University of Alberta

Developing Industrial Workflows from Process Data

by

Sridhar Dasani

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Process Control

Department of Chemical and Materials Engineering

Fall 2013
Edmonton, Alberta

Dedicated to:

Mummy, Daddy, Budda, Chinni, Ammulu

&

Shirdi Sai

for your love and affection.

Abstract

An industrial workflow represents a sequence of tasks performed by an operator in controlling or monitoring a process. Developing industrial workflows refers to the technique of capturing the best operating practice(s) to handle a specific event encountered in process operations. In this work, workflow strategies are developed that capture expert knowledge by analyzing the event logs of how the most experienced plant operators have dealt with infrequent process operations such as a boiler start-up and shutdown procedures. We explore various data mining methods to extract valuable operational knowledge from alarm and event archives and convert them into industrial workflows. Various challenges involved in pre-processing of text-based event logs are addressed in this thesis. The applicability of these workflow strategies is demonstrated with live orchestrated workflows developed for boiler and pipeline operations. The proposed procedure guides operators to deal with out-of-spec events and critical emergencies in a safe and efficient manner.

Acknowledgements

First and foremost, I am highly indebted to my supervisor Dr. Sirish L. Shah for being a continuous source of encouragement and my inspiration during my MSc program at the University of Alberta. Dr. Shah gave me freedom to explore all avenues during my research and at the same time he made sure that I progress towards the desired results. He gave me confidence and timely support personally and professionally and I am very fortunate to work under his supervision. I always cherish the time spent and the rich learning experience gained working under such a great mentor.

I am heartily thankful to my co-supervisor Dr. Tongwen Chen for his valuable guidance during my research. I would like to personally thank him for providing an opportunity to work with his students in a stimulating team environment.

My sincere thanks to our Sponsor Company and National Research Council Canada (NRC-IRAP) for their financial support. I would like to thank Ken McDonald and Scott Buffett from NRC-IRAP for sharing their vision and experience in this project.

I am grateful to Vinay A. Bavdekar for his encouragement and patience in reviewing and commenting on the contents of this thesis. My earnest thanks to OWBE group members; Yue Cheng, Ping Duan, Kenan Kigunda, Zihao Huang, Irvinder Noor Singh, and Vengatesh Muralidharan for their help and coordination till the completion of this project.

I would like to acknowledge our end-users who participated in this project; Gordon Meyer, Colin Lunty from Suncor Energy Pipelines and Robert Pollard, Jim Hyrve from the heating plant at the University of Alberta. Numerous brainstorming discussions with them helped me gain valuable operational knowledge of their processes.

My earnest thanks to the University of Alberta for supporting my graduate studies and stay in Edmonton. I would like to thank my thesis examination committee for their careful reading of this work and for providing constructive critiques.

I am highly indebted to my friend Sujith Sundarajan for his continual support during my graduate studies. I would like to take this opportunity to extend my sincere thanks to Swetha Parvathaneni, Ashish Kapur and their parents, for the great care and attention they had given to me during my stay in Edmonton. I would like to thank my friends and well-wishers Dr. Prakash Jagadeesan, Greg McMillan, Jim Cahill, Rajakumar Jeyaraman, Vazir Mohideen, Ravendrudu Ravilla, Komala Ravilla, Srinivasa Rao Ravuri, Ashok Prabhu Masilamani, Ayyappasamy Sudalaiyadum Perumal, Kotireddy Ambati, Chaitanya Koduri, Sai Kosuri, Hemanth Davuluri, Abhay Pratap Singh, Srikirshna Chanakya Bodepudi, Anuj Narang, Aditya Tulsyan, Olav Bruggeman, Srikar Parvathaneni, Jayesh Jethwani, Thanouzu Ramr, Amrutha Harsha, Sonia Kapoor, Sagar Mashetti, Kiran Sivapuram, Vajra Teji Kanneganti, Pritam Chordia, Arun Pathiran, Dinaesh Kumar Raju, Sudarshan Sundar, Nikhil Mittal, Swanand Khare, Masih Sekhavat, and Marziyeh Keshavarz for their support during my graduate studies.

Table of Contents

Chapter 1 1
Introduction 1
1.1 Motivation 1
1.2 Objectives 3
1.3 Contributions 5
1.4 Organization 6
Chapter 2 8
Workflow Mining and OWBE 8
2.1 Introduction 8
2.2 Workflow Mining Tools 8
2.2.1 Extracting Workflows using Disco 9
2.3 Data Pre-processing of Event Logs in Process Industry 13
2.4 Exploring OWBE Platform 13
2.4.1 Applying Workflow Mining Tools in Process Industry 14
2.4.2 Developing Software-based Workflows 15
2.4.3 Saving Workflows 16
2.4.4 Conformance Monitoring 18
2.5 Summary 19
Chapter 3 20
Case Study 1: Developing Workflows for Pipeline Operations 20
3.1 Introduction 20
3.2 Product Batching 22
3.2.1 Batch-cut Mechanism at Delivery Stations 23
3.3 Suncor Energy Inc. Pipeline Operations 24
3.3.1 Tank-swing or Tank-switching Process at Denver Station 25
3.4 Event Logs for Tank-swing Process 30
3.4.1 Interpreting SCADA Event Log Data 31
3.4.2 Event Log Pre-processing Steps for Tank-swing Workflows 36

3.4.3 Developing Tank-swing Workflows from Event Log Data 43
3.4.4 Workflow Conformance 48
3.5 Developing Software-based Workflows for "Response to Leak Detection Alarms" Procedure using OWBE 50
3.5.1 Shortcomings of Paper-based SOPs 50
3.5.2 Merits of Software-based Workflows 52
3.5.3 Annotated Workflow Diagrams Developed in Visio 2010 58
3.5.4 OWBE Triggering Emergency Response Notifications 60
3.6 Summary 61
Chapter 4 62
Case Study 2: Developing Workflows for Heating Plant Operations 62
4.1 Description of the Heating Plant at University of Alberta 62
4.1.2 Orchestrated Workflows for Boiler Operations 63
4.2 Burner Rotation Sequence during Start-up of Boiler #4 64
4.2.1 Event Logs for Burner Rotation Sequence during Boiler #4 Start-up Procedure 67
4.2.2 Event Log Pre-processing Steps for Burner Rotation Sequence 68
4.2.3 Developing Workflows for Burner Rotation Sequence from Event Logs 74
4.2.4 Workflow Conformance for Burner Rotation Sequence 78
4.3 Developing Software-based Workflows for "Boiler #6 Shutdown Procedure" from Paper-based SOPs 81
4.3.1 Annotated Workflow Diagrams Developed in Visio Premium 2010. 92
4.3.2 Boiler #6 Shutdown Procedure -A Perspective from Control Room. .94
4.3.3 Information not Captured in a Control Room during Boiler #6 Shutdown Operation 97
4.4 Summary 99
Chapter 5 100
OWBE Application 100
5.1 Introduction 100
5.2 OWBE Home Page 103

5.3 Importing Excel File into OWBE Platform (Step – 2) 105
5.4 Filtering Events with Event Filter (Step – 3) 106
5.5 Saving Workflows in Workflow Document Library (Step – 4) 109
5.6 Summary 109
Chapter 6 110
Concluding Remarks and Future Work 110
6.1 Concluding Remarks 110
6.2 Future Work 111
References 113

List of Tables

Table 2.1: Configuring event log data within Disco software 11
Table 3.1: Batch-cut matrix for different products of crude oil............................ 29
Table 3.2: Tri-state alarms and status messages from SCADA event logs........... 31
Table 3.3: Different consoles from Suncor pipeline operations 33
Table 3.4: Commands count per console by week.. 35
Table 3.5: Tank-swing workflow start and end events 41
Table 4.1: Alarm and event data archived in the PROVOX system..................... 67
Table 4.2: Type of events or actions in PROVOX event logs 71
Table 4.3: Assigning case IDs and interpreting a custom event message 72
Table 4.4: Burner rotation sequence workflow start and end events 74
Table 4.5: Customized PROVOX event messages created based on "PvStr"...... 76
Table 5.1: SCADA events associated with the tank-swing process 100

List of Figures

Figure 1.1: Workforce demographic shifts in process industry 2
Figure 1.2: Block diagram of the OWBE platform.. 4
Figure 2.1: Event log data representing travel itinerary of a tourist 9
Figure 2.2: Workflow representing travel itinerary of a tourist 12
Figure 2.3: Extracting workflows using Disco software 15
Figure 2.4: Developing software-based workflows 16
Figure 2.5: Developing workflows in Microsoft Visio Premium 2010 17
Figure 2.6: Conformance monitoring using OWBE 18
Figure 3.1: A liquid pipeline distribution network 21
Figure 3.2: Batch train of petroleum products in a pipeline 22
Figure 3.3: Batch-cut mechanism at pipeline delivery stations 24
Figure 3.4: Rocky mountain pipeline network (US pipeline operations) 25
Figure 3.5: Interpreting a "Batch-ID" sequence in tank-swing event logs 26
Figure 3.6: A Batch ID initiation process 27
Figure 3.7: Tank terminals of RM-06 pipeline at Denver station......................... 29
Figure 3.8: SCADA event logs from the Suncor Pipeline Control Centre 32
Figure 3.9: Trends of commands count per console by week.............................. 35
Figure 3.10: Filtering raw event logs based on "Denver station"......................... 37
Figure 3.11: Standardizing date and time format... 37
Figure 3.12: Separating stations and tag names.. 38
Figure 3.13: Normalizing BATCH IDs registered at Denver station 38
Figure 3.14: Differentiation of commands from status messages......................... 40
Figure 3.15: Custom message by concatenating event and alarm message.......... 40
Figure 3.16: Mandatory events in a tank-swing workflow 41
Figure 3.17: A complete task with case IDs assigned to events 42
Figure 3.18: Part1: Critical events captured in a tank-swing workflow 44
Figure 3.19: Part2: Critical events captured in a tank-swing workflow 45
Figure 3.20: Part3: Critical events captured in a tank-swing workflow 46

Figure 3.21: Workflow conformance monitoring 49
Figure 3.22: Snapshot of a paper-based "response to leak alarms" SOP............. 51
Figure 3.23: Page1: Operational procedure details 53
Figure 3.24: Master-Workflow: "Response to leak detection alarms" 54
Figure 3.25: SW-1: Threshold for leak detection system > 0 55
Figure 3.26: SW-2: Threshold for leak detection system > 4.6 55
Figure 3.27: SW-3: Pipeline leak alarm active for ten minutes 56
Figure 3.28: A XOML code snippet corresponding to "Master Workflow" annotated diagram developed in the Microsoft Visio Premium 2010 59
Figure 3.29: A leak alarm triggering an emergency workflow............................ 61
Figure 4.1: Schematic of a drum-type natural-gas fired boiler 64
Figure 4.2: Burners arrangement in boiler #4 at the heating plant, University of Alberta 65
Figure 4.3: Burner ignition rotation sequence developed from boiler #4 SOP 66
Figure 4.4: Filtering raw event logs based on "Boiler 4" 69
Figure 4.5: Filtering event logs based on "B4 Burners" 70
Figure 4.6: Burner rotation workflow developed based on "SpStr" value 75
Figure 4.7: Burner rotation workflow developed based on "PvStr" value 77
Figure 4.8: Workflow conformance monitoring for burner rotation sequence..... 79
Figure 4.9: Auditable workflow activities for operational insight 80
Figure 4.10: Snapshot of a paper-based "boiler #6 shutdown procedure" SOP ... 81
Figure 4.11: Master Workflow – Boiler #6 shutdown procedure 82
Figure 4.12: Sub-workflow-1: Prepare to take boiler off-line 83
Figure 4.13: Sub-workflow-2: Monitor the system for pressure swings 84
Figure 4.14: Sub-workflow-3: Removing burners from service 85
Figure 4.15: Sub-workflow-4: Transferring combustion control.......................... 86
Figure 4.16: Sub-workflow-5: Opening vent 87
Figure 4.17: Sub-workflow-6: Continue lowering boiler master.......................... 88
Figure 4.18: Sub-workflow-7: Taking last burners offline 89
Figure 4.19: Sub-workflow-8: Boiler blow down... 90
Figure 4.20: Sub-workflow-9: Securing boiler 90

Figure 4.21: Sub-workflow-10: Put boiler in hot standby 91
Figure 4.22: Sub-workflow-11: Taking boiler for maintenance 92
Figure 4.23: A XOML code snippet corresponding to "Master Workflow" annotated diagram developed in Microsoft Visio Premium 2010 93
Figure 4.24: Profiles of boiler pressure, steam flow rate, steam temperature, and feedwater flow rate during boiler #6 shutdown operations.................................. 95
Figure 4.25: Operator console faceplate during boiler #6 shutdown operation.... 96
Figure 4.26: Verbal communications over phone during planned shutdown 97
Figure 4.27: Supervisor's log sheet obtained from the control room 98
Figure 5.1: Tank-swing workflow developed from "Denver Batch Cut" SOP .. 101
Figure 5.2: OWBE architecture diagram with developed web parts 102
Figure 5.3: OWBE web parts assembled within OWBE webpage 104
Figure 5.4: Importing Excel file and saving it as "RDF graph" 105
Figure 5.5: Using calender control and event filter 106
Figure 5.6: Filtering events based on timestamps and attributes 107
Figure 5.7: Setting an event as a "trigger" 107
Figure 5.8: Saving an event trigger 108
Figure 5.9: Selecting filtered events as a sequence.. 108
Figure 5.10: Saving selected events as workflows 108
Figure 5.11: "Thesis_Demo.xml" workflow when opened in an IE.................. 109

List of Abbreviations

API	American Petroleum Institute
ASM	Abnormal Situation Management Consortium
ASCII	American Standard Code for Information Interchange
A&E Logs	Alarm and Event Logs
ATMOS	Pipeline Leak Detection Software from ATMOS International
BPEL	Business Process Execution Language
BPM	Business Process Management
BMS	Burner Management System
CBD	Continuous blowdown valve (in a boiler operation)
Disco	Disco, Automated Process Discovery Software
DCS	Distributed Control System
ERP	Enterprise Resource Planning
FD	Forced Draft fan
HMI	Human-Machine-Interface
ISA	International Society of Automation
MES	Manufacturing Execution Systems
NRC-IRAP	National Research Council Canada – Industrial Research Assistance Program
OWBE	Orchestrated Workflows by Example Project

OSIsoft PI	Data archiving system from OSIsoft LLC
OPC	OLE for Process Control (Open Platform Communications)
OMNI	Flow Computers from OMNI Inc.
OPC UA	OPC Unified Architecture Standard
ProM	ProM, open-source framework for process mining
PROVOX	A DCS from FISHER (Emerson Process Management)
PLC	Programmable Logic Controllers
P&ID	Piping and Instrumentation Diagram
RM Pipeline	Rocky Mountain Pipeline
RTU	Remote Terminal Unit
RDF	Resource Description Framework
SIS	Safety Instrumented System
SOP	Standard Operating Procedures
SCADA	Supervisory Control and Data Acquisition System
SPCC	Suncor Pipelines Control Centre
SQL	Structured Query Language
T&T	Technology and Training
TAN	Total Acidic Number
ULSD	Ultra Low Sulfur Diesel
XOML	Extensible Object Markup Language
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Motivation

In today's competitive global market, some of the key operational constraints confronting the process industry are shortage of skilled workforce and procedural violations during plant start-up and shutdown operations which result in process upsets, thereby negatively impacting the environmental performance. Operational inconsistencies during grade transitions also introduce product variability that affects quality and financial performance of a company [1]. Process industry is going through a demographic shift in terms of workforce wherein a new generation of automation professionals are entering the field, whereas experienced professionals are leaving or retiring [1]. Shortage of highly experienced workforce is one of the imminent challenges regarding process industry over the next decade [2]. This looming shortage of skilled workforce is very severe in the hydrocarbon industry [3].

According to a survey conducted by the American Petroleum Institute (API) in 2004, the average age of the workforce in the hydrocarbon industry was estimated to be around 49 years [4]. Hence, the majority of those experienced operational crews is close to retirement by now. By 2009, there has been an estimated shortage of 38 percent of engineers and a shortage of 28 percent of instrumentation and electrical workers [4]. According to a report released by the International Society of Automation (ISA) in 2007, the annual demand for new automation engineers in the United States was estimated to be around 15,000 [5]. The U.S. Bureau of Labor Statistics also estimates that the demand for automation professionals will increase by 6% over the next decade [6]. According to a talent crisis report issued by Deloitte Research [4], a major transition in the workforce mix has occurred in 2010 where a significant number of new and inexperienced engineers have joined the operational workforce as shown in Figure 1.1.

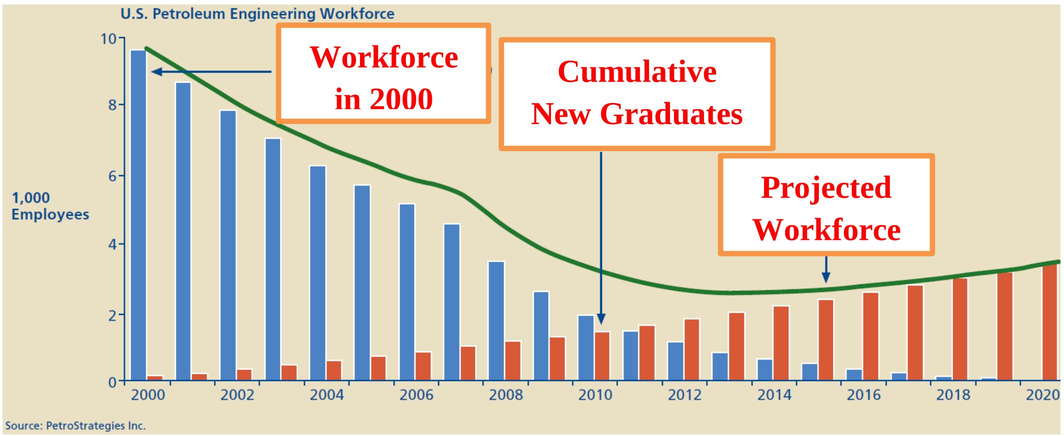


Figure 1.1: Workforce demographic shifts in process industry [1]

The Abnormal Situation Management (ASM) consortium reports that, an average of 42 percent of abnormal situations in process operations is directly attributed to the skills and experience of the operational crew while handling a complex procedure such as a plant start-up and shutdown operation [7]. Similarly, operational inconsistencies during grade changes hurt product quality and contribute to wastage of raw materials and energy [8]. Operational effectiveness varies among operators and depends on their expertise gained over several years of experience. Experienced workforce of engineers, operators and technicians gain valuable operational knowledge over many years of experience and these skills are very critical for safe, reliable and profitable operations of the process [1], [8]. When these experienced workforces retire or leave, the critical

[1] Exhibit 2. Sampath, R., & Robinson, M. (2005). The Talent Crisis in Upstream Oil & Gas. Retrieved August 13, 2013, from http://www.deloitte.com/assets/Dcom-Canada/Local Assets/Documents/CA_ER_TalentCrisisUpstreamOilGas_18_11_2005.pdf

operational knowledge captured in their heads permanently disappears from the organization. Process industry finds it very difficult to capture this valuable operational knowledge of the retiring workforce and transfer it to a new generation of automation professionals. To meet global competition, and to improve operational, financial and environmental performance, companies are investing millions of dollars and man hours in technology advancement and training their workforce (T&T investment) [9]. In this work, we focus on the development of a software-based platform, which can capture events from a particular process operation. These captured events can then be used to develop workflows, which can retain critical operational knowledge of the experienced workforce as a corporate asset within the organization. This software-based platform is termed as "Orchestrated Workflows by Example" (OWBE). The OWBE platform is instrumental in transferring technical know-how of plant operations to the next generation of engineers, operators and technicians.

1.2 Objectives

The Orchestrated Workflows by Example platform is a collaborative research project between the Sponsor Company, NRC-IRAP (National Research Council Canada), and the University of Alberta. The web parts developed for OWBE platform are an outcome of the team effort at the University of Alberta. The underlying block diagram of OWBE platform is shown in Figure 1.2. OWBE is a component of a data integration platform that archives all operator actions, application invocations, and process alarms and events into an event and alarm database. Simply put, operator's expertise in adjusting set-points, tuning a control loop, troubleshooting an equipment and such tasks are archived in the form of events within an event log database. By analyzing and data-mining these text-based event messages, it is possible to capture operator's knowledge while executing a work procedure such as start-up and shutdown of processes, and troubleshooting an unstable control loop. The OWBE platform developed should have the capability to allow operators to select, filter and capture their own actions while they execute their work procedures.

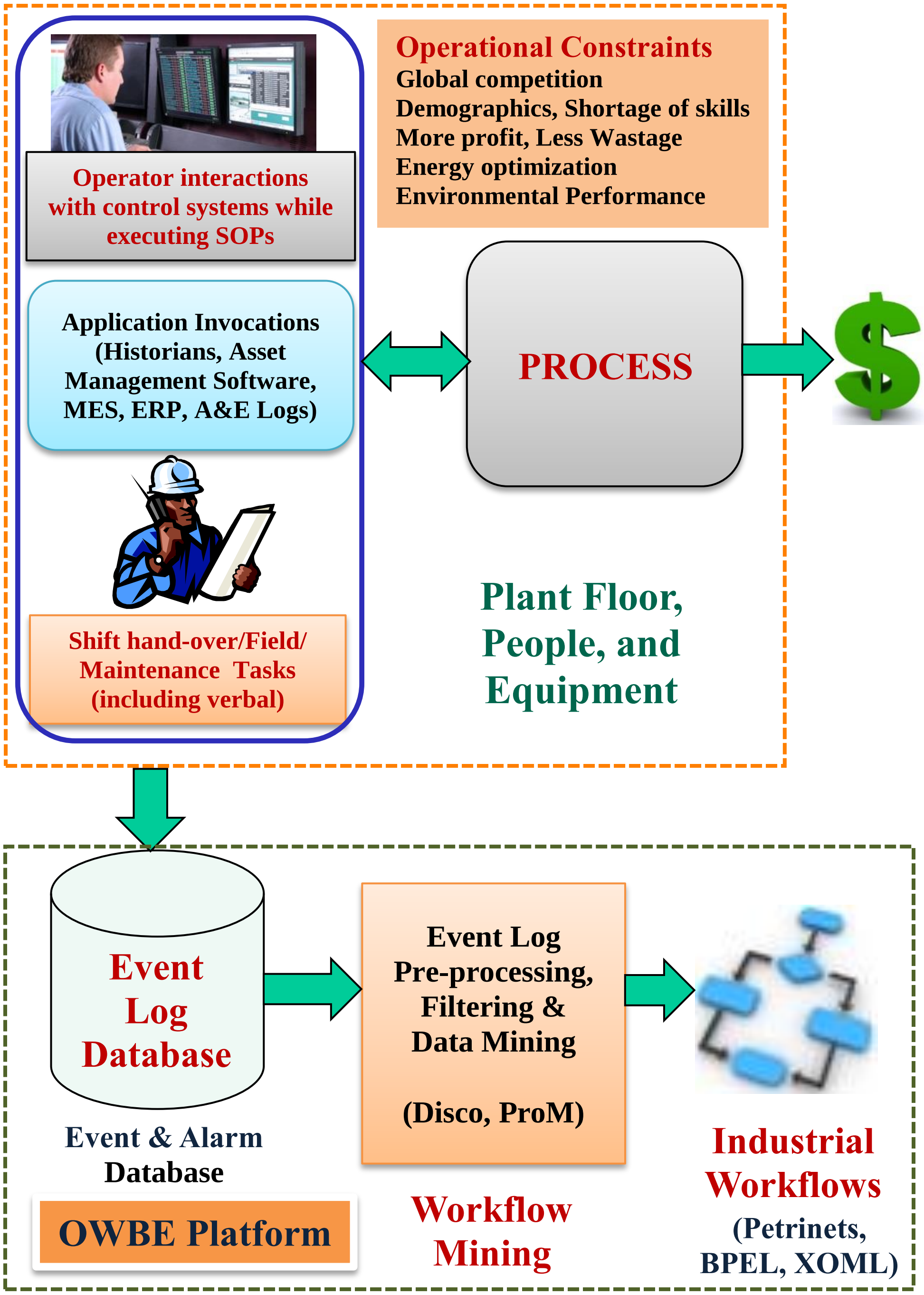


Figure 1.2: Block diagram of the OWBE platform

As shown in Figure 1.2, all the events generated from a plant are archived in an event log database. These event messages are pre-processed and filtered to extract industrial workflows using workflow mining tools such as Disco software [10] (refer to Chapter 2). An industrial workflow is a sequence of tasks or actions that describe an operational procedure. It could be as simple as a sequence of operator actions to control level in a tank, troubleshoot a control loop and start-up of a boiler. Industrial workflows are often referred to as *orchestrated* workflows since they involve a team of people who coordinate, manage and execute a complex operational procedure to its completion. For instance, a pipeline start-up procedure involves people from various cross-functional teams such as SCADA (Supervisory Control and Data Acquisition) analysts, control room operators, field personnel and technicians who work together while executing a pipeline start-up sequence.

1.3 Contributions

The work carried out in this thesis is aimed at developing procedures for creating workflows and demonstrating the effectiveness of the OWBE platform by application on industrial case studies. These case studies are intended to aid the Sponsor Company in building a case for commercialization of the OWBE technology. Some of the key contributions are as follows:

1. The concepts of OWBE technology are demonstrated to the end-users via industrial case studies (Suncor Energy Pipelines and Heating plant at the University of Alberta).
2. Identify opportunities and case studies where OWBE technology can fit into their day-to-day process operations.
3. Conduct interviews with supervisors and operators to understand their expectations of OWBE software and its associated User Interface (UI).
4. Based on discussions with the end-users, provide input to OWBE web parts development team for tailoring the User Interface and features

associated with "Event Loader" & "Event Filter" functions in the OWBE platform.

5. Understand and solve challenges in data pre-processing to handle event messages from a variety of information systems used in the process industry (Emerson Distributed Control System, PROVOX system, Supervisory Control and Data Acquisition system, and OSIsoft PI system).
6. Explore various workflow mining tools (automated process discovery from event logs) available in Business Process Management (BPM) and identify opportunities to use them in the context of process industry.

1.4 Organization

The thesis is organized as follows:

- Chapter 2 gives a brief introduction to the concept of workflow mining and workflow mining tools available for data analysis. We discuss various functionalities of the OWBE platform with a block diagram.
- In chapter 3 we discuss the first industrial case study from Suncor Energy Pipeline operations. This chapter gives a brief overview of the tank-swing process and the challenges involved in pre-processing of SCADA event logs associated with it. We develop tank-swing workflows from operator actions and compare them with standard operating procedure(s) to check for conformance monitoring.
- Chapter 4 looks at the second industrial case study from the heating plant at the University of Alberta. This chapter gives a brief introduction to boiler operations and burner rotation sequence. Workflows for burner rotation sequence are developed from the PROVOX event logs and are compared with the standard operating procedure(s) to check for conformance monitoring.
- Chapter 5 demonstrates various functionalities of the OWBE platform developed by the group at the University of Alberta. This chapter

provides users with a step-by-step procedure to use OWBE for constructing workflows from the event logs.

- Concluding remarks, future work and the recommendations for commercializing OWBE platform are discussed in Chapter 6.

Chapter 2

Workflow Mining and OWBE

2.1 Introduction

Workflow mining[2] or process mining is defined as *"a process management technique that allows for the analysis of business processes based on event logs. The basic idea is to extract knowledge from event logs recorded by an information system. Process mining aims at improving this by providing techniques and tools for discovering process, control, data, organizational, and social structures from event logs"* [11]. Workflow mining techniques are widespread in the areas of business process management (BPM) (Tiwari, Turner, & Majeed, 2008) [11], [12]. In our work, we intend to use these techniques in the context of operations related to the process industry.

2.2 Workflow Mining Tools

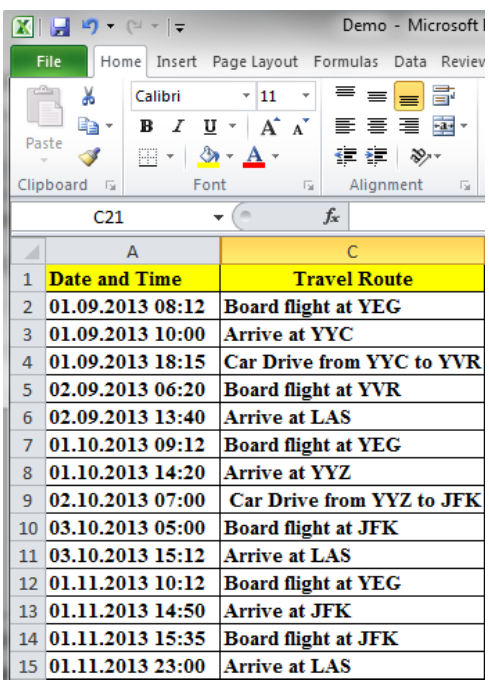
Several open-source workflow mining tools are available for extracting process models from the event logs. The workflow mining framework ProM [13], developed at the Department of Mathematics and Computing Science, Eindhoven University of Technology (TU/e) is an open-source based tool, which is widely-used for academic research. Workflows extracted from ProM are saved in the form of Petri nets (Van Dongen, De Medeiros, Verbeek, Weijters, & Van Der Aalst, 2005) [13]. Some of the proprietary workflow mining tools that are readily available for data analysis include Futura Reflect, Disco, Nitro, and Interstage Automated Process Discovery [11]. For extracting workflows from the event logs in our case studies, we use a demo version of Disco, an automated process

[2] http://www.processmining.org/, Math&CS department, Eindhoven University of Technology.

discovery software developed by Fluxicon (a spin-off of process mining research at TU/e) [10]. Disco software is a user-friendly tool that accepts event messages in the form of Excel files and the resultant workflows are exported as PDF, PNG image, and JPEG images for easy visualization [10]. To automatically extract workflows from the event logs, Disco uses a built-in data mining algorithm developed by Christian W. Günther [14], [15]. In this work, some of the terminologies associated with Disco software are tailored to use it in the context of the process industry for extracting industrial workflows.

2.2.1 Extracting Workflows using Disco

Using a simple example, we illustrate how Disco software automatically extracts workflows from the event logs. Let us consider a contrived example of a tourist from Edmonton who plans to visit Las Vegas during the first week of every month. The tourist travels for pleasure by splitting up his journey partly by air travel and partly by driving and visiting in between places. However, he starts on



	A	C
1	Date and Time	Travel Route
2	01.09.2013 08:12	Board flight at YEG
3	01.09.2013 10:00	Arrive at YYC
4	01.09.2013 18:15	Car Drive from YYC to YVR
5	02.09.2013 06:20	Board flight at YVR
6	02.09.2013 13:40	Arrive at LAS
7	01.10.2013 09:12	Board flight at YEG
8	01.10.2013 14:20	Arrive at YYZ
9	02.10.2013 07:00	Car Drive from YYZ to JFK
10	03.10.2013 05:00	Board flight at JFK
11	03.10.2013 15:12	Arrive at LAS
12	01.11.2013 10:12	Board flight at YEG
13	01.11.2013 14:50	Arrive at JFK
14	01.11.2013 15:35	Board flight at JFK
15	01.11.2013 23:00	Arrive at LAS

YEG: Edmonton Airport
YYC: Calgary Airport
YYZ: Toronto Airport
YVR: Vancouver Airport
JFK: New York Airport
LAS: Las Vegas Airport

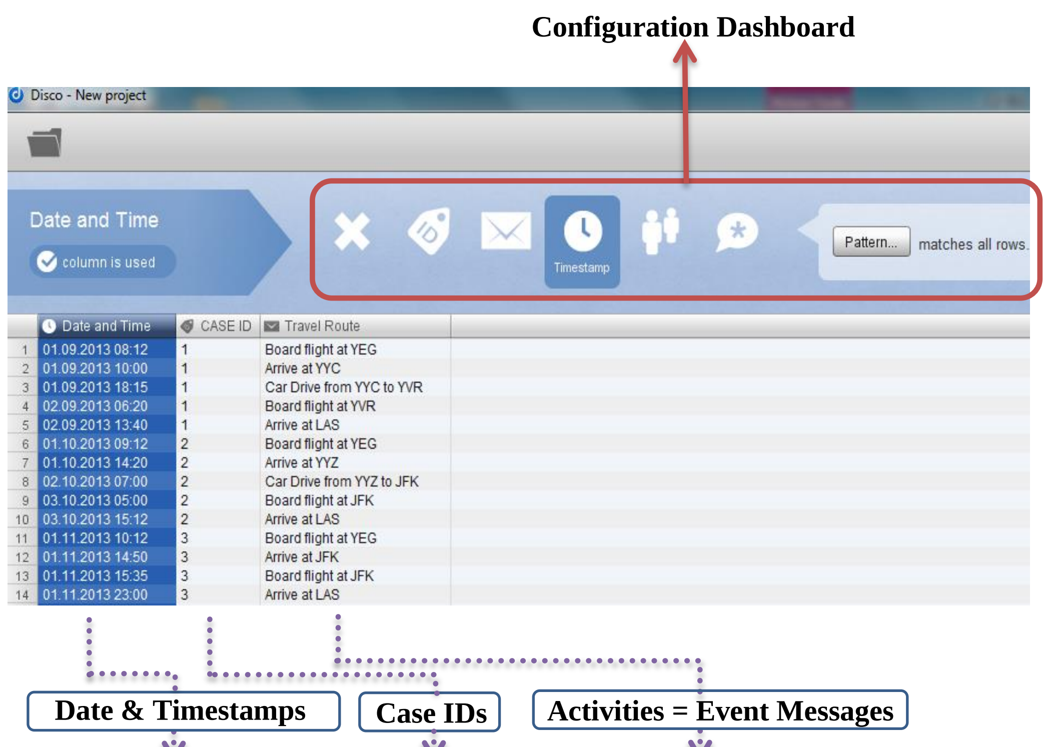
Figure 2.1: Event log data representing travel itinerary of a tourist

the first day of every month from Edmonton. Figure 2.1 shows a snapshot of a Microsoft Office Excel spreadsheet where the first column denotes the date and time information associated with his travel and the second column shows the travel route associated with his itinerary during three consecutive months in 2013. As shown in Table 2.1, each of the Excel columns is referred to as “attributes”. Each of the row represents “events” associated with his journey. A “task” represents a complete scope of the process. In our running example, each task represents an itinerary that starts with the event “Board flight at YEG” and ends with the event “Arrive at LAS”.

The three mandatory data attributes required to automatically extract workflows from the event logs are *“Date and Time”, “Case ID”, and “Activities”* [16]. *Date and Time* attribute represents the information about the actual date of occurrence of an event and the timestamp associated with it. This column helps Disco to understand the order of the activities archived in each task.

A *CaseID* represents a unique event in the process. All the events and activities within each specific task have same CaseIDs. CaseIDs help to differentiate events from task to task. The processing of assigning unique IDs to the event messages is called “Case Identification”. Case IDs also set the start and end events of a process. For the purpose of extracting workflows from the event logs, we manually set these CaseIDs throughout this thesis. CaseIDs are very similar to tuning parameters that decide which events are to be included and excluded to meet the analysis goals.

The *activities* column contains text-based event messages that represent tasks executed in a process. These event messages appear as blocks in the workflow diagram. Users can set any piece of information as an activity based on their requirements. In our running example, if the tourist wishes to see the flight service associated with his itinerary as well, it is possible to customize these event messages by concatenating the flight numbers along with the text messages denoting his travel route. For instance, the event message "Board flight at YEG" from Table 2.1 can be customized to "Board flight at YEG_AIR_CANADA_Flight_777-300ER for a better analysis in the workflow diagram extracted from the event logs.



Date and Time	CASE ID	Travel Route
01.09.2013 08:12	1	Board flight at YEG
01.09.2013 10:00	1	Arrive at YYC
01.09.2013 18:15	1	Car Drive from YYC to YVR
02.09.2013 06:20	1	Board flight at YVR
02.09.2013 13:40	1	Arrive at LAS
01.10.2013 09:12	2	Board flight at YEG
01.10.2013 14:20	2	Arrive at YYZ
02.10.2013 07:00	2	Car Drive from YYZ to JFK
03.10.2013 05:00	2	Board flight at JFK
03.10.2013 15:12	2	Arrive at LAS
01.11.2013 10:12	3	Board flight at YEG
01.11.2013 14:50	3	Arrive at JFK
01.11.2013 15:35	3	Board flight at JFK
01.11.2013 23:00	3	Arrive at LAS

Task 1 (Case ID 1), Task 2 (Case ID 2), Task 3 (Case ID 3)

Table 2.1: Configuring event log data within Disco software

Table 2.1 shows Disco's configuration dashboard for importing Excel spreadsheets and to configure aforementioned attributes to automatically extract workflows. There are 14 event messages associated with his itinerary (for three consecutive months) that are categorized into 3 tasks based on the CaseIDs.

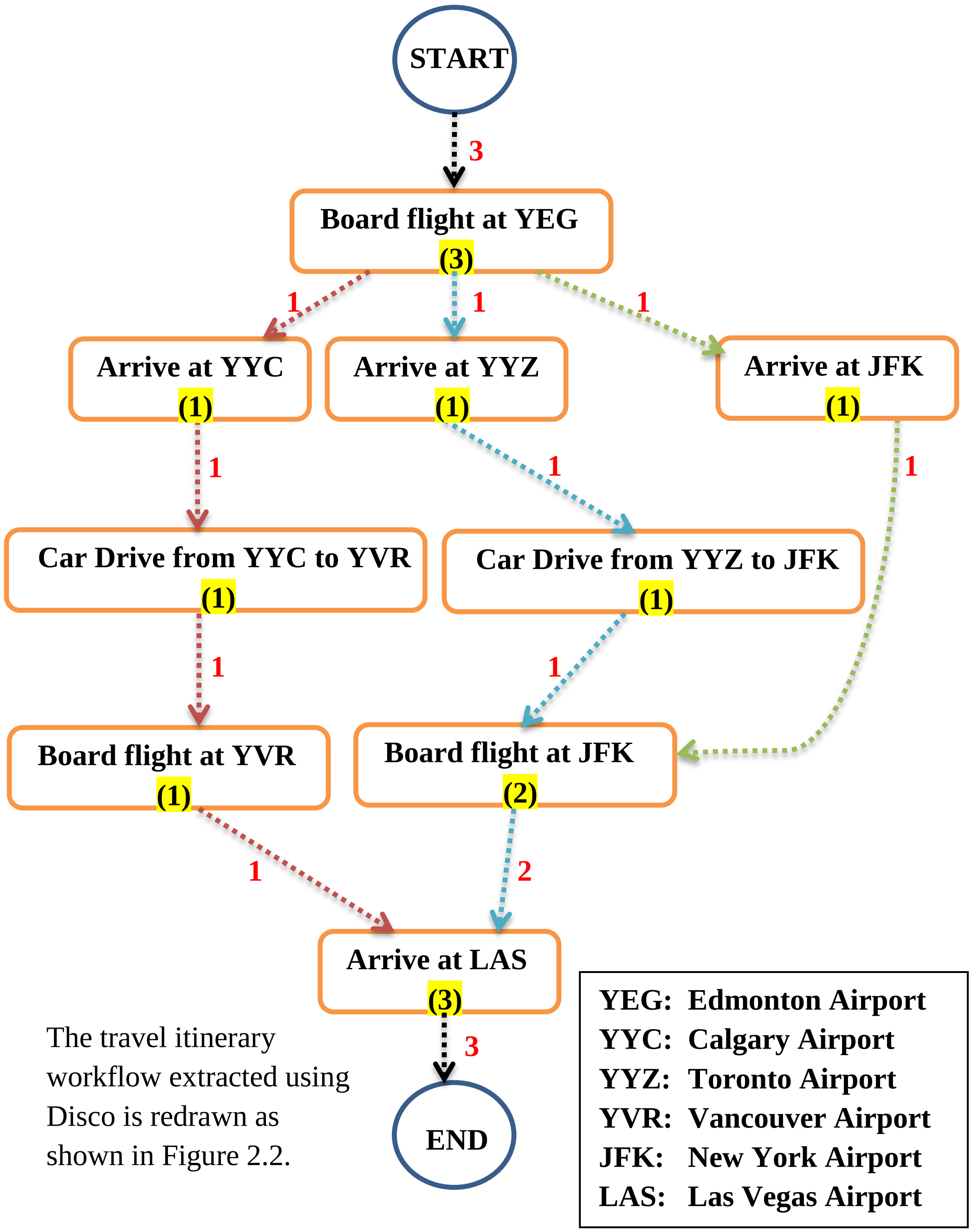


Figure 2.2: Workflow representing travel itinerary of a tourist

Figure 2.2 shows the travel itinerary workflow extracted from the event logs shown in Figure 2.1. Each text-based event message appears as a workflow activity block in the workflow diagram. The frequency of occurrence of an event is shown with a number within each block. A directed arrow connecting two blocks indicates the path of actions or events occurred during his travel. A number on the directed arrow indicates the number of times a particular transition occurred between those two blocks (during travel) in the direction of the arrow.

2.3 Data Pre-processing of Event Logs in Process Industry

Events logs retrieved from contemporary information systems used in the process industry are vendor specific. For instance, the structure of archived event logs associated with Emerson DeltaV Event Chronicle is different from the PROVOX event and alarm logs. A few of the automation vendors accommodate interoperable open standards for data connectivity which are compliant to OPC Foundation (OPC alarm & event archives) [17].

Hence, to extend the concept of automated workflow extraction to the process industry, significant amount of data pre-processing has to be carried out to convert event messages retrieved from a variety of information systems into a standardized format acceptable for use with workflow mining tools such as Disco. For a user, an intrinsic understanding of the physical processes and the information systems is necessary to perform event log pre-processing. Chapters 3 and 4 illustrate how event log pre-processing enables Disco to automatically extract workflows for the two industrial case studies.

2.4 Exploring OWBE Platform

Thanks to the state-of-the-art information systems used in the process industry, the majority of the operator interactions with the plant floor (process and equipment) are archived in the event log database. The OWBE platform helps operators to select, filter and capture their own actions while they execute a work procedure. The OWBE platform is developed as a modular component of a data integration platform that collects all the data related to plant operations. Hence, this data is a rich source of operator actions, which are archived in the form of event and alarm messages. Figure 2.3 shows the component flow diagram of the OWBE platform. The details of the components constituting the OWBE platform and their functionalities are given in the following subsections. The functionality of the entire platform is demonstrated on two industrial case studies, which are discussed in Chapters 3 and 4.

2.4.1 Applying Workflow Mining Tools in Process Industry

From Figure 2.3, the blocks which are shown in red depicts how operator actions, application invocations, plant event and alarm messages are archived into an event and alarm database [11]. Operator actions such as set-point changes, controller tuning steps, control loop diagnostic steps, and instrument troubleshooting tasks are archived in the event log database. Application invocations such as operators viewing process trends, process historian, requests for system diagnostic information are also archived in the event log database. These event messages are collected in the form of Excel spreadsheets where data pre-processing is done to convert raw event messages into a standardized text format acceptable for use with workflow mining software such as Disco.

Similarly, event log data is collected from third party real-time data archiving applications such as OsiSoft PI systems, DeltaV Event Chronicle, PROVOX alarm and event logs in an Excel spreadsheet for data pre-processing. These pre-processed event messages are then uploaded into Disco software to automatically extract workflows. Each block in a workflow diagram represents an ongoing event or a task performed by an operator while operating a process. Hence, a workflow represents a model of a plant operation that captures the operator’s actions. Hence, when an expert (or experienced) operator is operating the process, his/her expertise on the process operations can be captured by using these tools.

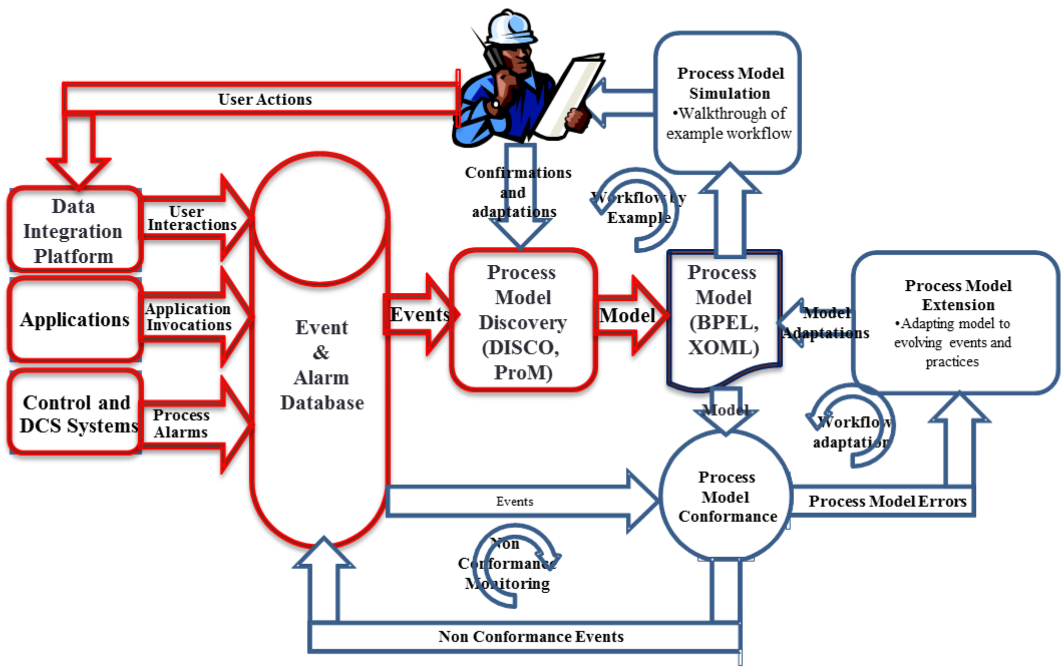


Figure 2.3: Extracting workflows using Disco software [10]

2.4.2 Developing Software-based Workflows

Written procedures or paper-based SOP(s) are often outdated and are left on the shelf [18]. Moreover, in situations where plant operations are disrupted due to unforeseen events, it is cumbersome to sift through paper-based SOPs for the procedures required to bring the equipment back to safe conditions or shut it down safely. In order to adapt to the changing operating environments and time evolving events, experienced operators often follow their own workaround practices which can be different from a written procedure. This is similar to a "handy black-book" that contains hidden treasure of operational knowledge and is accessible only by the owner. Written procedures do not capture these current operator actions and hence experienced operators take away this expertise when they leave the organization.

The OWBE platform facilitates companies to capture expert knowledge from their operational crew. Operators are able to select, filter and capture their own actions while they execute their work procedures. Operators are able to turn paper-based procedures into live orchestrated workflows that capture ongoing events and current actions thereby keeping the operating practices up-to-date.

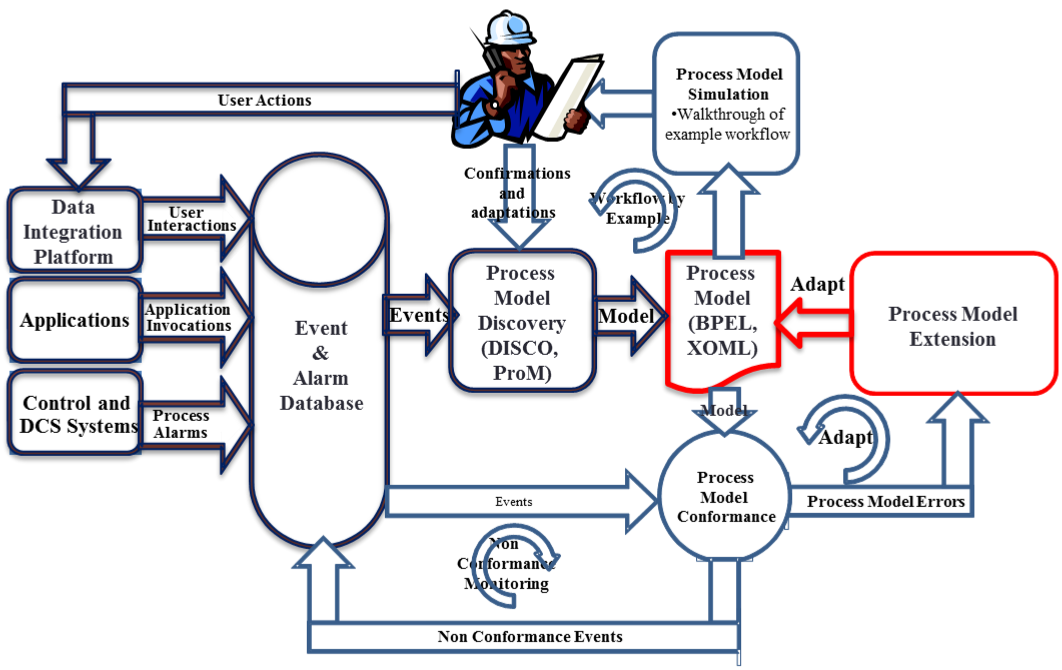


Figure 2.4: Developing software-based workflows

From Figure 2.4, the blocks which are shown in red indicates how modifications to the operating procedures can be easily incorporated into a software-based workflow on-the-fly. The concept of developing live orchestrated workflows using Microsoft Visio Premium 2010 is demonstrated using two industrial case studies in Chapters 3 and 4.

2.4.3 Saving Workflows

In the later stages of OWBE development, workflows are developed and configured in Microsoft SharePoint Designer 2010 and are deployed as web service applications. Hence, workflows will be saved in a XOML format (renamed as XAML), a Microsoft standard for documenting workflows in XML. In some business processes, workflows are also documented using Business

Process Execution Language (BPEL) as shown in Figure 2.4. In this work, workflows which are extracted using workflow mining tools such as Disco are saved as image (PNG or JPEG) files.

Also, SharePoint workflow stencils (workflow activities) available in the Microsoft Visio Premium 2010 [19] are extensively used to turn paper-based SOP(s) into live orchestrated software-based workflows [20]. Operators will be able to drag, drop and connect these workflow activities to author an annotated diagram that captures operator's expert knowledge. These annotated diagrams are then saved as Visio Workflow Interchange (.vwi) files which contain coded XAML (or XOML) information about the sequence of tasks executed by an operator during plant start-up or shutdown operations.

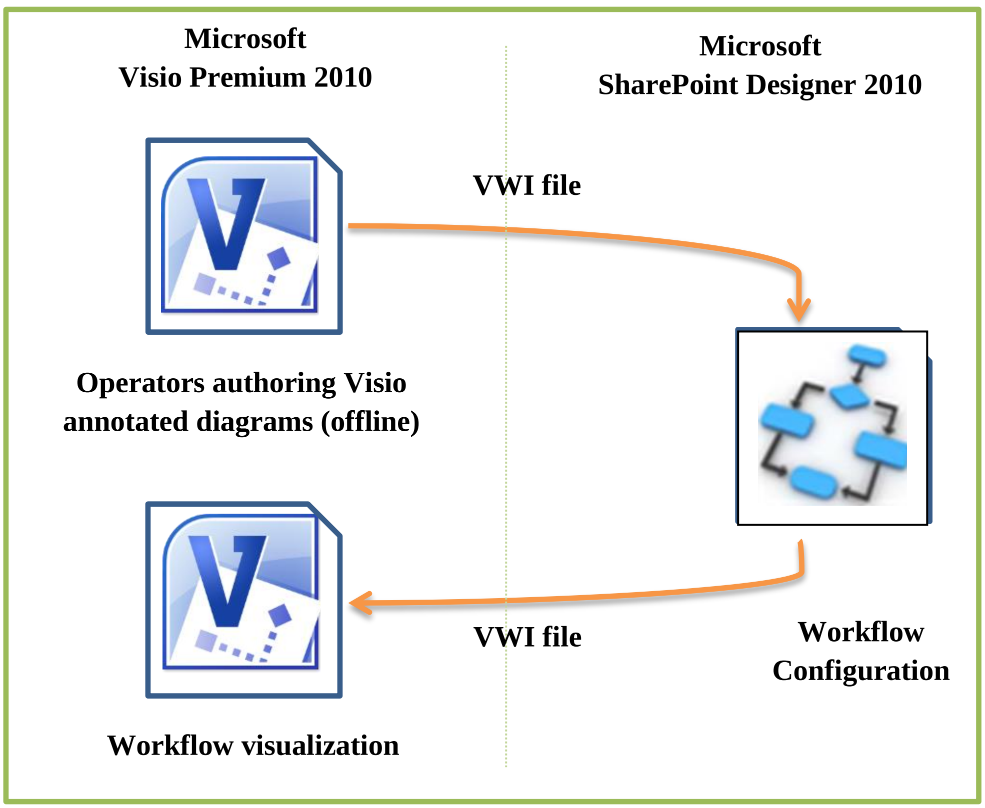


Figure 2.5: Developing workflows in Microsoft Visio Premium 2010 [19]

As shown in Figure 2.5, Visio Workflow Interchange files are directly imported into Microsoft SharePoint Designer 2010 for further configuration. This enables operators to create workflow diagrams offline and then configure them in OWBE using web services. Chapter 3 and 4 illustrate how operators are able to author, edit, and publish their self-service workflows in the Microsoft Visio Premium 2010.

2.4.4 Conformance Monitoring

The concept of conformance monitoring is shown using red colored blocks in Figure 2.6. Event log database continuously archives events and alarms that are generated in the plant. Conformance monitoring helps to monitor ongoing events and operator actions by comparing them with an existing workflow model. Any deviation observed is further investigated to understand and solve operational problems associated with process, equipment and human factors. Conformance monitoring also helps to identify operators who deviate from standard operating

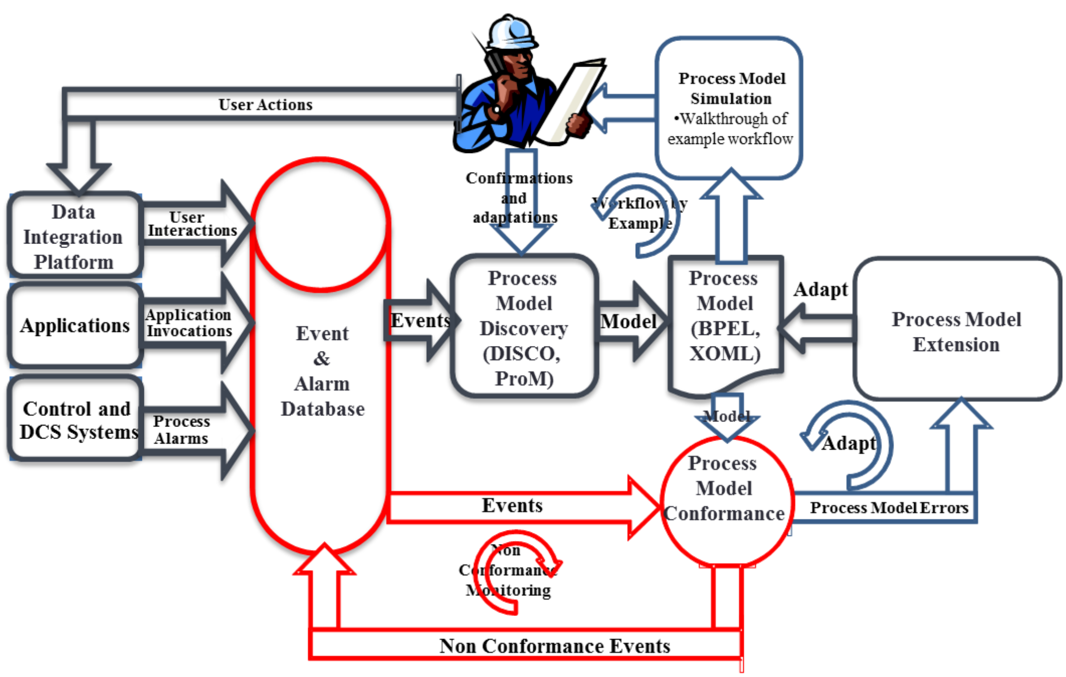


Figure 2.6: Conformance monitoring using OWBE

procedure(s) or who fail to follow a correct procedure while operating a process. In a famous ISA publication on Safety Instrumented Systems (SIS), Gruhn and Cheddie stated that *"If procedures are to be accounted for as a protection layer, they need to be documented, people need to be trained to follow them, and their use must be audited"* [21]. This excerpt emphasizes the importance of conformance monitoring of operator actions with respect to process safety. Conformance monitoring provides documented information (audit) on operator actions while executing a procedure. In this work, the concept of conformance monitoring is demonstrated using two industrial case studies from boiler and pipeline operations.

2.5 Summary

This chapter gives a brief introduction to the concept of workflow mining and the tools available to extract workflows from the event logs. A brief overview of the components within OWBE platform is also presented in this chapter.

Chapter 3

Case Study 1: Developing Workflows for Pipeline Operations

3.1 Introduction

Pipeline transportation is a reliable, economical, and safest mode of carrying large volumes of crude oil and petroleum products over large distances [22]. Crude oil (oilsands synthetic crude), refined petroleum products and natural gas are the major types of fluids transported by pipelines. This chapter specifically focusses on liquid pipeline operations. Pipelines are classified into four major categories based on their location, length, and volume of liquid they transport [23]. Crude oil is transported from offshore drilling platforms, tanker vessels and onshore oil wells to intermediary crude storage tanks through gathering pipelines as shown in Figure 3.1. Feeder lines transfer crude oil from these intermediary tanks to trunk lines or nearby refineries for further processing into petroleum products. Refined products are then transported over large diameter transmission lines to consumer markets within a country or across the continent [24].

Distribution terminals are centres where crude oil or petroleum products are taken off transmission pipelines into terminal storage tanks. From these storage tanks, distribution lines transport jet fuel to local aviation terminals and heating oil to household and industrial applications. Distribution terminals also feed gasoline to marketing terminals where gasoline is moved by trucks to gas stations, stores and consumer markets. Figure 3.1 gives a schematic of a pipeline network starting from crude oil gathering and ending with the distribution. However, the actual configuration of a pipeline transportation network varies in scale and complexity.

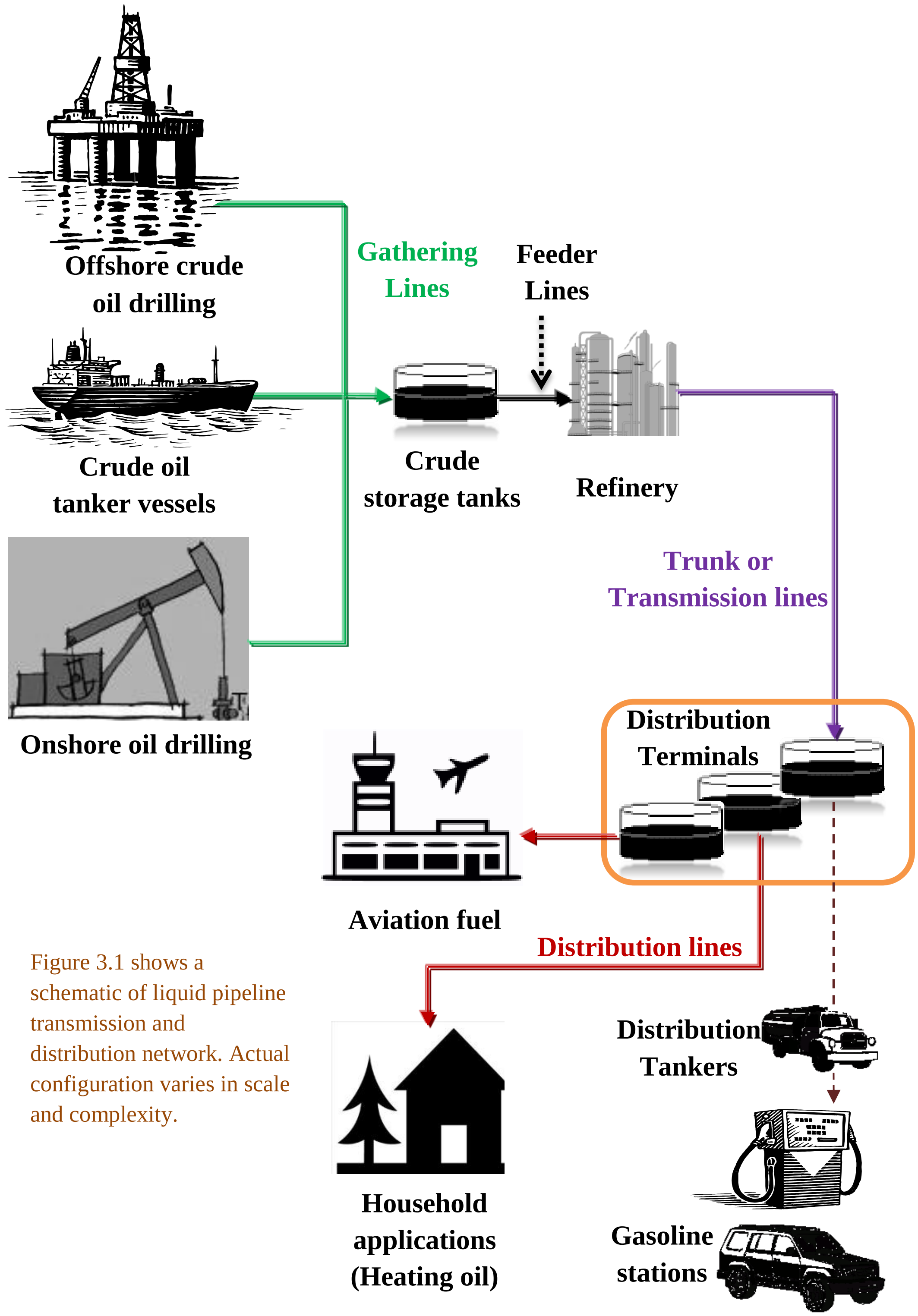


Figure 3.1 shows a schematic of liquid pipeline transmission and distribution network. Actual configuration varies in scale and complexity.

Figure 3.1: A liquid pipeline distribution network

3.2 Product Batching

Pipeline operators move different types of crude oil, refined petroleum products or grades of a same product in a single pipeline through a process known as "product batching" [25], [26]. Batching helps meet the diversified demands of the energy consumers using the same pipeline. Batching also enables pipeline operators to meet their operational and financial objectives. A batch is a specific quantity (measured in barrels) of a distinct crude oil or petroleum product with common physical properties [27]. For custody transfer, different products of crude oil (or refined products) or different grades of a similar crude oil (or refined product) are initially stored in segregated storage tanks at pipeline origination stations and are delivered to shipper nominated tanks at the delivery stations. Several batches of petroleum products of the same type are butted against each other forming a batch train and are moved sequentially in a pipeline. Figure 3.2 shows a typical sequence or a batch train of petroleum products. Batches of petroleum products of the same type are moved together in cycles to minimize product mixing and contamination as shown in Figure 3.2 [28]. When two grades abut with each other in a distillate cycle, an "interface" zone is developed which

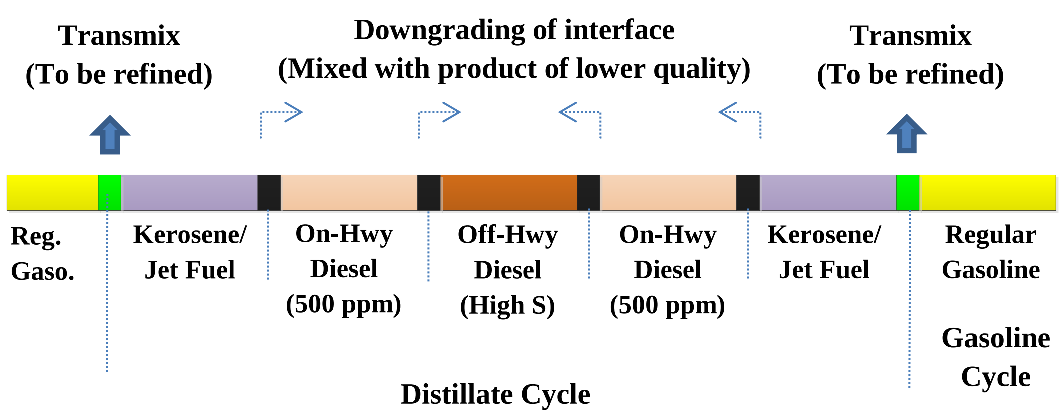


Figure 3.2: Batch train of petroleum products in a pipeline[3]

[3] Redrawn based on a typical refined product sequence, Energy Information Administration (EIA), (May, 2001) "The Transition to Ultra-Low-Sulfur Diesel Fuel: Effects on Prices and Supply".

downgrades the product of higher quality. A similar situation occurs when premium and regular gasoline are butted against each other in a gasoline cycle which reduces the volume of higher quality premium gasoline product [25]. A "transmix" is formed as an interface when two different products abut with each other. A transmix is formed when gasoline and distillate are interfaced as shown in Figure 3.2. At the receiving terminals, the transmix is separated and routed to a transmix tank to avoid contamination of either gasoline or distillate batch. This transmix is then reprocessed or refined into saleable products and then introduce again into shipper nominated storage tanks. Batch scheduling system helps pipeline operators to schedule monthly batch cycles beforehand which enables them to custody-transfer products of the same type in a sequence thereby minimizing product contamination [25].

3.2.1 Batch-cut Mechanism at Delivery Stations

Batches of petroleum products injected sequentially into the pipeline are separated into distinct products at delivery stations using batch-cut mechanism. Batch-cut mechanism is a critical process which accurately detects the interface between different batches and route distinct types and grades of products into distinctive shipper designated storage tanks at a delivery station or distribution terminal. The batch cutting process relies on a fundamental principle that each product (as a batch) injected into a pipeline has a distinct specific gravity from its adjacent product. Even various grades of a similar product have marginally varying specific gravity values [26], [28]. The batch cutting process involves continuous monitoring of specific gravity and accumulated batch volume of a batch train when it approaches a tank terminal. A density measuring meter (gravitometer) is installed ahead of a delivery station (at metering stations) and is used to detect variations in specific gravity of the pipeline product stream. Completion of custody-transfer of a particular batch (current) and the commencement of the next incoming batch is indicated by a change in specific gravity at the batch interface.

An alarm generated from gravitometer alerts the operators at the control centre and prepares them to "make a batch-cut" by swinging or switching cut valves. An accurate batch-cut ensures that batches in a pipeline are routed to designated tanks and prevents batches from product contamination within a tank [29].

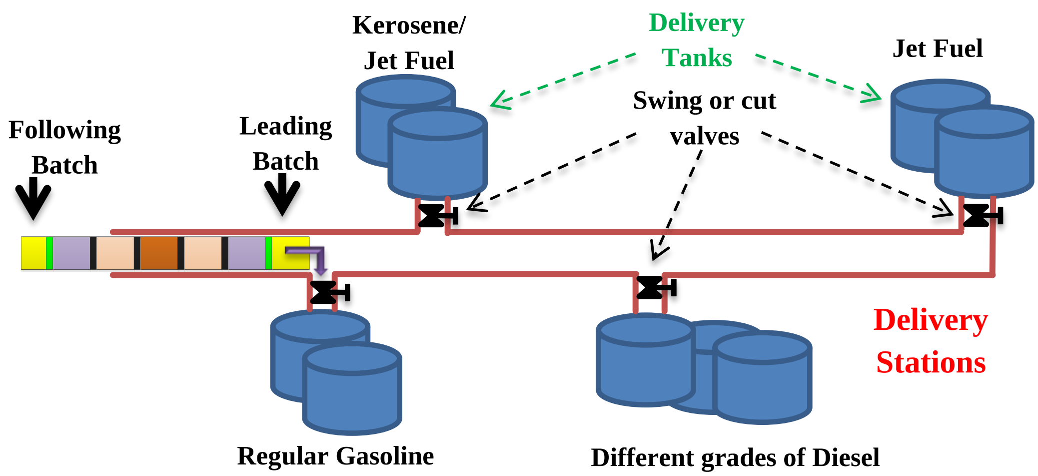


Figure 3.3: Batch-cut mechanism at pipeline delivery stations

Figure 3.3 shows a schematic of batch-cut mechanism at a delivery station. However, the actual configuration of a batch-cut mechanism and related instrumentation at a delivery station or terminal varies in scale and complexity.

3.3 Suncor Energy Inc. Pipeline Operations

Suncor Energy Inc. owns and operates OSPL Pipeline (Oilsands Pipeline) within Canada and Rocky Mountain Pipeline (RM) in the United States. The SCADA geographical schematic of the RM pipeline is shown in Figure 3.4 where it extends from Guernsey in the district of Wyoming to Denver in the state of Colorado. For our case study, we consider pipeline RM – 06 which starts from the Cheyenne pumping station and ends at the Denver receiving station. Our prime focus is on Denver receiving station as we analyze SCADA event logs and develop workflows for tank-swing process which are associated with the Denver receiving station.

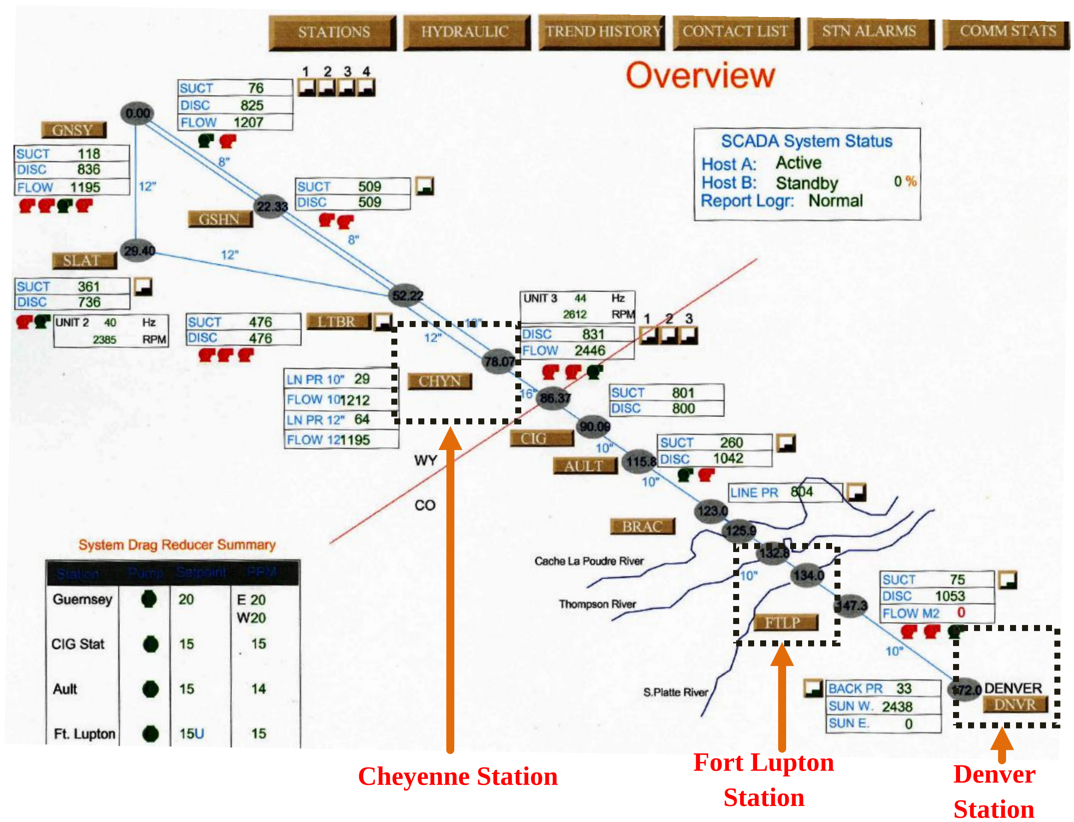
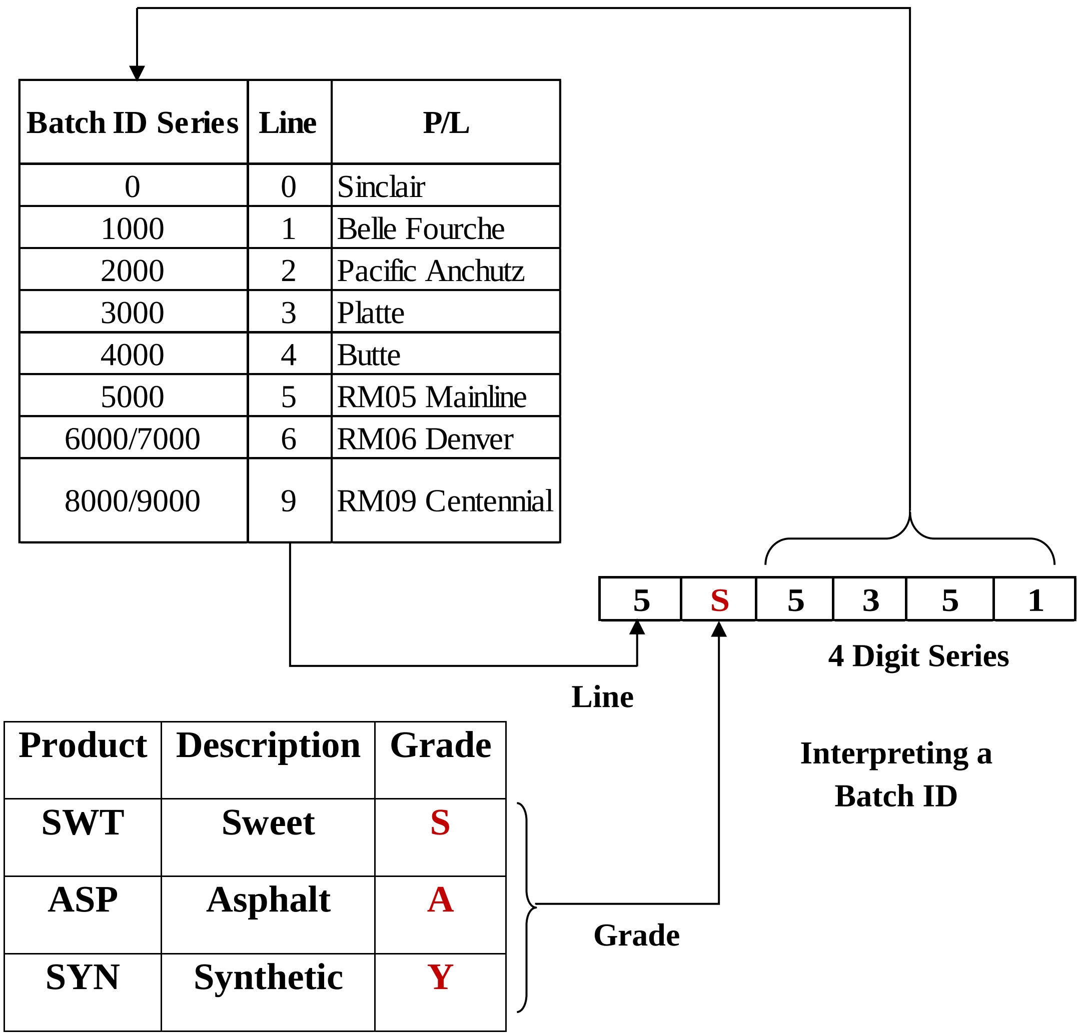


Figure 3.4: Rocky mountain pipeline network (US pipeline operations)

3.3.1 Tank-swing or Tank-switching Process at Denver Station

Tank-swing or switching is a part of the batch-cut mechanism where an operator switches delivery tanks (by manipulating cut valves) at the Denver delivery station and cuts a batch ticket. Each batch of petroleum products to be moved through the RM pipeline is assigned a unique BATCH ID during batch scheduling. Batch ID contains information pertaining to a specific batch such as; product information, origin and destination locations (transfer route), flow rates, batch size and estimated time of lift and delivery at destination terminals to name a few [29]. Figure 3.5 describes how to interpret a Batch ID to identify the pipeline route and the type of crude being transferred by Suncor's Rocky Mountain Pipeline. [29].

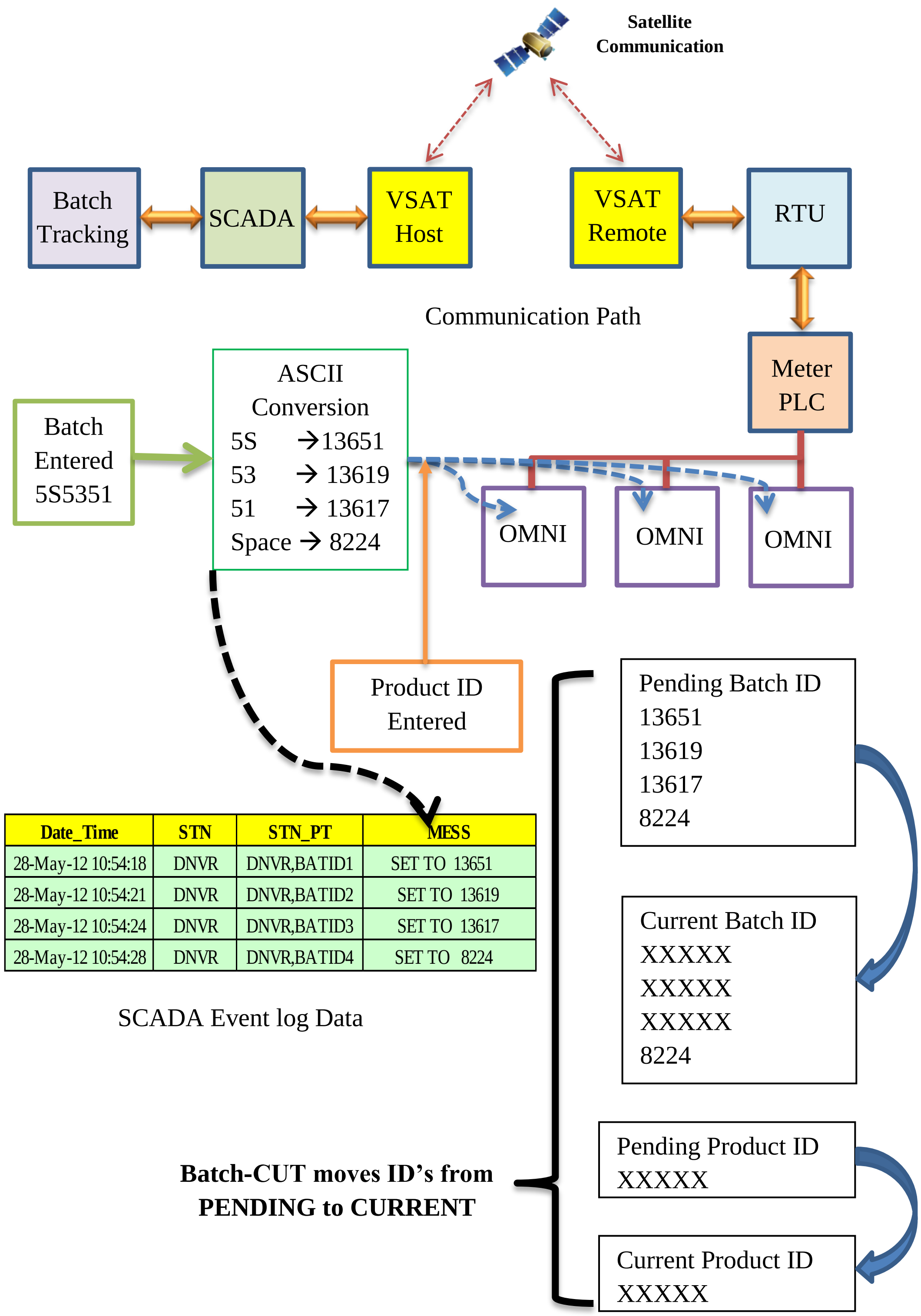


Batch ID Series	Line	P/L
0	0	Sinclair
1000	1	Belle Fourche
2000	2	Pacific Anchutz
3000	3	Platte
4000	4	Butte
5000	5	RM05 Mainline
6000/7000	6	RM06 Denver
8000/9000	9	RM09 Centennial

Product	Description	Grade
SWT	Sweet	S
ASP	Asphalt	A
SYN	Synthetic	Y

Figure 3.5: Interpreting a "Batch-ID" sequence in tank-swing event logs

When a product is moved through the pipeline, the SCADA host system collects metered batch volume information from field meters through a process called ticketing. A delivery of the scheduled batch volume to a delivery tank or shipper nominated meter receipt is recorded in the form of "a ticket" [30]. Ticketing at Cheyenne station is initiated by entering Batch ID (and Product ID) into the SCADA host system during batch scheduling. A Batch ID (like the one shown in Figure 3.5) is first converted into ASCII characters as shown in Figure 3.6. The SCADA host then loads Batch IDs into OMNI flow computers at the Denver receiving station.



Date_Time	STN	STN_PT	MESS
28-May-12 10:54:18	DNVR	DNVR,BATID1	SET TO 13651
28-May-12 10:54:21	DNVR	DNVR,BATID2	SET TO 13619
28-May-12 10:54:24	DNVR	DNVR,BATID3	SET TO 13617
28-May-12 10:54:28	DNVR	DNVR,BATID4	SET TO 8224

Figure 3.6: A Batch ID initiation process

OMNI flow computers are programmed with batch setup information in the batch stack [31]. A batch-cut moves pending IDs in batch stack to current IDs as shown in Figure 3.6. When a current batch ends, the next batch will commence immediately.

The batch-cut procedure at the Denver station ensures that all Cheyenne and Fort Lupton barrels are correctly ticketed to shipper nominated tanks at the Denver receiving station. All the pending batch tracking data is imported into OMNI computers by issuing Denver "Batch Launch" command.

Figure 3.7 shows the schematic of delivery tanks at the Denver receiving station. Valves corresponding to "Valve V07, V08, V09, and V10" are tank-swing valves or cut valves. According to Figure 3.7, Tanks 776 and 778 are now being used to store Synthetic Asphalt Blend crude. Storage Tank 58 is used to store Synthetic Asphalt Blend and Butte Sweet products depending on the shippers requirement. Tank 775 is a Sweet crude storage tank and should be kept very pure by restricting delivery of any high TAN products into it. High TAN products are corrosive and have a total acid number greater than 1.0 mg KOH/g [32]. East refinery Tanks 6, 26, and 38 are used for storing Sweet crude.

A gravitometer is used at Meter M1 (metering station at the Denver receiving terminal) to continuously monitor the specific gravity of the pipeline product stream entering the Denver station. A batch volume accumulator at Meter M1 continuously records the volume of product being transferred through the pipeline as shown in Figure 3.7.

- To ensure that refinery nominated barrels are correctly delivered to the tanks, alarms limits on batch volume accumulator in Meter 1 are set to:

Alarm limit on M-1 Batch Volume Accumulator = **Linefill (350 bbl) + Scheduled Delivery Volume**

- Similarly, to ensure the operators would accurately identify the batch interface and make the batch-cut, alarm limits on gravitometer is set according to the specific gravity limits for various types of crudes being transferred by Suncor Energy as shown in Table 3.1 [29].

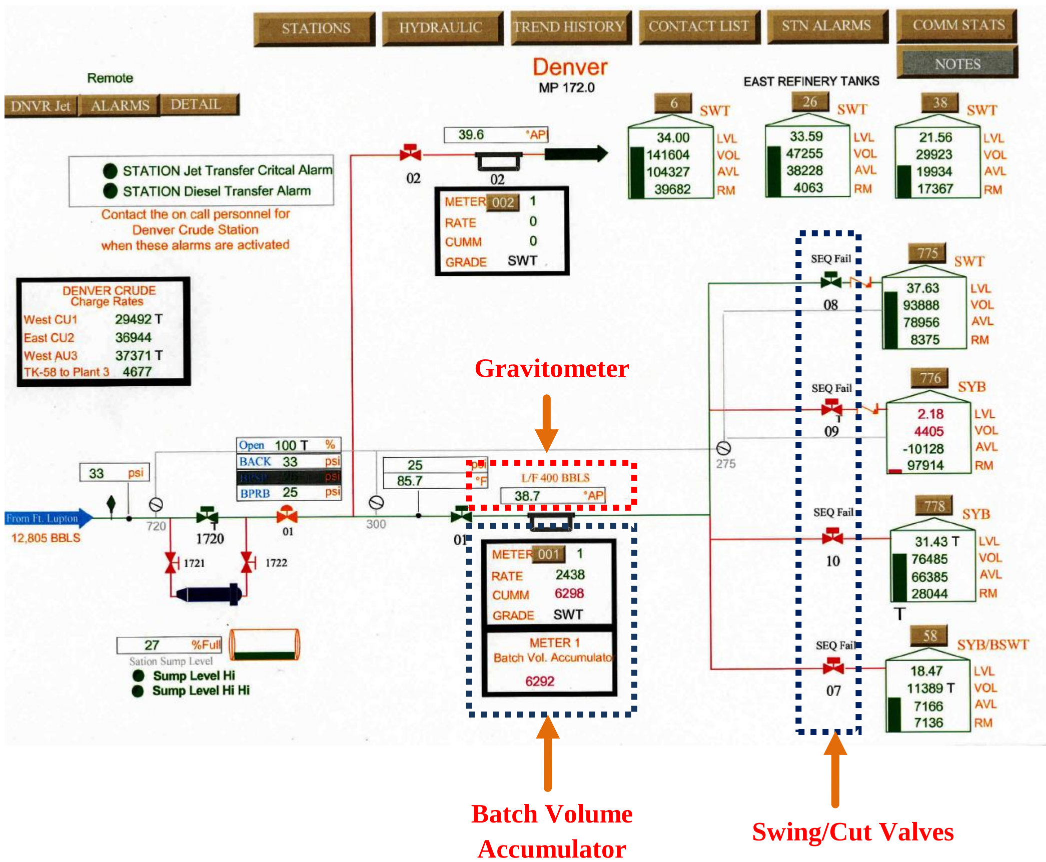


Figure 3.7: Tank terminals of RM – 06 pipeline at Denver station

SYN	30 API	26 API	30 API	X
SWT	Near last change 37 API	30 API	X	30 API
SYB	30 API	X	30 API	26 API
BSWT	X	30 API	Near last change 37 API	30 API
	BSWT	**SYB**	**SWT**	**SYN**

Table 3.1: Batch-cut matrix for different products of crude oil

Let us now consider a situation where SYB to BSWT interfaced batches are approaching the Denver receiving station; here SYB crude is the current leading batch and BSWT crude is the following batch (SWT crude follows BSWT). Also a scheduled volume of 500 bbl of BSWT to Tank 776 is set in batch scheduling.

- M-1 gravitometer alarm limits are set to 28, 29, 30 API corresponding to HI, HIHI, HIHIHI limits of specific gravity values. 30 API is the cut point from SYB to BSWT.
- Pop M-1 at 30 API from SYB to BSWT (& reset M-1 gravity limits to 35, 36, 37 API for next incoming SWT batch).
- When 350 bbl linefill is reached, swing the valves (close V07 and open V09) and initiate ticket cut command (which cuts current SYB batch from batch tracking data).
- BSWT batch delivery commences delivery to Tank 776.
- For BSWT batch, alarm limits on M-1 Batch volume accumulator alarm limit are set to: (500 +350) bbl = 850bbl. This accumulator limit alarms are used to make tank swings.

PLCs at the Denver station automatically closes the V07 valve when the V09 valve is opened, so the operator sends one command from SPCC. This event is triggered from alarms on the batch volume accumulator and/or gravitometer.

3.4 Event Logs for Tank-swing Process

In the following sections, we analyze SCADA event logs associated with tank-swing process at the Denver receiving station. The SCADA host system archives all commands issued at SPCC (Suncor Pipeline Control Centre) and all the field data points associated with tank-swing process such as Batch ID initiation, tank-swing warnings, gravity alarms, valve manipulation, and ticket cut commands. The following sections give a detailed description of how to interpret SCADA event logs and how to develop tank-swing workflows from the event logs.

3.4.1 Interpreting SCADA Event Log Data

Suncor Energy Pipeline Limited is equipped with SCADA host systems, flow computers and Remote Terminal Units (RTUs) supplied by Willowglen Systems (version v5.0). Event log data and SCADA operator schematics discussed in this work are obtained from Willowglen's SCADA host systems located at SPCC (Suncor Pipeline Control Centre), Sherwood Park, Edmonton. Recently, Suncor upgraded their monitoring and control host systems with a newer version of SCADA solutions engineered by Survalent Technology.

Data types archived in a SCADA system could be broadly classified into four categories namely discrete, analogue, internal, and parameter types [33]. Discrete type data includes binary inputs from field instrumentation such as alarms (low or high) or status messages corresponding to a valve "opening or closing". Binary output commands from the SCADA host system to start or shut down a pump also constitute discrete data points. Suncor's host system supports more than two states to represent an alarm. For instance, alarms are configured with three levels of priority namely "LO, LOLO, LOLOLO" and "HI, HIHI, HIHIHI" to indicate a field instrument low and high volume alarms respectively. Similarly, the status of a control valve is configured with three states namely "Open, Travel, and Closed" as shown in Table 3.2.

TIME_STAMP	STN	EVENT_MESSAGE	ALARM_MSG
28-05-2012 10:55:15	DNVR	DNVR TK SWING WARNING	HIHIHI 910.0 9
28-05-2012 10:55:15	DNVR	DNVR TK SWING WARNING	HIHI 862.0 850
28-05-2012 10:55:15	DNVR	DNVR TK SWING WARNING	HI 803.0 800.0
28-05-2012 11:12:04	DNVR	VALVE 08 TANK 775 STAT	OPEN
28-05-2012 11:12:04	DNVR	VALVE 10 TANK 778 STAT	TRAVEL
28-05-2012 11:12:37	DNVR	VALVE 10 TANK 778 STAT	CLOSED

Table 3.2: Tri-state alarms and status messages from SCADA event logs

A SCADA master continuously monitors field discrete data points to update operator screens and generate alarms. The event logs are extracted from PI systems of SPCC and are essentially an electronic version of what operators would see on their screens in the control room. For the purpose of developing a workflow, we consider only discrete data types which include event and alarm messages generated from the field and commands issued by the host system. Figure 3.8 shows a snapshot of raw event log spreadsheet obtained from SPCC.

Date_Time	STN	STN_PT	MESS	ALM
28-May-12 07:59:57	GATP	GATP,V005O	PIG LNCH KICK VLV POS	CLOSE FROM SPCC-
28-May-12 08:00:00	GATP	GATP,V010O	PIG LNCH ISO VLV POS	CLOSE FROM SPCC-
28-May-12 08:00:11	GATP	GATP,V005O	PIG LNCH KICK VLV POS	TRAVEL
28-May-12 08:00:11	GATP	GATP,V010O	PIG LNCH ISO VLV POS	TRAVEL
28-May-12 08:00:30	DNVR	DNVR,M1ACC1	DNVR TK SWING WARNING	HI 803.0 800.0
28-May-12 08:01:18	GATP	GATP,V005O	PIG LNCH KICK VLV POS	CLOSED
28-May-12 08:01:29	GATP	GATP,V010O	PIG LNCH ISO VLV POS	CLOSED
28-May-12 08:01:45	DNVR	DNVR,M1ACC1	DNVR TK SWING WARNING	HIHI 862.0 850
28-May-12 08:02:45	DNVR	DNVR,M1ACC1	DNVR TK SWING WARNING	HIHIHI 910.0 9
28-May-12 08:03:15	SCOP	SCOP-UBDVOLI	.CNTDWN.FINISHED	
28-May-12 08:03:15	OSPL	OSPL,CMD80	CMD80 FAIL SCT COUNTDOWN	ALARM
28-May-12 08:03:15	SEQ0	SEQ080FINISH	FROM SPCC-CONSOLE::SCAD	
28-May-12 08:06:45	GNSY	GNSY,TFLWML	MAINLINE METER IN/OUT	LO -123.0 -60.0
28-May-12 08:07:35	AULT	AULT,FRM4M8	SET TO 2741(2804)	FROM SEQ12
28-May-12 08:07:37	FLUP	FLUP,FRCHP	SET TO 1387(1368)	FROM SEQ12
28-May-12 08:08:48	OSPL	OSPL:BATCH	380 : 60 MIN WRN TO GRAP	
28-May-12 08:09:09	GRAP	GRAP,SUCLVP	SET TO 650(550)	FROM SPCC-
28-May-12 08:15:21	ATT2	ATT2,V1212C	203-V-1212 TK8-BP2	TRAVEL
28-May-12 08:15:32	ATT2	ATT2,V1216C	BP2 INLT VLV TK-9 STAT	TRAVEL
28-May-12 08:17:38	AULT	AULT,FRM4M8	SET TO 2692(2741)	FROM SEQ12
28-May-12 08:17:41	FLUP	FLUP,FRCHP	SET TO 1349(1387)	FROM SEQ12
28-May-12 08:18:07	ATT2	ATT2,V1212C	203-V-1212 TK8-BP2	OPEN
28-May-12 08:18:17	DPMP	DPMP,F5SPD2	FAN05 SPD INDICATOR 2	ON
28-May-12 08:18:30	ATT2	ATT2,V1216C	BP2 INLT VLV TK-9 STAT	CLOSED
28-May-12 08:23:00	MTSB	MTSB,PLCLM	300-PLC-5 LOGIC MISMATCH	ALARM
28-May-12 08:25:53	GNSY	GNSY,M10ACC	METER 10 MAINLINE PUMP	HI 3801.0 3800.0
28-May-12 08:26:10	TARP	TARP,TIMESP	SET TO 0(0)	FROM SEQ04
28-May-12 08:26:23	WLMR	WLMR,TIMESP	SET TO 0(0)	FROM SEQ04
28-May-12 08:26:28	SCMP	SCMP,TIMESP	SET TO 0(0)	FROM SEQ04
28-May-12 08:26:44	GIBS	GIBS,TIMESP	SET TO 0(0)	FROM SEQ04
28-May-12 08:26:48	ENBR	ENBR,TIMESP	SET TO 0(0)	FROM SEQ04
28-May-12 08:26:56	GATP	GATP,TIMESP	SET TO 0(0)	FROM SEQ04

Time Stamp **Station** **Tag Name** **Event Message** **Alarm Status**

Figure 3.8 : SCADA event logs from the Suncor Pipeline Control Centre

This event log spreadsheet has five columns and is very similar to the "events and alarms log screen" as seen by an operator in the control room. These five columns are the following:

1. **Date_Time:**
   It represents the time-stamped information of discrete points which includes alarms and event messages and SCADA host issued commands.
2. **STN:**
   This column represents zones or consoles or station points situated along the pipeline. This attribute of the event log helps operators to differentiate events and commands from different consoles. Table 3.3 shows a list of text data entries that appear in SCADA event logs corresponding to different geographical consoles or stations of RM pipeline operations.

United States Consoles	MTT Console	Canadian Consoles
AULT (Ault)		GATP
BRAC (Bracewell)	MTSB (MTT Suncor Blend Console)	HANP
CHYN (Cheyenne)		MARP
DNVR (Denver)		WANP
FLUP (Fort Lupton)		GRAP
GNSY (Guernsey)		NEWP
GSHN		SCOP
LTBR		WLMR
SLAT		(Oil Sands Pipeline)

Table 3.3: Different consoles from Suncor pipeline operations

3. **STN_PT:**
   Station points indicate the tag name of the instruments, equipment and operator interfaces associated with pipeline operations. Points include

OMNI flow computers, leak detection system, remote terminal units (RTU), station pumps, valves, communication modules, push buttons, measuring instruments such as flow transmitters, pressure transmitters, volume accumulators, gravity meters, temperature transmitters, and site programmable logic controllers (PLC).

4. **MESS:**

   This column contains all the commands issued by the SCADA host system, operator set-point adjustments, event and alarm messages generated and acknowledged by the host system (operator). This is an important attribute of event log database which is very instrumental in tracking operational problems and for generating operational reports such as alarm summary, day-in day-out shift summary, product movement and tracking, communication, and system availability reports. This column of the database also helps operators to create basic "host command count" charts and "work-load comparision" charts among the consoles.

5. **ALM_State (or CMD State)**

   "ALM_State" gives status message of alarms and their limits associated with various field discrete points. These status messages convey mission critical information such as communication failure, RTU failure, leak alarm and its location in the pipeline, low and high limits of a field measurement. "ALM_State" is often referred to as "CMD_State" (command state) which provides information on status of commands issued by an operator, for instance, a pump stop and start command.

The SCADA host is interfaced with the OSIsoft PI System which stores all alarm and event messages, their status points and host commands as historical data base. A query-driven user interface allows operators to retrieve alarm - command counts and their pre-configured reports. Command load is a measure of operator command counts per console. Alarm loading is a measure of stimulus inputs of the operator. Command counts and alarm loading are important metrics which place cognitive demands on control centre operators. Table 3.4 shows commands per console by week during different months of a year[4].

CONSOLE	17-Jan-11	18-Apr-11	15-Aug-11	14-Nov-11	23-Apr-12	25-Jun-12
CAN_Console	2449	1747	2019	2313	2165	1799
MTT_Console	639	329	526	388	603	440
US_Console(Z8)	2408	505	3071	1954	2380	2485
Overall Total	5496	2581	5616	4655	5148	4724

Table 3.4: Commands count per console by week

Figure 3.9 shows a historical command count chart for a period of 18 months which identifies the US and Canadian consoles as the busiest ones when compared to MTT console (a sudden dip of command counts for the US console during 18th April 2011 is due to the outage of the US pipelines during that period).

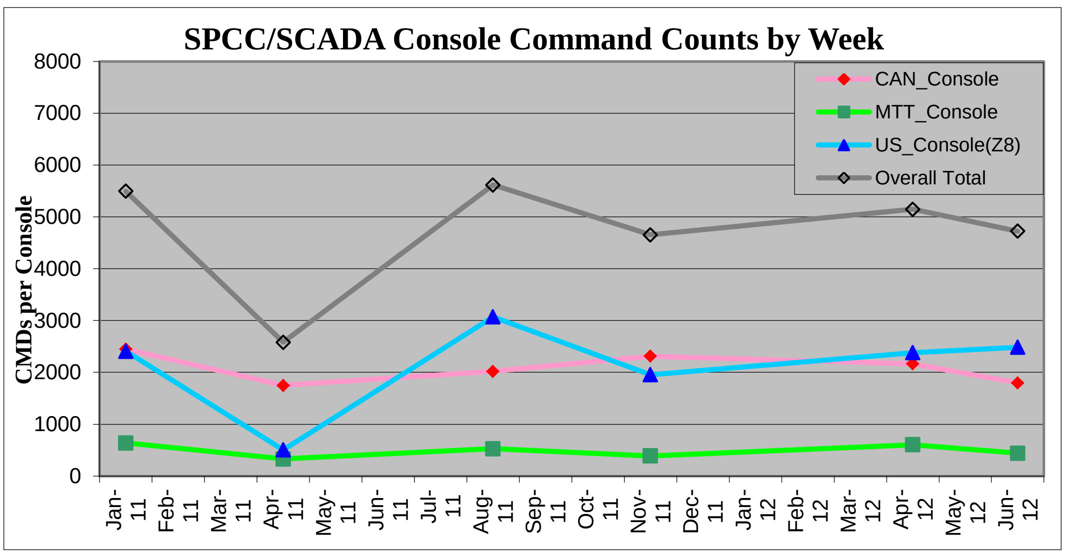


Figure 3.9: Trends of commands count per console by week[2]

Operators' communication over the phone, shift-hand-over and other operational tasks are critical for pipeline operations, however these are not recorded in the database.

[4] Colin Lunty, e-mail message to author, July 5, 2012

Suncor uses command and alarm count charts as a measure to survey the operator's performance on specific tasks.

[Note: Command load and alarms counts at the US console is trended along with the estimates of phone time, shift-hand-over time to provide an annual assessment for the PHMSA (Pipeline and Hazardous Materials Safety Administration, U.S Department of Transportation)[5]].

3.4.2 Event Log Pre-processing Steps for Tank-swing Workflows

This section analyzes the raw event log file obtained from PI systems. The pre-processing steps involve filtering the event log to retain only those events, which are relevant to the tank-swing process at the Denver receiving station or console. This also ensures that the event log file is compatible with Disco [10], which can then be used to generate the tank-swing workflows automatically.

The pre-processing procedure involves following steps:

1. **Filtering raw event log file**

   The event logs are parsed up by zone or console. One of the main challenges in filtering SCADA events is the lack of operator logon names (authenticity) associated with commands or events record. As events and commands from all consoles are recorded chronologically in the host system, lack of unique instrument tag names further toughen the process of event log analysis.

   In order to extract events and alarm messages corresponding to the tank-swing process, all the events corresponding to Denver station are filtered first by setting the filter criterion to:

   Filter → STN : DNVR (as shown in Figure 3.10)

[5] Colin Lunty, e-mail message to author, July 5, 2012

Date_Time	STN	STN_PT	MESS	ALM
28-May-12 07:59:57	GATP	GATP,V005O	PIG LNCH KICK VLV POS	CLOSE FROM SPCC-
28-May-12 08:00:00	GATP	GATP,V010O	PIG LNCH ISO VLV POS	CLOSE FROM SPCC-
28-May-12 08:00:11	GATP	GATP,V005O	PIG LNCH KICK VLV POS	TRAVEL
28-May-12 08:00:11	GATP	GATP,V010O	PIG LNCH ISO VLV POS	TRAVEL
28-May-12 08:00:30	DNVR	DNVR,M1ACC1	DNVR TK SWING WARNING	HI 803.0 800.0
28-May-12 08:01:18	GATP	GATP,V005O	PIG LNCH KICK VLV POS	CLOSED
28-May-12 08:01:29	GATP	GATP,V010O	PIG LNCH ISO VLV POS	CLOSED
28-May-12 08:01:45	DNVR	DNVR,M1ACC1	DNVR TK SWING WARNING	HIHI 862.0 850
28-May-12 08:02:45	DNVR	DNVR,M1ACC1	DNVR TK SWING WARNING	HIHIHI 910.0 9
28-May-12 08:03:15	SCOP	SCOP-UBDVOLI	.CNTDWN.FINISHED	
28-May-12 08:03:15	OSPL	OSPL,CMD80	CMD80 FAIL SCT COUNTDOWN	ALARM

Filter → STN : DNVR

Date_Time	STN	STN_PT	MESS	ALM
28-May-12 08:00:30	DNVR	DNVR,M1ACC1	DNVR TK SWING WARNING	HI 803.0 800.0
28-May-12 08:01:45	DNVR	DNVR,M1ACC1	DNVR TK SWING WARNING	HIHI 862.0 850
28-May-12 08:02:45	DNVR	DNVR,M1ACC1	DNVR TK SWING WARNING	HIHIHI 910.0 9
28-May-12 08:31:45	DNVR	DNVR,TOTFLW	CHY/FT.LUPT TO DNVR FLOW	LO -360.5 -300.0
28-May-12 08:33:15	DNVR	DNVR,TOTFLW	CHY/FT.LUPT TO DNVR FLOW	LO -360.5 -300.0
28-May-12 08:43:45	DNVR	DNVR,TOTFLW	CHY/FT.LUPT TO DNVR FLOW	LO -301.0 -300.0
28-May-12 08:44:00	DNVR	DNVR,TOTFLW	CHY/FT.LUPT TO DNVR FLOW	LO -301.0 -300.0
28-May-12 10:52:50	DNVR	DNVR,M1GRV	IN STN GRAV MTR1 SUNCOR	HIHI 28.0 28
28-May-12 10:53:37	DNVR	DNVR,M1GRV	IN STN GRAV MTR1 SUNCOR	HIHIHI 29.1

Figure 3.10: Filtering raw event logs based on "Denver station"

2. **Standardize date and time format**

The date and timestamps of the SCADA event logs are re-formatted to a standardized representation acceptable for use with workflow mining tools such as Disco as shown in Figure 3.11. Disco views "date and timestamp" values with *year*, *month*, *day*, *hour*, *minute* attribute as an integer value and *second* attribute as a decimal value. This is also an official definition of dateTime format in XML Schema [34].

Date_Time	STN	STN_PT	MESS	ALM
28-May-12 10:54:14	DNVR	DNVR,CMD142	CMD142.OMNI.BTCH.ID.SEQ.	ON

↓

Date_Time	STN	STN_PT	MESS	ALM
28-05-2012 10:54:14	DNVR	DNVR,CMD142	CMD142.OMNI.BTCH.ID.SEQ.	ON

Figure 3.11: Standardizing date and time format

3. Split STN_PT into "Station" and "Tag"

STN_PT column of the raw event log file is split into two different entities namely "Station_Point" and "Tag",which simplifies the task of identifying individual tags of equipments for each console as shown in Figure 3.12.

Date_Time	STN	STN_PT	MESS	ALM
28-May-12 11:04:05	DNVR	DNVR,M1ACC	METER 1 SUNCOR RECEIVE	HIHIHI 456.0 4

STD_TIME_STAMP	STATION	STATION_POINT	TAG	EVENT_MESSAGE	ALARM_MESSAGE
28-05-2012 11:04:05	DNVR	DNVR,M1ACC	M1ACC	METER 1 SUNCOR RECEIVE	HIHIHI 456.0 4

Figure 3.12: Separating stations and tag names

4. Normalize BATCH IDs

Operators enter BATCH IDs into OMNI flow computers in a sequence. BATCH ID strings are sent automatically from the SCADA system. The procedure to decipher a BATCH ID is discussed in Section 3.3.1. Figure 3.13 shows two sets of BATCH ID sequence fed into OMNI computers for two different batches of products for delivery at Denver station.

Date_Time	STN	STN_PT	MESS	ALM
28-May-12 10:54:18	DNVR	DNVR,BATID1	SET TO 13907(13890)	FROM SEQ14
28-May-12 10:54:21	DNVR	DNVR,BATID2	SET TO 13881(13881)	FROM SEQ14
28-May-12 10:54:24	DNVR	DNVR,BATID3	SET TO 13877(13876)	FROM SEQ14
28-May-12 10:54:28	DNVR	DNVR,BATID4	SET TO 8224(8224)	FROM SEQ14
28-May-12 14:13:56	DNVR	DNVR,BATID1	SET TO 13890(13907)	FROM SEQ14
28-May-12 14:13:59	DNVR	DNVR,BATID2	SET TO 13881(13881)	FROM SEQ14
28-May-12 14:14:02	DNVR	DNVR,BATID3	SET TO 13878(13877)	FROM SEQ14
28-May-12 14:14:05	DNVR	DNVR,BATID4	SET TO 8224(8224)	FROM SEQ14

(SET TO "NUMERICAL VALUE" → SET TO "A" BARRELS)

28-May-12 10:54:18	DNVR	DNVR,BATID1	SET TO A BARRELS	FROM SEQ14
28-May-12 10:54:21	DNVR	DNVR,BATID2	SET TO B BARRELS	FROM SEQ14
28-May-12 10:54:24	DNVR	DNVR,BATID3	SET TO C BARRELS	FROM SEQ14
28-May-12 10:54:28	DNVR	DNVR,BATID4	SET TO D BARRELS	FROM SEQ14

Figure 3.13: Normalizing BATCH IDs registered at Denver station

It is interesting to observe that entries for “BATCH ID1 to BATCH ID4” are always registered together. As these BATCH IDs are unique, workflow mining tools consider each string of BATCH ID as a separate event and creates several blocks of tasks within a workflow.

To make workflow representation simpler, numeric strings of BATCH IDs are replaced with generic text named entries (like “SET TO A BARRELS”). It makes sense, since we are interested in operator actions that describe the process of keying in new BATCH IDs into the flow computer instead of looking at individual numerical values.

5. **Concatenate event and alarm messages**

Concatenating event messages with alarm messages is a critical step in data pre-processing. Figure 3.14 shows a segment of the event logs corresponding to manipulation of control valves V08 and V10 located in Denver receiving station (refer to Figure 3.7). The first three event messages look alike “VALVE 10 TANK 778 STAT” as they do not convey any further information on valve status. For instance, when an operator issue a command to open the control valve V10 to pass products to Tank 778, SCADA stores this action in the form of “VALVE 10 TANK 778 STAT” in event message column (MESS) and “OPEN FROM SPCC” in ALM column. When that valve is actually opened, it is stored as “VALVE 10 TANK 778 STAT” in event message column (MESS) and “OPEN” in ALM column. Workflow mining tools like Disco and ProM [10] are very sensitive to “text information” and they consider similar text messages as repetition of a same action. When workflow actions (sequence of tasks) are developed solely on these identical text messages, they fail to differentiate a “valve open” command message from a “valve position” status message. Thus, the sequence of tasks which are critical in the batch-cut process are not captured in the tank-swing workflow.

Event messages look alike

Date_Time	STN	STN_PT	MESS	ALM
28-May-12 14:11:55	DNVR	DNVR,V10O	VALVE 10 TANK 778 STAT	OPEN FROM SPCC
28-May-12 14:12:15	DNVR	DNVR,V10O	VALVE 10 TANK 778 STAT	TRAVEL
28-May-12 14:12:49	DNVR	DNVR,V10O	VALVE 10 TANK 778 STAT	OPEN
28-May-12 14:12:53	DNVR	DNVR,V08O	VALVE 08 TANK 775 STAT	TRAVEL
28-May-12 14:13:31	DNVR	DNVR,V08O	VALVE 08 TANK 775 STAT	CLOSED

"Valve 08" status message

"Valve 10" status message

"Open valve 10" command from SPCC

Figure 3.14: Differentiation of commands from status messages

Date_Time	STN	STN_PT	CUSTOM_MESSAGE
28-May-12 14:11:55	DNVR	DNVR,V10O	VALVE 10 TANK 778 STAT_OPEN_SPCC
28-May-12 14:12:15	DNVR	DNVR,V10O	VALVE 10 TANK 778 STAT_TRAVEL
28-May-12 14:12:49	DNVR	DNVR,V10O	VALVE 10 TANK 778 STAT_OPEN
28-May-12 14:12:53	DNVR	DNVR,V08O	VALVE 08 TANK 775 STAT_TRAVEL
28-May-12 14:13:31	DNVR	DNVR,V08O	VALVE 08 TANK 775 STAT_CLOSED

Unique event message string

Figure 3.15: Custom message by concatenating event and alarm message

It is, therefore, necessary to create unique text messages from SCADA event logs, which will enable workflow mining tools to capture all critical tasks in the workflow, which occur on a single piece of equipment. One way of creating such unique messages is by concatenating the text character strings (CUSTOM_MESSAGE) of "event and alarm messages" from end-to-end, for each and every event archived in the SCADA history. An example of such a concatenated unique message is shown in Figure 3.15.

6. Case Identification

Case identification is an important attribute required for Disco to automatically extract workflows from event logs. Section 2.2.1 provides a detailed explanation of case identification and its significance in extracting sequences of tasks from event messages using Disco. In order to assign case IDs to the event logs, a workflow "start event" and "end event" are first identified as listed in Table 3.5. All the mandatory events that constitute a workflow (golden path of operation) are identified and assigned case IDs manually. Figure 3.16 lists all the mandatory events in a tank-swing workflow:

Workflow Start & End Events	TAG	EVENT_MESSAGE	ALARM_MESSAGE
START	CMD142	CMD142.OMNI.BTCH.ID.SEQ.	ON
END	BLNCH1	MTR#1 TICKET CUT CMD	YES

Table 3.5: Tank-swing workflow start and end events

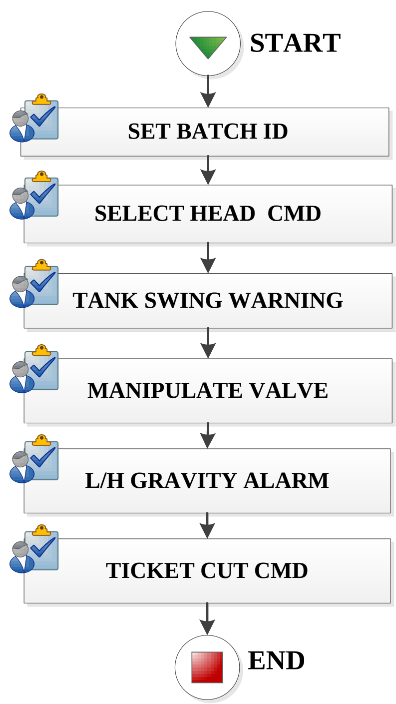


Figure 3.16: Mandatory events in a tank-swing workflow

Standardized Date & time stamps

Station/Console

Tag name

Custom Event Message

Case IDs Assigned to events

STD_TIME_STAMP	STATION	TAG	CUSTOM_MESSAGE	CASE_ID
28-05-2012 10:54:14	DNVR	CMD142	CMD142.OMNI.BTCH.ID.SEQ.	1
28-05-2012 10:54:18	DNVR	BATID1	SET TO A BARRELS	1
28-05-2012 10:54:21	DNVR	BATID2	SET TO B BARRELS	1
28-05-2012 10:54:24	DNVR	BATID3	SET TO C BARRELS	1
28-05-2012 10:54:28	DNVR	BATID4	SET TO D BARRELS	1
28-05-2012 10:54:31	DNVR	MTRID	SET TO 1(0)	1
28-05-2012 10:54:35	DNVR	M1PRID	SET TO 1(4)	1
28-05-2012 10:54:55	DNVR	M1HCMD	MTR 1 HEAD SELECT CMD	1
28-05-2012 10:55:01	DNVR	BATID1	SET TO A BARRELS	1
28-05-2012 10:55:04	DNVR	BATID2	SET TO B BARRELS	1
28-05-2012 10:55:05	DNVR	M1BTST	MTR1.BTCH/COMPAR STATUS	1
28-05-2012 10:55:08	DNVR	BATID3	SET TO C BARRELS	1
28-05-2012 10:55:10	DNVR	M1ACC	METER 1 SUNCOR RECEIVE	1
28-05-2012 10:55:10	DNVR	M1ACC	METER 1 SUNCOR RECEIVE	1
28-05-2012 10:55:10	DNVR	M1ACC	METER 1 SUNCOR RECEIVE	1
28-05-2012 10:55:12	DNVR	BATID4	SET TO D BARRELS	1
28-05-2012 10:55:13	DNVR	M1HCMD	MTR 1 HEAD SELECT CMD	1
28-05-2012 10:55:13	DNVR	M1CH1	MTR 1 CONOCO HEAD BIT1	1
28-05-2012 10:55:13	DNVR	M1CH3	MTR 1 CONOCO HEAD BIT3	1
28-05-2012 10:55:15	DNVR	M1PRID	SET TO 1(1)	1
28-05-2012 10:55:15	DNVR	M1ACC1	DNVR TK SWING WARNING	1
28-05-2012 10:55:15	DNVR	M1ACC1	DNVR TK SWING WARNING	1
28-05-2012 10:55:15	DNVR	M1ACC1	DNVR TK SWING WARNING	1
28-05-2012 10:55:38	DNVR	BATID1	SET TO A BARRELS	1
28-05-2012 10:55:41	DNVR	BATID2	SET TO B BARRELS	1
28-05-2012 10:55:44	DNVR	BATID3	SET TO C BARRELS	1
28-05-2012 10:55:48	DNVR	BATID4	SET TO D BARRELS	1
28-05-2012 10:55:51	DNVR	M1PRID	SET TO 1(1)	1
28-05-2012 10:55:53	DNVR	M1HCMD	MTR 1 HEAD SELECT CMD	1
28-05-2012 10:56:13	DNVR	MTRID	SET TO 0(1)	1
28-05-2012 10:56:13	DNVR	BLNCH1	MTR#1 TICKET CUT CMD	1
28-05-2012 10:56:17	DNVR	BLNCH1	MTR#1 TICKET CUT CMD	1
28-05-2012 10:56:17	DNVR	CMD142	CMD142.OMNI.BTCH.ID.SEQ.	1
28-05-2012 10:56:17	DNVR	CMD142	CMD142.OMNI.BTCH.ID.SEQ.	1
28-05-2012 10:56:17	DNVR	BLNCH1	MTR#1 TICKET CUT CMD	1
28-05-2012 10:56:20	DNVR	CMD142	CMD142.OMNI.BTCH.ID.SEQ.	1
28-05-2012 11:02:03	DNVR	M1ACC	METER 1 SUNCOR RECEIVE	1
28-05-2012 11:03:04	DNVR	M1ACC	METER 1 SUNCOR RECEIVE	1
28-05-2012 11:04:05	DNVR	M1ACC	METER 1 SUNCOR RECEIVE	1
28-05-2012 11:04:31	DNVR	M1GRV	IN STN GRAV MTR1 SUNCOR	1
28-05-2012 11:04:31	DNVR	M1GRV	IN STN GRAV MTR1 SUNCOR	1
28-05-2012 11:05:07	DNVR	M1BTST	MTR1.BTCH/COMPAR STATUS	1
28-05-2012 11:11:00	DNVR	M1ACC1	DNVR TK SWING WARNING	1
28-05-2012 11:11:13	DNVR	V08O	/ALVE 08 TANK 775 STAT_OPEN_SPCC	1
28-05-2012 11:11:19	DNVR	V08O	VALVE 08 TANK 775 STAT_TRAVEL	1
28-05-2012 11:12:00	DNVR	M1ACC1	DNVR TK SWING WARNING	1
28-05-2012 11:12:04	DNVR	V08O	VALVE 08 TANK 775 STAT_OPEN	1
28-05-2012 11:12:04	DNVR	V10O	VALVE 10 TANK 778 STAT_TRAVEL	1
28-05-2012 11:12:37	DNVR	V10O	VALVE 10 TANK 778 STAT_CLOSED	1
28-05-2012 11:13:00	DNVR	M1ACC1	DNVR TK SWING WARNING	1
28-05-2012 12:31:15	DNVR	TOTFLW	CHY/FT.LUPT TO DNVR FLOW	1
28-05-2012 12:32:30	DNVR	TOTFLW	CHY/FT.LUPT TO DNVR FLOW	1
28-05-2012 14:11:19	DNVR	M1GRV	IN STN GRAV MTR1 SUNCOR	1
28-05-2012 14:11:49	DNVR	M1GRV	IN STN GRAV MTR1 SUNCOR	1
28-05-2012 14:11:55	DNVR	V10O	/ALVE 10 TANK 778 STAT_OPEN_SPCC	1
28-05-2012 14:12:15	DNVR	V10O	VALVE 10 TANK 778 STAT_TRAVEL	1
28-05-2012 14:12:49	DNVR	V10O	VALVE 10 TANK 778 STAT_OPEN	1
28-05-2012 14:12:53	DNVR	V08O	VALVE 08 TANK 775 STAT_TRAVEL	1
28-05-2012 14:13:31	DNVR	V08O	VALVE 08 TANK 775 STAT_CLOSED	1
28-05-2012 14:13:50	DNVR	BLNCH1	MTR#1 TICKET CUT CMD	1
28-05-2012 14:13:52	DNVR	BLNCH1	MTR#1 TICKET CUT CMD	1

Pre-processed event log data obtained by implementing steps 1 to 6 from Section 3.4.2. Case IDs are assigned to events corresponding to a complete “TASK” or “Tank-swing” Process

Figure 3.17: A complete task with case IDs assigned to events

A complete tank-swing process is considered a "task". So, each task starts with the event "CMD142.OMNI.BTCH.ID.SEQ." and ends with the event "MTR#1 TICKET CUT CMD". Each task contains all mandatory events associated with a tank-swing process and is assigned case IDs manually. All events or instances within a single task are assigned with same case IDs. Figure 3.17 shows a list of events related to a specific tank-swing process (task) which is identified as case ID "1".

3.4.3 Developing Tank-swing Workflows from Event Log Data

In this section, we will discuss how tank-swing workflows are automatically extracted from event logs using workflow-mining tools such as Disco (Fluxicon Process Laboratories) [10]. A step by step procedure to upload an event file and to extract workflows using Disco is discussed in Section 2.2.1. Event logs are pre-processed as illustrated in Section 3.4.2, before they can be uploaded into Disco.

We analyze SCADA event logs archived during a period of one week from 28 May 2012 to 04 June 2012. This one week's event logs spreadsheet has 3500 events archived only for the tank-swing process at the Denver receiving station. Out of 3500 events or instances, 5 tasks are identified and assigned case IDs which results in 300 events. A task is a complete tank-swing process that starts with the event "CMD142.OMNI.BTCH.ID.SEQ." and ends with the event "MTR#1 TICKET CUT CMD". A task also contains all the mandatory events associated with a tank-swing process which are highlighted in Figure 3.16. As SCADA event logs do not show unique tag names, it is hard to distinguish events from task to task during case identification.

"Date & Timestamp","Case ID", and "Event message" are the three main spreadsheet attributes required to generate a tank-swing workflow using Disco. A workflow extracted from SCADA event log data is very dense and hence for better visualization, tank-swing workflow is zoomed in on as three portions which are shown in Figures 3.18, 3.19 and, 3.20 respectively.

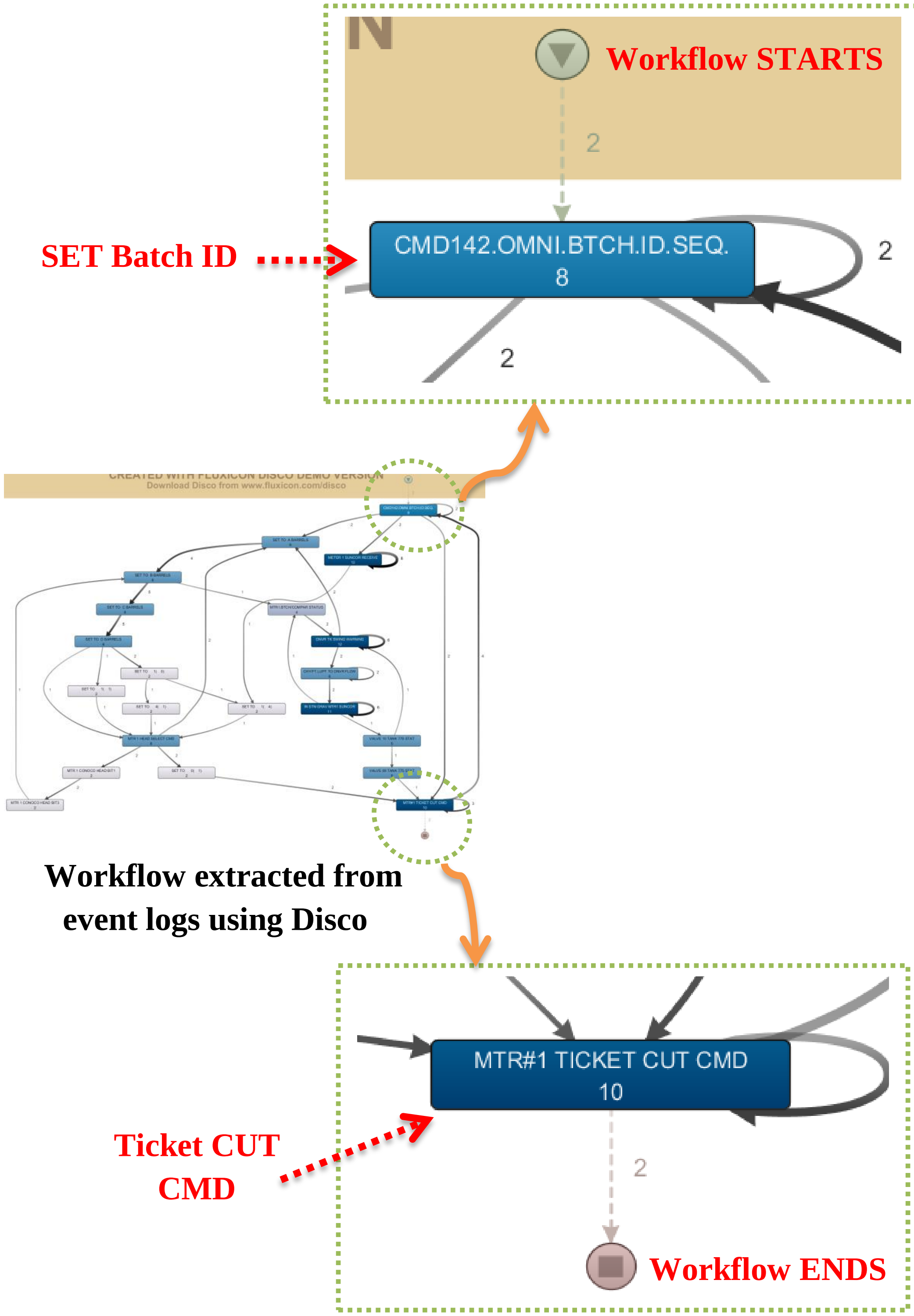


Figure 3.18: Part 1 : Critical events captured in a tank-swing workflow

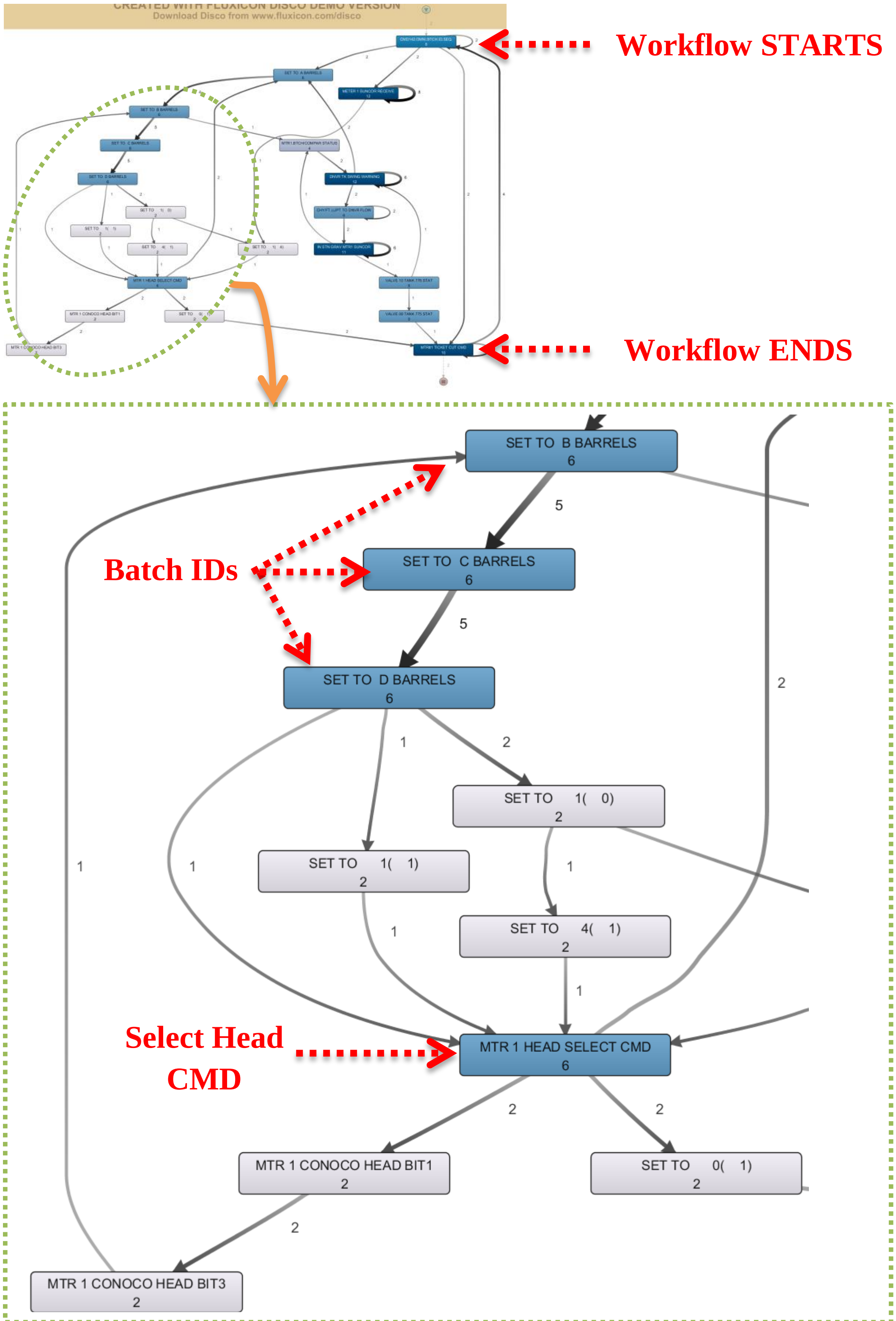


Figure 3.19: Part 2 : Critical events captured in a tank-swing workflow

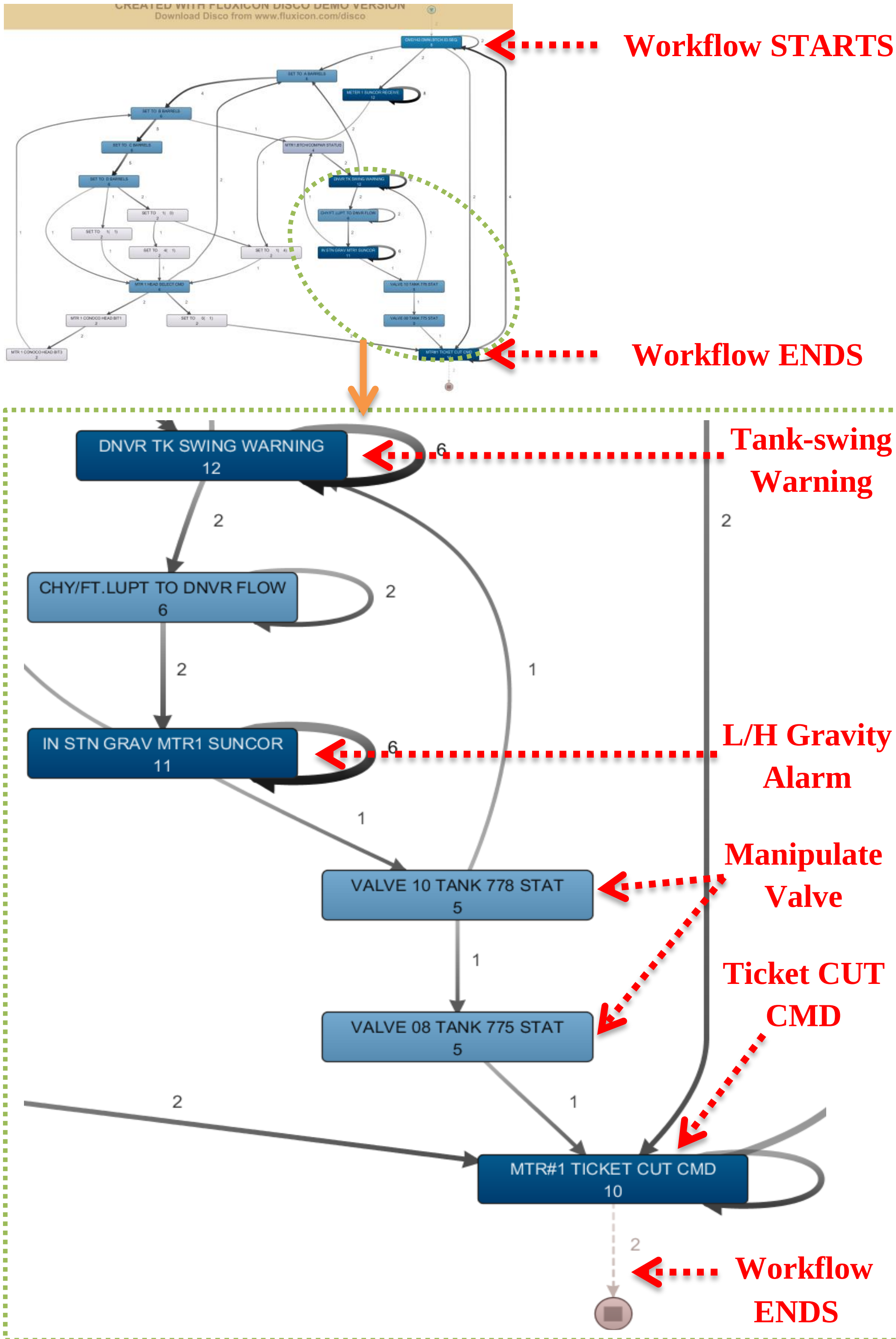


Figure 3.20: Part 3 :Critical events captured in a tank-swing workflow

Even the pre-processed event logs are not sanitized, because operators at SPCC may issue repeated commands to open or close a valve. Also, alarms from the batch volume accumulator and gravitometer are set to three levels of priority during a batch-cut process. Hence, within each task, the event log spreadsheet contains many repeated events such as valve "open and close" commands from SPCC, tank-swing warnings and gravity alarms to name a few. During pre-processing, these repeated operator actions or alarms cannot be ignored as they are very critical events for pipeline auditing purposes. When a valve is manipulated, an operator may change set-points for backpressure and flow rates in stages to maintain line pressure for custody transfer. These actions and events are not associated directly with a tank-swing process, however they are recorded chronologically as event messages. So, the event logs supplied to Disco have significant outliers and repetitive events. As Disco is very sensitive to text information, the workflows extracted using Disco are very dense, noisy and hard to understand.

The prominence of an event is indicated by the color-thickness of a block in the workflow diagram. The frequency of occurrence of an event is shown with a number within each block. So, higher the frequency number, the thicker the block would be. From Figure 3.20, the event "DNVR TK SWING WARNING" is shown with a thick colored block as that event appears 12 times within the event log. A directed arrow connecting two blocks indicates the path of actions or events during a tank-swing process. A number on the directed arrow indicates the number of times a particular transition occurred between those two blocks (actions or events) in the direction of the arrow. Tank-swing workflows extracted from event logs are instrumental in getting an operational insight into a tank-swing process. By analyzing current event logs of an expert operator, valuable operational knowledge is captured in the form of a workflow. However, intrinsic understanding of pipeline operations is essential to pre-process an event log and appreciate a workflow extracted using workflow mining tools.

3.4.4 Workflow Conformance

Workflows help to continuously monitor and compare operator-interactions and current-events with standard operating procedure(s) and identify operational deviations that could compromise process safety and efficiency. Workflow conformance also helps to identify operators who deviate from standard operating procedure(s) or who fail to follow a correct procedure while operating a process.

Very often a process deviates from its expected behavior for several reasons namely; changing operating conditions or constraints, operators failing to follow an established procedure, operational inconsistency during start-up and shutdowns, process and equipment failure to name a few. Process deviations are recognized in terms of alarms and non-conformance events that arise during an abnormal plant behavior. When a workflow is extracted from event logs, these abnormal alarms and non-conformance events appear as message blocks along with anticipated events. Hence, the process deviations are identified on-the-fly by comparing workflows with stipulated operating procedures.

A tank-swing process is a critical operational sequence in a batch-cut mechanism. Inconsistent operator actions while swinging cut valves hurt batch-cut mechanism that could negatively impact the batch quality of crude or petroleum products transported to a delivery terminal. Especially, when the pipeline operators are dealing with Ultra Low Sulfur Diesel (ULSD containing 15 ppm sulfur content) [35], operator errors are intolerable during a tank-swing process. Operational errors that occur due to human factors, procedure violations or faulty instrumentation also ruin shipper contracts by delivering inaccurate shipper-nominated batch volume to a storage tank.

Workflows capture operator actions and event messages of a tank-swing process in the form of message blocks as shown in Figures 3.18, 3.19, and 3.20. Figure 3.21 compares a tank-swing workflow extracted from ongoing events against an established operating practice (SOPs) for conformance checking. Incidents of

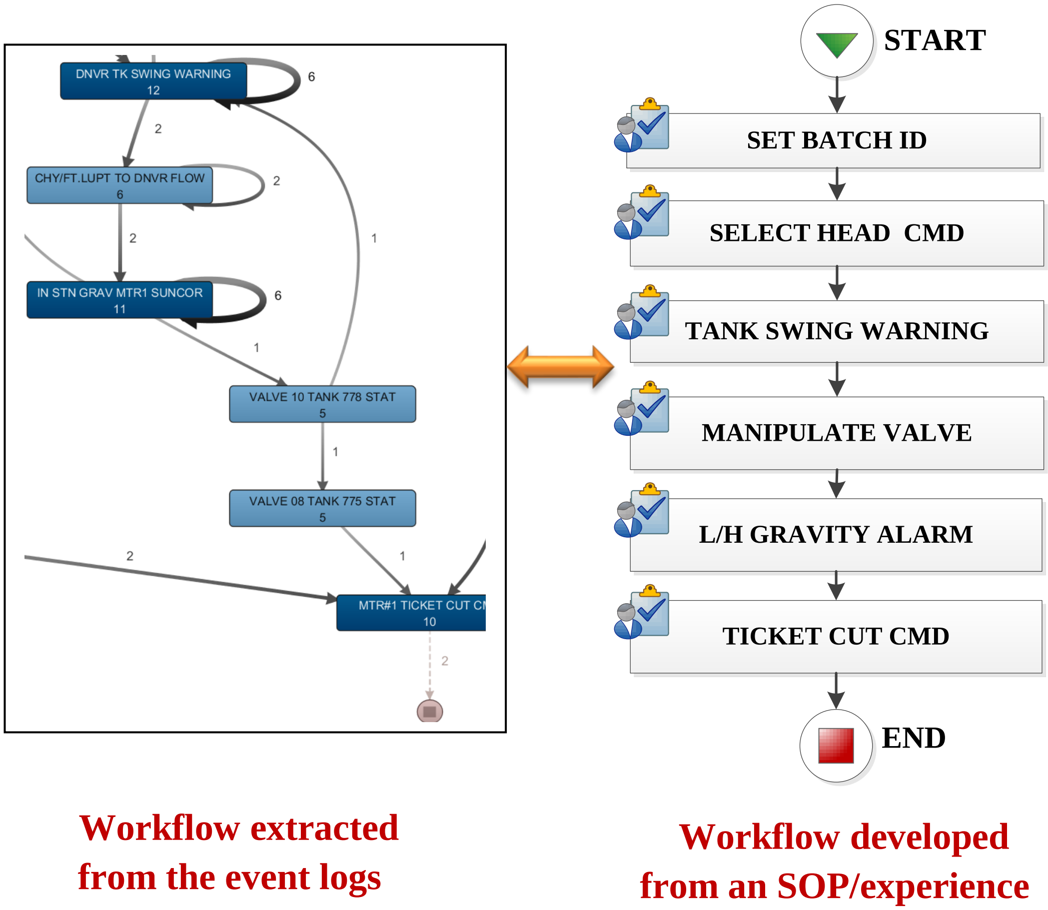


Figure 3.21: Workflow conformance monitoring

operator errors such as failing to acknowledge a tank-swing warning, manipulating an inappropriate valve, selecting a wrong tank are easily identified during a workflow conformance check.

By comparing current events with authorized operating practices, pipeline operators are able to identify problems associated with human factors, process, and instrumentation in a pipeline operation. Workflow conformance is an indicator of preventive maintenance and highlights the opportunity for process enhancement and optimization. As software-based workflows are auditable, workflow conformance also provides documented information on actions taken during an abnormal situation in order to meet the regulatory compliance.

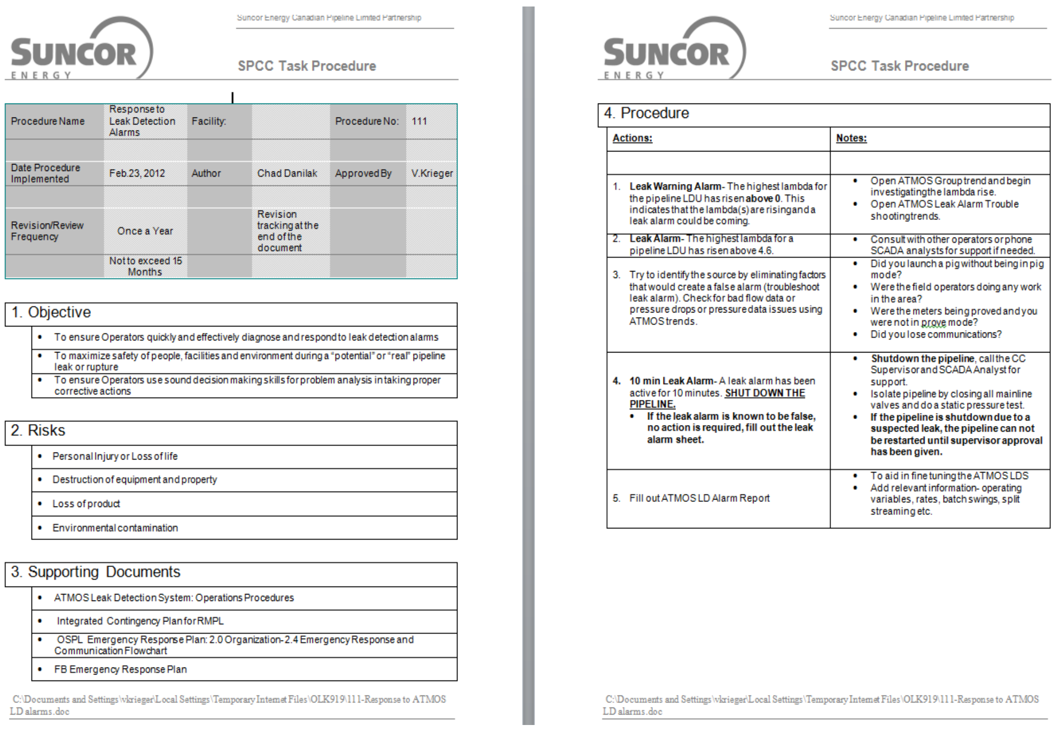
3.5 Developing Software-based Workflows for "Response to Leak Detection Alarms" Procedure using OWBE

Orchestrated workflows by Example (OWBE) platform helps users to create, publish, execute and improve self-service workflows catering to their changing operating environment. The following discussion examines how OWBE can turn a paper-based standard operating procedure of a pipeline emergency response task into a "live" orchestration of people, processes, and equipment. This section highlights the opportunity for implementing OWBE as a triggering engine to initiate notifications to plant personnel during emergency situations such as pipeline leakage. This section also proposes a strategy to configure OWBE's SharePoint workflow engine to handle automated execution of mission critical workflows which enable pipeline companies to safely shut down a pipeline during abnormal situations.

Leakage is a catastrophic event in the pipeline industry as it involves destruction of equipment, loss of life, loss of valuable products and environmental contamination. Unauthorized digging, floods, corrosion, operational faults, and mechanical failures are some of the factors contributing to pipeline leakage [36]. Pipeline accidents incur huge financial liabilities for pipeline companies in terms of damage compensation. There is no upper bound on damage compensation a pipeline operator would pay for clean-up costs for oil spills in Canada [37]. To ensure safety and integrity, pipeline operators closely monitor and control pipeline operations from control centres (365x7x24) using Supervisory Control and Data Acquisition (SCADA) systems equipped with state-of-the-art leak detection system, instrumentation, and satellite communication [38].

3.5.1 Shortcomings of Paper-based SOPs

Pipeline companies rely heavily on sophisticated leak detection systems which provide accurate estimates of leak size and leak location within a pipeline network. When a “potential” “or “real” leak or rupture is indicated by a leak alarm generated by a leak detection system, operators at the control centre initiate a standard operating procedure which involves various troubleshooting steps to effectively diagnose and respond to leak alarms [39].



Suncor Energy Canadian Pipeline Limited Partnership

SUNCOR ENERGY

SPCC Task Procedure

Procedure Name	Response to Leak Detection Alarms	Facility:		Procedure No:	111
Date Procedure Implemented	Feb.23, 2012	Author	Chad Danilak	Approved By	V.Krieger
Revision/Review Frequency	Once a Year		Revision tracking at the end of the document		
	Not to exceed 15 Months				

1. Objective

- To ensure Operators quickly and effectively diagnose and respond to leak detection alarms
- To maximize safety of people, facilities and environment during a “potential” or “real” pipeline leak or rupture
- To ensure Operators use sound decision making skills for problem analysis in taking proper corrective actions

2. Risks

- Personal Injury or Loss of life
- Destruction of equipment and property
- Loss of product
- Environmental contamination

3. Supporting Documents

- ATMOS Leak Detection System: Operations Procedures
- Integrated Contingency Plan for RMPL
- OSPL Emergency Response Plan: 2.0 Organization- 2.4 Emergency Response and Communication Flowchart
- FB Emergency Response Plan

C:\Documents and Settings\vkrieger\Local Settings\Temporary Internet Files\OLK919\111-Response to ATMOS LD alarms.doc

Suncor Energy Canadian Pipeline Limited Partnership

SUNCOR ENERGY

SPCC Task Procedure

4. Procedure

Actions:	**Notes:**
1. **Leak Warning Alarm**- The highest lambda for the pipeline LDU has risen **above 0**. This indicates that the lambda(s) are rising and a leak alarm could be coming.	• Open ATMOS Group trend and begin investigating the lambda rise. • Open ATMOS Leak Alarm Trouble shooting trends.
2. **Leak Alarm**- The highest lambda for a pipeline LDU has risen above 4.6.	• Consult with other operators or phone SCADA analysts for support if needed.
3. Try to identify the source by eliminating factors that would create a false alarm (troubleshoot leak alarm). Check for bad flow data or pressure drops or pressure data issues using ATMOS trends.	• Did you launch a pig without being in pig mode? • Were the field operators doing any work in the area? • Were the meters being proved and you were not in prove mode? • Did you lose communications?
4. 10 min Leak Alarm- A leak alarm has been active for 10 minutes. **SHUT DOWN THE PIPELINE.** • **If the leak alarm is known to be false, no action is required, fill out the leak alarm sheet.**	• **Shutdown the pipeline**, call the CC Supervisor and SCADA Analyst for support. • Isolate pipeline by closing all mainline valves and do a static pressure test. • **If the pipeline is shutdown due to a suspected leak, the pipeline can not be restarted until supervisor approval has been given.**
5. Fill out ATMOS LD Alarm Report	• To aid in fine tuning the ATMOS LDS • Add relevant information- operating variables, rates, batch swings, split streaming etc.

C:\Documents and Settings\vkrieger\Local Settings\Temporary Internet Files\OLK919\111-Response to ATMOS LD alarms.doc

Figure 3.22: Snapshot of a paper-based “response to leak alarms” SOP

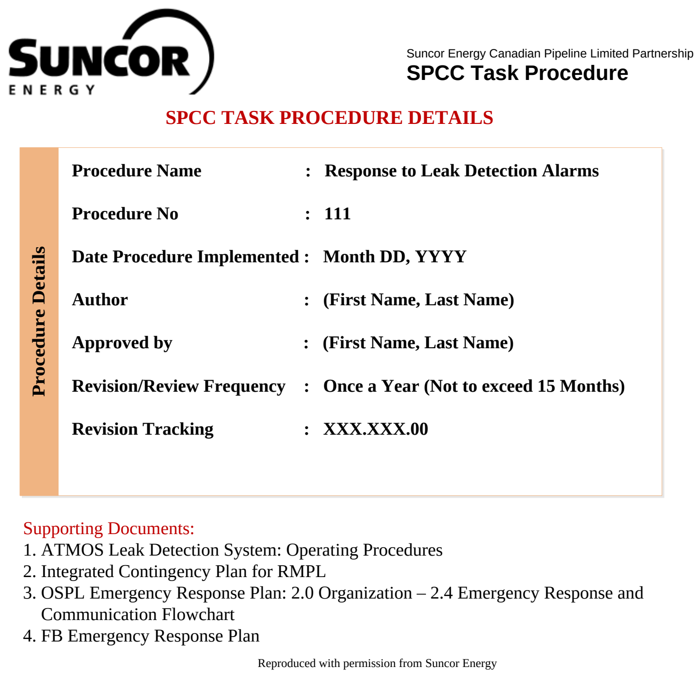
Figure 3.22 shows a snapshot of a paper-based “response to leak alarms” SOP which is implemented by Suncor Energy Pipelines. This emergency procedure guides operators to monitor various process variable trends, consult SCADA analysts, identify false alarms, deploy aerial patrols, communicate with field personnel over the phone and determine the root cause of a leak [39].

Factors such as inaccurate process measurement, faulty instruments, lost communication, improper meter proving, miscommunication during a pig launch, field operator intervention affect the performance of a leak detection system. A false alarm demands operators’ cognitive skills as they could overlook a real leak event [40]. During a leak investigation, operators at control centre may consult with other on-board operators, SCADA analysts and field personnel under time constraints. For instance, at Suncor Energy Pipelines, an incipient or real pipeline leak is indicated by a rising statistical parameter over a threshold value within the leak detection system. When a leak alarm from the leak detection system is active for ten minutes, the pertinent pipeline will be shut down [39]. Shutting down a pipeline due to a false alarm will degrade operational efficiency. This will pose additional stress on the operators involved in the leak investigation to quickly recognize the situation, diagnose the problem and determine the root cause of a leak alarm. For operators following paper-based SOPs or their mental map of the tasks to deal with such an emergency procedure, there is a high probability of missing some of the critical steps involved.

Paper-based SOPs are updated periodically, every 6 months or annually depending upon the corporate policies. Written procedure documents (paper-based SOP) are not always followed; often they fall out of date and are left on the shelf [18]. An emergency response SOP shown in Figure 3.22 is not a "good for all" procedure to deal with all kinds of pipeline leak alarms encountered in the control centre. Pipelines transport a wide variety of crude oil and petroleum products. Apart from factors discussed in Section 3.5.1, factors such as variation in fluid properties, product batching sequence, pipeline configuration, and operational characteristics also have a profound impact on the rate of leak alarms [40]. An operator's intrinsic understanding of the pipeline operations and operational characteristics of the fluid they transport is very critical in identifying the root causes of leak alarms. Written procedures do not capture this intrinsic understanding of an expert operator in justifying a leak alarm.

3.5.2 Merits of Software-based Workflows

A software-based workflow (electronic workflow) developed in OWBE can be instrumental in capturing this valuable operational knowledge from an expert and transfer it to other on-board operational crew. An electronic workflow is implemented in a web browser and provides step-by-step instructions to an operator while handling a leak alarm. Figure 3.23 shows the starting page of the proposed electronic workflow for "response to leak alarms" emergency procedure. This web page contains clickable hyperlinks that retrieve all procedural relevant information such as author contact details, P&ID diagrams, plant floor layout, revision history, established operating conditions and related documentation.



SUNCOR ENERGY

Suncor Energy Canadian Pipeline Limited Partnership
SPCC Task Procedure

SPCC TASK PROCEDURE DETAILS

Procedure Details

Procedure Name	:	**Response to Leak Detection Alarms**
Procedure No	:	**111**
Date Procedure Implemented	:	**Month DD, YYYY**
Author	:	**(First Name, Last Name)**
Approved by	:	**(First Name, Last Name)**
Revision/Review Frequency	:	**Once a Year (Not to exceed 15 Months)**
Revision Tracking	:	**XXX.XXX.00**

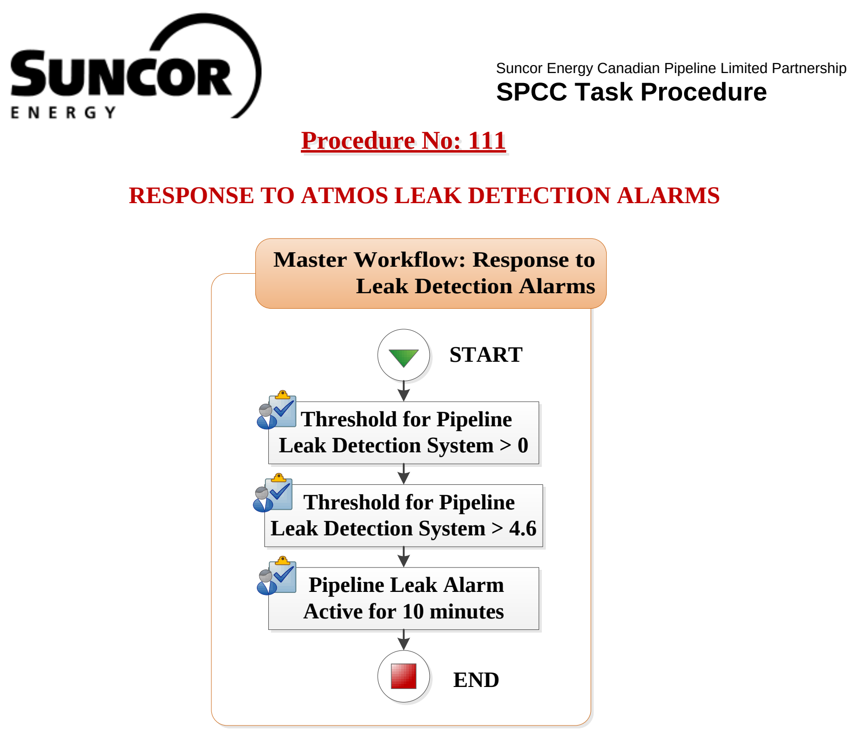
Supporting Documents:
1. ATMOS Leak Detection System: Operating Procedures
2. Integrated Contingency Plan for RMPL
3. OSPL Emergency Response Plan: 2.0 Organization – 2.4 Emergency Response and Communication Flowchart
4. FB Emergency Response Plan

Reproduced with permission from Suncor Energy

Figure 3.23: Page 1: Operational procedure details

By leveraging the benefits of e-mail notifications in Windows Workflow foundation [41], notifications of any procedural modifications can be sent on-the - fly to cross-functional teams like management, operators and maintenance personnel. This strategy will provide a seamless communication of operational and process information among people, equipment and plant floor. Configuration of automated e-mail notifications is discussed in Section 3.5.4. Thus, OWBE provides opportunities for streamlined modularization of operating procedures and create higher level of visibility for operators on the operational characteristics of the processes and the equipment they deal with.

Figure 3.24 shows a master-workflow that establishes the main window of the proposed electronic workflow when deployed as a console application or a workflow web service. The start and stop blocks indicate the commencement and termination of stipulated operator actions in response to leak alarms. The master-workflow is divided into three sub-workflows (SW-1, SW-2 and SW-3) based on threshold values for the leak detection system. Each sub-workflow block is self-explanatory and consists of sequence of established operating procedures and operators' tasks to deal with emergency situations. These sub-workflows are clickable blocks that navigate users through the entire operating procedure. The block diagrams of sub-workflows SW-1, SW-2, and SW-3 are shown in Figures 3.25, 3.26 and 3.27 respectively.



Reproduced with permission from Suncor Energy

Figure 3.24: Master-Workflow: "Response to leak detection alarms"

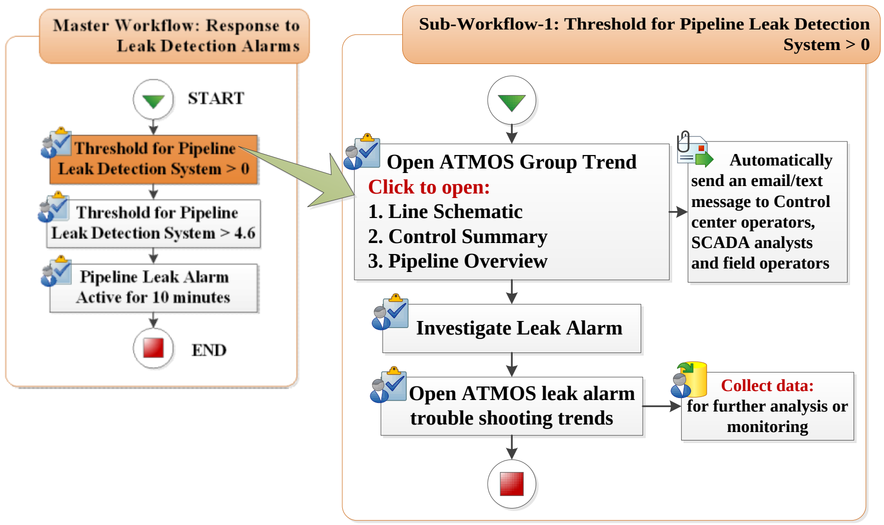


Figure 3.25: SW-1: Threshold for leak detection system > 0

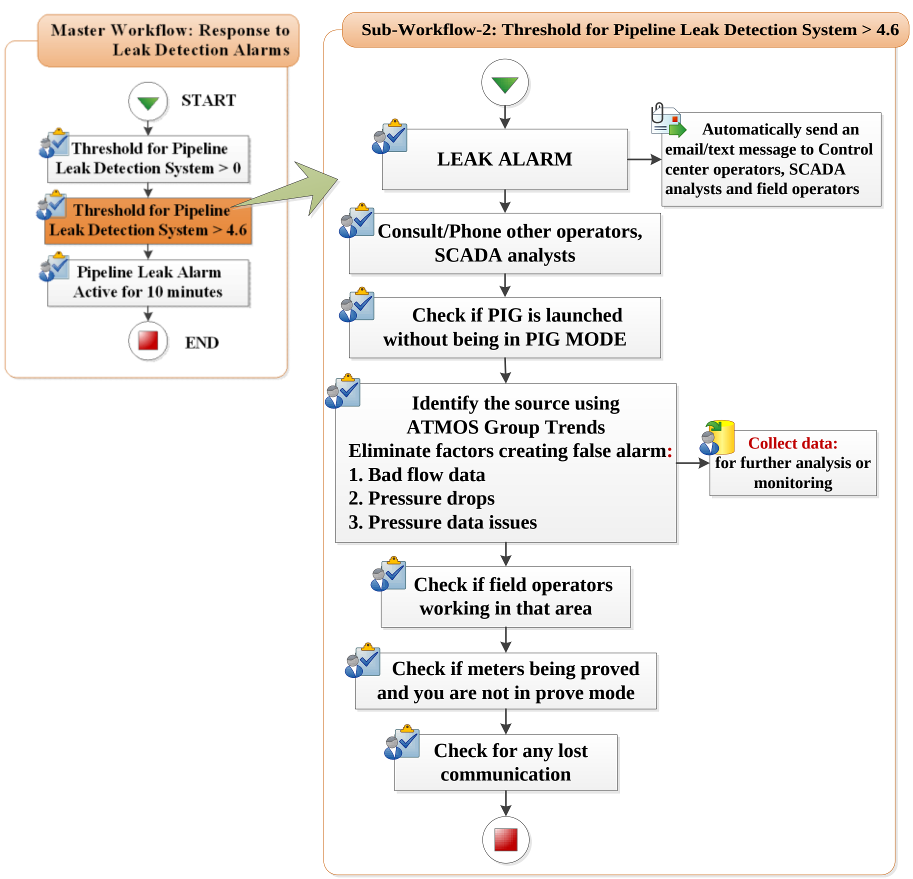


Figure 3.26: SW-2: Threshold for leak detection system > 4.6

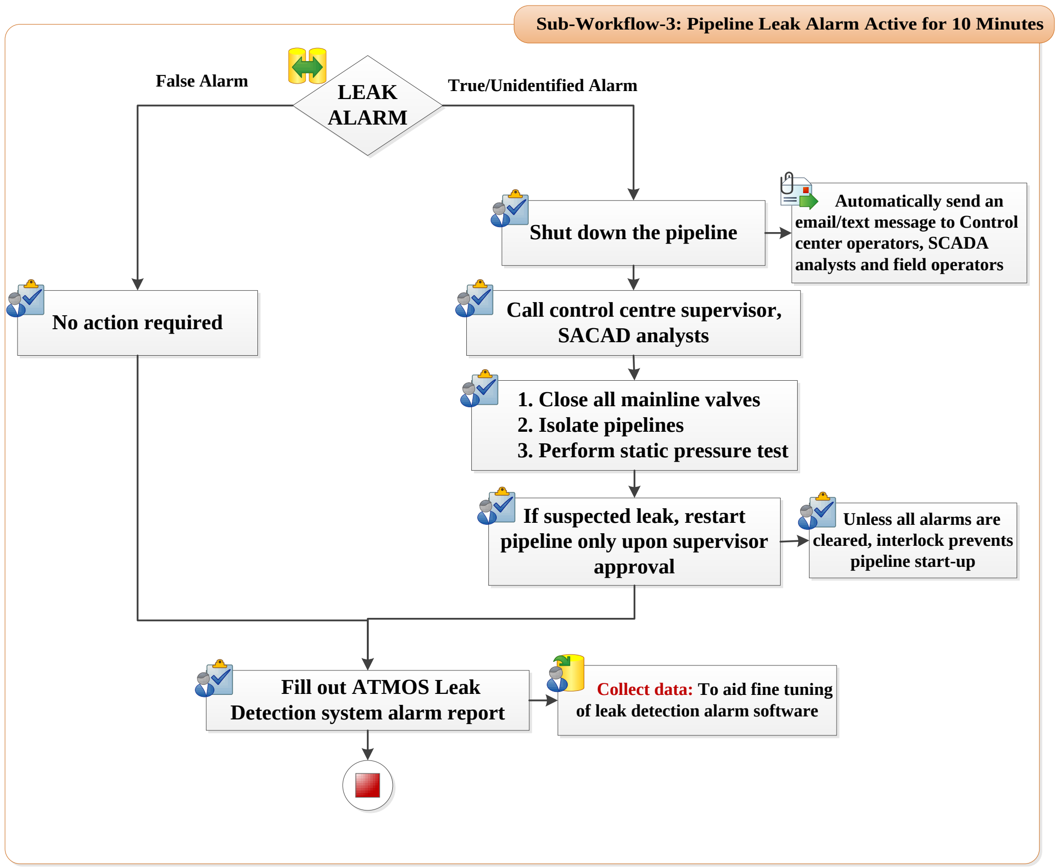


Figure 3.27: SW-3: Pipeline leak alarm active for ten minutes

1. Sub-Workflow 1 (SW-1):

The operating tasks listed in sub-workflow 1 (SW-1) are effective when the threshold for the leak detection system is greater than zero; a warning for an incipient leak alarm. This part of the workflow guides operators to investigate group trends of process variables such as flow and pressure measurements along the pipeline. Operators can quickly navigate to pipeline schematics, pressure and flow profiles, and control summary by clicking appropriate hyperlinked blocks as shown in Figure 3.25. OWBE could be programmed to send e-mail notifications to plant personnel based on an alarm limit check against incoming real time measurement data or events.

2. **Sub-Workflow 2 (SW-2):**

The sequence of operating tasks listed in sub-workflow 2 is effective when the threshold for the leak detection system is greater than 4.6; indicating a confirmed leak alarm. The sequence of steps shown in Figure 3.26 helps operators to recognize the situation, diagnose the problem and determine the root cause of the leak alarm. This is a crucial work practice as it demands the operator's cognitive ability to resolve the situation within a period of ten minutes. When a confirmed leak alarm pops, sub-workflow 2 is able to send automated notifications to alert the operational crew. This workflow guides operators through a sequence of troubleshooting steps such as communicating with SCADA analysts and on board operators over the phone, checking if a pig has been launched into the pipeline, checking if meters are proved ('proving' refers to the process of regular inspection and calibration) but not acknowledged at the control centre, identifying noisy and frozen data, inspecting of any lost communication.

Crude oil flows very slowly (approximately 5 km/hr) through a pipeline [42]. Hence, the custody transfer of crude oil is a long running process and spans over several shifts in the control centre. By leveraging the benefits of OWBE, shift operators are able to document shift-turnover information such as procedural modifications, system failures, frozen data, pump station upsets, RTU failures, false leak alarms, pig launch, meter proving, field maintenance intervention, and any lost communication which would have occurred during their shift. This will assist the incoming shift crew to fully understand the current operational status of the pipeline before they take ownership. Electronic workflows allow operators to click on a block to retrieve pertinent information entered by outgoing shift operators. Tailoring to changing operating environments, any procedural modifications recommended by an expert operator are incorporated on-the-fly and hence the procedures are kept up-to-date.

Alarm and event logs archived during a leak alarm are documented to provide a database of **"operational signature"** (what actions are executed by whom, where and when). Sub-workflow 2 ensures that operators quickly, effectively diagnose and respond to leak detection alarms.

3. **Sub-Workflow 3 (SW-3):**

   If a leak alarm is active for ten minutes and if the operator is unable to determine the root cause; that pertinent pipeline is shut down. If the alarm is due to a suspected leak, the pipeline cannot be restarted until the supervisor approval has been issued. Sub-workflow 3 guides operators through pipeline shutdown sequence and to perform static pressure test post-shutdown. Notifications on pipeline shutdown are communicated to all stakeholders involved in the operation. Besides the operational crew, shutdown notifications are communicated to product shipper, nomination management team, and revenue accounting team. If the leak alarm is known to be false, no action is required. In both the cases, information pertaining to operating variables and batch swings is documented in the leak alarm report which helps fine tune the leak detection system [39].

3.5.3 Annotated Workflow Diagrams Developed in Visio 2010

Software-based workflows capture and document expert operational knowledge in the form of workflow activities. Workflows shown in Figures 3.24 through 3.27 are annotated diagrams developed in the Microsoft Visio Premium 2010. These annotated diagrams show how workflow activities would look like when configured in OWBE. Workflow activities which are developed in Visio 2010 are exported for configuration in Microsoft SharePoint Designer 2010 which is the backbone for OWBE platform [20]. Figure 3.28 shows a snippet of XOML code corresponding to master-workflow shown in Figure 3.24. (The skeleton of a workflow activity is defined in a XOML file as discussed in Section 2.4.3).

```
<ns0:RootWorkflowActivityWithData x:Class="Microsoft.SharePoint.Workflow.ROOT"
x:Name="ROOT" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/workflow"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:ns0="clr-
namespace:Microsoft.SharePoint.WorkflowActions;Assembly=Microsoft.SharePoint.Workfl
owActions, Version=14.0.0.0, Culture=neutral, PublicKeyToken=null">

<ns0:RootWorkflowActivityWithData.WorkflowFields>
<ns0:WorkflowDataField Name="__list" Type="System.String" />
<ns0:WorkflowDataField Name="__item"
Type="Microsoft.SharePoint.Workflow.SPItemKey" />
<ns0:WorkflowDataField Name="__context"
Type="Microsoft.SharePoint.WorkflowActions.WorkflowContext" />
<ns0:WorkflowDataField Name="__initParams"
Type="Microsoft.SharePoint.Workflow.SPWorkflowActivationProperties" />
<ns0:WorkflowDataField Name="__workflowId" Type="System.Guid" />
<ns0:WorkflowDataField Name="calc" Type="System.Double" />
</ns0:RootWorkflowActivityWithData.WorkflowFields>
<ns0:OnWorkflowActivated WorkflowProperties="{ActivityBind
ROOT,Path=__initParams}" x:Name="ID1">

<ns0:OnWorkflowActivated.CorrelationToken>

<wf0:CorrelationToken Name="refObject" OwnerActivityName="ROOT"
xmlns:wf0="http://schemas.microsoft.com/winfx/2006/xaml/workflow" />

</ns0:OnWorkflowActivated.CorrelationToken>

</ns0:OnWorkflowActivated>

<ns0:ApplyActivation __Context="{ActivityBind ROOT,Path=__context}" x:Name="ID2"
__WorkflowProperties="{ActivityBind ROOT,Path=__initParams}" />

<SequenceActivity x:Name="ID3">

<ns0:TodoItemTask x:Name="ID4" ContentTypeId="{x:Null}" Title="{x:Null}"
__Context="{ActivityBind ROOT,Path=__context}" Description="ShapeGuid={4567586F-
5D76-4793-8671-F72FA0EFDD4D};PageId=0;ShapeId=8;ShapeText= Threshold for
Pipeline Leak Detection System &gt; 0 " AssignedTo="{x:Null}" />

<ns0:TodoItemTask x:Name="ID5" ContentTypeId="{x:Null}" Title="{x:Null}"
__Context="{ActivityBind ROOT,Path=__context}" Description="ShapeGuid={424B13B1-
1AF0-420C-B864-94CA8CE19580};PageId=0;ShapeId=12;ShapeText=  Threshold for
Pipeline Leak Detection System &gt; 4.6 " AssignedTo="{x:Null}" />

<ns0:TodoItemTask x:Name="ID6" ContentTypeId="{x:Null}" Title="{x:Null}"
__Context="{ActivityBind ROOT,Path=__context}" Description="ShapeGuid={87E4A6D7-
2ED0-4EC6-B5EA-D9A4D81DE348};PageId=0;ShapeId=16;ShapeText= Pipeline Leak
Alarm &#xA;Active for 10 minutes  " AssignedTo="{x:Null}" />

</SequenceActivity>

</ns0:RootWorkflowActivityWithData>
```

Figure 3.28: A XOML code snippet corresponding to “Master Workflow” annotated diagram developed in the Microsoft Visio Premium 2010

3.5.4 OWBE Triggering Emergency Response Notifications

Effective communication among operational team, field personnel, and management is a paramount task during an abnormal situation. In pipeline operations, when the threshold for the leak detection system exceeds a critical limit, OWBE can be configured to initiate communication workflows (an email or text notification) to operators, SCADA analysts and field personnel. These communication workflows could be email notifications followed by a sequence of complex work practices to engage concerned plant personnel to resolve an issue. These notifications serve as a precursor of an emergency situation demanding further investigation. For instance, a threshold for leak detection system greater than 4.6 confirms a leak alarm and the operator should resolve the issue within a period of ten minutes to prevent a pipeline from mandatory shutdown (Figure 3.26 & 3.27). Based on this leak alarm, OWBE is able to trigger an emergency response notification to quickly engage concerned plant personnel to avoid an impending emergency situation. With OWBE, automatic escalation of alerts to other operators and the workforce is possible when a shift-in-charge operator is away, busy or unable to respond to the leak alarms.

Contemporary SCADA host systems are integrated with OPC Servers for data connectivity [43]. When a pipeline leak alarm is generated by the leak detection system, the SCADA host reports this alarm over OPC UA, which can then be used to trigger automated workflows. For instance, when a leak alarm is active for ten minutes, operators at the Suncor Pipelines control centre (SPCC) shut down the pertinent pipeline as shown in Figure 3.27. As this operation is a mission critical procedure, an automated emergency shutdown workflow can be developed within OWBE to make sure that all the required tasks are performed consistently no matter who is executing them. Figure 3.29 shows an artistic rendition of how an emergency pipeline shutdown workflow can be triggered by a leak alarm.

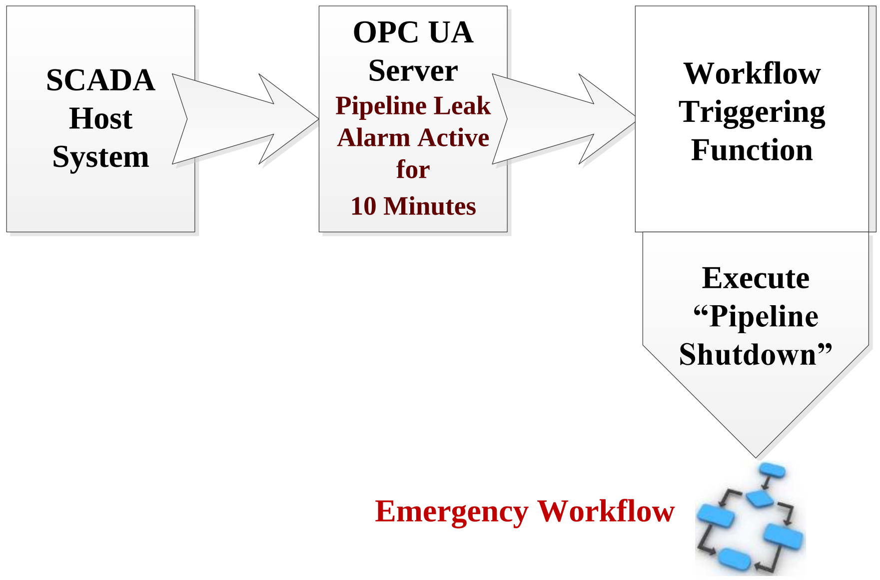


Figure 3.29: A leak alarm triggering an emergency workflow

This will enable pipeline operators to practice best operating procedures to safely and efficiently deal with out-of-spec events and critical emergencies.

3.6 Summary

Workflows are developed from the event logs of an industrial case study, namely the tank-swing process at Suncor Energy Pipelines. This chapter gives a brief overview of the tank-swing process and the challenges involved in pre-processing of event logs associated with it. The developed workflows are checked for conformance with standard operating procedures to identify any procedural violations. This chapter also illustrates how operators are able to author, publish and modify their operational workflows catering to changing process demands.

Chapter 4

Case Study 2: Developing Workflows for Heating Plant Operations

4.1 Description of the Heating Plant at University of Alberta

The information related to physical setup, capacity and system configuration of the heating plant discussed throughout this chapter, is obtained from the Department of Facilities and Operations at University of Alberta [44]. The heating plant supplies steam required for campus space heating and research facilities. The plant has a steam production capacity of 650 tonnes per hour and the steam load varies with season. The plant has two boilers which produce steam at 2,760 kPag and three boilers producing steam at 6,200 kPag [44]. All the boilers are natural-gas fired; however boiler #2 can be oil-fired. The plant is equipped with a 13.3 MW back pressure steam turbine generator and a 26.4 MW condensing steam turbine generator. Boilers producing steam at 6,200 kPag are connected to 13.3 MW back pressure generator where the steam pressure and temperature is reduced through the turbine. Steam is exhausted from the turbine at 1,035 kPag and is supplied for campus space heating and research facilities. Hence, this heating plant is a cogeneration facility as it produces both process heat and electricity. The heating plant and its auxiliary processes have more than 3,400 field instrument tags which are controlled and monitored by PROVOX and DeltaV distributed control systems (DCS) [44]. During the initial setup of the plant, all the boilers were configured with FISHER PROVOX controllers, however, most of them have been upgraded to EMERSON DeltaV DCS. For real-time data monitoring, all the control systems are equipped with OSIsoft PI server that collects all control and monitoring related information of the plant operations.

4.1.2 Orchestrated Workflows for Boiler Start-up and Shutdown Operations

The core concept of orchestrated workflows by example (OWBE) project is to allow operators and plant personnel automate their own workflow processes by analyzing and capturing their actions while they execute their work procedures. The first and foremost step in developing live orchestrated workflows for boiler operations is to gain inherent knowledge of the process by consulting experienced plant personnel, walking through their standard operating procedures (SOPs) and witnessing sequence of tasks as they execute their work process from the control room. A comprehensive review is carried out to develop boiler workflows by analyzing various information sources available from the heating plant such as paper-based standard operating procedures, operator log book, shift supervisor's log, PROVOX alarm and event data, DeltaV process historians and operator's verbal communication (between field and the control room).

In this chapter, our focus is to develop workflows for boiler operations from event logs and standard operating procedure(s) as mentioned below:

- Burner rotation sequence from event logs during the start-up of boiler #4 at the University of Alberta on November 09, 2012.
- Develop software-based workflows from paper-based SOPs for boiler #6 shutdown operation on April 25, 2013 (we witnessed this boiler shutdown sequence by sitting next to an operator in the control room).

Boiler #4 is controlled by a PROVOX system whereas boiler #6 is on DeltaV DCS. Event logs corresponding to boiler #4 and boiler #6 are disparate due to the dissimilarity in system architecture. Also, for boiler #6, data logging is configured in such a way that in the absence of alarms, events are not archived in the database. These control systems are integrated with PI Server where event data is retrieved in the form of Excel files for analysis and report generation.

4.2 Burner Rotation Sequence during Start-up of Boiler #4

A schematic of a natural-gas fired drum-type boiler is shown in Figure 4.1. A drum-type boiler has two drums; namely steam drum and mud drum. These two

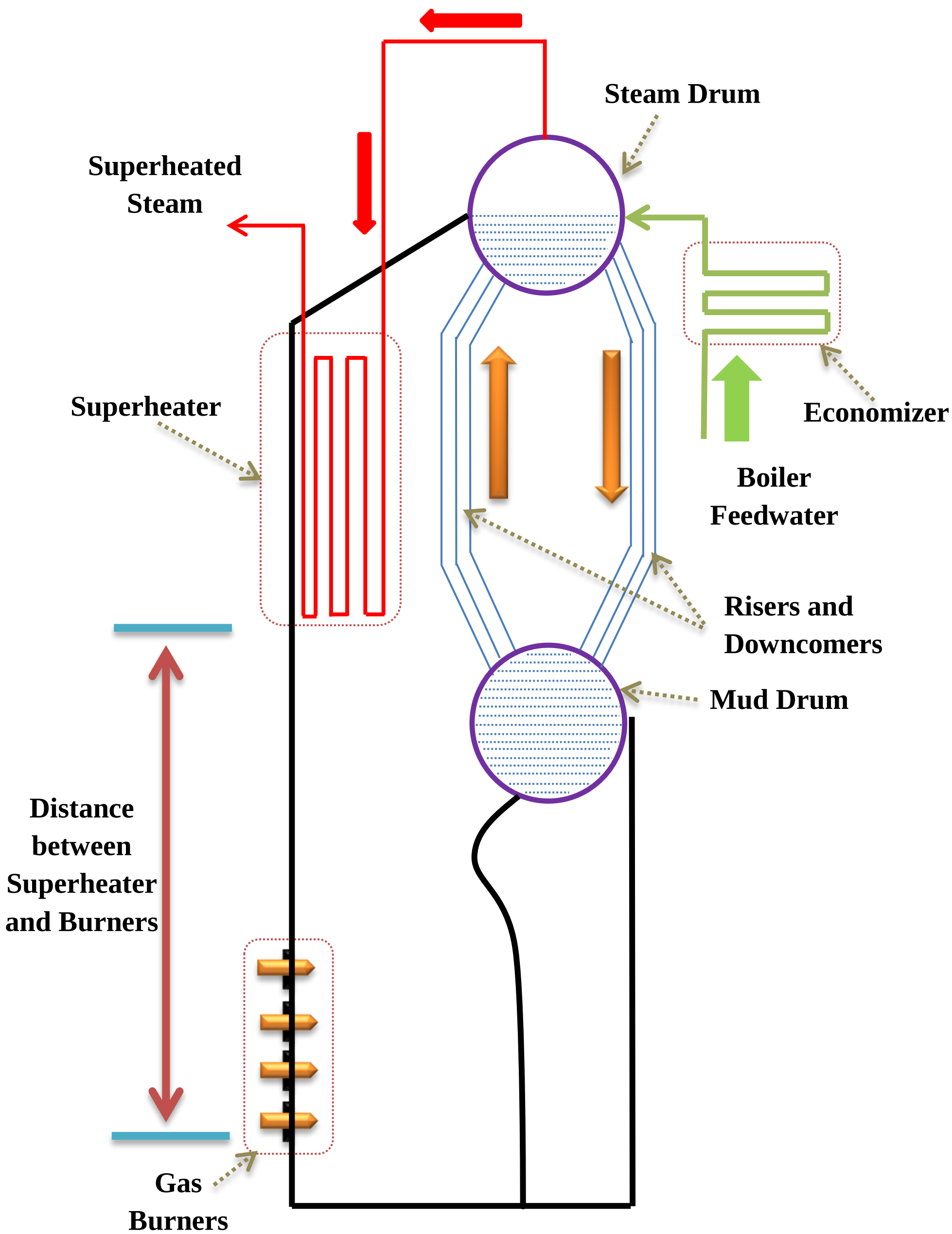


Figure 4.1: Schematic of a drum-type natural-gas fired boiler

drums are connected through a series of water-tubes called risers and downcomers as shown in Figure 4.1. Steam is generated when these water-tubes are exposed to heat within the gas combustion furnace where a mixture of natural gas and air is ignited at a gas burner. Gas burners ensure proper mixing and combustion of natural gas and air in the boiler furnace. Steam and water rises up in the tubes that are closer to the furnace (risers) and water drops in the tubes which are farther away from the furnace (downcomers) creating a water circulation phenomenon [45]. Steam accumulated at the steam drum (top drum) flows through a series of tubes which are exposed to the furnace where any moisture content left out in the steam is completely removed (resulting in dry steam). This part of the boiler section where the temperature of the steam is raised is called a superheater. As shown in Figure 4.1, most of the industrial type boilers are equipped with multiple burners. Figure 4.2 shows the physical orientation of the six burners associated with boiler #4 in the heating plant. The warm-up rate of a boiler during a start-up is specified by the boiler manufacturer [45]. Rapid warming up of a boiler during start-up results in temperature swings creating thermal stress on thick metallic tubes associated with boiler drums, superheaters, and economizers. Hence, during a start-up, boilers are warmed-up gradually by rotating burners. With reference to Figure 4.2, burner #1 is ignited first for 20 minutes; then burner #2 is ignited for 20 minutes while burner #1 is shut off, and then burner #3 is ignited for 20 minutes, so that the heat transfer within boiler #4 remains uniform [46].

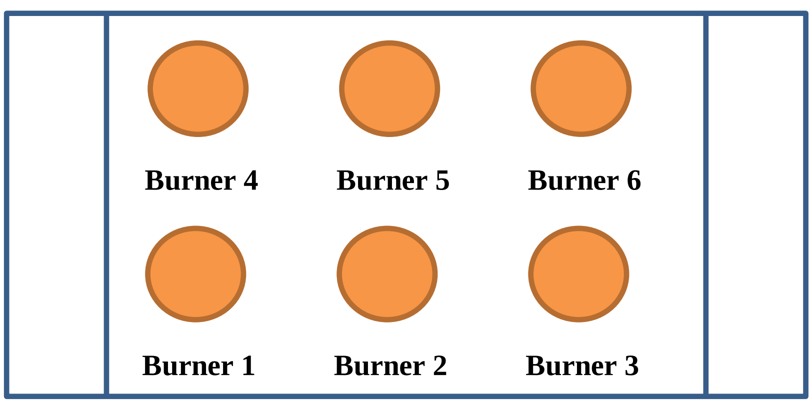


Figure 4.2: Burners arrangement in boiler #4 at the heating plant, University of Alberta

Based on the design specifications of boiler #4, the rate of warm-up from a cold setting is limited to 38 deg C rise per hour (1500 kPa per hour) while rotating the burners. Figure 4.3 shows a predetermined burner rotation sequence for boiler #4.

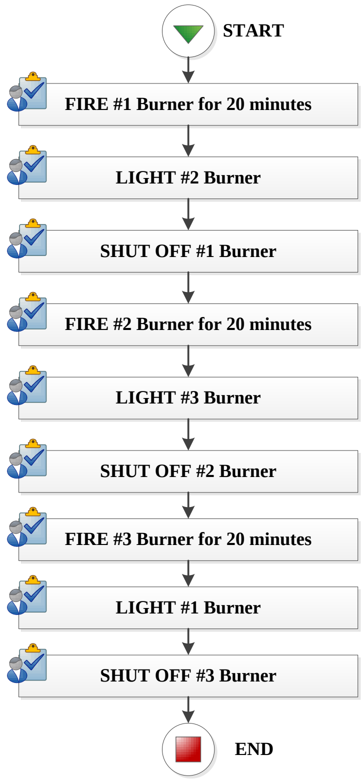


Figure 4.3: Burner ignition rotation sequence developed from boiler #4 SOP

From Figure 4.3, it is noticeable that each burner is ignited for 20 minutes before switching to another burner. The subsequent burner is ignited before shutting down the current burner. This is done to avoid a purge of the burner section gases, which automatically occurs in case of a flame blowout. The number of burners ignited and rotated during a warm-up period depends on whether boiler #4 is brought to operating conditions from a cold-setting or from a hot standby. During a boiler start-up, operators at the heating plant always follow the burner rotation sequence as shown in Figure 4.3 to prevent uneven heating of superheater headers [46]. In the following subsections, we extract workflows for burner rotation sequence based on the ongoing events archived in the PROVOX event logs during the start-up of boiler #4 on November 09, 2012.

4.2.1 Event Logs for Burner Rotation Sequence during Boiler #4 Start-up Procedure

Alarm and event messages associated with various instruments and equipment in the heating plant are archived chronologically in the PROVOX system. Such equipment includes boilers, turbines, feedwater pumps, gas analyzers, deaerators, demineralizers, superheaters, economizers, FD fans, compressors, diesel generators, burner management system (BMS), flame scanners, and burners to name a few. In this section, we analyze alarm and event messages which are associated with the start-up of boiler #4 on November 09, 2012. Boiler #4 is a 900 PSIg (6,200 kPag) natural-gas fired boiler which runs at full capacity during the winter terms. The campus heating load will be at peak during November and hence the boiler is started up from a hot standby condition. The boiler start-up procedure begins at 02:00 PM and is brought to operating conditions by 05:00 PM on the same day. During this period of three hours, a total of 735 alarms and events is archived in the PROVOX system. Table 4.1 shows a snapshot of the PROVOX alarm and event log spreadsheet.

Ord	MsgId	EventDate	EventTime	Area	PointTag	PointDesc	ModeStr	PvValue	PvStr	SpValue	SpStr	User
2384933	20	11/09/2012	2:51:17 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	2	Off Line	
2384934	20	11/09/2012	2:51:20 PM	DOORS	DS139019	ST13901 2000		0	Closed			
2384935	20	11/09/2012	2:51:20 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	1	On Line	
2384936	22	11/09/2012	2:51:22 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	1	On Line	HP2
2384937	20	11/09/2012	2:51:37 PM	DOORS	MD201019	MtnDet @2010		0	Normal			
2384938	20	11/09/2012	2:51:38 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		0	Ok			
2384939	20	11/09/2012	2:51:40 PM	SECURITY-HP	HPMS-3	Stb Tunl Mtn		1	Motion			
2384940	20	11/09/2012	2:51:41 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	1	On Line	
2384941	20	11/09/2012	2:51:42 PM	DOORS	DS139019	ST13901 2000		1	Open			
2384942	20	11/09/2012	2:51:45 PM	BOILER-4	FRC-4003	A/F TRIM	MAN	89.5625		89.5625		
2384943	20	11/09/2012	2:51:45 PM	SECURITY-HP	HPMS-3	Stb Tunl Mtn		0	Ok			
2384944	20	11/09/2012	2:51:47 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		1	Motion			
2384945	20	11/09/2012	2:51:48 PM	BOILER-4	FRC-4003	A/F TRIM	MAN	100		100		
2384946	20	11/09/2012	2:51:57 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		0	Ok			
2384947	20	11/09/2012	2:52:02 PM	DOORS	DS139019	ST13901 2000		0	Closed			
2384948	20	11/09/2012	2:52:08 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		1	Motion			
2384949	22	11/09/2012	2:52:09 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	1	On Line	HP2
2384950	20	11/09/2012	2:52:15 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	2	Off Line	
2384951	20	11/09/2012	2:52:17 PM	BOILER-4	LT-4004A	B4 LVL A		8.076172		0		
2384952	20	11/09/2012	2:52:19 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		0	Ok			
2384953	20	11/09/2012	2:52:24 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		1	Motion			
2384954	20	11/09/2012	2:52:35 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	2	Off Line	
2384955	20	11/09/2012	2:52:38 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	1	On Line	
2384956	20	11/09/2012	2:52:41 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		0	Ok			
2384957	22	11/09/2012	2:52:41 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	1	On Line	HP2

Table 4.1: Alarm and event data archived in the PROVOX system

4.2.2 Event Log Pre-processing Steps for Burner Rotation Sequence

The PROVOX control system is equipped with an OSIsoft PI server with gathers all real-time events and alarms associated with the start-up of boiler #4. Operators collect archived event data from the PI system in the form of Excel spreadsheets. The raw event log spreadsheet is very complex and contains about 225 Excel columns representing various data attributes related to plant operations. Some of the data attributes such as system nodes, system configuration alerts, device address, alarm grouping, and alarm priorities (which are not shown in Table 4.1) are less relevant to workflow mining tasks [10]. To develop workflows for burner rotation sequence, we will be specifically focussing on custom event messages as shown in Table 4.3. Hence, pre-processing is carried out on raw event log spreadsheets by filtering events relevant to burner rotation sequence. Event log pre-processing also ensures that the event log file is compatible with Disco software [10], which can then be used to generate the burner rotation workflows automatically. The pre-processing procedure involves following steps:

1. **Filtering raw event log file**

   The heating plant at the University of Alberta has multiple boilers supplying steam to a common steam header. Each boiler operation is equipped with a dedicated operator workstation and auxiliary equipment. For process control and monitoring purposes, boilers and their associated equipment are categorized into distinct areas or system nodes within the PROVOX control system hierarchy. Hence, the event logs are parsed up by system nodes or equipment area by setting appropriate filtering criterion in the PROVOX event log spreadsheet. The first step is to filter events associated with boiler #4 by setting Excel filter criterion to:

   Filter → Area: "BOILER 4" (as shown in Figure 4.4):

Ord	MsgId	EventDate	EventTime	Area	PointTag	PointDesc	ModeStr	PvValue	PvStr	SpValue	SpStr	User
2384933	20	11/09/2012	2:51:17 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	2	Off Line	
2384934	20	11/09/2012	2:51:20 PM	DOORS	DS139019	ST13901 2000		0	Closed			
2384935	20	11/09/2012	2:51:20 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	1	On Line	
2384936	22	11/09/2012	2:51:22 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	1	On Line	HP2
2384937	20	11/09/2012	2:51:37 PM	DOORS	MD201019	MtnDet @2010		0	Normal			
2384938	20	11/09/2012	2:51:38 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		0	Ok			
2384939	20	11/09/2012	2:51:40 PM	SECURITY-HP	HPMS-3	Stb Tunl Mtn		1	Motion			
2384940	20	11/09/2012	2:51:41 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	1	On Line	
2384941	20	11/09/2012	2:51:42 PM	DOORS	DS139019	ST13901 2000		1	Open			
2384942	20	11/09/2012	2:51:45 PM	BOILER-4	FRC-4003	A/F TRIM	MAN	89.5625		89.5625		
2384943	20	11/09/2012	2:51:45 PM	SECURITY-HP	HPMS-3	Stb Tunl Mtn		0	Ok			
2384944	20	11/09/2012	2:51:47 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		1	Motion			
2384945	20	11/09/2012	2:51:48 PM	BOILER-4	FRC-4003	A/F TRIM	MAN	100		100		
2384946	20	11/09/2012	2:51:57 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		0	Ok			
2384947	20	11/09/2012	2:52:02 PM	DOORS	DS139019	ST13901 2000		0	Closed			
2384948	20	11/09/2012	2:52:08 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		1	Motion			
2384949	22	11/09/2012	2:52:09 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	1	On Line	HP2
2384950	20	11/09/2012	2:52:15 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	2	Off Line	
2384951	20	11/09/2012	2:52:17 PM	BOILER-4	LT-4004A	B4 LVL A		8.076172		0		
2384952	20	11/09/2012	2:52:19 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		0	Ok			
2384953	20	11/09/2012	2:52:24 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		1	Motion			
2384954	20	11/09/2012	2:52:35 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	2	Off Line	
2384955	20	11/09/2012	2:52:38 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	1	On Line	
2384956	20	11/09/2012	2:52:41 PM	SECURITY-HP	HPMS-4	SE Tunl Mtn		0	Ok			
2384957	22	11/09/2012	2:52:41 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	1	On Line	HP2

Filter → Area : "BOILER 4"

Ord	MsgId	EventDate	EventTime	Area	PointTag	PointDesc	ModeStr	PvValue	PvStr	SpValue	SpStr	User
2384933	20	11/09/2012	2:51:17 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	2	Off Line	
2384935	20	11/09/2012	2:51:20 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	1	On Line	
2384936	22	11/09/2012	2:51:22 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	1	On Line	HP2
2384940	20	11/09/2012	2:51:41 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	1	On Line	
2384942	20	11/09/2012	2:51:45 PM	BOILER-4	FRC-4003	A/F TRIM	MAN	89.5625		89.5625		
2384945	20	11/09/2012	2:51:48 PM	BOILER-4	FRC-4003	A/F TRIM	MAN	100		100		
2384949	22	11/09/2012	2:52:09 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	1	On Line	HP2
2384950	20	11/09/2012	2:52:15 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	2	Off Line	
2384951	20	11/09/2012	2:52:17 PM	BOILER-4	LT-4004A	B4 LVL A		8.076172		0		
2384954	20	11/09/2012	2:52:35 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	2	Off Line	
2384955	20	11/09/2012	2:52:38 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	1	On Line	
2384957	22	11/09/2012	2:52:41 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	1	On Line	HP2

Figure 4.4: Filtering raw event logs based on "BOILER 4"

The second step in event log pre-processing is to filter events associated with burner rotation sequence by setting Excel filter criterion to (as shown in Figure 4.5):

Filter → PointDesc: Select boiler #4 burners

("B4 Bur1 Sta" + "B4 Bur2 Sta" + "B4 Bur3 Sta" + "B4 Bur4 Sta")

Here "B4" represents boiler #4 and "Burn1" represents burner 1 in boiler #4. Although boiler #4 is equipped with six burners, only four of them (burner #1, #2, #3, and #4) are ignited during boiler start-up to meet the steam load on November 09, 2012.

Ord	MsgId	EventDate	EventTime	Area	PointTag	PointDesc	ModeStr	PvValue	PvStr	SpValue	SpStr	User
2384933	20	11/09/2012	2:51:17 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	2	Off Line	
2384935	20	11/09/2012	2:51:20 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	1	On Line	
2384936	22	11/09/2012	2:51:22 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	1	On Line	HP2
2384940	20	11/09/2012	2:51:41 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	1	On Line	
2384942	20	11/09/2012	2:51:45 PM	BOILER-4	FRC-4003	A/F TRIM	MAN	89.5625		89.5625		
2384945	20	11/09/2012	2:51:48 PM	BOILER-4	FRC-4003	A/F TRIM	MAN	100		100		
2384949	22	11/09/2012	2:52:09 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	1	On Line	HP2
2384950	20	11/09/2012	2:52:15 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	2	Off Line	
2384951	20	11/09/2012	2:52:17 PM	BOILER-4	LT-4004A	B4 LVL A		8.076172		0		
2384954	20	11/09/2012	2:52:35 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	2	Off Line	
2384955	20	11/09/2012	2:52:38 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	1	On Line	
2384957	22	11/09/2012	2:52:41 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	1	On Line	HP2

Filter → PointDesc : Selects all "B4 Burners"

Ord	MsgId	EventDate	EventTime	Area	PointTag	PointDesc	ModeStr	PvValue	PvStr	SpValue	SpStr	User
2384933	20	11/09/2012	2:51:17 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	2	Off Line	
2384935	20	11/09/2012	2:51:20 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	1	On Line	
2384936	22	11/09/2012	2:51:22 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	1	On Line	1	On Line	HP2
2384940	20	11/09/2012	2:51:41 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	1	On Line	
2384949	22	11/09/2012	2:52:09 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	1	On Line	HP2
2384950	20	11/09/2012	2:52:15 PM	BOILER-4	BB-4050	B4 Burn1 Sta	MAN	2	Off Line	2	Off Line	
2384954	20	11/09/2012	2:52:35 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	2	Off Line	
2384955	20	11/09/2012	2:52:38 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	1	On Line	
2384957	22	11/09/2012	2:52:41 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	1	On Line	1	On Line	HP2
2384959	20	11/09/2012	2:52:50 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	2	Off Line	1	On Line	
2384963	22	11/09/2012	2:52:56 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	2	Off Line	1	On Line	HP2
2384967	20	11/09/2012	2:53:07 PM	BOILER-4	BB-4020	B4 Burn2 Sta	MAN	2	Off Line	2	Off Line	
2384968	20	11/09/2012	2:53:16 PM	BOILER-4	BB-4040	B4 Burn4 Sta	MAN	2	Off Line	2	Off Line	
2384970	20	11/09/2012	2:53:32 PM	BOILER-4	BB-4030	B4 Burn3 Sta	MAN	2	Off Line	2	Off Line	
2384976	20	11/09/2012	2:54:11 PM	BOILER-4	BB-4030	B4 Burn3 Sta	MAN	1	On Line	2	Off Line	
2384977	20	11/09/2012	2:54:14 PM	BOILER-4	BB-4030	B4 Burn3 Sta	MAN	1	On Line	1	On Line	
2384981	22	11/09/2012	2:54:30 PM	BOILER-4	BB-4030	B4 Burn3 Sta	MAN	1	On Line	1	On Line	HP2
2385326	20	11/09/2012	4:22:30 PM	BOILER-4	BB-4030	B4 Burn3 Sta	MAN	2	Off Line	1	On Line	
2385327	22	11/09/2012	4:22:36 PM	BOILER-4	BB-4030	B4 Burn3 Sta	MAN	2	Off Line	1	On Line	HP2
2385329	20	11/09/2012	4:24:03 PM	BOILER-4	BB-4030	B4 Burn3 Sta	MAN	1	On Line	1	On Line	
2385423	20	11/09/2012	4:53:35 PM	BOILER-4	BB-4020	B4 Burn2 Sta	MAN	1	On Line	2	Off Line	
2385424	20	11/09/2012	4:53:38 PM	BOILER-4	BB-4020	B4 Burn2 Sta	MAN	1	On Line	1	On Line	
2385425	22	11/09/2012	4:53:41 PM	BOILER-4	BB-4020	B4 Burn2 Sta	MAN	1	On Line	1	On Line	HP2

Figure 4.5: Filtering event logs based on "B4 Burners"

From Figure 4.5, it is evident that 23 events are associated with burner rotation sequence during the start-up operation which has spanned over three hours.

2. **Interpreting the filtered event log**

The filtered event log shown in Figure 4.5 has 13 spreadsheet columns. The significance of these data attributes is discussed below:

- **"Ord" :** A unique system specific identifier assigned to events that are archived chronologically in the PROVOX system. (This data attribute is ignored while performing workflow mining tasks within Disco).

- **"MsgID" :** This column represents the type of event messages associated with each row entry. This data attribute is helpful in differentiating an operator action from an instrument status message.

MsgID	Type of event or action
1	PROVOX Operator Console Revision Notification
10	Approval for connecting or disconnecting Console Data Server
11	Instances when Operator logged on
20, 22	Equipment status messages such as open, close, motion, online and offline
30,31	Operator set point changes E.g. boiler feed pumps

Table 4.2: Type of events or actions in PROVOX event logs

- **"EventDate" :** Date of occurrence of the event
- **"EventTime" :** Actual time stamp of the event
- **"Area" :** Unit or piece of equipment under consideration (E.g. Boiler, turbine, fans, and demineralizers etc.)
- **"PointTag" :** Tag name of the instrument
- **"PointDesc" :** Description of the tag
- **"PvValue" :** Current status of an instrument ('1' represents On Line, '2' represents Off Line)
- **"PvStr" :** Text denoting the current status of an instrument ('On Line' and 'Off Line')
- **"SpValue" :** Setpoint status of an instrument ('1' represents On Line, '2' represents Off Line)
- **"SpStr" :** Text denoting the set-point status of an instrument ('On Line' and 'Off Line' status set by operator)
- **"User" :** Operator authenticity

3. Creating custom event messages

Creating a custom event message string (Custom_Event_Message) is a critical step in the extraction of workflows using Disco [10]. Although the PROVOX system archives all events and alarms associated with boiler operation, any single data attribute (an Excel sheet column) alone does not convey meaningful information of the entire process by itself.

Case_ID	Time_Stamp	PointTag	PointDesc	Custom_Event_Message
1	02:51:17	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(Off Line)
1	02:51:20	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(On Line)
1	02:51:22	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(On Line)
1	02:51:41	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(On Line)
1	02:52:09	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(On Line)
1	02:52:15	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(Off Line)
1	02:52:35	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(Off Line)
1	02:52:38	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(On Line)
1	02:52:41	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(On Line)
1	02:52:50	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(On Line)
1	02:52:56	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(On Line)
1	02:53:07	BB-4020	B4 Burn2 Sta	BB-4020-B4 Burn2 Sta(Off Line)
1	02:53:16	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(Off Line)
1	02:53:32	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(Off Line)
1	02:54:11	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(Off Line)
1	02:54:14	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(On Line)
1	02:54:30	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(On Line)
1	04:22:30	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(On Line)
1	04:22:36	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(On Line)
1	04:24:03	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(On Line)
1	04:53:35	BB-4020	B4 Burn2 Sta	BB-4020-B4 Burn2 Sta(Off Line)
1	04:53:38	BB-4020	B4 Burn2 Sta	BB-4020-B4 Burn2 Sta(On Line)
1	04:53:41	BB-4020	B4 Burn2 Sta	BB-4020-B4 Burn2 Sta(On Line)

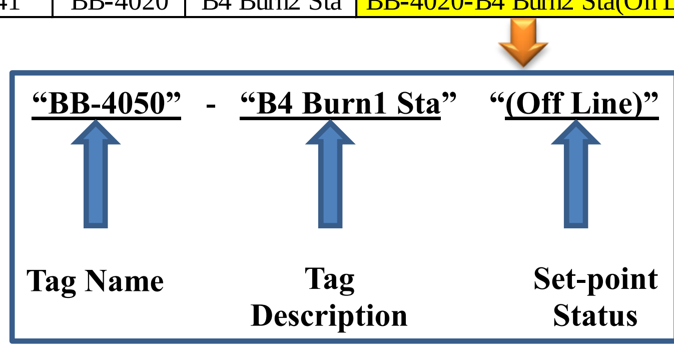


Table 4.3: Assigning case IDs and interpreting a custom event message

It is, therefore, necessary to create unique text messages from the event logs, which will enable workflow mining tools to capture all critical tasks associated with the burner rotation sequence. One way of creating such unique messages is by concatenating the text character strings of each and every event associated with spreadsheet columns “PointTag”, “PointDesc”, and “SpStr” from end-to-end. An example of such a concatenated unique message is shown in Table 4.3. Event messages archived under “SpStr” column take two states namely; ‘On Line’ or ‘Off Line’ which represents the status of burners as set by the control room operator during boiler warm-up operation. Custom_Event_Message strings also help to identify unique actions performed by the operator or control system while rotating the burners. Each of these custom messages represents a workflow activity block. Intrinsic understanding of boiler #4 operation and event logs is essential to create a custom event message. Each custom event message can be deciphered into three components for our analysis as shown in Table 4.3.

4. Case Identification

Case identification is an important data pre-processing step that enables Disco to automatically extract workflows from the event logs. The significance of case identification in extracting sequence of tasks from event messages is illustrated in Section 2.2.1. In order to assign case IDs to the event logs, a workflow “start event” and “end event” are identified as listed in Table 4.4. Start and end events are identified based on process knowledge, standard operating procedure or by consulting an expert operator (refer to Figure 4.3 in Section 4.2).

Case IDs are assigned manually to custom event messages associated with the burner rotation sequence in the order they appear in the event log (often called ongoing or current events) as shown in Table 4.3.

Workflow Start & End Events	Custom Event Message
START	BB-4050-B4 Burn1 Sta(Off Line)
END	BB-4030-B4 Burn3 Sta(Off Line)

Table 4.4: Burner rotation sequence workflow start and end events

A burner rotation sequence during the start-up of boiler #4 is considered one complete "task". According to the standard operating procedure for start-up of boiler #4 (Figure 4.3), each task shall start with the event "BB-4050-B4 Burn1 Sta (Off Line)" and ends with the event "BB-4030-B4 Burn3 Sta (Off Line)"; assuming only first three burners are ignited during a boiler warm-up. However, events associated with four burners (#1, #2, #3, and #4) appear in the event logs during the start-up of boiler #4 on November 09, 2012.

4.2.3 Developing Workflows for Burner Rotation Sequence from Event Logs

Workflows are automatically extracted from event logs using workflow-mining tools such as Disco (Fluxicon Process Laboratories) [10]. "Case ID", "Time_Stamp", and "Custom_Event_Message" are the three mandatory spreadsheet attributes required to generate workflows for a burner rotation sequence using Disco. A step by step procedure to upload an event file and extract workflows using Disco is discussed in Section 2.2.1. Event logs are pre-processed as illustrated in Section 4.2.2, before they can be uploaded into Disco. There are 23 custom event messages associated with burner rotation sequence during the start-up of boiler #4. The workflow shown in Figure 4.6 is extracted from custom event messages which are created based on set-point status of the burners ("SpStr") as described in Table 4.3. A control room operator sets these states when a burner is ignited or shut off during the burner rotation sequence. All the events associated with the burners which are archived under the spreadsheet column "SpStr" take two states "On Line" or "Off Line".

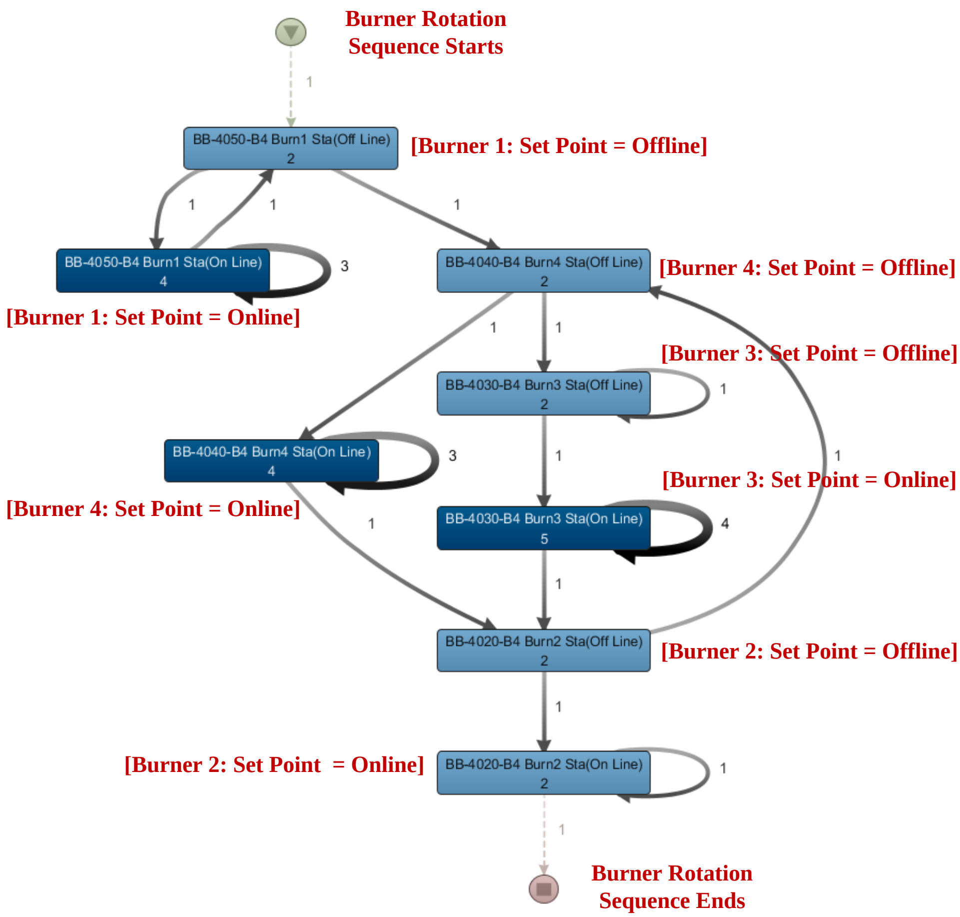


Figure 4.6: Burner rotation workflow developed based on "SpStr" value

A burner rotation workflow developed based on the set-point status of the burners (Figure 4.6) captures operator actions during the burner rotation. According to the workflow shown in Figure 4.6, burner #1 is taken online (ignited) first and then switched to burner #4 while shutting off (offline) burner #1. After igniting burner #4, ignition is switched to burner #3 while burner #4 is shut off. Finally burner #2 is taken online. Warm-up of boiler #4 continues with burner #2 and burner #3 ignited while the boiler reaches operating steam conditions. The prominence of an operator action is indicated by the color-thickness of a block in the workflow diagram. For instance, the custom event message "BB-4030-B4 Burn3 Sta (On Line)" appears five times in the event log which is indicated by a thick-colored block in the workflow diagram as shown in Figure 4.6. Similarly, a directed arrow connecting two blocks indicates the path of actions or events during the burner rotation sequence. A number on the directed arrow indicates the number of times a particular transition occurred between two blocks in the direction of the arrow.

The PROVOX system also archives current status of the burners "PvStr", which takes two states "On Line" or "Off Line" as shown in Figure 4.5. A burner which is currently ignited is indicated by "On Line" whereas "Off Line" indicates a burner which is shut off. Table 4.5 shows custom event messages that are created

Case_ID	Time_Stamp	PointTag	PointDesc	Custom_Event_Message
1	2:51:17 AM	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(On Line)
1	2:51:20 AM	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(On Line)
1	2:51:22 AM	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(On Line)
1	2:51:41 AM	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(Off Line)
1	2:52:09 AM	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(Off Line)
1	2:52:15 AM	BB-4050	B4 Burn1 Sta	BB-4050-B4 Burn1 Sta(Off Line)
1	2:52:35 AM	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(On Line)
1	2:52:38 AM	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(On Line)
1	2:52:41 AM	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(On Line)
1	2:52:50 AM	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(Off Line)
1	2:52:56 AM	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(Off Line)
1	2:53:07 AM	BB-4020	B4 Burn2 Sta	BB-4020-B4 Burn2 Sta(Off Line)
1	2:53:16 AM	BB-4040	B4 Burn4 Sta	BB-4040-B4 Burn4 Sta(Off Line)
1	2:53:32 AM	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(Off Line)
1	2:54:11 AM	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(On Line)
1	2:54:14 AM	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(On Line)
1	2:54:30 AM	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(On Line)
1	4:22:30 AM	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(Off Line)
1	4:22:36 AM	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(Off Line)
1	4:24:03 AM	BB-4030	B4 Burn3 Sta	BB-4030-B4 Burn3 Sta(On Line)
1	4:53:35 AM	BB-4020	B4 Burn2 Sta	BB-4020-B4 Burn2 Sta(On Line)
1	4:53:38 AM	BB-4020	B4 Burn2 Sta	BB-4020-B4 Burn2 Sta(On Line)
1	4:53:41 AM	BB-4020	B4 Burn2 Sta	BB-4020-B4 Burn2 Sta(On Line)

Table 4.5: Customized PROVOX event messages created based on "PvStr"

by concatenating the text character strings of each and every event associated with spreadsheet columns "PointTag", "PointDesc", and "PvStr" from end-to-end.

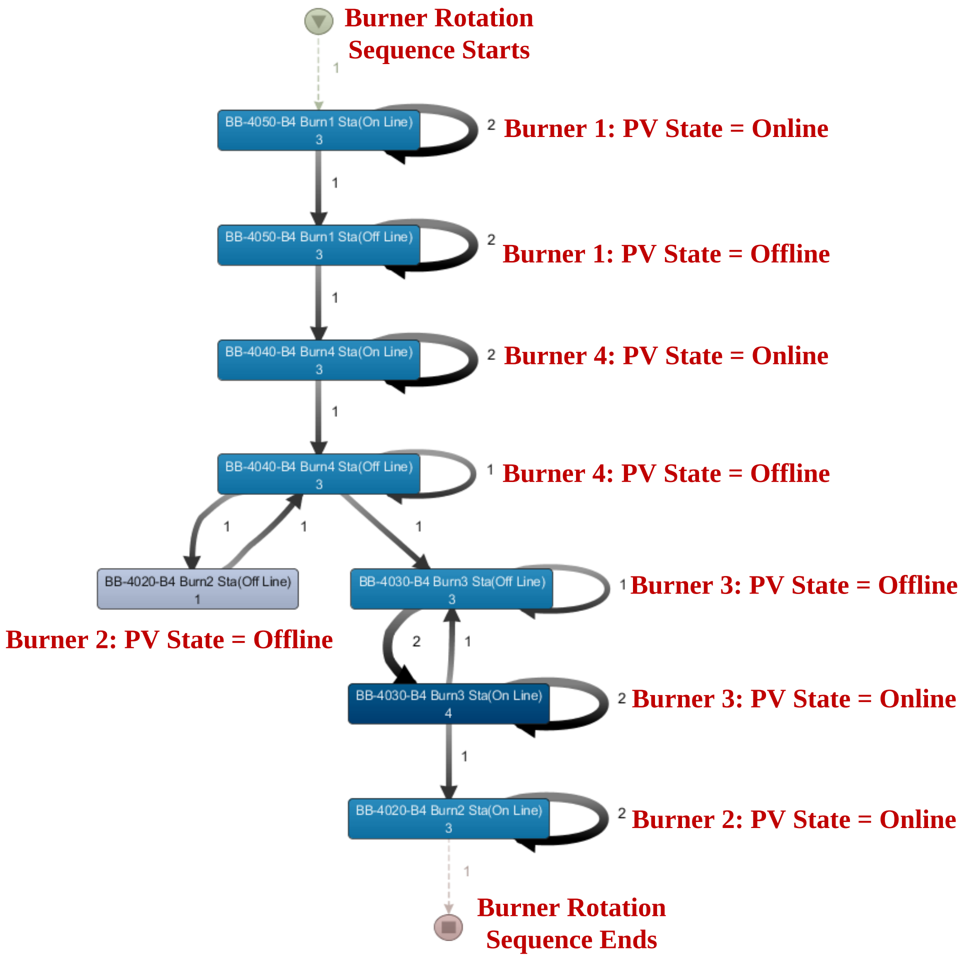


Figure 4.7: Burner rotation workflow developed based on "PvStr" value

Figure 4.7 shows a burner rotation workflow which is extracted from event logs based on the current status of the burners (PvStr). We can infer from this workflow that burner #1 is ignited first and then switched to burner #4 while shutting off burner #1. After igniting burner #4, ignition is switched to burner #3 while burner #4 is shut off. Boiler #4 is brought to operating conditions by continuing with burners #2 and #3. A burner rotation workflow developed based on the current status of the burners (Figure 4.7) captures ongoing events during the burner rotation. Workflows extracted from the event logs of an expert operator capture valuable operational knowledge and can be retained in the control room for training new operators.

4.2.4 Workflow Conformance for Burner Rotation Sequence

According to the standard operating procedure for boiler #4 start-up operation shown in Figure 4.3, it is a recommended practice that the burners which are located in the bottom (#1, #2, and #3) are ignited first during the burner rotation to ensure uniform heating of the boiler components. Especially, superheater headers shall be subjected to slow warm up to prevent hand-hole welds from cracking. This practice of burner rotation sequence also ensures that boiler tubes and superheater headers are not thermally stressed during a warm-up period [46].

Workflow conformance helps supervisors to continuously monitor and compare operator actions and ongoing events with standard operating procedure(s) and identify any procedural violations. Figure 4.8 compares a burner rotation workflow developed from the standard operating procedure with a workflow developed from operator actions. Similarly, it is possible to compare a burner rotation workflow developed from the standard operating procedure with a workflow developed from ongoing events (refer to Figure 4.7). In either of the cases, workflow conformance is instrumental in providing an operational insight into the burner rotation sequence during the start-up of boiler #4. Any deviations observed in the form of non-conforming events (often called a non-conformance workflow) can be further investigated to identify problems associated with human factors, processes, and instrumentation. A workflow that does not conform to the standard operating procedure(s) provides an opportunity for preventive maintenance of the burner components. Figure 4.9 shows an auditable workflow redrawn based on the concept of “Case explorer” associated with Disco software [10]. An auditable workflow provides a detailed information on what actions are completed, by whom, and when. Workflow conformance is able to provide documented information on burner operations in order to meet regulatory compliance [46].

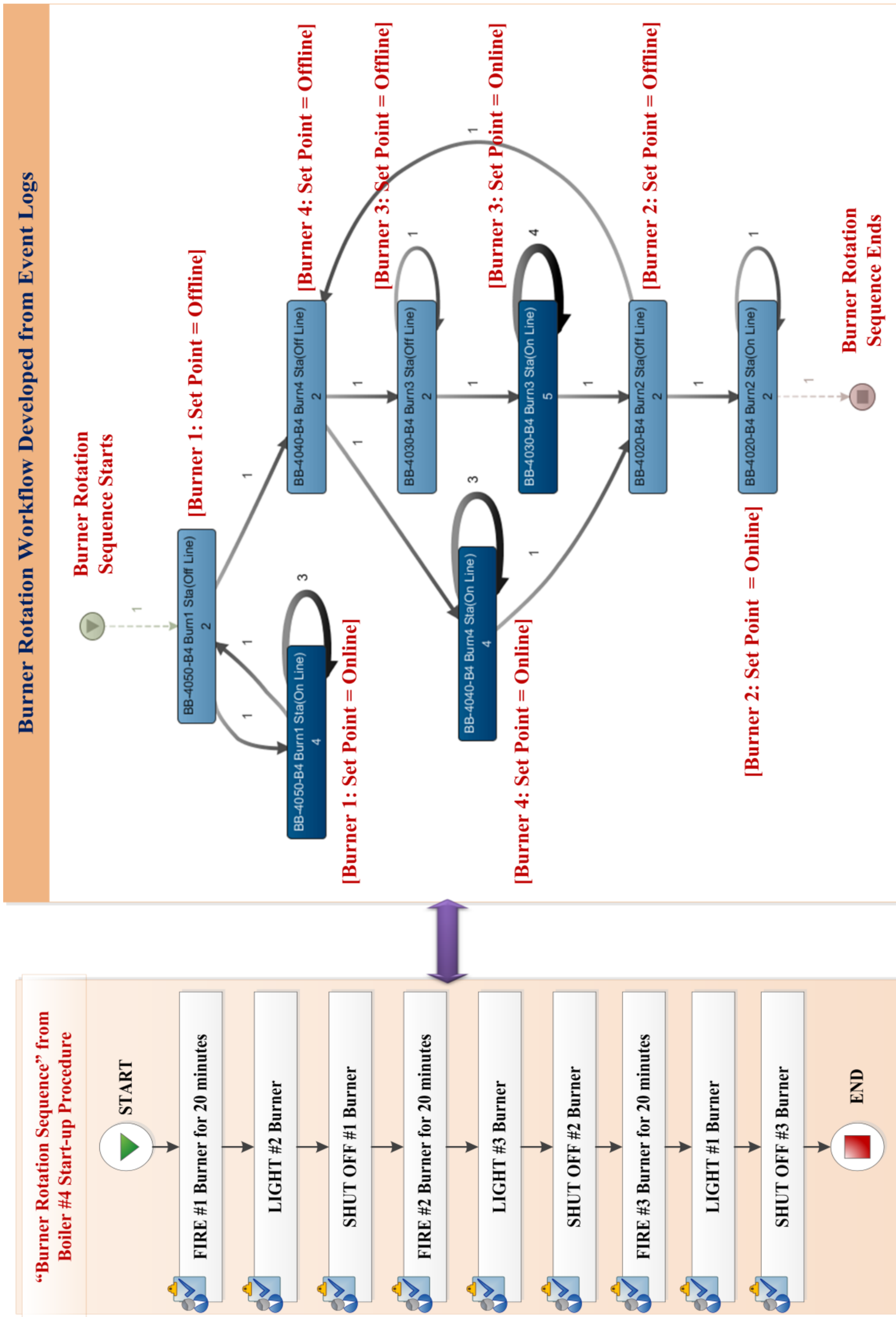


Figure 4.8: Workflow conformance monitoring for burner rotation sequence

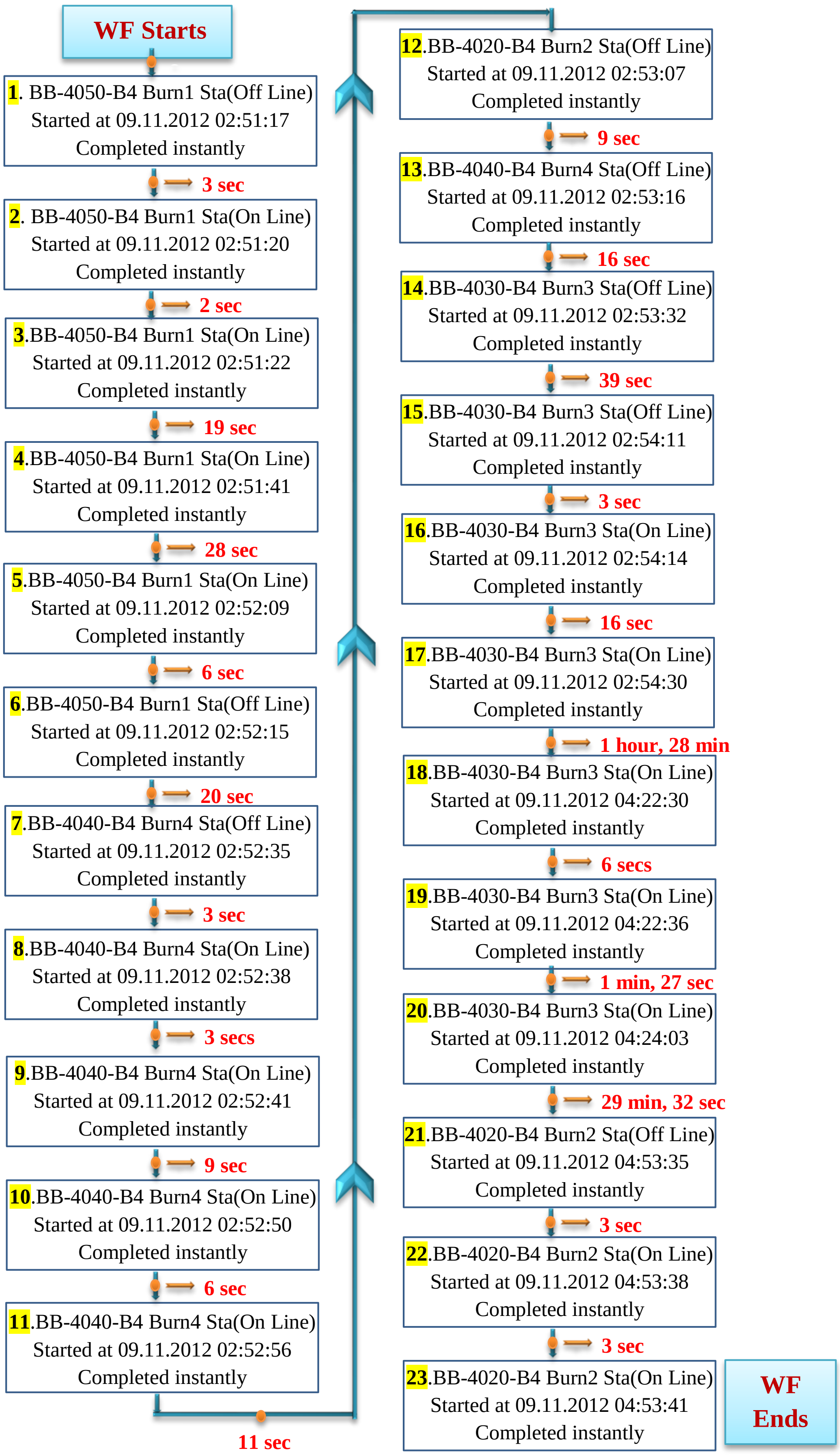
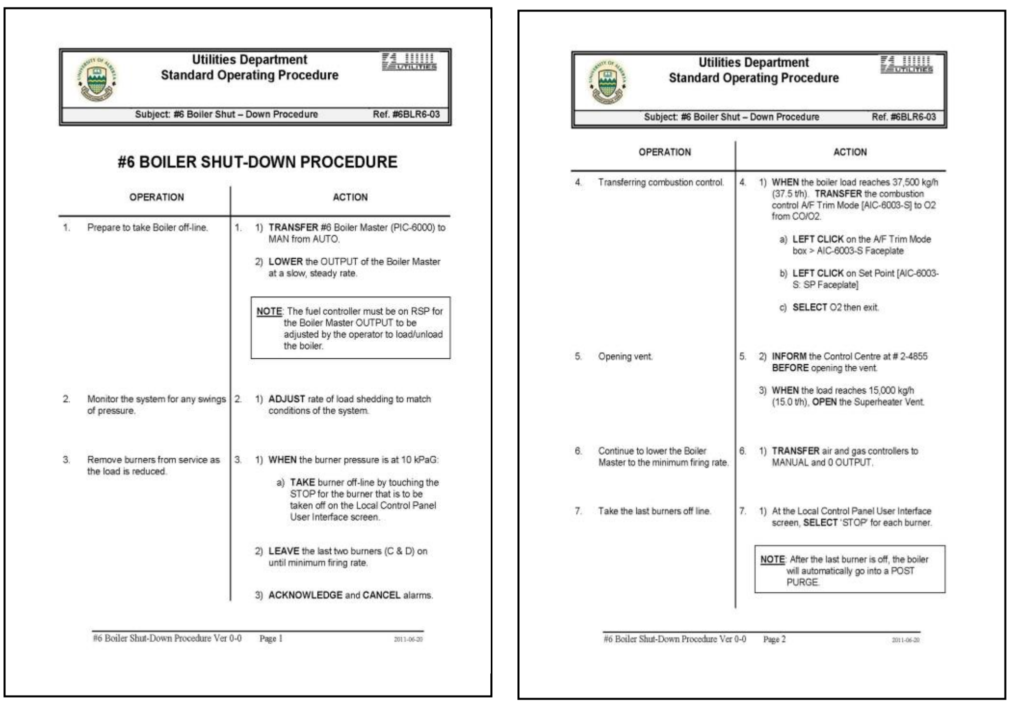


Figure 4.9: Auditable workflow activities for operational insight

4.3 Developing Software-based Workflows for "Boiler #6 Shutdown Procedure" from Paper-based SOPs

A paper-based standard operating procedure for a boiler shutdown operation spans over several pages as shown in Figure 4.10. It is a daunting task for an operator to navigate through paper-based SOPs while handling infrequent situations like boiler start-ups and shutdowns. There is a high probability for a reader to skip or oversee a critical process related information while reading through a lengthy document.



Utilities Department
Standard Operating Procedure

Subject: #6 Boiler Shut – Down Procedure — Ref. #6BLR6-03

#6 BOILER SHUT-DOWN PROCEDURE

OPERATION	ACTION
1. Prepare to take Boiler off-line.	1. 1) **TRANSFER** #6 Boiler Master (PIC-6000) to MAN from AUTO. 2) **LOWER** the OUTPUT of the Boiler Master at a slow, steady rate. NOTE: The fuel controller must be on RSP for the Boiler Master OUTPUT to be adjusted by the operator to load/unload the boiler.
2. Monitor the system for any swings of pressure.	2. 1) **ADJUST** rate of load shedding to match conditions of the system.
3. Remove burners from service as the load is reduced.	3. 1) **WHEN** the burner pressure is at 10 kPaG: a) **TAKE** burner off-line by touching the STOP for the burner that is to be taken off on the Local Control Panel User Interface screen. 2) **LEAVE** the last two burners (C & D) on until minimum firing rate. 3) **ACKNOWLEDGE** and **CANCEL** alarms.

#6 Boiler Shut-Down Procedure Ver 0-0 Page 1 2011-06-20

Utilities Department
Standard Operating Procedure

Subject: #6 Boiler Shut – Down Procedure — Ref. #6BLR6-03

OPERATION	ACTION
4. Transferring combustion control.	4. 1) **WHEN** the boiler load reaches 37,500 kg/h (37.5 t/h). **TRANSFER** the combustion control A/F Trim Mode [AIC-6003-S] to O2 from CO/O2. a) **LEFT CLICK** on the A/F Trim Mode box > AIC-6003-S Faceplate b) **LEFT CLICK** on Set Point [AIC-6003-S: SP Faceplate] c) **SELECT** O2 then exit.
5. Opening vent.	5. 2) **INFORM** the Control Centre at # 2-4855 **BEFORE** opening the vent. 3) **WHEN** the load reaches 15,000 kg/h (15.0 t/h), **OPEN** the Superheater Vent.
6. Continue to lower the Boiler Master to the minimum firing rate.	6. 1) **TRANSFER** air and gas controllers to MANUAL and 0 OUTPUT.
7. Take the last burners off line.	7. 1) At the Local Control Panel User Interface screen, **SELECT** 'STOP' for each burner. NOTE: After the last burner is off, the boiler will automatically go into a POST PURGE.

#6 Boiler Shut-Down Procedure Ver 0-0 Page 2 2011-06-20

Figure 4.10: Snapshot of a paper-based "boiler #6 shutdown procedure" SOP

In this section, we develop a software-based workflow for shutdown procedure associated with boiler #6. Figure 4.11 shows a master-workflow that establishes the main window of the proposed software-based workflow when deployed as a console application using OWBE web services. This shutdown procedure has 11 sub-workflows wherein each sub-workflow contains a sequence of operator actions that shall be performed before moving on to a next sub-workflow. Each sub-workflow is self-explanatory and is hyperlinked so that operators are able to navigate through the entire procedure by clicking each block of sub-workflows.

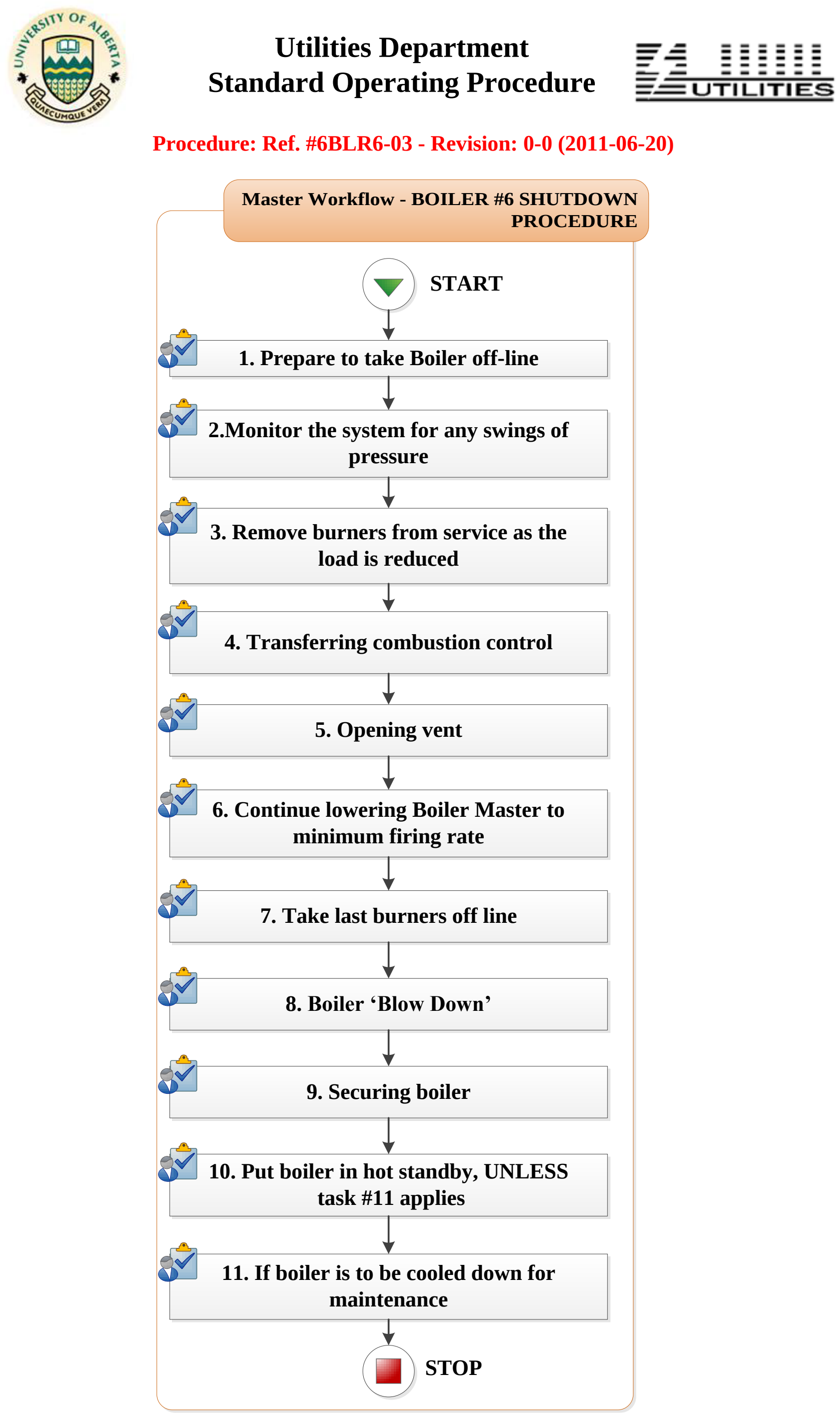


Figure 4.11: Master Workflow- Boiler #6 shutdown procedure

i. Sub-Workflow-1:

The tasks shown in sub-workflow-1 instruct the operators to take over control of boiler master PIC-6000. The boiler master controller (PIC-6000) is changed to MANUAL mode from AUTO as shown in Figure 4.12. This enables operators to manually adjust the boiler master output (lower the firing rate) gradually during a shutdown operation.

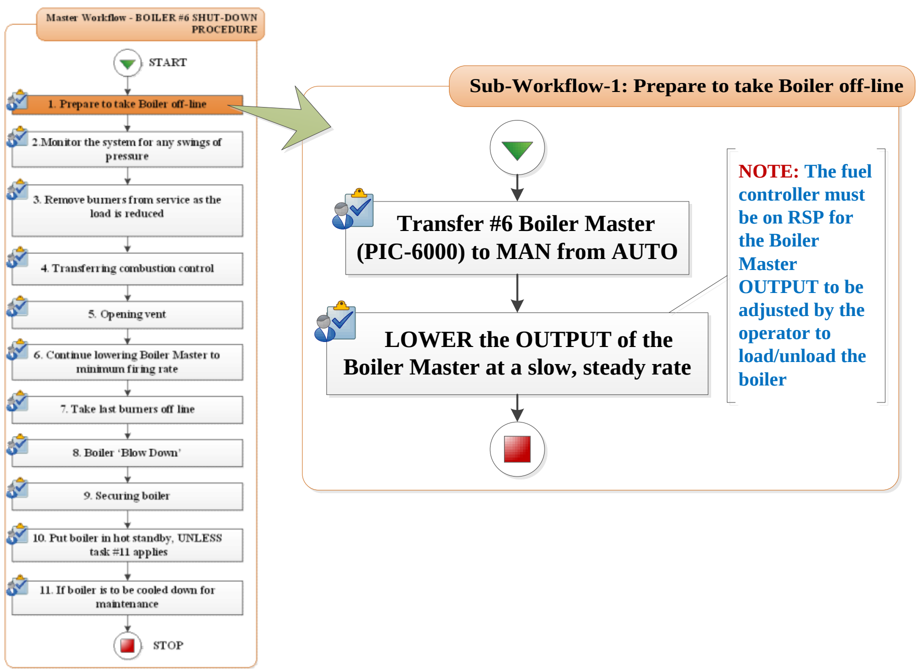


Figure 4.12: Sub-workflow-1: Prepare to take boiler off-line

Operators are able to enter a summary note of all manual set point changes during the shutdown operation. While the boiler master controller is in MANUAL mode, it is essential that the fuel controller (FIC-6001) shall be put in a CASCADE mode (Remote Set Point). This crucial information can be configured as a pop-up alert within sub-workflow-1.

ii. Sub-Workflow-2:

Sub-workflow-2 helps operators to monitor various process variable profiles during a shutdown procedure. The heating plant at the University of Alberta has multiple boilers supplying steam to a common steam header. Based on the demand for steam from the header, the plant master controller adjusts the load among operating boilers. Shutting down an active boiler could cause load fluctuations in other operating boilers. Hence the shutdown operation of boiler#6 should be performed without bumps, such that the rate of load shedding on boiler #6 shall match with the conditions of the plant and total steam load scheduled for that shift.

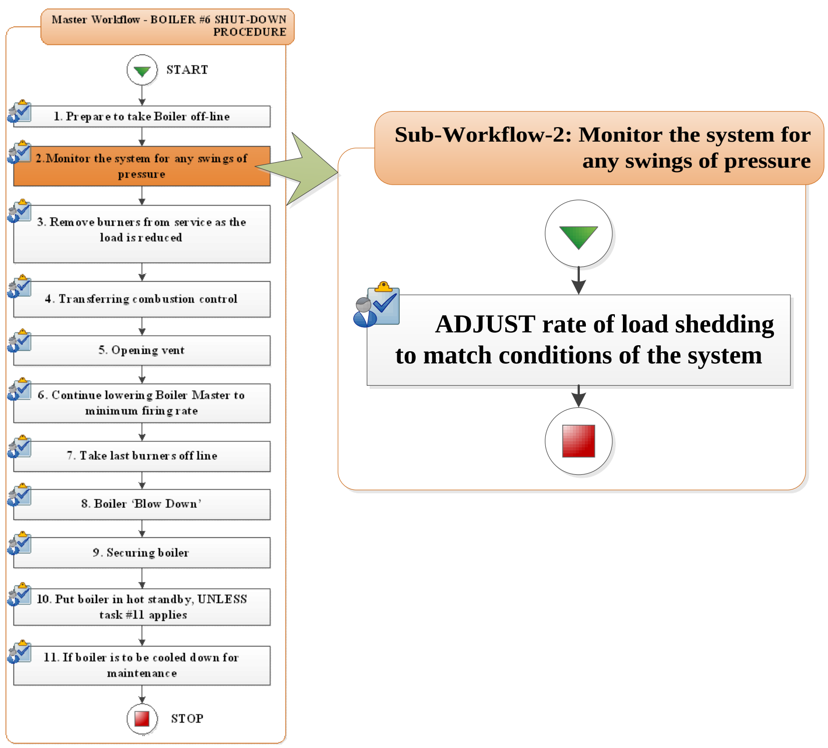


Figure 4.13: Sub-workflow-2: Monitor the system for pressure swings

An experienced operator or a shift supervisor will be able to incorporate the details of load shedding rate, boiler master set points trended over time which can help incoming shift operators understand the past and current plant behavior before they take ownership as shown in Figure 4.13. By integrating real-time process data into sub-workflow-2, notifications can be configured to alert operators if the rate of load shedding is too steep which could cause pressure swings.

iii. Sub-Workflow-3:

The operational tasks listed in sub-workflow-3 guide operators to remove the burners from service as the load is reduced on boiler #6. When the burner gas pressure is at 10kPag, upper burners A and B are shut down. It is a recommended practice that always upper burners are first removed from service while reducing load on the boiler #6.

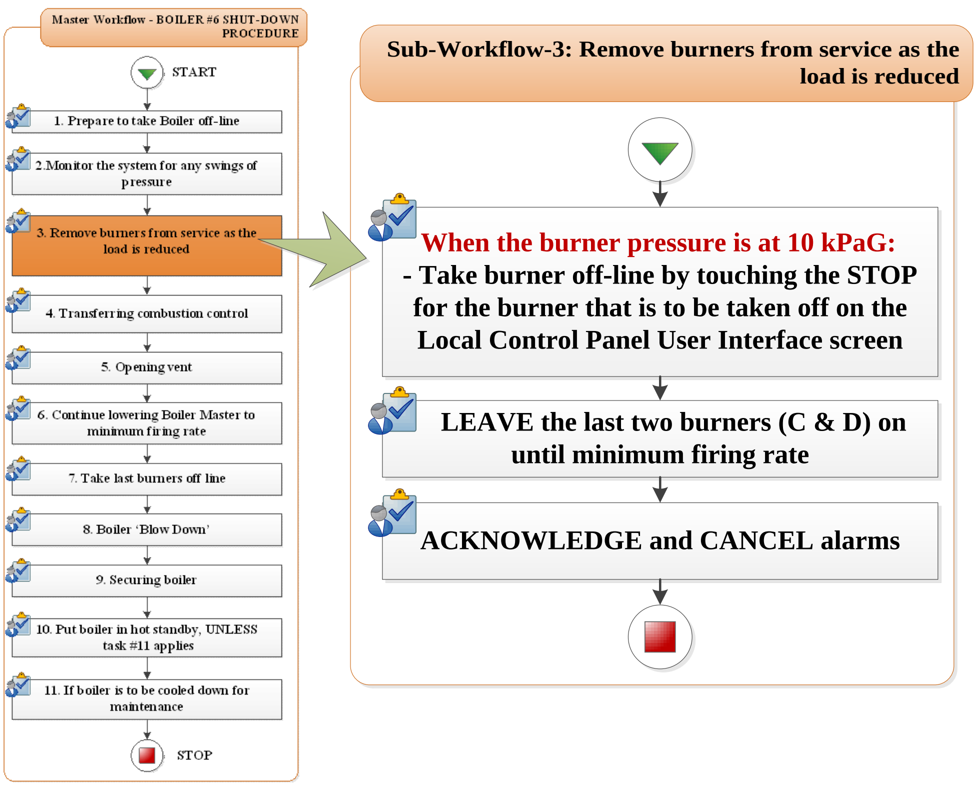


Figure 4.14: Sub-workflow-3: Removing burners from service

A field operator should proceed to the boiler #6 burner control panel to remove burners A and B from service by touching STOP on the local control panel user interface screen. This information has to be communicated and confirmed with the operator in the control room over the phone. When a burner is shut down, the flame scanner senses a loss of flame and de-energizes the Maxon gas valve causing it to close. Alarms associated with flame failure are acknowledged and cancelled. The lower two burners (C, D) are left in service until the minimum firing rate for boiler #6 as shown in Figure 4.14.

iv. Sub-Workflow-4:

For boiler #6, when the load reaches approximately 24% of the maximum capacity (37,500 kg/hr), CO control becomes too erratic. Operational tasks shown in Figure 4.15 assist operators to smoothly transfer combustion control to O_2 mode. As shown in Figure 4.15, it is a recommended practice that operators click on the faceplate of Air-to-Fuel (A/F) Trim Mode box (AIC-6003-S) and select O_2 as the setpoint. O_2 controller is then put in MANUAL mode to adjust O_2 according to the load.

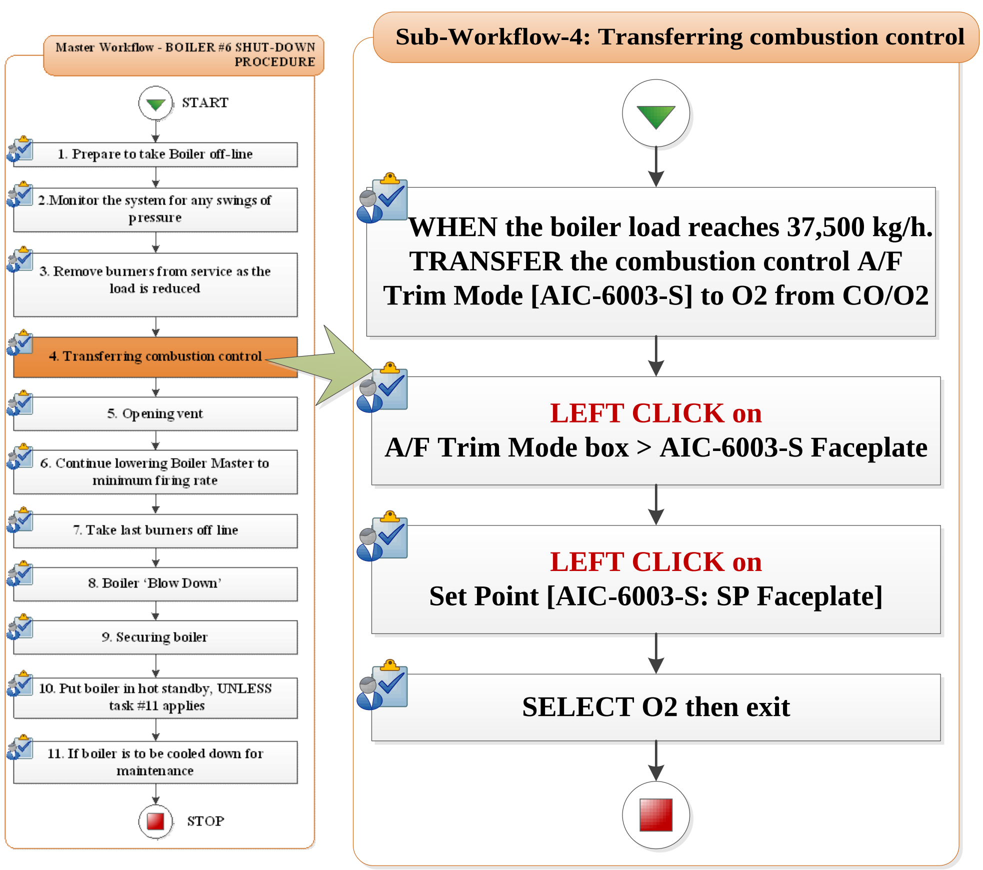


Figure 4.15: Sub-workflow-4: Transferring combustion control

v. Sub-Workflow-5:

As the boiler load decreases during the shutdown operation, superheater vent is opened to avoid overheating of the superheater tubes. Before opening the superheater vent, a field operator shall inform the control room operator at #2-4855 over radio communication as shown in Figure 4.16. Failure to open superheater vent when the load is reduced to 15,000 kg/hr would escalate the failure rate of superheater metal tubes due to high temperatures. During this sequence, verbal communication (over phone) between the field and control room are not captured in the event logs.

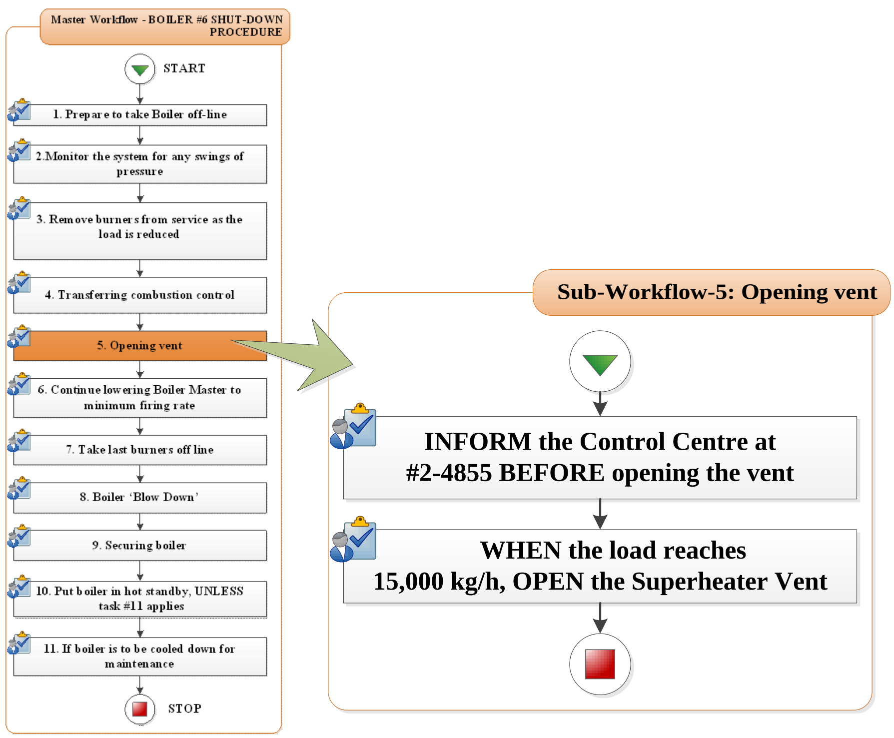


Figure 4.16: Sub-workflow-5: Opening vent

Using sub-workflow-5, control room operators would be able to enter information on abnormal superheater outlet temperatures during their shift; field operators would be able to notify any mechanical problems associated with superheater vent and drain valves and warping of superheater tubes for preventive maintenance when the boiler is offline.

vi. Sub-Workflow-6:

Sub-workflow-6 instructs the operators to transfer air and gas controllers (FIC-6001 and FIC-6002) to MANUAL mode and the outputs of both controllers are set to zero as shown in Figure 4.17.

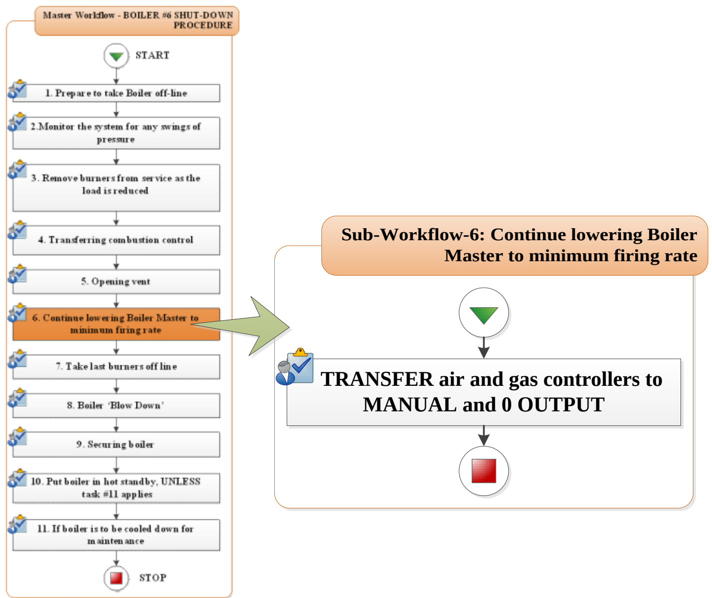


Figure 4.17: Sub-workflow-6: Continue lowering boiler master

vii. Sub-Workflow-7:

When the outputs of air and fuel controllers are set to zero, bottom burners (C, D) are taken offline. The sequence of steps shown in Figure 4.18 helps operators to safely shut down the last burners C and D. When the last two burners have been taken offline, the boiler will automatically go into post purge. After boiler purge, forced draft fan (FD fan) is shut down and superheater vent is closed. It is an important operational sequence to stop an FD fan after boiler purge is complete. Desuperheating spray controller is put in MANUAL mode and the desuperheating upstream valve is closed. All the alarms associated with this operational sequence are acknowledged and cancelled. Alarms grouping status (PPA state) are set to "shutdown".

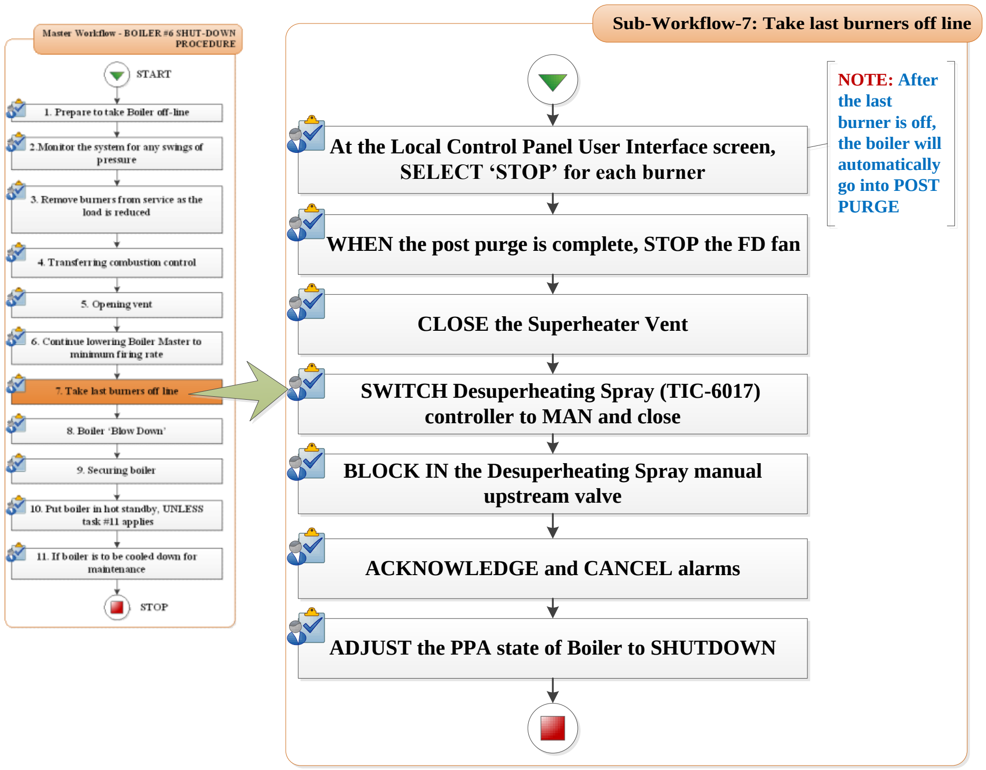


Figure 4.18: Sub-workflow-7: Taking last burners offline

viii. Sub-Workflow-8:

Sub-workflow-8 shows operator actions to blow down the boiler mud drum as shown in Figure 4.19. It is important to note that the blow down of mud drum and sidewall headers should be performed while pressure is still on the boiler to remove sludge.

ix. Sub-Workflow-9:

The operational tasks listed in sub-workflow-9 guide operators to secure the boiler #6 after boiler blow down. Drum level is carefully monitored and drum level is raised to +5 cm. Feedwater flow controller (FIC-6005) is put in MANUAL mode. Continuous blow down valve is closed, phosphate injection pump is stopped, chemical lines at steam drum and pumps are shut down, and main gas valve is closed as shown in Figure 4.20.

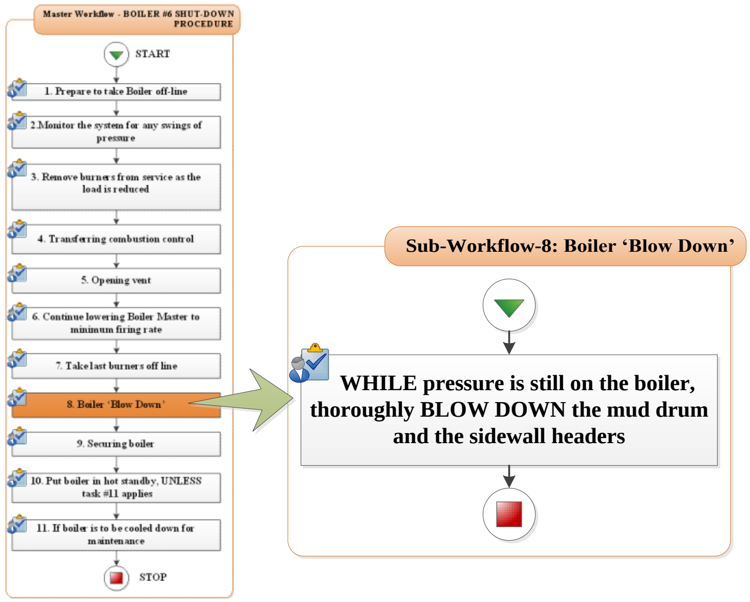


Figure 4.19: Sub-workflow-8: Boiler blow down

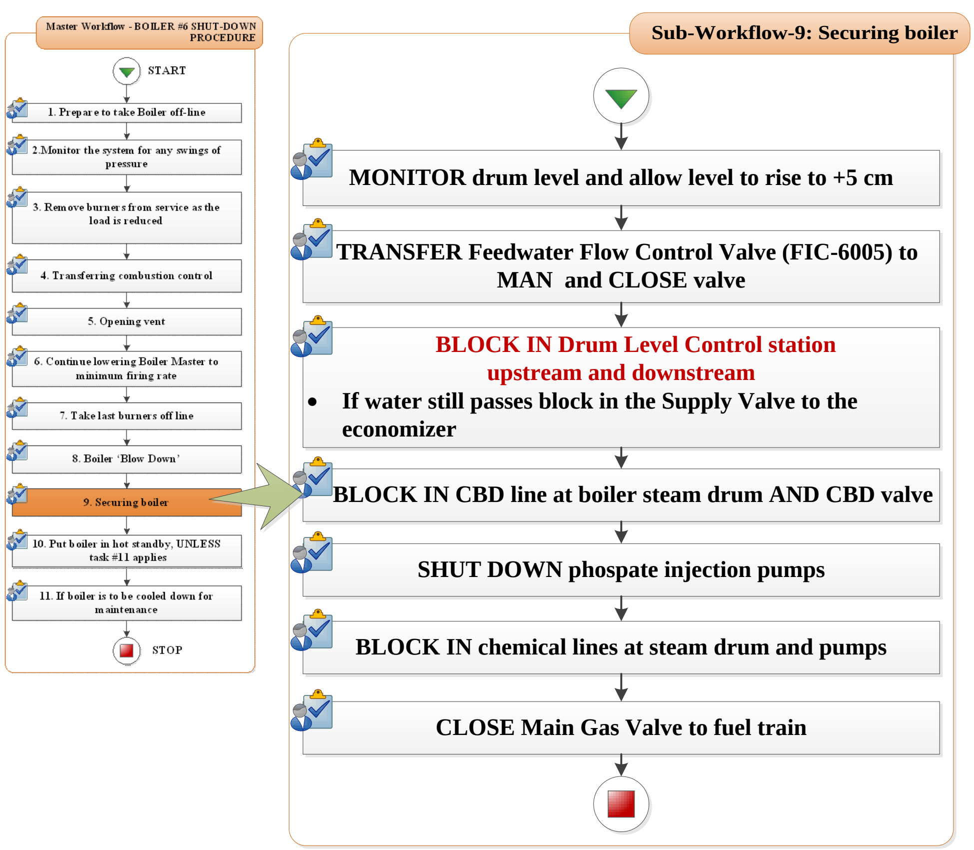


Figure 4.20: Sub-workflow-9: Securing boiler

x. Sub-Workflow-10:

The sequence of operating tasks listed in sub-workflow-10 is effective when boiler #6 is taken for hot standby. The mud drum steam coil supply valves are opened to pressure lines up. This sub-workflow also ensures that the steam trap system is in service as shown in Figure 4.21. (The steam trap system is a provision to remove condensate during a hot standby condition).

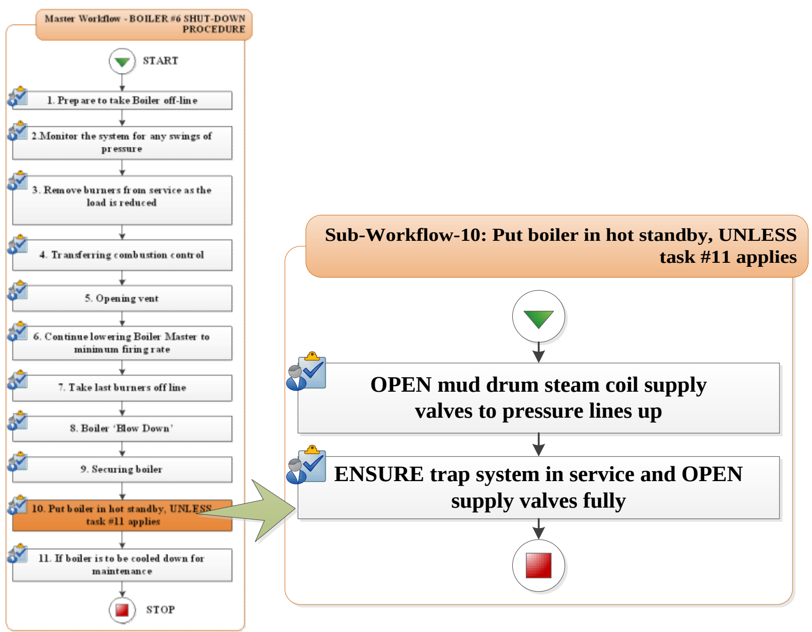


Figure 4.21: Sub-workflow-10: Put boiler in hot standby

xi. Sub-Workflow-11:

Figure 4.22 addresses the steps involved in cooling down boiler #6 if it is shut down for a scheduled maintenance work. This sub-workflow ensures that mud drum steam coils are not in service and the non-return valve is closed so that boiler #6 is completely isolated for maintenance. Drum vent is opened when the steam pressure reaches a safe limit of 170 kPag. Operating tasks listed in sub-workflows 1 through 11 guide operators to safely shut down boiler #6 for a hot standby or a scheduled maintenance.

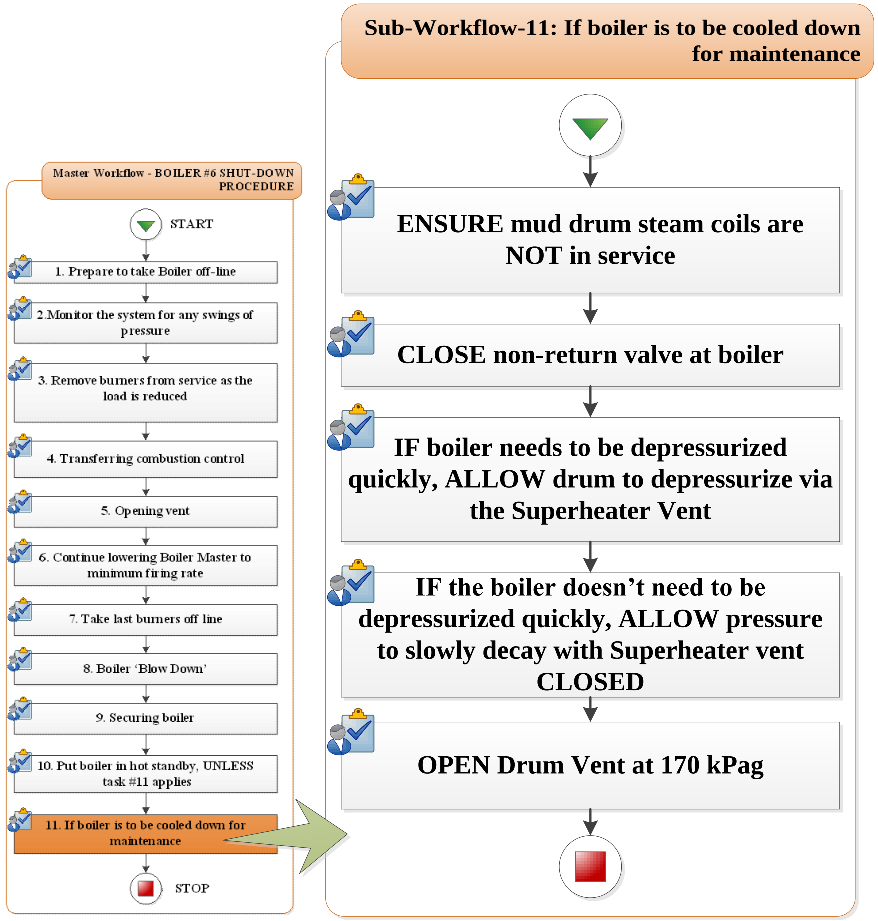


Figure 4.22: Sub-workflow-11: Taking boiler for maintenance

4.3.1 Annotated Workflow Diagrams Developed in Visio Premium 2010

Using annotated workflow diagrams in Visio Premium 2010, an SOP for boiler #6 shutdown procedure is converted into software-based workflows as shown in Figures 4.11 through 4.22. Initially, workflows are developed using SharePoint workflow diagrams in Visio and are exported for configuration in Microsoft SharePoint Designer 2010 [20]. These annotated diagrams show how workflow activities would look like when configured in OWBE and deployed as web service console applications. Workflow activities are validated within the Visio environment before they are exported to SharePoint (refer to Section 2.4.3). An XOML code snippet shown in Figure 4.23 captures all operator actions associated with master-workflow (shown in Figure 4.11) in the form of workflow activities.

```
<ns0:RootWorkflowActivityWithData x:Class="Microsoft.SharePoint.Workflow.ROOT" x:Name="ROOT"
xmlns=http://schemas.microsoft.com/winfx/2006/xaml/workflow
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:ns0="clr
namespace:Microsoft.SharePoint.WorkflowActions;Assembly=Microsoft.SharePoint.WorkflowActions,Version=14.
0.0.0, Culture=neutral, PublicKeyToken=null"> <ns0:RootWorkflowActivityWithData.WorkflowFields>
<ns0:WorkflowDataField Name="__list" Type="System.String" /> <ns0:WorkflowDataField Name="__item"
Type="Microsoft.SharePoint.Workflow.SPItemKey" /> <ns0:WorkflowDataField Name="__context"
Type="Microsoft.SharePoint.WorkflowActions.WorkflowContext" /> <ns0:WorkflowDataField
Name="__initParams" Type="Microsoft.SharePoint.Workflow.SPWorkflowActivationProperties" />
<ns0:WorkflowDataField Name="__workflowId" Type="System.Guid" /><ns0:WorkflowDataField Name="calc"
Type="System.Double" /></ns0:RootWorkflowActivityWithData.WorkflowFields> <ns0:OnWorkflowActivated
WorkflowProperties="{ActivityBind ROOT,Path=__initParams}"
x:Name="ID1"><ns0:OnWorkflowActivated.CorrelationToken> <wf0:CorrelationToken Name="refObject"
OwnerActivityName="ROOT" xmlns:wf0="http://schemas.microsoft.com/winfx/2006/xaml/workflow" />
</ns0:OnWorkflowActivated.CorrelationToken> </ns0:OnWorkflowActivated>
<ns0:ApplyActivation __Context="{ActivityBind ROOT,Path=__context}" x:Name="ID2"
__WorkflowProperties="{ActivityBind ROOT,Path=__initParams}" /><SequenceActivity x:Name="ID3">
<ns0:TodoItemTask x:Name="ID4" ContentTypeId="{x:Null}" Title="{x:Null}" __Context="{ActivityBind
ROOT,Path=__context}" Description="ShapeGuid={0141EF43-6A3E-435D-AF82-
238017BA46BE};PageId=0;ShapeId=1;ShapeText=1. Prepare to take Boiler off-line" AssignedTo="{x:Null}" />
<ns0:TodoItemTask x:Name="ID5" ContentTypeId="{x:Null}" Title="{x:Null}" __Context="{ActivityBind
ROOT,Path=__context}" Description="ShapeGuid={90C41350-8BD2-40EF-9C2A-
17EFF8E2A4E7};PageId=0;ShapeId=11;ShapeText=2.Monitor the system for any swings of pressure"
AssignedTo="{x:Null}" />
<ns0:TodoItemTask x:Name="ID6" ContentTypeId="{x:Null}" Title="{x:Null}" __Context="{ActivityBind
ROOT,Path=__context}" Description="ShapeGuid={A3531B39-500F-4917-9D5C-
1A86B6D02407};PageId=0;ShapeId=14;ShapeText=3. Remove burners from service as the &#xA;load is
reduced" AssignedTo="{x:Null}" />
<ns0:TodoItemTask x:Name="ID7" ContentTypeId="{x:Null}" Title="{x:Null}" __Context="{ActivityBind
ROOT,Path=__context}" Description="ShapeGuid={E340D906-65DE-4CB5-8F74-
97DE0FE9620A};PageId=0;ShapeId=17;ShapeText=4. Transferring combustion control"
AssignedTo="{x:Null}" />
<ns0:TodoItemTask x:Name="ID8" ContentTypeId="{x:Null}" Title="{x:Null}" __Context="{ActivityBind
ROOT,Path=__context}" Description="ShapeGuid={063A77AC-1274-41A5-B71B-
E132E67C891B};PageId=0;ShapeId=7;ShapeText=5. Opening vent" AssignedTo="{x:Null}" />
<ns0:TodoItemTask x:Name="ID9" ContentTypeId="{x:Null}" Title="{x:Null}" __Context="{ActivityBind
ROOT,Path=__context}" Description="ShapeGuid={568F5193-EFD5-4304-B7E4-
D76E6E0E5E47};PageId=0;ShapeId=26;ShapeText=6. Continue lowering Boiler Master to minimum firing
rate" AssignedTo="{x:Null}" />
<ns0:TodoItemTask x:Name="ID10" ContentTypeId="{x:Null}" Title="{x:Null}" __Context="{ActivityBind
ROOT,Path=__context}" Description="ShapeGuid={F5274337-AF17-4D6F-8AFE-
728921D1B41F};PageId=0;ShapeId=23;ShapeText=7. Take last burners off line" AssignedTo="{x:Null}" />
<ns0:TodoItemTask x:Name="ID11" ContentTypeId="{x:Null}" Title="{x:Null}" __Context="{ActivityBind
ROOT,Path=__context}" Description="ShapeGuid={EE02C114-BC1B-4AD5-BD68-
4ECA55195BE9};PageId=0;ShapeId=29;ShapeText=8. Boiler ‘Blow Down’" AssignedTo="{x:Null}" />
<ns0:TodoItemTask x:Name="ID12" ContentTypeId="{x:Null}" Title="{x:Null}" __Context="{ActivityBind
ROOT,Path=__context}" Description="ShapeGuid={5C1D1DA0-9974-4D69-A9A9-
765F9EE2E8A4};PageId=0;ShapeId=32;ShapeText=9. Securing boiler" AssignedTo="{x:Null}" />
<ns0:TodoItemTask x:Name="ID13" ContentTypeId="{x:Null}" Title="{x:Null}" __Context="{ActivityBind
ROOT,Path=__context}" Description="ShapeGuid={EF3163ED-0E01-4BA9-863C-
F2045F6D3F4C};PageId=0;ShapeId=35;ShapeText=10. Put boiler in hot standby, UNLESS&#xA; task #11
applies" AssignedTo="{x:Null}" />
<ns0:TodoItemTask x:Name="ID14" ContentTypeId="{x:Null}" Title="{x:Null}" __Context="{ActivityBind
ROOT,Path=__context}" Description="ShapeGuid={99366758-8858-4294-B799-
88445A5BAF9E};PageId=0;ShapeId=38;ShapeText=11. If boiler is to be cooled down for maintenance"
AssignedTo="{x:Null}" /> </SequenceActivity> </ns0:RootWorkflowActivityWithData>
```

Figure 4.23: A XOML code snippet corresponding to “Master Workflow” annotated diagram developed in Microsoft Visio Premium 2010

4.3.2 Boiler #6 Shutdown Procedure – A Perspective from Control Room

This section provides the details of operator actions as recorded from the control room during a planned shutdown of boiler #6 on April 25, 2013. We witnessed shutdown operation of boiler #6 by sitting next to an operator and noting down all the operational sequences performed (07:58 AM to 09:45 AM), while taking the boiler offline. Boiler #6 can safely handle a steam load of 10:1 turndown ratio (150 tonnes per hour – 15 tonnes per hour). At 07:58 AM, boiler #6 is running at 65 tonnes per hour steam load while preparing for a planned shutdown. Boiler controller (PIC-6000) is the MASTER controller that follows a boiler operation curve to adjust various process variables for safe and economic operation of the boiler. Boiler Master controller (PIC-6000) is put in MANUAL mode to gradually take load off the boiler as shown in the operator console faceplate in Figure 4.25.

Burners are gradually taken offline to reduce boiler load during shutdown operation. Upper burners (A, B) are taken offline first and then gradually lower burners (C, D) are shut down. According to boiler #6 design specifications, when the boiler load reaches a lower value of 37.5 tonnes per hour (approximately 24% of the maximum capacity after shutting down burners A and B), carbon emission (CO) control becomes too erratic, and hence the combustion control "Air-to-Fuel A/F Trim Mode (AIC-6003-S)" is transferred to O_2 SP (set-point) as shown in Figure 4.25. O_2 controller is then put in MANUAL mode to adjust O_2 according to the load.

At 8:45 AM, only two burners (C, D) are online with a steam load of 30 tonnes per hour (FT-6000) and boiler feedwater rate is at 30 tonnes per hour (FIC-6005). When steam flow is reduced, the superheater vent is opened to avoid overheating of the superheater headers. The operator continues to lower boiler MASTER and adjusts gas controllers gradually at a decrement of 2% until the output is 0%. At 9:14 AM, boiler steam load reaches 24 tonnes per hour where the two lower burners (C, D) are taken offline. At 09:17 AM, a field operator manually throttles a valve to open the superheater vent. The forced draft (FD) fan is taken offline and stack louvers are closed while the superheater vent is closed.

While pressure is still on the boiler, the operator performs blowdown of mud drum and sidewall headers at 09:20 AM. In order to secure the boiler, drum level is monitored carefully.

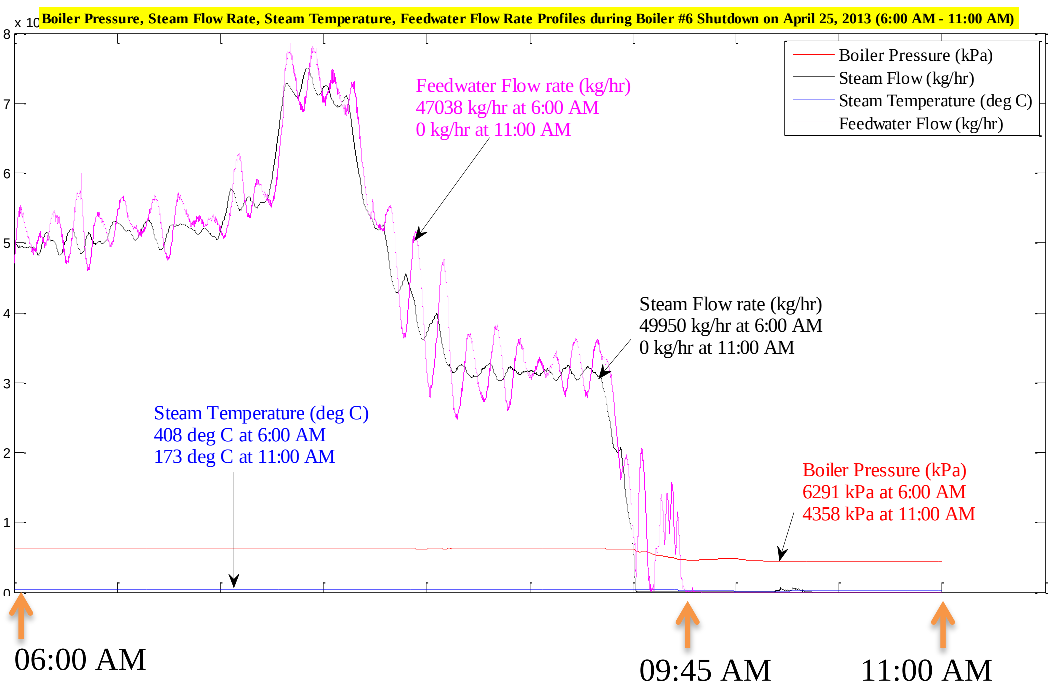


Figure 4.24: Profiles of boiler pressure, steam flow rate, steam temperature, and feedwater flow rate during boiler #6 shutdown operations

The feedwater control valve is closed and then chemical injection pumps are shut down at around 9:37 AM. Burner Management System (BMS) is finally turned off and the main gas valve to the boiler is closed. These tasks ensure that the boiler is safely shutdown and can be taken for maintenance if necessary. By 09:45 AM, boiler #6 is completely offline. Figure 4.24 shows the profiles of boiler pressure (kPa), steam flow rate (kg/hr), steam temperature (deg C), and feedwater flow rate (kg/hr) during the shutdown operation. Figure 4.25 is a snapshot of the operator workstation captured during the shutdown operation.

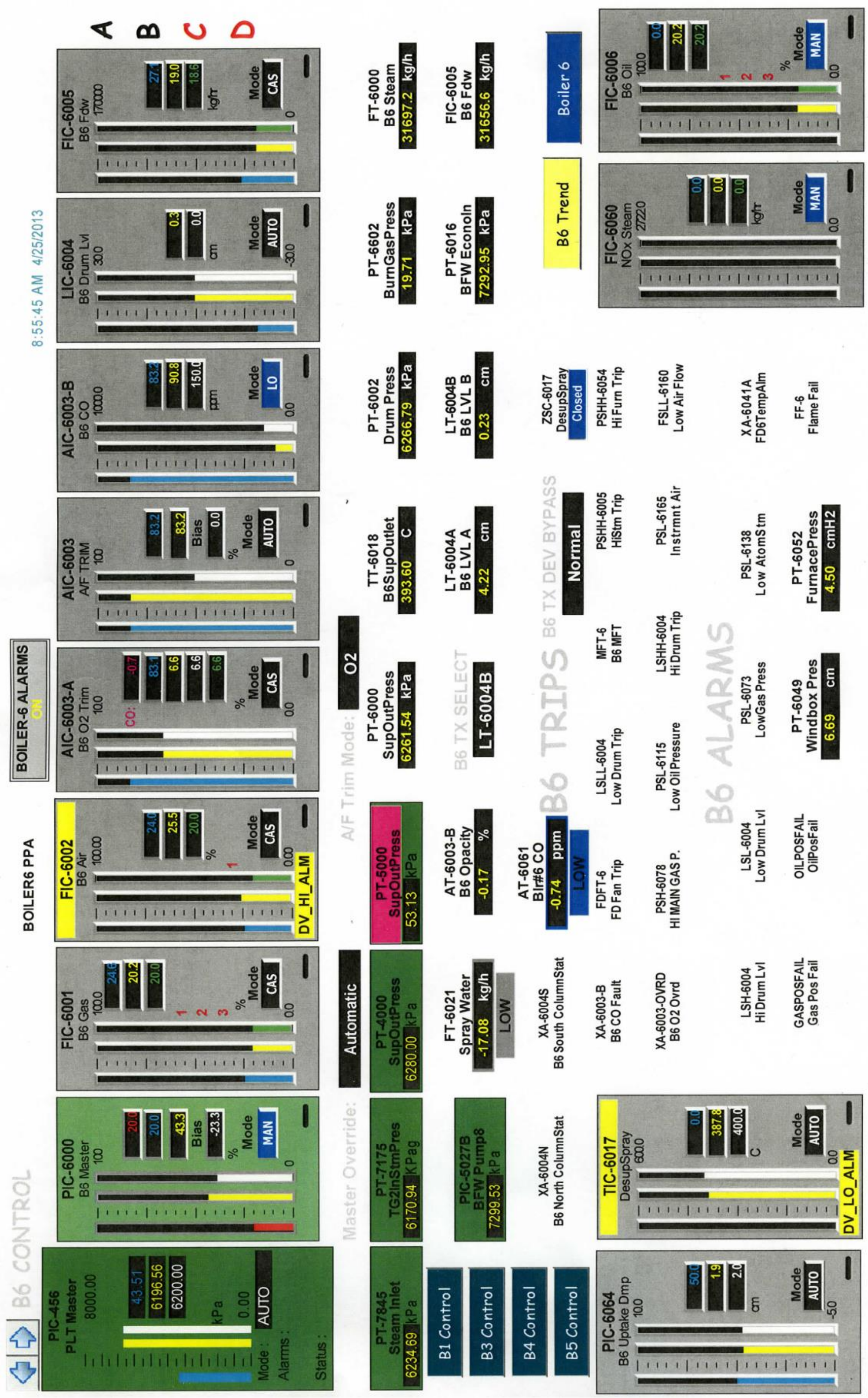


Figure 4.25: Operator console faceplate during Boiler #6 shutdown operation

4.3.3 Information not Captured in a Control Room during Boiler #6 Shutdown Operation

During a boiler shutdown operation, some of the operational tasks depend on verbal communications. For boiler #6 shutdown operation on April 25, 2013, three plant personnel are involved and are constantly communicating over radio (utilities service manager, control room operator and a field operator). Figure 4.26 shows two examples of operator actions performed based on verbal communications over the phone between the field and control room.

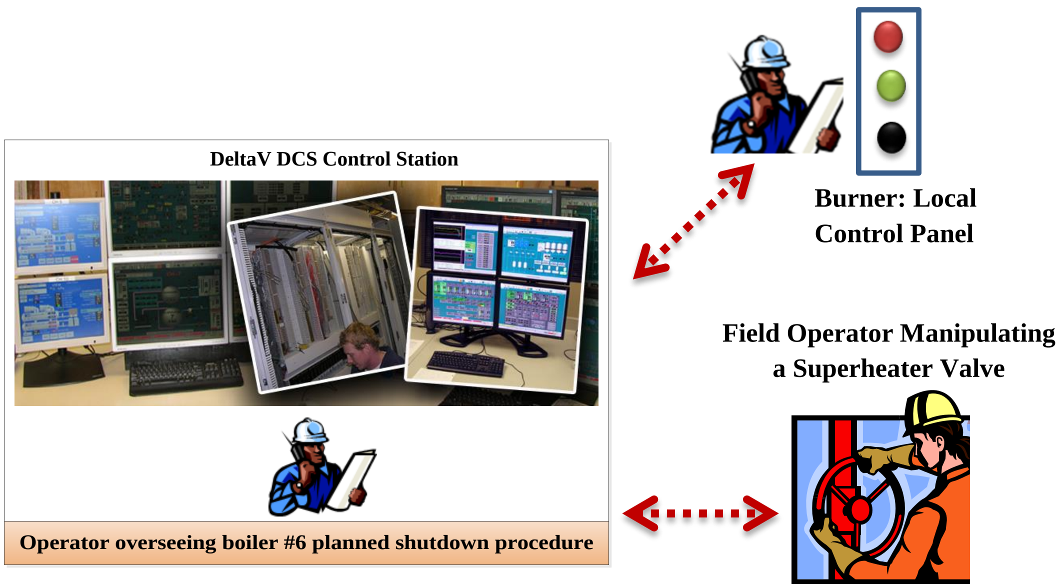
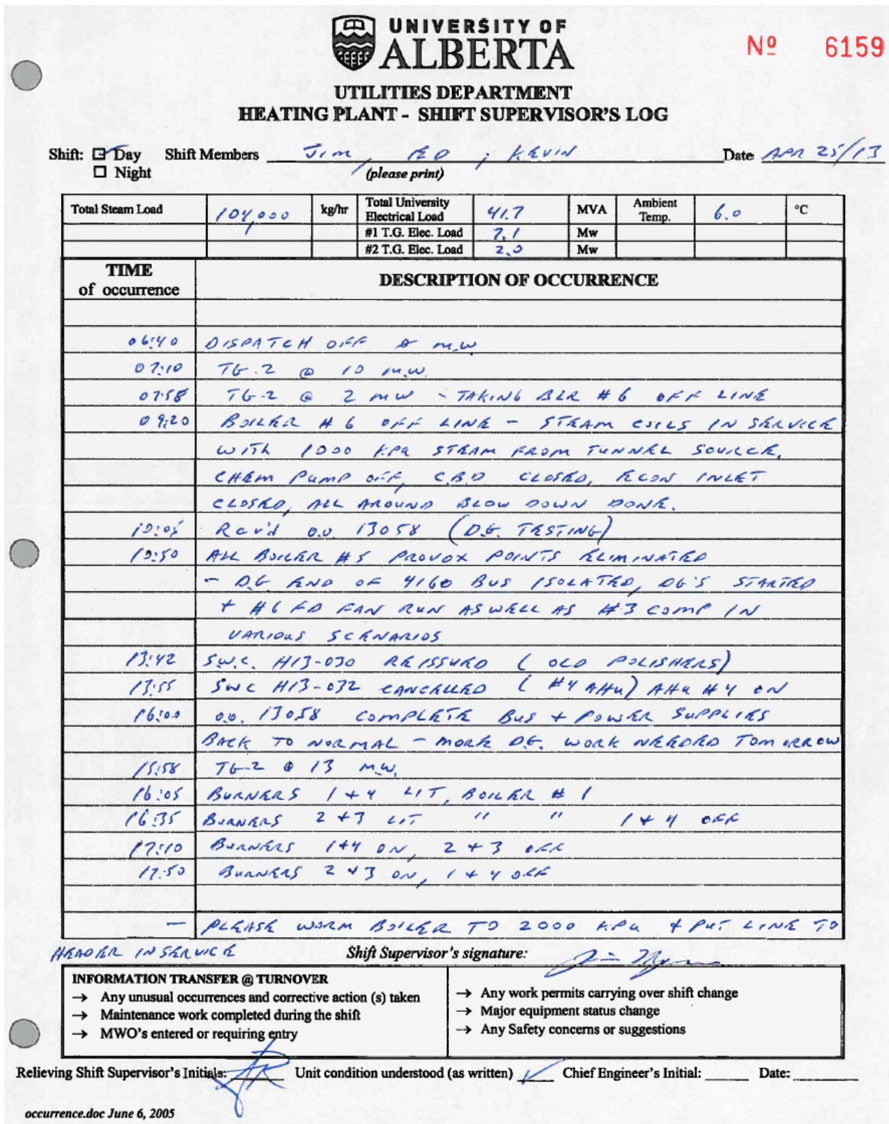


Figure 4.26: Verbal communications over phone during planned shutdown[6]

To shut off a burner, a field operator has to reach the burner supervisory cock and turn it clockwise 90 degrees. Then the operator has to proceed to boiler burner control panel and select "STOP" for each burner at the local control panel user interface. Finally, the purge button has to be activated by the operator. These sequences are communicated to the control room over the phone, so that the control room operator would confirm and acknowledge flame failure alarms. Any unusual occurrences notified and corrective actions undertaken through verbal

[6] Courtesy of Heating Plant Facility Operations at University of Alberta

communications are not captured in the paper-based standard operating procedure. A similar situation is witnessed whenever burners are removed and the steam load is reduced during a boiler shutdown operation. A field operator has to reach to the superheater vent valve and turn it open to avoid overheating of superheater headers. This information has to be communicated to the control room over the phone, before the control room operator would proceed lowering the boiler MASTER further.


UNIVERSITY OF ALBERTA

Nº 6159

UTILITIES DEPARTMENT
HEATING PLANT - SHIFT SUPERVISOR'S LOG

Shift: ☑ Day ☐ Night Shift Members JIM, ED ; KEVIN (please print) Date APR 25/13

Total Steam Load	104,000	kg/hr	Total University Electrical Load	41.7	MVA	Ambient Temp.	6.0	°C
			#1 T.G. Elec. Load	7.1	Mw			
			#2 T.G. Elec. Load	2.0	Mw			

TIME of occurrence	DESCRIPTION OF OCCURRENCE
06:40	DISPATCH OFF Ø MW
07:10	TG-2 @ 10 MW.
07:58	TG-2 @ 2 MW - TAKING BLR #6 OFF LINE
09:20	BOILER #6 OFF LINE - STEAM COILS IN SERVICE
	WITH 1000 KPA STEAM FROM TUNNEL SOURCE,
	CHEM PUMP OFF, C.B.D CLOSED, ECON INLET
	CLOSED, ALL AROUND BLOW DOWN DONE.
10:05	REC'D O.U. 13058 (D.G. TESTING)
10:50	ALL BOILER #5 PROVOX POINTS ELIMINATED
	- D.G. AND OF 4160 BUS ISOLATED, DG'S STARTED
	+ #6 FD FAN RUN AS WELL AS #3 COMP IN
	VARIOUS SCENARIOS
13:42	S.W.C. H13-030 REISSUED (OLD POLISHERS)
13:55	SWC H13-032 CANCELLED (#4 AHU) AHU #4 ON
16:00	O.U. 13058 COMPLETE BUS + POWER SUPPLIES
	BACK TO NORMAL - MORE D.G. WORK NEEDED TOMORROW
15:58	TG-2 @ 13 MW.
16:05	BURNERS 1+4 LIT, BOILER #1
16:35	BURNERS 2+3 LIT " " 1+4 OFF
17:10	BURNERS 1+4 ON, 2+3 OFF
17:50	BURNERS 2+3 ON, 1+4 OFF
—	PLEASE WARM BOILER TO 2000 KPA + PUT LINE TO

HEADER IN SERVICE

Shift Supervisor's signature:

INFORMATION TRANSFER @ TURNOVER
→ Any unusual occurrences and corrective action (s) taken
→ Maintenance work completed during the shift
→ MWO's entered or requiring entry
→ Any work permits carrying over shift change
→ Major equipment status change
→ Any Safety concerns or suggestions

Relieving Shift Supervisor's Initials: AR Unit condition understood (as written) ✓ Chief Engineer's Initial: ______ Date: ________

occurrence.doc June 6, 2005

Figure 4.27: Supervisor's log sheet obtained from the control room

Similarly, several operator actions associated with operations, maintenance work, equipment status changes, any safety concerns which are notified over the phone during a shift is lost when the shift operator leaves the control room or fails to document the events in a shift log sheet (paper-based) as shown in Figure 4.27. This information is very critical for safe and economical operation of the heating plant. OWBE platform helps operators to analyze and capture these actions while they execute their work procedures. Hence, software-based workflows provide an opportunity to document operator actions associated with procedural and verbal communications for auditing and regulatory compliance.

4.4 Summary

In this chapter, workflows are developed from event logs and paper-based SOPs of an industrial case study, namely boiler start-up and shutdown operations at the heating plant at the University of Alberta. This chapter gives a brief introduction to boiler operations and burner rotation sequence. Workflows for the burner rotation sequence are developed from the PROVOX event logs and are compared with the standard operating procedure(s) to check for conformance monitoring. The difference between the two workflows point towards a possible requirement for preventive maintenance. Conformance checking helps plant personnel to identify incipient problems related to the burner management system. The benefits of developing a software-based workflow for a boiler shutdown operation over a paper-based SOP(s) are also demonstrated in this case study.

Chapter 5

OWBE Application

5.1 Introduction

For demonstrating various features and functionalities of web parts developed for OWBE platform, we consider SCADA events associated with the tank-swing process from Suncor Energy Pipeline operations. Detailed explanation of tank-swing process and the procedure for pre-processing of associated SCADA event log data is illustrated in Section 3.4. Table 5.1 shows pre-processed events corresponding to tank-swing process at Denver receiving station during the month of May 2012. A tank-swing process typically involves six characteristic events; namely "**issuing a batch ID**", " **issuing a head select command**", "**tank swing warning**", "**valve manipulation**", "**low or high gravity alarm**", and "**ticket cut command**".

SCADA_ TIME_ STAMP	ALARM_ MESSAGE	EVENT_ MESSAGE	STN	STATION POINT	TAG
28-05-2012 10:54:14	ON	CMD142.OMNI.BTCH .ID.SEQ.	DNVR	DNVR,CMD 142	CMD1 42
28-05-2012 10:54:55	START FROM SEQ14	MTR 1 HEAD SELECT CMD	DNVR	DNVR,M1H CMD	M1HC MD
28-05-2012 10:55:15	HIHIHI 910.0 9	DNVR TK SWING WARNING	DNVR	DNVR,M1A CC1	M1 ACC1
28-05-2012 11:12:04	OPEN	VALVE 08 TANK 775 STAT	DNVR	DNVR, V080	V080
28-05-2012 11:12:04	TRAVEL	VALVE 10 TANK 778 STAT	DNVR	DNVR, V080	V080
28-05-2012 11:12:37	CLOSED	VALVE 10 TANK 778 STAT	DNVR	DNVR, V10O	V10O
28-05-2012 14:11:19	LO 36.0 36.0	IN STN GRAV MTR1 SUNCOR	DNVR	DNVR, M1GRV	M1 GRV
28-05-2012 14:13:50	START FROM SPCC-	MTR#1 TICKET CUT CMD	DNVR	DNVR, BLNCH1	BLNC H1

Table 5.1: SCADA events associated with the tank-swing process

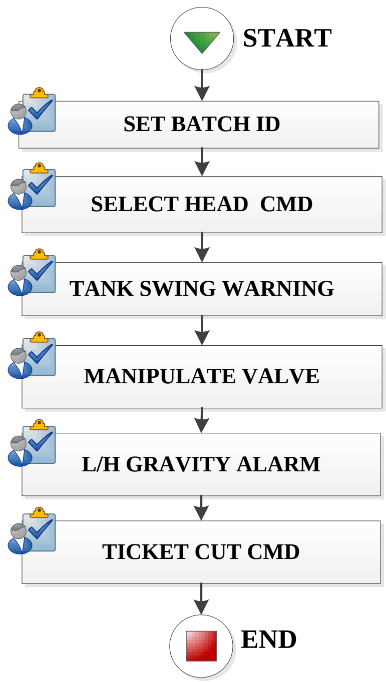


Figure 5.1: Tank-swing workflow developed from "Denver Batch Cut" SOP

Figure 5.1 shows a simplified workflow that represents the sequence of events, also called a "golden path of operation" associated with the tank-swing process. Using OWBE User Interface (UI), users would be able to filter and select events of interest based on attributes like date, time, tag, station, pipeline number, or geographical location as long as these data fields are archived into the event log database. Users will then drag and drop pertinent events into the workflow design surface to construct a tank-swing workflow.

The OWBE platform essentially contains four functional web parts that communicate seamlessly to accomplish a task of workflow creation. OWBE will be assembled from this set of web parts, hosted within a webpage. "Event Loader" and "Event Filter" web parts are developed by Ping Duan and Yue Cheng respectively at the University of Alberta. Kenan Kigunda and Zihao Huang developed "Event Trigger" and "Workflow Document Library" web parts which save workflows within OWBE platform. These four web parts are assembled in the OWBE architecture diagram as shown in Figure 5.2. With regard to the implementation, a user will run through four important steps. Step 1 involves pre-processing of event log file into a list of activities understood by the user. This is done at spreadsheet level before the event logs are uploaded into OWBE platform (refer to Section 3.4). Step 2 represents the event loader which uploads pre-processed event log file into OWBE platform. Step 3 corresponds to filtering of relevant events chosen by the user, based on data attributes and creating workflows within the workflow design surface. Step 4 enables users to save filtered events or created workflows as XML files in the workflow document library (SharePoint 2010) as shown in Figure 5.2.

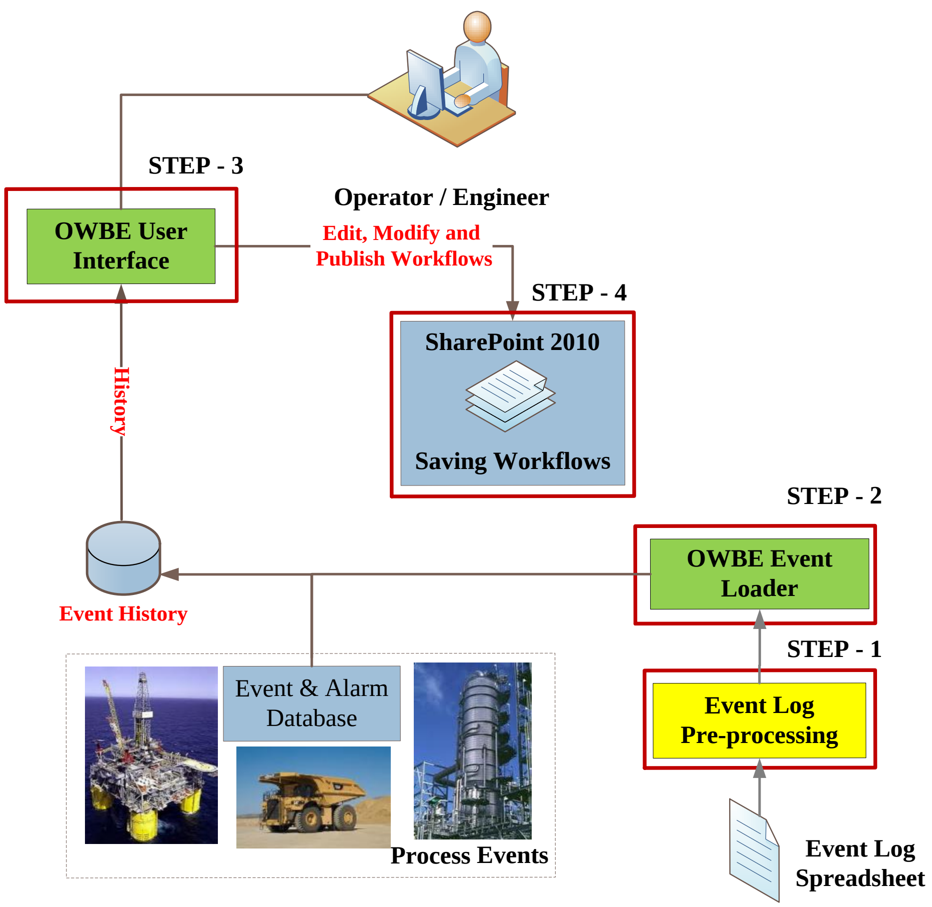


Figure 5.2: OWBE architecture diagram with developed web parts

Following steps demonstrate how to work with the developed OWBE platform, starting from loading an event log file and ending with saving workflows as XML files in the workflow document library.

- Step 1 (Pre-processing of Event Log Data):
  Firstly, SCADA event log data is pre-processed as discussed in Section 3.4. Data pre-processing is performed in an Excel file, outside of OWBE environment. Date, timestamps and event messages are used as filtering attributes to pre-process relevant events corresponding to tank-swing process at Denver receiving station as shown in Table 5.1.

- Step 2 (Event Loader and RDF Graph):
  A pre-processed event log file is uploaded into OWBE platform using "Event Loader" dashboard as shown in Figure 5.4. Excel sheet number or sheet name should be specified while uploading an Excel file. Uploaded Excel files are saved to SQL databases as named RDF graphs specified in "Graph" and "Graphview" dashboard.
- Step 3 (Filtering):
  "Event Filter" web part has a calendar control dashboard to select events based on timestamps. A user selectable filter includes provision for including and excluding event messages as shown in Figure 5.5.

- Step 4 (Save Workflows):
  Filtered events and workflows created are saved as XML files in the Workflow Document Library as shown in Figures 5.10 and 5.11.

5.2 OWBE Home Page

OWBE is deployed as a web service (console application) where users are able to access the OWBE platform by providing appropriate username and password authentication. Snapshot of OWBE's User Interface (UI) is shown in Figures 5.3.

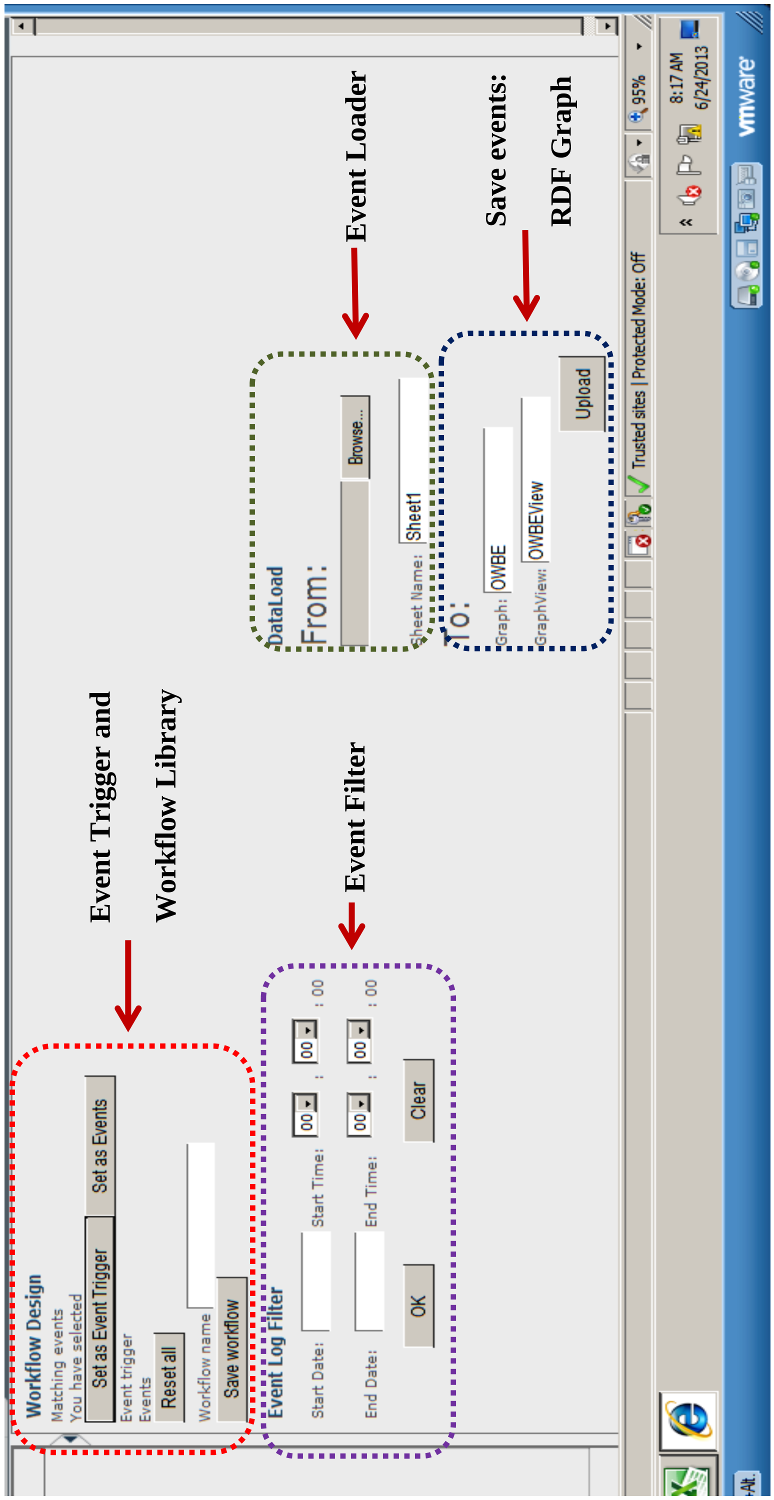


Figure 5.3: OWBE web parts assembled within OWBE webpage

5.3 Importing Excel File into OWBE Platform (Step – 2)

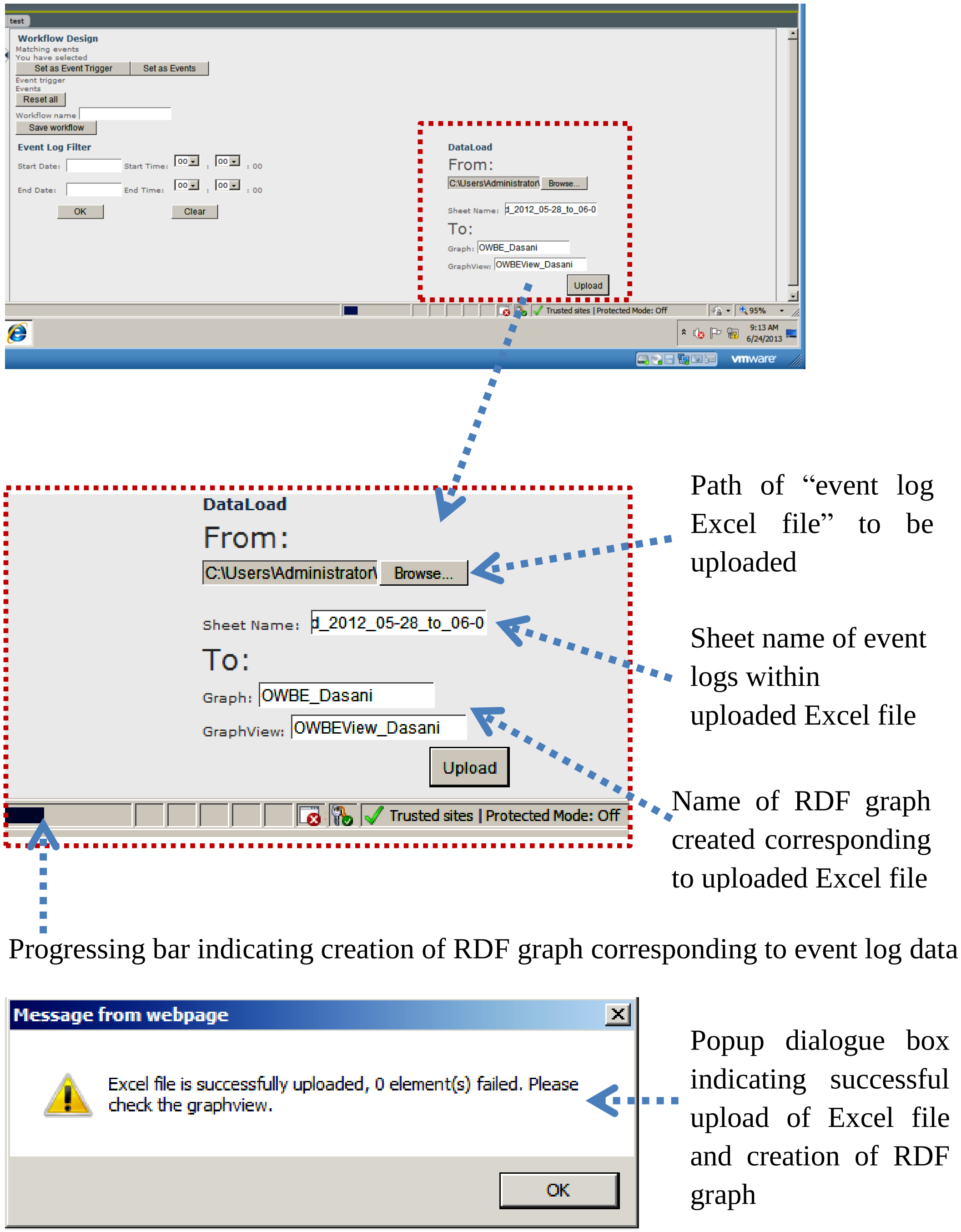


Figure 5.4: Importing Excel file and saving it as "RDF graph"

5.4 Filtering Events with Event Filter (Step – 3)

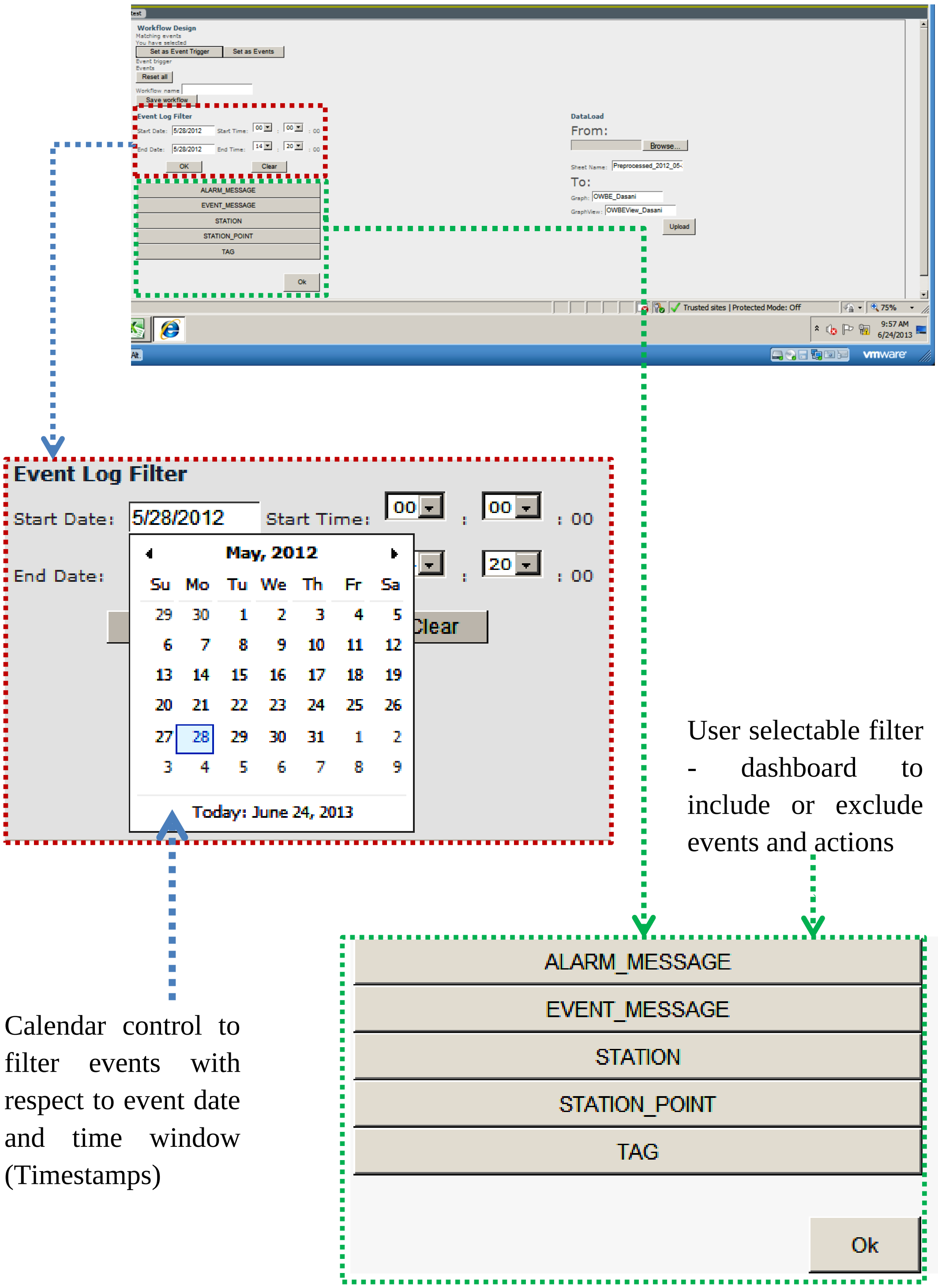


Figure 5.5: Using calendar control and event filter

A calendar control dashboard included in “Event Filter” web part enables users to select events based on timestamps. Start time and end time can be specified to select the relevant events. Once the user selects a time range of events, all event attributes (columns in an Excel file) are shown in the user interface (UI) as shown in Figure 5.6. A user selectable filter is built in this web part to include and exclude events which are of interest in order to further reduce the search space. In our running example, the events shown in Table 5.1 are included in developing tank-swing workflow. Figure 5.6 shows all filtered events corresponding to the tank-swing process which occurred between 0:00 to 14:20 on May 28, 2012.

Workflow Design

Matching events

	SCADA_TIME_STAMP	ALARM_MESSAGE	EVENT_MESSAGE	STATION	STATION_POINT	TAG
Select	2012-05-28T10:54:14	ON	CMD142.OMNI.BTCH.ID.SEQ.	DNVR	DNVR,CMD142	CMD142
Select	2012-05-28T10:54:55	START FROM SEQ14	MTR 1 HEAD SELECT CMD	DNVR	DNVR,M1HCMD	M1HCMD
Select	2012-05-28T10:55:13	YES	MTR 1 HEAD SELECT CMD	DNVR	DNVR,M1HCMD	M1HCMD
Select	2012-05-28T10:55:15	HI 803.0 800.0	DNVR TK SWING WARNING	DNVR	DNVR,M1ACC1	M1ACC1
Select	2012-05-28T10:55:15	HIHI 862.0 850	DNVR TK SWING WARNING	DNVR	DNVR,M1ACC1	M1ACC1
Select	2012-05-28T10:55:15	HIHIHI 910.0 9	DNVR TK SWING WARNING	DNVR	DNVR,M1ACC1	M1ACC1
Select	2012-05-28T10:55:53	NO	MTR 1 HEAD SELECT CMD	DNVR	DNVR,M1HCMD	M1HCMD
Select	2012-05-28T10:56:13	STOP FROM SEQ14	MTR#1 TICKET CUT CMD	DNVR	DNVR,BLNCH1	BLNCH1
Select	2012-05-28T10:56:17	NO	MTR#1 TICKET CUT CMD	DNVR	DNVR,BLNCH1	BLNCH1
Select	2012-05-28T10:56:17	OFF	CMD142.OMNI.BTCH.ID.SEQ.	DNVR	DNVR,CMD142	CMD142
Select	2012-05-28T10:56:17	ON	CMD142.OMNI.BTCH.ID.SEQ.	DNVR	DNVR,CMD142	CMD142
Select	2012-05-28T10:56:17	STOP FROM SEQ14	MTR#1 TICKET CUT CMD	DNVR	DNVR,BLNCH1	BLNCH1
Select	2012-05-28T10:56:20	OFF	CMD142.OMNI.BTCH.ID.SEQ.	DNVR	DNVR,CMD142	CMD142
Select	2012-05-28T11:04:31	HIHI 28.0 28	IN STN GRAV MTR1 SUNCOR	DNVR	DNVR,M1GRV	M1GRV
Select	2012-05-28T11:04:31	HIHIHI 29.1	IN STN GRAV MTR1 SUNCOR	DNVR	DNVR,M1GRV	M1GRV
Select	2012-05-28T11:11:00	HI 801.0 800.0	DNVR TK SWING WARNING	DNVR	DNVR,M1ACC1	M1ACC1
Select	2012-05-28T11:11:13	OPEN FROM SPCC-	VALVE 08 TANK 775 STAT	DNVR	DNVR,V08O	V08O
Select	2012-05-28T11:11:19	TRAVEL	VALVE 08 TANK 775 STAT	DNVR	DNVR,V08O	V08O
Select	2012-05-28T11:12:00	HIHI 853.0 850	DNVR TK SWING WARNING	DNVR	DNVR,M1ACC1	M1ACC1
Select	2012-05-28T11:12:04	OPEN	VALVE 08 TANK 775 STAT	DNVR	DNVR,V08O	V08O

Figure 5.6: Filtering events based on timestamps and attributes

Tank-swing workflow starts with event “CMD142.OMNI.BTCH.ID.SEQ” and is set as “Event Trigger” as it initiates tank-swing process. By clicking “Set as Event Trigger” button in Figure 5.7, any event can be set as a trigger.

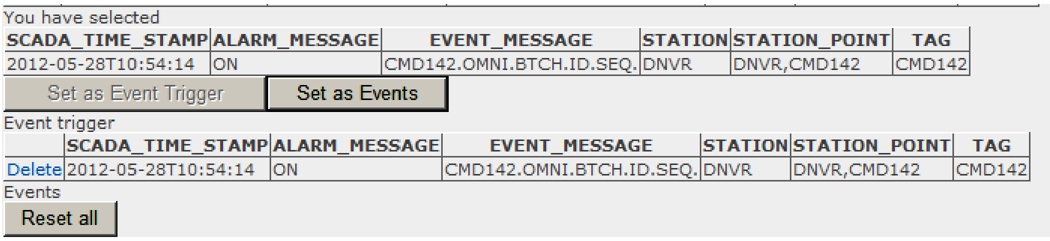
You have selected

SCADA_TIME_STAMP	ALARM_MESSAGE	EVENT_MESSAGE	STATION	STATION_POINT	TAG
2012-05-28T10:54:14	ON	CMD142.OMNI.BTCH.ID.SEQ.	DNVR	DNVR,CMD142	CMD142

Set as Event Trigger | Set as Events

Event trigger

Figure 5.7: Setting an event as a “trigger”


You have selected

SCADA_TIME_STAMP	ALARM_MESSAGE	EVENT_MESSAGE	STATION	STATION_POINT	TAG
2012-05-28T10:54:14	ON	CMD142.OMNI.BTCH.ID.SEQ.	DNVR	DNVR,CMD142	CMD142

Set as Event Trigger | Set as Events

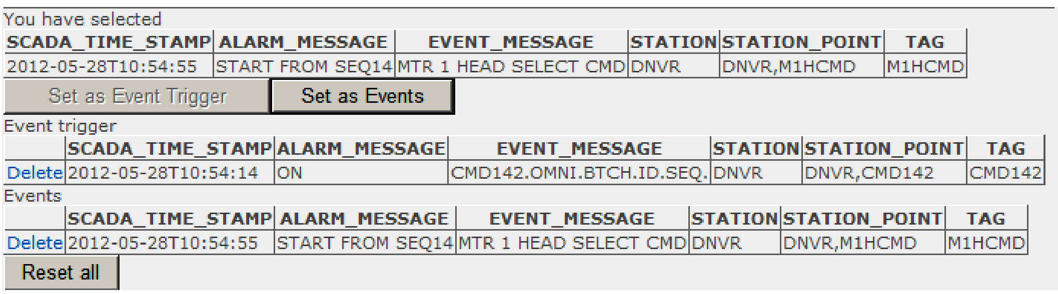
Event trigger

	SCADA_TIME_STAMP	ALARM_MESSAGE	EVENT_MESSAGE	STATION	STATION_POINT	TAG
Delete	2012-05-28T10:54:14	ON	CMD142.OMNI.BTCH.ID.SEQ.	DNVR	DNVR,CMD142	CMD142

Events

Reset all

Figure 5.8: Saving an event trigger


You have selected

SCADA_TIME_STAMP	ALARM_MESSAGE	EVENT_MESSAGE	STATION	STATION_POINT	TAG
2012-05-28T10:54:55	START FROM SEQ14	MTR 1 HEAD SELECT CMD	DNVR	DNVR,M1HCMD	M1HCMD

Set as Event Trigger | Set as Events

Event trigger

	SCADA_TIME_STAMP	ALARM_MESSAGE	EVENT_MESSAGE	STATION	STATION_POINT	TAG
Delete	2012-05-28T10:54:14	ON	CMD142.OMNI.BTCH.ID.SEQ.	DNVR	DNVR,CMD142	CMD142

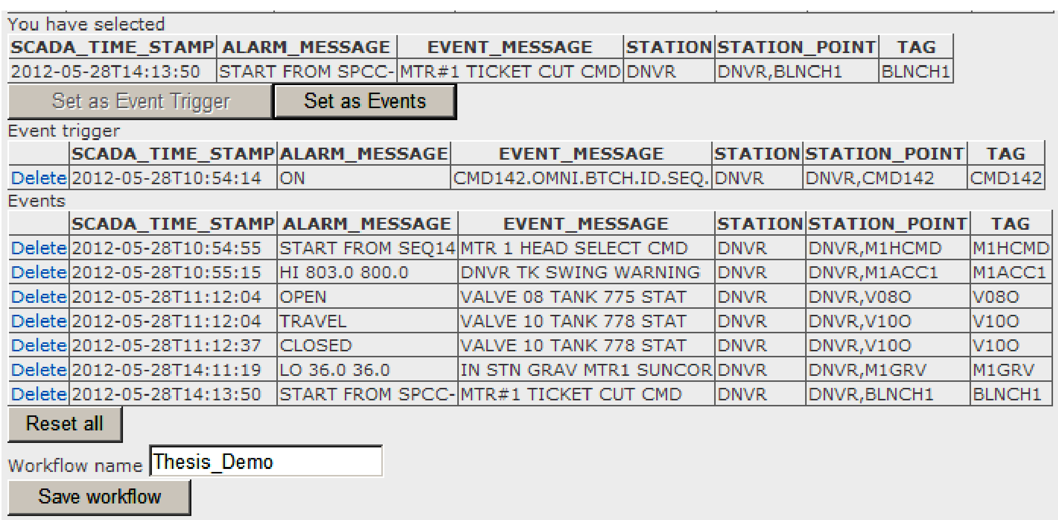
Events

	SCADA_TIME_STAMP	ALARM_MESSAGE	EVENT_MESSAGE	STATION	STATION_POINT	TAG
Delete	2012-05-28T10:54:55	START FROM SEQ14	MTR 1 HEAD SELECT CMD	DNVR	DNVR,M1HCMD	M1HCMD

Reset all

Figure 5.9: Selecting filtered events as a sequence

When an event is identified as a trigger, it will be moved to "Event Trigger" dashboard and the rest of the events are subsequently moved to "Events" dashboard as shown in Figure 5.8 and 5.9. OWBE uses an event trigger to start a workflow.


You have selected

SCADA_TIME_STAMP	ALARM_MESSAGE	EVENT_MESSAGE	STATION	STATION_POINT	TAG
2012-05-28T14:13:50	START FROM SPCC-	MTR#1 TICKET CUT CMD	DNVR	DNVR,BLNCH1	BLNCH1

Set as Event Trigger | Set as Events

Event trigger

	SCADA_TIME_STAMP	ALARM_MESSAGE	EVENT_MESSAGE	STATION	STATION_POINT	TAG
Delete	2012-05-28T10:54:14	ON	CMD142.OMNI.BTCH.ID.SEQ.	DNVR	DNVR,CMD142	CMD142

Events

	SCADA_TIME_STAMP	ALARM_MESSAGE	EVENT_MESSAGE	STATION	STATION_POINT	TAG
Delete	2012-05-28T10:54:55	START FROM SEQ14	MTR 1 HEAD SELECT CMD	DNVR	DNVR,M1HCMD	M1HCMD
Delete	2012-05-28T10:55:15	HI 803.0 800.0	DNVR TK SWING WARNING	DNVR	DNVR,M1ACC1	M1ACC1
Delete	2012-05-28T11:12:04	OPEN	VALVE 08 TANK 775 STAT	DNVR	DNVR,V08O	V08O
Delete	2012-05-28T11:12:04	TRAVEL	VALVE 10 TANK 778 STAT	DNVR	DNVR,V10O	V10O
Delete	2012-05-28T11:12:37	CLOSED	VALVE 10 TANK 778 STAT	DNVR	DNVR,V10O	V10O
Delete	2012-05-28T14:11:19	LO 36.0 36.0	IN STN GRAV MTR1 SUNCOR	DNVR	DNVR,M1GRV	M1GRV
Delete	2012-05-28T14:13:50	START FROM SPCC-	MTR#1 TICKET CUT CMD	DNVR	DNVR,BLNCH1	BLNCH1

Reset all

Workflow name Thesis_Demo

Save workflow

Figure 5.10: Saving selected events as workflows

Users will need to adjust the sequence of events in the workflow list by selecting them in a sequence as shown in Figure 5.10. This application also provides users the ability to add ad-hoc events to the workflow list to build updated workflows catering to changing operating environment for their specific processes. “Save workflow” button in Figure 5.10 enables users to save filtered events as workflows in the document library.

5.5 Saving Workflows in Workflow Document Library (Step – 4)

Saving workflows in the document library correspond to step 4 of OWBE implementation. Filtered workflow activities are stored in a workflow document library within SharePoint as XML documents. All of the saved workflows can be read or edited by clicking on pertinent XML files in the document library. Our running example, “Thesis_Demo.xml” workflow can be viewed in Microsoft’s internet explorer (IE) as show in Figure 5.11.

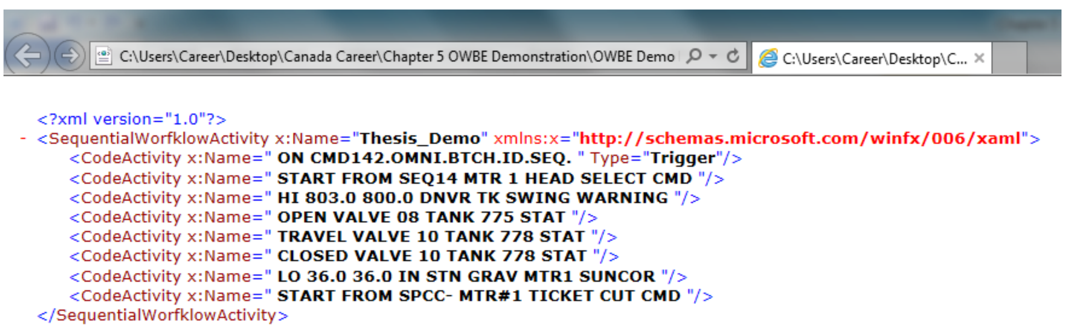


Figure 5.11: “Thesis_Demo.xml” workflow when opened in an IE

5.6 Summary

In this chapter, the architecture diagram of the OWBE platform, developed by the group, is discussed. Various functionalities of the web parts are demonstrated using event data obtained from an industrial case study, namely the tank-swing data provided by Suncor Pipelines. The demonstration aims to provide the users a step-by-step procedure to obtain the set of relevant events for constructing a workflow using the OWBE platform.

Chapter 6

Concluding Remarks and Future Work

6.1 Concluding Remarks

In this thesis, we have established a framework to develop industrial workflows from event logs and paper-based standard operating procedure(s). The significance of capturing operational knowledge from experienced workforce is demonstrated using two industrial case studies. This work explored various challenges involved in data pre-processing of different types of event log databases used in the process industry. We have suggested methods to turn these disparate event log messages into a standardized format acceptable for use with workflow mining tools. From the industrial case studies discussed in this thesis, we can see that a user requires an intrinsic understanding of the plant operations and event log archiving systems to pre-process the data and come up with messages that uniquely convey the process activities. We have also shown that the workflows developed can be used for conformance monitoring, which can help supervisors in monitoring operator actions and identifying deviations in set procedures. We have also demonstrated how workflow conformance can be instrumental in identifying problems associated with instrumentation, processes and human factors, which might affect smooth process operations. The workflow strategies developed in this work enable operators to select, filter and capture their own actions while they perform their work procedures. These captured workflows are retained as permanent corporate assets of expert knowledge which can then be used to train a new generation of operational crew.

Throughout this work, a comprehensive review of operating procedures and a substantial amount of discussions have been carried out with plant operators and supervisors of the two industrial case studies which has provided key design inputs for the development of "Event Loader" and "Event Filter" functions in the OWBE platform.

The outcomes of this work are summarized below:

1. Developed guidelines to analyze event log data and extract industrial workflows that can capture operational knowledge of the processes.
2. Developed strategies that help plant personnel to turn paper-based standard operating procedures into software-based workflows catering to their changing operating environment, which in turn reduces ambiguity in operating practices.
3. Developed workflows for start-up of boiler operations at the Heating Plant in the University of Alberta. Further, workflows are also developed for the sub-sequence of burner rotation by extracting events from customized event logs.
4. The benefits of software-based workflows as a conformance monitoring tool are demonstrated through the burner rotation workflows.
5. Developed workflows for Suncor Energy Pipeline operations (tank-swing and leak detection processes) which can provide higher operational visibility for process improvement.
6. Developed a framework to document work procedures for auditing and regulatory compliance.

6.2 Future Work

Data pre-processing is one of the key challenges that needs to be addressed in the future development of the OWBE platform. Throughout this work, event log data pre-processing is done manually in a Microsoft Excel environment. Data pre-processing is not addressed in the project scope document for the first phase development of OWBE platform. Based on the outcomes of this work, in order to deal with enormous amount of plant operational data, it was recommended to the Sponsor Company that it is essential to accommodate a feature in the OWBE platform to handle a variety of event messages generated from information systems supplied by different vendors.

Based on brainstorming discussions with end users regarding the feasibility and adaptability of using OWBE technology in their day to day operations, industrial workflows are viewed as an effective tool that can guide operators with the best operating practices to safely and efficiently deal with out-of-spec events and abnormal situations. However, an important issue to address in the future development of OWBE platform is how to integrate OWBE software within an operator console. Based on the findings of this work, it is recommended that the Sponsor Company consider Human-Machine-Interface (HMI) design factors while integrating OWBE platform into the live operating environment. Industrial workflows and the OWBE platform will pave the way for seamless flow of process and operational information among the process, people and equipment. The OWBE platform can significantly benefit Canadian process industry by providing tools for improving operator skills and achieve superior operational visibility.

References

[1] D. Walker, P., E. Cammy, N., J. Ellis, B., & D. Seibert, K. (2011). *OPERATIONS SKILLS FOR THE 21ST CENTURY*. Illinois, USA. Retrieved from http://www.uop.com/?document=operations-skills-for-the-21st-century&download=1

[2] Bolick, J. (2008). *Challenges in Process Control* (p. 36). Hauptsitzung. Retrieved from http://www.honeywell.com/sites/servlet/com.merx.npoint.servlets.AttachmentServlet?annid=A141039BC-2342-2FF7-CD4E-FC39E50C8DFA

[3] Taylor, J. (n.d.). Knowledge transfer strategies to beat skills shortages. Retrieved August 14, 2013, from http://www.computerweekly.com/opinion/Knowledge-transfer-strategies-to-beat-skills-shortages

[4] Sampath, R., & Robinson, M. (2005). The Talent Crisis in Upstream Oil & Gas. Retrieved August 13, 2013, from http://www.deloitte.com/assets/Dcom-Canada/Local Assets/Document/CA_ER_TalentCrisisUpstreamOilGas_18_11_2005.pdf

[5] ISA. (2007). ISA Leads Charge to Develop an Automation Engineering Degree Program. Retrieved August 14, 2013, from http://www.isa.org/Template.cfm?Section=Press_Releases5&template=/TaggedPage/DetailDisplay.cfm&ContentID=60162

[6] Spindler, J. (2013). Next-generation workforce: Process automation training. Retrieved August 14, 2013, from http://www.controleng.com/single-article/next-generation-workforce-process-automation-training/ca1be8d8fb976ee7e34ae07ec3daaa53.html

[7] *Abnormal Situation Management - Effective Automation to Improve Operator Performance*. (2007). Retrieved from http://www.asmconsortium.net/Documents/2007 ASM Overview -- MKO Symposium.pdf

[8] Wilkins, M. (2011). Improving operations in continuous processes through procedural automation. Retrieved August 14, 2013, from http://www.isa.org/InTechTemplate.cfm?template=/ContentManagement/ContentDisplay.cfm&ContentID=87848

[9] The Elephant in the Room. (2013). Retrieved August 14, 2013, from http://automation.isa.org/2013/03/the-elephant-in-the-room/

[10] Rozinat, A., & W. Günther, C. (2012). Disco: Process Mining Software. The Netherlands: Fluxicon Process Laboratories. Retrieved from http://fluxicon.com/disco/

[11] Process mining. (n.d.). *Wikipedia*. Retrieved August 17, 2013, from http://en.wikipedia.org/wiki/Process_mining

[12] Tiwari, A., Turner, C. J., & Majeed, B. (2008). A review of business process mining: state-of-the-art and future trends. *Business Process Management Journal*, *14*, 5–22. doi:10.1108/14637150810849373

[13] Van Dongen, B. F., De Medeiros, A. K. A., Verbeek, H. M. W., Weijters, A. J. M. M., & Van Der Aalst, W. M. P. (2005). The ProM framework: A new era in process mining tool support. (G. Ciardo & P. Darondeau, Eds.) *Applications and Theory of Petri Nets 2005*, *3536*(i), 444–454. doi:10.1007/11494744_25

[14] Rozinat, A. (2010). ProM Tips — Which Mining Algorithm Should You Use? Retrieved August 17, 2013, from http://fluxicon.com/blog/2010/10/prom-tips-mining-algorithm/

[15] Günther, C. W., & Van Der Aalst, W. M. P. (2007). Fuzzy Mining – Adaptive Process Simplification Based on Multi-Perspective Metrics. *Business Process Management*, *4714*, 328–343. doi:10.1007/978-3-540-75183-0

[16] Rozinat, A. (2012). *Disco User's Guide* (pp. 3–6). Retrieved from http://fluxicon.com/disco/files/Disco-User-Guide.pdf

[17] The OPC Foundation - The Interoperability Standard for Industrial Automation & Other Related Domains. (2011). Retrieved August 17, 2013, from

http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC

[18] Tokuda, S., M. Rantanen, E., & C.M. Gonzalez de Sather, J. (2004). AN ELECTRONIC INSTRUCTION MANUAL AND CHECKLIST FOR STEAM BOILER START-UP. In *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting*. Santa Minica.

[19] Microsoft Visio Premium 2010. (2010). Microsoft Corporation.

[20] Create, import, and export SharePoint workflows in Visio. (n.d.). Retrieved August 17, 2013, from http://office.microsoft.com/en-ca/visio-help/create-import-and-export-sharepoint-workflows-in-visio-HA101888007.aspx?CTT=5&origin=HA101893811

[21] Gruhn, P., & Cheddie, H. (2006). *Safety Instrumented Systems: Design, Analysis, and Justification* (Second., p. 54). NC, USA: International Society of Automation.

[22] Why Pipelines are Needed. (n.d.). *Canadian Energy Pipeline Association*. Retrieved August 04, 2013, from http://www.cepa.com/about-pipelines/why-pipelines

[23] Types of Pipelines. (n.d.). *Canadian Energy Pipeline Association*. Retrieved July 19, 2013, from http://www.cepa.com/about-pipelines/types-of-pipelines

[24] Shell Pipeline Control Center Houston, Texas. (n.d.). *Shell Pipeline Company LP*. Retrieved July 19, 2013, from http://splc-r.com/yahoo_site_admin/assets/docs/Control_Center_Overview1.28200240.pdf

[25] Mohitpur, M., Szabo, J., & Van Hardeveld, T. (2005). *Pipeline Operation & Maintenance: A Practical Approach* (pp. 85–90). New York: ASME Press.

[26] *Appendix C. Pipeline Regions and Operations: The Transition to Ultra-Low-Sulfur Diesel Fuel: Effects on Prices and Supply (Report #: SR-OIAF/2001-01)*. (n.d.). Washington, DC 20585. Retrieved from http://www.eia.gov/oiaf/servicerpt/ulsd/index.html

[27] Pipeline Terminology. (n.d.). *Shell Pipeline Company LP*. Retrieved July 19, 2013, from http://www.shell.us/products-services/solutions-for-businesses/pipeline/pipeline-america/terminology.html

[28] Kendell, J. (Energy I. A. (2001). *The Transition to Ultra-Low-Sulfur Diesel Fuel: Effects on Prices and Supply* (pp. 89–93). Washington, DC 20585. Retrieved from www.eia.doe.gov/oiaf/servicerpt/ulsd/chapter3.html

[29] Spitzke, D. (Suncor). (2009). *SPCC Task Procedure: Denver Batch Cut Protocol - Procedure 102*. Edmonton.

[30] S. Yoon, M., Bruce Warren, C., & Adam, S. (n.d.). *Pipeline System Automation and Control : Pipeline Engineering Monograph Series* (p. 257). New York: ASME Press.

[31] *OMNI Flow Computers: Basic Operation (Volume 2)*. (n.d.). Retrieved from http://www.omniflow.com/documents/manuals/V2_Rev_A.pdf

[32] *Crude Oil Price Differentials and Differences in Oil Qualities: A Statistical Analysis*. (2005). Washington, D.C. Retrieved from http://www.esmap.org/sites/esmap.org/files/08105.Technical Paper_Crude Oil Price Differentials and Differences in Oil Qualities A Statistical Analysis.pdf

[33] S. Yoon, M., Bruce Warren, C., & Adam, S. (n.d.). *Pipeline System Automation and Control: Pipleline Engineering Monograph Series* (pp. 21–22). New York: ASME Press.

[34] Permanente, K., V. Biron, P., & Malhotra, A. (2004). XML Schema Part 2: Datatypes Second Edition. *W3C Recommendation 28 October 2004*. Retrieved July 19, 2013, from http://www.w3.org/TR/xmlschema-2/#dateTime

[35] Trench, C. (2007). Ultra Low Sulfur Diesel at One Year. *In the Pipe: Association of Oil Pipelines American Petroleum Institute*, *1*(3). Retrieved from http://www.enewsbuilder.net/inthepipe/e_article000979449.cfm

[36] S. Yoon, M., Bruce Warren, C., & Adam, S. (n.d.). *Pipeline System Automation and Control : Pipeline Engineering Monograph Series* (p. 257). New York: ASME Press.

[37] Frequently Asked Questions (FAQs) Concerning Federally-Regulated Petroleum Pipelines in Canada. (n.d.). Retrieved July 18, 2013, from http://www.nrcan.gc.ca/energy/sources/natural-gas/pipeline-faq/2248

[38] Protecting pipeline integrity – it's a full-time job! (n.d.). *Canadian Energy Pipeline Association*. Retrieved July 18, 2013, from http://www.cepa.com/protecting-pipeline-integrity-its-a-full-time-job

[39] Danilak, C. (Suncor), & Kireger, V. (Suncor). (2012). *SPCC Task Procedure: Response to Leak Detection Alarms - Procedure No: 111*. Edmonton.

[40] S. Yoon, M., Bruce Warren, C., & Adam, S. (n.d.). *Pipeline System Automation and Control : Pipeline Engineering Monograph Series* (p. 304). New York: ASME Press.

[41] Workflow example: Send a notification message. (n.d.). *Microsoft Workflow Foundation*. Retrieved July 19, 2013, from http://office.microsoft.com/en-ca/sharepoint-designer-help/workflow-example-send-a-notification-message-HA010182908.aspx

[42] Liquids Pipelines. (n.d.). *Canadian Energy Pipeline Association*. Retrieved July 19, 2013, from http://www.cepa.com/about-pipelines/types-of-pipelines/liquids-pipelines

[43] SCADA Willowglen DACP OPC Server. (n.d.). Retrieved July 18, 2013, from http://www.matrikonopc.com/opc-drivers/255/base-driver-details.aspx

[44] Heating Plant. (2013). *Department of Facilities and Operations, University of Alberta*. Retrieved August 02, 2013, from http://www.facilities.ualberta.ca/FO_Utilities/HeatingPlant.aspx

[45] Heselton, K. (2005). *Boiler Operator's Handbook*. Lilburn, Georgia: THE FAIRMONT PRESS, INC.

[46] Utilities Department, U. of A. (2010). *#4 Boiler Start-up Procedure*. Edmonton.