensional* recommender systems [14]. Context-aware recommender systems use additional information of context as a part of the rating function:

$R : User \times Item \times Context \rightarrow Rating$

where, context represents additional conditions or situations where the user provides a specific rating to item. Use of contextual information results in providing better recommendations [14]. The ratings, hence are modeled as the function of not only items and users, but also of the context. The input data for traditional recommender systems is based on tuples of the form $\langle user, item, rating \rangle$. In contrast, context-aware recommender systems (CARS), are built based on additional of contextual information with tuples of the form $\langle user, item, context, rating \rangle$, where each specific record includes not only the rating of a user on a specific item, but also the contextual information in which the item was rated by this user. A common illustration of a two dimensional model (2D model), of traditional recommender systems and multi-dimensional model used to represent CARS is shown in Figure 5.3.

With the users representing resources, items representing the tasks and rating representing performance outcomes (such as completion times of tasks), CARS can be used to recommend tasks to a resource based on ratings of other similar users (resources).

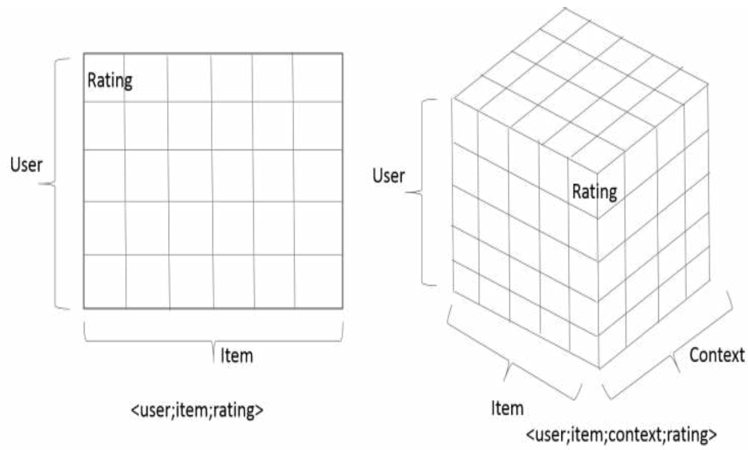


Figure 5.3: 2D model for traditional recommender systems and multi-dimensional model for CARS as discussed in [14]

Multiple methods have been used to build context-aware recommender systems:

- *Contextual pre-filtering*: In this approach, context is used to select the relevant ($\langle user, item, rating \rangle$) data for generating recommendations. On the subset of user-item pairs, ratings are predicted using any traditional collaborative filtering methods (detailed in Section 2.8). An example of using context pre-filter is to select data with users and ratings at specific time or location (context).

- *Contextual post-filtering*: This approach uses all the data for predicting the ratings. Then, the obtained ratings are adjusted using the information of context by i) filtering out recommendations that do not have the same context, ii) adjusting or calibrating the rating using the contextual information.

- *Contextual modeling*: This approach uses an unsupervised method where the recommendation function or user's rating for an item along with contextual information is learnt (built using approaches such as decision tree, support vector machine, or other technique).

Efficient contextual pre-filtering techniques using neighborhood based methods such as user splitting [130], item splitting [131] and UI splitting [132] have been proposed are known to have lower prediction errors of ratings. Item splitting splits items based on the context. The split is done when the a contextual dimension in which items are rated, significantly differ. Statistical test such as t-test, Kruskal-Wallis test can be used to evaluate if the means of ratings differ significantly across the values of the contextual dimension. Hence, the same item under different contextual dimensions would be treated as a different item. User splitting splits users instead, when the ratings are significantly different for different contextual dimensions. UI splitting applies item splitting and user splitting together.

5.5 Modeling CARS for Task Allocation

The elements of a context-aware recommender system are users, items, rating of users to items, context and the similarity measure to identify neighbors. In this section the resource, task, context are modeled and similarity of resources is defined, to build a context-aware recommender for task allocation.

5.5.1 Resource

The resource model described by Muehlen et al. [33] is used, to model resources (user). In Muehlen's resource model, each resource owns some *roles* that represents capabilities and privileges to perform tasks, occupies *positions* in organization units, that further provide privileges to perform task. Model of a resource is essential to ensure that the recommender does not recommend tasks that are out of a resource's capacity or privilege. A resource is represented by a set of attributes D_R representing role, position, organization and other relevant information. These attributes characterize the resource and are static - they do not change during the execution of a task. Hence, a resource r is represented by attribute-value pairs $v_r = (v_r^1, v_r^2 \dots v_r^{D_R})$.

5.5.2 Task

Item is a task that needs to be completed by a resource. Task is an executing instance of an activity in a process. Task is characterized by attributes of the process instance it belongs to, and the attributes specific to the task. Task attributes are endogenously determined elements (i.e., attributes whose values are determined via the execution of the task) as well data provided as input to the task. For example, for a task that verifies a loan application, the loan amount would be a task attribute. A set of attributes D_T is used to denote process and task data in the usual sense, i.e., data provided as input to a process or task, data modified or impacted by a process or task and data generated as output by a process or task. Hence, a task t is represented by attribute-value pairs $v_t = (v_t^1, v_t^2 \ldots v_t^{D_T})$.

5.5.3 Context

Context is an important model element in the presented approach. Saidani et al. [16] define a meta-model of context for a business process. The meta-model comprises of context entity and context attributes. Context entities are connected to each other using context relationships. I leverage this meta-model and use context entities such as activity and resource, and their related contextual attributes. Contextual attributes are referred to as contextual dimensions (as attributes for a contextual dimension is defined later in this section). While previous work has considered context for the overall process, here context is modeled for tasks in the process. Contextual entities and dimensions captured in the model vary with the situation [16] - *"There is no context without context: the notion of context should be defined in terms of a purpose."*. Figure 5.4 illustrates the context model used for the purpose of task allocation recommendation. The contextual entities are task and resource. The generic contextual dimensions for task and resource are defined in the model. In addition, domain specific contextual dimensions would need to be defined and added. An example of a domain specific dimension for a resource would be a 'number of years in organization'. Task specific contextual dimensions such as time of the day of executing task, the duration of the task and time to finish are self explanatory. Generic contextual attributes of resource that impact task allocation decisions are presented. These contextual dimensions are based on the resource behavior measures described in section 2.2.3:

Workload can be either the number of tasks waiting at the start of execution of a task or the number of tasks that have been completed over a particular period [12]. It defines 'how busy' a resource is or has been when committing to a task. $WL(r,t) \rightarrow N$, where $WL(r,t)$ is the workload of a resource r at time t.

Availability indicates whether a resource is available to perform a task within a specific time limitation. Huang et al. [57] define resource availability measure, to predict if a resource is available at some time in the future. A simpler measure of availability of a resource r at time t is $Avail(r,t) \rightarrow \{true, false\}$,

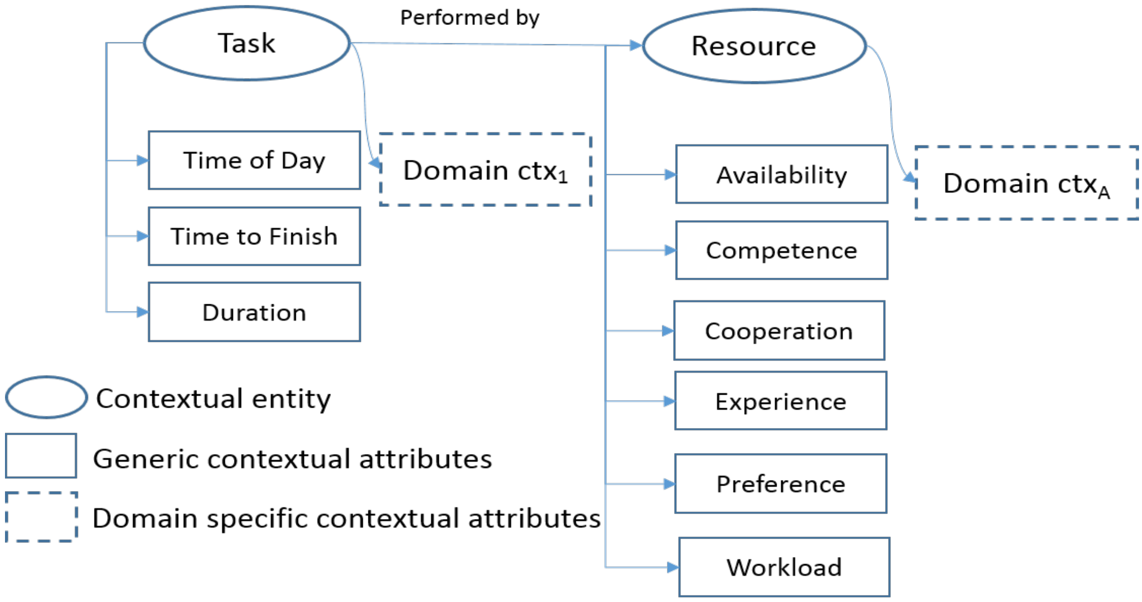


Figure 5.4: Context model used for task recommendation

a boolean true or false where the $Avail(r,t) = false$ if $WL(r,t) \geq \tau$ where τ is defined for a specific task.

Competence is the ability to perform a certain type of task [57]. If a resource performs a certain type of task by using lower cost than the others, it means that the resource has a higher competence level than others to perform the task. The cost can be defined based on business process (e.g. completion time, quality).

Cooperation is the ability of working with other resources. Kumar et al. [60], define compatibility or cooperation as a measure of the degree to which resources cooperate with one another in a process. Cooperation between resources who perform tasks where there are hand offs, is measured as described in [60].

Experience is acquired and improved by performing tasks [59]. The number of times a task has been performed and the duration or time period for which the task is performed, is used to measure experience.

Preference is acquired knowledge or attitude to do a certain kind of task. For example, if a resource commits for a type of task frequently, the preference to the task is high. Preference $\rho(a,r)$ of a resource r on task type a is given as: $\rho(a,r) = Card(a,r)/Card(a)$, where $Card(a,r)$ is the number of tasks of task type a, resource r has completed and $Card(a)$ is the total number of tasks of type a completed by all resources.

Moreover, each contextual dimension c, can be defined by a set of q attributes $\{c_1, \ldots c_q\}$ having a hierarchical structure and capturing a particular type of context (e.g., experience of a resource). The values taken by attribute c_q define finer (more granular) levels, while c_1 values define coarser (less granular) levels of contextual knowledge. For example, Figure 5.5 presents a two-level hierarchy for the contextual attribute c specifying experience of a resource to a task. While the root (coarsest level) of the hierarchy defines *experience* on an activity or task, the next level is defined by attribute $c_1 = \{experience\_case, experience\_customer\}$, which identifies the experience of a resource handling the specific case (or other tasks related to the case) and handling a specific customer.

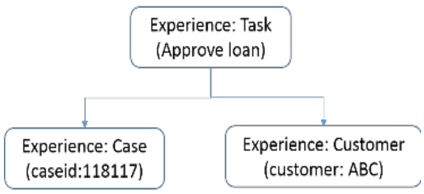


Figure 5.5: Hierarchy structure of a contextual dimension

5.5.4 Resource similarity

Various similarity measures that calculate the similarity among resources or users, have been defined in the implementation of CF algorithms. Correlation-based similarity of two resources u and v is measured by computing $Pearson - r$ correlation $corr_{u,v}$. The correlation between two user's ratings on common tasks, is used to determine similarity. The correlation used from [104] is as follows:

$$s(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r_u})(r_{v,i} - \bar{r_v})}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r_u})^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r_v})^2}} \tag{5.1}$$

Where I_u are items or tasks executed by u and I_v are items or tasks executed by v. $r_{u,i}, r_{v,i}$ is the rating of item i by user u and v respectively. $\bar{r_u}, \bar{r_v}$ is the average rating of the user u, v respectively. Once the similarity is computed, k neighbors are selected and the prediction of a rating on task i for a resource u is arrived at by computing the sum of the ratings given by the neighbors users. Each rating is weighted by the corresponding similarity $s(u, v)$.

5.5.5 Rating

In CF, users provide ratings to as many items as possible. Here, the outcome of past task executions is used to compute the rating of a resource to a task. Outcomes are typically performance indicators defined for the business process. Time to complete a task, quality level or percentage of tasks meeting a deadline are some examples of outcomes. Rating is an ordered set and needs to be on a common scale for all users. A sigmoid function is used to compute ratings. The computation of rating for a resource r_a, with completion time of task t as the outcome is given by a sigmoid function:

$$R(r_a, t) = \frac{1}{1 + e^{-k(\mu_t - \mu_{r_a,t})}} \tag{5.2}$$

where μ_t is the mean completion time of the task and $\mu_{r_a,t}$ is the mean completion time of the task t by the resource r_a. The parameter k can be varied to get the required rating interval. In particular, if the variance in outcome is high, k should be smaller to be more sensitive to these variances, similarly, if the variance is low, k should be higher. If there are multiple performance indicators, a rating can be arrived at by selecting from or combining different indicators. The ratings can be further scaled up to a suitable interval of [0,10].

Figure 5.6 shows the distribution of ratings derived for the completion times of a tasks from the event log [116], where k is based on the standard deviation σ of the completion times: $k = (0.25\sigma, 0.5\sigma, 0.75\sigma)$. Here a lower value of k would be preferred as it has a larger distribution of ratings which is suitable for identifying range of performance outcomes.

5.6 Data Extraction and Training

An important requirement for building and deploying CARS is the availability of contextual information along with historical task executions or the data required to train the recommender system. The current approach infers contextual dimensions such as preference, workload, cooperation and competence from event logs. Figure 5.7 provides a snapshot of the real-world event log [133], containing the details of the task, the resource owning the task, start time and completion time. The context information such as hour of the day when the task is created, the preference of the resource, the workload and other relevant context information are extracted and the $\langle Resource, Task, Context, Outcome\rangle$ is derived. In the scenario where context is not used, there can be multiple outcomes for the same user and item. as illustrated by the task pertaining to product 'PROD424' in Figure 5.7. The completion time takes {3,5} minutes. Here, aggregation techniques such as average completion time or the median completion time is used. A similar aggregation technique would be applied if there are multiple ratings for a resource on a task with same contextual dimensions.

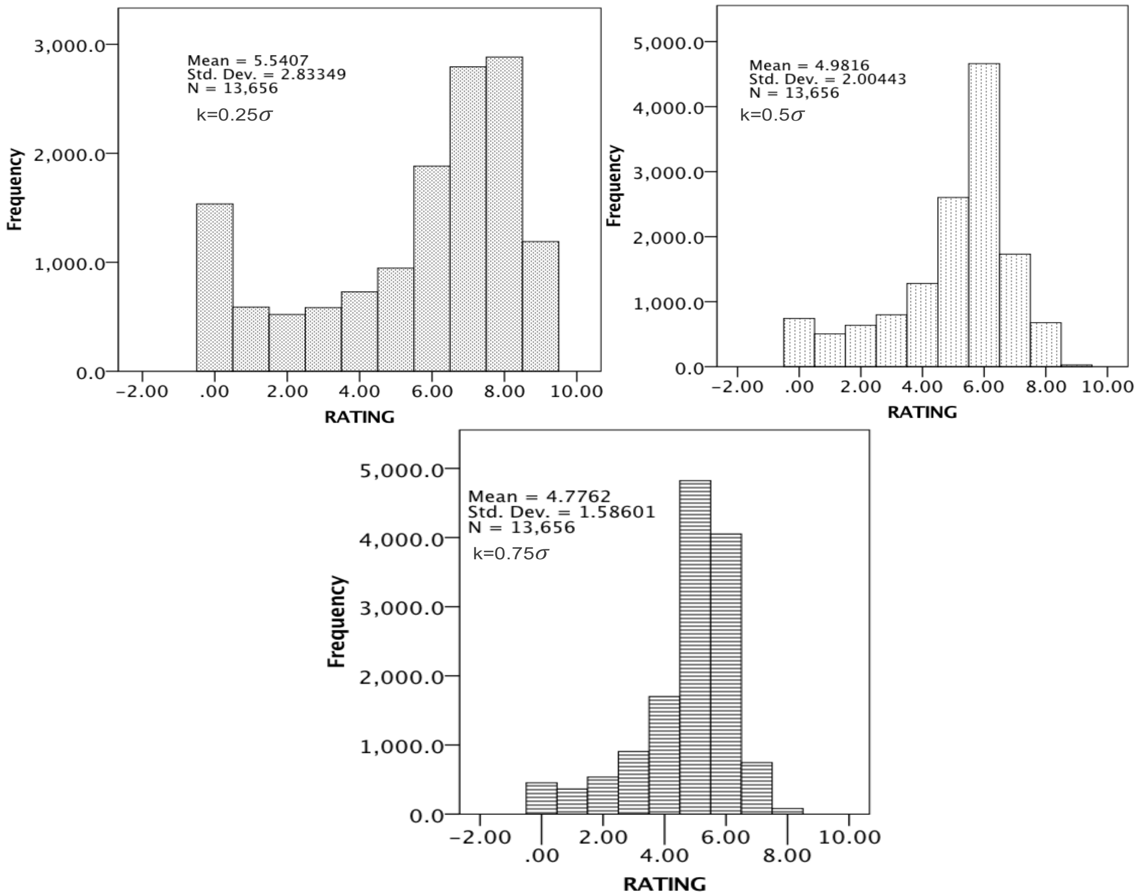


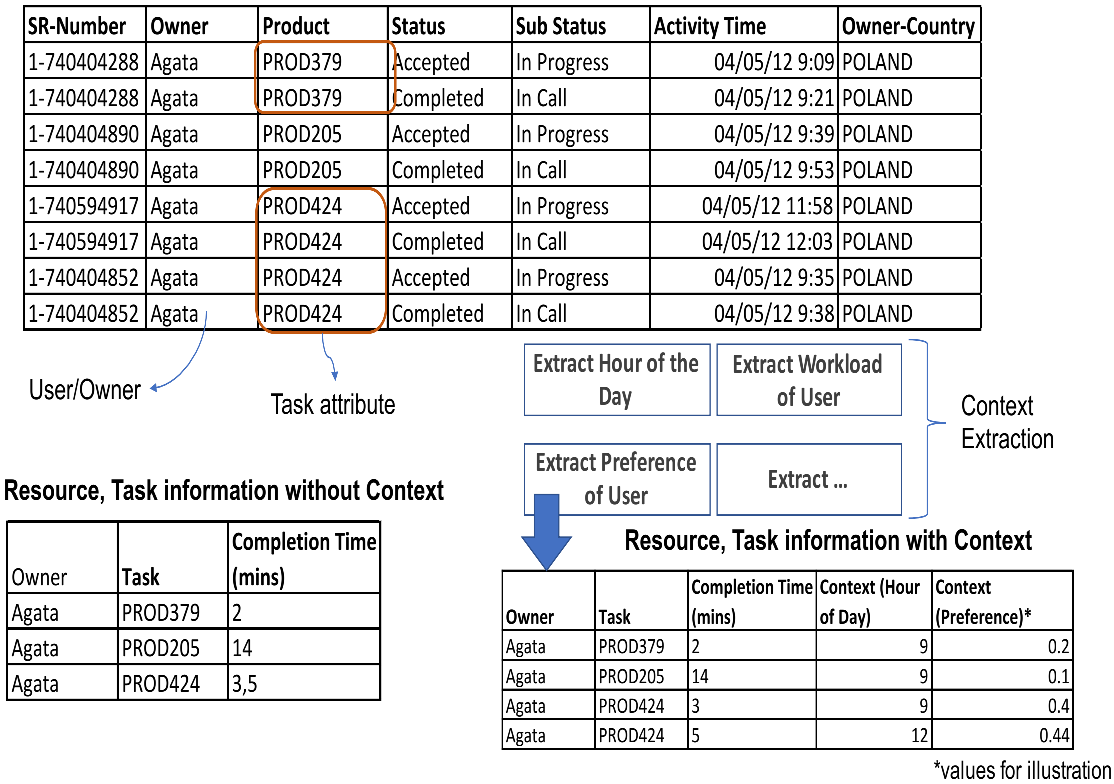
Figure 5.6: Distribution of rating with different values of k

5.6.1 Context-aware task recommendation

Information of the resource, task and context is used to predict the rating. Formally, with the multi-dimension data model, D_R and D_T are the dimensions of the resource and task respectively. The dimension D_R is a subset of Cartesian product of some attributes of the resource. For example, a resource dimension is defined as $Resource \subseteq Name \times Role \times Department$. Similarly, the task dimension is defined as $Task \subseteq Name \times Type$. Finally, the dimensions of context such as, $D_{workload}, D_{time}$ are included (and other relevant contextual dimensions). Given all the dimensions, the rating function F is defined as:

$F : D_R \times D_T \times D_{workload} \times D_{time} \rightarrow Rating$

There are multiple approaches to using contextual information in the recommendation process. In this work, I use contextual pre-filtering approaches such as user splitting, item splitting, and UI splitting as these methods are known to have lower prediction errors of ratings and have been evaluated in earlier studies [130], [131], [132].



SR-Number	Owner	Product	Status	Sub Status	Activity Time	Owner-Country
1-740404288	Agata	PROD379	Accepted	In Progress	04/05/12 9:09	POLAND
1-740404288	Agata	PROD379	Completed	In Call	04/05/12 9:21	POLAND
1-740404890	Agata	PROD205	Accepted	In Progress	04/05/12 9:39	POLAND
1-740404890	Agata	PROD205	Completed	In Call	04/05/12 9:53	POLAND
1-740594917	Agata	PROD424	Accepted	In Progress	04/05/12 11:58	POLAND
1-740594917	Agata	PROD424	Completed	In Call	04/05/12 12:03	POLAND
1-740404852	Agata	PROD424	Accepted	In Progress	04/05/12 9:35	POLAND
1-740404852	Agata	PROD424	Completed	In Call	04/05/12 9:38	POLAND

Owner	Task	Completion Time (mins)
Agata	PROD379	2
Agata	PROD205	14
Agata	PROD424	3,5

Owner	Task	Completion Time (mins)	Context (Hour of Day)	Context (Preference)*
Agata	PROD379	2	9	0.2
Agata	PROD205	14	9	0.1
Agata	PROD424	3	9	0.4
Agata	PROD424	5	12	0.44

Figure 5.7: Extracting contextual dimensions for user and task from past execution log

5.7 Evaluation

In this section, the evaluation of the approach is presented. First, the set up for the evaluation is presented. Then evaluations on two real-life event logs are detailed.

5.7.1 Evaluation setup

In order to conduct the evaluation, collaborative filtering based recommender is implemented using CARSKit [a][134]. Figure 5.8 depicts the procedure for evaluating context-aware recommender system. Two real-world event logs are used. Based on the identified performance outcome (completion time, quality), ratings are computed for each resource-task pair. The event logs are enriched by computing additional information about context, using information of the task, resource executing the task, the task's start and end times. The data without contextual information and data with contextual information, are utilized to carry out the validation using k-fold cross validation with $k = 10$. The data is divided into 10 folds or subsets. In the approach without contextual data (marked as 1), rating of a task for a resource is predicted and compared with the actual rating. In the context enriched approach,

[a]https://github.com/irecsys/CARSKit

additional contextual information is used to predict the rating of a task for a resource under that specific context.

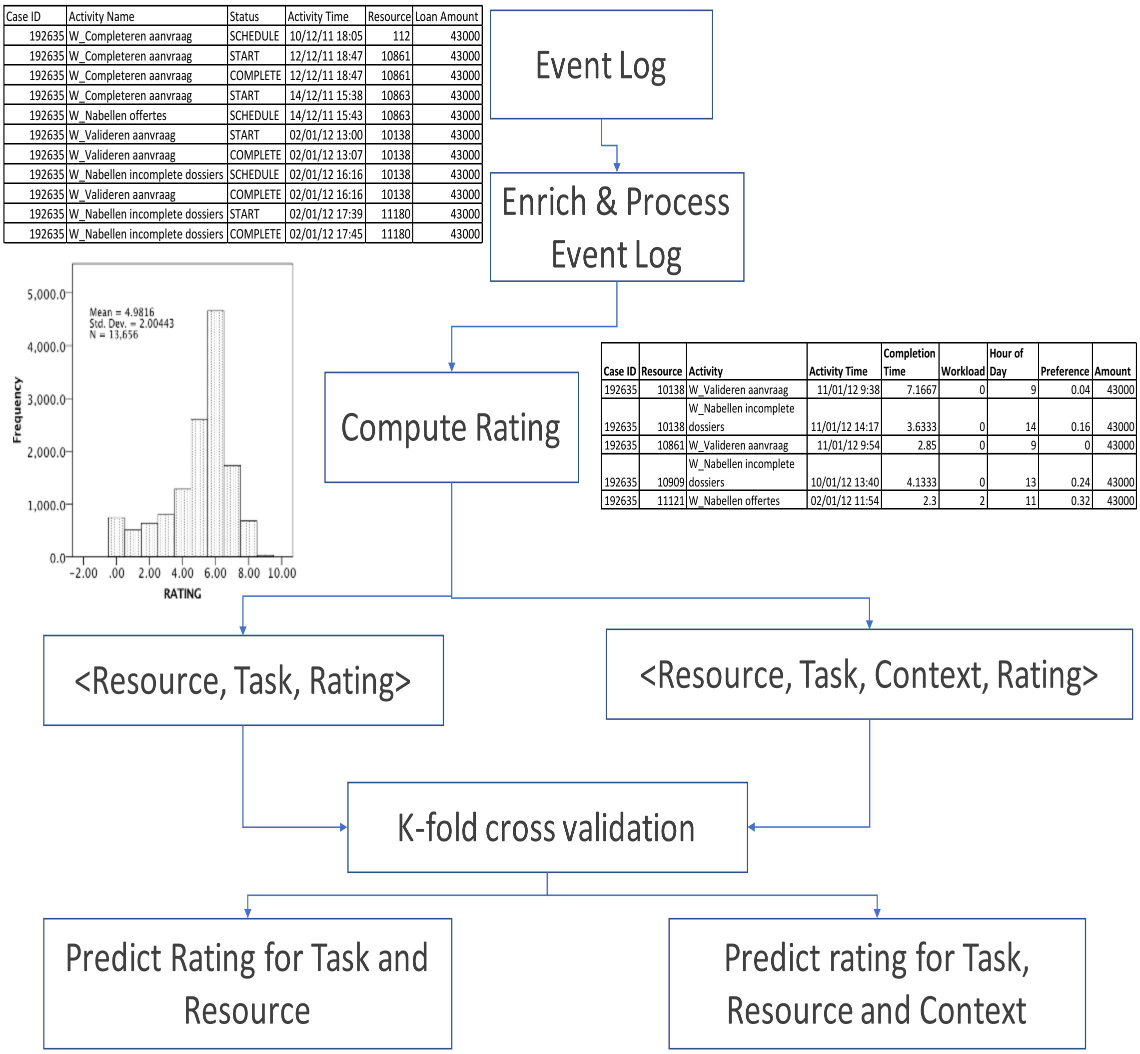


Figure 5.8: Evaluation procedure

5.7.2 Performance Measures

The common measures used in evaluating the accuracy of a recommender model are based on Absolute error ($|ActualRating - PredictedRating|$). $ActualRating$ is the real rating assigned to a task and user pair. $PredictedRating$ is the outcome of the recommender system. Mean Absolute Error (MAE)[135] is used to compare the performance of the model and is defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |ActualRating_i - PredictedRating_i| \tag{5.3}$$

where N is the number of tasks and resource pairs used for evaluating the performance of the model (test set).

The *Root Mean Squared Error* RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(|ActualRating_i - PredictedRating_i|)^2} \tag{5.4}$$

In addition, to compare the performance of the model without using context and using contextual dimensions, the statistical significance of the absolute errors achieved with the two models is verified using the Mann Whitney-U Rank Test [136]. The Mann Whitney-U test is a non-parametric statistical test used to compare mean variances in two independent samples. Mann Whitney-U test is used, as the training set and test set are not the same when we build the recommender systems with and without context.

5.7.3 Incident Management Event logs

To validate the effectiveness of using context in a real-life business process providing services, the 2013 edition of the BPI challenge event log of Volvo IT services [133], is used. The log (event log 1) contains events of an incident management process. Each incident is a task that relates to a glitch in a product. An IT service personnel or resource works on the incident and restores service. The event log contains the information about the product associated to the incident, impact of the incident, resources who worked on the incident, time and status of the incident. Product related to the incident is used to categorize tasks or items. In the log, there is not much information about resource other than name of the resource.

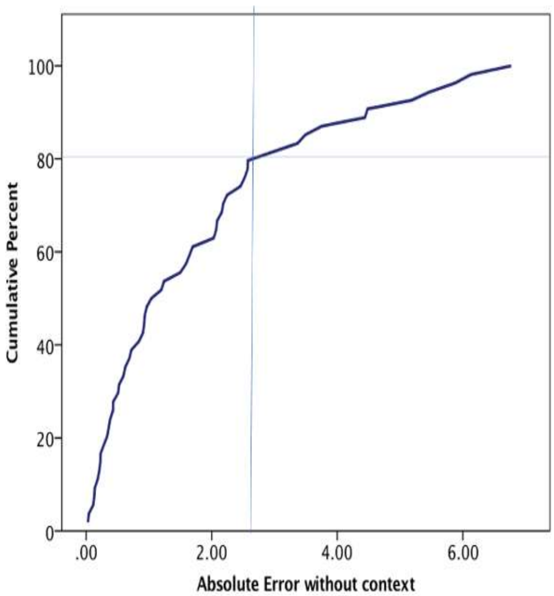
The logs are enriched with additional contextual information: time or hour of the day the task is created, workload and preference of a resource. The workload of the resource at a specific time, is computed by evaluating the number of active incidents in the queue of the resource at that time. The preference for task (item) is the ratio of the number of tasks executed by the resource and associated to the product, to the total number of tasks associated to the product.

Completion time of the task is used to compute the ratings. A sigmoid function considering the mean completion times for incidents of a product are computed. For predicting ratings, only a subset of incidents where one single resource has worked on it, is considered. Overall 3460 process instances supported by 346 unique resources or owners, are considered. Event logs involving multiple resources, do not provide clarity on the time spent by each resource on an incident and hence are not used.

The number of neighbors k, for predicting rating is set to 5. The mean absolute errors for completion time with and without context is shown in Table 5.1. The MAE and RMSE reduces when additional contextual dimensions are used to predict the rating of tasks. The cumulative percentage distribution of absolute error between the predicted and actual ratings, with and without context is shown in Figure 5.9. Use of contextual dimensions results in larger percentage of predictions with lower absolute errors (80% of the data has absolute error less than 2.0).

CARS	MAE	RMSE	Mann-Whitney U Test with baseline
$\langle Resource, Product \rangle$ (baseline)	1.571	2.07	-
$\langle Resource, Product, TimeOfDay \rangle$	1.366	2.01	$p < 0.01$
$\langle Resource, Product, Preference \rangle$	1.346	2.04	$p < 0.01$
$\langle Resource, Product, Workload \rangle$	1.466	1.90	$p < 0.01$
$\langle Resource, Product, Time, Preference \rangle$	1.287	1.96	$p < 0.01$
$\langle Resource, Product, Time, Workload \rangle$	1.30	2.01	$p < 0.01$
$\langle Resource, Product, Pref, Workload \rangle$	1.30	1.98	$p < 0.01$
$\langle Resource, Product, Time, Pref, Workload \rangle$	1.26	1.95	$p < 0.01$

Table 5.1: Results of using contextual dimensions to predict performance outcomes for event log 1



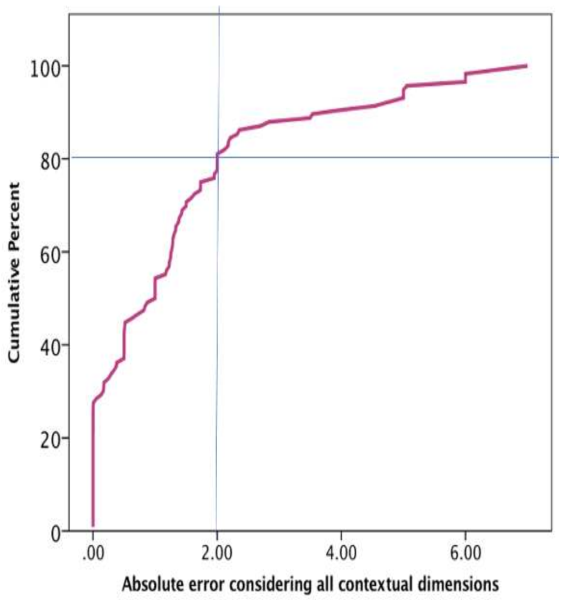


Figure 5.9: Cummulative percentage distribution of absolute error for event log 1

5.7.4 Financial Institute Event logs

The second study uses event logs of 2012 edition of BPI challenge (event log 2), taken from a Dutch financial institute [116]. The event log represents an application process for a personal loan or overdraft. The loan amount requested by the customer is indicated as an attribute in the logs. While there are over 1000 event types or activities present in the log, event types that indicate manual effort exerted by the bank's staff are evaluated. The manual effort is limited to 6 task types (associated to each activity). Task name and amount of loan requested are used as task attributes. The loan amount is categorized into 7 bins (based on the counts of loan applications). There is no additional information about the resource. Resource who have executed at least 100 tasks in the months of February and March are considered. Hence, there are 2709 process instances and 39 resources. Number of neighbors is set to 5. Workload of a resource at a specific time and preference of a resource to a task is computed from the log. Preference of a resource to a task is computed as the ratio of the number of tasks of the task type completed by this resource to the total number of tasks of the same task type completed by all resources. The rating is computed based on the completion times of task (activity and loan amount are considered as task or item attributes). The rating of a task for a resource with and without contextual information is predicted and MAE, RMSE as indicated in Table 5.2 are computed. The cumulative percentage distribution of absolute error between the predicted and actual ratings, without context, with different contextual dimensions (time of the day, workload, time of the day and workload) is shown in Figure 5.10. Adding workload as context, does improve the MAE but marginally. The absolute errors are lowest when time of the day is used as context, with 80% of the data having the absolute error less than 3.0.

CARS	MAE	RMSE	Mann-Whitney U Test with baseline
$\langle Resource, Task\rangle$ (baseline)	2.00	2.51	-
$\langle Resource, Task, TimeOfDay\rangle$	1.73	2.16	$p < 0.01$
$\langle Resource, Task, Preference\rangle$	1.78	2.12	$p < 0.01$
$\langle Resource, Task, Workload\rangle$	1.98	2.46	$p < 0.05$
$\langle Resource, Task, Time, Workload\rangle$	1.84	2.06	$p < 0.01$
$\langle Resource, Task, Time, Pref\rangle$	1.71	2.10	$p < 0.01$
$\langle Resource, Task, Pref, Workload\rangle$	1.76	2.08	$p < 0.01$
$\langle Resource, Task, Time, Pref, Workload\rangle$	1.85	2.16	$p < 0.01$

Table 5.2: Results of using contextual dimensions to predict performance outcomes for event log 2

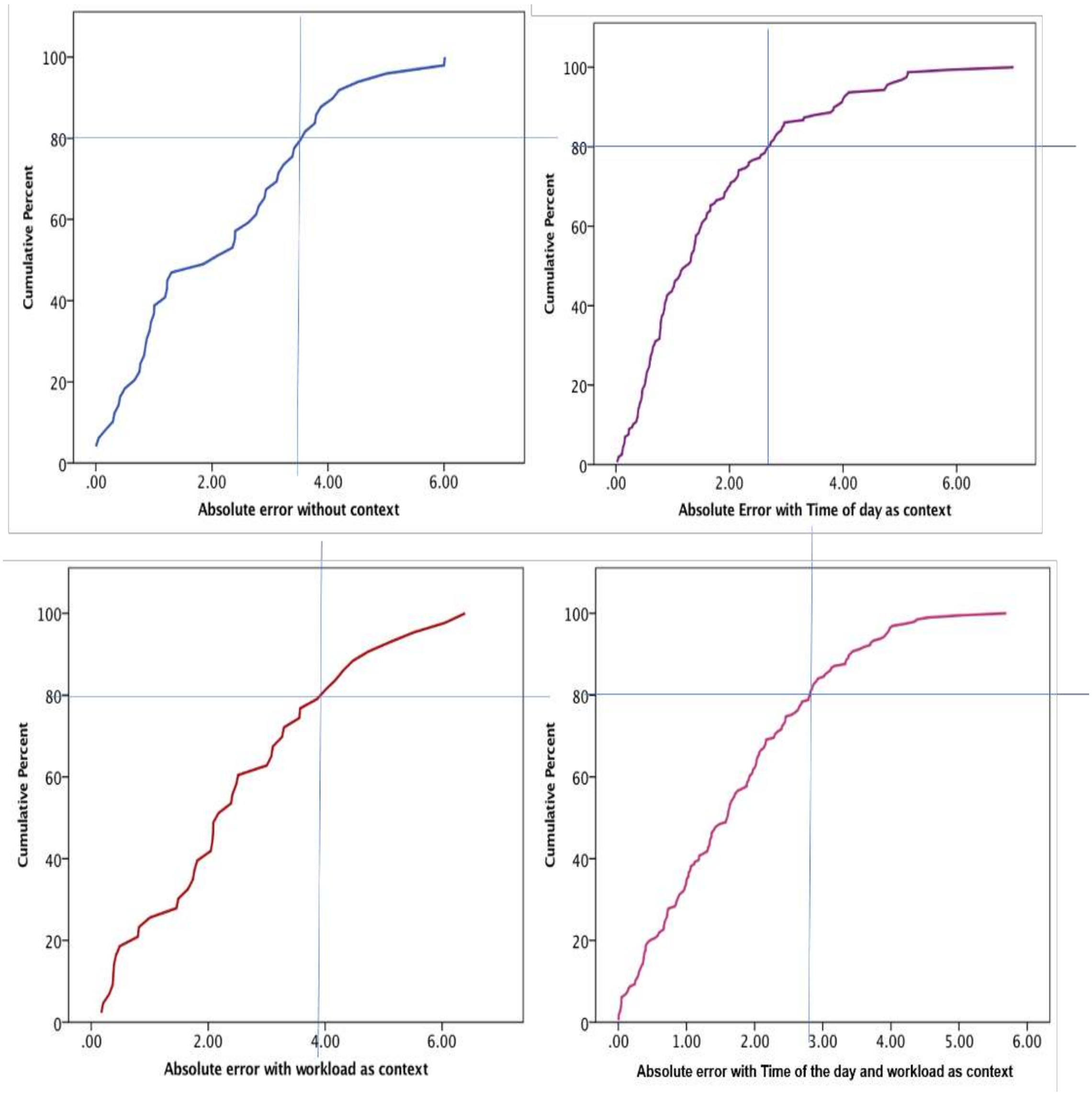


Figure 5.10: Cummulative percentage distribution of absolute error for event log 2

5.7.5 Discussion

The results of evaluation indicate that the ratings of a task for a resource are influenced by context. The results for event log 1 demonstrate a decrease in MAE and RMSE with the addition of each of the contextual dimensions. Adding all contextual dimensions reduces MAE. In event log 2, addition of contextual information such as time of the day, preference decreases MAE. However, addition of workload has a marginal reduction in MAE and inclusion of all contextual dimensions has higher MAE compared to single dimensions such as preference or time of the day. While context improves the MAE, selection of contextual dimensions needs to be adapted to the relevant business process. For the evaluation, event logs were analyzed for a time period of 3 months or less, and hence this is limited as CARS requires sufficiently large data that captures ratings in varying situations. Measuring and using additional contextual dimensions on a larger event log would be a useful activity. The models built for evaluation do not contain any domain specific contextual dimensions (due to lack of any additional information other than the log). It would be useful to build a model that includes domain specific contextual dimensions.

In real-world recommender systems, there could be a possibility that none of the resources are suitable for a task in their specific context. The task would be rated low for all resources. Such a situation could lead to a task not being picked up or completed on time. For handling such scenarios, additional alert mechanisms have to be built into the service system, to identify tasks that have rating below a specific threshold for all the active resources.

5.7.6 Threats to Validity

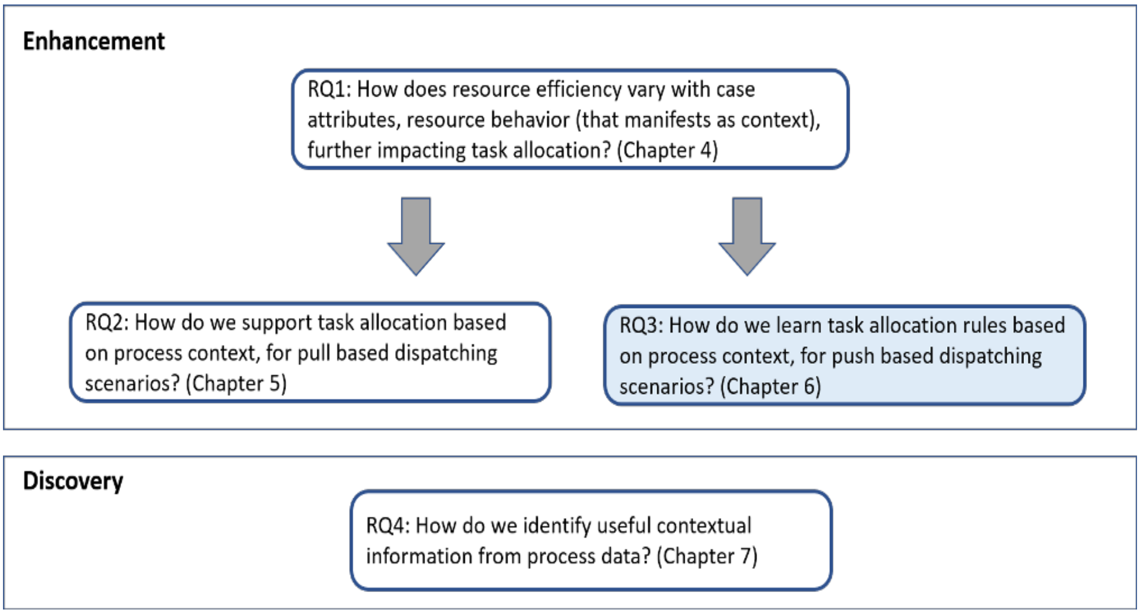
Threats to *external validity* concerns the generalization of the results from the study. This threat has been limited, by evaluating two different real-life processes and their event logs. Limitations of using event logs and lack of domain information (section 3.5), does impact the ability to provide a comprehensive list of contextual dimensions. Threats to *internal validity* arise when there are errors or biases. In this study, standard definitions of resource behavior metrics, case attributes and the ranking function have been used. There are no instrumentation errors related to changes in measuring dependent variable, as the ranking function computed on the training and test data is the same. The choice of measurements is considered as a threat to *construct validity.* Appropriate measures such as mean absolute error and root mean squared error, commonly used to measure the performance of recommender systems, have been used in the study.

5.8 Chapter Summary

This chapter shows how history of past task executions and their associated contexts can be mined to provide guidance in recommending suitable tasks to resources. Research in the past has analyzed resource behavior or context, but in isolation. The work presented in this chapter, uses context in conjunction with outcomes and provides guidance by using as input, outcomes of similar resources in similar context. Real-world event logs are used to extract resource context and discover influence of the context on performance outcome of tasks. It is however important to evaluate the interplay of contextual dimensions and their impact on the performance outcome for each business process. In the study, workload does not play a significant role in predicting outcome ratings in event log 2 (financial institute) but reduces the prediction error for the event log 1 (IT incident management). This dissertation highlights a data-driven and context-aware task allocation recommendation

Chapter 6

Learning Context-Aware Allocation Decisions



A large body of work addresses the problem of resource allocation by considering resources to have homogeneous performance or efficiency [9], [52]–[56]. However, in practice team leads and managers allocate tasks by recognizing and learning characteristics of resources and their behavior, as it impacts their efficiency. A dispatching system capable of (machine) learning the influence of resource behavior (or context), on task allocation and process performance would be essential. This chapter, presents multiple machine learning methods that use context, task, and resource attributes as input features to analyze the influence of these features on the process outcome. Such methods can be used to identify contextual factors that impact the performance outcome.

6.1 Introduction

Push based task allocation is commonly used in business processes having stringent contractual agreements on completion times, such as IT data center operations. A dispatcher, who could be a system or a person, is responsible for allocating the tasks of the process to relevant human resources or knowledge workers. The effectiveness (or even optimality) of resource allocation decisions (i.e., decisions on what resources to allocate to each process task) becomes one of the critical determinants of process performance. Event logs with with performance measures and process context can be a rich source of information for resource allocation recommendations. Context based recommendations, described in Chapter 5, requires a large amount of event data to be collected for many resources and contexts. When the past process execution data is limited, or when resources keep changing, allocation patterns need to be identified, to make allocation decisions. In this chapter, I propose an approach to support resource allocation decisions by learning allocation rules considering process context and process performance data from past process executions.

The notion of *process context* plays a key role in this decision making. As discussed earlier, process context is defined as that body of *exogenous knowledge potentially relevant to the execution of the process that is available at the start of the execution of the process, and that is not impacted/modified via the execution of the process* (in general, exogenous knowledge impacting the execution of a process can be dynamic, changing during the execution of the process, but the focus on only the knowledge that holds at the start of the execution of a process is a simplifying assumption). The process context can impact resource allocation decisions in a variety of ways. Consider a document printing process that takes as input a document and goes through a series of steps resulting in the document being printed. During office hours, the process might allocate a high-throughput (and high carbon-footprint) printer to the print task, but allocate a slower (but lower carbon-footprint) printer outside of office hours. The differential resourcing of the print task is driven by the context (specifically the time of day) which does not form part of the process data (generated or consumed by the process) but is exogenous. Process context includes resource behavior (described in section 2.2.3, as resource behavior is typically not determined, impacted, or provided as input to the execution of a process, and thus correctly belongs to the process context). Thus, for handling an insurance claim from a high priority customer, we might allocate an experienced employee as a resource (the experience or other attributes of employees do not form part of the process data - they are neither generated, impacted or consumed by the process - but have a bearing on the execution of the process). It may be noted that processes can always be re-designed to incorporate context dimensions as process inputs, but such an approach is not particularly useful given the complexity of the process designs that it would result in (consider, for example, the complexity of a process design that incorporates XOR branches for each distinct resourcing modality for a task). The proposed approach involves the use of two data mining techniques: (1) Decision tree learning and (2) the k-nearest neighbor (k-NN) algorithm. With the former, process context and a history of past process instances (each instance consisting of set of tasks executed, the relevant process data and a set of outcomes or performance indicators) are considered to compute a decision tree which enables in predicting the performance of a process instance. The decision tree thus obtained can also be used to extract rules correlating contextual knowledge with process data when the intent is to guarantee a certain set of outcomes (in other words, a certain performance profile). Given that resource characteristics typically form part of the process context, these rules can be valuable in determining the attributes of the resources necessary for achieving desired performance. With the k-NN approach, k-NN regression is used to determine from the nearest neighbors of a process instance, those values of the process context (and particularly those that characterize resources) that would likely lead to the desired outcomes. The evaluation of the approach is presented using both a real-world dataset and a synthetic dataset.

The approach proposed is of considerable practical value. Conventionally, the decision taken by a project or team lead (in many practical process resourcing settings) is based on human judgment, experience and on her implicit understanding of the context. Consequently, resource allocation activity is subjective and relies on the experience of a project or team lead. Automated, data-driven support can be used to reduce human errors and aid human judgement.

6.2 Example

For the purpose of illustration, an example process that is adopted throughout this chapter is presented. Consider a process that supports enhancements to an existing enterprise resource planning application with feature requests from customers. Enhancements are small incremental changes that are made to an existing application. Examples include adding a new report, updating a web page and so on. Figure 6.1 illustrates the business process. The process starts with a 'Developer' understanding the requirements and creating a technical design or specification (task 'Create Technical Specification'). A 'Reviewer' reviews the design (task 'Review Design'). After the review, a decision is made to either rework on the design or proceed with the implementation of the feature and test it ('Rework Design' or 'Implement and Test' respectively). This is followed by review of the code by the reviewer ('Review

Code'). A decision is made to either rework on the code or end the process if there are no major updates required in the implementation. If there are changes required, the developer reworks on the code ('Rework Code') and ends the process.

In this process, there are certain attributes, defined as a part of the process design: the enhancement complexity indicating the complexity of the work, technology module (e.g. database updates, web page update or creating a new report) and the resources authorized to work on these tasks based on their roles and capabilities. There are certain aspects that are dependent on the environment or situation during process execution: the utilization of reviewer at a specific instance of time (the number of other design or code review tasks the reviewer is currently working or number of tasks waiting in the queue of the reviewer), the preference of a developer to work on a given type of enhancement and the collaboration of the developer with the reviewer. These aspects do impact the process execution but are not modeled as a part of the design and become contextual characteristics of a process instance or resource. In this process, if the implementation (or development) of an enhancement for a module had been handled by one developer in the past, then it is preferable to allocate a similar enhancement work to that developer. Contextual characteristics of the resource and process instance are considered during task allocation and forms a part of the experience gained by person allocating tasks.

Process outcomes are another important aspect that are defined and need to be assessed during execution. In this process, there is a goal set for the completion time of the process: A simple enhancement should take no more than 2 days and a complex enhancement type should take no more than 5 days. A process instance may be successful or may fail in meeting the goal. The approach presented in dissertation, uses the process outcome, process instance attributes, contextual characteristics of the process instance and resources involved to discern allocation rules .

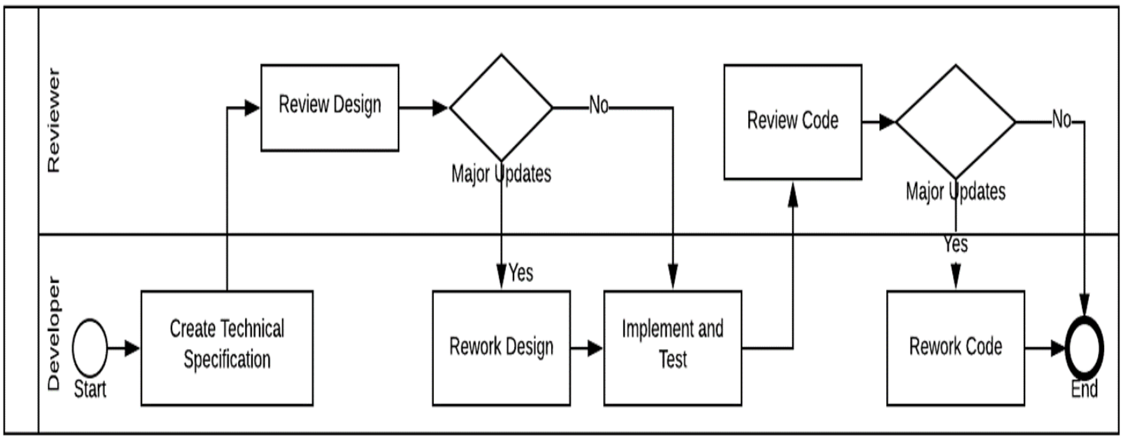


Figure 6.1: Process model to make enhancements to an enterprise application

6.3 General Setting

In this section, the notion of process context and the key data items that are used by the data mining machinery is detailed. The intent of process context is to capture the knowledge/data that does not fall under the ambit of the traditional notion of *process data* (or *process attributes*) but can be an important determinant of the performance of a process instance. It is critical that only exogenously determined data (i.e., determined not by the process but by the "rest of the world") constitute the process context. In contrast, process attributes (or process data) include endogenously determined elements (i.e., attributes whose values are determined via the execution of the process) as well data provided as input to the process. In general, the process context can be dynamic, i.e., exogenously determined knowledge relevant to the process might change while the process executes. For the purposes of this dissertation, a simplifying assumption is made, that only the context that holds at the start of the execution of the process is of interest. In this work, process context is defined to contain exogeneous information of the process that includes resource behavior. Thus the experience of a 'developer' is part of the process context in the example in the previous section. Contextual knowledge unrelated to resources can also be of interest. For instance, a history of process executions of a insurance claim handling process might suggest that these tend to perform poorly (in terms of completion time, cost or number of problem escalations) during periods of financial market volatility. Thus financial market volatility might be an important contextual dimension that determines the performance of the claim handling process. In this dissertation, the term contextual dimension and contextual attribute are used interchangeably - attribute is a term that is widely accepted in the business process modeling community, while dimension is commonly used when context is referred to in other domains such as e-commerce and ubiquitous computing.

The context can be of two types i) generic and relevant to all processes and ii) domain specific [83]. Some of the generic contextual characteristics defined in [83], are reusable across processes, while the domain specific contextual characteristics need to be identified by a domain experts.

Process context is modeled by a set of attribute-value pairs C. Other approaches to modeling the context are possible, such as truth-functional assertions in an appropriate logic, but this approach is quite general, and the overall framework remains valid even if alternative representation schemes are adopted, for the context. The knowledge about the resources available to a process is also part of the contextual knowledge that can brought to bear (resource attributes are typically not part of process data, and hence satisfy the definition of what can be deemed to be contextual knowledge). As resources are the critical entities in task allocation, for convenience, knowledge about resources is denoted as $C_r \subseteq C$ and to denote those parts of contextual knowledge that do not pertain to resources by C_p where $C_p = C - C_r$. A set of attribute-value pairs X is used to denote *process data* in the usual sense, i.e., data provided as input to a process, data modified or impacted by a process and data generated as output by a process. The *signature* of X (i.e., the schema for process data) is associated with a process design while an actual set of attribute-value pairs are associated with a process instance. A is used to denote the set of all activities that form part of a process design. Finally, I am interested in the (non-functional) *outcomes* (or *performance*) of a process (the aim is to predict these for a process instance, and to provision processes to achieve desired outcomes). A set of non-functional attributes (or QoS factor)-value pairs O is used, to denote the outcome of a given process instance. The *signature* of O is associated with a process design, and represents the set of non-functional attributes that can be used to assess the performance of an instance of that design.

The approach relies on being able to mine an *execution history* represented by a set of *process instances* and their associated *process contexts*. On occasion, a record of a partially-executed process instance is also leveraged for determining the best resource to allocate to process task (based on knowledge mines from the the execution history).

Definition 1. *Process Instance A* process instance *is a tuple* $PI = \langle v_x, v_a, C, v_o \rangle$, *where:*

- $v_x = (v_x^1, \ldots v_x^i) \subseteq X$, *is a set of attribute-value pairs representing available process data for that instance.*
- $v_o = (v_o^1, \ldots v_o^j) \subseteq O$, *is a set of* $\langle$*non-functional-attribute, value*$\rangle$ *pairs or* outcomes.
- $v_a = (a_i | a_i \in A \wedge f_{executed}(a_i) = true)$, *set of activities that were executed in that process instance,* $(f_{executed}(a_i) = true) => $ *activity* a_i *was executed in the process.*
- C, *is a set of attribute-value pairs of the process context*

6.4 Proposed Approach

The approach consists of three phases: the modeling phase, the data extraction phase and the learning phase (see Fig. 6.2). The *modeling phase* involves defining the process instance (process data, process activities, process context and process

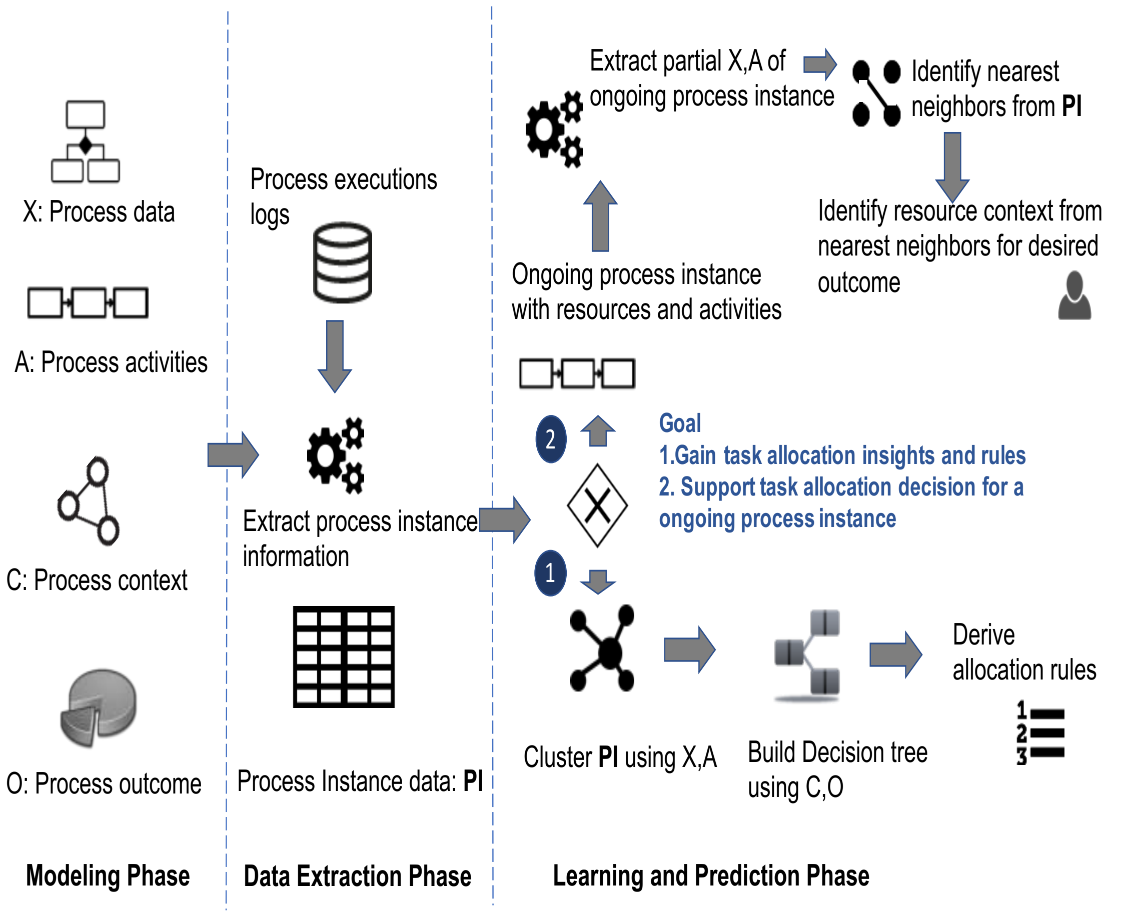


Figure 6.2: Approach for context-aware analysis of resource allocations

outcome). The *data extraction phase*, involves using historical process execution logs to extract the process instance data. Relevant performance outcome measures such as completion time of the task or quality of the task are extracted or derived from the event logs. In the *learning and prediction phase*, the intent is to provide data-enabled decision support for allocating resources to process tasks. This is achieved in two ways: (1) By applying *decision tree learning* and (2) By deploying the *k-nearest neighbor algorithm*.

Decision tree learning: The key problem to solve is as follows. *Given:*

- An execution history of process instances and their associated process attributes as defined above and
- A description of the process context as defined above,

Compute:

- A decision tree which enables us to predict the performance of a process instance.

Given the decision tree that is mined, the following questions can be answered:

- Given a specification of context and process data, predict the performance of the process.
- Extract rules from the decision tree that identify what states of the context and what process data are likely to lead to a given process outcome. These rules provide important guidance in process provisioning decisions.

In both the above modes of leveraging the decision tree that is learnt, I rely on the important observation that *the context often contains detailed knowledge about the resources that might potentially be used in a process instance.* The knowledge about resource-task pairs is represented, in the context. For example the experience of the resource used for the 'Implement and test' task is represented as a separate context attribute as compared to the experience of a reviewer performing the 'Review Design' task. The resource context attributes tells us what the experience of the *developer* was when performing a specific task, independent of the identity of the specific individual and not about resources in isolation (e.g., a specific person, or a specific machine).

First, process instances are clustered using the process attributes (v_x) and process activities (v_a, indicating the path of the process). *Clustering*, groups the process instances in a way that similar clusters have similar process attributes and execution paths. The process attribute values are used as features. The process execution path can be encoded as features in multiple ways [137]. A frequency based encoding is used where each feature represents the activity and the value of the feature is the frequency of the activity in the case. Two-step clustering[a] method is used, as it is capable of handling both categorical and numerical data, and identifies the optimal number of clusters from the data. However, any Gaussian mixture model based clustering with a suitable distance metric to identify (dis)similar process instances can be used [138]. The intent behind clustering is to mine decision trees only from clusters of similar process instances, and not from across the board.

The next step is to generate a decision tree model using the outcome(s) (v_o) as the target variable(s) and context attributes as predictor/independent variables (C). The approach is best illustrated in the example in Figure 6.3. The leaf node of decision tree in Figure 6.3 is the process outcome. At each branch, a branching criterion is used for determining which predictor variable is best suited to split process instances. At the first branch, experience of the resource working on 'Implement and Test' is used to split the process instances. 49.6% of the process instances have the value $ImplementationTask.ResourceExperience <= 0$ (representing lower experienced resource). The remaining 50.4% of the instances have

[a]http://www-01.ibm.com/software/analytics/spss

$ImplementationTask.ResourceExperience > 0$ (representing higher experienced resource). The next split of the tree, is based on the process context 'caseHandling' which indicates if the same reviewer is performing the the 'Review Design' and 'Review Code' task. The percentage of process instances having a specific value of the process outcome, is available at each node. The next predictor used for splitting the tree is the workload of the developer working on the 'Implementation and Test' task. There are additional splits based experience of the resource working on the 'Review Design' (the branches further on, have not been detailed due to lack of space). Given the attributes of a resource-task pair, the tree helps predict the process outcome. In Figure 6.3, if $enhancementType = complex$ and if the experience of the developer is low and the reviewers of the design task and code are different (represented by the branch $caseHandling \leq 0$), then the probability of meeting the service level is low $15\%, (0.43 * 0.35 = 0.15)$.

k-Nearest Neighbor (k-NN): This approach is one of the options available when the intent is to provide decision support for allocating resources to process tasks in partially executed process instances. The process data, the sequence of tasks executed thus far in the process instance and the desired outcomes (assignments of values to non-functional attributes), are provided as input. The k-nearest neighbor algorithm [98] identifies past process instances that are similar to the current instance. k-NN regression is used to identify the contextual conditions (specifically, those parts of the context that represent knowledge about the resource-task pairs) which would lead to the desired outcome/performance of the instance. k-NN regression thus provides the attributes of the resource to be deployed for a given task. Using the same setting as the example in Figure 6.3, k-NN regression might tell, based on neighbors most similar in terms of process data and the partial sequence of tasks executed, that using a developer with less workload is most likely to lead to a good outcome (in this case, service level being met). k-NN regression relies on averaging attribute values of the nearest neighbors. For discrete-valued attributes, a majority voting of the nearest neighbors is used.

6.5 Evaluation

This section presents two evaluations: first, using synthetic execution logs and second, using a subset of a real-world event log. Evaluation of the synthetic data aims to verify the ability of using the approach to discover context dependent task allocation rules. The real-world data is used to validate the possibility of extracting context and gain insights using event logs.

6.5.1 Evaluation using simulated process instances

The synthetic data is created by simulating process instances of enterprise application enhancement process, described in Section 6.2. The context comprises of the process context C_p and resource context C_r.

Attributes of $C_p = \{enhancementSpecification, customerTimeZone, caseHandling\}$
$enhancementSpecification$ captures how well the specification has been defined by the customer. If the customers have provided clear requirements ($= true$), the design specification would be well defined. A $false$ value, indicates low clarity and hence, the specification would need to be refined at multiple stages during the enhancement.
$customerTimeZone$ is the difference in number of hours between time zone of the customer and the time zone of the development team.
$caseHandling$ is a domain specific context attribute and is set to $true$, if the reviewer who reviewed the design, reviews the implemented code and is set to $false$ if they are two different reviewers.

Attributes of $C_r = \{Experience, Preference, Collaboration, Utilization\}$
Context of a resource includes availability, competency, experience, collaboration sensitivity, age, gender and so on [83]. Further, some of these resource contextual characteristics include behavior of the resource such as utilization, preference and collaboration have been identified and measured in the previous work [13], [57], and described in section 2.2.3.

The schema for process data is given by

$X = \{complexity, moduleName\}$
The $complexity$ can be set to 'complex' or 'simple' and is decided based a well defined set of information provided by the customer and $moduleName$ indicates the business module that needs a change: supply chain, financial module, account management and so on.

$O = \{completionTime, metServiceLevel\}$
$completionTime$ is the time taken for the process to complete. $metServiceLevel$ refers to meeting the service levels defined for a customer. In the example scenario, if the enhancement is complex, then the $metServiceLevel$ is true if $completionTime \leq 5d$, (d indicating days), and if the enhancement type is simple, then $metServiceLevel$ is true if $completionTime \leq 2d$.

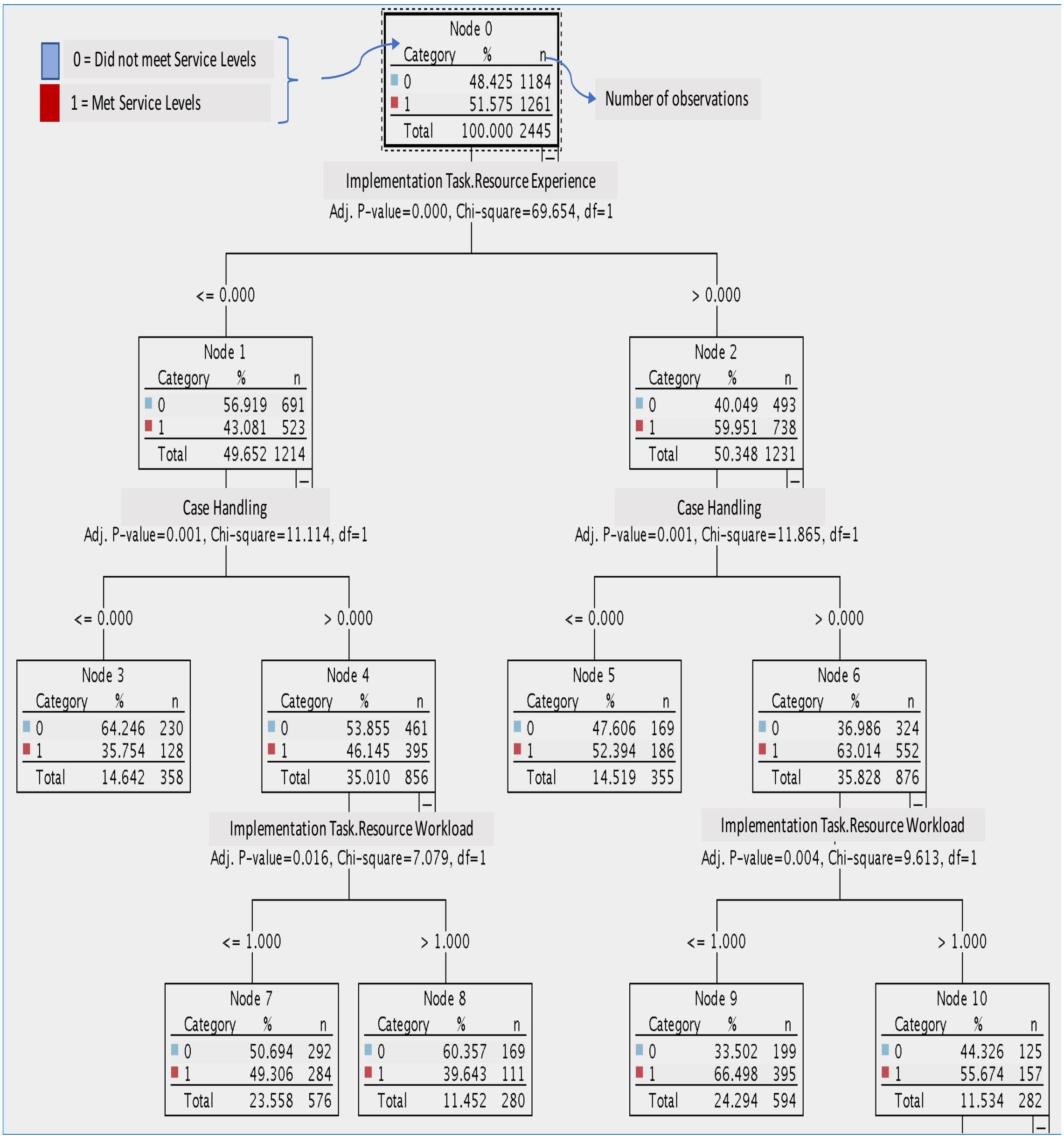


Figure 6.3: Decision tree depicting one path from root node to leaf nodes for 'complex' enhancements predicting 'metServiceLevel'

Process Instances generated for the model

A simulation model is used to generate process instances based on the example process. Gaussian distribution functions are used to generate values for the context of process, resources and process attributes. The completion time is generated by considering the context and attributes of the process as indicated in code listing 1. The input $\mu_{complex}$, $\sigma_{complex}$, μ_{simple}, σ_{simple} are the mean and standard deviation of the completion time for tasks with high and low complexity, respectively. There is additional randomness added to the generation of completion time to imitate real-world settings. Ten thousand process instances are simulated. The generated process instance data is used to evaluate resource allocation rules using decision tree learning and k-nearest neighbor methods.

```
Input: ProcessInstance, μ_complex, σ_complex, μ_simple, σ_simple
Output: completionTime
c = getNextGaussianValue()
if c ≤ 0.4 then
    complexity = complex
    completionTime = μ_complex + getNextGaussianValue() * σ_complex
end
else
    complexity = simple
    completionTime = μ_simple + getNextGaussianValue() * σ_simple
end
if complexity = complex then
    if designReviewWorkload = high then
        completionTime+ = getRandomValueInRange(0, 0.15) * μ_complex
    end
    if ImplementationWorkload = high then
        completionTime+ = getRandomValueInRange(0, 0.10) * μ_complex
    end
    if noCaseHandling then
        completionTime+ = getRandomValueInRange(0, 0.15) * μ_complex
    end
    if ImplementationExperience = high then
        completionTime+ = getRandomValueInRange(0, 0.25) * μ_complex
    end
    //additional updates to completion time based on other attributes
end
if complexity = simple then
    if ImplementationPreference = high then
        completionTime+ = getRandomValueInRange(0, 0.15) * μ_simple
    end
    if ImplementationWorkload = high then
        completionTime+ = getRandomValueInRange(0, 0.25) * μ_simple
    end
    if customerTimeZone > 6 then
        completionTime+ = getRandomValueInRange(0, 0.25) * μ_simple
    end
    if ImplementationExperience = high then
        completionTime+ = getRandomValueInRange(0, 0.07) * μ_simple
    end
    //additional updates to completion time based on other attributes
end
```

Code Listing 1: Computing completion time of simulated process instances

Decision tree learning: This step starts with clustering the process instances based on process attributes. The process instances are clustered based on a pro-

Predictor	Predictor Importance
Implementation Task.Resource Experience	0.329
Case Handling (Same reviewer for review design and review code)	0.183
Implementation Task.Resource Workload	0.11
Review Design Task.Resource Experience	0.10
Customer Timezone	0.08

Table 6.1: Importance of predictor with metServiceLevel as the target for 'complex' enhancement

cess attribute *complexity* indicating if the enhancement is simple or complex. A decision tree is built with the metServiceLevel as the target variable and context as predictor. Chi-square Automatic Interaction Detection (CHAID) algorithm is used to construct the decision tree [139]. Table 6.1 shows the predictor importance. The most important predictor is the experience of the resource performing the 'Implement and Test' task. The other resource context variables such as workload of the resource performing the implementation task, the experience of the reviewer working on the 'Review Design' task enable predicting the process outcome. The decision tree model (Figure 6.3) predicts the outcome with 64% accuracy. Table 6.2 presents the model evaluation metrics. The task allocation rules can be derived from the decision tree [140]. Production rules derived from the decision tree, can be much smaller than the number of leaves in the decision tree and are determined by filtering paths that are redundant or paths that do not improve the accuracy significantly. One of the production rules for task allocation would be:
$if(complexity = complex) \wedge (Experience.ImplementationAndTest = HIGH) \wedge$
$(CaseHandling = true) \wedge (Workload.ImplementationAndTest < 1)$
then $(metServciceLevel = true)$
The variable $Experience.ImplementationAndTest$ implies the value 'Experience', of the resource performing 'Implementation and test' activity.

Decision tree can be trained for the other cluster of process instances based on the process attribute *complexity* = *simple*. The predictor importance, the evaluation measures for *complexity* = *complex* and *complexity* = *simple* are presented in Table 6.3, and Table 6.4 respectively. Note that the predictor importance changes and hence, the resource allocation rules derived from the decision tree would be different.

Dataset	Precision	Recall	F1	Accuracy
Train	60.7	69.68	64.9	60.27
Test	57.29	64.82	60.82	58.46

Table 6.2: Decision tree prediction metrics for 'complex' enhancement

Predictor	Predictor Importance
Implementation Task.Resource Workload	0.315
Implementation Task.Resource Preference	0.29
Customer Timezone	0.11
Case Handling (Same reviewer for reviews design and review code)	0.07

Table 6.3: Importance of predictor with metServiceLevel as the target for 'simple' enhancement

Dataset	Precision	Recall	F1	Accuracy
Train	59.3	53.4	56.25	58.99
Test	63.04	54.23	58.3	59.696

Table 6.4: Decision tree prediction metrics for 'simple' enhancement

K-Nearest Neighbor: Another useful scenario would be in supporting the decision of task allocation, during process execution. In this scenario, a process may have executed partially (or is in its initial state). The *new* executing process instance and the target outcome of the executing process are given as input. In the example, the input is provided as $complexity = \text{`}complex', Timezone = 3$ and $completionTime = 3d$. The input values are matched against past process instances. K-Nearest Neighbor algorithm (K-NN), is used to find process instances that are closest to the current process execution instance. There are distance functions that consider continuous and categorical data. Categorical values are transformed to use Euclidean distance function. For evaluation, Euclidean distance measure (described in Chapter 2), is used. Statistical packages such as SPSS[b] [141] provide an estimate of K. For the experiment, K is set to 5.

The context values of the nearest neighbors (average for continuous values and majority voting for discrete values), is used as input to find the matching resources. Table 6.5 shows the key context attributes required for the complex enhancement (process attribute) and an outcome or completionTime = 3 days and additional information of timezone=3hours (process context). The matching resource can be identified by selecting resources with the same experience, workload and preference.

A similar K-NN model, used for a process requiring enhancement that is 'simple' with an outcome or completionTime =2 days, indicates that resource with lower experience and higher preference is capable of meeting the outcome. Hence, resource context required for a process outcome varies with the process attribute values.

[b] http://www-01.ibm.com/software/analytics/spss/products/modeler/

Case Handling	Implementation Task.Resource Experience	Design Review Task.Resource Experience	Implementation Task.Workload	Implementation Task.Resource Preference
1	HIGH	HIGH	1	0.11
1	HIGH	HIGH	1	0.64
1	HIGH	LOW	0	0.14
1	LOW	HIGH	0	0.68
1	LOW	HIGH	2	0.78

Table 6.5: Nearest neighbors and resource recommendations for complex enhancement with 3 days as completionTime

6.5.2 Evaluation using real-world event log

The approach is evaluated on a real-world event log. To this end, the logs from the BPI Challenge of 2013 [133] is used. The data set comprises of logs from an IT incident management system. An incident is created when there is an issue in the IT application. Each incident or issue, has an associated impact and relates to a product of the enterprise. A resource or worker, is allocated the task of resolving the incident.

Lack of information about the domain, limits the ability to model process attributes or the context of the process. Hence, the process context model is limited to generic attributes such as $TimeOfDay$ of the incident. The process attributes are the *impact* of the incident and *product* associated with the incident. The organization involved in resolving the incident is available. The support team is divided into '1st', '2nd' and '3rd' levels. The resource context is derived from event logs. Resource behavior measures that are computed from event logs, is used for capturing the resource context. The contextual dimensions used are computed as follows:

- Preference: It is the ratio of the number of incidents associated to a product that the resource has worked on in the past to the total number of incidents the resource has worked on.

- Utilization: the number of incidents that the resource worked on the past five days. This indicates how busy the resource has been.

- Experience: the experience of a resource is derived based on the assumption that a resource who has solved incidents that belong to support teams '2nd' or '3rd', is experienced [117]. This information is derived by counting the number of incidents a resource has solved that belonged to support teams 2nd or 3rd (representing levels). If the resource has solved these incidents, then it is likely their experience is higher than resources who have not.

The evaluation is done on a subset of the instances where a single resource resolves an incident. Event logs involving multiple resources, do not provide clarity the time spent by each resource on the incident and hence is not used. These account for 3460 incidents.

The process outcome is based on the completion time. A target binary variable is set to 1 if the completion time of the incident is lower than a threshold, and set to 0, if the completion time is higher than the threshold . A decision tree is built using the process attributes and context. The prediction accuracy and f1 score are presented in Table 6.6. Table 6.7 presents the predictor importance for the process outcome. The utilization of the resource impacts the outcome, followed by the preference and experience of the resource. The impact of the incident, is a process attribute, that influences the outcome. Experience of the resource has lower importance in the model. In this model, only two categories of experience levels (Low, High), were defined, based on the support team the resource belonged to.

Dataset	Precision	Recall	F1	Accuracy
Train	62.3	81.96	70.75	80.96
Test	58.63	78.49	67.13	78.49

Table 6.6: Prediction metrics for the incident management logs containing process instances belonging 'Org line C'

Predictor	Predictor Importance
Utilization	0.45
Incident Impact	0.21
Preference	0.16
Experience	0.16

Table 6.7: Importance of predictor with metServiceLevel as the target

There could be several other factors, that could influence the outcome, which have not been used for the evaluation of real-world event log. This requires access to additional information in the event logs and additional domain specific information. However, the current results indicate, that process context has an impact on the process outcome.

6.6 Threats to Validity

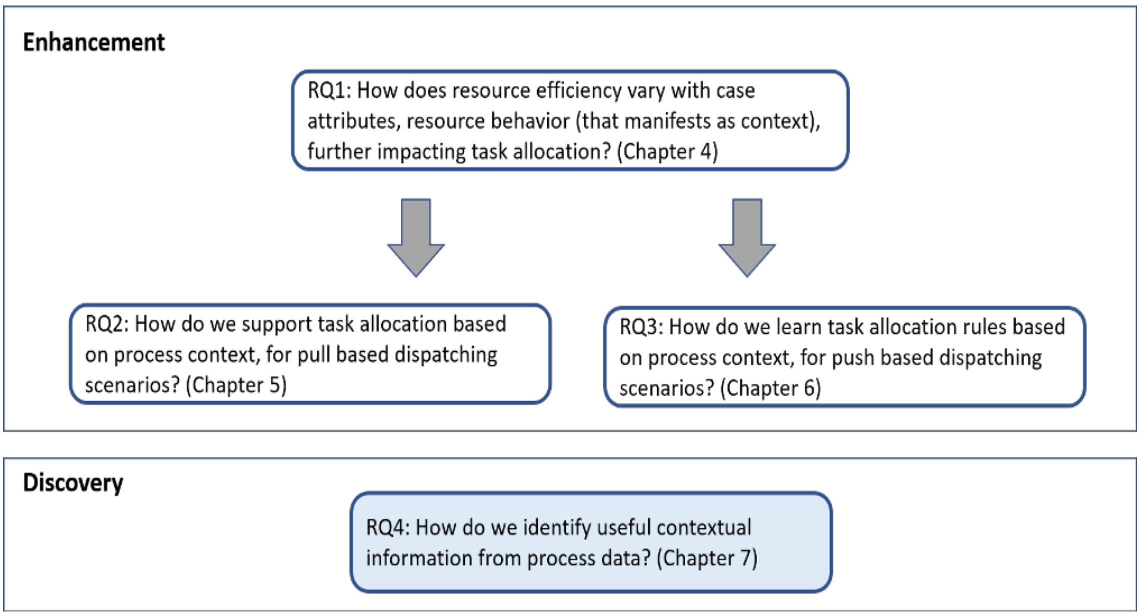
In this section, the limitation of the study with respect of construct validity, internal validity, and external validity, is identified. *Construct validity* denotes that the variables are measured correctly. The dependent variable (meeting service level) and independent variables (resource behavior, process context) have been studied previously. Standard metrics have been used to compute these features from event log. *External validity* concerns the generalization of the results from this study. The study was conducted using a generated synthetic log and one real-life event log. While insights can be drawn from the study, I do not claim that these results can be generalized. Further studies need to be conducted on other real-life event logs to affirm generalizability of the results. However, the results serve as the basis of using context when learning dispatching policies. Lack of information about the domain and the logs limits the ability of having a comprehensive list of input features or independent variables (as discussed in section 3.5). *Internal validity* is established for a study if it is free from systematic errors and biases. The real-life event log contained data collected over a period of 4 months. During this measurement interval, issues that can affect internal validity such as mortality (resources leaving the organization) could have occurred. However, since generic resource behavior measures are used, the impact of this threat is limited.

6.7 Chapter Summary

This chapter shows how a history of past process instances and their associated contexts can be mined to provide guidance in resource allocation decisions for a currently executing process instance. The work presented in this dissertation, uses resource context in conjunction with additional task context and outcome. There are multiple advantages of this scenario: i) in a push based dispatching system, an approach such as this would be useful in analyzing the resource context and making relevant recommendations. ii) It would be useful to get insights on the situations or process and resource context that either lead to a successful or failed process outcome. Such insight can be used to re-engineer and consider important contextual dimensions as a part of the process design. In the method, process and resource context have to be defined by domain experts that requires experience and deep understanding of the process execution. The next chapter, explores a method to mine process and resource context from execution logs.

Chapter 7

Mining Context from Unstructured Process Data



Process logs contain textual information with comments or notes added by resources, when performing tasks. Earlier studies have used textual information in process logs to identify suitable teams [75], predict deviant cases [69], and determine repetitive problems or solutions related to IT incidents [91], [92]. Not much work has been done in mining process context from textual data. This chapter presents a method of using the textual information to identify context that could impact process outcome. The work presented here is semi-automatic, filtering large amount of textual information, thus enabling a domain expert to manually categorize small amount of text snippets as context.

7.1 Introduction

Observing and analyzing impact of the context of a process or the environmental factors, on its execution outcome helps adapting and improving the process. Dourish [82], has presented two views of context (detailed in section 2.5). First, a representational view: context is a form of information that is stable, can be defined for an activity and is separable from the activity. Here, context is information described using a set of dimensions that can be observed and collected. Second, an interactional view: context is a property of information that may make it a context depending on the activity, can be dynamically defined and is produced by the activity. Modeling of context considers the representational view, which is termed as explicit context: information that is identified by domain experts and can be defined a priori. However, there are some situations that arise as a part of performing a task or an activity (interactional view), and may not be known a priori. These *implicit contextual dimensions* need to be discovered from various sources of information.

Saidani et al. [16] define a meta-model of context for a business process. The meta-model comprises of context entity, context attributes and context relationships. A domain expert can define a context model based on the meta-model and the contextual information can be observed from the process execution logs. For example, in the loan management application, a domain expert would indicate that the time of submitting the loan application is a contextual information, as the process path and outcome could vary depending on month of the financial year. The previous chapters have focused on learning and predicting using explicit process context extracted from structured information in event logs. Consider another example of an IT application maintenance process where, a problem ticket could contain the name of the application facing a glitch or issue, the severity of the issue and other details. Additional data, such as the knowledge worker or resource assigned to work on the problem ticket, the time the issue is created, are used to compute contextual dimensions such as the experience of the resource working on the ticket, the shift time when the issue was created. The process performance and behavior is analyzed based on the contextual dimensions. The contextual dimensions for the analysis are defined by domain experts. The term 'contextual dimensions' is used in line with existing literature on context aware recommender systems [14]. These dimensions are characterized as *explicit contextual dimensions*.

In practice, there are additional implicit contextual dimensions that arise from the task and could impact the process performance. For example, when performing the task of resolving an IT problem ticket, the resource may find that certain legacy applications require much more time as multiple interlinked applications need to be restarted, while an application using web services takes less time as it requires restart of just that specific web service. This information is implicit and once identified, the process re-design could assign different resolution times based on the new contextual dimension called 'application type' with two values - legacy application or service-oriented application. The source of identifying the underlying implicit context can be from unstructured information available as textual comments that are recorded during the process execution indicating, restarting of several related applications for a legacy application.

In this dissertation, the problem of exploiting unstructured textual data to discover implicit context is studied. In the proposed framework, phrases of textual data are extracted from relevant textual logs of process instances. These phrases or nuggets of information are clustered. The clusters are semi-automatically pruned by applying filtering rules considering performance outcome, to arrive at subset of textual clusters that are likely to relate to implicit contextual information and impact process outcome. The final decision of the information being a contextual dimension or not is made by domain experts. To the best of my knowledge, discovery of process context from unstructured or textual data available with process execution histories has not been considered so far. To summarize, the following contributions are made in this chapter:

- Introduce the research problem of mining context from textual information available during the process execution.
- Propose an unsupervised approach of identifying context, that is strongly suggestive of situations during process execution and salient to domain experts.
- Filter information mined from textual logs by correlating with process outcomes to identify relevant contextual dimensions.

7.2 Motivating Example

I motivate the problem using the textual information logged in a real-life business process for maintaining IT applications. Table 7.1, contains textual information logged by workers or resources involved in the process of maintaining IT applications. A problem is reported by a customer. The resource or worker allocated to the task, evaluates the problem, identifies and executes relevant resolution, confirms with the customer if the problem has been resolved. At every step in the process of analyzing and resolving the problem, the details are recorded in an incident management system (process aware information system). Examples in Table 7.1 are representative of typical challenges with textual logs of business processes: i) varying informativeness from being very brief to very detailed, ii) containing ill

No.	**Communication log of the problem tickets recorded by knowledge workers**
1	emailed user. *waiting for user to get back to me.* emailed user. looking for response. User confirmed that the issue is not replicated. Hence closing the incident.
2	Left a voicemail for customer at the number provided in this ticket. Requested he call option (one) for further assistance. Validated userid in the portal, made in Synch . Manually made in SYNC with that of GUI. Call made both on office phone and cell. *Voice sent on cell and office phone is not reachable.* 2nd call made to the customer. No response.. *3rd call made to the customer.* No response. Call closed due to no prior response from the customer.
3	Userid been unlocked, sent to user, pending confirmation. pwd sent to user, waiting for response. Second pwd sent to user, phone number provided is a warehouse phone number, nobody answers it. *No response from user,* closing the incident..
4	Peformed netmeeting with user and there are no authorization issues. user is able to run the reports. *Training issue.*
5	Requested customer to provide error screenshots. Users requested to logoff and then reopen the browser and then login again This is to check whether the users are able to view the required access or not.. Customer contacted to check whether the login access to portal is OK. *Customer confirmed for successful login.* Hence closing the ticket.
6	incorrect logon locks. unlocked the ID and reset the password. pinged user via IM. *Elli confirmed to close the incident.*
7	Password reset done in AAA and BBB for the user and user mailed. User ID unlocked. Customer confirmed of logging successfully. Hence closing ticket.
8	Validity date has been reset as per the record and sent to user. Awaiting confirmation.. Sent a agan for confirmation. Awaiting confirmation. Closing.
9	called, Attributes corrected & mail send to user
10	Received confirmation from user, closing the incident.

Table 7.1: Unstructured textual information captured during IT maintenance process

formed sentences with grammatical errors, typographical errors and abbreviations. The entry numbered 5, has detailed information of the steps taken to resolve the issue. The entries (9,10), have very limited information and hence are of little value. The characteristics of the textual information available in the maintenance of four IT applications is shown in Table 7.2. Textual data is small in terms of the number of words in a process instance log.

However, these logs reflect some common situations that arise when performing an activity. For example, 'Unavailability of the customer' could be a situation or a task context, and could impact the time taken to perform the task. The log contains both, i) information relevant to the specific process or task, and ii) information that represents context. Hence, the textual data can refer to multiple topics. In the following section, the background of concepts that can be applied to mine relevant information from the logs, specifically related to identifying multiple topics from textual documents is described.

Application	Number of process instances	Number of sentences	Average number of words per sentence	Average number of words per process log
Application Security	684	2235	10.25	44.35
Portal	210	1569	14.11	118.02
HR System	490	1482	11.87	41.38
Reporting	832	1267	9.71	20.02

Table 7.2: Characteristics of textual data in process logs of real-life IT application maintenance process

7.3 Background

This section presents well known natural language processing techniques that can be used together to mine contextual information from process logs.

7.3.1 Notations

The textual information logged during the execution of a process instance can be considered as a text document. Let each document $d_i \in D$ represent textual information logged for respective process instance $p_i \in P$. Each document could comprise of information on activities being performed, the actions taken when performing the activity and the situation or conditions during the execution of the activities. Hence, document d_i comprises of one or more topics of the topic set $T = \{t_1, t_2 \dots t_T\}$ with some topics representing the context of the process instance. The problem can be represented as a multi-label categorization of textual logs.

Further, each document d_i is represented by smaller constituents that relate to one or more topics. The smaller constituents or chunks of text are called *segments*, which in turn contain one or more sentences. A segment is small enough to contain information relevant to a single topic. In general, this assumption holds for communication logs containing short descriptions. Hence let S_i be the set of segments of document d_i, then $S = \bigcup_{i=1}^{|D|} S_i$, is a set of all segments. The goal is to find the topics T over S, and further find the topics for each document $T_i \subseteq T$ based on topics of the segments S_i of the document d_i, and hence the process instance p_i

7.3.2 Segmenting Document

The goal of breaking down the document into segments, is to identify smaller constituents that represent distinct information related to tasks or their context. There are multiple ways of segmenting text. The suitability of the method is based on the characteristics of the textual information in the process logs.

1. *Phrase extraction* using parts-of-speech (POS) patterns has been used to extract text segments [142],[91]. These are similar to regular expression patterns based on parts of speech. While, pattern based extraction has a high precision in extracting information, it has low recall as it filters phrases that do not match the POS pattern. For example, the phrases 're-provisioning completed', 'has been re-provisioned' and 're-provisioned and sent confirmation', have the same information, and yet have different POS tag patterns: 'VBG VBN', 'VBZ VBN VBN', 'VBN CC VBN NN' respectively (VBN is verb, CC is conjunction, and NN is noun, based on the listing of POS tags by Penn Treebank Project [143]). This method of segmentation is suitable when information logged by process participants is based on a standardized templates.

2. *Parse Tree* is a rooted tree that represents the syntactic structure of a sentence based on a grammar. There are two ways of constructing parse trees: 1) constituency relation that is based on phrase structure grammar, 2) dependency relation that is based on relations among words. Constituency parser can be used to break down the sentence to extract smaller noun or verb phrases. Noun and verb phrases can be used as segments of the document. Parse trees are suitable when there is very sparse data reported by the process participants. In such scenarios the information extracted, is limited to key actions recorded during process execution. For example, from the communication log on the first row in Table 7.1, verb phrases such as 'emailed user', 'waiting for user',

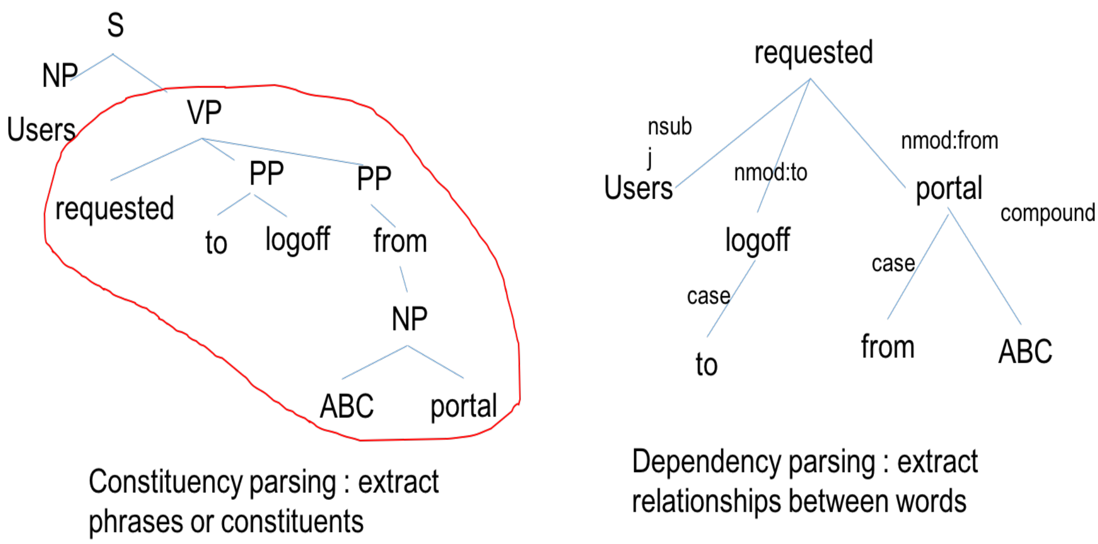


Figure 7.1: Constituency and Dependency Parse trees

'looking for response' can be extracted by using constituency parser. The two parse trees are illustrated in Figure 7.1.

3. *Extractive summarization* is an automatic text summarization method that, produces a summary of the text while retaining key information in a document [144]. There are two well known methods to summarization i) abstractive summarization, and ii) extractive summarization. Extractive summarization identifies important sections of the text and generates them verbatim. Distinct sentences of the document summary can be used as segments. Summarizing text is suitable when verbose comments are logged by process participants.

7.3.3 Clustering Methods

The extracted text segments can be categorized and grouped using different clustering methods. Common clustering methods and their suitability to grouping textual data available in process logs, is briefly discussed:

1. *Topic Modeling* Clustering approaches such as latent semantic analysis [145], probabilistic latent semantic analysis (pLSA) [146] and latent Dirichlet allocation (LDA) [113] have been used to identify representative set of words or topics. These approaches identify topics by exploiting the co-occurrence of words within documents and are well suited for multi-topic text labeling. However, they are not suitable for short documents containing limited number of words and sentences. Hence, while these methods are widely used in multi-class text categorization, they are unsuitable for textual data available in process logs.

2. *Partition based clustering* such as k-Means, k-Mediods, are the most widely used class of clustering algorithms [99]. These algorithms form clusters of data points, by iteratively minimizing a clustering criterion and relocating data points between clusters until a (locally) optimal partition is attained. An important requirement of partition based methods is the number of partitions or 'k' as input.

3. *Affinity Propagation* is one of the recent state-of-the-art clustering methods that has better clustering performance than partition based approaches such as k-Means [101]. Affinity propagation identifies a set of 'exemplars' and forms clusters around these exemplars. An exemplar is a data point that represents itself and some other data points. The input to the algorithm is pair-wise similarities of data points. Given the similarity matrix, affinity propagation starts by considering all data points as exemplars and runs through multiple iterations to maximize the similarity between the exemplar and their member data points.

7.3.4 Text Similarity

Next, the focus is on the key aspect of any clustering algorithm; the choice of (dis)similarity function or distance metric between data points (text segment pairs). A text segment, is represented as a vector and distance functions such as Euclidean distance or similarity functions such as cosine similarity are used.

1. *Bag-of-Words (BOW)*: Each text segment is represented as vector of word counts of dimensionality $|W|$, where W is the entire vocabulary of words.

2. *TF-IDF*: The bag-of-words representation divided by each word's document frequency (number of text segment it occurs). The representation ensures that commonly occurring words are given lower weight.

3. *Neural Bag-of-Words (NBOW)*: Each text segment is represented as a mean of the embeddings of words contained in the text segment. The embeddings of words are obtained using the word2vec tool [147]. The semantic relationships are retained in vector operations on word vectors, e.g., vec(Paris) - vec(France) + vec(Germany) is close to vec(Berlin). Hence, distances between embedded word vectors can be assumed to have semantic meaning.

4. *Word mover distance (WMD)*: WMD is suitable for short text documents (or text segments). It uses word2vec embeddings [148]. The word travel cost (or Euclidean distance), between individual word pairs is used to compute document distance metric. The distance between the two documents is the

minimum (weighted) cumulative cost required to move all words from d_i to d_j. When there are documents with different numbers of words, the distance function moves words to multiple similar words.

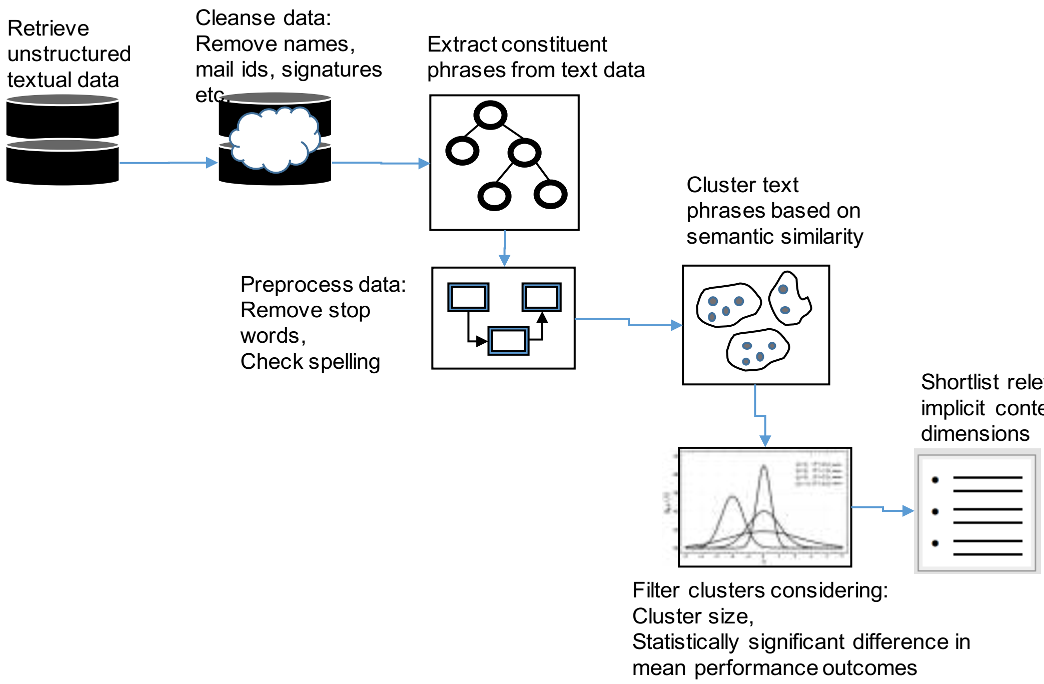


Figure 7.2: Overall approach to identify implicit contextual dimensions

7.4 Overall Approach

Our approach to infer or identify implicit context is organized into multiple steps, as shown in Figure 7.2. The approach comes down to answering three key questions: i) What are the common situations and actions taken by the performers of a process during its execution? ii) How many process instances are related to these situations? - is this a common or a rare situation? and iii) Are these representative of process context and do they impact the performance outcome of the process? The steps of the approach are discussed in detail:

7.4.1 Text Retrieval and Cleansing:

A tuple $\langle pid, ppi, text\_data \rangle$ containing the process instance identifier (pid), the process performance indicator (ppi) [149], and the unstructured textual information is extracted from execution logs. The use of each of these attributes, will be described in the following steps. The $text\_data$ for each process instance is referred to as a document. The document is processed to remove the names of people, IP addresses, HTTP addresses, and other textual data such as email signatures, phone numbers, that would not represent common actions or situations. The cleansing uses named entity recognizer[a], to detect person names, organization names. IP addresses, phone numbers, email addresses are cleaned from the text using regular expression parsers.

7.4.2 Text Segmentation:

In this step the document is broken down into text segments by extracting summaries, or by extracting phrases using constituency parsing. A suitable method is chosen based on the characteristics of textual log (sparsity, verbosity, or variety), as described in Section 7.3.2. Hence, we have $\langle pid, text\_segment\rangle$.

7.4.3 Text Preprocessing:

Each text segment goes through standard preprocessing steps i) lemmatization, where the base form of the words in the text segment are derived (e.g - allocate, allocation, allocating are replaced by their lemma 'allocate'). ii) stop word removal, where very frequent words that are likely to appear in all the documents and contain little information, are removed.

7.4.4 Clustering

The text segments are clustered using one of the similarity measures described in Section 7.3.4. This step results in grouping process instances having similar text segments. The process instance associated to each text segment and its performance indicator is used to form a tuple $\langle pid, cluster\_id, text\_segment, ppi\rangle$.

7.4.5 Filtering Clusters

The goal of this step is to identify clusters of text segments, that are important and useful to a domain expert and help discern contextual dimensions. Two filters can be applied:

Size Filter: The number of process instances associated with a cluster is a good indicator of its importance. Intuitively, if the size is very large, then the information content is a part of normal execution of the task. For example, if the number of process instances associated to the phrase 'confirming and closing loan application' is very large, it is indicative of a normal procedure. Similarly, a cluster containing very few process instances may not be useful as it may indicate an exception and has to be handled as a part of the process exception or process error management.

[a]https://nlp.stanford.edu/software/CRF-NER.html

An upper and lower bound on number of process instances is set to filter clusters.

Process Performance Filter: This filter helps identify clusters that have an impact on the performance indicators of the process. The performance indicators of a process can be the completion time, the quality outcome of the process, or any other process indicator as detailed in [149]. To verify if the performance indicators of the process instances of a cluster are significantly different from other process instances, two sample groups are considered - i) cluster group, and ii) other group. Performance indicators of all process instances in a cluster are taken as one sample (cluster group). Performance indicators of a randomly chosen set of process instances from other clusters are considered as the second independent sample (other group). The Mann-Whitney U test [129] is used to compare statistically, the variance in the performance indicators of the two groups. The test is run with multiple random samples of *other group* to reduce false positives or Type 1 error. The Mann-Whitney U test is one of the powerful nonparametric tests that makes no assumption on the distribution of data and is relevant for groups with small sample sizes (as clusters could be containing 10 process instances).

7.4.6 Context Identification

The final step of the approach is a manual verification by domain experts on the filtered set of clusters. The description in the text segments of filtered clusters are used by the domain experts to identify contextual situations that impact the performance of the process.

7.5 Experimental Evaluation

For the purpose of evaluation, first segment based clustering using different clustering methods, and similarity measures is evaluated, on a benchmark data set of multi-topic documents, as there is no benchmark textual data of business process available to evaluate the approach. Next, the overall approach detailed in Section 7.4, is used on a real-life business process textual log to identify the clusters that indicate contextual information.

7.5.1 Evaluating Clustering of Text Segments:

The Reuters-21578 text categorization collection is a text categorization benchmark [150]. The *Mode Apte* evaluation, is used in which unlabeled documents are removed. There are 10787 documents that belong to 90 categories. The collection has a training set containing 7768 documents and a test set containing 3019 documents. Two main constraints are set up on the data: 1) each document should be assigned to at least 3 topics or categories, 2) each category or topic must have at least 1% of the documents. The training set is used to set the parameters for affinity propagation, choose K for k-Means, and group text segments into the same number of clusters as the categories in the collection (68 categories in this case).

The quality of segment based clustering is evaluated on the test data containing over 900 segments on 95 multi-labeled documents, using the commonly used criterion of *precision, recall* and *F1 measure* [151]. Two approaches are used to compute the measures for multiple categories. The Precision, Recall, F1-measure is computed for each category. Finally, the overall measure is obtained by averaging category specific Precision, Recall and F1 measure. This is known as macro-averaging $(Prec_M, Rec_M, F1_M)$. The other approach is based on computing a confusion matrix of all the categories by summing the documents that fall in each of the four conditioned sets, namely true positives, true negatives, false positives, and false negatives. The Precision, Recall and F1 measure is computed with the overall confusion matrix. This second measure is known as micro-averaging $(Prec_\mu, Rec_\mu, F1_\mu)$.

The results are presented in Table 7.3. Text segments for each document are created by using extractive summaries. As K-Means algorithm is based on Euclidean distance between two pairs, word mover distance is not evaluated. The results indicate that using affinity propagation based clustering, provides better F1 scores as compared to K-Means. Euclidean distance of NBOW and WMD measures result in higher macro-average and micro-average F1.

		Macro-Average			**Micro-Average**		
Clustering	**Similarity**	$Prec_M$	Rec_M	$F1_M$	$Prec_\mu$	Rec_μ	$F1_\mu$
K-Means	BOW	0.772	0.442	0.491	0.385	0.490	0.431
	TF-IDF	0.583	0.586	0.534	0.552	0.447	0.495
	NBOW	0.665	0.538	0.530	0.55	0.467	0.503
Affinity Propagation	BOW	0.705	0.450	0.448	0.341	0.535	0.417
	TD-IDF	0.648	0.548	0.568	0.614	0.483	0.541
	NBOW	0.637	0.626	**0.580**	0.570	0.516	**0.542**
	WMD	0.652	0.593	**0.584**	0.631	0.470	**0.540**

Table 7.3: Comparative evaluation of multi-class categorization for various distance measures and clustering methods

7.5.2 Context Mining from Text Logs

The overall approach of identifying contextual information is evaluated on the IT maintenance process of three different applications of a large media and entertainment organization. The textual data recorded varies significantly for different application domains such as security, human resources, finance and web portal. The process consists of four main tasks: 1) customer creates an application problem ticket, 2) the worker acknowledges the receipt the ticket, 3) the worker analyzes the issue and resolves the problem, 4) on resolving the problem, the worker confirms with the user, and 5) the worker closes the ticket. At each step, the workers log their findings or progress. In some cases, emails sent or received by the customer and the worker is logged in the system. The communication or task logs associated with each process instance is analyzed.

To evaluate the overall approach of mining contextual factors from textual data, the pipeline of steps detailed in Section 7.4 are executed. Table 7.4 presents the descriptions derived from the text segments in the filtered clusters. As shown, for the 'Security' application, of the 2493 text segments extracted from all the process instance documents, clustering using affinity propagation with WMD, results in 119 groups or categories. The mean completion times of the process instances in these groups is compared to mean completion time of a random number of other process instances. A statistically significant variance in the mean completion time (the performance outcome), is used to filter few clusters. Further, filtering of clusters is done based on the size of the cluster. For example, 'confirm and close incident' is a very common text segment that is identified and associated with several process instances. It occurs in 50% of the process instances. It may hence, be a process completion step and not a situation or context. The highlighted descriptions in the table are examples of context.

Based on the cluster labels in Table 7.4 (that are derived from common text in the clusters), for the security application, it is observed that any process instance associated with *reset password* has lower completion time (indicated with a + sign in the table), as the task is extremely specific . The clusters further highlight a key situation of not being able to contact the customers, leading to the process being set to 'pending' status and the completion time being much higher than other process instances. Identifying such a situation can help re-design the process to account for customer unavailability. Similarly in the maintenance of the portal application, waiting for *more information* from the user leads to higher completion time of such tasks. A template with all relevant information recorded by the customers when creating the problem ticket, could be a plausible solution. In the HR domain application, the number filtered clustered were limited and the clusters did not provide useful insights on context.

Figure 7.3 visually depicts a subset of clusters of the textual segments. The NBOW vectors of text segments is represented on a two dimensional space. The textual segments are the noun and verb phrases extracted using constituency parser.

App. Domain	# Text Segments	#Clusters	**# Filtered**	**Cluster labels**
Security	2493	119	13	1. **(2nd call, 3rd call) made to the customer** 2. (researching, working, fixing) issue 3. (asked, sent, mailed) to check again 4. **waiting for (approval, confirmation)** 5. could not (read, get, contact) user 6. waiting for user 7. **reset password (sent, mailed) user** (+) 8. **changing status to pending** 9. **tried calling the user** 10. . . . . . .
Portal	2025	170	22	1. **sent to the user for (confirmation, information)** 2. **waiting for user (confirmation, email)** 3. **moved support issue to development** 4. **getting more details on the issue** 5. called and left a voice mail 6. . . . . . .
HR System	2092	189	27	1. **(were, tied to, failed) data issues** 2. closing the incident (+) 3. **need to upgrade to breakfix** 4. (write,call) back to me 5. . . . . . .

Table 7.4: Filtered Clusters of IT application maintenance process logs, (+) indicates clusters has lower completion times

7.5.3 Discussion

The approach presented in this chapter, helps identify information from the logs that could indicate contextual information or situations that should be addressed.

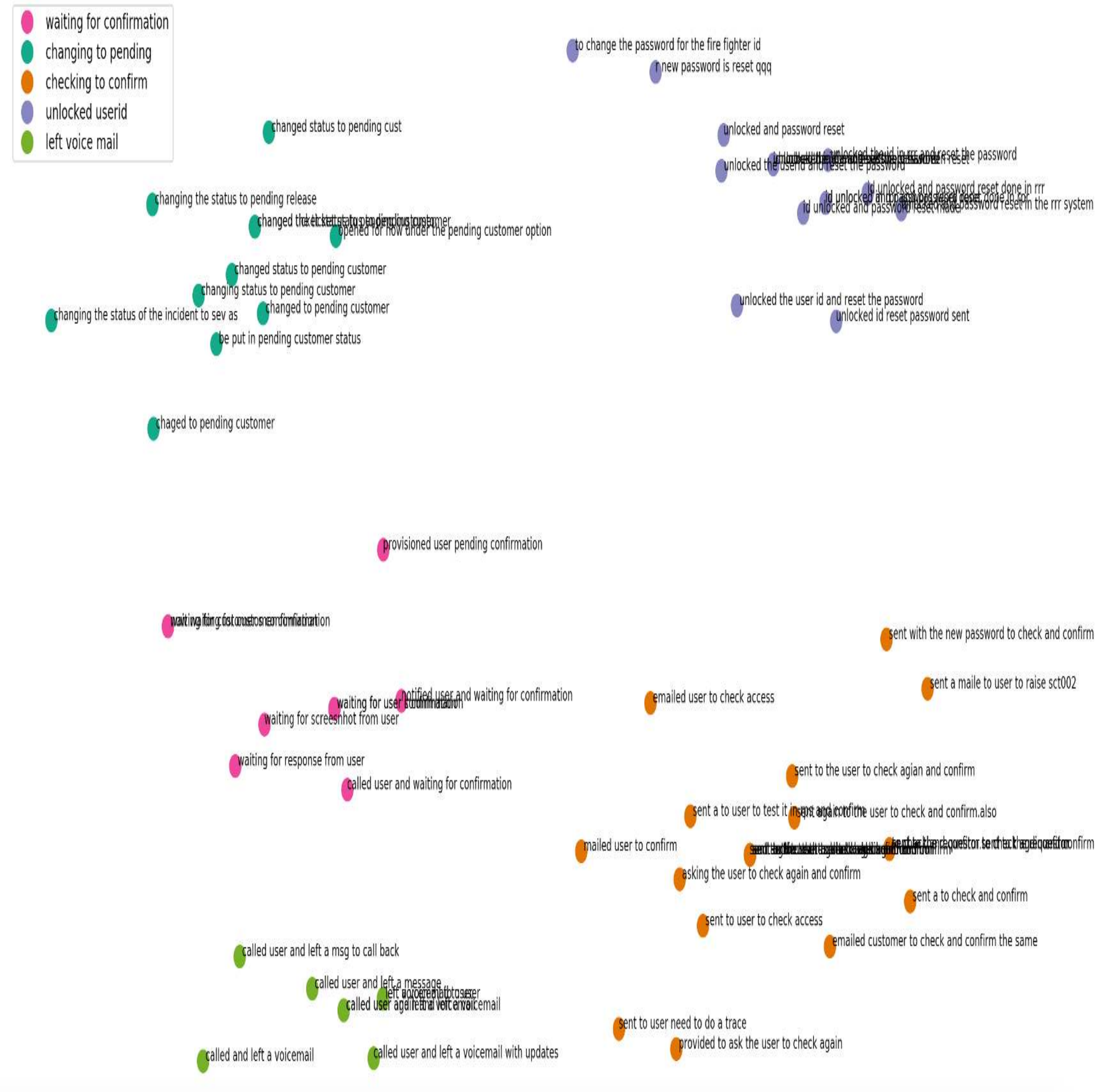


Figure 7.3: Visualizing clusters identified by the approach

Figure 7.4: Different clusters containing semantically similar phrases

There are some limitations that have been observed. Affinity propagation generates multiple clusters for phrases that semantically mean the same or are similar. This is because the word mover distance function is unable to capture the semantic similarity between the phrases. An example of this limitation is illustrated in Figure 7.4, where the clusters representing ‘awaiting customer confirmation’ and ‘awaiting reply’ semantically mean the same but are assigned to different clusters (as indicated by the colors). These clusters could be merged. However, these are limitations of existing semantic text matching techniques. The word embeddings currently used do not recognize ‘confirmation’ and ‘reply’ to be similar (the Euclidean distance between the two words is high). Hence, these phrases are considered different by the clustering algorithm. Generating word embeddings [147], on a large corpus of IT incident management textual data would help in using relevant training data and lead to improved identification of similar phrases.

7.5.4 Threats to Validity

Threats to *external validity* concerns the generalization of the results from the study. This threat has been addressed by evaluating it on textual data of 4 application domains, with over 300 users logging comments on over 2000 process instances. While insights can be drawn from our study, these results cannot be generalized in all business processes. However, the results serve as the basis of using textual data to discern relevant process context. Threats to *internal validity* arise when there are errors or biases. In this study, standard implementations of distance functions and cluster analysis, have been used. The clustering and filtering approach required some configuration parameters such as the minimum and maximum size of the clusters. These should not impact the applicability of the approach. The choice of measurements is considered as a threat to *construct validity*. Appropriate measures such as precision and recall were not used on textual data in process logs due to non-availability of labeled data. However, the metrics were evaluated on a multi-labeled benchmark data set to compare various methods of grouping textual information, used in the study.

7.6 Chapter Summary

In this chapter, a novel approach of leveraging textual logs captured during a process execution is proposed, for identifying useful and relevant situations or context. Using unstructured information extraction methods, an approach consisting of clustering process instances or tasks into unified groups, correlating them with process outcome and identifying a subset of salient groups, is used. The approach presented, is quite general, and can be applied to different application domains. PAIS store comments and textual logs from resources, for the purpose of knowledge sharing. Using this information to mine situations that lead to varying performance outcomes would be extremely valuable.

Chapter 8

Conclusion

8.1 Conclusion

Effective and optimal task allocation is one of the critical necessities for provisioning a business process. In processes involving human resources, task allocation is critical and challenging as the human performers have varying efficiencies that change with external factors such as the context (or situations). Existing research on process mining has analyzed task allocation from different perspectives such as time perspective, case perspective and organizational perspective, but each of them in isolation. To date, there has been limited work on considering all three perspectives together when allocating resources. Another important aspect that plays a crucial role is the context. Various external factors, in addition to case and the resources, influence the task allocation decisions. In real-life team leads and managers account for all three perspectives based on past experience and knowledge. To evaluate the use of data-driven approach to assist and enable better decision making, the thesis addressed the following research questions:

1. How does resource efficiency vary with case, resource behavior, and impact task allocation?

2. How do we support task allocation based on process context, for pull based dispatching scenarios?

3. How do we learn task allocation rules based on process context, for push based dispatching scenarios?

4. How do we identify useful contextual information from process data?

To address the first research question (Chapter 4), an approach was presented by considering a business process supporting resolution of problem tickets or issues.

Data from three teams involved in supporting problems that occur in operating systems, was collected for a period of three weeks. In much of the earlier work, the analysis of such a process involved considering resource efficiencies by evaluating the case attributes (customer, complexity) [8], [23]. In this dissertation, variance in service time was analyzed based on case attributes (complexity, priority) and resource behavior (experience). Statistically significant variance in mean service time was observed. These distinct service times were used as input to a simulation model representing the service system executing the business process. The experiments indicated that the performance outcome of the process was impacted when resource behavior was taken into account. The evaluation showed the need for considering resource behavior and case attributes when analyzing and making task allocation decisions. Further, the need for a data-driven approach to analyze and identify variances in resource efficiencies, was highlighted. The experiments were limited to a single study based on the data from one organization. Impact of resource behavior on service time has been validated in the past. However, there has been limited work on considering variances in resource efficiency and behavior to arrive at task allocation decisions. The key contribution of this work is an approach to verify the influence of process attributes and context on resource efficiency.

The second question is addressed by modeling contextual dimensions and building a context-aware recommender system (Chapter 5). First, resource behaviors were modeled as resource context. Task context was defined based on context attributes defined in [16]. The resource, task, context and performance outcome was modeled as user, item, context and rating of a context-aware recommender system respectively. This approach enabled considering all perspectives of organizational, case and timing for allocation decisions. The approach was evaluated on two real-life process event logs by extracting relevant resource context, task context and performance. The rating or performance predictions with and without context were evaluated. Results indicated reduction in the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) when context was used. The absolute errors were statistically significant as verified by Mann Whitney U test. The observations indicated that, certain contextual dimensions improve the prediction accuracy more than others dimensions. The influence of a contextual dimension varied for business processes. The evaluation was limited to generic contextual dimensions as there were no domain experts to model the domain context. Two key contributions of this work are: i) defining a context model using resource behavior indicators computed from event logs, and ii) building a recommender system that uses context in addition to task and resource attributes to improve task allocation decision.

The third research question is applicable in scenarios where there is a need to discern task allocation patterns based on contextual information (Chapter 6). This method is applicable in push based dispatching, where a dispatcher makes the allocation decision for all resources. In the study presented, context (includes resource and process context) and the case attributes were considered as independent variables. The performance outcome was considered as the dependent variable. A decision tree was trained using the context, case information and process outcome, extracted from event logs. Task allocation rules were inferred from the decision tree. For partially executed process instances, a k-nearest neighbor algorithm was used where, the case data and the process outcome were considered as independent variables and the contextual dimensions were predicted. Evaluation was done on a synthetic data set to identify the contextual variables impacting the performance outcomes. The evaluation on a real-life event log for predicting the outcome resulted in f1-score of 67% and an accuracy of 79% on the test data set. The predictors included contextual dimensions such as experience, preference, and utilitzation. The contribution of this work is a machine learning based method to derive resource allocation policies that take into consideration context and its influence on performance outcome.

The final research question was tackled by using natural language processing on unstructured logs that are recorded by resources when working on tasks (Chapter 7). There are several external contextual factors that occur when performing a task. Resources capture details of task execution in the form of text messages. In this dissertation, the messages were broken down into smaller phrases to make them atomic. These phrases were clustered using different clustering techniques [99], [101]. Distance measures such as word mover distance [148], and other measures using word vectors [147], were evaluated. The approach was first evaluated on a benchmark data set. Affinity propagation and word mover distance resulted in a macro-average f1-score of 58%. Experiments were carried out on a real-life process log and clusters were filtered to identify groups, which have an impact on process performance. The contribution of this work is a method to explore and discover contextual information that impacts performance outcome, from unstructured textual data available in process logs

8.2 Limitations and Future Work

In this section the limitations and extensions of current work that allow for future research are described.

8.2.1 Limitations

In addition to the limitations discussed in section 3.5, this section describes other challenges in the studies that were carried out. In Chapter 4, a limited set of process attributes and contextual dimensions were considered. There were no task specific context or domain attributes. The logs did not capture domain specific information such as types of operating system issues or time of the day when the issues were resolved. Hence, the study was limited to available amount of information. In addition the data collected from the three teams, were addressing the same domain of servicing operating systems.

Chapter 5 too, used a limited set of process attributes and contextual dimensions. Lack of domain information limited the ability to identify and use other contextual dimensions. A common challenge with context-aware recommender systems is that, addition of more dimensions causes sparseness of the data. In this dissertation, a limited set of resource behaviors were used. Addition of resource behaviors may not necessarily improve accuracy or quality of recommendations.

Chapter 6, evaluates the use of decision tree and k-nearest neighbor methods to predict the performance outcome of a process or task. These methods have been used because they are intuitive and decision tree can be used to produce a set of interpretable rules. However, they have lower accuracy as compared to other supervised learning methods. The evaluation on real-life event log, here too did not contain domain specific context or attributes.

Finally, In Chapter 7, an unsupervised method of extracting context has been explored. This approach has been evaluated on a single data set and hence the approach needs to be further analyzed on other process logs containing textual information.

8.2.2 Future work

In Chapter 4, data from three team was used from a single domain of IT services. A similar study considering domain attributes and additional contextual dimensions extracted from an event log in a different domain, containing richer information, would be useful to explore.

In Chapter 5, a single performance outcome (time), was evaluated. It would be useful to further extend this work and evaluate context recommendation for a combination of outcomes (time, cost or quality). K-NN based recommender has been used. Other model based recommender systems can be evaluated and compared for higher prediction accuracy. Further, it would be worthwhile to evaluate the global outcome of all process instances such as percentage of tasks that met the service level, total cost of task allocation, as the recommender system provides resource specific recommendations based on a local optimal rating for each resource.

The work in Chapter 6, can be extended to evaluate other machine learning methods and analyze the prediction accuracies. Another challenge commonly faced when using context is the unavailability of contextual information in event logs. It would be a valuable extension to define a specification of a process event log that enables storing and extraction of generic and domain specific contextual dimensions. An extension to the event log specification and implementation support in process aware systems to capture context, would lead to context-aware process analysis.

Finally, In Chapter 7, it would be useful to build a supervised model by creating a labeled data set containing contextual dimensions extracted from textual logs. This would require domain experts, but lack of such a data set makes it an imperative need. Building a data set would help evaluate suitability of context extraction methods from rich source of information maintained by resources when working on tasks. Filtering approaches, in addition to cluster size and performance outcomes, that help identify sets of relevant contextual clusters, will need to be evaluated.

In conclusion, this thesis compiles a set of methods that are data-driven and context-aware to help managers and resources allocate suitable tasks and help improve the overall process outcome. First, the variance in resource efficiency based on context was demonstrated (Chapter 4). A context model for resource allocation was built and used as input to a context-aware recommender system to predict resource efficiencies for a case, resource and context (Chapter 5). Next context, case and resource attributes were used to learn task allocation rules (Chapter 6). Finally, textual information available in process logs was used to semi-automatically mine context. The contextual factors mined can be used to improve the process or define process outcomes (Chapter 7). The results of this dissertation can be used as input to improve task allocation and process outcomes.

Bibliography

[1] W. M. P. van der Aalst and A. J. M. M. Weijters, "Process mining: A research agenda," *Comput. Ind.*, vol. 53, no. 3, pp. 231–244, Apr. 2004, ISSN: 0166-3615 (pages 1, 2, 8).

[2] M. Weske, *Business Process Management - Concepts, Languages, Architectures, 2nd Edition*. Springer, 2012, ISBN: 978-3-642-28615-5 (pages 1, 6).

[3] W. M. P. van der Aalst, *Process Mining: Data Science in Action*, 2nd ed. Heidelberg: Springer, 2016, ISBN: 978-3-662-49850-7 (pages 1, 2, 12, 13).

[4] A. J. M. M. Weijters and W. M. P. van der Aalst, "Rediscovering workflow models from event-based data using little thumb," *Integrated Computer-Aided Engineering*, vol. 10, no. 2, pp. 151–162, 2003 (page 1).

[5] M. Song and W. M. P. van der Aalst, "Towards comprehensive support for organizational mining," *Decis. Support Syst.*, vol. 46, no. 1, pp. 300–317, Dec. 2008, ISSN: 0167-9236 (pages 1, 2, 12–14, 29).

[6] E. Ramezani, D. Fahland, and W. M. P. van der Aalst, "Where did I misbehave? diagnostic information in compliance checking," in *Business Process Management - 10th International Conference, BPM 2012, Tallinn, Estonia, September 3-6, 2012. Proceedings*, 2012, pp. 262–278 (page 2).

[7] W. M. P. van der Aalst, J. Nakatumba, A. Rozinat, and N. Russell, "Business process simulation," in *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, J. v. Brocke and M. Rosemann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 313–338, ISBN: 978-3-642-00416-2 (pages 2, 16).

[8] G. Havur, C. Cabanillas, J. Mendling, and A. Polleres, "Automated resource allocation in business processes with answer set programming," in *Business Process Management Workshops - BPM 2015, 13th International Workshops, Innsbruck, Austria, August 31 - September 3, 2015, Revised Papers*, 2015, pp. 191–203 (pages 2, 3, 52, 127).

[9] Z. Huang, X. Lu, and H. Duan, "A task operation model for resource allocation optimization in business process management," in *IEEE Transactions on Systems, Man, and Cybernetics*, ser. 5, vol. Part A 42, 2012, pp. 1256–1270 (pages 2, 15, 29, 92).

[10] A. Senderovich, A. Rogge-Solti, A. Gal, J. Mendling, A. Mandelbaum, S. Kadish, and C. A. Bunnell, "Data-driven performance analysis of scheduled processes," in *Business Process Management - 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 - September 3, 2015, Proceedings*, 2015, pp. 35–52 (pages 2, 3).

[11] A. Rozinat and W. M. P. van der Aalst, "Decision mining in prom," in *Business Process Management, 4th International Conference, BPM 2006, Vienna, Austria, September 5-7, 2006, Proceedings*, 2006, pp. 420–425 (page 2).

[12] J. Nakatumba and W. M. P. van der Aalst, "Analyzing resource behavior using process mining," in *Business Process Management Workshops, BPM 2009 International Workshops, Ulm, Germany, September 7, 2009. Revised Papers*, 2009, pp. 69–80 (pages 2, 16, 18, 29, 52, 79).

[13] A. Pika, M. T. Wynn, C. J. Fidge, A. H. M. ter Hofstede, M. Leyer, and W. M. P. van der Aalst, "An extensible framework for analysing resource behaviour using event logs," in *CAiSE 2014, Greece, June 16-20, 2014. Proceedings*, 2014, pp. 564–579 (pages 2, 18, 29, 101).

[14] G. Adomavicius and A. Tuzhilin, "Recommender systems handbook," in, F. Ricci, L. Rokach, B. Shapira, and B. P. Kantor, Eds. Boston, MA: Springer US, 2011, ch. Context-Aware Recommender Systems, pp. 217–253, ISBN: 978-0-387-85820-3 (pages 3, 73, 76, 77, 110).

[15] J. Kiseleva, M. J. I. Müller, L. Bernardi, C. Davis, I. Kovacek, M. S. Einarsen, J. Kamps, A. Tuzhilin, and D. Hiemstra, "Where to go on your next trip?: Optimizing travel destinations based on user preferences," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, 2015, pp. 1097–1100 (page 3).

[16] O. Saidani, C. Rolland, and S. Nurcan, "Towards a generic context model for BPM," in *48th Hawaii International Conference on System Sciences, HICSS 2015, Kauai, Hawaii, USA, January 5-8, 2015*, 2015, pp. 4120–4129 (pages 3, 26, 27, 79, 110, 127).

[17] M. Rosemann and J. Recker, "Context-aware process design exploring the extrinsic drivers for process flexibility," in *Proceedings of the CAISE*06 Workshop on Business Process Modelling, Development, and Support BPMDS '06, Luxemburg, June 5-9, 2006*, 2006 (pages 3, 25).

[18] M. Bazire and P. Brézillon, "Understanding context before using it," in *Modeling and Using Context: 5th International and Interdisciplinary Conference CONTEXT 2005, Paris, France, July 5-8, 2005. Proceedings*, A. Dey, B. Kokinov, D. Leake, and R. Turner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 29–40, ISBN: 978-3-540-31890-3 (pages 3, 24).

[19] A. K. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 4–7, 2001 (pages 3, 24).

[20] J. Ghattas, P. Soffer, and M. Peleg, "A formal model for process context learning," in *Business Process Management Workshops, BPM 2009 International Workshops, Ulm, Germany, September 7, 2009. Revised Papers*, 2009, pp. 140–157 (pages 3, 27, 29).

[21] J. Ghattas, P. Soffer, and M. Peleg, "Improving business process decision making based on past experience," *Decision Support Systems*, vol. 59, pp. 93–107, 2014 (pages 3, 28, 29).

[22] J. Ghattas, M. Peleg, P. Soffer, and Y. Denekamp, "Learning the context of a clinical process," in *Business Process Management Workshops, BPM 2009 International Workshops, Ulm, Germany, September 7, 2009. Revised Papers*, 2009, pp. 545–556 (pages 3, 28, 29).

[23] Y. Diao and A. Heching, "Staffing optimization in complex service delivery systems," in *7th International Conference on Network and Service Management, CNSM 2011, Paris, France, October 24-28, 2011*, 2011, pp. 1–9 (pages 4, 21, 52, 57, 58, 66, 127).

[24] R. Sindhgatta, G. B. Dasgupta, and A. Ghose, "Analysis of operational data for expertise aware staffing," in *12th Int. Conf., BPM 2014. Proceedings*, 2014, pp. 317–332 (page 5).

[25] R. Sindhgatta, A. K. Ghose, and H. K. Dam, "Context-aware recommendation of task allocations in service systems," in *Service-Oriented Computing - 14th International Conference, ICSOC 2016, Banff, AB, Canada, October 10-13, 2016, Proceedings*, 2016, pp. 402–416 (page 5).

[26] R. Sindhgatta, A. Ghose, and H. K. Dam, "Context-aware analysis of past process executions to aid resource allocation decisions," in *CAiSE'16 Forum at the 28th International Conference on Advanced Information Systems En-*

gineering (CAiSE), Ljubljana, Slovenia, 13-17 June 2016, 2016, (to appear) (page 5).

[27] R. Sindhgatta, A. Ghose, and H. K. Dam, "Leveraging unstructured data to analyze implicit process context," in *Business Process Management Forum - BPM Forum 2018, Sydney, NSW, Australia, September 9-14, 2018, Proceedings*, 2018, pp. 143–158 (page 5).

[28] W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, "Business process management: A survey," in *Business Process Management, International Conference, BPM 2003, Eindhoven, The Netherlands, June 26-27, 2003, Proceedings*, 2003, pp. 1–12 (page 6).

[29] T. H. Davenport, *Process Innovation: Reengineering Work Through Information Technology*. Boston, MA, USA: Harvard Business School Press, 1993, ISBN: 0-87584-366-2 (page 6).

[30] W. M. P. van der Aalst and C. Stahl, *Modeling Business Processes - A Petri Net-Oriented Approach*, ser. Cooperative Information Systems series. MIT Press, 2011, ISBN: 978-0-262-01538-7 (page 7).

[31] A. Scheer, O. Thomas, and O. Adam, "Process modeling using event-driven process chains," in *Process-Aware Information Systems: Bridging People and Software Through Process Technology*, 2005 (page 7).

[32] J. Rumbaugh, I. Jacobson, and G. Booch, *Unified Modeling Language Reference Manual, The (2Nd Edition)*. Pearson Higher Education, 2004, ISBN: 0321245628 (page 7).

[33] M. zur Muehlen, "Organizational management in workflow applications - issues and perspectives," *Information Technology and Management*, vol. 5, no. 3-4, pp. 271–291, 2004 (pages 8, 78).

[34] C. Cabanillas, "Enhancing management of resource-aware business process," in *PhD thesis, Universidad De Sevilla*, 2012 (page 8).

[35] N. Russell, W. M. P. van der Aalst, A. H. M. ter Hofstede, and D. Edmond, "Workflow resource patterns: Identification, representation and tool support.," in *CAiSE*, ser. LNCS, vol. 3520, Springer, May 24, 2005, pp. 216–232 (pages 9, 44, 45).

[36] C. Cabanillas, M. Resinas, and A. R. Cortés, "RAL: A high-level user-oriented resource assignment language for business processes," in *Business Process Management Workshops - BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I*, 2011, pp. 50–61 (page 11).

[37] C. Wolter and A. Schaad, "Modeling of task-based authorization constraints in bpmn," in *Proceedings of the 5th International Conference on Business Process Management*, ser. BPM'07, Brisbane, Australia: Springer-Verlag, 2007, pp. 64–79, ISBN: 3-540-75182-3, 978-3-540-75182-3 (page 11).

[38] L. T. Ly, S. Rinderle, P. Dadam, and M. Reichert, "Mining staff assignment rules from event-based data," in *Business Process Management Workshops, BPM 2005 International Workshops, BPI, BPD, ENEI, BPRM, WS-COBPM, BPS, Nancy, France, September 5, 2005, Revised Selected Papers*, 2005, pp. 177–190 (pages 14, 29).

[39] S. Schönig, C. Cabanillas, S. Jablonski, and J. Mendling, "Mining the organisational perspective in agile business processes," in *Enterprise, Business-Process and Information Systems Modeling - 16th International Conference, BPMDS 2015, 20th International Conference, EMMSAD 2015, Held at CAiSE 2015, Stockholm, Sweden, June 8-9, 2015, Proceedings*, 2015, pp. 37–52 (pages 14, 29).

[40] F. M. Maggi, R. P. J. C. Bose, and W. M. P. van der Aalst, "Efficient discovery of understandable declarative process models from event logs," in *Advanced Information Systems Engineering - 24th International Conference, CAiSE 2012, Gdansk, Poland, June 25-29, 2012. Proceedings*, 2012, pp. 270–285 (page 14).

[41] M. Kuhlmann, D. Shohat, and G. Schimpf, "Role mining - revealing business roles for security administration using data mining technology," in *Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies*, ser. SACMAT '03, Como, Italy: ACM, 2003, pp. 179–186, ISBN: 1-58113-681-1 (pages 14, 29).

[42] A. Baumgrass, "Deriving current state RBAC models from event logs," in *Sixth International Conference on Availability, Reliability and Security, ARES 2011, Vienna, Austria, August 22-26, 2011*, 2011, pp. 667–672 (pages 14, 29).

[43] A. Burattin, A. Sperduti, and M. Veluscek, "Business models enhancement through discovery of roles," in *CIDM 2013, Singapore, 16-19 April, 2013*, 2013, pp. 103–110 (pages 14, 29).

[44] J. Vaidya, V. Atluri, and Q. Guo, "The role mining problem: Finding a minimal descriptive set of roles," in *Proc. of the 12th ACM SACMAT*, Sophia Antipolis, France, 2007, pp. 175–184, ISBN: 978-1-59593-745-2 (page 14).

[45] M. Frank, J. M. Buhman, and D. Basin, "Role mining with probabilistic models," *ACM Trans. Inf. Syst. Secur.*, vol. 15, no. 4, 15:1–15:28, Apr. 2013, ISSN: 1094-9224 (page 14).

[46] A. Kumar, W. M. P. Van Der Aalst, and E. M. W. Verbeek, "Dynamic work distribution in workflow management systems: How to balance quality and performance," *J. Manage. Inf. Syst.*, vol. 18, no. 3, pp. 157–193, Jan. 2002, ISSN: 0742-1222 (pages 15, 29).

[47] Z. Huang, W. M. P. van der Aalst, X. Lu, and H. Duan, "Reinforcement learning based resource allocation in business process management," *Data Knowl. Eng.*, vol. 70, no. 1, pp. 127–145, 2011 (pages 15, 29, 52).

[48] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, 1998 (page 15).

[49] C. Cabanillas, J. M. García, M. Resinas, D. Ruiz, J. Mendling, and A. R. Cortés, "Priority-based human resource allocation in business processes," in *Service-Oriented Computing - 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5, 2013, Proceedings*, 2013, pp. 374–388 (pages 15, 29).

[50] J. M. GarcíA, M. Junghans, D. Ruiz, S. Agarwal, and A. Ruiz-CortéS, "Integrating semantic web services ranking mechanisms using a common preference model," *Know.-Based Syst.*, vol. 49, pp. 22–36, Sep. 2013, ISSN: 0950-7051 (page 15).

[51] M. Mohagheghian, R. Sindhgatta, and A. K. Ghose, "An extended agent based model for service delivery optimization," in *PRIMA 2014: Principles and Practice of Multi-Agent Systems - 17th International Conference, Gold Coast, QLD, Australia, December 1-5, 2014. Proceedings*, 2014, pp. 270–285 (page 15).

[52] H. S. Gupta and B. Sengupta, "Scheduling service tickets in shared delivery," in *ICSOC*, 2012, pp. 79–95 (pages 15, 29, 92).

[53] C. Oliveira, R. Lima, T. Andre, and H. Reijers, "Modeling and analyzing resource-constrained business processes," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, Oct. 2009, pp. 2824–2830 (pages 15, 29, 92).

[54] C. L. J. Xu and X. Zhao, "Resource planning for massive number of process instances," in *On the Move to Meaningful Internet Systems,LNCS, Berlin, Germany*, vol. 5870, 2009, pp. 219–236 (pages 15, 29, 92).

[55] J. Xu, C. Liu, X. Zhao, and S. Yongchareon, "Business process scheduling with resource availability constraints," in *On the Move to Meaningful Internet Systems: OTM 2010: Confederated International Conferences: CoopIS, IS, DOA and ODBASE, Hersonissos, Crete, Greece, October 25-29, 2010, Proceedings, Part I*, R. Meersman, T. Dillon, and P. Herrero, Eds. Berlin,

Heidelberg: Springer Berlin Heidelberg, 2010, pp. 419–427, ISBN: 978-3-642-16934-2 (pages 15, 29, 92).

[56] J. Pflug and S. Rinderle-Ma, "Dynamic instance queuing in process-aware information systems," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13, Coimbra, Portugal, March 18-22, 2013*, 2013, pp. 1426–1433 (pages 15, 29, 92).

[57] Z. Huang, X. Lu, and H. Duan, "Resource behavior measure and application in business process management," *Expert Syst. Appl.*, vol. 39, no. 7, pp. 6458–6468, 2012 (pages 16, 29, 79, 80, 101).

[58] S. Kabicher-Fuchs and S. Rinderle-Ma, "Work experience in PAIS - concepts, measurements and potentials," in *Advanced Information Systems Engineering - 24th International Conference, CAiSE 2012, Gdansk, Poland, June 25-29, 2012. Proceedings*, 2012, pp. 678–694 (pages 17, 29).

[59] S. Kabicher-Fuchs, J. Mangler, and S. Rinderle-Ma, "Experience breeding in process-aware information systems," in *Advanced Information Systems Engineering - 25th International Conference, CAiSE 2013, Valencia, Spain, June 17-21, 2013. Proceedings*, 2013, pp. 594–609 (pages 17, 80).

[60] A. Kumar, R. M. Dijkman, and M. Song, "Optimal resource assignment in workflows for maximizing cooperation," in *Business Process Management - 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings*, 2013, pp. 235–250 (pages 17, 29, 30, 52, 80).

[61] A. Pika, M. Leyer, M. T. Wynn, C. J. Fidge, A. H. M. ter Hofstede, and W. M. P. van der Aalst, "Mining resource profiles from event logs," *ACM Trans. Management Inf. Syst.*, vol. 8, no. 1, 1:1–1:30, 2017 (pages 18, 29, 52).

[62] B. F. van Dongen, R. A. Crooy, and W. M. P. van der Aalst, "Cycle time prediction: When will this case finally be finished?" In *On the Move to Meaningful Internet Systems: OTM 2008, OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 9-14, 2008, Proceedings, Part I*, 2008, pp. 319–336 (pages 19, 29).

[63] W. M. P. van der Aalst, M. H. Schonenberg, and M. Song, "Time prediction based on process mining," *Inf. Syst.*, vol. 36, no. 2, pp. 450–475, 2011 (page 19).

[64] S. Suriadi, C. Ouyang, W. M. P. van der Aalst, and A. H. M. ter Hofstede, "Root cause analysis with enriched process logs," in *Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, 2012, pp. 174–186 (pages 20, 29).

[65] H. Schonenberg, B. Weber, B. F. van Dongen, and W. M. P. van der Aalst, "Supporting flexible processes through recommendations based on history," in *Business Process Management, 6th International Conference, BPM 2008, Milan, Italy, September 2-4, 2008. Proceedings*, 2008, pp. 51–66 (pages 20, 29).

[66] G. T. Lakshmanan, D. Shamsi, Y. N. Doganata, M. Unuvar, and R. Khalaf, "A markov prediction model for data-driven semi-structured business processes," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 97–126, Jan. 2015, ISSN: 0219-3116 (page 20).

[67] N. Tax, I. Verenich, M. L. Rosa, and M. Dumas, "Predictive business process monitoring with LSTM neural networks," in *Advanced Information Systems Engineering - 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings*, 2017, pp. 477–492 (page 20).

[68] F. M. Maggi, C. D. Francescomarino, M. Dumas, and C. Ghidini, "Predictive monitoring of business processes," in *Advanced Information Systems Engineering - 26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16-20, 2014. Proceedings*, 2014, pp. 457–472 (page 20).

[69] I. Teinemaa, M. Dumas, F. M. Maggi, and C. D. Francescomarino, "Predictive business process monitoring with structured and unstructured data," in *Business Process Management - 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings*, 2016, pp. 401–417 (pages 20, 31, 109).

[70] M. de Leoni, W. M. P. van der Aalst, and M. Dees, "A general framework for correlating business process characteristics," in *Business Process Management - 12th International Conference, BPM 2014, Haifa, Israel, September 7-11, 2014. Proceedings*, 2014, pp. 250–266 (page 21).

[71] P. P. Maglio, S. L. Vargo, N. Caswell, and J. Spohrer, "The service system is the basic abstraction of service science," *Inf. Syst. E-Business Management*, vol. 7, no. 4, pp. 395–406, 2009 (page 21).

[72] L. Ramaswamy and G. Banavar, "A formal model of service delivery," in *2008 IEEE International Conference on Services Computing (SCC 2008), 8-11 July 2008, Honolulu, Hawaii, USA*, 2008, pp. 517–520 (page 21).

[73] D. Banerjee, G. B. Dasgupta, and N. Desai, “Simulation-based evaluation of dispatching policies in service systems,” in *Winter Simulation Conference 2011, WSC'11, Phoenix, AZ, USA, December 11-14, 2011*, 2011, pp. 779–791 (pages 22, 57).

[74] Q. Shao, Y. Chen, S. Tao, X. Yan, and N. Anerousis, “Efficient ticket routing by resolution sequence mining,” in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08, Las Vegas, Nevada, USA: ACM, 2008, pp. 605–613, ISBN: 978-1-60558-193-4 (pages 22, 29, 32).

[75] S. Agarwal, R. Sindhgatta, and B. Sengupta, “Smartdispatch: Enabling efficient ticket dispatch in an it service environment,” in *KDD*, 2012, pp. 1393–1401 (pages 22, 29, 32, 109).

[76] P. Sun, S. Tao, X. Yan, N. Anerousis, and Y. Chen, “Content-aware resolution sequence mining for ticket routing,” in *Business Process Management - 8th International Conference, BPM 2010, Hoboken, NJ, USA, September 13-16, 2010. Proceedings*, 2010, pp. 243–259 (page 22).

[77] S. Agarwal, R. Sindhgatta, and G. B. Dasgupta, “Does one-size-fit-all suffice for service delivery clients?” In *Service-Oriented Computing - 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5, 2013, Proceedings*, 2013, pp. 177–191 (pages 22, 23, 29).

[78] G. B. Dasgupta, R. Sindhgatta, and S. Agarwal, “Behavioral analysis of service delivery models,” in *Service-Oriented Computing - 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5, 2013, Proceedings*, 2013, pp. 652–666 (pages 23, 29).

[79] B. Sengupta, A. Jain, K. Bhattacharya, H. L. Truong, and S. Dustdar, “Collective problem solving using social compute units,” *Int. J. Cooperative Inf. Syst.*, vol. 22, no. 4, 2013 (page 23).

[80] S. Dustdar and K. Bhattacharya, “The social compute unit,” *IEEE Internet Computing*, vol. 15, no. 3, pp. 64–69, May 2011, ISSN: 1089-7801 (page 23).

[81] J. Kiseleva, “Context mining and integration into predictive web analytics,” in *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*, 2013, pp. 383–388 (page 24).

[82] P. Dourish, “What we talk about when we talk about context,” *Personal Ubiquitous Comput.*, vol. 8, no. 1, pp. 19–30, Feb. 2004, ISSN: 1617-4909 (pages 25, 73, 110).

[83] O. Saidani and S. Nurcan, "Context-awareness for adequate business process modelling," in *Proceedings of the Third IEEE International Conference on Research Challenges in Information Science, RCIS 2009, Fès, Morocco, 22-24 April 2009*, 2009, pp. 177–186 (pages 25, 96, 101).

[84] S. Najar, O. Saidani, M. Kirsch-Pinheiro, C. Souveyet, and S. Nurcan, "Semantic representation of context models: A framework for analyzing and understanding," in *Proceedings of the 1st Workshop on Context, Information and Ontologies*, ser. CIAO '09, Heraklion, Greece: ACM, 2009, 6:1–6:10, ISBN: 978-1-60558-528-4 (page 25).

[85] K. Bessai and S. Nurcan, "Actor-driven approach for business process. how to take into account the work environment?" In *Enterprise, Business-Process and Information Systems Modeling, 10th International Workshop, BPMDS 2009, and 14th International Conference, EMMSAD 2009, held at CAiSE 2009, Amsterdam, The Netherlands, June 8-9, 2009. Proceedings*, 2009, pp. 187–196 (page 27).

[86] A. Ghose, G. Koliadis, and A. Chueng, "Process discovery from model and text artefacts," in *2007 IEEE International Conference on Services Computing - Workshops (SCW 2007), 9-13 July 2007, Salt Lake City, Utah, USA*, 2007, pp. 167–174 (page 30).

[87] F. Friedrich, J. Mendling, and F. Puhlmann, "Process model generation from natural language text," in *Advanced Information Systems Engineering - 23rd International Conference, CAiSE 2011, London, UK, June 20-24, 2011. Proceedings*, 2011, pp. 482–496 (page 31).

[88] A. Sinha and A. M. Paradkar, "Use cases to process specifications in business process modeling notation," in *IEEE International Conference on Web Services, ICWS 2010, Miami, Florida, USA, July 5-10, 2010*, 2010, pp. 473–480 (page 31).

[89] M. M. Botezatu, J. Bogojeska, I. Giurgiu, H. Völzer, and D. Wiesmann, "Multi-view incident ticket clustering for optimal ticket dispatching," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, 2015, pp. 1711–1720 (page 32).

[90] W. Zhou, L. Tang, C. Zeng, T. Li, L. Shwartz, and G. Ya. Grabarnik, "Resolution recommendation for event tickets in service management," *IEEE Trans. on Netw. and Serv. Manag.*, vol. 13, no. 4, pp. 954–967, Dec. 2016, ISSN: 1932-4537 (page 32).

[91] R. Potharaju, N. Jain, and C. Nita-Rotaru, "Juggling the jigsaw: Towards automated problem inference from network trouble tickets," in *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2013, Lombard, IL, USA, April 2-5, 2013*, 2013, pp. 127–141 (pages 32, 109, 114).

[92] V. Aggarwal, S. Agarwal, G. B. Dasgupta, G. Sridhara, and V. E, "React: A system for recommending actions for rapid resolution of IT service incidents," in *IEEE International Conference on Services Computing, SCC 2016, San Francisco, CA, USA, June 27 - July 2, 2016*, 2016, pp. 1–8 (pages 32, 109).

[93] S. Mani, K. Sankaranarayanan, V. S. Sinha, and P. T. Devanbu, "Panning requirement nuggets in stream of software maintenance tickets," in *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, (FSE-22), Hong Kong, China, November 16 - 22, 2014*, 2014, pp. 678–688 (page 32).

[94] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, ser. Adaptive computation and machine learning. MIT Press, 2012, ISBN: 978-0-262-01825-8 (pages 32, 33).

[95] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth, 1984, ISBN: 0-534-98053-8 (page 33).

[96] G. V. Kass, "An exploratory technique for investigating large quantities of categorical data," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 29, no. 2, pp. 119–127, 1980, ISSN: 00359254, 14679876 (page 33).

[97] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993, ISBN: 1-55860-238-0 (page 34).

[98] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607–616, 1996 (pages 34, 100).

[99] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, Berkeley, Calif.: University of California Press, 1967, pp. 281–297 (pages 35, 116, 128).

[100] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008, ISBN: 978-0-521-86571-5 (pages 35, 41).

[101] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007 (pages 36, 116, 128).

[102] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–58, Mar. 1997, ISSN: 0001-0782 (page 37).

[103] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proceedings of the 1st ACM Conference on Electronic Commerce*, ser. EC '99, Denver, Colorado, USA: ACM, 1999, pp. 158–166, ISBN: 1-58113-176-3 (page 37).

[104] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proceedings of the 2Nd ACM Conference on Electronic Commerce*, ser. EC '00, Minneapolis, Minnesota, USA: ACM, 2000, pp. 158–167, ISBN: 1-58113-272-7 (pages 37, 81).

[105] X. Ning, C. Desrosiers, and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender Systems Handbook*, 2015, pp. 37–76 (page 38).

[106] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995 (page 38).

[107] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)* Baltimore, MD, USA: Johns Hopkins University Press, 1996, ISBN: 0-8018-5414-8 (pages 38, 41).

[108] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011 (page 38).

[109] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*, 1995, pp. 1137–1145 (page 39).

[110] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999, ISBN: 020139829X (page 40).

[111] T. K. Landauer and S. T. Dumais, "Latent semantic analysis," *Scholarpedia*, vol. 3, no. 11, p. 4356, 2008 (page 41).

[112] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, no. 2-3, pp. 259–284, 1998 (page 41).

[113] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003 (pages 41, 115).

[114] L. Yao, D. Mimno, and A. McCallum, "Efficient methods for topic model inference on streaming document collections," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09, Paris, France: ACM, 2009, pp. 937–946, ISBN: 978-1-60558-495-9 (page 41).

[115] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 3rd ed. Sage Publications Ltd., 2008 (page 46).

[116] A. D. Bautista, L. Wangikar, and S. M. K. Akbar, "Process mining-driven optimization of a consumer loan approvals process - the BPIC 2012 challenge case study," in *Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, 2012, pp. 219–220 (pages 48, 74, 82, 87).

[117] C. J. Kang, Y. S. Kang, Y. S. Lee, S. Noh, H. C. Kim, W. C. Lim, J. Kim, and R. Hong, "Process mining-based understanding and analysis of volvo it's incident and problem management processes," in *Proceedings of the 3rd Business Process Intelligence Challenge co-located with 9th International Business Process Intelligence Workshop (BPI 2013), Beijing, China, August 26, 2013.*, 2013 (pages 48, 50, 106).

[118] R. P. J. C. Bose and W. M. P. van der Aalst, "Process mining applied to the BPI challenge 2012: Divide and conquer while discerning resources," in *Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, 2012, pp. 221–222 (page 48).

[119] I. Miles, N. Kastrinos, K. Flanagan, R. Bilderbeek, P. Den Hertog, W. Huntink, and M. Bouman, "Knowledge-intensive business services," *EIMS publication*, no. 15, 1995 (page 53).

[120] S. Laumer, N. Blinn, and A. Eckhardt, "Opening the black box of outsourcing knowledge intensive business processes–a longitudinal case study of outsourcing recruiting activities," in *2012 45th Hawaii International Conference on System Sciences*, Jan. 2012, pp. 3827–3836 (page 53).

[121] M. Unger, H. Leopold, and J. Mendling, "How much flexibility is good for knowledge intensive business processes: A study of the effects of informal work practices," in *48th Hawaii International Conference on System Sciences, HICSS 2015, Kauai, Hawaii, USA, January 5-8, 2015*, 2015, pp. 4990–4999 (page 53).

[122] R. S. Huckman and G. P. Pisano, "The firm specificity of individual performance: Evidence from cardiac surgery," *Management Science*, pp. 473–488, 2006 (page 53).

[123] A. Newell and P. S. Rosenbloom, "Mechanisms of skill acquisition and the law of practice," MIT Press, 1993, pp. 81–135, ISBN: 0-262-68071-8 (pages 53, 64).

[124] Y. Diao and A. Heching, "Analysis of operational data to improve performance in service delivery systems," in *8th International Conference on Network and Service Management, CNSM 2012, Las Vegas, NV, USA, October 22-26, 2012*, 2012, pp. 302–308 (pages 53, 60, 63).

[125] Y. Diao, A. Heching, D. M. Northcutt, and G. Stark, "Modeling a complex global service delivery system," in *Winter Simulation Conference*, 2011, pp. 690–702 (page 57).

[126] A. Borshchev, *The Big Book of Simulation Modeling. Multimethod Modeling with AnyLogic 6*. Kluwer, 2013, ISBN: 978-0-9895731-7-7 (page 62).

[127] R. Martí, M. Laguna, and F. Glover, "Principles of scatter search," *European Journal of Operational Research*, vol. 169, no. 2, pp. 359–372, 2006 (page 62).

[128] F. Glover and M. Laguna, *TABU search*. Kluwer, 1999, pp. I–XIX, 1–382, ISBN: 978-0-7923-9965-0 (page 62).

[129] S. Siegel and N. Castellan, *Nonparametric statistics for the behavioral sciences*, Second. McGraw–Hill, Inc., 1988 (pages 63, 74, 119).

[130] L. Baltrunas and X. Amatriain, "Towards time-dependant recommendation based on implicit feedback," in *In Workshop on context-aware recommender systems (CARS 09*, 2009 (pages 78, 83).

[131] L. Baltrunas and F. Ricci, "Context-dependent recommendations with items splitting," in *IIR 2010 - Proceedings of the First Italian Information Retrieval Workshop, Padua, Italy, January 27-28, 2010*, 2010, pp. 71–75 (pages 78, 83).

[132] Y. Zheng, B. Mobasher, and R. D. Burke, "The role of emotions in context-aware recommendation," in *Proceedings of the 3rd Workshop on Human Decision Making in Recommender Systems in conjunction with the 7th ACM Conference on Recommender Systems (RecSys 2013), Hong Kong, China, October 12, 2013.*, 2013, pp. 21–28 (pages 78, 83).

[133] B. F. van Dongen, B. Weber, D. R. Ferreira, and J. D. Weerdt, "Report: Business process intelligence challenge 2013," in *Business Process Management Workshops - BPM 2013 International Workshops, Beijing, China, August 26, 2013, Revised Papers*, 2013, pp. 79–87 (pages 82, 86, 106).

[134] Y. Zheng, B. Mobasher, and R. D. Burke, "Carskit: A java-based context-aware recommendation engine," in *IEEE International Conference on Data Mining Workshop, ICDMW 2015, Atlantic City, NJ, USA, November 14-17, 2015*, 2015, pp. 1668–1671 (page 84).

[135] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004 (page 85).

[136] A. Arcuri and L. Briand, "A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering," *Software Testing, Verification and Reliability*, vol. 24, no. 3, pp. 219–250, 2014, ISSN: 1099-1689 (page 86).

[137] A. Leontjeva, R. Conforti, C. D. Francescomarino, M. Dumas, and F. M. Maggi, "Complex symbolic sequence encodings for predictive monitoring of business processes," in *Business Process Management - 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 - September 3, 2015, Proceedings*, 2015, pp. 297–313 (page 99).

[138] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *The Computer Journal*, vol. 41, pp. 578–588, 1998 (page 99).

[139] L. I. Press, M. S. Rogers, and G. H. Shure, "An interactive technique for the analysis of multivariate data," *Behavioral Science*, vol. 14, no. 5, pp. 364–370, 1969, ISSN: 1099-1743 (page 104).

[140] J. R. Quinlan, "Generating production rules from decision trees," in *Proceedings of the 10th International Joint Conference on Artificial Intelligence. Milan, Italy, August 23-28, 1987*, 1987, pp. 304–307 (page 104).

[141] A. Field, *Discovering Statistics Using SPSS*. SAGE Publications, 2005, ISBN: 0761944524 (page 105).

[142] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 1535–1545 (page 114).

[143] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The penn treebank: Annotating predicate argument structure," in *Proceedings of the Workshop on Human Language Technology*, ser. HLT '94, Plainsboro, NJ: Association for Computational Linguistics, 1994, pp. 114–119, ISBN: 1-55860-357-3 (page 114).

[144] M. Allahyari, S. A. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text summarization techniques: A brief survey," *CoRR*, vol. abs/1707.02268, 2017 (page 115).

[145] S. Osiński, J. Stefanowski, and D. Weiss, "Lingo: Search results clustering algorithm based on singular value decomposition," in *Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM'04 Conference*, M. A. Kłopotek, S. T. Wierzchoń, and K. Trojanowski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 359–368 (page 115).

[146] T. Hofmann, "Probabilistic latent semantic indexing," in *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, 1999, pp. 50–57 (page 115).

[147] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013 (pages 116, 124, 128).

[148] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger, "From word embeddings to document distances," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015, pp. 957–966 (pages 116, 128).

[149] A. del-Río-Ortega, M. R. A. de Reyna, A. D. Toro, and A. R. Cortés, "Defining process performance indicators by using templates and patterns," in *Business Process Management - 10th International Conference, BPM 2012, Tallinn, Estonia, September 3-6, 2012. Proceedings*, 2012, pp. 223–228 (pages 117, 119).

[150] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proceedings of the 22Nd Annual International ACM SIGIR*, ser. SIGIR '99, Berkeley, California, USA, 1999, pp. 42–49 (page 119).

[151] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427–437, 2009 (page 120).