# A METHOD FOR FORMAL ANALYSIS AND SIMULATION OF STANDARD OPERATING PROCEDURES (SOPS) TO MEET SAFETY STANDARDS

by

Jomana Bashatah

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Systems Engineering and Operations Research

Committee:

_____ Dr. Lance Sherry, Dissertation Director

_____ Dr. George Donohue, Committee Member

_____ Dr. John Shortle, Committee Member

_____ Dr. Vivian Motti, Committee Member

_____ Dr. John Shortle, Department Chair

Date: _____ Spring Semester 2024
George Mason University
Fairfax, VA

# A Method for Formal Analysis and Simulation of Standard Operating Procedures (SOPs) to Meet Safety Standards

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Jomana Bashatah
Master of Science
George Mason University, 2019
Bachelor of Science
George Mason University, 2018

Director: Lance Sherry, Professor
Department of Systems Engineering and Operations Research

Spring Semester 2024
George Mason University
Fairfax, VA

Copyright 2024 Jomana Bashatah
All Rights Reserved

**DEDICATION**

## ACKNOWLEDGMENTS

## TABLE OF CONTENTS

Page
List of Tables ..... ix
List of Figures ..... x
List of Equations ..... xii
List of Abbreviations ..... xiii
Abstract ..... xv
Chapter One: Introduction ..... 1
1.1 Introduction to Standard Operating Procedures ..... 2
1.2 Scope ..... 3
1.3 Importance of Standard Operating Procedures (SOPs) ..... 4
1.3.1 Importance of SOP Design in Completing the SOP Step Before the Onset of a Hazardous Condition ..... 5
1.4 Regulatory Standards and Guidelines for SOP Development ..... 6
1.5 Industry Practices and Processes for SOP Development ..... 7
1.5.1 SOP Creation Process ..... 9
1.5.2 SOP Approval Process ..... 9
1.5.3 SOP Publishing Process ..... 10
1.5.4 SOP Training & Monitoring in Operations ..... 11
1.5.5 Bottlenecks with the Current Process ..... 11
1.6 Tools for SOP Development and Management ..... 11
1.6.1 SOP Creation Tools ..... 12
1.6.2 SOP Approval Tools ..... 12
1.6.3 SOP Publishing Tools ..... 13
1.6.4 SOP Training Tools ..... 13
1.6.5 SOP Monitoring Field Operation Tools ..... 13
1.6.6 Bottlenecks of Current Tools ..... 14
1.7 Problem Statements ..... 14
1.8 Unique Contributions ..... 15
1.8.1 List of Published Papers ..... 16
Chapter Two: Literature Survey of Models of HMI Interaction and SOP Languages ..... 19

2.1 Models of Human-Machine Interaction ..... 19
2.1.1 Goals, Operators, Methods, and Selection Rules (GOMS) ..... 20
2.1.2 Key Level Model ..... 21
2.1.3 Cognitive Perceptual Motor – GOMS (CPM-GOMS) ..... 22
2.1.4 Adaptive Control of Thought – Rational (ACT-R) ..... 23
2.1.5 SOAR ..... 25
2.1.6 Time-Based Models of Standard Operating Procedures ..... 26
2.1.6.1 Pilot-Autonomy Workload Simulation (PAWS) ..... 27
2.1.6.2 Operational Event Sequence Diagrams (OESDs) ..... 27
2.1.6.3 Dynamic Value Action Stream Diagrams ..... 28
2.1.6.3.1 Time Data ..... 29
2.1.6.4 HCIPA ..... 30
2.1.6.5 ToP & AOTW ..... 31
2.1.6.5.1 Time Data ..... 32
2.1.7 Summary of Models Related to Standard Operating Procedures (SOPs) ..... 32
2.2 Modelling Languages for Standard Operating Procedures (SOPs) ..... 33
2.2.1 Procedure Representation Language (PRL) ..... 33
2.2.2 PLEXIL ..... 35
2.2.3 BPMN ..... 36
2.2.4 Summary of Languages Used for Standard Operating Procedures ..... 36
2.3 Summary of Literature Survey ..... 37
Chapter Three: Methodology for Analysis and Simulation of SOP Steps ..... 39
3.1 Design of A SOP Analysis and Simulation System ..... 40
3.1.1 Generate SOP Steps ..... 43
3.1.2 Classify SOP Steps into SOP Step Components ..... 46
3.1.2.1 Canonical Structure of SOPs (e-PRL) ..... 47
3.1.2.2 Classify SOP Steps into e-PRL – Manual Method ..... 50
3.1.2.3 Classify SOP Steps into e-PRL - LLM-Based Chatbot Method ..... 53
3.1.3 Analyze SOP Content ..... 54
3.1.3.1 SOP Step Type ..... 55
3.1.3.2 SOP Step Recall Score ..... 57
3.1.3.3 Estimating Training Requirements ..... 60

3.1.4 Simulate SOP ..... 63
3.1.4.1 Human Performance Time Distribution Database ..... 73
3.1.5 Generate MBSE Action Diagram ..... 78
3.1.5.1 Create MBSE Action Diagram ..... 78
3.1.5.2 Convert Existing e-PRL SOP into MBSE Action Diagrams ..... 82
3.1.6 Summary of Methodology for Analysis and Simulation of SOP Steps ..... 83
Chapter Four: Case Studies ..... 85
4.1 Case Study One: Airline Operations SOPs ..... 86
4.1.1 Objective ..... 86
4.1.2 Data and Preprocessing ..... 86
4.1.3 Results ..... 87
4.1.3.1 SOP Step Type ..... 87
4.1.3.2 SOP Step Recall Score ..... 91
4.1.3.3 Estimating Training Requirements ..... 92
4.1.3.4 Simulations of SOPs ..... 93
4.1.4 Discussion ..... 95
4.2 Case Study Two: International Space Station (ISS) Habitable Airlock (HAL) SOPs ..... 96
4.2.1 Objective ..... 97
4.2.2 Data and Preprocessing ..... 97
4.2.3 Results ..... 98
4.2.3.1 SOP Step Type ..... 98
4.2.3.2 SOP Step Recall Score by SOP Step Type ..... 99
4.2.3.2.1 Action-Only Steps ..... 102
4.2.3.2.2 Action with Waiting & Verification Steps ..... 102
4.2.3.3 SOP Step Recall Score by SOP ..... 102
4.2.3.4 Estimating Training Requirements ..... 103
4.2.3.5 Simulation of SOPs ..... 104
4.2.3.6 MBSE Diagrams ..... 107
4.2.4 Discussion ..... 110
4.2.5 Summary ..... 111
4.2.5.1 Findings ..... 112
4.3 Case Study Three: Autonomous Vehicle SOPs ..... 112

4.3.1 Objective of the Case Study ..... 113
4.3.2 Data and Preprocessing ..... 113
4.1.2.1 Generate SOPs Steps from Owner's Manual ..... 115
4.1.2.2 Classify SOP Steps to Components ..... 121
4.3.3 Results ..... 129
4.3.3.1 SOP Step Type ..... 129
4.3.3.2 SOP Step Recall Score by SOP Step Type ..... 129
4.3.3.3 SOP Step Recall Score by SOP ..... 131
4.3.3.4 Estimating Training Requirements ..... 132
4.3.3.5 Simulations of SOPs ..... 133
4.3.3.6 MBSE Diagrams ..... 141
4.3.4 Time Analysis for the Process of Classification of SOP Step into Components: Manual Vs. Chatbot ..... 142
4.3.5 Discussion ..... 144
4.3.6 Summary ..... 146
4.3.6.1 Findings ..... 147
Chapter Five: Conclusions ..... 149
5.1 SOP Analysis for Operational Efficiency ..... 155
5.2 SOP Analysis for Training Time Predictions ..... 156
5.3 SOP Analysis for Process Improvement ..... 156
5.4 SOP Analysis for Regulatory Approval ..... 157
Appendix A ..... 159
Analysis of Aviation Accidents ..... 159
Appendix B ..... 162
List of Tasks for Human Performance Data ..... 162
Appendix C ..... 164
SOPs and SOP Steps Generated for Tesla Model 3 ..... 164
Appendix D ..... 167
Human Performance Database ..... 167
References ..... 175

## LIST OF TABLES

Table Page
Table 1 Summary of Problems and Contributions of this Dissertation ..... 16
Table 2 Questions for Decision Nodes ..... 51
Table 3 Metrics for SOP Content Analysis..... 56
Table 4 Implicit Elements Definitions and Examples ..... 59
Table 5 Number of e-PRL Elements Required by SOP Step Type ..... 60
Table 6 Average Number of Repetitions of a List Required to Reach Proficiency..... 61
Table 7 Number of Repetitions and Days to Proficiency ..... 63
Table 8 Characteristics of Data Obtained (in seconds)..... 76
Table 9 Parameters for Triangular Distributions and Goodness of Fit..... 77
Table 10 Mapping Between e-PRL and SOP Action Diagram..... 83
Table 11 Training, Certification, and Selectiveness of Three Case Studies..... 85
Table 12 SOP Step Types – Case 1 ..... 87
Table 13 Content Explicitly Present in the SOP Steps – Case 1 ..... 89
Table 14 Average SOP Step Recall Score by SOP - Case 1..... 91
Table 15 Training Estimates – Case 1 ..... 93
Table 16 SOP Step Types – Case Study 2 ..... 98
Table 17 Content Explicitly Present in the SOP Steps ..... 100
Table 18 SOP Step Recall Score By SOP..... 103
Table 19 Training Estimates ..... 103
Table 20 Calculated AOTW and Simulated PFtC - Case Study 2..... 105
Table 21 Procedures, Initiating Conditions, Alerts, and Hazardous Events that May Occur ..... 118
Table 22 Claude Classification Results of the Forward Collision Warning SOP..... 125
Table 23 Accuracy of Chatbot After Each Iteration ..... 128
Table 24 SOP Step Types – Case Study 2 ..... 129
Table 25 SOP Recall Score by SOP Step Type – Case 3 ..... 130
Table 26 SOP Step Recall Score by SOP – Case 3..... 131
Table 27 Training Estimates - Case 3 ..... 133
Table 28 Parameters and Values for Calculating AOTW..... 134
Table 29 PFtC for Simulated SOPs - Case 3 ..... 136
Table 30 Problematic Steps - Case 3 ..... 139
Table 31 Time Reduction by SOP for Each Participant ..... 144

## LIST OF FIGURES

Figure Page
Figure 1 Example of SOP (Swissair Air Conditioning Smoke) [2, App. B] ..... 3
Figure 2 Example SOP Step ..... 4
Figure 3 Process for Creating & Maintaining SOPs ..... 9
Figure 4 OESD Symbols [50, Tbl. 1] ..... 28
Figure 5 MODAPTS Codes ..... 30
Figure 6 Hierarchical Structure of PRL ..... 34
Figure 7 PRL and PLEXIL Structures [57, p. 5] ..... 36
Figure 8 Framework for SOP Analysis and Simulation System ..... 42
Figure 9 Prompt to Generate SOP ..... 44
Figure 10 Modified Version of Chatbot Generated SOP (Modifications Highlighted in Red) ..... 45
Figure 11 Final SOP After Manual Modifications ..... 46
Figure 12 Basic Structure of SOP Steps ..... 47
Figure 13 Canonical Structure of SOP Steps ..... 48
Figure 14 Process of Annotating SOPs ..... 51
Figure 15 Process for Training a Chatbot to Classify SOP Step Components ..... 54
Figure 16 Flowchart to Categorize SOP Steps to Types ..... 57
Figure 17 Example of SOP Step ..... 58
Figure 18 Excel-Based Program (User Entry) ..... 64
Figure 19 CADSOPs Interface ..... 65
Figure 20 Importing Existing Steps into Interface ..... 66
Figure 21 Assess Cue Type ..... 66
Figure 22 Assess Frequency ..... 66
Figure 23 Cell to Import Time Distributions ..... 67
Figure 24 Verify Imported Time Distributions ..... 67
Figure 25 Flowchart for SOP Simulation ..... 71
Figure 26 PBT, ToP, AOTW, and PFtC ..... 73
Figure 27 SOP Step Recall Score by SOP - Case 1 ..... 92
Figure 28 MBSE Visualization - Case 1 ..... 94
Figure 29 ToP & AOTW Scatter Plot ..... 95
Figure 30 ToP & AOTW Distributions ..... 95
Figure 31 Habitable Airlock (HAL) ..... 97
Figure 32 ToP VS AOTW Scatter Plot - Case Study 2 ..... 106
Figure 33 ToP and AOTW Distributions - Case Study 2 ..... 106
Figure 34 SOP Step in MBSE Action Diagram Form ..... 108
Figure 35 e-PRL Representation of SOP Step in Figure 34 ..... 109
Figure 36 Complex SOP Step with Waiting and Verification ..... 109
Figure 37 e-PRL Representation of Figure 36 ..... 110
Figure 38 Tesla Model 3 Owner's Manual ..... 114

Figure 39 Import Button in Claude ..... 115
Figure 40 Scenarios where Human Takeover is Required ..... 116
Figure 41 Collision Avoidance Features Where Human Intervention may be Required. ..... 117
Figure 42 Claude Prompt to Generate Procedure Steps for "Forward Collision Warning" SOP ..... 119
Figure 43 Claude Prompt and Response to Re-Write SOP steps in Trigger-Action Format. ..... 120
Figure 44 "Forward Collision Warning" Procedure after Modification ..... 121
Figure 45 Rules for Labeling SOP Text ..... 122
Figure 46 Teaching Claude the Rules in the Database ..... 123
Figure 47 Prompt to Classify SOP into e-PRL Components ..... 125
Figure 48 Prompt to Generate New Rules ..... 126
Figure 49 Prompt to Add Generated Rules to Database ..... 127
Figure 50 SOP Step Recall Score - Case 3 ..... 130
Figure 51 SOP Step Recall by SOP ..... 132
Figure 52 Cue Evaluation by SOP – Case 3 ..... 136
Figure 53 ToP & AOTW Distributions – Case 3 ..... 137
Figure 54 ToP vs. AOTW - Case 3 ..... 137
Figure 55 Cue Evaluations by SOP ..... 140
Figure 56 Action Type by SOP ..... 141
Figure 57 Forward Collision SOP ..... 142
Figure 58 Taxonomy of SOP Design Issues as Identified from Human Operator Intervention Tasks in Accident ..... 161

## LIST OF EQUATIONS


Equation Page
Equation 1 Standard Time Completion Ratio ..... 29
Equation 2 SOP Step Recall Score ..... 60
Equation 3 Total Number of Repetitions for an SOP as a Function of Number of Steps 62
Equation 4 Number of Steps in SOP with Recall Items ..... 62
Equation 5 Total Number of Repetitions for a Perfect Recall ..... 62
Equation 6 Number of Days Required to Reach Proficiency ..... 63
Equation 7 Triangular CDF ..... 68
Equation 8 Combined Area under a Triangular Distribution ..... 68
Equation 9 Closed Form Solution of Common Side H ..... 68
Equation 10 Ratio to Determine Random Variable Resides in Which Triangle ..... 69
Equation 11 Area of Triangle bound by x ..... 69
Equation 12 Closed Form Solution of h ..... 69
Equation 13 Proportional Property of Triangles ..... 69
Equation 14 Closed Form Solution of x ..... 69
Equation 15 Closed Form Solution of x When x Lies in Left Triangle ..... 69
Equation 16 Inverse Cumulative Distribution Function (CDF) ..... 70
Equation 17 Maximum of AOTW ..... 104
Equation 18 Mode of AOTW ..... 104
Equation 19 Minimum of AOTW ..... 104
Equation 20 Lead Vehicle Velocity in Ft/Sec ..... 134
Equation 21 Lead Vehicle Deceleration in Ft/Sec^2 ..... 134
Equation 22 Lead Vehicle Stopping Time in Seconds ..... 134
Equation 23 Following Car Velocity in Ft/Sec ..... 134
Equation 24 Following Car Deceleration in Ft/Sec^2 ..... 134
Equation 25 Following Car Stopping Time in Seconds ..... 134
Equation 26 Following Distance between Head of Follow Car and Tail of Lead Car ... 135
Equation 27 Following Distance in Seconds ..... 135
Equation 28 Following Vehicle Stopping Time ..... 135
Equation 29 Calculated AOTW - Case 3 ..... 135

## LIST OF ABBREVIATIONS

Adaptive Control of Thought - Rational ..... ACT-R
Advisory Circular ..... AC
Allowable Operational Time Window ..... AOTW
Autosteer Disable ..... AD
Blind Spot Collision ..... BSC
Business Process Execution Language ..... BPEL
Business Process Model and Notation ..... BPMN
Cognitive Perceptual Motor GOMS ..... CPM-GOMS
Extended Procedure Representation Language ..... e-PRL
Extensible Markup Language ..... XML
Extravehicular Activities ..... EVA
Federal Aviation Administration ..... FAA
Field of View ..... FoV
Flight Data Monitoring ..... FDM
Flight Operations Quality Assurance ..... FOQA
Forward Collision Warning ..... FCW
Goals, Operators, Methods, and Selection Rules ..... GOMS
Habitable Airlock ..... HAL
Human-Information Processing ..... HIP
Human Human-machine interaction ..... HMI
International Space Station ..... ISS
Intersection Turn Takeover ..... ITT
Key Level Model ..... KLM
Lane Departure Avoidance ..... LDA
Large Language Model ..... LLM
Long-term Term Memory ..... LTM
Model-Based Systems Engineering ..... MBSE
Model Human Processor ..... MHP
Modular Arrangement of Predetermined Time Standards ..... MODAPTS
Operational Event Sequence Diagram ..... OESD
Original Equipment Manufacturer ..... OEM
Observe, Orient, Decide, and Act ..... OODA
Pilot-Autonomy Workload Simulation ..... PAWS
Plan Execution Interchange Language ..... PLEXIL
Probability of Failure to Complete ..... PFtC
Procedure Buffer Time ..... PBT
Procedure Integrated Development Environment ..... PRIDE
Procedure Representation Language ..... PRL
Problem Space Computational Model ..... PSCM
Standard Operating Procedure ..... SOP

Safety Management System.....SMS
Time on Procedure.....ToP
Unexpected Autopilot Behavior Intervention.....UABI

## ABSTRACT

A METHOD FOR FORMAL ANALYSIS AND SIMULATION OF STANDARD OPERATING PROCEDURES (SOPS) TO MEET SAFETY STANDARDS

Jomana Bashatah, PhD

George Mason University, 2024

Dissertation Director: Dr. Lance Sherry

Standard Operating Procedures (SOPs) are the "glue" that holds the command-and-control center together. They are step-by-step instructions to guide the operators on how to control complex human-machine systems. While the machine is certified and the operators are licensed, SOPs are loosely regulated. Time and cost constraints limit the testing of SOPs to account for the variability noted in human performance, i.e., SOP execution time and the variability in the operational environment. Additionally, SOPs mainly exist as static text documents, i.e., Word documents, hindering the ability to revise SOPs and maintain configuration integrity consistently.

To address these limitations, this dissertation developed a framework for a digital SOP representation, metrics, and a simulation model to aid in creating, revising, and evaluating SOPs. A canonical structure, the extended Procedure Representation Language (e-PRL), was developed to decompose SOP steps into perceptual, cognitive, and motor elements. A method for using Large Language Models (LLMs) to generate SOP Steps from Owners Manuals, and a method to classify the text in the SOP steps into

e-PRL components was developed. Techniques, including Monte-Carlo simulations to assess human performance and quantitative metrics that evaluate SOP content and training requirements, were developed for the e-PRL representation.

Three case studies demonstrating the applicability of the methods are presented from the following domains: (1) aviation operational SOPs, (2) International Space Station (ISS) Habitable Airlock (HAL) SOPs, and (3) semi-autonomous vehicle SOPs. The implications of the results for each case study and the limitations and future work for the methods are discussed.

# CHAPTER ONE: INTRODUCTION

Standard Operating Procedures (SOPs) are the "glue" that holds the command-and-control center (e.g., airliner cockpit) together. The SOPs provide instructions for the operators on how to command the automation to control the machine, especially for abnormal and emergency scenarios. Despite the integral role of SOPs, they are lightly regulated, verified, and validated. Potential rationales for such limited oversight include (1) the ability of human operators to adapt and compensate for deficiencies in SOPs, (2) new SOPs have to be trained, and this provides a de facto (although informal) testing process, and (3) the substantial expenses required for human-in-the-loop testing of off-nominal scenarios.

Analyses of airline accidents, described in Appendix A , highlight the necessity of evaluating SOPs under variable conditions of the Allowable Operational Time Window (AOTW) and the Time on Procedure (ToP). Most accidents occurred when the AOTW was short, *and* the ToP took too long. The ability to evaluate the SOPs while accounting for the variability through analysis and simulation could contribute to increased safety margins.

With the advent of autonomous systems and the transition of human operators towards supervisory roles, the importance of well-designed SOPs is elevated. Additionally, digitization of SOPs may present opportunities for reducing the costs of creating, approving, training, and monitoring SOPs. The digital representation of the SOPs also provides the means to execute simulations of the SOPs.

Section 1 introduces Standard Operating Procedures (SOPs), the regulatory standards and guidelines for SOP development, the current process of SOP development and management, and the methods used for safety analysis of SOPs. This section identifies the issues with the existing SOP analysis and simulation process.

## 1.1 Introduction to Standard Operating Procedures

Standard Operating Procedures (SOPs) are formally recognized guidelines defining the execution of routine or emergency activities essential for safe and efficient operations [1]. SOPs are heavily used in safety-critical domains, such as aviation and Nuclear Power Plants (NPPs), where operational anomalies lead to catastrophic outcomes. An example of the Smoke in the Cockpit SOP, which aims to isolate the source of smoke on board an MD-11 in an emergency, is displayed in Figure 1.

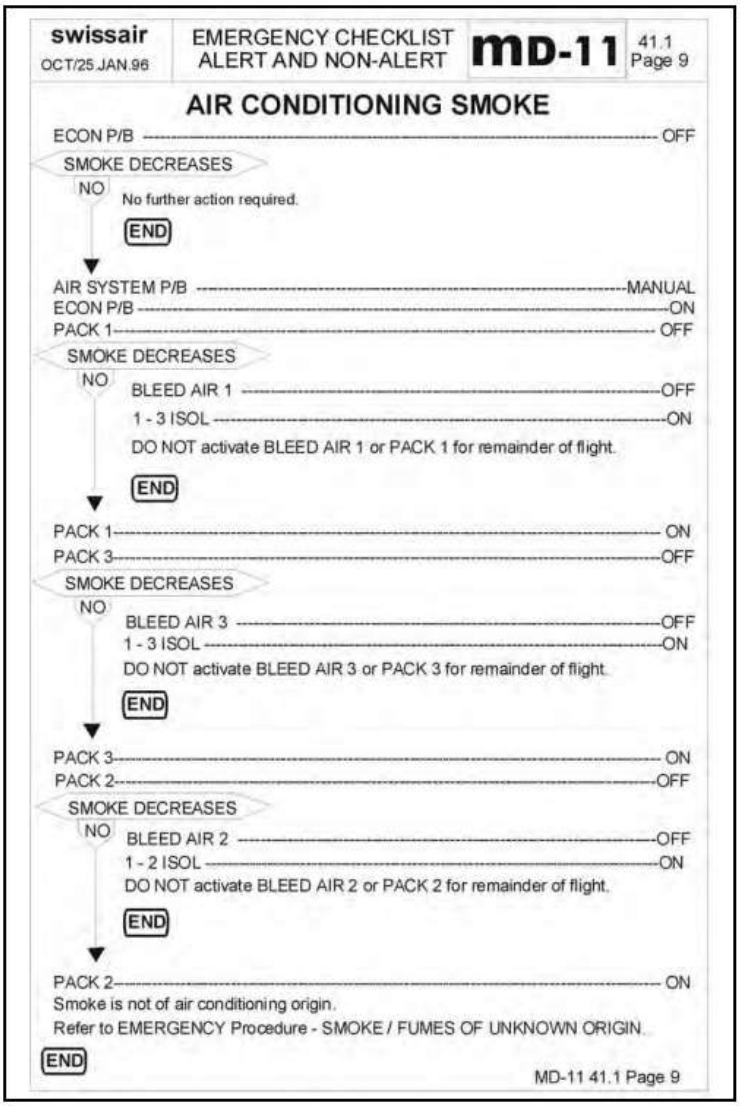



**Figure 1 Example of SOP (Swissair Air Conditioning Smoke) [2, App. B]**

## 1.2 SCOPE

SOPs are composed of two or more SOP steps. The SOP steps define the cognitive aspect (i.e., decision-making) and the physical motor aspects (e.g., pull lever). SOP steps can be represented in a canonical form of <Cognitive (trigger/decide)> <Motor (action)>. The cognitive part of the step captures the perception and decision-making,

while the motor part captures the physical motion. This dissertation is limited to the study of SOP steps similar to the step in Figure 2. This dissertation does not consider steps that provide abstract and high-level instructions without cognitive and motor events.

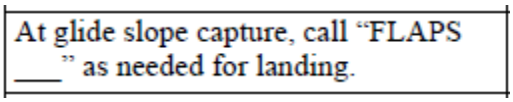



**Figure 2 Example SOP Step**

### 1.3 Importance of Standard Operating Procedures (SOPs)

In complex systems that involve humans collaborating with automation, such as autonomous vehicles, airliners, International Space Station (ISS) systems, and submarines, SOPs are crucial for safely and efficiently operating the system. Prior research has demonstrated that SOPs improve coordination, compliance, situation awareness, and decision-making under normal and abnormal conditions [1]. In emerging systems like autonomous vehicles, SOPs can guide operator monitoring and manual takeover transitions.

SOPs also form a core component of training programs across safety-critical domains. Training and evaluations against SOPs provide insight into trainees' grasp of critical knowledge and strategic decision-making [3]. For example, flight training simulators offer pilot trainees repetitive exposure to SOPs for standard and abnormal scenarios and enable objective skill measurement [4], [5].

In domains with a large pool of operators (e.g., airlines, command centers, etc.), SOPs enable a standardized response across different operators. This allows for mixed crew staffing and scheduling and ensures critical information is shared when required [6]. They also enable an organization to achieve uniformity in executing a specific function [7].

### 1.3.1 Importance of SOP Design in Completing the SOP Step Before the Onset of a Hazardous Condition

SOP design is vital to safe operations, especially during abnormal and emergencies. Poorly designed procedures have been cited in aviation accidents as a contributing factor when the variability of the Available Operational Time Window (AOTW) and the Time on Procedure (ToP) resulted in the SOP step not being completed on time before the vehicle reached a hazardous state.

Analysis of aviation accident reports shows that SOP Step design is a causal factor in completing the required task in the available time [8], [9]. Analysis of twelve Part 121 modern aircraft accidents highlighted three main categories of accident scenarios: (1) excessive procedure execution time (ToP), (2) insufficient allowable time (AOTW) to complete procedures, and (3) a combination of excessive procedure execution time (ToP) and insufficient allowable time (AOTW) (Appendix A).

Nine accidents fit the category of "excessive time to complete procedure." The analysis revealed that accidents resulted from delays in initiating the SOP due to ambiguous triggering information from the external environment (ABX 827 [10]), missing associations between cockpit alerts and creating the required SOP (AF 447 [11],

ABX 827 [10], NW 255 [12], LNI 610 [13]), overreliance on memory (AAL 965 [14], AAL 1420 [15]), or visual cues that are outside the operator's Field of View (FoV) (XL Germany 888 [16], SB 815).

Accidents where there is an insufficient AOTW (SR 111 [2], OZ 214 [17], TK 1951 [18], SQ 237 [19]) were accidents where a human operator couldn't complete any of the required emergency SOPs as a result of environmental phenomena. Accidents fitting into category 3 had excessive ToP and insufficient AOTW (TK 1951, OZ 214).

### 1.4 Regulatory Standards and Guidelines for SOP Development

The possibility of problems with SOP design, where there might be a high percentage of operators who cannot complete the SOP in the required time, made for regulations and policies to arise regarding developing SOPs. The Federal Aviation Administration (FAA) manages the quality of SOPs through two Advisory Circulars (ACs).

The FAA's AC 120-71B [17] guided SOP design, development, and evaluation for Title 14 Code of Federal Regulation (14 CFR) part 121 and part 135 air carriers. It described the general guidelines for writing an exemplary procedure (e.g., use short sentences, use active verbs, write steps as imperative), how the procedure should be organized, how vocabulary and the usage of abbreviations should be consistent, and the general format of procedures (including font, case, justification, etc.,).

The FAA's AC 120-92B [18] guided the development and implementation of Safety Management Systems (SMSs) to manage safety and risk within the operations of an airline. An SMS is a formal top-down approach to managing safety risks and ensuring

the effectiveness of risk mitigation strategies [19]. All hazards are identified when procedures are modified, and risks are mitigated through risk assessment and control techniques. According to this AC, designs and modifications must go through the SMS to identify all risks associated with the SOP.

While AC 120-71B provides high-level guidance on how an SOP should be written, and AC 120-92B provides high-level guidance on how to use an SMS to "approve" a modification to a procedure, both do not guide how to test or evaluate SOPs before approval. They also do not mention the role of time with regard to how procedures are considered.

Furthermore, Barshi et al. [6] developed a seven-step process for developing effective procedures based on a multi-year research effort examining the existing literature, safety reports, accident reports, and inputs from key stakeholders such as pilots, Original Equipment Manufacturers (OEMs), and airline managers. The report outlines human-centered best practices for developing reliable and adaptable procedures across technology and personnel changes. The proposed framework encompasses task analysis, leveraging operational data, and continuous post-deployment monitoring. Such techniques enable proactive risk identifications not fully captured in the current FAA guidance. That relies generally on Subject Matter Expert (SME) reviews.

### 1.5 Industry Practices and Processes for SOP Development

Interviews with eight stakeholders responsible for SOPs, including commercial airliners, military aircraft operators, International Space Station (ISS), and submarines,

identified the current process for creating and maintaining SOPs (Figure 3). The process is based on five main functions or phases:

1. SOP creation
2. SOP approval
3. SOP publishing
4. SOP training
5. SOP monitoring in operations

Each of the functions is described in the subsequent sections. This process aims to update procedures based on emerging issues continually. However, as discussed in later sections, over-reliance on qualitative SME assessments and small-scale evaluations leaves latent issues unaddressed that quantitative approaches could reveal proactively.

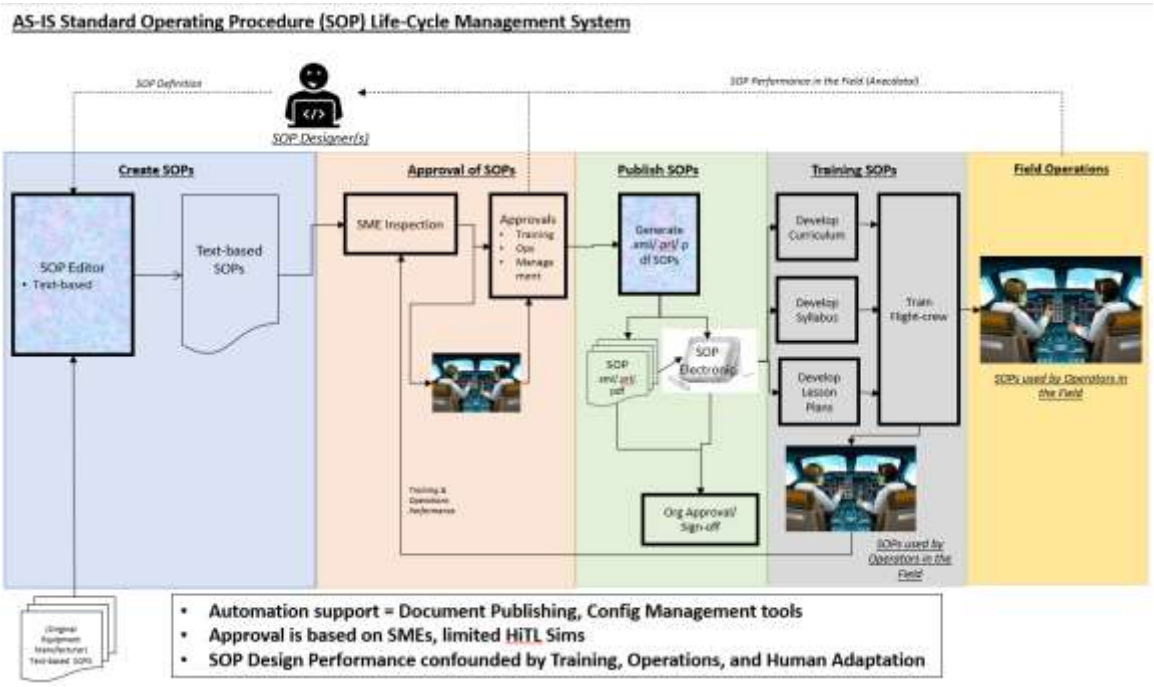



**Figure 3 Process for Creating & Maintaining SOPs**

### 1.5.1 SOP Creation Process

Typically, organizations receive SOPs from Original Equipment Manufacturers (OEMs) to meet the technology's engineering limits [6]. However, OEMs cannot produce customer-specific SOPs that meet the airlines' or operators' policies. Therefore, agencies must modify the SOPs to meet their company's policies and philosophies [1], [6]. SOP creation involves developing or revising text documents using editing tools such as Word or PDF.

### 1.5.2 SOP Approval Process

As described in interviews with industry experts (J. Abrigo, Air Combat Command, Training Support Squadron (ACC/TRSS) Det 7 Deputy Commander, personal

communication, May 5, 2022), (V. DaSilva, Sr. Principal Flight Simulator Trainer (Flight Engineer), Northrop Grumman, personal communication, May 19, 2022), and (T. Schwabe, First Officer, Swiss Air, personal communications, March 24, 2022), the revised SOPs are then run by Subject Matter Experts (SMEs) and tested on a sample of pilots depending on available resources. This qualitative human-centered evaluation looks at aspects such as procedure structure and clarity. The approval process is ad-hoc and leaves many potential inefficiencies unexplored.

From the literature, the quality of an SOP can be measured using subjective and objective measures. Subjectively, SOPs are evaluated by Subject Matter Experts (SMEs) [20]. Objective measures include testing in a simulator, on a sample of pilots, and testing operations in the field to ensure the procedure's outcome matches the intended result. Barshi et al. [6] describe one form of objective evaluation: data from current operations is collected and analyzed before being compared to data from operations after implementing a change to an SOP. The drawback of this method is that procedures are tested late in the development phase. i.e., only *after* the SOP has been implemented and distributed to the pilots; additionally, the full range of plausible human, machine, and environmental conditions cannot be covered using the proposed method, leaving many states where the SOP can fail [20], [21].

### 1.5.3 SOP Publishing Process

After expert inspection and simulator trial runs, approved SOPs get published electronically. Electronic formats, such as PDF or XML schemas, facilitate rapid organizational distribution.

### 1.5.4 SOP Training & Monitoring in Operations

After digital publications, operators undergo training. For example, pilots get trained on the published procedures using the simulator. Once in the field, operations performance is continuously measured through Flight Data Monitoring (FDM) and Flight Operations Quality Assurance (FOQA).

### 1.5.5 Bottlenecks with the Current Process

The current process presents several issues and bottlenecks: (1) limited testing and (2) lack of digitization. Firstly, SOPs lack thorough regulatory oversight and validation, especially for abnormal and emergency scenarios. Reasons for this include the cost-prohibitive nature of human-in-the-loop testing. Additionally, new SOPs are informally tested during training. This means the process depends on the human operator to identify any issues. Most importantly, the variability in operators and environmental conditions is not accounted for during testing, which limits the range of tested conditions. Secondly, SOPs primarily exist as static documents. This prevents the analysis and simulation of SOPs to identify any safety improvements.

## 1.6 TOOLS FOR SOP DEVELOPMENT AND MANAGEMENT

During the interviews, several tools used currently by the industry for procedure management were identified, including PRIDE, Arbortext, Procedure Genius, and ePROC. Each tool contains functions that can be directly linked to one of the five functions in Figure 3. All tools are Digital Procedure Management Tools, which provide organizations the ability to (1) create procedures, (2) distribute them to resources and personnel in a step-by-step electronic format, (3) monitor operator performance (some

tools gather task completion time, while others gather binary completion status), (4) publish procedures primarily to XML-based outputs, and (5) manage procedure revisions to ensure consistency and compliance. None of the tools use simulations or time-based measures to aid the management process.

### 1.6.1 SOP Creation Tools

Arbortext [22] is an advanced XML-based editor for authoring structured technical information documentation. Its customizable interface enables writers to create and edit XML-marked documentation. Authors can configure revisions and reuse content with Arbortext to ensure accurate and consistent documents. Vulcan is a customization of Arborterxt by Lockheed Martin (email P. Evans, personal communication, December 22, 2021).

PRIDE's procedure development tool, PRIDE Author [23], enables the creation of executable procedures through a drag-and-drop interface. Authors can construct procedures hierarchically from steps, blocks, and instructions covering actions, verifications, and other procedural constructs. PRIDE Author outputs and XML-based procedure capable of supporting human spaceflight operations and automation.

### 1.6.2 SOP Approval Tools

The Aviation Task Analysis Tool, described in [6], is a framework to analyze the constraints imposed on procedures by the technology, humans, and the operational environment. The tool utilizes a table or spreadsheet format, where each row lists the high-level phase. Columns capture the technological capabilities and limitations, human cognitive processes and constraints, and operational environment capabilities and

limitations. This structure facilitates identifying conflicts between mission needs and imposed constraints across those domains. Cells document observed problems and potential mitigating solutions per step.

### 1.6.3 SOP Publishing Tools

Arbortext enables procedure publishing to customers and users. Arbortext automatically assembles and publishes documents in printed and electronic formats based on the procedure's intended audience. It provides authors with the capability of aggregating content into publications that are dynamic and personalized.

### 1.6.4 SOP Training Tools

PRIDE View is an interactive procedure interface that provides procedures in a human-readable format. The procedure and boxes indicate whether the operator completed the step are presented to the operator. As the operator completes steps in the procedure, they are checked off to show completeness. It also provides a timer for steps that require a delay.

Like PRIDE View, Procedure Genius provides operators with a step-by-step procedure and enables users to check off steps as they are completed. It also allows operators to view procedures from any device.

### 1.6.5 SOP Monitoring Field Operation Tools

Procedure Genius [24] provides management tools to develop, track, and guide operators through procedures. Procedure Genius collects data, such as task completion time, to monitor workers in the field. PRIDE Monitor contains a database with records

for all actions. It collects real-time data and allows procedures to be analyzed based on historical data.

### 1.6.6 Bottlenecks of Current Tools

Tools provided thus far enable SOP workflow management rather than SOP analysis. The tools offer capabilities in authoring, publishing, distributing, and revising current SOPs. While applicable, these tools cannot quantitatively analyze SOPs. Thus, SOP creation and management rely heavily on the simulator's qualitative SME reviews and limited operational trials. Consequently, vulnerabilities in the design of SOPs can go unnoticed until after SOP deployment.

## 1.7 Problem Statements

Combing the bottlenecks of the current process identified in section 1.5.5 and the bottlenecks of the tools in section 1.6.6 yields the following problems within the current process:

1. The current testing process does not account for variability in ToP and/or AOTW. Accident analyses show that a short AOTW and/or a long ToP are causal factors. This gap is a result of:
    1.1. Lack of a machine-readable canonical structure for SOPs: SOPs are written and stored as static documents in an inconsistent natural language. Thus, there is an inability to parse SOP text.
    1.2. Lack of executable models that enable simulations to evaluate the quality of SOPs.

1.3. Lack of quantitative metrics to assess the content of SOPs and the human performance when executing the SOPs.

## 1.8 UNIQUE CONTRIBUTIONS

This dissertation addresses several issues in existing SOP practices. Testing of SOPs does not account for variability in ToP and/or AOTW. The proposed solution presents a formal SOP analysis and simulation framework that addresses this limitation. This is done through the following: (1) the development of a canonical SOP representation language (e-PRL) enabling machine readability, (2) the development of a method to convert text-based SOPs into the canonical structure, and (3) a computational analysis though (3.1) simulations of human performance and (3.2) quantitative SOP content metrics is proposed.

Table 1 summarizes the contributions. This research moves the SOP development from an ad-hoc text-based process towards a data-driven model-based approach that formalizes SOP language, structure, and analysis methods.

**Table 1 Summary of Problems and Contributions of this Dissertation**

| Problem | Contributions |
|---|---|
| 1. The current testing process does not account for variability in ToP and/or AOTW. Accident analyses show that a short AOTW and/or a long ToP are causal factors. | This dissertation presents a formal SOP analysis and simulation framework through an HMI model of SOPs that accounts for variability in human operators and environmental conditions. |
| 1.1 Lack of a machine-readable canonical structure for SOPs: SOPs are written and stored as static documents in an inconsistent natural language. Thus, there is an inability to parse SOP text. | 1.1.1 This dissertation presents a canonical procedure structure as an extension to the Procedure Representation Language (e-PRL).<br>1.1.2 This dissertation presents a formal methodology to convert text procedures into a canonical structure manually and using Large Language Models (LLMs). |
| 1.2 Lack of executable models that enable simulations to evaluate the quality of SOPs. | 1.2.1 This dissertation presents a model for SOP analysis through executable Model-Based Systems Engineering (MBSE) Action Diagrams. |
| 1.3 Lack of quantitative metrics to assess the content of SOPs and the human performance when executing the SOPs. | 1.3.1 This dissertation presents a set of quantitative metrics to evaluate the content of procedures (SOP Step Type, SOP Recall Score, and Training Requirements).<br>1.3.2 This dissertation presents a set of quantitative metrics to assess human performance (ToP VS AOTW, PFtC) |

### 1.8.1 List of Published Papers

This section lists eight published papers (1 journal and seven conferences) by the author related to this dissertation.

1. Bashatah, Jomana A., and Lance Sherry. “A Model-Based Approach for the Qualification of Standard Operating Procedures.” In *2021 Integrated*

*Communications Navigation and Surveillance Conference (ICNS)*, 1–10, (2021). https://doi.org/10.1109/ICNS52807.2021.9441587.

2. Bashatah, Jomana A., Lance Sherry, Steve Dam, Lauren Flenniken, Patrick Hartmann, and Tom Harold. "Analyzing Standard Operating Procedures Using Model-Based Systems Engineering Diagrams." *INCOSE International Symposium* 31, no. 1 (2021): 1130–44. https://doi.org/10.1002/j.2334-5837.2021.00891.x.

3. Bashatah, Jomana, and Lance Sherry. "Model-Based Analysis of Standard Operating Procedures' Role in Abnormal and Emergency Events." *INCOSE International Symposium*, (2022).

4. Bashatah, Jomana, and Lance Sherry. "Lessons Learned from Human Operator Intervention for AI Navigation and Flight Management Systems." In *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, 1–15, (2022). https://doi.org/10.1109/ICNS54818.2022.9771518.

5. Bashatah, Jomana, and Lance Sherry. "Usability Analysis of an MBSE Model of Standard Operating Procedures." In *INCOSE International Symposium*, 32:38–47, (2022). https://doi.org/10.1002/iis2.12870.

6. Bashatah, Jomana, and Lance Sherry. "Method for Formal Analysis of the Type and Content of Airline Standard Operating Procedures." In *2023 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, 1–8, (2023). https://doi.org/10.1109/ICNS58246.2023.10124313.

7. Bashatah, Jomana, and Lance Sherry. "Prompt Engineering to Classify Components of Standard Operating Procedure Steps Using Large Language Model

(LLM)-Based Chatbots.” In 2024 Integrated Communication, Navigation and Surveillance Conference (ICNS), (2024).

8. Bashatah, Jomana, and Lance Sherry. “Prompt Engineering for Classifying the Components of SOP Steps to Enable SOP Simulation and Analysis” submitted to Int J. Human Factors Modelling and Simulation – Under Review

# CHAPTER TWO: LITERATURE SURVEY OF MODELS OF HMI INTERACTION AND SOP LANGUAGES

This section consists of a review of the models of Human-Computer Interaction (HCI) related to Standard Operating Procedures (SOPs) for quantitative analysis and simulation of SOPs.

## 2.1 MODELS OF HUMAN-MACHINE INTERACTION

An operator's interaction with the computerized systems on board complex systems to fulfill a mission task is governed by SOPs. This interaction can be considered a "Command and Control" system [25], where the operator commands the system based on the state of the automation and the environment that it operates within. The operator's role in this command and control system is reflected in a series of "OODA loops" [26]. An OODA loop is a feedback loop consisting of four (4) phases: Observe, Orient, Decide, and Act. An operation governed by an OODA loop is as follows: the operator *observes* stimuli to initiate some response. The operator then *retrieves* information from mental models surrounding what should be done in response to this stimulus. Then, the operator *devises* the correct plan of action before *making a change* to an input device.

This concept is similar to the Human-Information Processing (HIP) system [27], where the operator acts on a "Recognize-Act" cycle. Applying HIP theory to HCI led to a task-analysis-based approximation and calculation of human performance [28]. The properties of this task-analysis-based calculation allowed for technical analyses of tasks without the need for empirical data. Models that fit that description, called Zero

Parameter models [27], enable task analysis to be completed early in the design phase because they do not require empirical data. Card, Moran, and Newell [28] introduced the Model Human Processor (MHP), providing a framework for these Zero Parameter models, and could successfully predict execution times for short tasks. These models include GOMS, along with its variations KLM and CPM-GOMS.

### 2.1.1 Goals, Operators, Methods, and Selection Rules (GOMS)

Card, Moran, and Newell [35] introduced the Goals, Operators, Methods, and Selection Rules (GOMS) model to produce Zero-parameter models. A GOMS model describes the Methods (GO**M**S) necessary to accomplish a Goal (**G**OMS). Methods are a series of operators (G**O**MS) performed by the user. Since multiple methods may be performed to achieve a rule, selection rules (GOM**S**) are employed to aid in choosing what method to accomplish that goal [29]. A fully developed GOMS model provides predictions of human performance and predictive estimates of how much additional time is spent when information is displayed differently [30].

Goals in a GOMS task analysis are defined as the outcome that the user wants to accomplish. Goals may be broken into subgoals using a hierarchical tree, where multiple goals can be active [30]. Defining goals must be extended to include the specific system and the context in which the system is intended to run to ensure that the system's design works well in the operational environment [29]. Operators are actions or efforts toward accomplishing the goal(s). Operators can be perceptual, cognitive, motor acts, or a composite of these. Note that operators are not users. Users perform the operations to change the environment or change their mental state. [30].

While goals define what the system must do, methods define how they are done on the plan and are driven by the system design [29]. They comprise a sequence of operators that accomplish a goal or subgoal. [30]. The process of describing methods is very complex due to the complexity of the human user's mental function. Determining what methods need to be described depends on answering one question: does this method affect the design of the interface? The GOMS task analysis should include the method if the answer is yes. Selection rules are necessary when more than one method can be used to accomplish a goal and can be achieved from the user's experience or training [30].

The difference between goals and operators is the level of detail. Suppose the analyst sees that an activity can be unpacked into more detail; that activity is a goal. In other words, an operator is not broken down, unlike goals, which are broken down [30]. To determine the level of detail to decompose operators, analysts follow two constraints: the lowest-level operators (primitive operators) must have constant time durations (a mouse click always has the same time duration regardless of user experience and surrounding context), and the time durations must be already estimated in the literature. These two constraints mean that most GOMS models will stop at the "Key Level"[30].

### 2.1.2 Key Level Model

Keystroke-Level Model, or KLM, another task-analysis-based model, is based on listing all the operators. The goals are decomposed until the primitive operators are at a key-stroke level. These primitive operators have pre-defined times and are used to predict the overall time of the task. The original key-level operators are (i) press on a button, (ii) point with a mouse, (iii) hands on a keyboard (or any other typing device), (iv) prepare

mentally to perform a task, (v) wait for the system to respond to an action. Each primitive action has a constant execution time or a parameterized value, such as the number of letters in a word to type [30].

KLM holds several advantages. Being the simplest form of GOMS makes it easier to apply to interface designs. It also allows a fast estimate of execution times for a task, making it the GOMS variation used to perform comparisons of design ideas, as well as perform sensitivity analysis and explore the design space. However, it is considered a restricted variation of GOMS since each task must explicitly have a method that accomplishes it and only contains primitive operators [31].

### 2.1.3 Cognitive Perceptual Motor – GOMS (CPM-GOMS)

According to Jonh and Kieras, 1996 [30] and John et al. 2002 [32], Cognitive, Perceptual, Motor GOMS or CPM-GOMS is the variation of GOMS to use when attempting to model overlapping activities. The key to this analysis is to decompose the goals into operators, where the low-level "primitive" operators are known as cognitive, perceptual, or motor actions. CPM-GOMS assumes that operators are not performed sequentially but in parallel depending on the task demands. Because of this dependency, a CPM-GOMS model is not depicted as a hierarchical tree; instead, the model is described as a pert chart. Another assumption of the CPM-GOMS model is that it only predicts the performance of highly skilled users.

The CPM-GOMS model uses the Model Human Processor (MHP) architecture. This architecture models humans as perceptual, cognitive, and motor processors.

Information is first recognized and stored in the perceptual processors, which triggers the cognitive processors to command the motor processors to act upon that information [30].

GOMS and its variations are built on task analysis techniques described above and computational cognitive architectures. Computational cognitive architectures "represent human information processing in terms explicit enough to be run as a computer program" [27, p. 69]. In addition, they provide predictions that can be empirically validated. Several architectures have been used in HCI and GOMS models, such as ACT-R/PM [33], Soar [34], and Apex [35].

### 2.1.4 Adaptive Control of Thought – Rational (ACT-R)

Adaptative Control of Thought (ACT) was first introduced by John Anderson in 1976 [36] as a theory of higher-level human cognition, such as deduction and problem-solving [37, p. 1], which are explained by lower-level functions such as pattern recognition. In 1993, ACT was expanded by Anderson [33] to include ACT-Rational (ACT-R). ACT-R assumes human cognition as a result of interactions between declarative knowledge, i.e., factual information, and procedural knowledge, i.e., unconscious/skilled behavior. Declarative knowledge in ACT-R is never lost, i.e., it becomes inactive and may be retrieved if needed. It is represented as chunks, a unit of declarative knowledge. Chunks are described as a propositional network of nodes with spreading activation. That means only a subset of the nodes is activated at a given instance, and activation spreads to other nodes down the network path. Procedural knowledge is modeled as a production system, following Newell's PS system [38]. In

production systems, a set of IF-THEN rules, the unit of procedural knowledge, are defined to specify what states of mind follow each other.

Thus far, ACT-R has been applied to purely cognitive domains in desktop-style interfaces and cannot estimate complex multi-modal interfaces such as aircraft. To overcome this issue, ACT-R/Perceptual Motor was introduced by Byrne & Anderson [39] as an extension to ACT-R that is well suited for predicting human performance in domains that integrate cognition with perceptual and motor systems [40]. ACT-R/PM adds four (4) modules (vision, motor, speech, audition) to the two (2) modules (procedural knowledge, declarative knowledge) introduced earlier in ACT-R. The modules run in parallel, meaning a person could be retrieving information from declarative memory while the vision module receives information and the motor module acts. Once a chunk has been seen, the basic knowledge of the chunk, such as the location, size, color, etc., is passed to the vision module before requesting to shift attention to that chunk. The motor module represents any physical actions, such as mouse clicks. The time for executing motor modules depends on the movement type and the distance the movement will travel. All execution times are based on Fitts' Law [41]. ACT-R/PM models use the exact learning mechanisms as ACT-R.

One drawback of this architecture is the assumption of execution times. ACT-R/PM assumes that execution times for parameters in the visual and motor module are drawn from rectangular distributions where the parameters are the mean values (found from Fitts Law and empirical data [42]) and the coefficient of variance is 1.3. Byrne (2001) [40, p. 13] provides values for several parameters. Moreover, Fitts' Law does not

estimate the time for finding or identifying a target but only estimates the movement time.

### 2.1.5 SOAR

Another cognitive architecture, Soar, has been under development for more than thirty years [43], [44]. The architecture of Soar is made up of (1) primitive long-term memories: (i) procedural memory, (ii) semantic memory, (iii) episodic memory along with (2) working memory, (3) perceptual short-term memory, and (4) perceptual long-term memory. Processes in Soar include (1) perception, (2) action, (3) learning through (i) reinforcement learning, (ii) chunking, (iii) semantic learning, and (iv) episodic learning. Information processing in Soar starts with information captured through the perception process and stored in the short-term perceptual memory. Symbolic structures from the information are then extracted and added to the working memory. Working memory retrieves knowledge from long-term memory and is the basis of action-taking. Each long-term memory, i.e., procedural, semantic, and episodic, is independent and has a separate learning mechanism associated with each type. Information processing is controlled by procedural memory.

Soar uses the Problem-Space Computational Model (PSCM) as the backbone for computations of human performance [43]. The problem-space framework specifies that an agent is always in a situation that requires applying knowledge to choose an action from a set of alternative actions to solve a problem. The set of alternative actions in this context is referred to as *operators*. The definition of problem-solving in this framework is the search for a problem (the initial state and a set of desired states) within a problem

space (the set of possible states the system can attain at a given time). This is accomplished through a problem-space search; the agent selects operators to move from its initial state to another state that gets the system closer to its objective over time. Thus, the PSCM is made up of functions to retrieve or generate [44] (1) problem spaces (set of available operators and available states), (2) operators (means by which an agent changes a state), and (3) an initial state. These functions are performed in procedural memory using production systems, i.e., behavior is controlled by applying clear production rules. Operator selection is made up of an elaboration phase and a decision procedure. During the elaboration phase, information in the working memory is matched to production rules (in LTM). All rules capable of firing do so, changing the state and suggesting or preferencing an operator. This allows the architecture to collect all operators and their associated changes before deciding. The appropriate operator is selected in the decision procedure phase, assuming no conflicts arise.

So far, ACT-R/PM and Soar can be used for modeling human behavior, thus predicting task completion times. On the other hand, Apex, developed by Michael Freed [42], attempts to predict errors that could arise during task completion. Apex is a computational framework for automating the creation of CPM-GOMS models, which are complex and time-consuming [32], [45].

### 2.1.6 Time-Based Models of Standard Operating Procedures

Several time-based models evaluate human-machine interactions when performing steps towards a primary goal, i.e., completing the steps in an SOP. Models

such as the Pilot-Autonomy Workload Simulation (PAWS), Operational Event Sequence Diagrams (OESDs), Dynamic Value Action Stream Diagram, HCIPA, ToP & AOTW have been used to assess the performance of a procedure.

#### *2.1.6.1 Pilot-Autonomy Workload Simulation (PAWS)*

The Pilot-Autonomy Workload Simulation (PAWS) estimates the pilot workload using a utilization-based approach [46]. PAWS utilizes a discrete event simulation to estimate the workload of procedure and checklist execution. To generate the estimates, a generalized task-flow model was combined with SMEs' judgment to estimate task completion times. The resulting task-completion time distributions were lognormal distributions with several associated issues. First, lognormal distributions can have infinite tails; thus, task completion times can be relatively long. Second, the estimates are based on SMEs, thus are subjective, and might not account for novice operators.

#### *2.1.6.2 Operational Event Sequence Diagrams (OESDs)*

Operational Event Sequence Diagrams (OESDs) were developed by Kurke [47] to represent operator decisions and multi-person tasks. Subsequently, these models have been used in HMI. Sorenson et al. [48] and Harris et al. [49] used such models to analyze activities on the flight deck. Moreover, Huddlestone et al. [50] used these models to study the taxi-in after landing phase of the flight. OESDs can offer a straightforward framework for evaluating the prospective task distribution. A time-based sequence of tasks, including interactions between operators and technological artifacts, is displayed in an OESD's output. Using a set of standardized symbols, it graphically represents task processes often established following an initial task analysis. To generate OESDs,

observations, analysis of SOPs, and interviews. Each step is represented by a symbol placed in one of several timelines, often known as "swim lanes," each committed to a specific human or automated agent. Symbols used to generate OESDs are shown in Figure 4.

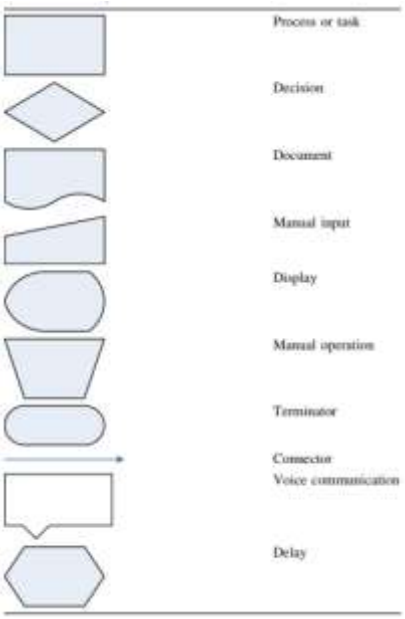



**Figure 4 OESD Symbols [50, Tbl. 1]**

### *2.1.6.3 Dynamic Value Action Stream Diagrams*

Value stream mapping is a production process technique used to visualize time and information flow through a system. Zhang et al. [51] proposed using dynamic value stream mapping to optimize astronauts' extravehicular activities (EVAs) during space missions. The authors developed a fuzzy action value stream map to analyze astronaut EVA procedures.

The astronauts' mission is decomposed using a task-analysis approach, where the atomic low-level actions are one of 18 therbligs [51] defined in Gilbreth's time and motion study. The initial time data for astronauts was predicted based on the MODular Arrangement of Predetermined Time Standards (MODAPTS) [27], which results in a standard value stream map. The actual time is observed in real-time, i.e., during the astronauts' EVA mission, and the standard times are updated so that a future value stream map is drawn.

After gathering real-time data, the completion rate compared to the standard time is assessed using Equation 1. A negative ratio corresponds to actions that take longer than expected.

**Equation 1 Standard Time Completion Ratio**

$$Standard\ time\ compeltion\ ratio = \frac{Low\ Time\ (UT) - Actual\ Time\ (AT)}{Upper\ Time\ (UT) - Lower\ Time\ (LT)}$$

*2.1.6.3.1 Time Data*

MODAPTS is a simple measure of the time it takes to carry out short to medium-length repetitive tasks. It is based on the fact that each action can be defined in terms of body movements, and body movements can be expressed in terms of an integer multiple of the basic finger movement, a mod (1 mod = 0.129 seconds). MODAPTS contains 21 predefined alphanumeric codes (Figure 5), where the alpha code represents the type of activity and the number reflects the integer multiple.

**Figure 5 MODAPTS Codes**

While MOADPTS assumes a predefined time for each action, Zhang et al. (2023) declare that the time for astronauts' actions is a fuzzy triangular number. This is due to the variance in physical abilities of different astronauts and external disturbances. Additionally, Cao (2017) [52] studied the difference in action times between ground personnel and weightless environments. Action times for subjects performing the same task in a weightless environment were considerably longer. Therefore, the basic mod value was assumed to have an optimistic value of 1.7 seconds and a pessimistic value of 2.1 seconds.

#### *2.1.6.4 HCIPA*

HCIPA [53] is a task analysis method to enable quantitative evaluation of SOPs using human performance metrics for training and operational use. HCIPA is based on

the RAFIV model that describes pilot cognition during HMI [54]. First, the operator mentally Reformulates how automation capabilities will be used to achieve mission tasks. Next, the operator must perform a physical action to transfer the formulation to the machine. This is done in three steps. The first step is to Access the appropriate display. Then, the operator Format required data entries. Finally, the operator Inserts the inputs. The last step in this model is to analyze feedback from the automation to Verify that (1) the automation accepted input, (2) the automation is performing as expected, and (3) the task is satisfying mission and task goals.

Steps in HCIPA can be classified into two main types of actions: (1) decision actions and (2) physical actions. Steps combine both a decision and an action [53].

Performance is evaluated using two metrics: (1) Training Time for Recall Steps and (2) Reliability of Recall Steps, also known as the Probability of Failure to Complete. Procedure steps that rely entirely on memory recall without salient cues impose heavy training loads and are prone to failure, especially for infrequent tasks. Hence, user interfaces minimize required memory recall through salient cues to aid operator performance.

#### *2.1.6.5 ToP & AOTW*

Kourdali and Sherry [20], [21] & Bashatah and Sherry [25], [55] used Monte Carlo (MC) simulations to estimate human performance when completing an SOP. Human performance of the SOP can be evaluated using time-based measures, which provide a means for simulation of human performance considering all human and environmental conditions. Human performance is measured by the time to complete a

procedure (ToP) compared to the allowable time to complete a procedure before a hazardous event occurs (AOTW) [20], [25]. When the ToP exceeds the AOTW, the SOP fails to meet its goal. The surplus or shortage of the ToP, when compared to the AOTW, results in Procedure Buffer Time (PBT). Repeated performance of the SOP by operators with various levels of experience (varying ToP) under varying operational conditions (varying AOTW) results in a distribution of PBT. When PBT is negative, the left tail of the distribution is the Probability of Failure to Complete (PFtC). At the heart of the MC simulations are time distributions driven by the saliency of cues and the frequency of completing steps.

*2.1.6.5.1 Time Data*

Aircraft flight data from Flight Data Monitoring (FDM) and Flight Operations Quality Assurance (FOQA) data [20] was leveraged to generate distributions characterizing the time needed to complete SOPs. Timestamps marking the start and end events for a given procedure segment across multiple flights provide empirical samples of duration that describe the Time on Procedure (ToP) distribution. The resulting time distributions capture variance in the human operator, the machine, and the operational environment.

### 2.1.7 Summary of Models Related to Standard Operating Procedures (SOPs)

Since Card, Moran, and Newell introduced the GOMS task analysis techniques in 1983, research has been carried out to help create these models. Multiple tools have surfaced that automate the creation of GOMS models. These models, however, carry some limitations. They assume constant times for execution time estimates and consider a

fixed category of users. Furthermore, these models do not count for any time requirements or time before a task must be done. Nor do they account for the variability in that time requirement.

Other task-analysis-based models, such as HCIPA, PAWS, OESDs, and ToP & AOTW, provide a framework for evaluating task times when completing SOPs. These models, however, come with certain limitations. HCIPA, based on the RAFIV model, decomposes the step into six steps, which can result in massive models for longer SOPs. PAWS utilizes lognormal distributions, which can result in excessively long tails for task times. OESDs do not capture explicitly each of the required elements. For example, the OESD model captures the display and the information on the display in one symbol. Moreover, it captures the decision but not the necessary information to make it or the display from which the data can be retrieved.

## 2.2 Modelling Languages for Standard Operating Procedures (SOPs)

Previous researchers have developed a set of languages used for SOPs. Explored in this section are (1) Procedure Representation Language (PRL), (2) PLEXIL, and (3) BPMN. While these languages have been used to model SOPs, they lack the granular decomposition of cognitive, perceptual, and motor aspects of SOP steps,

### 2.2.1 Procedure Representation Language (PRL)

The basis of PRIDE (introduced in section 1.6) is the Procedure Representation Language (PRL), which was developed to support the increased attention to automated procedure execution [56]. PRL is an executable language understood by the machine and the human, and its data structure is described in Figure 6.

Procedures in PRL consist of steps. Steps are made up of Blocks, which are wrappers that contain instructions. Instructions are the most complex components of PRL, and they hold the detailed information to accomplish the step, i.e., what action to perform. Each instruction performs a different action. Instructions are grouped to make up a procedural step based on a specific end goal or purpose. Finally, a set of procedural steps are grouped to compose the procedure.

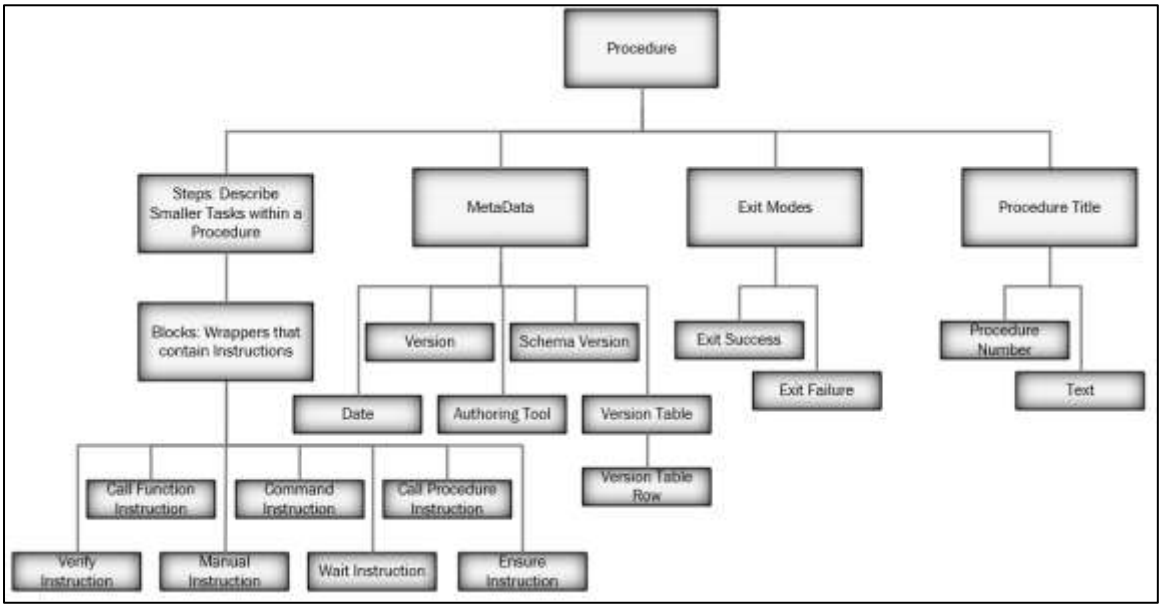



**Figure 6 Hierarchical Structure of PRL**

Instructions are the lowest level of PRL definition. They include Command, Verify, Ensure, Wait, Call Procedure, Call Function, and Manual. A computer command is issued to the automation via a command instruction. The Verify instruction compares a

recorded value with a target value. If TRUE, the comparison result triggers the instruction to be accomplished. If the comparison result is FALSE, the Ensure instructions command the automation (similar to command instruction, but only occurs after a Verify instruction is FALSE). A Wait instruction is used when an instruction depends on a predetermined amount of time or until a Boolean expression becomes TRUE. The Call Procedure instruction calls another procedure and can sometimes wait until the called procedure is completed. A user-defined function can be called using the Call Function instruction.

### 2.2.2 PLEXIL

The Plan Execution Interchange Language (PLEXIL) [57] is a language developed by NASA to represent executable automation plans. Like PRL, PLEXIL comprises a hierarchical structure where high-level tasks are decomposed into sub-tasks. In contrast to PRL's focus on encompassing human and computer execution, PLEXIL focuses solely on autonomous systems. PLEXIL's structure is a tree with nodes. Node execution is driven by start and end conditions. A comparison of PRL and PLEXIL is depicted in Figure 7.

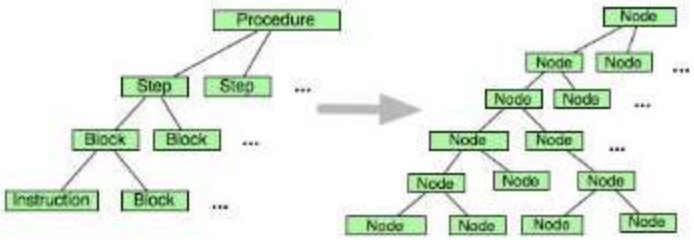



**Figure 7 PRL and PLEXIL Structures [57, p. 5]**

PRL and PLEXIL do not capture all the elements required to model SOP steps' cognitive and motor aspects. Both languages are focused more on automated procedure execution and lack the granularity needed to model human operators. SOP steps within the scope of this dissertation cannot be modeled using these languages.

### 2.2.3 BPMN

The Business Process Model and Notation (BPMN) is a modeling language that can also be used for procedures [58]. BPMN is based on a Business Process Diagram (BPD), a flowchart depicting activities and the execution order of each activity. This is similar to UML Activity Diagrams [59]. In addition, BPMN can be executed using the Business Process Execution Language (BPEL). A significant limitation of this language is that it lacks the hierarchical decomposition needed to explicitly define the perceptual, cognitive, and motor aspects of SOP steps.

### 2.2.4 Summary of Languages Used for Standard Operating Procedures

The Procedure Representation Language (PRL) provides a hierarchical machine-readable format with steps decomposed into blocks, and instructions focused on human-computer procedure execution. The Plan Execution Interchange Language (PLEXIL) is

another hierarchical structure but targets fully automated systems. The Business Process Model and Notation (BPMN) offers graphical workflow modeling constructs similar to UML activity diagrams and can be executed using the Business Process Execution Language (BPEL). However, none of these languages capture all the required elements to model the cognitive and physical aspects of human operators performing SOPs.

A consistent machine-readable representation of the different components of SOP steps is needed to enable quantitative analysis and simulations of procedures. This includes a granular decomposition of the SOP step specifying information elements, displays, decisions, and actions separately.

## 2.3 Summary of Literature Survey

Various human-machine interaction (HMI) models offer frameworks to enable quantitative assessments of execution times for operators carrying out procedures. Models such as GOMS, KLM, and CPM-GOMS leverage task analysis techniques to derive estimates without needing empirical data. While valuable, these models incorporate assumptions of constant execution times and user categories, which do not account for the known variance in the human operator and environmental conditions.

Other methodologies, including HCIPA, PAWS, OESDs, and the ToP & AOTW model, aim to capture time-based aspects but are limited because of their granular task decomposition or statistical distribution choices. Nevertheless, the ToP & AOTW approach incorporates simulations assessing completion times against available operational time windows while addressing human variability and environmental conditions. However, a gap that persists across all models surveyed is the lack of a

digital, executable representation that comprehensively captures cognitive and physical actions while executing SOP steps.

Several languages have emerged as potential solutions for digitally representing SOPs to facilitate automated or human-driven execution. Procedure Representation Language (PRL) and the Plan Execution Interchange Language (PLEXIL) are hierarchical machine-interpretable formats prioritizing autonomous procedure execution over human execution. Meanwhile, Business Process Model and Notation (BPMN) is a graphical workflow language similar to the Unified Modeling Language (UML) activity diagrams. However, none of these languages fundamentally capture the complexity of human perceptual, cognitive, and motor aspects fundamental to SOP execution.

# CHAPTER THREE: METHODOLOGY FOR ANALYSIS AND SIMULATION OF SOP STEPS

Standard Operating Procedures (SOPs) serve an integral role in guiding operator control of complex systems during emergency and routine scenarios. Despite their importance, typical methods for SOP testing lack quantitative analyses and simulations. SOP development has traditionally relied on testing on a limited number of pilots/simulators, a process unable to account for variabilities in SOP execution times and/or available operational time windows. Additionally, SOPs primarily reside as static documents in an inconsistent natural language rather than formal digitized representations, enabling executable modeling and simulations.

A survey of the literature identified models of Human Machine Interaction (HMI) that estimate user performance of SOP execution. Models such as GOMS, KLM, and CPM-GOMS employ zero-parameter models and task analysis techniques to predict the execution times of highly skilled users and do not account for any variance in the human operator. Other models, such as HCIPA, PAWS, and OESDS, aim to capture some variance through time-based measures but are constrained by their granularity and/or simplifying of statistical distributions. The ToP and AOTW model leverages Monte Carlo simulations to assess operator execution times against the available time while accounting for the variance in human operator and operational environment. Several languages, including PRL, PLEXIL, and BPMN, enable procedure automation, but they do not encapsulate the complexities of human perceptual, cognitive, and motor aspects that underly SOP execution.

The methodology addresses the problem that the current testing process does not account for variability in ToP and/or AOTW. Accident analyses show that a short AOTW and/or a long ToP are causal factors. However, traditional methods of testing do not account for this variance. This is because of the following:

(1) Lack of a machine-readable canonical structure for SOPs: SOPs are written and stored as static documents in an inconsistent natural language. Thus, there is an inability to parse SOP text.

(2) Lack of executable models that enable simulations to evaluate the quality of SOPs.

(3) Lack of quantitative metrics to assess the content of SOPs and the human performance when executing the SOPs.

The lack of analysis and quantitative testing of SOPs is solved through analysis of SOP step content and simulations of human performance. To do so, the digitization of SOPs is required. The digitization of SOPs involves the definition of a canonical structure and the conversion of text-based SOPs into models. A process to analyze SOPs by converting SOP steps into an executable model first and then examining simulation metrics and SOP content metrics has been developed.

### 3.1 Design of a SOP Analysis and Simulation System

The analysis process, displayed in Figure 8, starts with receiving text-based manuals from the Original Equipment Manufacturers (OEMs). These text-based manuals are converted to procedure manuals through a refined Large Language Model (LLM) model (section 3.1.1). The resulting procedures are then converted to a digital

representation (section 3.1.2.11 either through a manual annotation process (section 3.1.2.2) or through a trained LLM-based chatbot (section 3.1.2.3). The parsed procedures are stored in a database of procedures and steps and can be represented visually through SysML Action Diagrams (section 3.1.5). The SOP model is then analyzed using a set of defined metrics (section 3.1.3) and simulated to examine the human performance when completing the procedures (section 3.1.4). The following subsections describe each of these processes in detail.

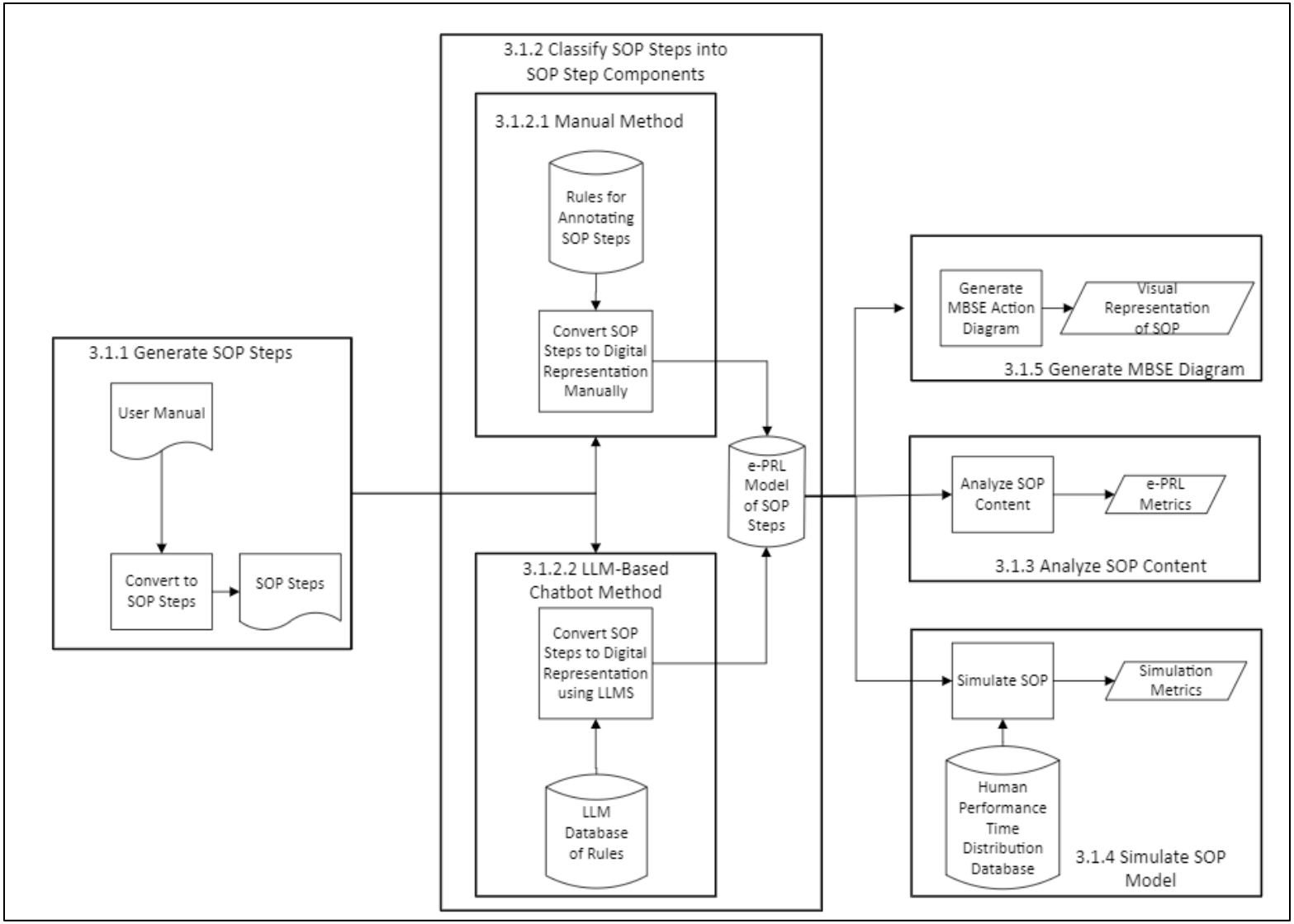



**Figure 8 Framework for SOP Analysis and Simulation System**

### 3.1.1 Generate SOP Steps

An owner's manual is a document supplied by the equipment manufacturer and serves as the reference guide for the equipment's safe operation. It is different than a procedure manual in that it contains long paragraphs of text as opposed to concise step-by-step instructions. Extracting step-by-step procedures from the owner's manual requires an extensive amount of time and understanding of the equipment and how it is used. The LLM-based chatbot is used to extract step-by-step procedures from the text.

To import the owner's manual, a chatbot's import button (highlighted in red in Figure 9) is used. Then, the chatbot is prompted to "write a procedure for operator intervention while using Full Self-Drive."

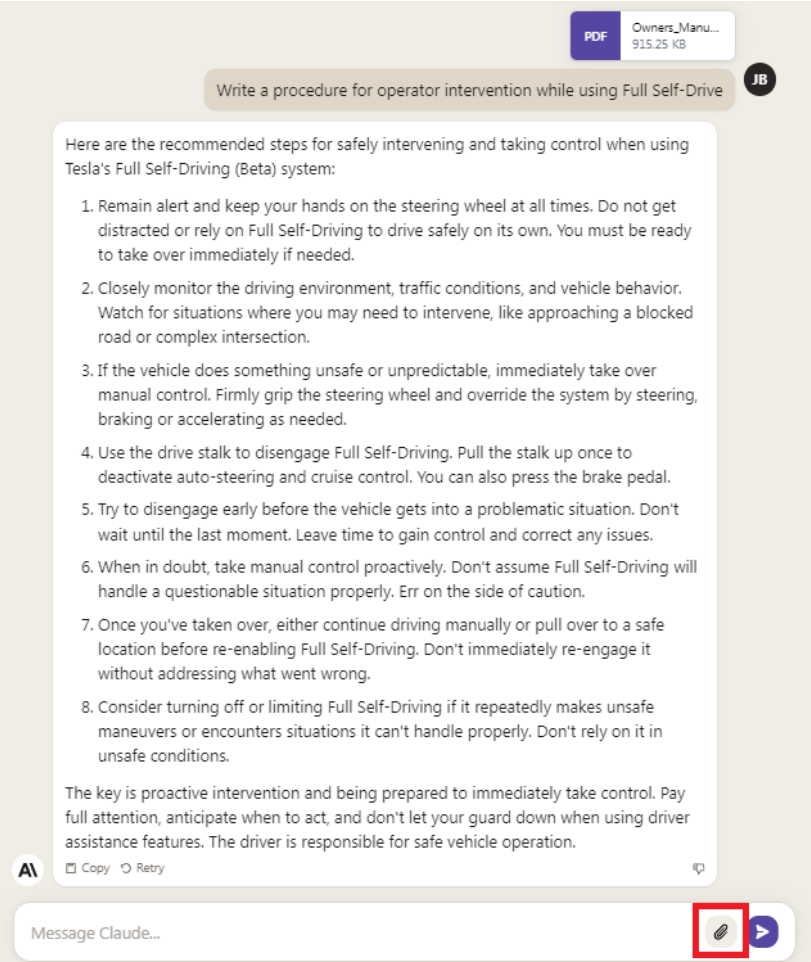



**Figure 9 Prompt to Generate SOP**

Two (2) of the steps generated by the chatbot are not trigger-action steps (#4 and #8). For example, "#4: Use the drive stalk to disengage Full Self Driving ….". This can be corrected by prompting the chatbot to understand that a step must contain a trigger and an action. The prompt "A step must contain an action and a trigger" is submitted to the chatbot (Figure 10). The chatbot produces the SOP in Figure 10, which resolves the issue with the absence of the trigger for #4 and #8.

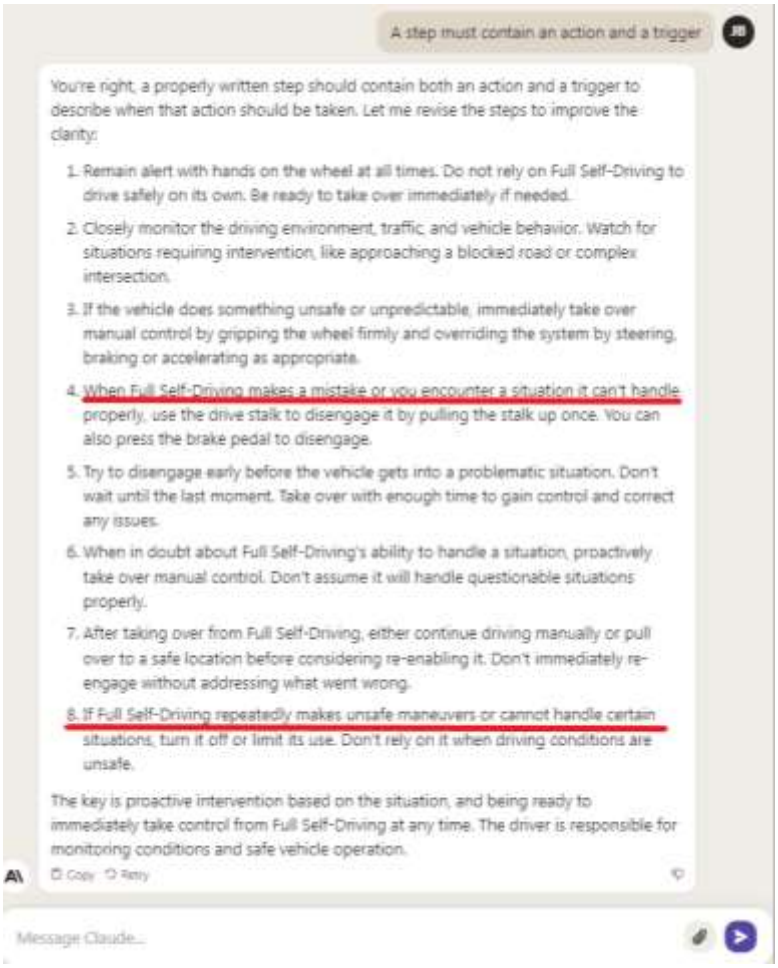



**Figure 10 Modified Version of Chatbot Generated SOP (Modifications Highlighted in Red)**

The procedure in Figure 10 identifies the triggering conditions for when an intervention could be required, such as “approaching a blocked road or complex intersection” or “when Full Self-Driving makes a mistake, or you encounter a situation it cannot handle.” It also identifies the correct course of action by “using the drive stalk to disengage” and “gripping the wheel firmly.”

The steps are further modified manually by excluding any non-actionable items and breaking compound steps into separate steps. This is to ensure that each SOP step contains a clear trigger and action. The new SOP and SOP steps are in Figure 11.

Operator Intervention while using FSD

1. Remain alert with hands on the wheel at all times.
2. Be ready to take over immediately if needed.
3. If the vehicle encounters a situation requiring intervention, like approaching a blocked road or complex intersection, closely monitor the driving environment, traffic, and vehicle behavior.
4. If the vehicle does something unsafe or unpredictable, immediately take over manual control by gripping the wheel firmly and overriding the system by steering, braking or accelerating as appropriate.
5. When Full Self-Driving makes a mistake or you encounter a situation it can't handle properly, use the drive stalk to disengage it by pulling the stalk up once.
6. When in doubt about Full Self-Driving's ability to handle a situation, proactively take over manual control.
7. After taking over from Full Self-Driving, either continue driving manually or pull over to a safe location before considering re-enabling it.

**Figure 11 Final SOP After Manual Modifications**

### 3.1.2 Classify SOP Steps into SOP Step Components

Taking existing SOPs and converting them to a canonical structure can be achieved through a manual process or through augmenting the process with an LLM-based chatbot.

#### *3.1.2.1 Canonical Structure of SOPs (e-PRL)*

To accomplish the objective of digitizing SOPs such that they can be executable models, the definition of a canonical structure is required. This section describes the canonical structure for SOPs that satisfies the goals mentioned above.

Operator-machine interaction is governed by trigger-decide-act cycles and reflects human cognition steps to (a) Observe, (b)Orient, (c) Decide, and (d) Act. The SOP steps follow this same model. Based on the SOP step model, the primary form of the SOP step structure can be defined. Each SOP step has the components (1) trigger (*perceptual*), (2) decision (*cognitive*), (3) action (*motor*), (4) waiting/timing (*perceptua*l), and (5) verification after waiting (*perceptual*). The basic structure of SOP steps is shown in Figure 12.

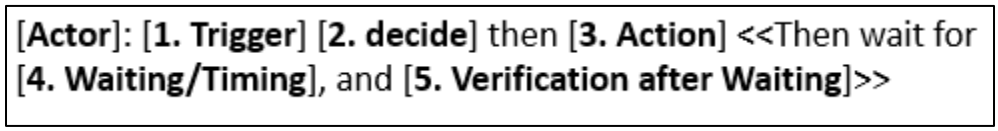



**Figure 12 Basic Structure of SOP Steps**

Each of the five components is further defined by the perceptual, cognitive, and motor elements (Figure 13)

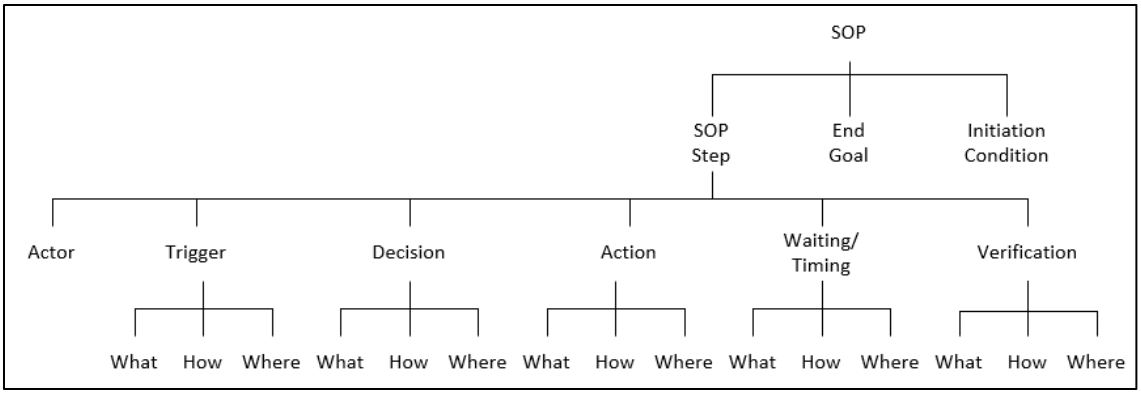



**Figure 13 Canonical Structure of SOP Steps**

Using the SOP step canonical structure in Figure 13, extended-PRL was defined. e-PRL is an extension of the Procedure Representation Language (PRL) [56] developed by Traclabs [23] for the International Space Station (ISS). e-PRL contains the following entities:

1. An operator who is responsible for the SOP step (Actor). E.g., *"Pilot Monitoring."*
2. Condition for initiating the SOP step (Trigger (What)). E.g., *"When aircraft reaches V1".*
3. Data required for initiating the SOP step (Trigger (How)). E.g., *"... V1 speed"*
4. Source of data for initiating the SOP step (Trigger (Where)). E.g., "*on the PFD airspeed tape."*
5. The decision to be made (Decide (What)). E.g., *"... if there are no alerts and the aircraft is configured appropriately for takeoff, then."*
6. Data required to make a decision (Decide (How)). E.g., "… *flaps set to FLAPS 20…."*

7. Source of data required to make a decision (Decide (Where)). E.g., "*... on the EICAS..."*
8. Action for the SOP step (Action (What)). E.g., *"Activate TO/GA Mode"*
9. The physical motion required for the action (Action (How)). E.g., *"by pressing...."*
10. The input device is used to complete the action (Action (Where)). E.g., *"... TO/GA Button on Throttle Lever..."*
    a. Some steps require the operator to wait and verify that a condition has been met. In that case, the following entities are also required:
11. Waiting requirements, if any, must be met (Waiting (What)). E.g., *"... and wait for engines to stabilize...."*
12. Data needed for fulfilling waiting requirements (Waiting (How)). E.g., *"... 50% N1..."*
13. Source of data needed to fulfill waiting requirement (Waiting (Where)). E.g., *"... on the EICAS..."*
14. Verification action after a waiting requirement has been met (Verification (What)). E.g.," *... verify engines have been stabilized...."*
15. Data required for verification (Verification (How)). E.g., *"... 50% N1... "*
16. Source of data required for verification (Verification (Where)). E.g., *"... on the EICAS..."*

#### *3.1.2.2 Classify SOP Steps into e-PRL – Manual Method*

The classification of SOP steps into components is defined as the process of (1) decomposing the SOP step into components and (2) classifying each component into the appropriate label. The former is known as SOP step annotation. To annotate the SOP step manually, the entities (or labels) must first be defined. The entities for this analysis were defined in section 3.1.2.1, which are the e-PRL elements. After defining the entities, the SOPs can be uploaded into an annotation tool, and the text is annotated using a defined (Figure 14). For each decision node, a path is taken based on the answers to the questions in Table 2.

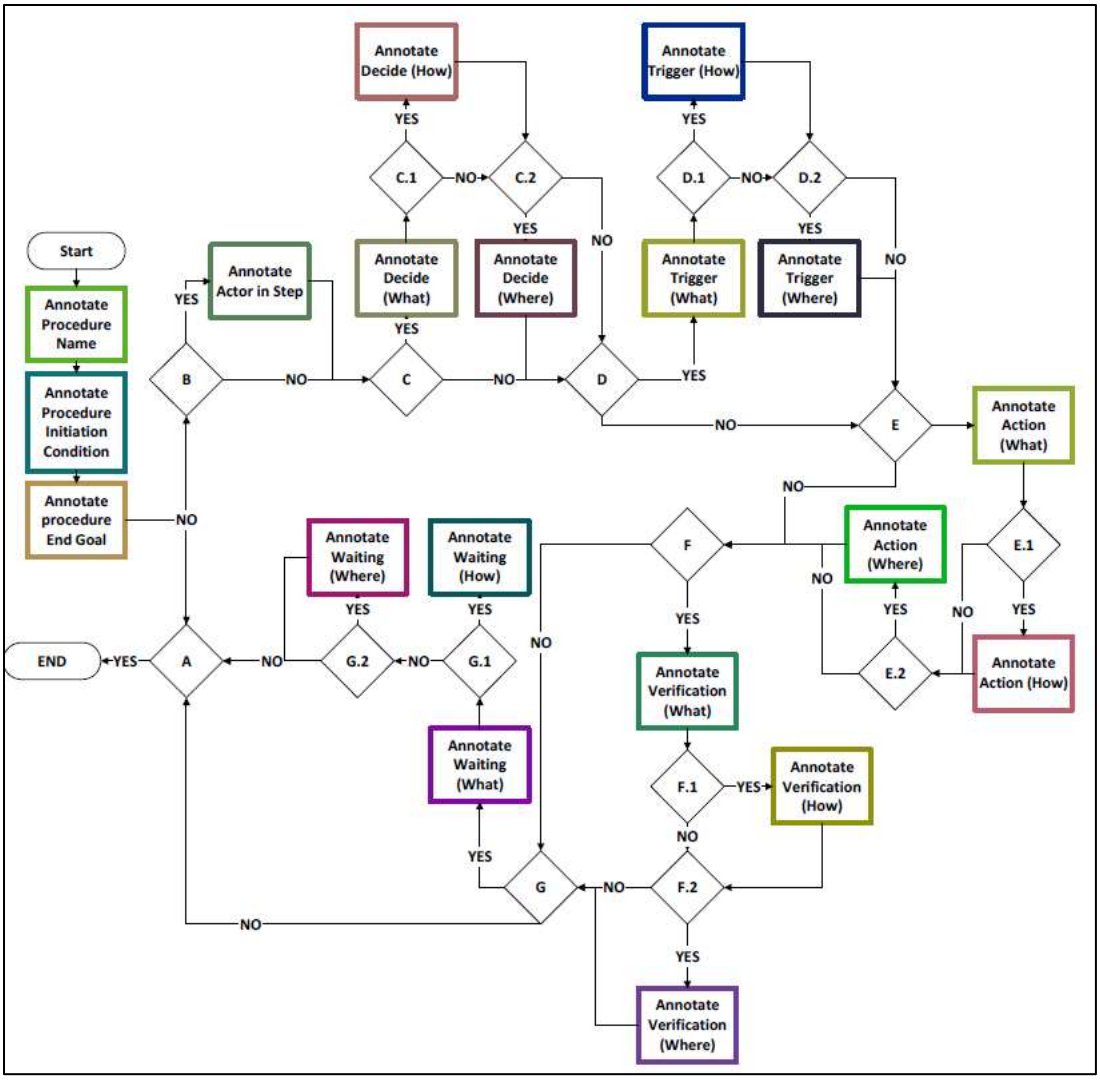



**Figure 14 Process of Annotating SOPs**

**Table 2 Questions for Decision Nodes**

| Decision Node | Decision | Path |
|---|---|---|
| A | Are All SOP Steps Annotated? | Yes: End annotation process. |
| | | No: Go to decision node B. |
| B | Is there an actor? i.e., PM, PF, Pilot Monitoring, Pilot Flying | Yes: Annotate the actor, then go to decision node C. |
| | | No: Go to decision node C. |
| C | Is there an explicit decision? i.e., "IF" | Yes: Annotate the decision, then go to node C.1. |

| | | |
|---|---|---|
| | | No: go to decision node D. |
| C.1 | Is the data required for the decision explicit? | Yes: Annotate the data required for the decision, then go to decision node C.2. |
| | | No: Go to decision node C.2 |
| C.2 | Is the source of the data explicit? | Yes: Annotate the source of the data, then go to decision node D. |
| | | No: Go to decision node D. |
| D | Is there an explicit initiation condition, i.e., "WHEN," "DURING," "AFTER," "BEFORE" | Yes: Annotate the trigger, then go to decision node D.1. |
| | | No: Go to decision node E. |
| D.1 | Is the data required for initiating the SOP explicit? i.e., speed, altitude. | Yes: Annotate the data required for initiating the SOP, then go to decision node D.2. |
| | | No: Go to decision node D.2. |
| D.2 | Is the source of the data (output display) explicit? i.e., "PFD," "ALTIMETER," "AIRSPEED INDICATOR." | Yes: Annotate the source of the data, then go to decision node E. |
| | | No: Go to decision node E. |
| E | Is the action explicitly stated? | Yes: Annotate the action, then go to decision node E.1. |
| | | No: Go to decision node F. |
| E.1 | Is the required physical motion explicitly stated? i.e., "PUSH," "SELECT," "ROTATE," "FLIP," "TURN," "PUSH AND HOLD," call-out | Yes: Annotate the physical motion, then go to decision node E.2. |
| | | No: Go to decision node E.2. |
| E.2 | Is the input device required for the action explicitly stated? i.e., "THRUST LEVERS," "BAROMETRIC PRESSURE KNOB," "MCP," "EICAS." | Yes: Annotate the input device, then go to decision node F. |
| | | No: Go to decision node F. |
| F | Is there an explicit verification? i.e., “VERIFY” | Yes: Annotate the verification action, then go to decision node F.1. |
| | | No: Go to decision node G. |
| F.1 | Is the data required for the verification explicitly stated? | Yes: Annotate the data needed for verification, then go to decision node F.2. |
| | | No: Go to decision node F.2. |

| | | |
|---|---|---|
| F.2 | Is the source of the data required for the verification explicitly stated? | Yes: Annotate the source of the data required for verification, then go to decision node G. |
| | | No: Go to decision node G. |
| G | Is there an explicit waiting requirement? i.e., "UNTIL" | Yes: Annotate the waiting requirement, then go to decision node G.1. |
| | | No: Go to decision node A. |
| G.1 | Is the data needed to fulfill the waiting requirement explicitly stated? | Yes: Annotate the data required to satisfy the waiting requirement, then go to decision node G.2. |
| | | No: Go to decision node G.2. |
| G.2 | Is the source of the data needed to fulfill the waiting requirement explicitly stated? | Yes: Annotate the source of the data needed to fulfill the waiting requirement, then go to decision node A. |
| | | No: Go to decision node A. |

#### *3.1.2.3 Classify SOP Steps into e-PRL - LLM-Based Chatbot Method*

Classification of SOP text into components (e-PRL) using an LLM-based chatbot (Figure 15) requires that the chatbot first understands what e-PRL and its labels are. This can be done by prompting the chatbot to create an internal database where the rules for classifying SOP text into e-PRL labels are defined. Next, the chatbot learns these rules by prompting the chatbot to learn the rules in the database. Next, the chatbot is asked to classify the text in a procedure into e-PRL labels. Next, a table containing the correct classifications is imported into the chatbot, and the chatbot is prompted to generate a list of new rules by comparing the correct classifications with the chatbot-generated classifications. The new rules are added to the chatbot's internal database, and the

process can be iterated again with more SOPs until an acceptable number of SOP steps and their components have been correctly classified.

The exact number of times the chatbot is required to understand the rules of labeling is not set fully. Instead, the user can continue the refinement process until satisfied with the chatbot's performance. Figure 15 shows the process for developing an SOP chatbot that can label or classify SOP text into e-PRL models.

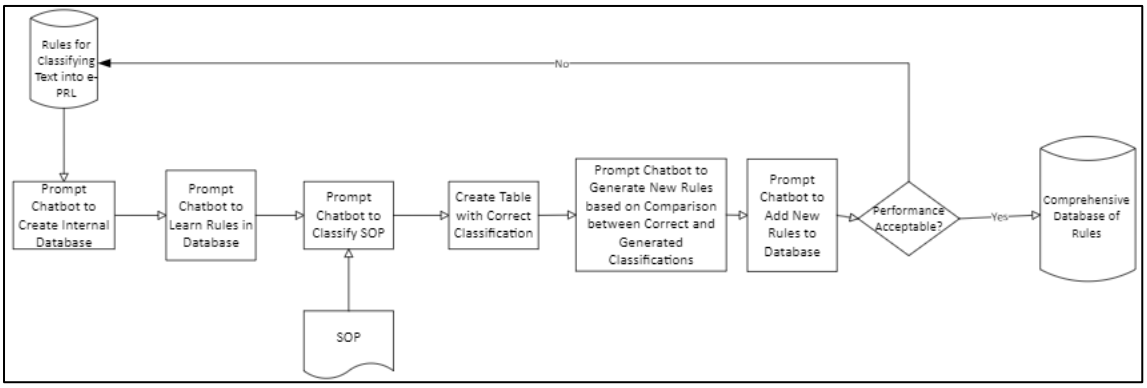



**Figure 15 Process for Training a Chatbot to Classify SOP Step Components**

### 3.1.3 Analyze SOP Content

Leveraging SOP step counts, memory metrics and training estimates facilitates the human-centered development of SOPs that consider human cognitive workload limitations and abilities. Analyzing SOP content by categorizing the steps in the SOP and quantifying memory and training metrics aids SOP designers in improving SOP development and training.

The categorization of SOP steps into (1) action-only, (2) decision-action, (3) action with waiting & verification, and (4) decision-action with waiting & verification yields insights into the complexity of the SOP. SOP steps that contain waiting requirements are subject to failure, especially if a verification requirement does not follow these steps. Steps that contain decisions should have the data required for the decision, as well as the source of the data identified, such that the operator can make the correct decision promptly.

Additionally, the SOP step recall score measures the memory demands placed on the operator. SOPs that require more memory (i.e., workload) may lead to errors and/or longer training times. Identifying these steps is crucial so that designers can modify these steps to relieve the operator from unnecessary workload. Finally, by estimating the total repetitions and training days needed for an SOP, SOP instructional programs can be planned accordingly.

Statistics of the SOP can be divided into two types: (1) Statistics for the type of SOP steps and (2) the content of the SOP steps. The following sections describe each of the metrics developed.

#### *3.1.3.1 SOP Step Type*

The type of SOP step is derived from the SOP canonical structure described in section 3.1.2.1. The basic form of an SOP step with (1) trigger, (2) decision, (3) action, (4) waiting, and (5) verification yields four (4) SOP step types. (1) Action Only Steps, (2) Decision Steps, (3) Action with Waiting & Verification Steps, and (4) Decision with

Waiting & Verification Steps. Table 3 summarizes the metrics associated with the type of SOP steps, which are basic counting functions that count the number of each type.

**Table 3 Metrics for SOP Content Analysis**

| Metric | Description |
|---|---|
| Number of Action-Only Steps | Number of steps characterized as "Action-Only" steps according to the flowchart in Fig 13. |
| Number of Decision-Action Steps | Number of steps characterized as "Decision-Action" steps according to the flowchart in Fig 13. |
| Number of Action with Waiting & Verification Steps | The number of steps characterized as "Action with Waiting-Verification" is according to the flowchart in Fig 13. |
| Number of Decision-Action with Waiting & Verification Steps | The number of steps characterized as "Decision-Action with Waiting Verification" steps according to the flowchart in Fig 13. |

The flowchart in Figure 16 is used to identify each SOP step type.

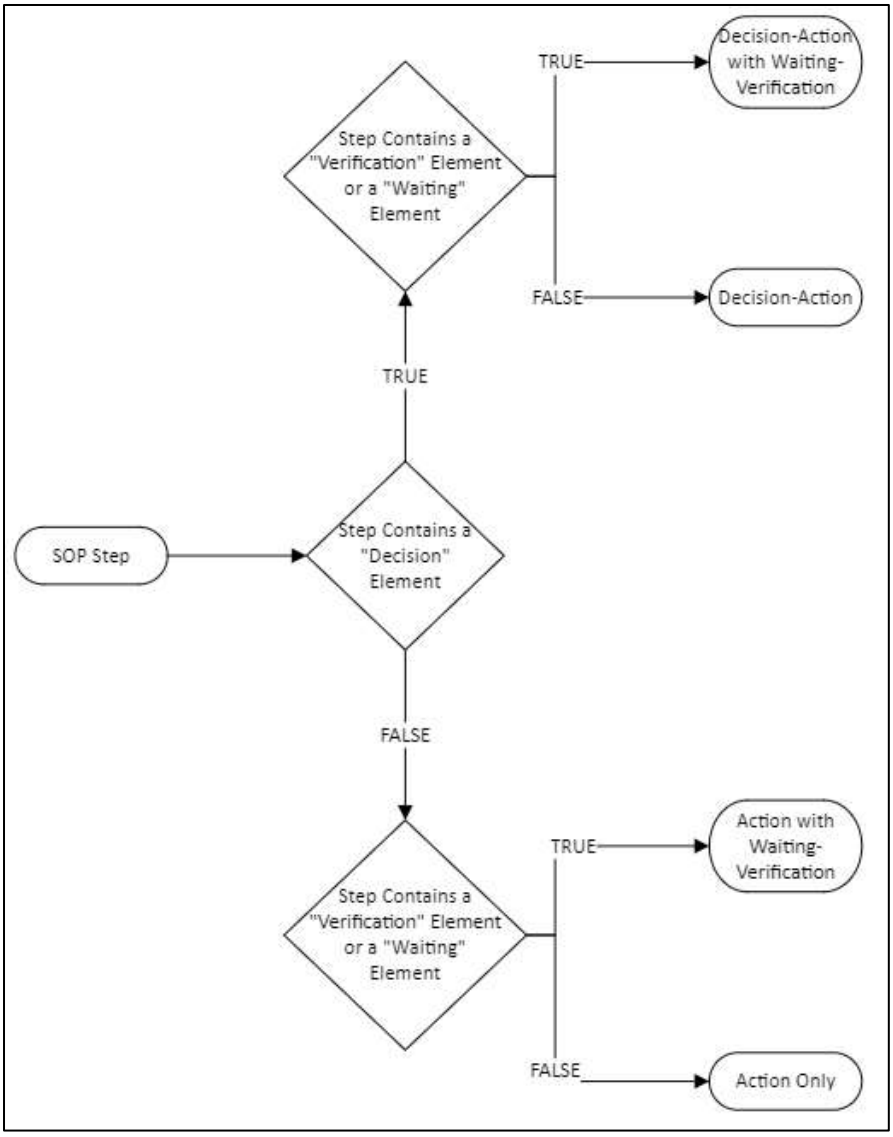



**Figure 16 Flowchart to Categorize SOP Steps to Types**

### *3.1.3.2 SOP Step Recall Score*

The SOP Step recall score is a metric that gauges the ratio of what the operator has to recall or remember when completing the step. This score is computed based on the presence of implicit/explicit e-PRL elements in the SOP step, depending on its type.

Explicit elements mean that the SOP explicitly states the element. An example of an SOP step with explicit elements is in Figure 17. Since the SOP step explicitly identified the physical action, then the “Action (How)” is an explicit element.

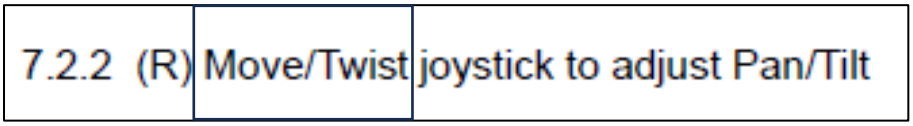



**Figure 17 Example of SOP Step**

**Implicit elements are the exact opposite of explicit elements, meaning that implicit elements are e-PRL elements that have been removed from the SOP step but are implied through the operator’s knowledge. There are three (3) types of implicit information: (1) implied – sequential, (2) implied – instantaneous, and (3) implied – system description.**

Table 4 summarizes the three types, their definitions, and the e-PRL elements they are associated with and provides an example for each case.

**Table 4 Implicit Elements Definitions and Examples**

| Implicit Element Type | Definition | e-PRL Element | Example | Notes |
|---|---|---|---|---|
| Implicit – Sequential | Information is absent from the SOP step due to the sequential nature of the steps. | Trigger (What) | 1. ® Select "Grab HC Focus"<br>2. ® Select "Auto" (turns blue) | The steps are done sequentially; therefore, the initiating conditions are implied. |
| Implicit – Instantaneous | Information is absent from the SOP step due to its instantaneous nature. | Waiting (What), Waiting (Where), Waiting (How) | ® Select "Auto" (turns blue) | After the operator selects "Auto," the operator must wait for the system to respond by turning the button blue. This response is instantaneous. |
| Implicit – System Description | Information absent from the SOP step due to the operator's system description knowledge of how to operate input devices and where to look for cues. | Action (How), Action (Where), Trigger (How), Trigger (Where), Verification (Where), Verification (How), Decide (Where), | ® Select "Grab HC Focus" | How the operator selects "Grab HC Focus," and the device used to make the selection is part of the operator's system description knowledge. |

| | | | | |
|---|---|---|---|---|
| | | Decide (How) | | |

It is crucial to capture the number of implied–system description elements since that information is memorized, and the operator must recall those elements when completing the SOP. Thus, the implied – --system description elements drive the SOP step recall score modeled by Equation 2.

**Equation 2 SOP Step Recall Score**

$$100 * (\frac{Number\ of\ Implied\ (System - Description)\ e - PRL\ Elements\ in\ an\ SOP\ Step}{Number\ of\ implied\ e - PRL\ Elements\ available\ in\ a\ step})$$

The number of implied e-PRL elements available in a step, the denominator in Equation 2, is shown in Table 5.

**Table 5 Number of e-PRL Elements Required by SOP Step Type**

| SOP Step Type | Number of e-PRL Elements Required |
|---|---|
| Action-Only | 4 |
| Decision-Action | 6 |
| Action with Waiting-Verification | 8 |
| Decision-Action with Waiting-Verification | 10 |

### *3.1.3.3 Estimating Training Requirements*

Matessa and Polson (2006) [60] provided a model for estimating the number of repetitions required for a list of length L. Table 6 presents the number of repetitions needed each day as a function of List Length (L) as described in [60]. It is important to

note that the first two rows in Table 6 were not part of the original analysis done by Matessa & Polson but were assumed to be true for the rest of this analysis. Furthermore, list lengths 16 and 24 (rows 7 and 8 in Table 6) show that the operator was never able to reach proficiency within the eight days initially analyzed. Therefore, an assumption was made that the operator was able to achieve proficiency on Day 9 for a list length of 16 items. No further assumptions were made for list length 24. Bolded and italicized are the assumptions made for this analysis that were not a part of the original analysis done by [60].

**Table 6 Average Number of Repetitions of a List Required to Reach Proficiency**

| List Length | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 | Day 8 | Day 9 | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| ***1*** | ***1*** | | | | | | | | | ***1*** |
| ***2*** | ***1.1*** | ***1*** | | | | | | | | ***2.1*** |
| 4 | 1.5 | 1.2 | 1.1 | 1 | | | | | | 4.7 |
| 8 | 4.5 | 2.4 | 1.8 | 1.5 | 1.4 | 1 | | | | 12.5 |
| 12 | 8.5 | 3.9 | 2.7 | 2.1 | 1.8 | 1.6 | 1 | | | 21.7 |
| 16 | 15.5 | 6.6 | 4.2 | 3.2 | 2.6 | 2.2 | 2 | 1.8 | ***1*** | ***39.2*** |
| 24 | 23 | 9.5 | 5.9 | 4.3 | 3.4 | 2.9 | 2 | 1.8 | ***Proficiency not reached.*** | ***52.8*** |

According to Matessa & Polson [60], training a procedure is analogous to memorizing a list of actions. To account for the list length not in the table, a regression analysis was conducted. There is a linear relationship between the number of repetitions required and the number of steps (based on the data in Table 6), with an R^2 value of 0.977. The relationship is defined by Equation 3, where Num Steps is the total number of steps in an SOP:

**Equation 3 Total Number of Repetitions for an SOP as a Function of Number of Steps**

$$Total\ Number\ of\ Repetitions\ (TNR)_{SOP} = 2.35(Num_Steps_{SOP}) - 3.24$$

Equation 3 estimates the total number of repetitions for the SOP as a function of the number of steps in the SOP. However, steps with a higher recall score would be treated as a 1+SOP Recall score. The updated number of steps can be modeled with Equation 4:

**Equation 4 Number of Steps in SOP with Recall Items**

$$Num\ Steps_{SOP} = \sum_{i=1}^{num\ steps} 1 + SOP\ Recall\ Score_i$$

Combing Equation 3 and Equation 4 yields the total number of repetitions (TNR) for an SOP in Equation 5:

**Equation 5 Total Number of Repetitions for a Perfect Recall**

$$Total\ Number\ of\ Repetitions\ (TNR)_{SOP} = 2.35\left(\sum_{i=1}^{num\ steps} 1 + SOP\ Recall\ Score_i\right) - 3.24$$

The Number of Days (ND) required to reach proficiency can be derived from Table 7. The Total Number of Repetitions (TNR) was taken directly from Table 6, column 11. The Days to Proficiency column was calculated by counting the number of non-empty cells in each row.

**Table 7 Number of Repetitions and Days to Proficiency**

| Total Number of Repetitions | Days to Proficiency |
|---|---|
| 1 | 1 |
| 2.1 | 2 |
| 4.7 | 4 |
| 12.5 | 6 |
| 21.7 | 7 |
| 39.2 | 9 |

To account for data not covered in the table, a regression analysis was conducted to understand the relationship between the two variables. The relationship between the two variables, a logarithmic relationship with an R^2 value of 0.99, is described by Equation 6:

**Equation 6 Number of Days Required to Reach Proficiency**

$$Number\ of\ Days\ (ND)_{SOP} = 2.16 \ln(TNR_{SOP}) + 0.68$$

### 3.1.4 Simulate SOP

The Computer-Aided Design of SOPS (CADSOPs) is an Excel-based user interface to aid in the process of creating, storing, and simulating SOPs (Figure 18).

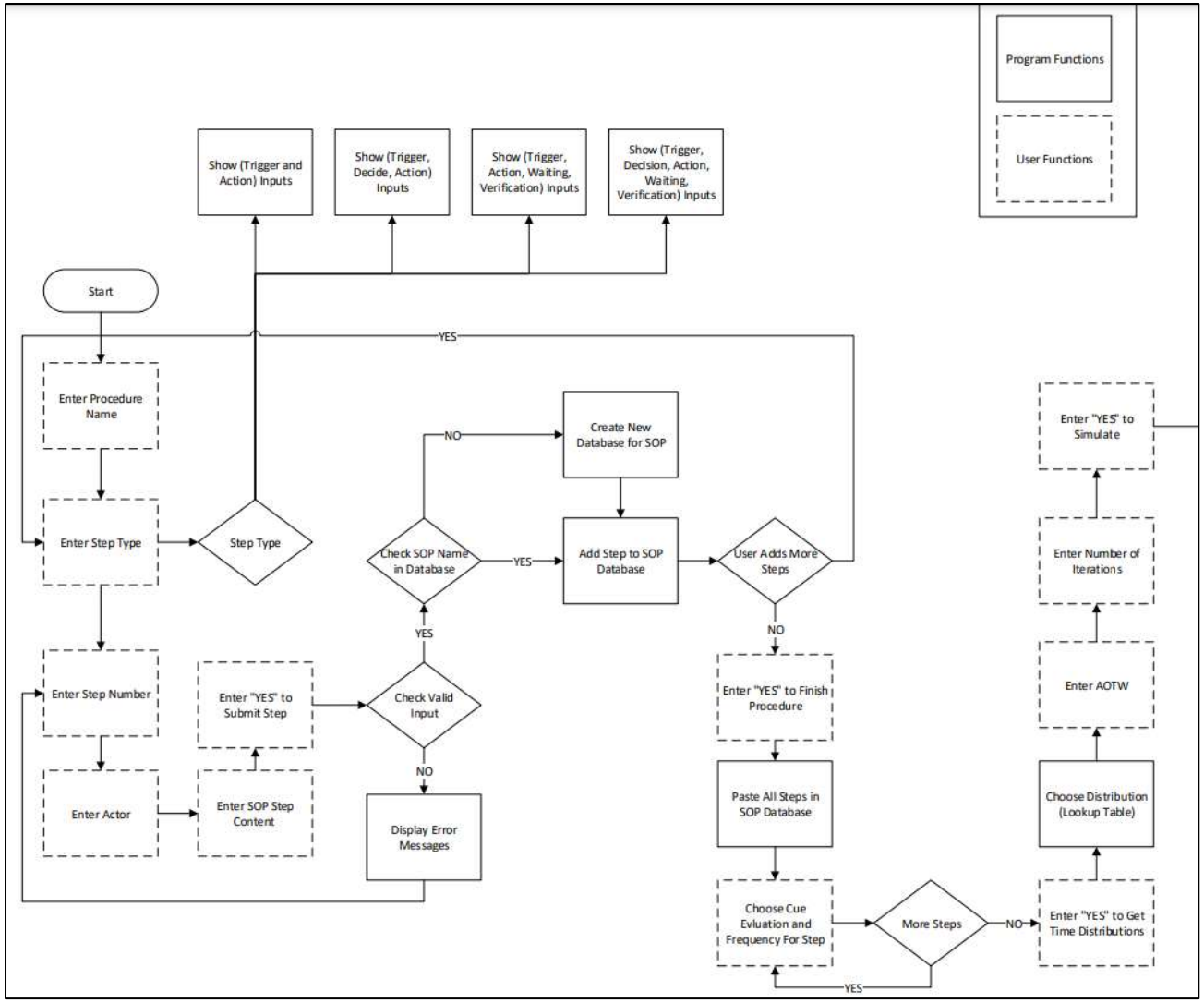



**Figure 18 Excel-Based Program (User Entry)**

The CADSOPs interface allows the user to input SOP steps one by one using the interface shown in Figure 19. Each SOP step input is followed by submitting a “YES” in the “Submit Step” cell, which then exports the entered step into a new sheet if the procedure name is non-existent in the current database or exports the step into an existing tab if the procedure name exists in the database.

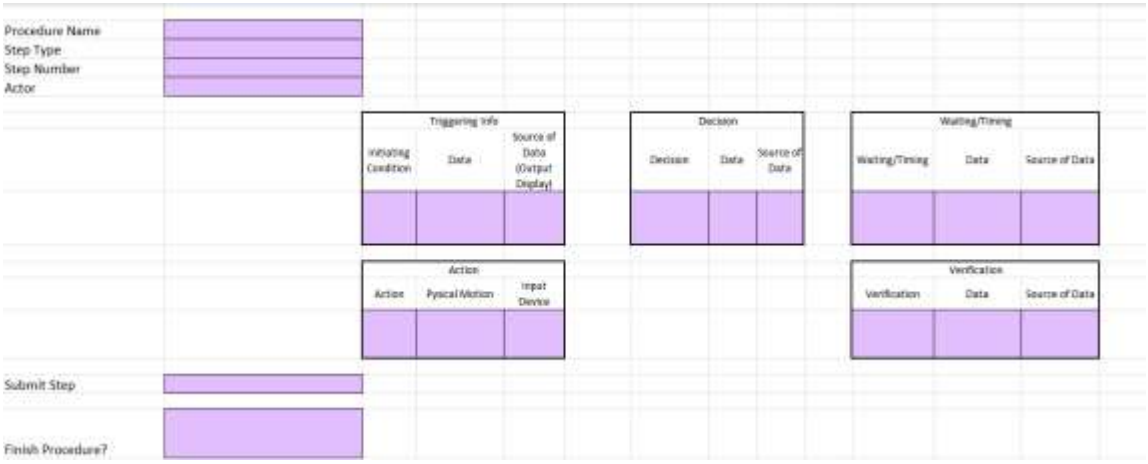



**Figure 19 CADSOPs Interface**

After submitting all of the SOP steps into the program, the user is then prompted to submit "YES" in the "Finish Procedure" cell. This drives the program to import all of the steps of that procedure into the current sheet to allow the user to assess the cue type and frequency, as shown in Figure 20.

**Figure 20 Importing Existing Steps into Interface**

To assess cue type and frequency, the user is prompted to select from the drop list, as shown in Figure 21 and Figure 22.

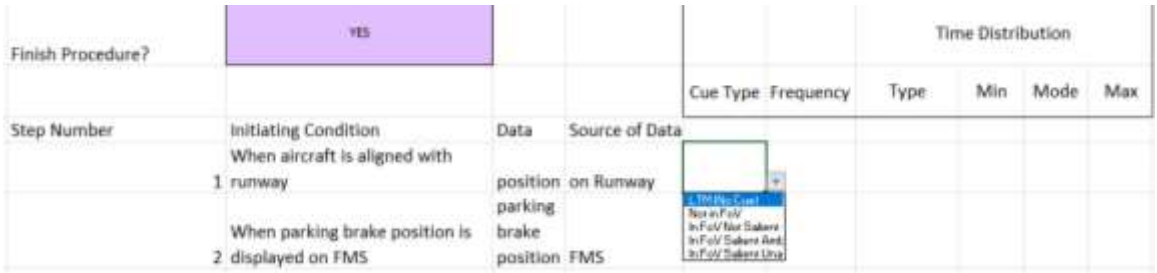


**Figure 21 Assess Cue Type**

**Figure 22 Assess Frequency**

Based on the user-specified cue type and frequencies, the user can then import the appropriate time distribution by submitting “YES” under the “Get Distributions” cell shown in Figure 23.

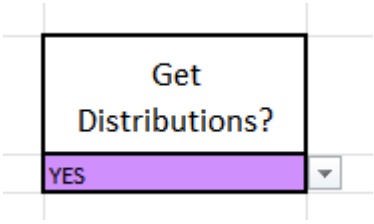



**Figure 23 Cell to Import Time Distributions**

The resulting time distribution is printed in the same input sheet to be verified by the user (see Figure 24).

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Finish Procedure? | YES | | | | | Time Distribution | | | |
| | | | | | Cue Type | Frequency | Type | Min | Mode | Max |
| | Step Number | Initiating Condition | Data | Source of Data | | | | | | |
| 1 | | When aircraft is aligned with runway | position | on Runway | Not In Fo\ | Infrequent | Triang | 0.03 | 43.7 | 109.1 |

**Figure 24 Verify Imported Time Distributions**

Finally, the SOP can be simulated based on the user-specified AOTW and number of iterations. The flowchart in Figure 25 depicts the overall process for running a Monte-Carlo simulation to generate the ToP and AOTW of an SOP. The process begins by initializing required variables (ToP & AOTW arrays, number of iterations, procedure

steps) and any counters (j and p). A nested loop iterates through each procedure step (j) and each Monte-Carlo iteration (p). For each iteration (p), a random variable U is generated using Excel's RND() function for the trigger and action of each step. These random variables are transformed using the inverse transformation method.

The inverse transformation method works by generating a random variable x, where $0 \leq x \leq 1$, then setting that random variable equal to the triangular cumulative distribution function, and solving for x. Given that the cumulative distribution function F(x) of a triangular distribution (Equation 7):

**Equation 7 Triangular CDF**

$$F(x) = \begin{cases} 0 & if\ x \leq a \\ \dfrac{(x-a)^2}{(b-a)(c-a)} & if\ a < x \leq b \\ 1 - \dfrac{(c-x)^2}{(c-a)(c-b)} & if\ b < x < c \\ 1 & if\ c \leq x \end{cases}$$

The distribution has two triangles, the left one extending from x=a to x=b and the right one extending from x=b to x=c, with a common side H. The combined areas of the two triangles must be equal to one. Thus, Equation 8 must hold:

**Equation 8 Combined Area under a Triangular Distribution**

$$\frac{1}{2}(b-a)(H) + \frac{1}{2}(c-b)(H) = 1$$

Applying geometry to Equation 8 to isolate H yields Equation 9.

**Equation 9 Closed Form Solution of Common Side H**

$$H = \frac{2}{c-a}$$

To transform the random variable U, it is necessary to determine first if the random variable is from the left or right triangle, i.e., left or right of the mode. This is

done by computing the ratio of the mode minus the minimum to the maximum minus the minimum (Equation 10). If U is less than the ratio, then it is known that $a < x < b$, and x falls into the left triangle.

**Equation 10 Ratio to Determine Random Variable Resides in Which Triangle**

$$\frac{b-a}{c-a}$$

When the random variable x falls in the left triangle, then the distance from a to x must be determined. The new area, which equals the random variable U, can be donated by Equation 11, where h is the new height.

**Equation 11 Area of Triangle bound by x**

$$\frac{1}{2}(x-a)(h) = U$$

Solving for h yields Equation 12:

**Equation 12 Closed Form Solution of h**

$$h = -\frac{2U}{a-x}$$

By triangle proportion properties, the proportion of the height H to the distance between b and a is equal to the ratio of the height h to the distance between x and a. Therefore, Equation 13 must hold.

**Equation 13 Proportional Property of Triangles**

$$\frac{H}{b-a} = \frac{h}{x-a}$$

Solving for x in Equation 13 yields Equation 14.

**Equation 14 Closed Form Solution of x**

$$x = a + \frac{h(b-a)}{H}$$

Substituting Equation 12 and Equation 9 in Equation 14 yields Equation 15:

**Equation 15 Closed Form Solution of x When x Lies in Left Triangle**

$$x = a + \sqrt{U(c-a)(b-a)}$$

Applying the same logic, the transformation of a triangular random variable can be written as the inverse of the cumulative distribution, given by Equation 16.

**Equation 16 Inverse Cumulative Distribution Function (CDF)**

$$F^{-1}(U) = \begin{cases} a + \sqrt{U(c-a)(b-a)} & if\ U < \dfrac{b-a}{c-a} \\ c - \sqrt{(1-U)(c-b)(c-a)} & if\ U \geq \dfrac{c-b}{c-a} \end{cases}$$

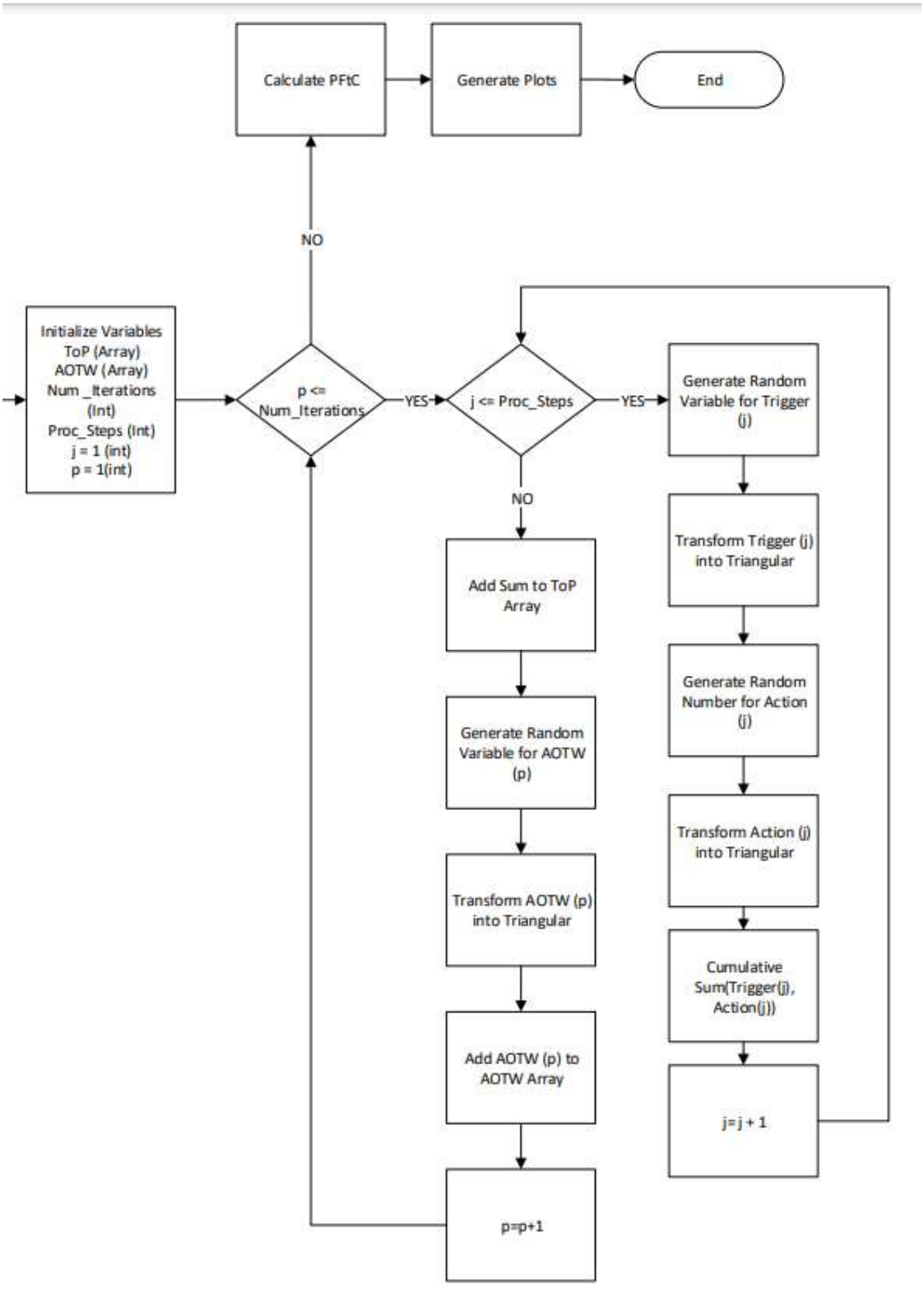



**Figure 25 Flowchart for SOP Simulation**

The simulation analysis of SOPs is captured through multiple iterations of the Monte-Carlo simulation. The performance of the SOP is measured by the Time on Procedure (ToP). The ToP differs each time the SOP is completed due to the varying level of expertise of the operators performing the SOP, attention, fatigue, and other human factors. The Allowable Operation Time Window (AOTW) is defined by the time in which the procedure must be completed [21]. The AOTW can also differ due to variability in aircraft performance and the operational environment, which explains why it is given as a distribution and not a fixed number.

When the ToP exceeds the AOTW, it indicates that the SOP failed to meet its goal. From the ToP and AOTW, the Procedure Buffer Time (PBT) can be derived by the following relation: PBT is the difference in AOTW and ToP for each time the SOP is performed. Because distributions govern AOTW and ToP, PBT is represented as a distribution, where the left tail below zero (i.e., ToP> AOTW) is defined as the Probability of Failure to Complete (PFtC). The longer the left tail is, the more unreliable the SOP is. Figure 26 shows the dynamic metrics of the SOP as the ToP distribution (in red), the AOTW distribution (in blue), the PBT distribution (in purple), and the PFtC (in red).

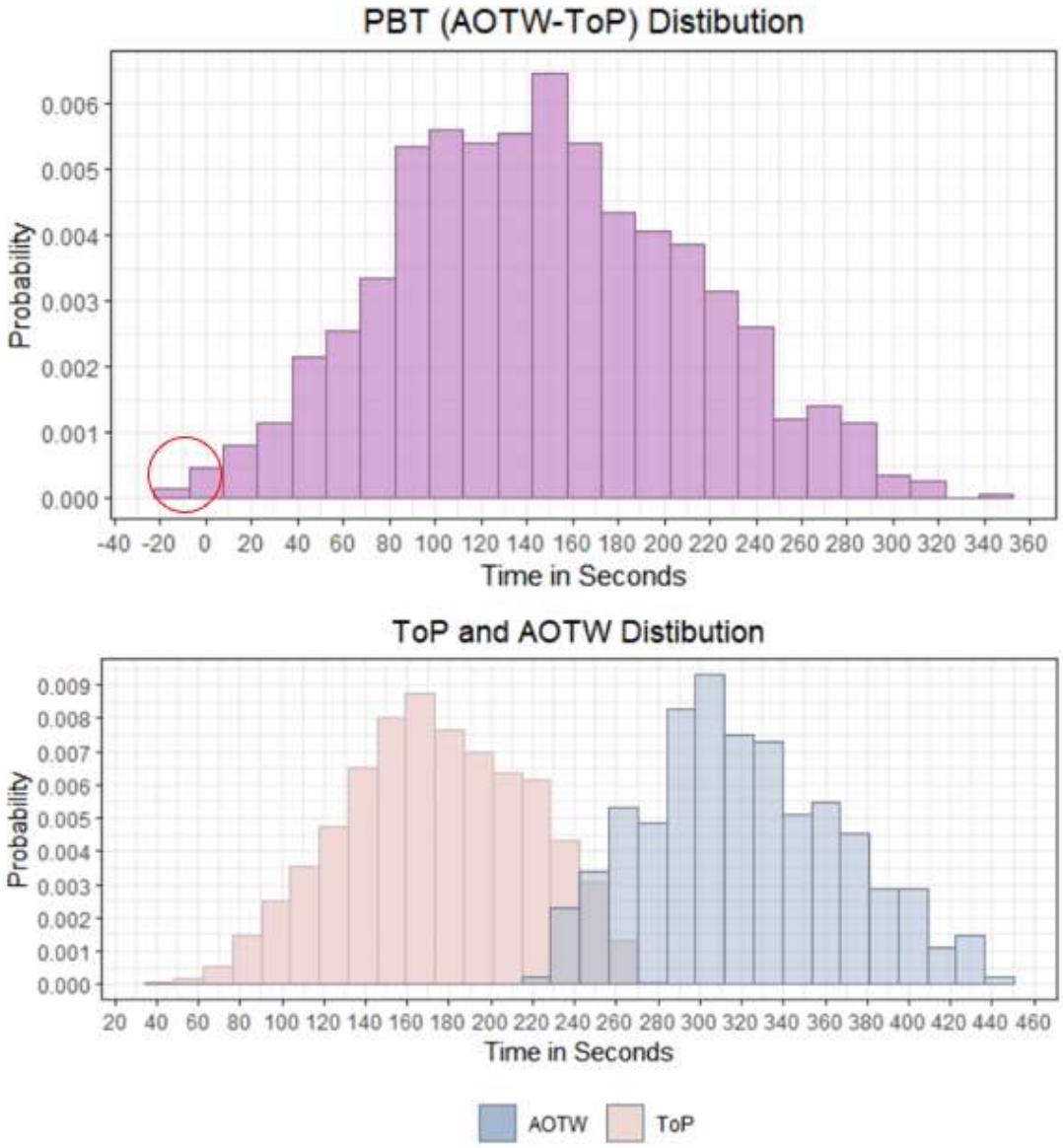



**Figure 26 PBT, ToP, AOTW, and PFtC**

### *3.1.4.1 Human Performance Time Distribution Database*

To derive time distributions for the simulation model, a study that aimed to collect and analyze human performance data was conducted. The motivation behind the study

was based on the assumption that time distributions for operators operating complex systems (aircraft, autonomous vehicles, etc.) may be similar to users completing tasks in Microsoft Office applications. Therefore, this study collected timing data for common MS Office tasks completed by multiple users to derive time distributions that could be used in SOP modeling.

Twenty-one (21) participants, ages ranging from 18 to 65 years. Three participants were graduate students in an engineering discipline, two students were undergraduate students in an engineering discipline, twelve participants were senior high school students, and 4 participants were instructors (either college or high school). All participants reported using MS Office applications daily for schoolwork. None of the participants ever received any certification.

Participants completed twelve MS Office tasks in Word, PowerPoint, and Excel. There were three daily tasks, three frequent tasks (done about once or twice a week), three infrequent tasks (done about once or twice a month), and three rare tasks (done about once or twice a year). The tasks involved predefined initial conditions and instructions. The order of tasks presented was randomized for each participant to avoid any bias. Each step was assigned a "cue type," categorizing the type of cue available to the user for completing that step. Timing data was captured through mouse clicks using software tools such as Steps Recorder and XNote.

The cue types included (1) not in Field of View (FoV), in Field of View (FoV) but not salient, (3) in the Field of View (FoV), salient but ambiguous, (4) in the Feld of View (FoV), salient, and unambiguous, and (5) no cue (long term memory). Tasks had

between 3-5 steps each, with some alternate ways of completing tasks. The final dataset was analyzed to obtain timing distributions by cue type and frequency. A list of the tasks is provided in Appendix B.

Key characteristics of the time distributions were derived, including median, mean, fifth and ninety-fifth percentiles, and skew. Outliers were removed using the IQR method. The data for Always and Frequent frequencies were grouped, and the Infrequent and Rare datasets were grouped because of the small sample size. Table 8 provides the characteristics of the data obtained in seconds.

**Table 8 Characteristics of Data Obtained (in seconds)**

| Frequency | Type of Cue | 5th %ile | Median | Mean | Mode | 95th %ile | StDev | Skew |
|---|---|---|---|---|---|---|---|---|
| Always & Frequent | No Cue | 0.91 | 6.25 | 7.12 | 1.05 | 16.95 | 5.24 | 0.90 |
| | Not in FoV | 7.12 | 18.48 | 24.05 | NA | 50.65 | 16.84 | 1.55 |
| | In FoV, Not Salient | 1.12 | 2.65 | 6.96 | NA | 29.79 | 10.13 | 2.40 |
| | In FoV, Salient, Ambiguous | 1.05 | 2.89 | 11.04 | NA | 40.92 | 17.64 | 2.61 |
| | In FoV, Salient, Unambiguous | 1.37 | 2.27 | 4.78 | 2.13 | 12.31 | 4.61 | 1.76 |
| Infrequent & Rare | No Cue | 1.35 | 10.59 | 20.50 | 14.82 | 72.32 | 23.52 | 1.72 |
| | Not in FoV | NA | NA | NA | NA | NA | NA | NA |
| | In FoV, Not Salient | 1 | 10.40 | 25.37 | 1 | 83.72 | 28.53 | 1.13 |
| | In FoV, Salient, Ambiguous | 1.81 | 11.34 | 24.31 | NA | 76.71 | 26.78 | 1.25 |
| | In FoV, Salient, Unambiguous | 1.23 | 2.58 | 3.92 | NA | 11.87 | 3.29 | 1.65 |

Triangular distributions were fitted to the frequent task data. Table 9 provides the parameters for each distribution. A two-sided Kolmogorov-Smirnov (KS) test was conducted to test the null hypothesis that the sample was drawn from a triangular distribution with the parameters specified in Table 9. In summary, triangular distributions with the parameters in Table 9 were not a good representation of the data.

**Table 9 Parameters for Triangular Distributions and Goodness of Fit**

| Frequency | Type of Cue | Distribution | P-value | |
|---|---|---|---|---|
| Always & Frequent | No Cue | Triang(0.4, 1.05,25.28) | 0.004 | Reject Null Hypothesis |
| | Not in FoV | Triang(5.94, 24,1, 65.47) | 0.04 | Reject Null Hypothesis |
| | In FoV, Not Salient | Triang(0.94, 1.75, 48.05) | 2.2e-19 | Reject Null Hypothesis |
| | In FoV, Salient, Ambiguous | Triang(0.88,13.47, 80.26) | 3.8e-15 | Reject Null Hypothesis |
| | In FoV, Salient, Unambiguous | Triang(1.22, 2.13, 22.07) | 1.6e-10 | Reject Null Hypothesis |
| Infrequent & Rare | No Cue | Triang(0.44, 14.82, 103.84) | 2.2e-16 | Reject Null Hypothesis |
| | Not in FoV | NA | NA | NA |
| | In FoV, Not Salient | Triang(0.91, 1, 111.14) | 3.1e-15 | Reject Null Hypothesis |
| | In FoV, Salient, Ambiguous | Triang(1, 1, 110) | 3.9e-11 | Reject Null Hypothesis |
| | In FoV, Salient, Unambiguous | Triang(1, 6, 14.25) | 2.2e-16 | Reject Null Hypothesis |

Several parameters were found to lack sufficient data, including tasks "not in the Field of View," which reflected a limitation of the MS Office task interface. Additional limitations were brought on by the task design and low subject numbers.

### 3.1.5 Generate MBSE Action Diagram

An SOP Action Diagram consists of (i) actions, (ii) branches, and (iii) inputs/outputs (I/Os). Actors perform actions depicted by squares. The branches represent actors, and each actor performs the actions on its branch. An action can be triggered by a preceding action done by the same actor, or it may be triggered by an action from another actor, using I/Os to depict the information flow between the two actions. I/Os are defined as parallelograms.

This section provides step-by-step instructions to create an SOP Action Diagram and a method for converting existing e-PRL SOPs into MBSE Action Diagrams.

#### *3.1.5.1 Create MBSE Action Diagram*

Step 1: Identify and create branches for the Action Diagram (AD). Seven actors perform actions in an SOP Action Diagram. The actors are:

1. Environment
2. Plant
3. Automation Input Device
4. Automation Output Display
5. Operator(s)
6. Operator(s) Long Term Memory (LTM)
7. Operator(s) Working Memory (WM)

After identifying the actors, they are added to the diagram as branches. There must be a branch for each actor, and following the order above is preferred to minimize clutter in the action diagram produced.

Step 2: For each operator, insert operator actions. Operator actions are next identified directly from the SOP. The actions are added to the diagram by adding a square on the appropriate branch. Each action in the action diagram should be preceded by the word “Action:” followed by the action from the SOP.

Step 3: For each operator action, add a “See and Decide,” “Hear and Decide,” or “Remember and Decide” action to the action diagram. Each action must be preceded by a “See and Decide,” “Hear and Decide,” or “Remember and Decide” action to act as the decision-making action. Decision-making actions are actions where the actor must act after observing a trigger to the action. These decision-making actions are added to aid in the process of assessing action performance discussed in step 15.

Step 4: Add the initiation action and cue of the SOP. Each SOP is initiated by an I/O that triggers the first See/Hear/Remember and Decide action on the operator’s branch. The trigger, also known as a cue, may come from (i) the Automation Output Display, (ii) the Environment, (iii) the operator’s LTM, or (iv) the operator’s WM. The initiating cue can be either visual, aural or from memory.

Step 5: Insert Automation Output Display and Automation Input Device actions. This step is initiated by extracting all nonoperator actions and sorting them into Automation Output Device and Automation Input Display actions. Automation input and output device actions could be extracted by analyzing operator actions. For example, if an operator's action was to raise the landing gear lever, then the modeler may expect to see an Automation Input Device action to do so. The same operator action can be used to find

the Automation Output Display action, as an action to display landing gear raised confirmation is expected. The identified actions are then added to the diagram.

Step 6: Identify the cues of the SOP and identify the type of each cue: visual, aural, and memory. First, identify the cues present in the SOP. Cues may be easily extracted from the SOP, or they may be implied by looking at operator actions. For example, if an operator's action was to call out VOne speed, then there should be an aural cue as a result. After identifying as many cues as possible, the next step is sorting each cue according to the type, either visual, auditory, or memory. Visual cues are cues that can be seen, typically through the Automation Output Display or the Environment. Aural cues are those that can be heard, and they are usually produced through the operator or the Automation Output Display. Memory cues are those generated by the Operator's long-term memory or working memory.

Step 7: Identify the source action/actor of each cue identified in Step 6. For each cue identified in step 6, identify the source action that generates the cue. The originating actor is also identified in this step. The Environment and the Automation Output Display typically generate visual and aural cues. The WM or LTM generates memory cues.

Step 8: For each "See/Hear and Decide" or "Remember and Decide" action, add an appropriate cue as an input to the "See/Hear and Decide" action from the list identified in Step 6. Using the cues identified in step 6, add the cues to the diagram, connecting them to appropriate See/Hear/Remember and Decide actions. During the first iteration, not all cues may be connected to Operator actions. However, eventually, all See/Hear/Remember and Decide actions will be triggered by a cue.

Step 9: For the list of actions identified in Step 7, add the actions to the diagram, connecting the actions to appropriate cues as determined in Step 6. Add each action identified in Step 7 to the diagram, making sure to connect each action to the appropriate cue.

Step 10: Re-write the cues to follow the "Cue Wording Rules." For all I/Os that are connected to an input and output, rewrite the I/Os based on what actor originated them and what actor they are going to.

1. Outputs from the Plant to the Automation Output Display: nouns, e.g., climb rate, lights. Sound.
2. Outputs from the Automation Output Display to the Operator(s): nouns, e.g. altitude, speed.
3. Outputs from the Automation Input Device to the Plant: verb-noun, e.g., extend flaps or raise landing gear.
4. Outputs from the Operator(s) to the Automation Input Display: verb-noun, e.g., move flap handle, raise landing gear lever.
5. Outputs from an Operator to another Operator: A word or phrase coming from an operator. E.g., "V-One".
6. Outputs from the LTM – WM to an Operator: Verb sentence, E.g., Recall check for V-One speed.

Step 11: For each operator action, add an Automation Input Device action and I/O as appropriate. Connect any Operator physical actions to the actions on the Automation

Input Device branch on the diagram, making sure to add I/Os between connecting actions following the rules in step 10.

Step 12: For each Automation Input Device action, add an output to a Machine Action and add I/Os as appropriate. For each Automation Input Device action, add a Plant action as a result, making sure to add I/Os following the rules in step 10.

Step 13: For each Machine Action, add an output to an Automation Output Display Action and I/O as appropriate. Connect Plant actions to Automation Output Displays as appropriate, again making sure to add I/Os that follow the wording rules specified in step 10.

Step 14: Compare SOP to the action diagram, and go back to step 5 as appropriate. This concludes the first iteration of creating an SOP Action Diagram. Look through the SOP and compare it to the generated Action Diagram. Go back and redo the steps as appropriate.

#### *3.1.5.2 Convert Existing e-PRL SOP into MBSE Action Diagrams*

If a procedure is in e-PRL format, then the SOP can be converted into an MBSE Action diagram with the help of Table 10. There is a one-to-one mapping between e-PRL elements and the SysML Action diagram. That means that once an SOP is in e-PRL form, the SOP can be converted into an Action Diagram automatically.

**Table 10 Mapping Between e-PRL and SOP Action Diagram**

| e-PRL Element | SOP Action Diagram Element |
|---|---|
| Procedure Name | Diagram Name |
| Procedure Initiation Condition | Action on Environment Branch<br>It could be from the plant, output device, or LTM |
| Procedure End Goal | I/O from Procedure Initiation Condition to first operator action |
| Actor | Branch Asset |
| Decide (What) | Action on Actor Branch |
| Decide (Where) | Action on Output Display Branch |
| Decide (How) | I/O from Decide (Where) to Decide (What) |
| Action (What) | See & Decide: Action on Actor Branch |
| Action (Where) | Action on Input Device Branch |
| Action (How) | I/O from Action (What) to Action (Where) |
| Waiting (What) | Action on Actor Branch |
| Waiting (Where) | Action on Output Display Branch |
| Waiting (How) | I/O from Waiting (Where) to Waiting (What) |
| Verification (What) | Action on Actor Branch |
| Verification (Where) | Action on Output Display Branch |
| Verification (How) | I/O from Verification (Where) to Verification (What) |

### 3.1.6 Summary of Methodology for Analysis and Simulation of SOP Steps

This chapter presented a methodology to enable quantitative analysis and simulation of SOPs. Significant contributions are achieved through the digitization of the SOP steps into a machine-readable format. This is done through a refined Large Language Model that can parse SOP text into a canonical structure. Digitized SOPs can be visualized, analyzed using quantitative metrics, and simulated to assess human performance and its variability.

This methodology is applied to three case studies in chapter four: (1) Aviation Operational SOPs, (2) International Space Station (ISS) Habitable Airlock (HAL) SOPs, and (3) Autonomous Vehicle SOPs.

# CHAPTER FOUR: CASE STUDIES

To validate the methodology and the model of SOP steps developed, the method was applied to three case studies in three different domains: (1) aviation operational SOPs, (2) International Space Station (ISS) Habitable Airlock (HAL) SOPs, and (3) autonomous vehicle SOPs. The three cases differ in the training and certification that the operator undergoes. Domains such as aviation or ISS are selective in choosing operators and require the operator to be trained and certified before operations. On the other hand, autonomous vehicle operators are loosely trained and licensed to operate standard, i.e., non-autonomous vehicles. Table 11 summarizes the differences between the three case studies.

**Table 11 Training, Certification, and Selectiveness of Three Case Studies**

| Case Study | Operator Training | Operator Certification | Domain Selectiveness of Operators |
|---|---|---|---|
| Aviation Operational SOPs | Operators trained on airlines' operational SOPs. Recurrent training every 9-12 months. | Certified and licensed by FAA. | Selective |
| ISS HAL | Operators trained for approximately 300 hours in simulators. | Certified by NASA. | Extremely Selective |
| Semi-Autonomous Vehicles | Operators are trained in driver's education courses operating non-autonomous vehicles. | Licensed by DMV. | Not Selective |

For each of the three case studies, SOPs were converted to the canonical structure (e-PRL) either through the established manual or automatic processes. Metrics such as step type, step recall score, and training requirements were computed for the set of SOPs. Next, the structured SOPs were simulated to assess variable human performance (ToP) against operational time constraints (AOTW). Finally, SysML diagrams were produced for a sample of the SOPs to demonstrate the consistency of the developed SOP model with Model-Based Systems Engineering (MBSE) approaches.

## 4.1 Case Study One: Airline Operations SOPs

In aviation, SOPs represented in the Quick Reference Handbooks (QRH) and the Airplane Flight Manual (AFM) or Flight Crew Operating Manual (FCOM) provide step-by-step guidance to identify and mitigate unsafe conditions, as well as operate the airliner in normal conditions. In the past, accidents have occurred where emergency procedure execution times exceeded the amount of time that the crew had. A total of fifteen (15) SOPs were analyzed in this case study.

### 4.1.1 Objective

Demonstrate the application of the SOP analysis methodology for a sample of fifteen procedures in a safety-critical domain with selective operators. All SOPs were normal operation SOPs, including two emergency SOPs.

### 4.1.2 Data and Preprocessing

Fifteen (15) SOPs were obtained from the A318/A319/A320/A321 FCOM and the Boeing 747-441 Operations Manual. All SOPs were normal SOPs, i.e., done regularly

and for non-emergency use, except for two SOPs, which are emergency SOPs. The 15 SOPs had a total of 171 SOP steps.

### 4.1.3 Results

The following subsections explain the results obtained.

#### *4.1.3.1 SOP Step Type*

Table 12 summarizes the categorization of SOP steps. While the majority of the SOP steps were action-only steps (70%), some steps had a decision (14%), waiting & verification (12%), and a decision waiting & verification (4%). Steps that have a waiting requirement should be followed by a verification requirement to ensure that the machine reaches its expected state and to avoid race conditions. i.e., ensure that the operators wait the required amount of time before executing the next step.

**Table 12 SOP Step Types – Case 1**

| SOP Step Type | Percent of SOP Steps |
|---|---|
| Action-Only | 70% |
| Decision-Action | 14% |
| Action with Waiting-Verification | 12% |
| Decision-Action with Waiting – Verification | 4% |

Table 13 presents the percentage present of each e-PRL label in the SOP step across all four SOP step types. The action-only steps, which specify only a trigger and an

action, frequently state what the operator should be looking for (21%) but rarely say where the operator has to look to perceive the trigger (3%). The triggering condition was stated 55% of the time due to the sequential nature of the SOP steps, i.e., whenever it is missing, it is because the SOP step is triggered by the end of the previous step.

The decision-action steps, which require a trigger, a decision, and an action, also displayed the consecutiveness of SOP steps by explicitly stating 83% of the triggering conditions (17% of the steps were sequential and did not require a triggering condition). These steps specified the cue that the operator looks at 83% of the time and what output display the cue would be presented on 67% of the time.

It is crucial to analyze steps with any waiting and verification (action with waiting & verification and decision-action with waiting & verification). Across both step types, 70% (19 out of 27) of the steps included a waiting requirement but did not follow with a verification requirement. Many of these steps included a verification without stating a waiting requirement. This means that the waiting in this case was instantaneous and therefore omitted from the SOP.

.

**Table 13 Content Explicitly Present in the SOP Steps – Case 1**

| SOP Element | Action-Only | Decision-Action | Action with Waiting & Verification | Decision-Action with Waiting & Verification |
|---|---|---|---|---|
| Percentage | 70% (120 out of 171) | 14% (24 out of 171) | 12% (21 out of 171) | 4% (6 out of 171) |
| Actor | 10.00% (12 Out of 120) | 79% (12 Out of 24) | 10% (2 Out of 21) | 33% (2 Out of 6) |
| Trigger | | | | |
| What | 55% (66 Out of 120) | 83% (20 Out of 24) | 57% (12 Out of 21) | 83% (5 Out of 6) |
| Where | 3% (3 Out of 120) | 67% (16 Out of 24) | 5% (1 Out of 21) | 17% (1 Out of 6) |
| How | 21% (25 Out of 120) | 83% (20 Out of 24) | 48% (10 Out of 21) | 67% (4 Out of 6) |
| Decide | | | | |
| What | | 100% (24 Out of 24) | | 83% (5 Out of 6) |
| Where | | 0% (0 Out of 24) | | 17% (1 Out of 6) |
| How | | 0% (0 Out of 24) | | 67% (4 Out of 6) |
| Action | | | | |
| What | 100% (120 Out of 120) | 100% (24 Out of 24) | 100% (21 Out of 21) | 100% (6 Out of 6) |
| Where | 81% (97 Out of 120) | 100% (24 Out of 24) | 90% (19 Out of 21) | 100% (6 Out of 6) |
| How | 43% (51 Out of 120) | 92% (22 Out of 24) | 62% (13 Out of 21) | 100% (6 Out of 6) |
| Waiting | | | | |

| | | | | |
|---|---|---|---|---|
| What | | | 29%<br>(6 Out of 21) | 50%<br>(3 Out of 6) |
| Where | | | 14%<br>(3 Out of 21) | 0%<br>(0 Out of 6) |
| How | | | 14%<br>(3 Out of 21) | 17%<br>(1 Out of 6) |
| Verification | | | | |
| What | | | 76%<br>(16 Out of 21) | 50%<br>(3 Out of 6) |
| Where | | | 71%<br>(15 Out of 21) | 50%<br>(3 Out of 6) |
| How | | | 71%<br>(15 Out of 21) | 50%<br>(3 Out of 6) |

#### *4.1.3.2 SOP Step Recall Score*

Table 14 displays the average SOP recall score for each SOP. The SOP step recall score ranges from 25%-81%, with the majority of the SOPs having an average SOP recall score of 30%-60%. This suggests that these procedures generally require a moderate amount of information recall. Figure 27 displays the SOP step recall for each SOP, along with the SOP step type. SOPs with a high average SOP step recall score, such as "SA," had four steps with a recall score of 100% and nine steps with a recall score of 75%.

**Table 14 Average SOP Step Recall Score by SOP - Case 1**

| SOP | Average SOP Step Recall Score |
|---|---|
| AL_747 | 50% |
| AS_747 | 52% |
| FLCCD | 46% |
| AFDS | 25% |
| HH | 44% |
| HS | 46% |
| AH | 25% |
| VSCD | 34% |
| AO | 35% |
| LA | 58% |
| SA | 81% |
| SF | 71% |
| TO_320 | 62% |
| TO_747 | 43% |
| SP | 38% |

**Figure 27 SOP Step Recall Score by SOP - Case 1**

### *4.1.3.3 Estimating Training Requirements*

The total number of repetitions required for each SOP, along with the number of days to proficiency, are summarized in Table 15. The number of steps column is the number of steps in the SOP. The results show extreme variability across the set of SOPs in the number of steps in each SOP, ranging from just one step to SOPs with 55 steps that need to be memorized for training purposes.

**Table 15 Training Estimates – Case 1**

| SOP | Num Steps | Average SOP Recall Score | Total Number of Repetitions (TNR) | Number of Days (ND) |
|---|---|---|---|---|
| AL_747 | 8 | 50% | 26 | 7.6 |
| AS_747 | 8 | 52% | 26 | 7.6 |
| FLCCD | 3 | 46% | 8 | 4.9 |
| AFDS | 2 | 25% | 3 | 2.8 |
| HH | 2 | 44% | 4 | 3.4 |
| HS | 3 | 46% | 8 | 4.9 |
| AH | 1 | 25% | 3 | 2.4 |
| VSCD | 4 | 34% | 10 | 5.5 |
| AO | 8 | 35% | 23 | 7.4 |
| LA | 16 | 58% | 57 | 9.4 |
| SA | 16 | 81% | 66 | 9.7 |
| SF | 15 | 71% | 57 | 9.4 |
| TO_320 | 55 | 62% | 206 | 12.2 |
| TO_747 | 28 | 43% | 91 | 10.4 |
| SP | 2 | 38% | 4 | 3.2 |

#### *4.1.3.4 Simulations of SOPs*

A segment of the Takeoff SOP was simulated. The Allowable Operational Time Window (AOTW) for the Takeoff Roll segment was obtained [20]. The Probability of Failure to Complete (PFtC) was 5.72%.

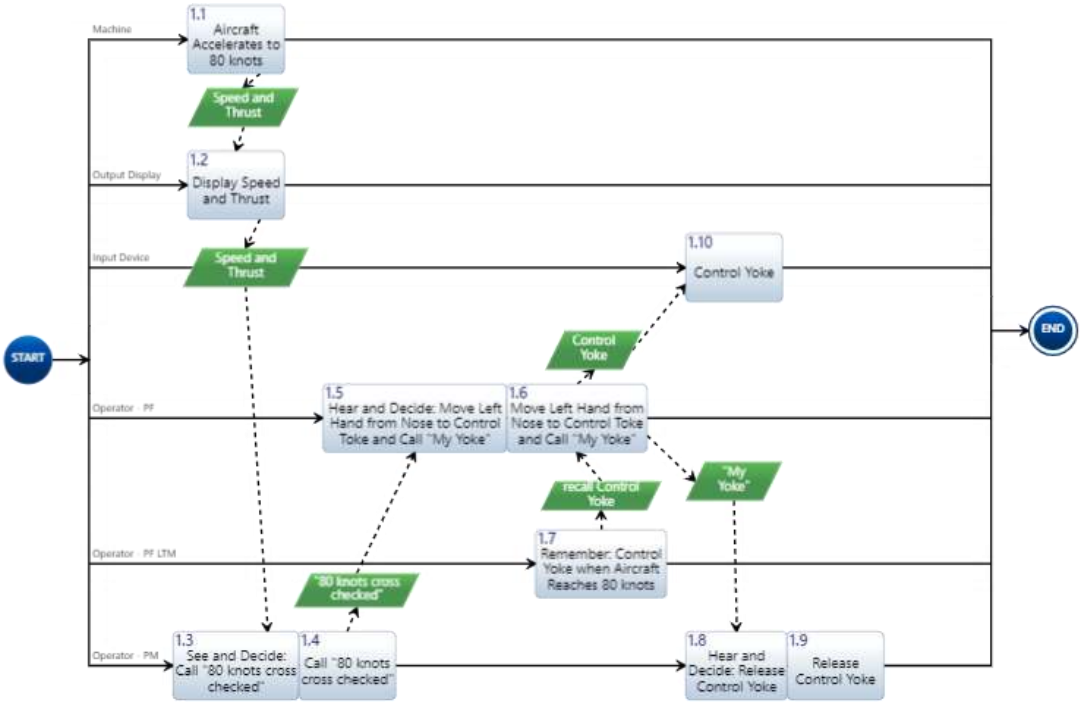



**Figure 28 MBSE Visualization - Case 1**

The aural call-out triggering the two steps (1.5 & 1.8) in Figure 28 was present in the operator's auditory perception and not lost in the noise. However, one cue has ambiguous semantics (i.e., the PM calls 80 knots, and the PF must control the yoke). The visual cue triggering the first step (1.3) was present in the operator's FoV, not lost in the clutter and semantically unambiguous. Figure 29 and Figure 30 display the simulation metrics for the takeoff roll segment.

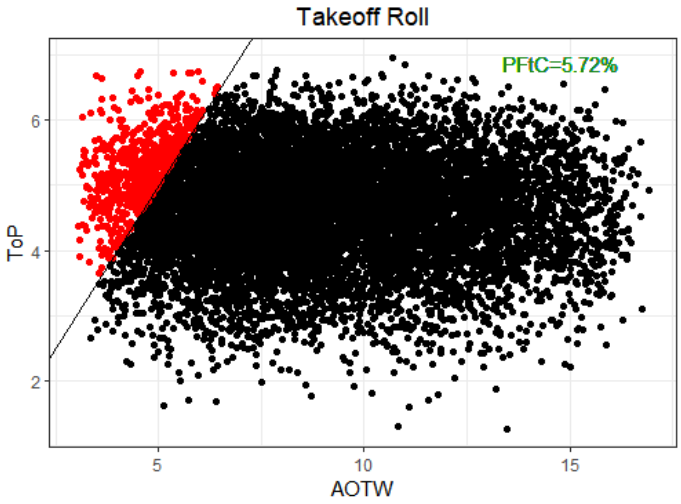



**Figure 29 ToP & AOTW Scatter Plot**

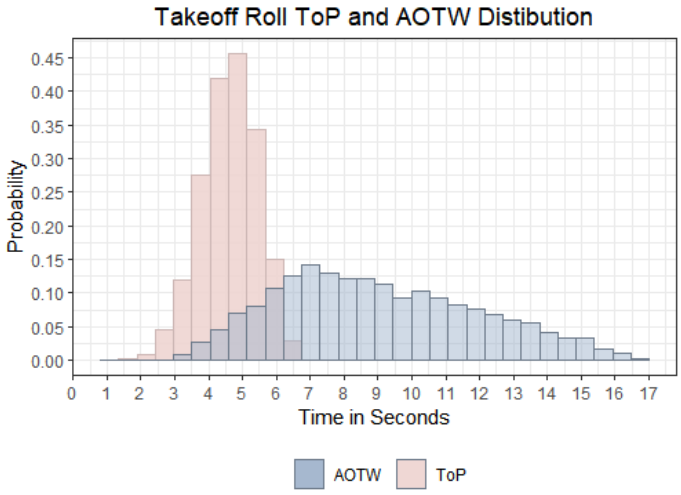



**Figure 30 ToP & AOTW Distributions**

### 4.1.4 Discussion

The results demonstrate the applicability of this method in a sample of fifteen aviation operational SOPs. The SOPs contained a total of 171 steps, 70% of which were Action-Only, 14% were Decision-Action, 12% were Action with Waiting -Verification,

and 4% were Decision-Action with Waiting – Verification. Among the steps with waiting requirements, the majority of the steps did not contain a verification after waiting.

Across all SOP step types, the triggering conditions were frequently stated (~60%). Less often present (~34%) in the SOP was the data that the operator sees to initiate a step, and rarely stated (~12%) in the SOP was the output display that the operator looks at. The input devices and the physical motion required to manipulate the input device were present in the SOP 85% and 54% of the time, respectively. This highlights the importance of recall from the operator's system description knowledge, as these gaps must be filled.

The SOP step recall for the set of SOPs ranged from 25% to 81%. This requires a range of 3 to 206 repetitions and a range of 2.4 to 12.2 days. This was due to the large number of steps in an SOP and/or a high SOP step recall score.

Finally, a segment of the takeoff SOP was simulated, and a Probability of Failure to Complete (PFtC) of 5.72% was achieved. This was because of the presence of a cue that was characterized as having ambiguous semantics.

## 4.2 Case Study Two: International Space Station (ISS) Habitable Airlock (HAL) SOPs

The Habitable Airlock (HAL) is a cabin designed to support astronauts during short-term scientific studies when detached from larger vehicles in missions to the Moon, Mars, and beyond (Figure 31). To prepare the HAL for Extravehicular Activities (EVA), operators use established procedures. Once the HAL is prepared and detached, operators

use another set of SOPs (non-EVA procedures) to collect samples, make observations, and deploy systems to manipulate objects from other airlocks.

**Figure 31 Habitable Airlock (HAL)**

### 4.2.1 Objective

The objective of this case study is to demonstrate the methodology in a sample of training procedures where operators' roles are time-sensitive and the operator selection process is highly selective.

### 4.2.2 Data and Preprocessing

The International Space Station (ISS) provided a total of 21 procedures. The procedures were categorized according to the content of each procedure. Thirteen (13) procedures were classified as "manual manipulation of physical items" or EVA

procedures. These were procedures that did not contain any Human-Machine Interaction (HMI) but were necessary to configure the HAL for EVA (Figure 31). These SOPs were removed from the analysis, as this analysis strictly applies to procedures with HMI.

Eight (8) of the procedures contained steps where the operator interacts with a machine (non-EVA procedures). The steps in each SOP were manually classified and stored in an Excel sheet for analysis. One individual completed the classification to assure consistency in the manual classifications. The eight SOPs had a total of 321 SOP steps and 893 e-PRL elements.

### 4.2.3 Results

#### *4.2.3.1 SOP Step Type*

The analysis uncovered statistics for the four main types of SOP steps, as seen in Table 16.

**Table 16 SOP Step Types – Case Study 2**

| SOP Step Type | Percent of SOP Steps |
|---|---|
| Action-Only | 73% |
| Decision-Action | 0% |
| Action with Waiting-Verification | 27% |
| Decision-Action with Waiting - Verification | 0% |

The Action-Only step, which consists of a (1) trigger and a (2) action, was the most prevalent type of SOP step. The Decision-Action and the Decision with Waiting and Verification steps were not present in the analysis of the eight procedures. The Action

with Waiting and Verification step consists of (1) trigger, (2) action, (3) waiting, and (4) verification and constitutes about 27% of the SOP steps.

#### *4.2.3.2 SOP Step Recall Score by SOP Step Type*

The content of the SOP steps by type was analyzed to determine which elements were explicitly specified in the SOP steps. Table 17 summarizes the results, and the following sub-sections describe the results in detail.

**Table 17 Content Explicitly Present in the SOP Steps**

| SOP Element | Action-Only | Decision-Action | Action with Waiting & Verification | Decision-Action with Waiting & Verification |
|---|---|---|---|---|
| Percentage | 73% (233 out of 321) | 0.00% (0 out of 321) | 27% (88 out of 321) | 0.00% (0 out of 321) |
| Actor | 0.00% (0 Out of 233) | 0% (0 out of 0) | 0.00% (0 Out of 88) | 0% (0 out of 0) |
| Trigger | | | | |
| What | 3% (8 Out of 233) | 0% (0 out of 0) | 1% (1 Out of 88) | 0% (0 out of 0) |
| Where | 58% (136 Out of 233) | 0% (0 out of 0) | 74% (65 Out of 88) | 0% (0 out of 0) |
| How | 48% (111 Out of 233) | 0% (0 out of 0) | 77% (68 Out of 88) | 0% (0 out of 0) |
| Decide | | | | |
| What | | 0% (0 out of 0) | | 0% (0 out of 0) |
| Where | | 0% (0 out of 0) | | 0% (0 out of 0) |
| How | | 0% (0 out of 0) | | 0% (0 out of 0) |
| Action | | | | |
| What | 100% (233 Out of 233) | 0% (0 out of 0) | 100% (88 Out of 88) | 0% (0 out of 0) |
| Where | 9% (22 Out of 233) | 0% (0 out of 0) | 1% (1 Out of 88) | 0% (0 out of 0) |
| How | 2% (4 Out of 233) | 0% (0 out of 0) | 1% (1 Out of 88) | 0% (0 out of 0) |
| Waiting | | | | |

| | | | | |
|---|---|---|---|---|
| What | | | 6%<br>(5 Out of 88) | 0%<br>(0 out of 0) |
| Where | | | 10%<br>(9 Out of 88) | 0%<br>(0 out of 0) |
| How | | | 18%<br>(16 Out of 88) | 0%<br>(0 out of 0) |
| Verification | | | | |
| What | | | 42%<br>(37 Out of 88) | 0%<br>(0 out of 0) |
| Where | | | 24%<br>(21 Out of 88) | 0%<br>(0 out of 0) |
| How | | | 76%<br>(67 Out of 88) | 0%<br>(0 out of 0) |
| Average SOP Step Recall Score | 71% | 0 % | 42% | 0% |

*4.2.3.2.1 Action-Only Steps*

Seventy-three percent (73%) of all SOP steps were Action Only. This type of SOP step only requires an action and a trigger (other e-PRL elements in the table grayed out). The trigger was explicitly present only 3% of the time, and the source of the data (the output display) was specified 58% of the time. The data, or the cue presented to the operator, was present 48% of the time. The average SOP step recall score was 71%, meaning that, on average, 71% of an Action-Only step would require the operator's memory.

*4.2.3.2.2 Action with Waiting & Verification Steps*

Twenty-seven percent (27%) of the SOP steps were Action with Waiting Verification. These steps have the form of (1) Trigger, (2) Action, (3) Waiting, and (4) Verification. One percent (1%) of the SOP steps in this category contained a trigger; however, the data for the trigger was present 77% of the time. The source of the data was present 74% of the time. Forty-two (42%) of the Action with Waiting and Verification steps contained a waiting requirement, followed by a verification requirement, leaving 58% of those steps without any verification. The average SOP step recall score was 42%, meaning that, on average, an operator would have to recall 42% of the Action-with Waiting & Verification step.

***4.2.3.3 SOP Step Recall Score by SOP***

Table 18 displays the average SOP step recall for each SOP in the analysis.

**Table 18 SOP Step Recall Score By SOP**

| SOP | Average SOP Step Recall (Action-Only) | Average SOP Step Recall (Action with Waiting & Verification) | Average SOP Step Recall Score |
|---|---|---|---|
| 5.101 | 78% | 13% | 71% |
| 5.111 | 36% | 61% | 56% |
| 5.121 | 71% | 35% | 61% |
| 5.122 | 66% | 34% | 54% |
| 5.999 | 69% | 33% | 58% |
| 3.1 | 100% | 69% | 96% |
| 3.1A | 100% | 73% | 88% |
| 3.1F | 100% | 73% | 88% |

#### *4.2.3.4 Estimating Training Requirements*

The total number of repetitions required for each SOP, along with the number of days to proficiency, are summarized in Table 19. The number of steps column is the number of steps in the SOP. The updated number of steps is a calculated column using Equation 4.

**Table 19 Training Estimates**

| SOP | Num Steps | Average SOP Recall Score | Total Number of Repetitions (TNR) | Number of Days (ND) |
|---|---|---|---|---|
| 5.111 | 103 | 31% | 316 Reps | 13.1 Days |
| 5.101 | 10 | 71% | 38 Reps | 8.5 Days |
| 3.1 | 17 | 96% | 76 Reps | 10 Days |
| 3.1_F | 17 | 88% | 72 Reps | 9.9 Days |
| 3.1_A | 17 | 88% | 72 Reps | 9.9 Days |
| 5.121 | 54 | 61% | 202 Reps | 12.1 Days |
| 5.122 | 52 | 54% | 185 Reps | 11.9 Days |
| 5.999 | 51 | 58% | 187 Reps | 12 Days |

#### *4.2.3.5 Simulation of SOPs*

All eight (8) SOPs were simulated with 7500 iterations. Since the SOPs did not specify an Allowable Operational Time Window (AOTW) for each SOP, an AOTW was calculated for each SOP by assuming that all steps were in the Field of View (FoV), salient, and unambiguous, having a time distribution of Triang(0.3, 5.2, 24.3) seconds. Therefore, the maximum, mode, and minimum of the AOTW for an SOP are given by Equation 17, Equation 18, and Equation 19, respectively. Table 20 summarizes the calculated AOTW assumed for each SOP, along with the SOP Recall Score and the PFtC.

**Equation 17 Maximum of AOTW**

$$Max_{AOTW} = Num\ Steps * 24.3$$

**Equation 18 Mode of AOTW**

$$Mode_{AOTW} = Num\ Steps * 5.2$$

**Equation 19 Minimum of AOTW**

$$Min_{AOTW} = Num\ Steps * 3.1$$

**Table 20 Calculated AOTW and Simulated PFtC - Case Study 2**

| SOP | AOTW | PFtC |
|---|---|---|
| 5.111 | Triang(375.1, 629.2, 2940.3) s | 34% |
| 5.101 | Triang(34.1, 57.2, 267.3) s | 45% |
| 3.1 | Triang(58.9, 98.8, 461.7) s | 37% |
| 3.1_F | Triang(52.2, 88.4, 413.1) s | 34% |
| 3.1_A | Triang(77.5, 130, 607.5) s | 38% |
| 5.121 | Triang(210.8, 353.6, 1652.4) s | 31% |
| 5.122 | Triang(213.9, 358.8, 1676.7) s | 34% |
| 5.999 | Triang(201.5, 338, 1579.5)s | 29% |

Each of the eight SOPs was simulated. The AOTW assumed is given in Table 20. The ToP vs. AOTW scatter plot for each SOP and the ToP and AOTW distributions are displayed in Figure 32 and Figure 33, respectively.

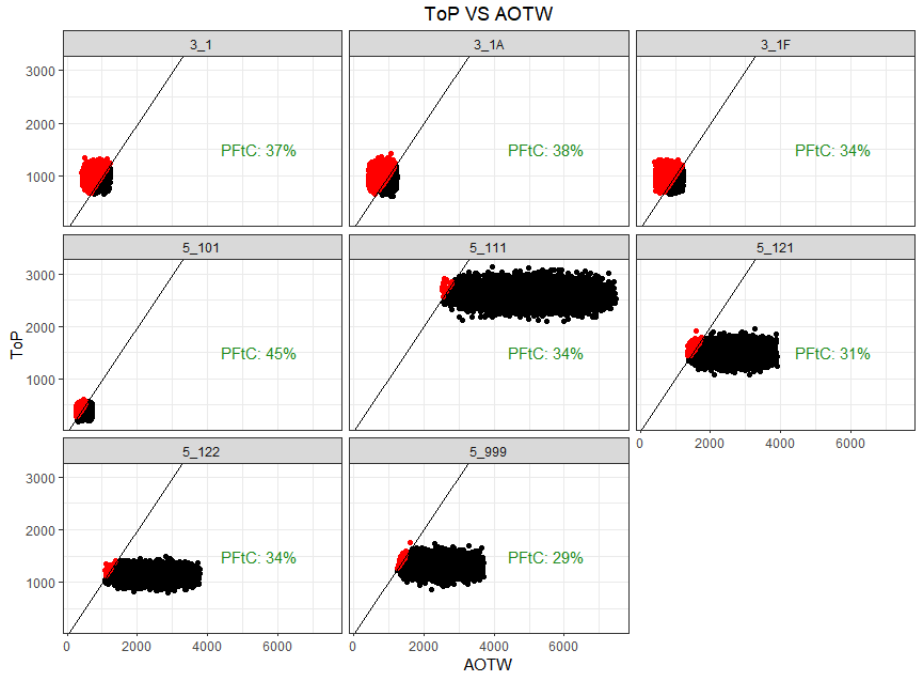



**Figure 32 ToP VS AOTW Scatter Plot - Case Study 2**

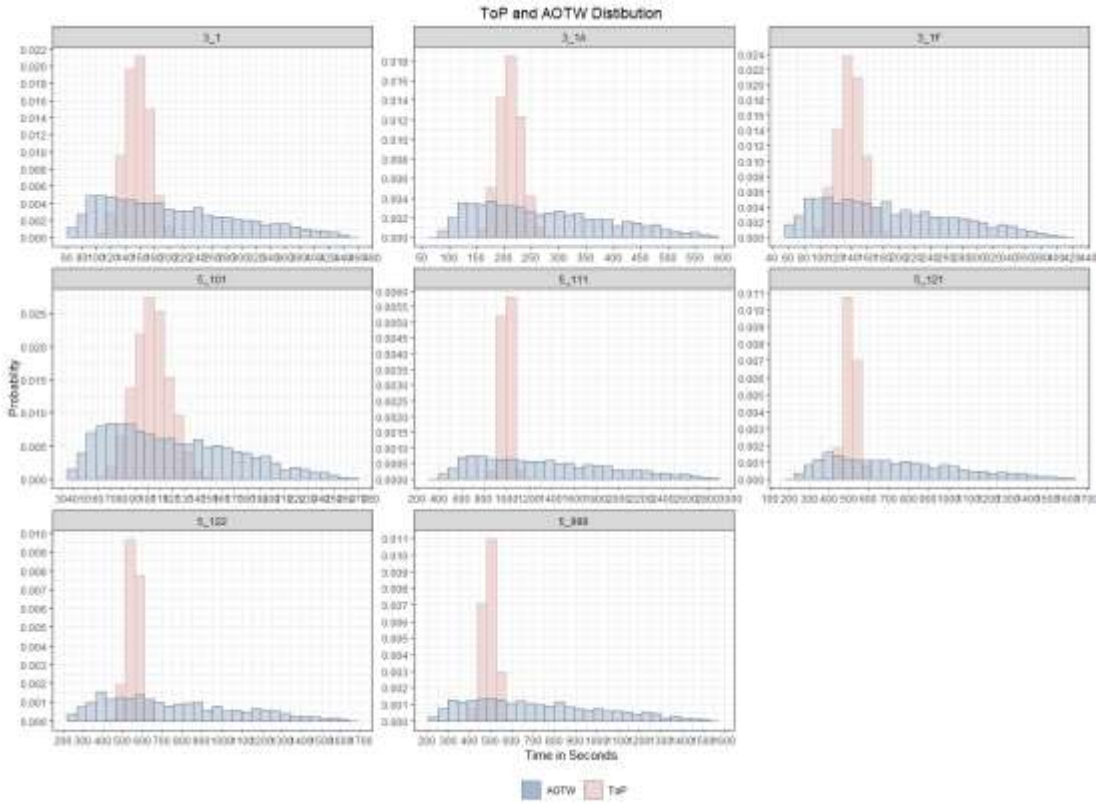



**Figure 33 ToP and AOTW Distributions - Case Study 2**

#### *4.2.3.6 MBSE Diagrams*

A step from the 5.111 SOP is documented in Figure 34. The e-PRL representation of the same step is in Figure 35. This step was categorized as an Action-Only Step. A more complex step, with an instantaneous waiting and a verification requirement, is in Figure 36, and its e-PRL representation is in Figure 37. Note how the implied system description elements are translated into Long-Term Memory (LTM) items in the MBSE Action Diagrams.

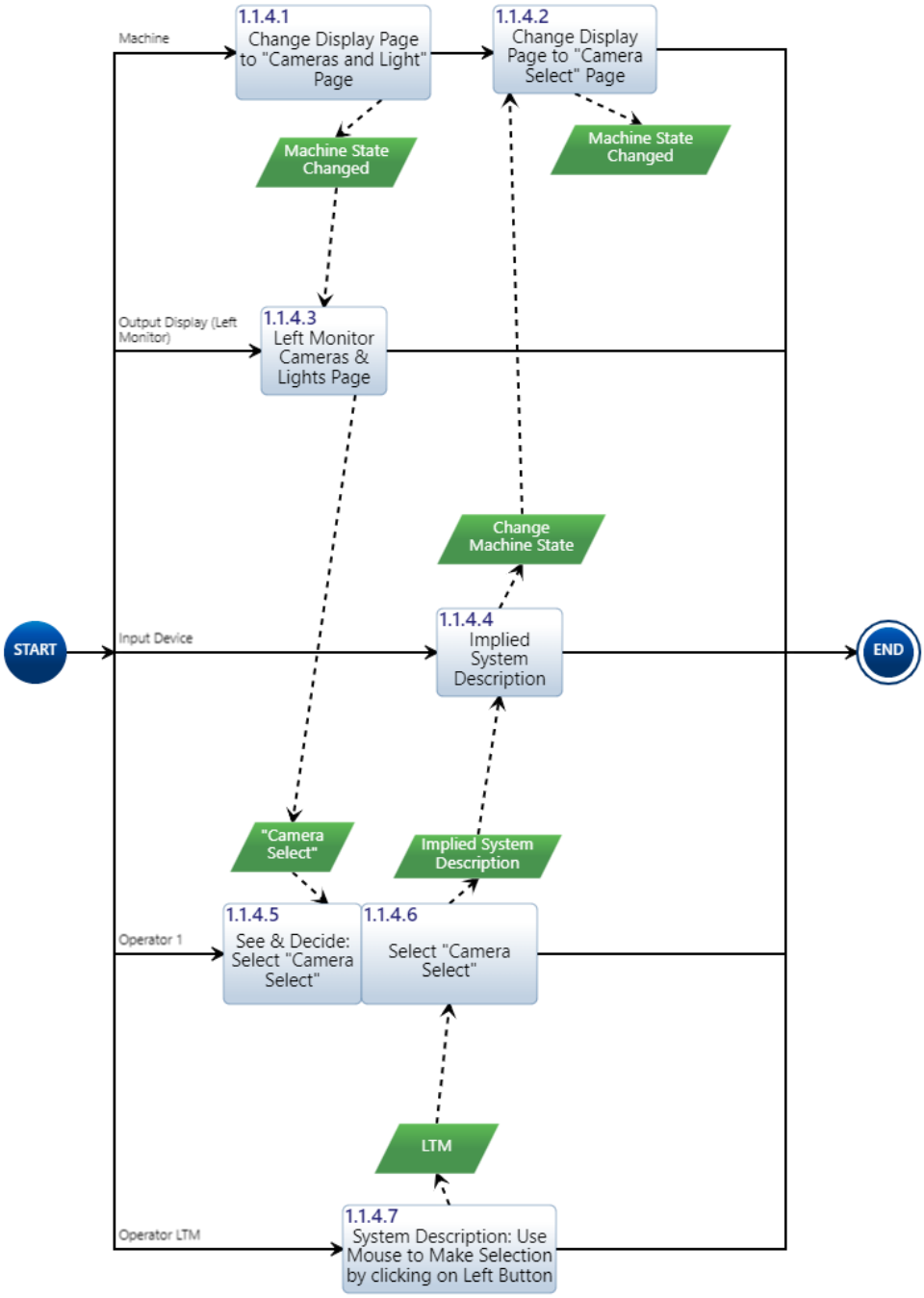



**Figure 34 SOP Step in MBSE Action Diagram Form**

| | | | | | Triggering Information | | | Decision | | | Action | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Procedure File Number | Step Number | ACTOR | Initiating Condition (What) | Data (How) | Source of Data (Where) | Decision (What) | Data (How) | Source of Data (Where) | Action (What) | Physical Motion (How) | Input Device (Where) |
| | | HAL_5_111.pdf | 4 | | Implied/ Sequential | "Camera Select" | DCAPP main screen left monitor | | | | Select "Camera Select" | Implied/ System Description | Implied/ System Description |

**Figure 35 e-PRL Representation of SOP Step in Figure 34**

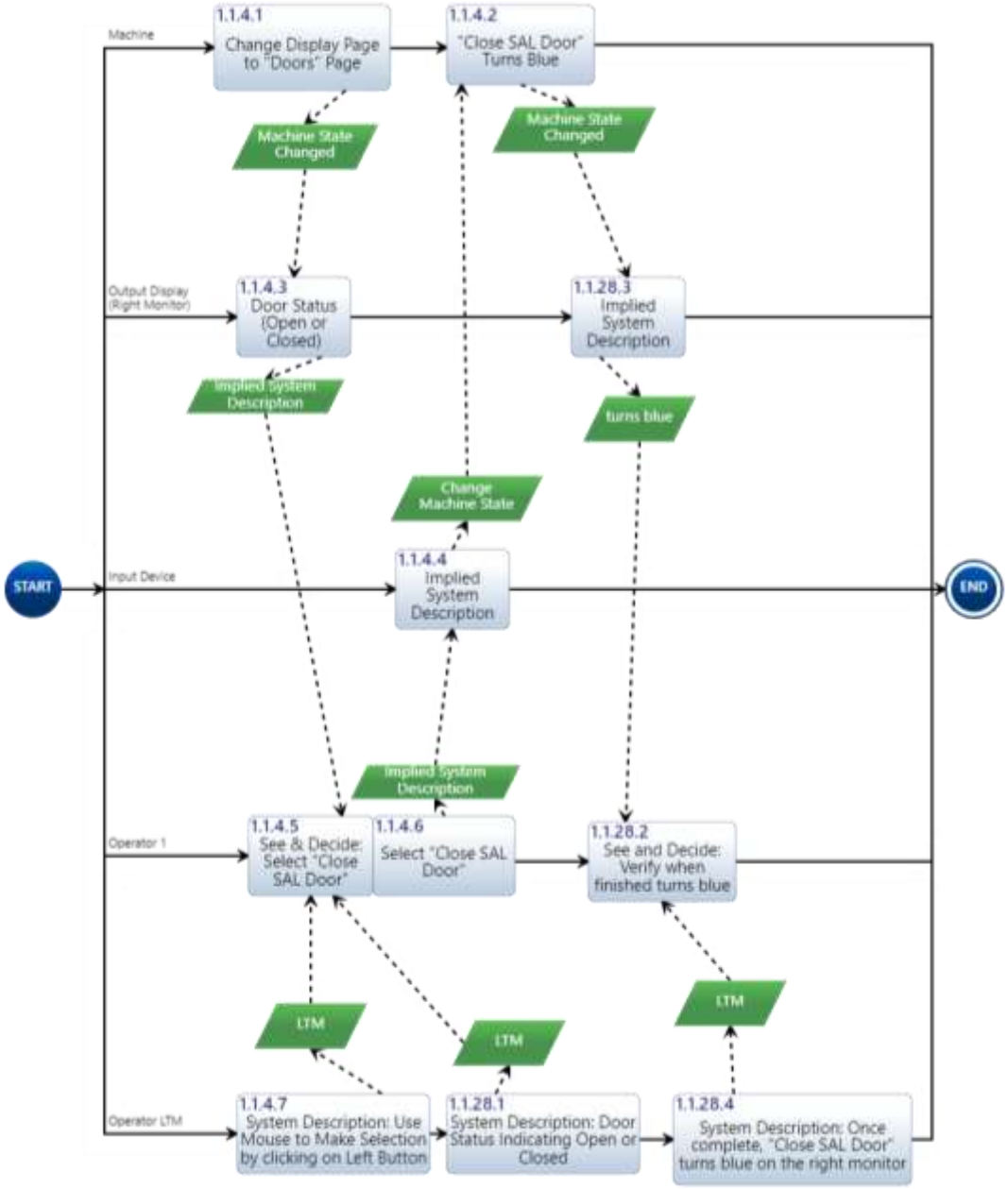



**Figure 36 Complex SOP Step with Waiting and Verification**

| | | | Triggering Information | | | Decision | | | Action | | | Waiting/Timing | | | Verification | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Procedure File Number | Step Number | ACTOR | Initiating Condition (What) | Data (How) | Source of Data (Where) | Decision (What) | Data (How) | Source of Data (Where) | Action (What) | Physical Motion (How) | Input Device (Where) | Waiting/ Timing (What) | Data (How) | Source of Data (Where) | Verification (What) | Data (How) | Source of Data (Where) |
| HAL_3_11 1.pdf | 28 | | If the door is open | Implied/ System Description | DCAPP main screen right monitor | | | | select "Close SAL Door" | Implied/ System Description | Implied/ System Description | Implied/I nstantan eous | Implied/I nstantan eous | Implied/I nstantan eous | when finished | turns Blue | Implied/ System Description |

**Figure 37 e-PRL Representation of Figure 36**

### 4.2.4 Discussion

The results demonstrate the applicability of the SOP e-PRL model in the ISS HAL domain. The analysis found that 73% of the total SOP steps were Action-Only steps. These steps had a higher SOP recall score, indicating that the operators would need to rely heavily on memory to execute the majority of the steps. The remaining 27% were Action with Waiting & Verification steps and had an average recall score of 42%. Across both step types, the triggering condition was rarely stated, mainly because the steps were completed sequentially and did not require any trigger. The data needed to trigger the next step was present 58% of the time in the Action Only steps and 74% of the time in the Action with Waiting & Verification steps. Additionally, knowledge regarding what input device to use and how to use it was consistently absent from both step types. This poses a risk to operators, especially novice operators, and results in longer training times.

The estimates of training requirements show that these SOPs required a range of 8 to 13 days per SOP. The longest SOP requires 316 repetitions to reach proficiency, which is eight times as many as the shortest SOP, which requires 38 repetitions. This highlights opportunities to balance training loads by spacing SOPs with more repetitions between simpler ones.

Examining the SOP step recall score for each SOP reveals a tradeoff between simple/shorter procedures with high recall scores and complex/longer procedures with lower recall scores. In the case of SOP 5.101, even though the SOP step recall score was high at 71%, the procedure has ten steps that the operators can reasonably recall information from memory. On the other hand, SOP 5.111, with over 100 steps and a recall score of 51%, places excessive demands on the operators' memory. They are forced to retain far more information over a prolonged period of time, which creates an unnecessary cognitive workload for the operators.

Most importantly, this analysis examines the absence of written verification requirements for Action with Waiting & Verification steps. The data shows that 58% of the steps that should include a verification requirement omit that element from the written step. Without specifying what constitutes a correct system state or acceptable thresholds, operators must either rely on memory or judgment depending on their mental model.

### 4.2.5 Summary

The case study demonstrated the methodology for procedure evaluation, estimating training requirements, and simulating human performance. Results highlight that 73% of the total 321 SOP steps were Action-Only steps, which rarely explicitly stated triggers and information regarding input devices and how to manipulate them. Meanwhile, the Action-with Waiting & Verification steps had lower recall scores but were missing a crucial element required for this step type, which is the verification.

Estimating training requirements for the SOPs showed some variability in the number of repetitions required, ranging from 38 reps to 316 reps. Simulating the human

performance generated reasonable PFtC; however, this was done on the assumption that the AOTW was calculated from the ToP distribution. Overall, the methodology facilitated diagnostics into the set of SOPs that could result in enhancements to the ISS HAL operations.

#### *4.2.5.1 Findings*

Based on the results and discussion, this analysis recommends the following:

1) To reduce training times:

a) Explicitly specify elements in the SOP such that training repetitions and days required can be shortened. Having SOPs with gaps leads to imposing memorization strains on trainees, which leads to the need for more repetitions and longer training days.

b) Explicitly specify details on interfaces used and how to use them to enable effective operation, especially for novice users.

2) To reduce PFtC:

a) Reduce LTM items, especially for steps requiring verification, such that operators do not rely solely on memory or one's mental model.

## 4.3 CASE STUDY THREE: AUTONOMOUS VEHICLE SOPS

As vehicles with Full Self Drive (FSD) capabilities are becoming ubiquitous, the role of the operator is changing. Specifically, the human operator is a fallback and is required to ensure safety whenever automation reaches its limit. Operators still have a legal responsibility to monitor the performance of the FSD system and to intervene in a

timely manner to avoid hazards. Published Standard Operating Procedures (SOPs) define what the intervention is and how the operator shall perform this intervention.

This case study demonstrates how to use Large Language Models (LLMs)--based chatbots to aid in generating SOP steps from an owner's manual, as well as to reduce the time to convert an SOP into SOP step components, i.e., e-PRL classification.

### 4.3.1 Objective of the Case Study

SOPs are critical for liability purposes and operator training. State Motor Vehicle agencies are responsible for the safe operation of cars in their state. This is accomplished by providing training, testing, and license endorsement to certify vehicle safety. It is not apparent to what degree these agencies evaluate the SOPs.

The objective of this case study is to evaluate the current SOPs of a semi-autonomous vehicle (Tesla Model 3) with Full-Self-Driving (FSD) capabilities concerning the content provided in the SOP, as well as operational safety. Content in the SOP relates to training requirements, and operational safety relates to the efficacy of the cues presented to the operators and whether the operators can take over the semi-autonomous vehicle before a hazard occurs.

### 4.3.2 Data and Preprocessing

The Tesla Model 3 Owner's Manual was used [61] (Figure 38). The manual is a 300-page document with images describing the input/output displays and text that covers the operation of the vehicle. The Autopilot section used for this case study contained 43 two-column styled pages.

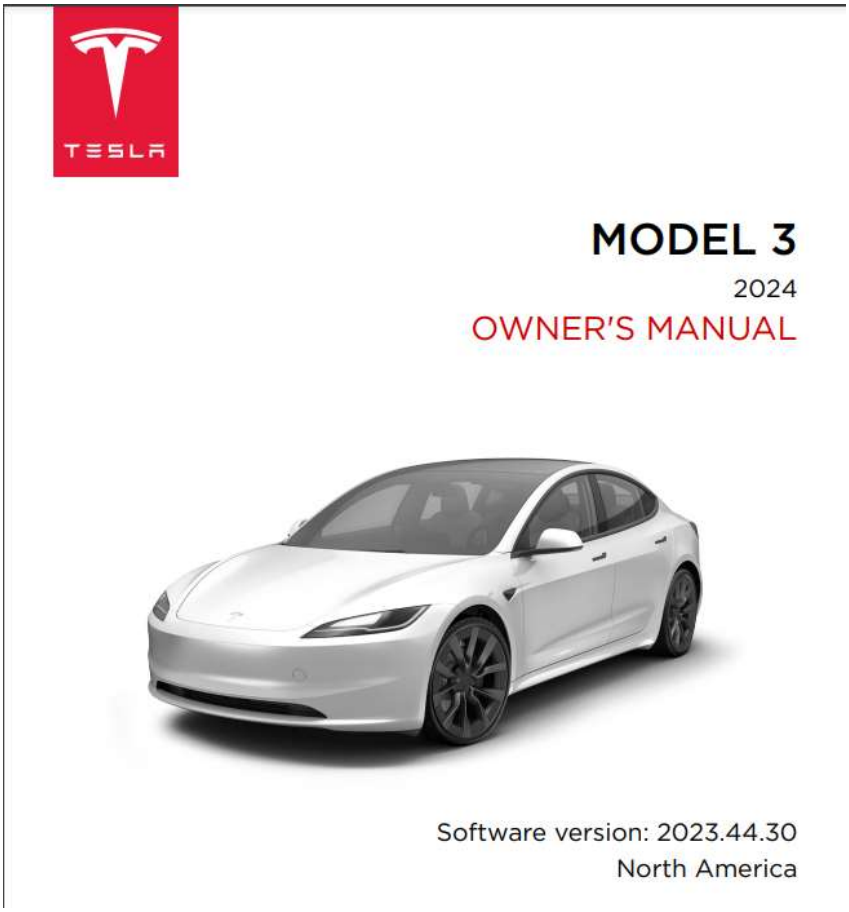



**Figure 38 Tesla Model 3 Owner's Manual**

SOPs for regular, everyday interaction with the vehicle, such as engaging or disengaging autopilot, were not analyzed for this analysis, as they are not associated with any risk if not completed in time. The only procedures examined are SOPs for human takeover, as they do have a time limit associated with them, and it is crucial to gauge whether operators can complete the takeover request before a hazard occurs.

#### *4.1.2.1 Generate SOPs Steps from Owner's Manual*

This section describes how to convert an owner's manual to SOPs using an LLM-based chatbot.

To generate procedures for human intervention requests, the first step is to understand when a human takeover request is required to be initiated. To do so, the manual was imported using the import button (Figure 39). Then, the LLM-based chatbot, Claude, was asked to generate a list of scenarios where a human takeover is required. This generated a list of scenarios that would require SOPs (Figure 40).

**Figure 39 Import Button in Claude**

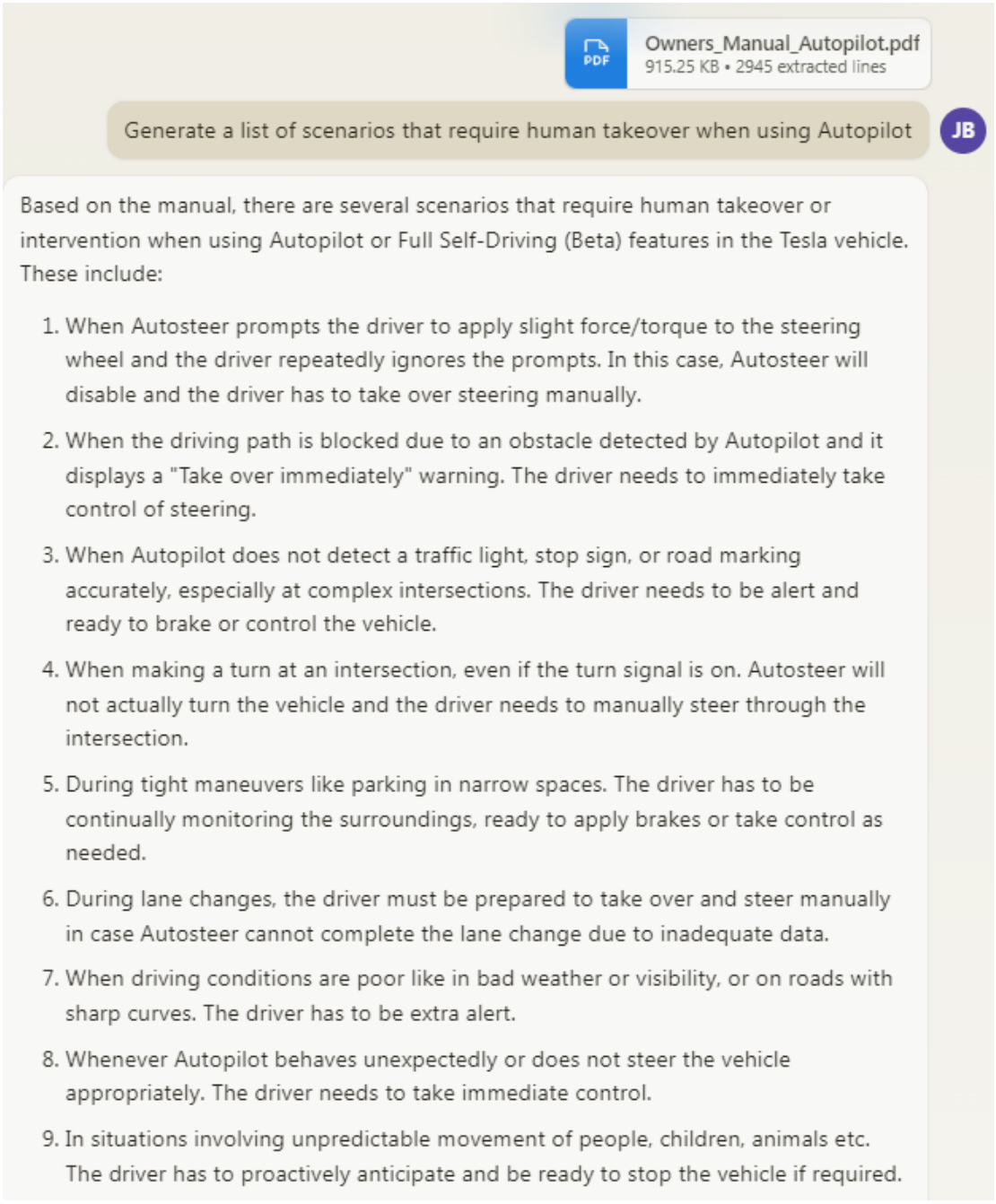
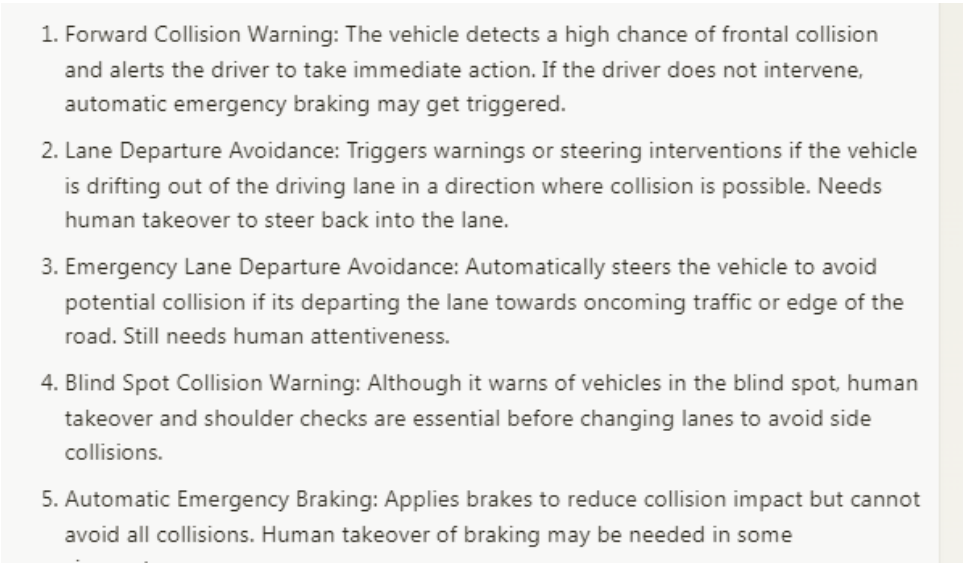



**Figure 40 Scenarios where Human Takeover is Required**

In addition to the scenarios listed above, the Owners' Manual mentions several collision avoidance features that require human takeover, which are Forward Collision,

Lane Departure Avoidance, Emergency Lane Departure, Blind Spot Collision Warning, and Automatic Emergency Braking (Figure 41).



1. Forward Collision Warning: The vehicle detects a high chance of frontal collision and alerts the driver to take immediate action. If the driver does not intervene, automatic emergency braking may get triggered.
2. Lane Departure Avoidance: Triggers warnings or steering interventions if the vehicle is drifting out of the driving lane in a direction where collision is possible. Needs human takeover to steer back into the lane.
3. Emergency Lane Departure Avoidance: Automatically steers the vehicle to avoid potential collision if its departing the lane towards oncoming traffic or edge of the road. Still needs human attentiveness.
4. Blind Spot Collision Warning: Although it warns of vehicles in the blind spot, human takeover and shoulder checks are essential before changing lanes to avoid side collisions.
5. Automatic Emergency Braking: Applies brakes to reduce collision impact but cannot avoid all collisions. Human takeover of braking may be needed in some

**Figure 41 Collision Avoidance Features Where Human Intervention may be Required.**

The SOPs were derived from the lists in Figure 40, which were LLM-based chatbot generated, and Figure 41, which were manually generated. The list of SOPs analyzed in this case study is shown in Table 21.

**Table 21 Procedures, Initiating Conditions, Alerts, and Hazardous Events that May Occur**

| SOP | Initiation Condition | Alerts Provided from the Vehicle for Triggering Condition | Hazardous Event |
|---|---|---|---|
| Forward Collision Warning Procedure | Initiated when a vehicle detects a high probability of an imminent frontal collision with another object | The vehicle sounds a chime, highlights the vehicle in front in red on the touchscreen, and the steering wheel vibrates. | The vehicle collides with an object in front. |
| Lane Departure Avoidance Procedure | Initiated when the vehicle drifts out of its driving lane towards another lane or road edge. | The vehicle sounds a chime, displays a red line highlighting the lane marking being crossed, and the steering wheel vibrates. | A vehicle collides with an object or veers off-road. |
| Blind Spot Collision (Side Collision) Procedure | Initiated when the vehicle detects an object near the vehicle's blind spot/side while a lane change maneuver is being attempted. | The vehicle sounds a chime, and the display displays colored lines radiating from the image of the vehicle. | A vehicle collides with an object (side collision). |
| Autosteer Disable Procedure | Initiated when vehicle disables "Autosteer" due to the operator ignoring hands-on torque reminders and/or encountering a situation where Autosteer is unable to steer accurately. | The vehicle sounds a chime and displays a "Takeover Steering Immediately" message. | The vehicle slows down gradually until it comes to a complete stop. |
| Intersection Turn Takeover Procedure | Initiated when the vehicle approaches an intersection in Autosteer mode and needs to make a turn according to the navigation route. | The vehicle sounds a chime; the navigation display shows an upcoming turn required, and the boundaries of the intersection on visualization turn blue to indicate Autosteer is active. | The vehicle stops mid-intersection. |
| Unexpected Autopilot Behavior Intervention Procedure | Initiated when the vehicle makes sudden inaccurate or erratic steering actions such as swerving or unexpectedly slowing down | The vehicle displays a "take over immediately" message, sounds a chime, and/or the steering wheel vibrates. | Rear-end Collision: If the vehicle erratically brakes.<br>Frontal Collision: If the vehicle erratically accelerates. |

To get the step-by-step procedure, the chatbot was prompted to write the steps for a procedure (Figure 42).

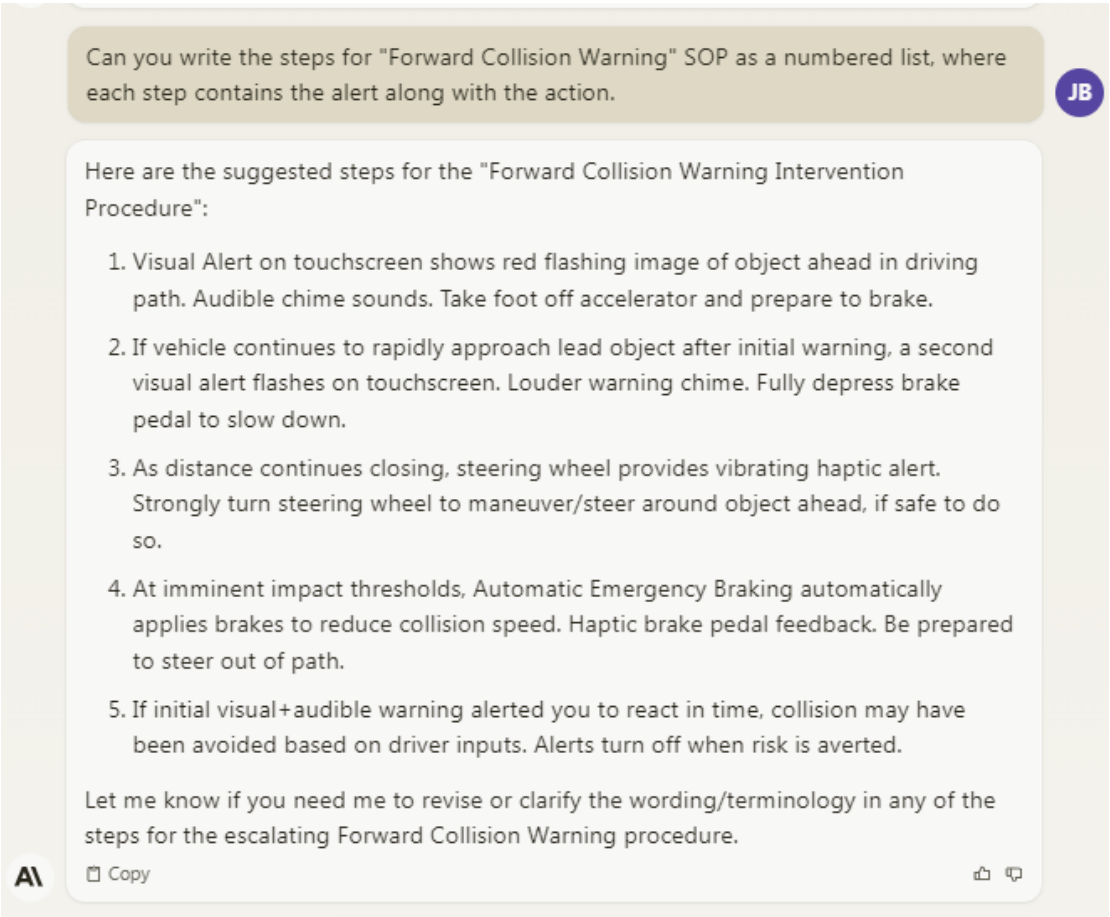



**Figure 42 Claude Prompt to Generate Procedure Steps for "Forward Collision Warning" SOP**

The chatbot was prompted to rewrite the steps to be in the Trigger-Action format.

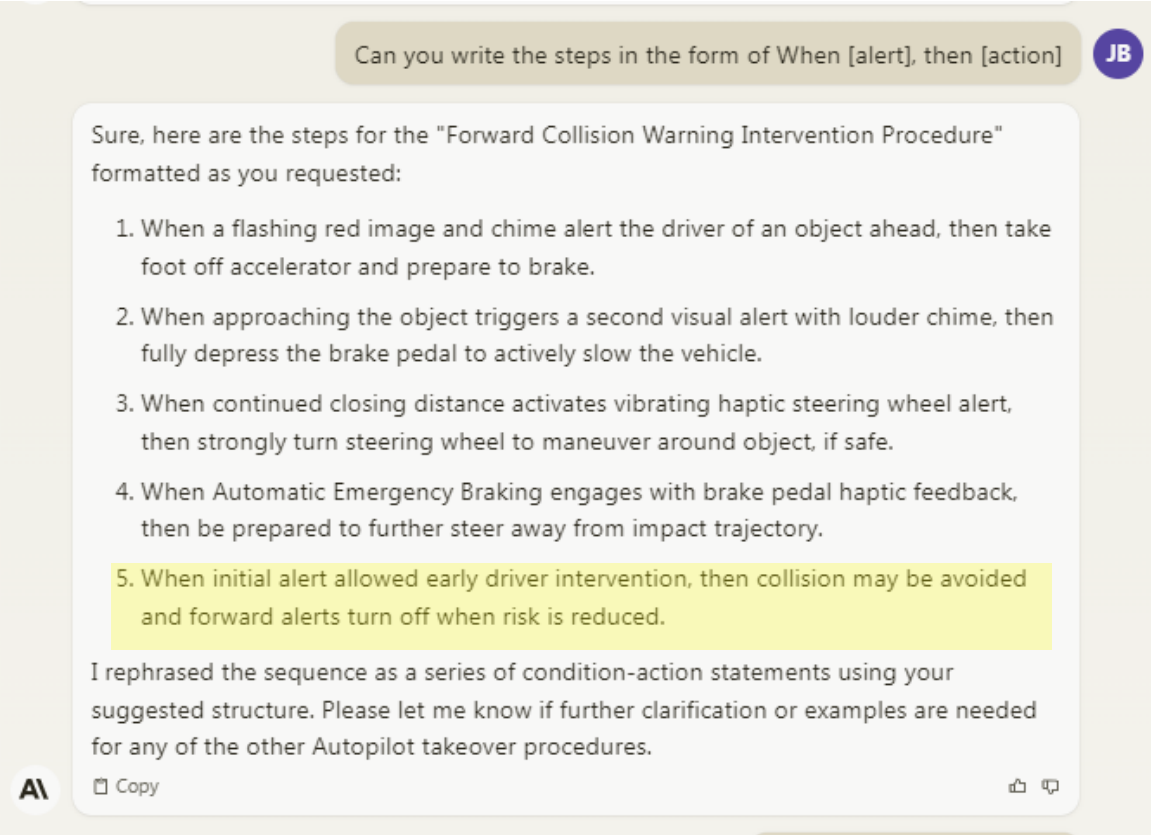



**Figure 43 Claude Prompt and Response to Re-Write SOP steps in Trigger-Action Format.**

The resulting procedure is modified manually to remove any non-actions, e.g., step 5 in Figure 43. The final version of the SOP used in the analysis is displayed in Figure 44. The SOPs generated as part of this case study are given in Appendix C.

Takeover when there is a potential forward collision:

1. When you hear a loud chime and an obstacle is highlighted in red on the touchscreen, place your foot hovered over the brake pedal to prepare for braking.
2. If the chime becomes louder, then fully depress the brake pedal to slow down the vehicle.
3. If the steering wheel begins to vibrate, then strongly turn steering wheel to maneuver around the obstacle, if safe.
4. If Automatic Emergency Braking engages with brake pedal haptic feedback, then operate vehicle manually until you re-engage Autopilot.

**Figure 44 "Forward Collision Warning" Procedure after Modification**

#### *4.1.2.2 Classify SOP Steps to Components*

This section describes how to classify the SOP into e-PRL components. The first step to classifying SOP text into an e-PRL model is to teach Claude the rules of labeling or annotating. The approach taken was to create an internal database of rules in the LLM-based chatbot for classifying text into e-PRL components. The rules for labeling text into e-PRL, along with the prompt used, are displayed in Figure 45 and were extracted from [62].

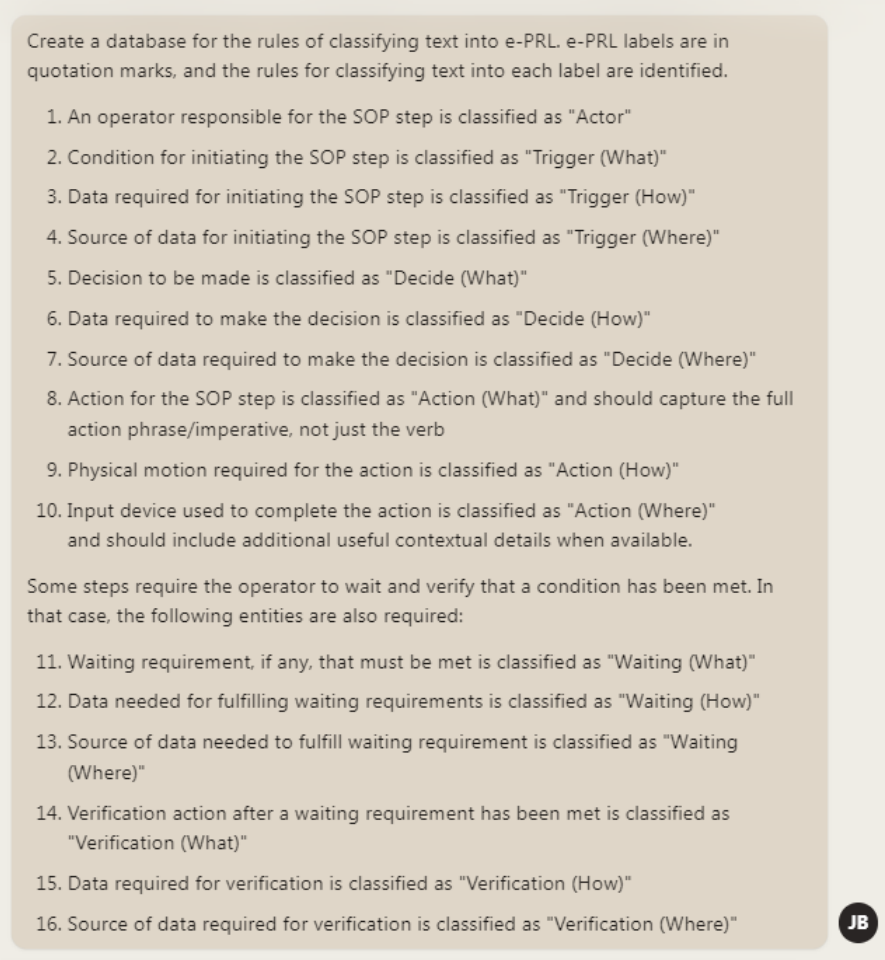



**Figure 45 Rules for Labeling SOP Text**

Next, Claude was asked to learn the rules in the database and use them for future classifications (Figure 46). The chatbot also detailed the approach that will be used to do the text classification in an attempt to clarify its reasoning. That is, the chatbot says, "Whenever you provide me with a new SOP step description, I will parse the text, extract relevant entities, and classify them into the standard e-PRL taxonomy based on the guidelines you outlined."

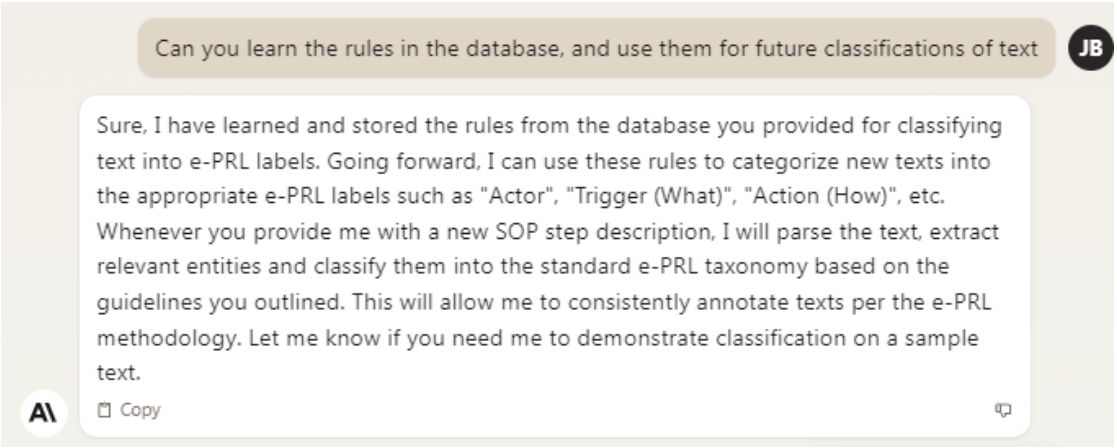



**Figure 46 Teaching Claude the Rules in the Database**

Next, Claude was asked to classify the text in the SOP steps into SOP step components. The classification task includes first breaking the SOP step into chunks and then classifying each chunk into the appropriate SOP step component. Figure 47 displays the prompt used to classify the text in an SOP into SOP step components. The results that Claude provided are shown in Table 22 Claude Classification Results of the Forward Collision Warning SOP

| Step | Trigger (What) | Action (How) | Action (Where) | Waiting (What) | Waiting (How) | Waiting (Where) |
|---|---|---|---|---|---|---|
| 1 | When you hear a loud chime, an obstacle is highlighted in red. | Prepare for braking | | brake pedal | | |
| 2 | If the chime becomes louder | Slow down vehicle | | brake pedal | | |
| 3 | If the steering wheel begins to vibrate | Maneuver around obstacle | | steering wheel | | |
| 4 | If Automatic Emergency Braking engages with brake pedal haptic feedback | Operate vehicle | | | | Until you re-engage Autopilot |

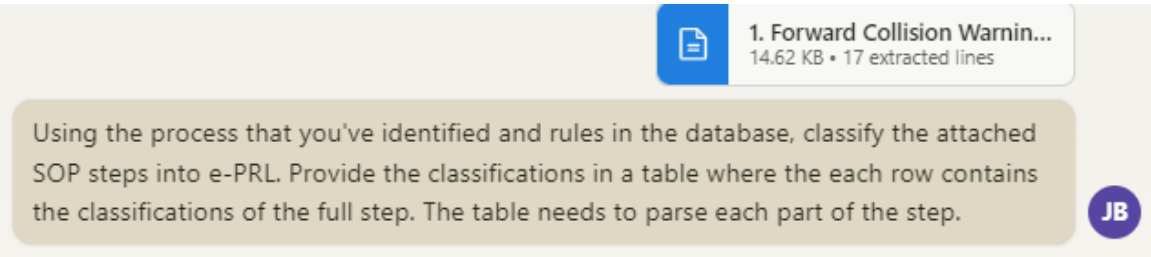



**Figure 47 Prompt to Classify SOP into e-PRL Components**

**Table 22 Claude Classification Results of the Forward Collision Warning SOP**

| Step | Trigger (What) | Action (How) | Action (Where) | Waiting (What) | Waiting (How) | Waiting (Where) |
|---|---|---|---|---|---|---|
| 1 | When you hear a loud chime, an obstacle is highlighted in red. | Prepare for braking | | brake pedal | | |
| 2 | If the chime becomes louder | Slow down vehicle | | brake pedal | | |
| 3 | If the steering wheel begins to vibrate | Maneuver around obstacle | | steering wheel | | |
| 4 | If Automatic Emergency Braking engages with brake pedal haptic feedback | Operate vehicle | | | | Until you re-engage Autopilot |

Next, a table with the correct classification was generated manually, i.e., by the author. The purpose of this step is to provide Claude with the correct classification so that Claude can understand what is correctly and incorrectly done. Based on the comparison,

Claude was asked to generate a list of new rules that can be added to the database. The prompt and response can be seen in Figure 48.

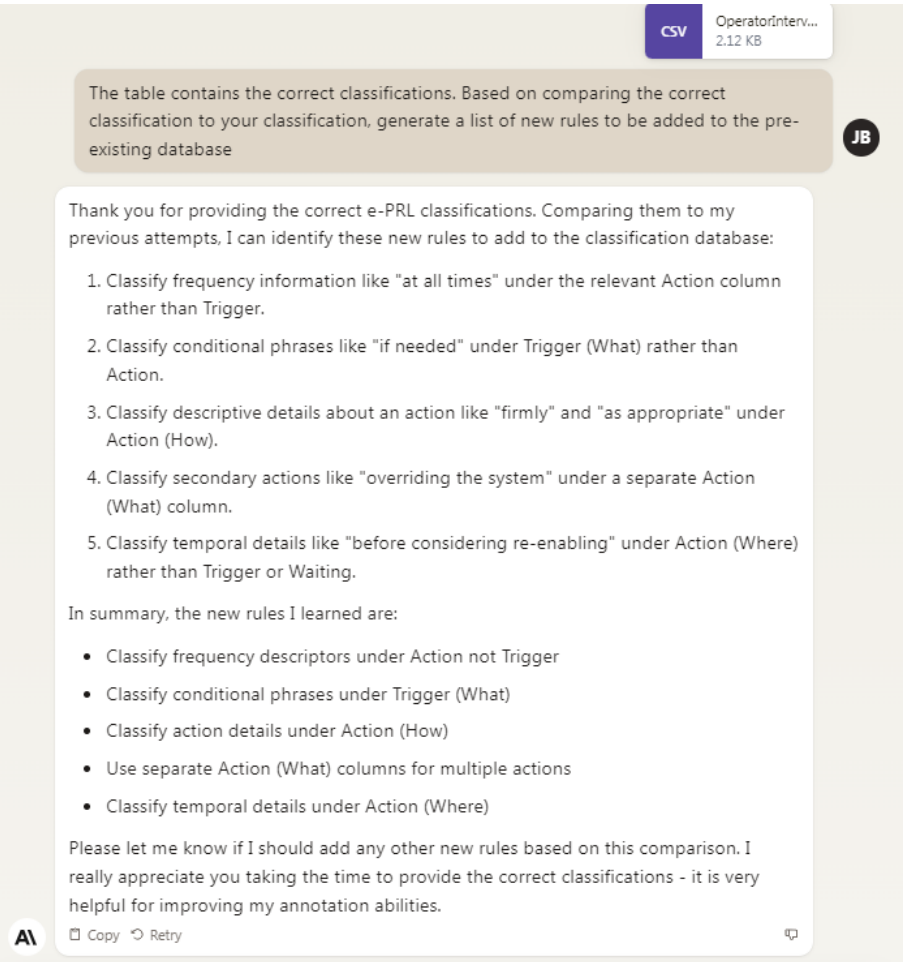



**Figure 48 Prompt to Generate New Rules**

Finally, the new set of rules was added to the existing Claude database (Figure 49), and Claude was prompted to learn the updated database. The process then starts over again until the SOP Chatbot achieves acceptable performance. Acceptable performance is when the LLM-based chatbot stops improving with the addition of new rules. That is, the

LLM-based chatbot has the same accuracy for two consecutive SOPs, which was at 80% in this case.

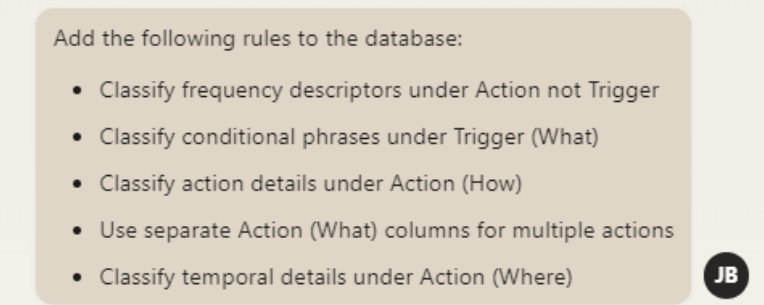



**Figure 49 Prompt to Add Generated Rules to Database**

The chatbot received a total of six (6) SOPs and their correct classifications. The classification accuracy of the LLM-based chatbot improved from 40% to 80% (Table 23). The accuracy of the classifications produced by the chatbot during each iteration was computed based on a comparison between the table of correct classification and Claude's classification. The number of correctly populated cells was divided by the total number of populated cells to compute the accuracy.

**Table 23 Accuracy of Chatbot After Each Iteration**

| Iteration | Percentage of Correctly Classified SOP Step Components in an SOP | Notes: New Rules Added to the Database of the LLM-based Chatbot |
|---|---|---|
| 1 | 40% | Classify frequency information like "at all times" under the relevant Action column rather than Trigger.<br>Classify conditional phrases like "if needed" under Trigger (What) rather than Action.<br>Classify descriptive details about an action like "firmly" and "as appropriate" under Action (How). |
| 2 | 50% | Classify device details like "touchscreen" under Action (Where), not Action (What)<br>Separate compound actions like "monitor and be prepared" into two Action (What) rows.<br>Classify secondary effects of actions under Verification (What), not Action (What)<br>Classify prompts/requests under Trigger (What) rather than Action (Where) |
| 3 | 56% | Classify descriptive instructions like "Roll up to increase speed, down to decrease" under Action (How), not Action (What)<br>Classify triggers relating to cancel/re-engage under Trigger (What)<br>Omit device details already captured in Action (Where) from Verification (Where) |
| 4 | 62% | Avoid adding information not stated in the SOP.<br>Waiting and Verification columns can be left blank if not applicable.<br>For Verification after an Action, only capture the key verification step |
| 5 | 80% | Summarize trigger conditions clearly and concisely.<br>Carry prior conditions into the next row's Trigger (What) if relevant.<br>Use present tense verbs in Trigger (What).<br>Avoid device repetition between Action and Verification (Where) |
| 6 | 80% | NA |

### 4.3.3 Results

There were twenty-five (25) steps in the six procedures analyzed. The following sub-sections discuss the results for each of the metrics.

#### *4.3.3.1 SOP Step Type*

The analysis uncovered statistics for the four main types of SOP steps, as seen in Table 24. The Action-Only step was the most prevalent type of SOP step, followed by the Action with Waiting and Verification step. There was one step that included a decision to be made, and none of the steps had both a decision and a verification.

**Table 24 SOP Step Types – Case Study 2**

| SOP Step Type | Percent of SOP Steps (out of 25 steps) |
|---|---|
| Action-Only | 80% |
| Decision-Action | 4% |
| Action with Waiting-Verification | 16% |
| Decision-Action with Waiting - Verification | 0% |

#### *4.3.3.2 SOP Step Recall Score by SOP Step Type*

Overall, the Action-Only step required the most recall at 50%, followed by the Decision-Action step, which required 33% recall. The Action-with Waiting & Verification step had the least amount of recall, with a 19% SOP step recall score. Table

25 summarizes the SOP step recall score by SOP step type. Figure 50 shows the count of SOP steps and the SOP step recall score.

**Table 25 SOP Recall Score by SOP Step Type – Case 3**

| SOP Step Type | Average SOP Step Recall Score |
|---|---|
| Action-Only | 50% |
| Decision-Action | 33% |
| Action with Waiting & Verification | 19% |

**Figure 50 SOP Step Recall Score - Case 3**

#### *4.3.3.3 SOP Step Recall Score by SOP*

The Intersection Turn Takeover SOP had the highest recall score, even though this SOP was one of the simplest, i.e., it had 100% Action-Only steps. On the other hand, procedures that had more complex steps, such as Forward Collision Warning SOP and Blind Spot Collision SOP, which had either a decision or verification, had lower recall scores. Table 26 summarizes the average SOP Step Recall score by SOP, and Figure 51 shows the count of SOP steps in each SOP step recall score.

**Table 26 SOP Step Recall Score by SOP – Case 3**

| SOP | Average SOP Step Recall Score (Action-Only) | Average SOP Step Recall Score (Decision-Action) | Average SOP Step Recall Score (Action with Waiting & Verification) | SOP Recall Score |
|---|---|---|---|---|
| Forward Collision Warning | 25% | 33% | NA | 27% |
| Lane Departure Avoidance | 33% | NA | 25% | 31% |
| Blind Spot Collision | 38% | NA | 13% | 25% |
| Autosteer Disable | 50% | NA | NA | 50% |
| Intersection Turn Takeover | 80% | NA | NA | 80% |
| Unexpected Autopilot Behavior Intervention | 58% | NA | 25% | 50% |

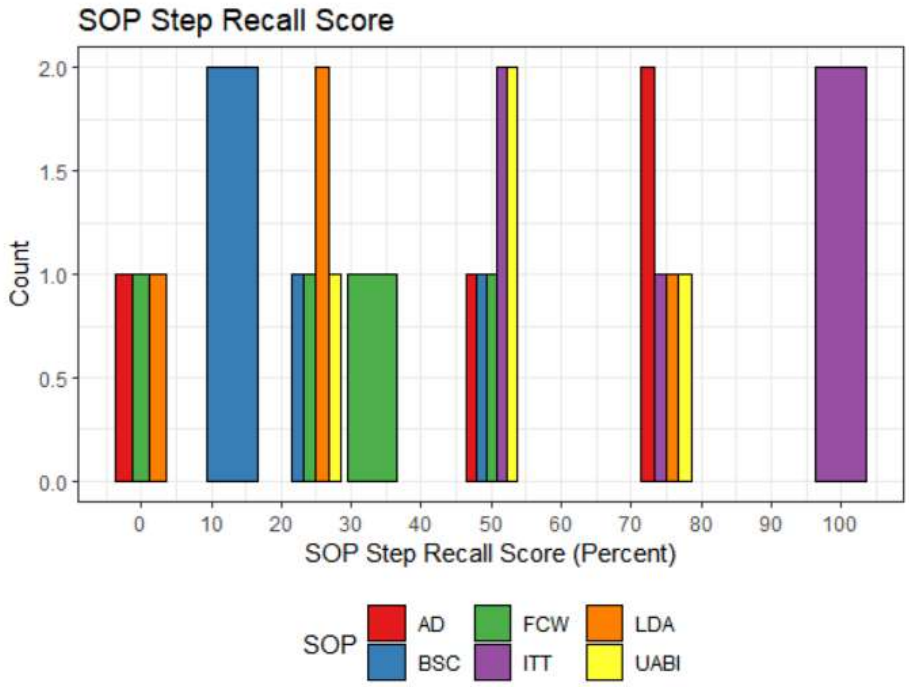



**Figure 51 SOP Step Recall by SOP**

### *4.3.3.4 Estimating Training Requirements*

The Total Number of Repetitions (TNR) and the Number of Days (ND) were calculated for each procedure. TNR ranged from 9 reps to 18 reps, with ND ranging between 5.3 days to 6.9 days. Table 27 summarizes estimated training metrics for the sample of SOPs.

**Table 27 Training Estimates - Case 3**

| SOP | Num Steps | Average SOP Recall Score | Total Number of Repetitions (TNR) | Number of Days (ND) |
|---|---|---|---|---|
| Forward Collision Warning | 4 | 27% | 9 Reps | 5.3 Days |
| Lane Departure Avoidance | 4 | 31% | 10 Reps | 5.4 Days |
| Blind Spot Collision | 4 | 25% | 9 Reps | 5.3 Days |
| Autosteer Disable | 4 | 50% | 11 Reps | 5.8 Days |
| Intersection Turn Takeover | 5 | 80% | 18 Reps | 6.9 Days |
| Unexpected Autopilot Behavior Intervention | 4 | 50% | 11 Reps | 5.8 Days |

#### *4.3.3.5 Simulations of SOPs*

The calculated AOTW for this case study was assumed to be for highway streets and is given in Equation 29. The computation is based on (1) lead vehicle speed, (2) lead vehicle deceleration rate, (3) following vehicle speed, (4) following vehicle deceleration rate, (5) human reaction time, and (6) car length. The distribution for each parameter is displayed in Table 28.

**Table 28 Parameters and Values for Calculating AOTW**

| Parameter | Assumed Distribution |
|---|---|
| Lead Vehicle Speed | Triang(50, 55, 60) mph |
| Lead Vehicle Deceleration Rate | Triang(0.238, 0.442, 1.09) g's [63] |
| Following Vehicle Speed | Triang(50, 55, 60) mph |
| Following Vehicle Deceleration Rate | Triang(0.238, 0.442, 1.09) g's [63] |
| Human Reaction | Triang(0.004, 1.66, 6.889) s [63] |
| Car Length | Triang(12.62, 14.7, 17.5) ft |

The collision model takes the initial speed of the lead and follow vehicles in mph and the follow and lead deceleration and converts it to ft/sec and ft/sec^2 using Equation 20, Equation 23, Equation 21, and Equation 24, respectively. Then, the lead and follow vehicles' stopping times based on their velocities are computed using Equation 22 and Equation 25.

**Equation 20 Lead Vehicle Velocity in Ft/Sec**

$$vel_{lead}(fps) = lead\ vehicle\ speed\ (mph) * 1.46667$$

**Equation 21 Lead Vehicle Deceleration in Ft/Sec^2**

$$dec_{lead}(fps^2) = lead\ vehicle\ deceleration(g's) * 32.2$$

**Equation 22 Lead Vehicle Stopping Time in Seconds**

$$Stop\ time_{lead}(s) = \frac{vel_{lead}}{dec_{lead}}$$

**Equation 23 Following Car Velocity in Ft/Sec**

$$vel_{follow}(fps) = folow\ vehicle\ speed(mph) * 1.46667$$

**Equation 24 Following Car Deceleration in Ft/Sec^2**

$$dec_{follow}(fps^2) = follow\ vehicle\ deceleration(g's) * 32.2$$

**Equation 25 Following Car Stopping Time in Seconds**

$$stop\ time_{follow}(s) = \frac{vel_{follow}}{dec_{follow}} + reaction\ time_{follow}(s)$$

Then, the distance between the head of the following car and the tail of the lead car is computed using Equation 26, where separation is between 0.5 and 6 car lengths,

with 0.25 increments. Finally, the following distance is converted to time using Equation 27, and the stopping time for the following vehicle is computed using Equation 28.

**Equation 26 Following Distance between Head of Follow Car and Tail of Lead Car**

$$follow\ distance\ (ft) = car\ length(ft) * \ separtion\ (in\ car\ lengths)$$

**Equation 27 Following Distance in Seconds**

$$follow\ distance\ time\ (s) = \frac{follow\ distance}{vel_{follow}}$$

**Equation 28 Following Vehicle Stopping Time**

$$follow\ distance\ stop\ time\ (s) = stop\ time_{follow} - follow\ distance\ time$$

After running the model for 2500 iterations, the minimum, average, and maximum of the follow distance stop time made up the distribution of the Allowable Operational Time Window (AOTW) that an operator has to intervene within before two vehicles collide.

**Equation 29 Calculated AOTW - Case 3**

$$AOTW = Triang(2.04, 7.82, 16.51)\ s$$

The six procedures were simulated for 10,000 runs. The PFtC for each SOP resulting from the simulation is reported in Table 29. The Forward Collision Warning SOP, the Blind Spot Collision SOP, and the Intersection Turn Takeover SOP had the highest Probability of Failure to Complete (PFtC). This is because of the cues associated with each step. The Forward Collision Warning SOP had two steps that had ambiguous cues and two steps that had cues lost in the clutter (not salient). The Autosteer Disable SOP had a notable decrease in the PFtC. This is because all the steps were unambiguous. The count of SOP steps assigned to each cue type is displayed in Figure 52.

**Table 29 PFtC for Simulated SOPs - Case 3**

| SOP | PFtC | Cue Characteristics |
|---|---|---|
| Forward Collision Warning | 41.51% | In FoV, Salient, Ambiguous: 2 Steps<br>In FoV, Not Salient: 2 Steps |
| Lane Departure Avoidance | 10.55% | In FoV, Salient, Unambiguous: 4 Steps<br>In FoV, Salient, Ambiguous: 1 Step |
| Blind Spot Collision | 63.47% | In FoV, Salient, Ambiguous: 6 Steps |
| Autosteer Disable | 2.69% | In FoV, Salient, Unambiguous: 4 Steps |
| Intersection Turn Takeover | 41.6% | In FoV, Salient, Unambiguous: 3 Steps<br>No Cue (Long Term Memory): 1 Step |
| Unexpected Autopilot Behavior Intervention | 16.45% | In FoV, Salient, Unambiguous: 3 Steps<br>In FoV, Salient, Ambiguous: 2 Steps |

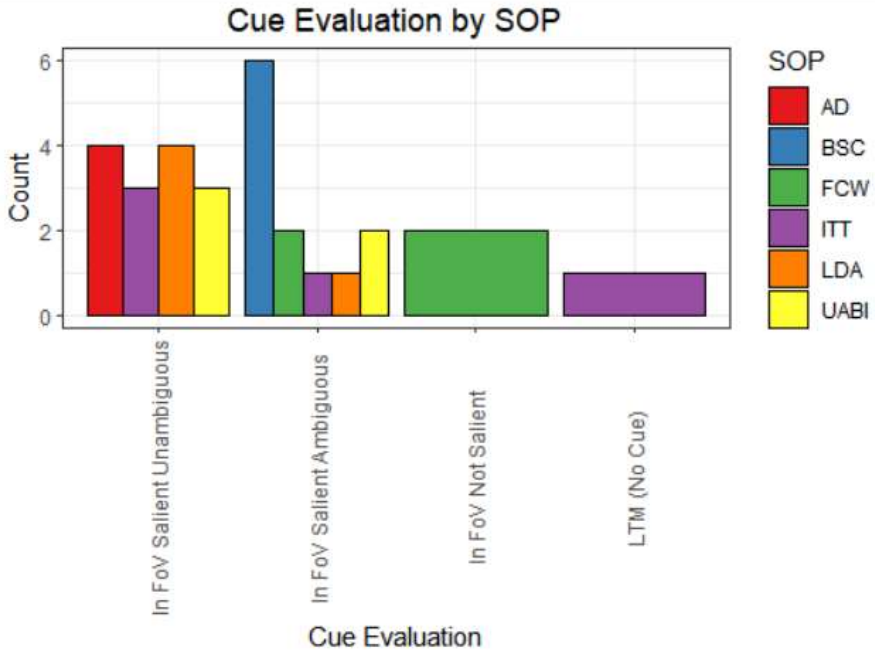



**Figure 52 Cue Evaluation by SOP – Case 3**

The simulation results are shown as ToP and AOTW distributions (Figure 53) and AOTW vs ToP scatterplots (Figure 54). Any intersection between the ToP and AOTW distribution is undesirable. Note that this phenomenon only occurs when the lead car has a sudden stop, which is a low probability.

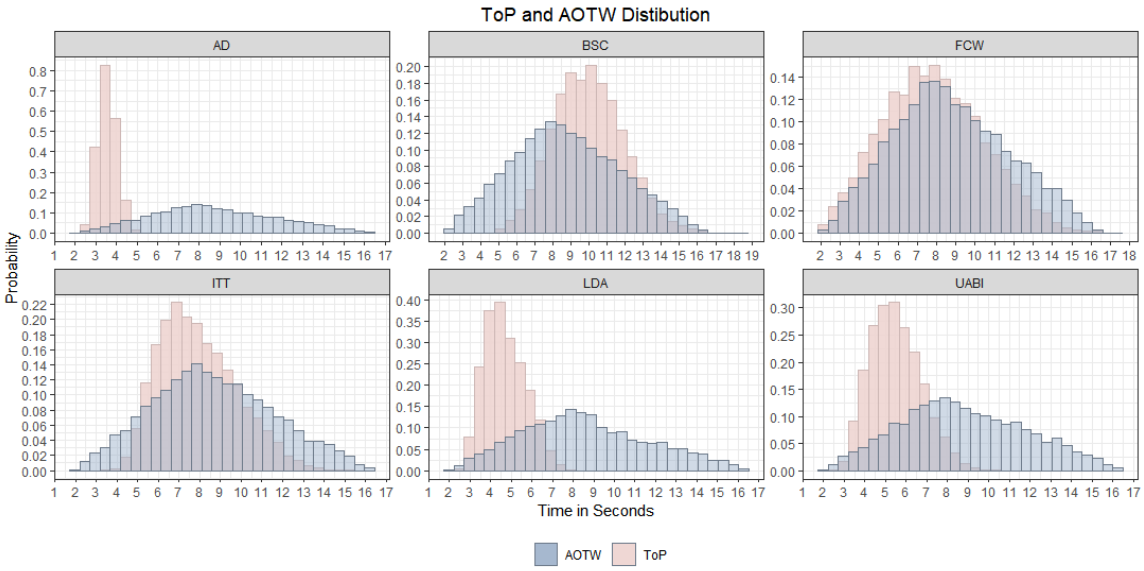



**Figure 53 ToP & AOTW Distributions – Case 3**

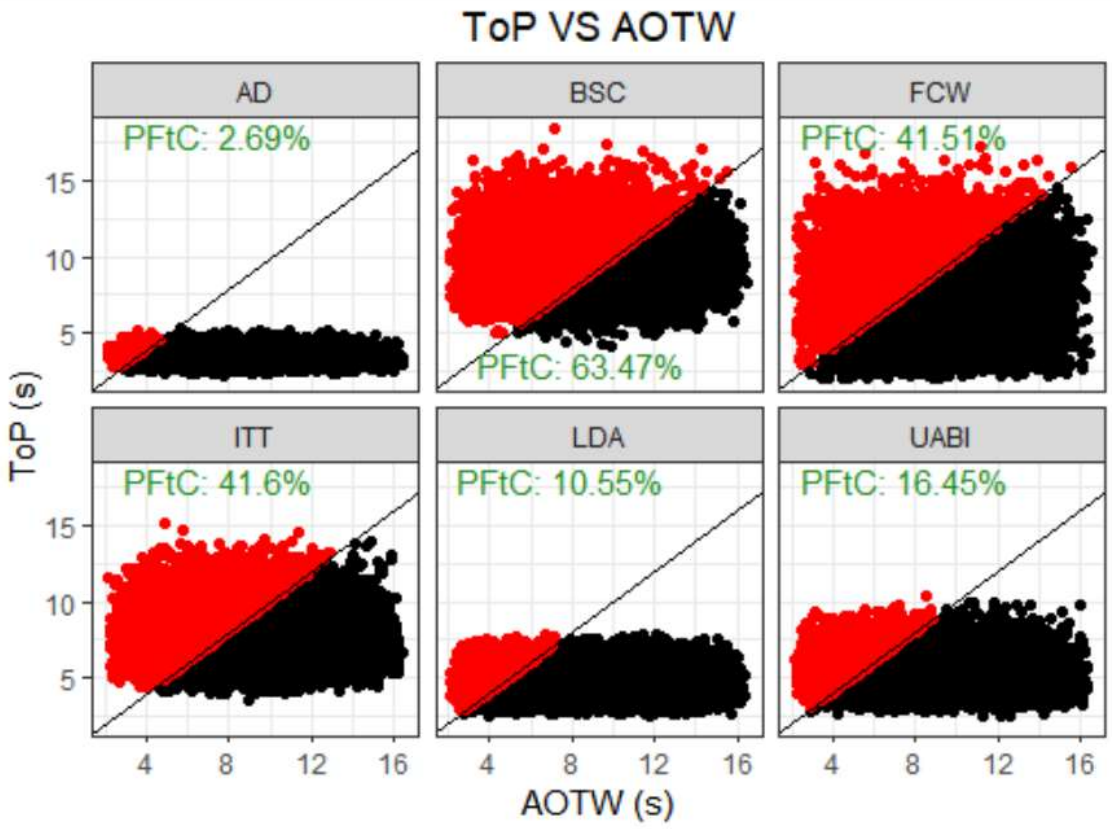



**Figure 54 ToP vs. AOTW - Case 3**

Problematic steps were identified for each SOP, along with its cue evaluation and why it was evaluated that way (Table 30). For example, ambiguous chimes were used for multiple steps, and ambiguous icons (i.e., gray or blue steering wheel and speed icon) were used to communicate engaged/disengaged systems.

The count of SOP steps falling into each cue category (Figure 55) shows that all steps in the Blind Spot Collision (BSC) SOP were triggered by ambiguous cues. The Forward Collision Warning (FCW) SOP had two steps that were triggered by cues that were not salient and two steps that were triggered by ambiguous cues. The Lane Departure Avoidance (LDA) SOP had four steps that were triggered by unambiguous cues and one step that was triggered by an ambiguous cue.

**Table 30 Problematic Steps - Case 3**

| SOP Step | Cue Evaluation | Reasoning |
|---|---|---|
| FCW 1. When you hear a loud chime and an obstacle is highlighted in red on the touchscreen, place your foot hovering over the brake pedal to prepare for braking. | In FoV, Salient Ambiguous | The cue "loud chime" was a cue for multiple steps; therefore, it has an ambiguous meaning. The chime could be a result of an imminent frontal crash or when you exceed a certain speed.<br>An obstacle highlighted in red does not send a direct message that the vehicle is about to crash. The operator must rely on other cues, i.e., what they can see outside, to determine that an imminent frontal crash is about to occur. |
| BSC 3. Verify that TACC has been disabled by visually glancing at the display and confirming the speed icon turns gray. | In FoV, Salient Ambiguous | The connection between Traffic Aware Cruise Control (TACC) and the speed icon is not clear. Color coordination (i.e., gray = off, blue = on) is also ambiguous. |
| BSC 4. When all autopilot systems have been disabled, accelerate or brake manually using the pedals to avoid the obstacle. | In FoV, Salient Ambiguous | The cue to indicate that all autopilot systems have been disabled is a gray steering wheel icon. The connection between the cue and what it means is ambiguous, along with the color coordination. |
| FCW 2. If the chime becomes louder, then fully depress the brake pedal to slow down the vehicle. | In FoV, Not Salient | The chime can be lost if other sounds are present. |

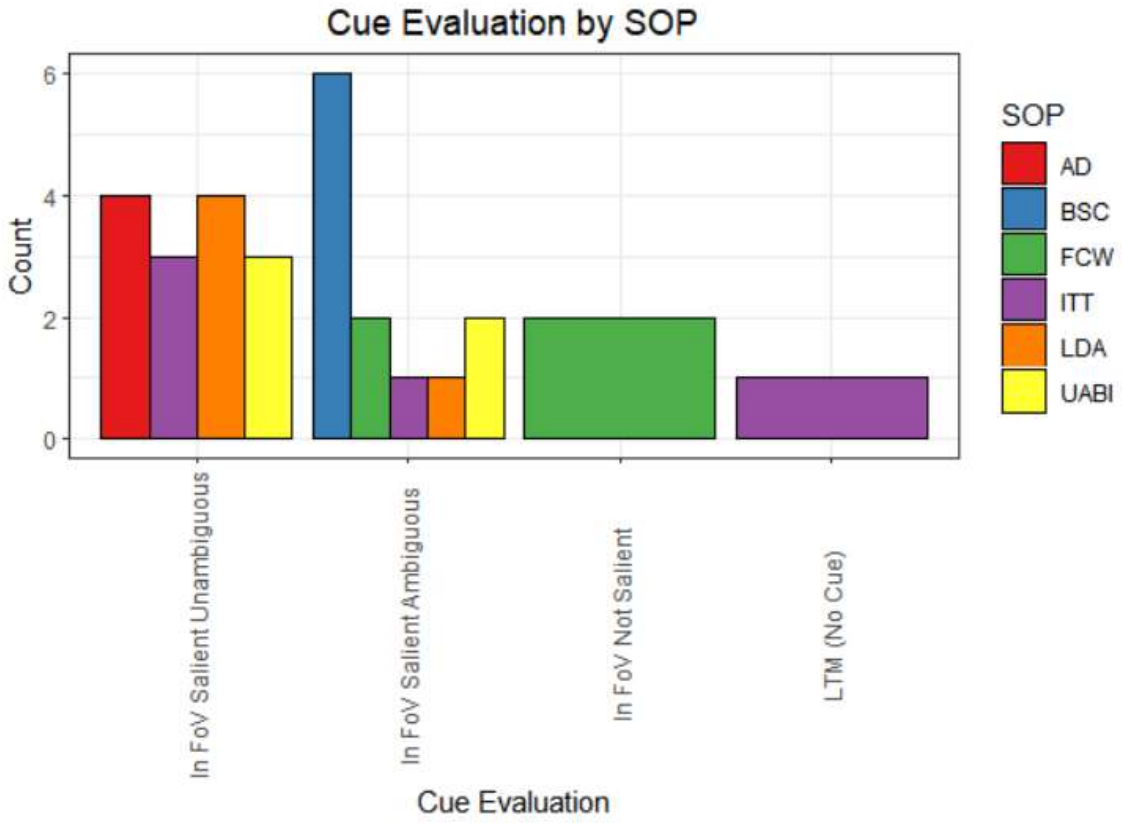



**Figure 55 Cue Evaluations by SOP**

The input device used for each action in each SOP is summarized in Figure 56. Seven SOP steps from five different SOPs included the same input device (drive stalk). The drive stalk is an input device that performs multiple functions, depending on how it is manipulated and for how long. The stalk drive can go in four different directions (upward, downward, towards the user, and away from the user) and can perform various functions if manipulated once or twice. For example, pulling the drive stalk twice down engages Autosteer, and pulling it once shifts into drive.

Multi-functional, moded devices, or devices that can perform multiple functions depending on which direction they are pulled/pushed, could lead to difficulties in an emergency. For example, the multi-functional moded device can be manipulated in the wrong way since the operator is accustomed to using it one way during nominal scenarios. Additionally, this leads to more memory demands on the operator to remember which function is done by pulling/pushing in which direction.

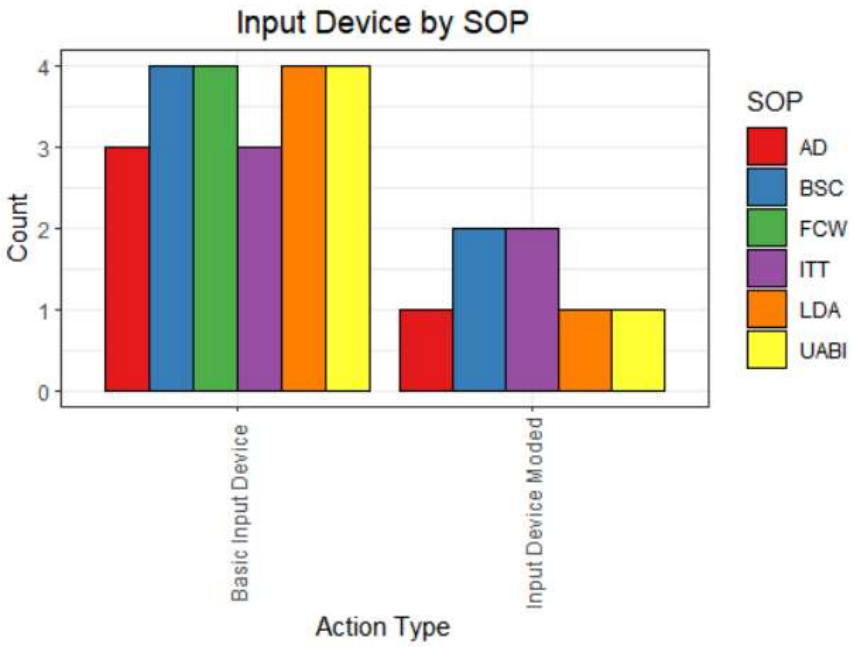



**Figure 56 Action Type by SOP**

### *4.3.3.6 MBSE Diagrams*

The MBSE Action Diagram representation of the Forward Collision SOP is displayed in Figure 57. The action diagram is a visualization of the SOP and can be integrated into the design of the vehicle during the design phase.

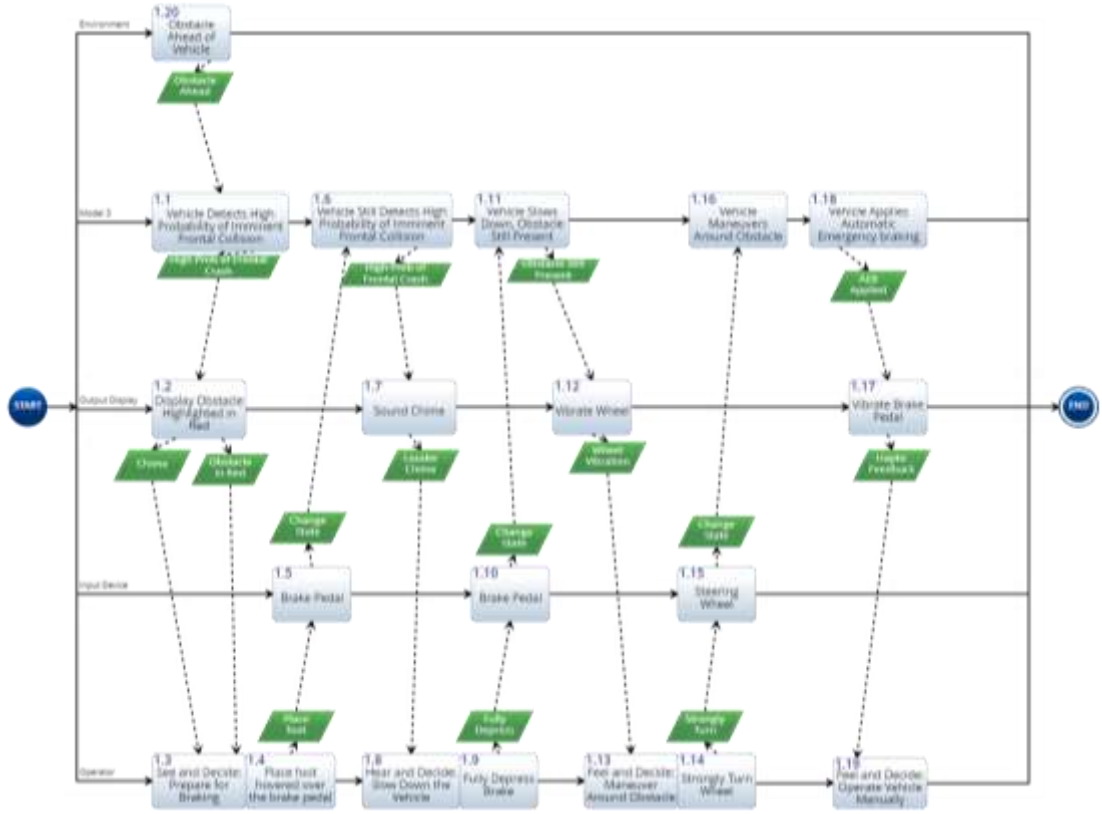



**Figure 57 Forward Collision SOP**

### 4.3.4 Time Analysis for the Process of Classification of SOP Step into Components: Manual Vs. Chatbot

To examine the potential time savings of using an LLM-based chatbot to automatically classify SOP steps into components, a time study was conducted with four participants and six procedures. The participants were graduate students in the Systems Engineering department. All participants have experience driving a non-autonomous vehicle and were exposed to e-PRL and classification equally through a period of two years. The approach was to time the participants while they were manually classifying the steps into components in an Excel sheet, as well as time them while prompting the

chatbot to do the classification and do a manual check. Finally, the total time for both classification methods was compared. The order of presentation of the procedures to the participants was randomized; however, all participants were asked to do the manual classification first, then the chatbot classification.

The chatbot SOP classification methodology involves importing the SOP into the chatbot, performing the classification by prompting with a specific prompt, copying and pasting the classifications into an empty Excel sheet, and manually checking the chatbot's classification. The manual process involves creating the appropriate headers in a blank Excel sheet and then manually populating the cells with the proper text.

Results show that using the chatbot reduced total SOP classification time by an average of 36.63% across the three SOP types and subjects. Individual subject time savings ranged from 25% to 65%. Table 31 displays the results of time savings for each participant in each SOP.

**Table 31 Time Reduction by SOP for Each Participant**

| SOP | Time Decrease |
|---|---|
| Forward Collision | 59.95% |
| | -14.78% |
| | 44.17% |
| | 16.86% |
| Lane Departure | 33.62% |
| | 25.52% |
| | 61.63% |
| | 39.69% |
| Side Collision | 39.69% |
| | 36.42% |
| | 64.98% |
| | 31.81% |

The normality of the data was first tested to test the significance of the time reduction. A K-S test was conducted, and the p-value was greater than 0.05, failing to reject the hypothesis of normality. Therefore, the assumption of normality holds in the data, and a parametric test was conducted.

The Manual group had higher values (M = 7.65, SD = 2.81) than the Claude group (M = 4.78, SD = 2.17). The null hypothesis that the Manual times had a smaller or equal value compared to the Claude timings was tested using a one-tailed t-test for paired samples. The result was statistically significant ($t(11) = 4.75$, $p = <.001$. The null hypothesis was thus rejected. Therefore, the time reduction was statistically significant between the two groups.

### 4.3.5 Discussion

Results revealed that most SOP steps were Action-Only (80%), with fewer Decisions (4%) or Actions with Waiting & Verification (16%).

Action-only steps had the highest recall score (50% vs 33% for Decision-Action and 19% for Action with Waiting & Verification). Across the six SOPs, estimated total training repetitions ranged from 9 reps to 18 reps, demanding 5-7 training days to reach proficiency. The Intersection Turn Takeover SOP notably had the highest recall score (80%) and required the most repetitions (18 over seven days) due to increased step complexity.

Problematic SOP steps were identified based on cue evaluations, considering factors like cue salience and ambiguity. Having cues in the FoV, Salient, and Unambiguous cues could further improve ToP times and reduce collisions through designing effective perceptual (i.e., visual, aural, haptic) triggers. Additionally, moded devices seem to be used frequently. For example, the drive stalk can be pushed up (once or twice quickly) or pulled down (once or twice quickly). Pushing the stalk up and holding it for one to two seconds can put the car in neutral. These devices can be problematic, especially during human takeover when the operator must complete several critical steps.

Finally, a time analysis comparing the time for manual classification of three SOPs across four participants shows the potential of using chatbots to achieve time and cost savings in SOP analysis. While the results show an average time reduction of 37% when using the chatbot, the results may be biased. This is because the participants were asked to do the manual classification first. This could potentially reduce the time for doing the “manual check” after the chatbot generates the classifications since the participants know what the correct classification is. An alternative approach to minimize

this bias is to generate a set of new SOPs and randomize the order in which the participants are asked to classify the SOP. i.e., half the participants do the manual classification first, and the other half do the chatbot classification first.

### 4.3.6 Summary

This case study examined SOPs for semi-autonomous vehicles to estimate training requirements and evaluate procedure quality. Six SOPs for critical human driver takeover events were generated and analyzed, including forward collision, lane departure, unexpected autopilot behavior, autosteer disable, intersection turn takeover, and blind spot collision.

An LLM-based chatbot generated the set of procedures from the Tesla Model 3 Owner's Manual. The steps were refined into a trigger-action format. The LLM-based chatbot was then trained through multiple iterations until 80% of the SOP step components were correctly classified by the chatbot. The trained chatbot was able to aid with the classification of SOP steps into SOP step components. A time study compared chatbot-assisted classifications to manual classifications. Using the chatbot, participants were able to reduce the time for SOP step component classification by 37%.

Among the twenty-five (25) SOP steps analyzed, 80% were Action-Only steps, 16% were Action with Waiting & Verification, and 4% were Decision-Action steps. Action-only steps had the highest average SOP recall score of 50%, indicating the highest memory demand. Estimated total training repetitions spanned from 9 to 18, requiring 5-7 training days for each SOP.

#### *4.3.6.1 Findings*

Based on the results and discussion, this analysis recommends the following:

1) Improve user-interface design:
    a) Ensure cues are in the FoV, salient, and unambiguous such that the PFtC can be lowered.
        i) Color code modes in a more intuitive way (engaged = green, disabled = gray).
        ii) Use distinct sounds for different issues instead of a chime for every failure.
        iii) Provide more detailed messages as to what the operator must do. For example, "brake now" instead of a highlighted object in red, or "Blind Spot Obstacle Detected" instead of a highlighted object.
    b) Moded devices are problematic when the same input device performs four different functions depending on the direction in which it is pulled/pushed in and for how long it is held. This creates the need for the operator to recall which direction performs which function. During emergencies, the incorrect function may be performed due to this.

Recommendations:

1) Augmenting the design phase of future vehicles with an MBSE model of SOPs can aid in identifying elusive cues, input & output displays, and SOP step sequences.
2) State DMVs should require additional training explicitly focused on human handover requests. SOPs analyzed in this case study had a high PFtC, further

confirming that extra attention should be placed on SOPs and input/output devices.

3) Policymakers should mandate vehicle manufacturers to define clear written SOPs that could be analyzed and simulated for human performance. SOPs should have a PFtC less than a safety threshold to ensure that:
    a. Manufacturers have thought of every possible human takeover scenario and generated the appropriate human response.
    b. Generated responses in the SOPs can be completed in time in the event of a hazardous event occurs.

# CHAPTER FIVE: CONCLUSIONS

Standard Operating Procedures (SOPs) are the glue that holds the operator and automation together in complex human-machine systems. While the automation is certified and the operator is licensed, SOPs are loosely regulated. Analyzing the current SOP development and evaluation process revealed that it needs more rigorous analysis and testing methods.

Testing is based on Subject Matter Expert (SME) inspections and an informal testing feedback process through training. These testing methods do not consider the human operator's or operational environment's variabilities. Additionally, any type of testing is qualitative; thus, the process lacks quantitative measures.

One contributing factor that prevents more rigorous vetting of SOPs is that SOPs reside as text in static/flat files, with no underlying structure to the SOP step that can be executed or analyzed.

To address that gap, this dissertation developed methods to (1) generate SOP steps in a digital form, (2) classify SOP steps into SOP step components in a digital form, (3) analyze SOP content, (4) simulate SOP for human performance, and (5) generate MBSE models to integrate within the models used for design and testing of the machine. This five-step process hinges on digitizing the SOP steps to analyze the SOP.

The digitization of the SOPs steps requires the definition of a canonical structure. This structure follows the trigger-decide-act cycle that follows human cognition steps. The components of each SOP step include a perceptual trigger, cognitive decision, motor

action, perceptual waiting/timing, and perceptual verification after waiting. Perceptual, cognitive, and motor elements further define these components to outline the SOP step canonical structure.

Extended-PRL (e-PRL) was developed as an extension of the Procedure Representation Language (PRL) that was developed by Traclabs for the International Space Station using the SOP step canonical structure. e-PRL includes entities such as the operator responsible for the SOP step, conditions and data for initiating the step, decisions to be made, data required for those decisions, actions for the step, needed physical motion, and input devices used. Some steps may require waiting and verification. In such cases, additional entities are needed, including waiting requirements, data for fulfilling those requirements, sources of data, verification actions, data required for verification, and sources of that data. The canonical structure defines the necessary components and entities for digitizing SOPs as executable models, allowing for effective operator-machine interaction and reflecting human cognition steps.

The first step in the process is to generate SOP steps from an owner's manual. An owner's manual is a document the Original Equipment Manufacturer (OEM) provides as a reference guide for safe operation. Extracting step-by-step procedures from the manual requires much time and understanding of the equipment. A method for using an LLM-based chatbot was developed to extract SOP Steps from the text in the owner's manual.

The chatbot's import button is used to import the owner's manual. The chatbot is then prompted to write a procedure for operator intervention during Full Self-Drive usage. However, some of the generated steps are not trigger-action steps. To correct this,

the chatbot is prompted to understand that a step must contain a trigger and an action. The chatbot produces an improved SOP that includes triggering conditions and the correct course of action. The steps are further modified manually to remove non-actionable items and break compound steps. This ensures that each SOP step has a clear trigger and action.

Once an SOP is generated or received from the OEM, the SOP is converted into digital representation (e-PRL) through a manual process or an LLM-based Chatbot method developed in this dissertation. The manual method involves following a flowchart to decompose the step into components and assign each component an appropriate e-PRL label. This process, known as SOP step annotation, requires the entities to be defined. This analysis described the entities, called e-PRL elements, in section 3.1.2.1. Once the entities are defined, the SOPs can be uploaded to an annotation tool and annotated according to the logic developed in section 3.1.2.2.

The LLM-based Chatbot method developed in this dissertation involves the development of rules and data such that the chatbot can classify text in the SOP into e-PRL components. First, the chatbot needs to understand what e-PRL and its labels are. This is achieved by creating an internal database with rules for classification. For example, the rules specify that if the chunk of text contains “PF,” then that chunk of text would be labeled as “Actor.” The chatbot then learns these rules from the database. Next, the chatbot is asked to classify the text into e-PRL labels. A table with correct classifications is imported, and the chatbot generates a list of new rules by comparing

these with its classifications. These new rules are added to the database, and the process is repeated with more SOPs until 80% accuracy.

Once an SOP is in e-PRL format, the SOP can be analyzed. Metrics were developed to assess SOP quality (i.e., ease of use) and training time. This aids SOP designers in improving SOP development and training.

Categorizing SOP steps into different types, such as action-only, decision-action, action with waiting and verification, and decision-action with waiting and verification, provides insights into their complexity. Waiting requirements and verification are crucial for successful SOP execution. Steps with decisions should include necessary data and its source to facilitate timely and correct decision-making by operators.

Furthermore, the memory demands placed on operators can be measured using the SOP step recall score. Steps that require excessive memory may lead to errors and longer training times, necessitating modifications to reduce unnecessary workload. Estimating total repetitions and training days helps plan SOP instructional programs accordingly. SOP statistics can be categorized into two types: statistics on the type of SOP steps and statistics on the content of SOP steps.

The SOP step types are derived from the SOP canonical structure. There are four types: Action Only Steps, Decision Steps, Action with Waiting & Verification Steps, and Decision with Waiting & Verification Steps. The logic for categorizing each step was defined in section 3.1.3.1. Based on the type of the SOP step, the SOP step recall score is computed. The SOP Step recall score measures the operator's ability to recall information during training for a step. It is based on explicit or implicit e-PRL elements in the SOP

step. Explicit elements are clearly stated in the SOP, while implicit elements are implied through the operator's knowledge. There are three types of implicit information: sequential, instantaneous, and system description. The implied system description elements contribute to the operator's recall score. The equation used to calculate the recall score was provided in section 3.1.3.2.

Matessa and Polson (2006) studied estimating the number of repetitions required for a list of actions. They provided a model and table to showcase the number of repetitions needed each day based on the length of the list. The data was synthesized from Ebbinghaus experiments but only went up to a list length of 24. A regression analysis was conducted to account for list lengths not in the data, revealing a linear relationship between repetitions required and the number of steps in the procedure. Additionally, a logarithmic relationship was found between the number of days required to reach proficiency and the total number of repetitions.

To simulate the SOP for human performance and operational efficiency assessment, a Computer-Aided Design of SOPS (CADSOPs) was developed. CADSOPs is an Excel-based user interface that aids in creating, storing, and simulating SOPs. The CADSOPs interface allows users to input SOP steps individually and then export them to a new sheet or an existing tab based on the procedure name. After inputting all SOP steps, users can assess the cue type and frequency on the current sheet by selecting options from a drop list. Users can also import the appropriate time distribution based on cue type and frequency. The resulting time distributions are printed on the input sheet for verification. Users can then simulate the SOP by specifying the AOTW and number of

iterations. This simulation process is illustrated in a flowchart and involves generating random variables for each step using Excel's RND() function and applying the inverse transformation method.

The performance of the SOP is measured by the Time on Procedure (ToP), which can vary due to factors such as operator expertise, attention, fatigue, and other human factors (see [21], [55]). The Allowable Operational Time Window (AOTW) is the time the procedure must be completed and can also vary due to aircraft performance and the operational environment. When the ToP exceeds the AOTW, the SOP has failed to meet its goal. The Procedure Buffer Time (PBT) is the difference between the AOTW and ToP for each SOP performance and is represented as a distribution. The left tail of the PBT distribution represents the Probability of Failure to Complete (PFtC), with a longer left tail indicating a more unreliable SOP. Figures illustrate the dynamic metrics of the SOP, including the ToP distribution, AOTW distribution, PBT distribution, and PFtC.

Finally, the SOP can be visualized in an MBSE environment as an SOP Action Diagram. SOP Action Diagram consists of actions, branches, and inputs/outputs (I/Os). Actors perform actions depicted by squares. The branches represent actors, and each actor performs the actions on its branch. An action can be triggered by a preceding action done by the same actor or by an action from another actor using I/Os. Rules and mapping were provided to convert e-PRL representations of SOPs into SysML Action Diagrams.

Three case studies were presented to demonstrate the applicability of the method in general and e-PRL as a model of SOPs. Analysis of a sample of SOPs in each domain (aviation operations, International Space Station (ISS) Habitable Airlock (HAL), and a

semi-autonomous vehicle) provided analytical insights into step complexity, training requirements, problematic cues, input devices, and overall procedure completion reliability. Applications of this method are (1) improving operational safety, (2) predicting training time, (3) process improvement, and (4) developing regulatory standards of new autonomous systems.

### 5.1 SOP Analysis for Operational Efficiency

The methodology facilitates comprehensive and continual evaluation of SOP quality using data-driven models early in the design phase of the human-machine system. By integrating models of the SOP with models of the machine, manufacturers can address usability issues in the user interface during the design phase before implementing the design. For example, the vehicle manufacturer may address an issue with an ambiguous chime by replacing it with an aural message pointing the operator to the issue. Note that the same chime is used to indicate at least 4 different events.

The simulation of human performance against operational time constraints can expose specific steps with a high probability of failure. This occurs when cues meant to trigger steps need to be more clear, salient, or completely missing. By pinpointing such problematic steps and evaluating their perceptual triggers, user interface design can be refined appropriately. For instance, alerts can be made more visually or aurally distinct, the timing and content of messages can be optimized, and modes can be designed to minimize confusion. That way, the burden is taken off the human operator, who typically must compensate for any design issues by demanding more memory.

### 5.2 SOP Analysis for Training Time Predictions

This method also facilitates a data-driven way to forecast the repetitions and days needed for trainees to internalize procedures fully. Predictions can closely reflect the realities of the instructional process by accounting for the impact of recall demands in steps through scores derived from the SOP analysis. Such insights enable better planning of logistical aspects around simulators, instructors, and schedules. Further, the total repetitions can inform the optimal sequencing of procedures to balance workload and pace progress effectively. For instance, shorter procedures with fewer memory elements can be scheduled between more complex, lengthy ones. Training duration is thus optimized to meet proficiency milestones.

### 5.3 SOP Analysis for Process Improvement

This method can also be used alongside existing workflows to reduce costs, enhance training, and monitor deployed SOPs more effectively. Organizations can use this method to analyze their current processes and evaluate them against time limits to see whether a procedure can be completed in time when needed. Organizations can also identify waste through this method, i.e., processes that are taking too long, further reducing the cost of operations.

A great example of this is hospital operations. Many emergencies can come up in a hospital that requires hospital personnel to react and act in a timely manner. For example, a code pink is initiated in the event of a child's abduction at the hospital. While hospital staff are trained on what to do in the case of a code pink, these scenarios rarely occur. This means that once the code pink is initiated, it is likely that staff will need more

time than expected to perform this procedure. Applying this method to such SOPs will aid in assessing how feasible each step of the SOP is and how long it would take in the event of an abduction.

Additionally, nurses in each hospital have a set of SOPs that are performed when a patient is admitted. Assessing these SOPs would catch inefficiencies in the process. For example, if patients were asked for their medical history multiple times during the intake process, as different nurses were not entering the patient's medical history in a logbook. By assessing the SOP and adding a step for completing the patient's medical history in the logbook, the process would take less time, and the quality of the intake process would increase.

### 5.4 SOP Analysis for Regulatory Approval

The responsibility of the government to its citizens is to protect them. The aviation industry has a well-established regulatory process to ensure the safe design and operation of aircraft.

In contrast, the semi-autonomous car industry is still in its infancy and has fewer (or very few) regulations. The design of these vehicles burdens the human operator with intervention whenever the automation does something unexpected. While autonomous driving systems have the potential to improve road safety significantly, they are a double-edged sword. The National Highway Traffic Safety Administration (NHTSA) published that as of January 16, 2024, Tesla vehicles have reported 1,084 accidents since the NHTSA issued the Standing General Order (SGO), which requires all vehicle

manufacturers to report crashes involving vehicles equipped with level 2 advanced driver assistance systems.

The autonomous vehicle space presents an opportunity to implement new and innovative approaches to testing SOPs that can ensure the human operator intervenes in a timely manner. The analysis presented in this dissertation showed that the Tesla Model 3 SOPs were buried in the text within the owner's manual. Autonomous vehicle manufacturers would be more inclined to analyze and test SOPs if regulations existed to require testing, and this method contributes formal means to enforce such requirements. Further, the instructions in the form of SOPs can be used for training and licensure of drivers.

Ultimately, as automation becomes further embedded across vehicles, airliners, spacecraft, and beyond, the role of SOPs grows in guiding safe and efficient human intervention. Without new testing methods, getting the desired safety level could take decades.

# APPENDIX A

## ANALYSIS OF AVIATION ACCIDENTS

The accidents were categorized using the following rules:

1. Insufficient AOTW: Simulated ToP was in excess of AOTW when the SOP was performed in an efficient manner, and the AOTW occurred in less than the 33%-tile.
2. Excessive ToP: Accident AOTW was more than 34% of the simulated ToP.
    a. Excessive ToP - Ambiguous SOP Triggers – SOP Triggering Event is based on External Cue: according to the accident report, the triggering cue was ambiguous. An example is the presence of turbulence due to an onset of aerodynamic stall that could be confused with natural/clear air turbulence prior to 1.05 VStall or Stick-shaker.
    b. Excessive ToP - Ambiguous SOP Triggers – No/Weak Association between Cockpit Alert and SOP Triggering Event: according to the accident reports, the triggering cue was not associated with the correct procedure.
    c. Excessive ToP - Unambiguous SOP Triggers but Ambiguous Cues for SOP Steps – Cue for SOP Step in FoV but Not Salient or Ambiguous Semantics: triggering cues to start the SOP were

unambiguous, but a trigger to a step in the procedure was not in the Field of View (FoV) of the operator.

d. Excessive ToP - Unambiguous SOP Triggers but Ambiguous Cues for SOP Steps – Cue for SOP Step is not present: triggering cues to start the SOP was unambiguous, but a trigger to a step in the procedure was based on Long Term Memory.

Analysis of twelve Part 121 modern airliner accident reports yielded the taxonomy of SOP issues summarized in Figure 58. SOP issues result in either an excessive ToP or an insufficient AOTW. Accidents with insufficient AOTW occur when there is no time to complete the SOP. It is a phenomenon simply dictated by environmental conditions, e.g., fire.

An excessive ToP, however, could be due to a couple of underlying reasons, either an ambiguous SOP trigger or memorization items in the SOP. Ambiguous SOP initiation triggers could be because the trigger relies on the external environment, such as Precision Approach Path Indicator (PAPI) lights or the visible natural horizon. Ambiguous SOP initiation triggers could also be due to a missing association between an alert in the cockpit and the appropriate procedure. The missing association could be either because of an ambiguous alert or a non-ambiguous alert but a non-attentive pilot. The other cause of excessive ToP is memorization items in the SOP due to missing steps requiring the use of Long-Term Memory (LTM) or steps with cues not in the Field of View (FoV) of the operator. Figure 58 depicts the hierarchical structure of the SOP issues found in the analysis.

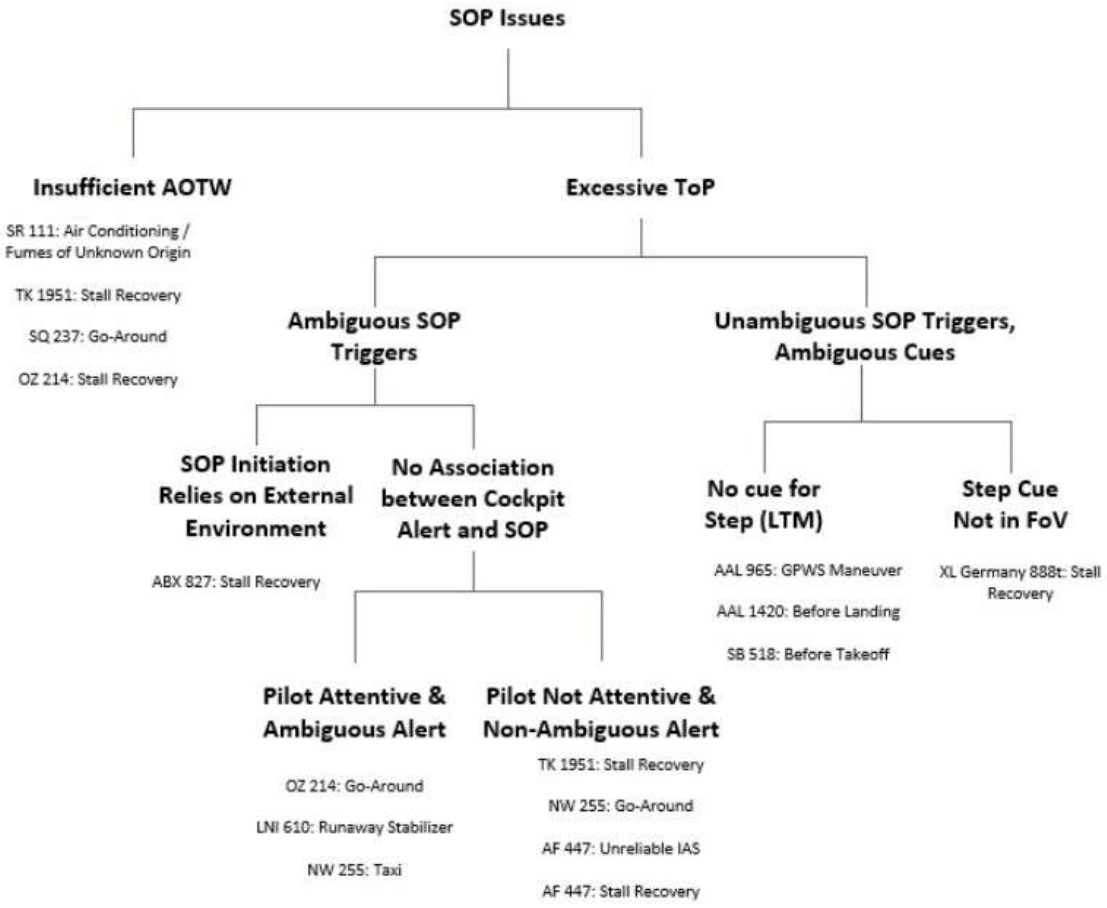



**Figure 58 Taxonomy of SOP Design Issues as Identified from Human Operator Intervention Tasks in Accident**

# APPENDIX B

## LIST OF TASKS FOR HUMAN PERFORMANCE DATA

### Daily Tasks:

1. Font Change (Word; 4) – Change

Initial Condition: A Word document with the Home Tab open and the word "Hello" written on the document in Calibri Size 11 Font
Task: "Please change the font of the word 'Hello' to Times New Roman."

2. Create Bullet Points (Word; 2)

Initial Condition: An empty Word Document on the Home Tab
Task: "Please create a bullet point with the filled circular dot."

3. Start Slide Show from the Beginning (PowerPoint; 1)

Initial Conditions: A PowerPoint presentation in the home tab with one slide filled with text and images.
Task: "Please start the presentation from the beginning."

### Frequent Tasks:

4. Copy text (Word; 5 – Ctrl + C, 1 – Right Click on Highlighted Text)

Initial Condition: A Word Document in the home tab with the word "Hello" written on it in Calibri Size 11 Font.
Task: "Please copy the word 'Hello' in a way that it can be pasted by the computer."

5. Highlight Cell (Excel; 3 – "Fill Cell")

Initial Condition: An Excel Sheet in the home tab with the word "1" written in a cell.
Task: "Please highlight the cell with the '1' in a different color."

6. Change the Layout of a Slide (PowerPoint; 2)

Initial Condition: A PowerPoint presentation in the home tab with a blank slide
Task: "Please change the layout of slide 1 to a 'Title and Content' Slide."

### Infrequent Tasks:

7. Alphabetical Sorting (Word; 4)

Initial Condition: A Word Document in the home tab that has the words "Dog," "Bat," "Ant," "Cat," and "Elephant," in that specific order.
Task: "Please sort the list in alphabetical order using Word's built-in feature."

8. Group Objects (PowerPoint; 1 – Inside "Arrange" Dropdown)

Initial Condition: A PowerPoint Presentation in the home tab with one slide. The slide will have two squares on it.
Task: “Please group the two squares together”

9. Insert Document on Computer (Excel; 5)

Initial Condition: An Excel Sheet in the home tab that is empty. An empty PDF document named “Blank PDF” is in an isolated folder on the computer.
Task: “Please embed the document ‘Hello’ into the Excel sheet.”

**Rare Tasks:**

10. Create a Unique Keyboard Shortcut (Word, 5)

Initial Condition: A Word Document in the home tab that is empty.
Task: “Please create a new keyboard shortcut to ‘Add a Digital Signature.’ Make the Shortcut Key ‘Ctrl + Shift + Y’”

11. Change the Character Spacing (PowerPoint, 2)

Initial Condition: A PowerPoint Presentation in the home tab with one slide. This slide will have “Hello” written on it.
Task: “Please change the character spacing of ‘Hello’ to loose.”

12. Turn off Wrap Text (Excel, 4)

**APPENDIX C**

**SOPS AND SOP STEPS GENERATED FOR TESLA MODEL 3**

# Forward Collision Warning Procedure

Triggering Condition: This is initiated when a vehicle detects a high probability of an imminent frontal collision with another object.

1. When you hear a loud chime and an obstacle is highlighted in red on the touchscreen, place your foot hovering over the brake pedal to prepare for braking.
2. If the chime becomes louder, then fully depress the brake pedal to slow down the vehicle.
3. If the steering wheel begins to vibrate, then turn the steering wheel strongly to maneuver around the obstacle if it is safe.
4. If Automatic Emergency Braking engages with brake pedal haptic feedback, then operate the vehicle manually until you re-engage Autopilot.

# Lane Departure Avoidance Procedure

Initiated when the vehicle drifts out of its driving lane towards another lane or road edge.

1. When the vehicle displays a red line highlighting the lane marking being crossed, and you hear a chime, then reduce pressure on the accelerator pedal to prepare for steering back into the lane.
2. If the drift continues, and the vehicle provides a steering wheel vibration, then firmly turn the steering wheel in the opposite direction to stay in lane.
3. If automatic corrective steering input is applied, shown on the touchscreen, continue monitoring that it prevents potential side collision.
4. If the vehicle continues drifting towards the lane markings or the edge of the road, push the drive stalk upward once to cancel autosteer. Verify that autosteer has been disabled by visually glancing at the display and confirming the steering wheel icon turns gray.

# Blind Spot Collision (Side Collision) Procedure

Initiated when the vehicle detects an object near the vehicle's blind spot/side while a lane change maneuver is being attempted.

1. When colored lines appear on the touchscreen indicating object proximity in the blind spot region, check side mirrors to confirm the presence of a detected object.
2. If the colored lines turn to red, indicating impact risk has increased, push the drive stalk upward once to disable autosteer. Verify that autosteer has been disabled by visually glancing at the display and confirming the steering wheel icon turns gray.
3. Once autosteer has been disabled, push the autopilot stalk away from you to disable Traffic Aware Cruise Control. Verify that TACC has been disabled by visually glancing at the display and confirming the speed icon turns gray.
4. When all autopilot systems have been disabled, accelerate or brake manually using the pedals to avoid the obstacle.

# Autosteer Disable Procedure

Initiated when vehicle disables "Autosteer" due to the operator ignoring hands-on torque reminders and/or encountering a situation where Autosteer is unable to steer accurately.

1. Once touchscreen displays take over the steering, message immediately, firmly grip the wheel to steer manually.
2. Take manual control of speed by pushing the brake/accelerator pedal.
3. When appropriate, based on road conditions, engage hazard lights.
4. Once pulled over, put the gear in Park to disable Autosteer completely before re-engaging it.

# Intersection Turn Takeover Procedure

Initiated when the vehicle approaches an intersection in Autosteer mode and needs to make a turn according to the navigation route.

1. When an upcoming turn is displayed on the navigation path, reduce cruising speed in anticipation of exiting from Autosteer.
2. As the intersection nears with no automatic turn path shown on the touchscreen, apply brakes to slow further and cancel Autosteer.
3. Activate the appropriate turn indicator.
4. Once at a slower, safe speed, manually steer the vehicle into the intended turn lane.
5. After fully completing the turn at an intersection, Autosteer can be re-enabled on the next straight driving segment by pulling the lever fully down twice

# Unexpected Autopilot Behavior Intervention Procedure

Initiated when the vehicle makes sudden inaccurate or erratic steering actions such as swerving or unexpectedly slowing down.

1. When the vehicle swerves erratically, or the steering seems inaccurate, keep a firm grip on the wheel and prepare to manually override.
2. If the vehicle engages in hazard lights, immediately press the brake pedal to regain control of the vehicle.
3. Take over steering by turning the wheel in the intended direction of travel to override Autosteer.
4. After manual override, disable Autosteer by pushing the lever fully upward once. Verify that autosteer has been disabled by visually glancing at the display and confirming the steering wheel icon turns gray.

# APPENDIX D

## HUMAN PERFORMANCE DATABASE

| Authors | Paper Title | Link | ToP | AOTW |
|---|---|---|---|---|
| Philip A. Oppenheimer | Pilot Response in Time Critical Aircraft Upset / Loss of Control Inflight (LOC-I) Events | https://apstraining.com/wp-content/uploads/2017/10/PilotResponseinTimeCriticalLOCI-Events.pdf | | 6-10 seconds for Loss of Control |
| | | | | 8–10 seconds under optimal circumstances and much longer under threat.” [4] This exceeds the 6-10 second critical window described above |
| Alexander J. Stimpson, Jason C. Ryan, and Mary L. Cummings | Assessing Pilot Workload in Single-Pilot Operations with Advanced Autonomy | https://hal.pratt.duke.edu/sites/hal.pratt.duke.edu/files/u10/hfes%202016%20final.pdf | Request before take-off check Lognormal(0.6931, 0.1). | |
| | | | Verify flight controls check completed Lognormal(2.0794, 0.1) | |

| | | | | |
|---|---|---|---|---|
| | | | Cabin check (communicate w/ crew) Lognormal(2.9957, 0.5) Respond to cabin check Lognormal(0.6931, 0.1) | |
| | | | Request flight controls check Lognormal(0.6931, 0.1) | |
| | | | Verify flight controls Lognormal(2.0794, 0.1) | |
| | | | Respond for verification Lognormal(0.6931, 0.1) | |
| Bartosz Binias, Dariusz Myszor, Henryk Palus, Krzysztof A. Cyran | Prediction of Pilot's Reaction Time Based on EEG Signals | https://www.frontiersin.org/articles/10.3389/fninf.2020.00006/full | Response to Unexpected Event 114 ms | |
| James Wilborn and John Foster | Defining Commercial Transport Loss-of-Control: A Quantitative Approach | https://arc.aiaa.org/doi/10.2514/6.2004-4811 | | The (LOC-I) situation progressed from an initial upset to a state of lost control in less than 10 seconds.” |
| FAA | | | the average reaction time for a cockpit crew to recognize emergencies is about 34 seconds | crews typically have about 88 seconds to respond to an emergency before it becomes critical. |
| Yiyuan Zheng, Yanyu Lu, Yuwen Jie, Zhiqiang Zhao, Shan Fu | Test Pilot and Airline Pilot Differences in Facing Unexpected Events | https://pubmed.ncbi.nlm.nih.gov/36757229/ | reaction time of test pilots (Mean = 23.38 s) was significantly shorter than airline pilots (Mean | |

| | | | | |
|---|---|---|---|---|
| | | | = 42.63 s) in Unreliable Airspeed condition | |
| CAA | Simulator-based Study of Emergencies Yields Insights into Pilot's Reaction Times | https://pubmed.ncbi.nlm.nih.gov/36757229/ | 2-4 seconds | Corrective action delay of 1 sec in cruise and 0.3 secs in other phases of flight |
| | | | 90th percentile 4.4 to 5.7 secs total reaction time. 2.07 to 2.95 secs detection time. Recommends design criteria of 6 seconds for total (assuming no distraction) and 3 secs for detection | |
| BEA | Serious incident to the Cessna Citation 525B registered F-HCIC on 22 July 2013 at 10 NM north west of Paris Orly airport climbing through FL 180 | https://bea.aero/uploads/tx_elydbrapports/f-ic130722.en.pdf | | |
| Chen et.al. 2023 | Where Failures May Occur in Automated Driving: A Fault Tree Analysis Approach | https://www.researchgate.net/publication/362539455_Where_Failures_May_Occur_in_Automated_Driving_A_Fault_Tree_Analysis_Approach | driver response times in 2 accident scenarios | 2 accident scenarios |
| Burian et.al, 2016 | Un-Alerted Smoke and Fire: Checklist Content | | Timely response by the flight deck and cabin | |

| | | | | |
|---|---|---|---|---|
| | and Intended Crew Response | | crews is essential (Federal Aviation Administration [FAA], 2014) | |
| FAA guidance for test flights in AC 25-7D, | Flight Test Guide for Certification of Transport Category Airplanes, and AC 25.1329-1C, | Barshi report | Approval of Flight Guidance Systems, require test pilots to delay initiation of response to flight control or flight guidance malfunctions to account for pilot recognition time and pilot reaction time. Often, recognition time is assumed to be 1 second, and reaction time is assumed to be 3 seconds. Thus, test pilots are told that "Recovery action should not be initiated until 3 seconds after the recognition point" (AC 25.1329-1C). | |
| | | | No studies were found that substantiate the FAA guidance concerning pilot recognition time and pilot reaction time. | |

| | | | | |
|---|---|---|---|---|
| DOT/FAA/AM-97/24; DOT/FAA/AM02/19; DOT/FAA/AM-05/23 | | | Several FAA studies with general aviation pilots demonstrate that these general aviation pilots may take many seconds, and in some cases many minutes, to recognize and respond to malfunctions | |
| Casner, Geven, & Williams, 2013 | The effectiveness of airline pilot training for abnormal events | https://pubmed.ncbi.nlm.nih.gov/23829023/ | substantially longer recognition times and reactions times, even in the case of expected events, than the times given in AC 25-7D and AC 25.1329-1C | |
| NTSB/AAR-14/01 | | | pilots may take a significantly longer time to recognize a malfunction and respond to it than the test flight 3 guidance suggests. For example, the NTSB states: “When a flight crew is confronted with a sudden, abnormal event, responses are more likely to be delayed or inappropriate.” | |

| | | | | |
|---|---|---|---|---|
| AC 25.1329-1A, Automatic Pilot Systems Approval, dated July 8, 1968 | | | The 3-second reaction time assumption dates back decades, to where the performance of the autopilot was constantly monitored by the crew in flight | |
| NTSB | Assumptions Used in the Safety Assessment Process and the Effects of Multiple Alerts and Indications on Pilot Performance | https://www.ntsb.gov/investigations/accidentreports/reports/asr1901.pdf | Pilot Recognition and Response in the Safety Assessment | |
| Burian et.al. 2017 | The Challenge of Aviation Emergency and Abnormal Situations | | | |
| Degani et.al, 2013 | Information Organization in the Airline Cockpit: Lessons From Flight 236 | https://journals.sagepub.com/doi/abs/10.1177/155534134129 83 | all relevant data to avoid the emergency were available to the aircraft computer systems, the design choices made about what to display and how to display it kept the pilots “in the dark.” | |
| Accidents | Airbus A321 operated by American Airlines as flight 1851 suffered a landing accident at Charlotte-Douglas | https://aviation-safety.net/database/record.php?id=20150815-0 | | The flight encountered a small microburst on short final about 7 seconds prior to the landing attempt characterized by wind shift from a headwind to a |

| | | | | |
|---|---|---|---|---|
| | International Airport, NC (CLT) | | | tailwind and back to a headwind. |
| Dixit VV, Chand S, Nair DJ | Autonomous Vehicles: Disengagements, Accidents and Reaction Times. PLOS ONE 11(12): e0168054. | https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0168054 | reaction time for AV disengagement for Mercedez-Benz cars and Google cars. Average for Mercedez-Benz was 0.83 seconds with SD 0.53 and average for Google was 0.84 with SD 0.90. | |
| Christian Roesener, Michael Harth, Hendrik Weber, Johanna Josten and Lutz Eckstein | **Modelling Human Driver Performance for Safety Assessment of Road Vehicle Automation** | https://ieeexplore-ieee-org.mutex.gmu.edu/document/8569669 | reaction time (defined as perception time and information processing time) fro braking was a gamma distribution with parameters 0.23, 1.5 seconds (estimated from fig 8) | |
| TyronLouw∗,GustavMarkkula,ErwinBoer,RuthMadigan,OliverCarsten,NatashaMerat | Coming back into the loop: Drivers' perceptual-motor performance in critical events after automated driving | https://www-sciencedirect-com.mutex.gmu.edu/science/article/pii/S0001457517302907 | take-over time (time when manual takeover is required and when driver disengages ADS) and action time (time when manual takeover is required and to when driver stated significant deceleration or wheel turning) for 5 conditions. Take-over time for base condition had mean of | |

| | | | | |
|---|---|---|---|---|
| | | | 3.7 s and 1.22 sd. time for action had mean of 4.95 s and sd of 0.91 | |

## REFERENCES


[1] A. Degani and E. L. Wiener, “Procedures in complex systems: the airline cockpit,” *IEEE Trans. Syst. Man Cybern. - Part Syst. Hum.*, vol. 27, no. 3, pp. 302–312, May 1997, doi: 10.1109/3468.568739.

[2] “In-Flight Fire Leading to Collision with Water Swissair Transport Limited McDonnell Douglas MD-11 HB-IWF Peggy’s Cove, Nova Scotia 5 nm SW 2 September 1998,” Transportation Safety Board of Canada, Aviation Investigation Report A98H0003, Mar. 2003.

[3] C. S. Burke, E. Salas, K. Wilson-Donnelly, and H. Priest, “How to turn a team of experts into an expert medical team: guidance from the aviation and military communities,” *BMJ Qual. Saf.*, vol. 13, no. suppl 1, pp. i96–i104, Oct. 2004, doi: 10.1136/qshc.2004.009829.

[4] V. Socha *et al.*, “Training of Pilots Using Flight Simulator and its Impact on Piloting Precision,” presented at the Transport Means 2016, Juodkrante, Lithuania, Oct. 2016.

[5] E. Salas, J. Held, and J. Weissmuller, “Performance Measurement in Simulation-Based Training: A Review and Best Practices,” in *Simulation in Aviation Training*, Routledge, 2002.

[6] I. Barshi, R. Mauro, A. Degani, and L. Loukopoulou, “Designing Flightdeck Procedures,” p. 81, Oct. 2016.

[7] G. Amare, “Reviewing the Values of a Standard Operating Procedure,” *Ethiop. J. Health Sci.*, vol. 22, no. 3, pp. 205–208, Nov. 2012.

[8] J. Bashatah and L. Sherry, “Lessons Learned From Human Operator Intervention for Ai Navigation and Flight Management Systems,” in *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, Apr. 2022, pp. 1–15. doi: 10.1109/ICNS54818.2022.9771518.

[9] J. Bashatah and L. Sherry, “Model-Based Analysis of Standard Operating Procedures’ Role in Abnormal and Emergency Events,” *INCOSE Int. Symp.*, 2022.

[10] “Uncontrolled Flight Into Terrain ABX Air (Airborne Express) Douglas DC-8-63, N827AX Narrows, Virginia December 22, 1996,” National Transportation Safety Board, AAR-97-05, Jul. 1997. Accessed: Sep. 27, 2021. [Online]. Available: https://www.ntsb.gov/investigations/Pages/DCA96MA054.aspx

[11] “Final Report on the Accident on 1st June 2009 to the Airbus A330-203 Registered F-GZCP Operated by Air France flight AF 447 Rio de Janeiro-Paris,” BEA, Aviation Investigation Report, Jul. 2012.

[12] “Northwest Airlines, INC. McDonnel Douglas DC-9-82, N312RC Detroit Metropolitan Wayne County Airport Romulus, Michigan August 16, 1987,” Aviation Investigation Report AAR-88/05, May 1988.
[13] “Aircraft Accident Investigation Report PT. Lion Mentari Airlines Boeing 737-8 (MAX); PK-LQP Tanjung Karawang, West Java Republic of Indonesia 29 October 2018,” Komite Nasional Keselamatan Transportasi, Aviation Investigation Report KNKT.18.10.35.04, 2019.
[14] “Controlled Flight into Terrain American Airlines Flight 965 Boeing 757-223, N651AA Near Cali, Columbia December 20, 1995,” Aeronautical Civil, Republic of Colombia, Aviation Investigation Report, Sep. 1996.
[15] “Runway Overrun During Landing American Airlines Flight 1420 McDonnell Douglas MD-82, N215AA Little Rock, Arkansas, June 1, 1999,” National Transportation Safety Board, Aviation Investigation Report AAR0102, Oct. 2001.
[16] “Accident on 27 November 2008 off the coast of Canet-Plage (66) to the Airbus A320-232 registered D-AXLA operated by XL Airways Germany,” BEA, 2010.
[17] “AC 120-71B - Standard Operating Procedures and Pilot Monitoring Duties for Flight Deck Crewmembers – Document Information.” Accessed: Sep. 16, 2022. [Online]. Available: https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentid/1030486
[18] “AC 120-92B - Safety Management Systems for Aviation Service Providers – Document Information.” Accessed: Sep. 16, 2022. [Online]. Available: https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentid/1026670
[19] “Safety Management System (SMS) | Federal Aviation Administration.” Accessed: Sep. 16, 2022. [Online]. Available: https://www.faa.gov/about/initiatives/sms
[20] H. K. Kourdali and L. Sherry, “Available Operational Time Window: A Method for Evaluating and Monitoring Airline Procedures,” *J. Cogn. Eng. Decis. Mak.*, vol. 11, no. 4, pp. 371–381, Dec. 2017, doi: 10.1177/1555343417727190.
[21] H. K. Kourdali and L. Sherry, “A Systems engineering method and simulation of standard operating procedures,” p. 9, 2016.
[22] “Arbortext | XML Content Management and Authoring | PTC.” Accessed: Sep. 16, 2022. [Online]. Available: https://www.ptc.com/en/products/arbortext
[23] “PRIDE,” TRACLabs. Accessed: Sep. 09, 2021. [Online]. Available: https://traclabs.com/projects/pride/
[24] T. Software, “Procedure Genius – Tietronix Software.” Accessed: Sep. 16, 2022. [Online]. Available: https://www.tietronix.com/procedure-genius/
[25] J. A. Bashatah, L. Sherry, S. Dam, L. Flenniken, P. Hartmann, and T. Harold, “Analyzing Standard Operating Procedures Using Model-Based Systems Engineering Diagrams,” *INCOSE Int. Symp.*, vol. 31, no. 1, pp. 1130–1144, 2021, doi: 10.1002/j.2334-5837.2021.00891.x.
[26] I.-C. Moon, K. M. Carley, and T. G. Kim, *Modeling and Simulating Command and Control*. in Springer Briefs in Computer Science. London: Springer, 2013. doi: 10.1007/978-1-4471-5037-4.

[27] J. M. Carroll *et al.*, *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*. San Francisco, UNITED STATES: Elsevier Science & Technology, 2003. Accessed: Sep. 20, 2022. [Online]. Available: http://ebookcentral.proquest.com/lib/gmu/detail.action?docID=294610
[28] S. K. Card, T. P. Moran, and A. Newell, *the psychology of human-computer interaction*. Hillsdale, N.J. : L. Erlbaum Associates, 1983.
[29] D. Kieras, "A Guide to GOMS Task Analysis," Jun. 1994.
[30] B. E. John and D. E. Kieras, "The GOMS family of user interface analysis techniques: comparison and contrast," *ACM Trans. Comput.-Hum. Interact.*, vol. 3, no. 4, pp. 320–351, Dec. 1996, doi: 10.1145/235833.236054.
[31] B. E. John and D. E. Kieras, "The GOMS Family of Analysis Techniques: Tools for Design and Evaluation,:," Defense Technical Information Center, Fort Belvoir, VA, Aug. 1994. doi: 10.21236/ADA309174.
[32] B. John, A. Vera, M. Matessa, M. Freed, and R. Remington, "Automating CPM-GOMS," presented at the Conference on Human Factors in Computing Systems, Minneapolis, Minnesota, Apr. 2002, p. 8.
[33] J. R. Anderson, *Rules of the Mind*. New York: Psychology Press, 2013. doi: 10.4324/9781315806938.
[34] A. Newell, *Unified Theories of Cognition*. USA: Harvard University Press, 1990.
[35] M. A. Freed, "Simulating human performance in complex, dynamic environments," Ph.D., Northwestern University, United States -- Illinois, 1998. Accessed: Jun. 07, 2021. [Online]. Available: http://www.proquest.com/docview/304449656/abstract/92B5C767E17F4D74PQ/1
[36] J. R. Anderson, *Language, memory, and thought.* in Language, memory, and thought. Oxford, England: Lawrence Erlbaum, 1976, pp. xiii, 546.
[37] J. R. Anderson, "The architecture of cognition.," *Archit. Cogn.*, pp. xi, 345–xi, 345, 1983.
[38] A. Newell, "PRODUCTION SYSTEMS: MODELS OF CONTROL STRUCTURES," in *Visual Information Processing*, W. G. CHASE, Ed., Academic Press, 1973, pp. 463–526. doi: https://doi.org/10.1016/B978-0-12-170150-5.50016-0.
[39] J. R. Anderson and C. Lebiere, *The atomic components of thought.* in The atomic components of thought. Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers, 1998, pp. xi, 490.
[40] M. D. Byrne, "ACT-R/PM and menu selection: applying a cognitive architecture to HCI," *Int J Hum Comput Stud*, vol. 55, pp. 41–84, 2001.
[41] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement.," *J. Exp. Psychol.*, vol. 47, pp. 381–391, 1954, doi: 10.1037/h0055392.
[42] J. Anderson, M. Matessa, and C. Lebiere, "ACT-R: A Theory of Higher Level Cognition and Its Relation to Visual Attention," *Hum.-Comput. Interact.*, vol. 12, pp. 439–462, Dec. 1997, doi: 10.1207/s15327051hci1204_5.
[43] J. E. Laird, *The Soar Cognitive Architecture*. The MIT Press, 2012.

[44] J. E. Laird, A. Newell, and P. S. Rosenbloom, “SOAR: An architecture for general intelligence,” *Artif. Intell.*, vol. 33, no. 1, pp. 1–64, Sep. 1987, doi: 10.1016/0004-3702(87)90050-6.
[45] M. A. Freed, M. Matessa, R. Remington, and A. Vera, “How Apex automates CPM-GOMS,” in *Proceedings of the Fifth International Conference on Cognitive Modeling*, 2003, pp. 93–98.
[46] A. J. Stimpson, J. C. Ryan, and M. L. Cummings, “Assessing Pilot Workload in Single-Pilot Operations with Advanced Autonomy,” *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 60, no. 1, pp. 675–679, Sep. 2016, doi: 10.1177/1541931213601155.
[47] M. I. Kurke, “Operational Sequence Diagrams in System Design,” *Hum. Factors*, vol. 3, no. 1, pp. 66–73, Mar. 1961, doi: 10.1177/001872086100300107.
[48] L. Sorensen, N. Stanton, and A. Banks, “Back to SA school: Contrasting three approaches to situation awareness in the cockpit,” *Theor. Issues Ergon. Sci.*, vol. 12, pp. 451–471, Nov. 2011, doi: 10.1080/1463922X.2010.491874.
[49] D. Harris, N. A. Stanton, and A. Starr, “Spot the difference: Operational event sequence diagrams as a formal method for work allocation in the development of single-pilot operations for commercial aircraft,” *Ergonomics*, vol. 58, no. 11, pp. 1773–1791, 2015, doi: 10.1080/00140139.2015.1044574.
[50] J. Huddlestone, R. Sears, and D. Harris, “The use of operational event sequence diagrams and work domain analysis techniques for the specification of the crewing configuration of a single-pilot commercial aircraft,” *Cogn. Technol. Work*, vol. 19, no. 2, pp. 289–302, Sep. 2017, doi: 10.1007/s10111-017-0423-5.
[51] H. Shan, P. Peng, L. Zhang, Z. Meng, and X. Shan, “Study using dynamic value stream mapping to optimize astronauts’ EVA processes and actions.”
[52] S. Cao, Y. Zhang, Y. Guo, and S. Chen, “Human Performance Time Model of Spacecraft Control Panel Operation in Simulated Microgravity,” *Aerosp. Med. Hum. Perform.*, vol. 88, no. 8, pp. 743–751, Aug. 2017, doi: 10.3357/AMHP.4668.2017.
[53] L. Sherry, M. Medina, M. Feary, and J. Otiker, “Automated tool for task analysis of NextGen automation,” in *2008 Integrated Communications, Navigation and Surveillance Conference*, May 2008, pp. 1–9. doi: 10.1109/ICNSURV.2008.4559185.
[54] L. Sherry, P. Polson, and M. Fear, “Designing User-Interfaces for the Cockpit: Five Common Design Errors and How to Avoid Them,” Nov. 2002, doi: 10.4271/2002-01-2968.
[55] J. A. Bashatah and L. Sherry, “A Model-Based Approach for the Qualification of Standard Operating Procedures,” in *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*, Apr. 2021, pp. 1–10. doi: 10.1109/ICNS52807.2021.9441587.
[56] D. Kortenkamp, R. P. Bonasso, D. Schreckenghost, K. M. Dalal, V. Verma, and L. Wang, “A Procedure Representation Language for Human Spaceflight Operations,” p. 8, 2008.
[57] M. Dalal and J. Frank, “Bridging the gap between human and automated procedure execution,” *IEEE Aerosp. Conf. Proc.*, Mar. 2010, doi: 10.1109/AERO.2010.5446851.

[58] R. C. Simpson, “An XML Representation for Crew Procedures.” Aug. 01, 2005. Accessed: Oct. 22, 2022. [Online]. Available: https://ntrs.nasa.gov/citations/20050202022
[59] S. White, “Introduction to BPMN.” Jul. 06, 2004. Accessed: Oct. 22, 2022. [Online]. Available: https://www.bptrends.com/introduction-to-bpmn/
[60] M. Matessa and P. Polson, “List Models of Procedure Learning,” Jan. 2006.
[61] “Model 3 Owner’s Manual,” Tesla. Accessed: Jul. 05, 2023. [Online]. Available: https://www.tesla.com/ownersmanual/model3/en_us/GUID-32E9F9FD-0014-4EB4-8D96-A8BE99DBE1A2.html
[62] J. Bashatah and L. Sherry, “Method for Formal Analysis of the Type and Content of Airline Standard Operating Procedures,” in *2023 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, Apr. 2023, pp. 1–8. doi: 10.1109/ICNS58246.2023.10124313.
[63] J. S. Wood and S. Zhang, “Evaluating Relationships between Perception-Reaction Times, Emergency Deceleration Rates, and Crash Outcomes using Naturalistic Driving Data,” *Transp. Res. Rec.*, vol. 2675, no. 1, pp. 213–223, Jan. 2021, doi: 10.1177/0361198120966602.
[64] S. Soares, A. Lobo, S. Ferreira, L. Cunha, and A. Couto, “Takeover performance evaluation using driving simulation: a systematic review and meta-analysis,” *Eur. Transp. Res. Rev.*, vol. 13, no. 1, p. 47, Sep. 2021, doi: 10.1186/s12544-021-00505-2.
[65] M. Kuehn, T. Vogelpohl, and M. Vollrath, “TAKEOVER TIMES IN HIGHLY AUTOMATED DRIVING (LEVEL 3)”.
[66] H. Guo, K. Xie, and M. Keyvan-Ekbatani, “Modeling driver’s evasive behavior during safety–critical lane changes: Two-dimensional time-to-collision and deep reinforcement learning,” *Accid. Anal. Prev.*, vol. 186, p. 107063, Jun. 2023, doi: 10.1016/j.aap.2023.107063.
[67] B. Wolfe, B. Seppelt, B. Mehler, B. Reimer, and R. Rosenholtz, “Rapid holistic perception and evasion of road hazards,” *J. Exp. Psychol. Gen.*, vol. 149, no. 3, pp. 490–500, Mar. 2020, doi: 10.1037/xge0000665.

## BIOGRAPHY

Jomana Bashatah received her Bachelor of Science in Systems Engineering from George Mason University in 2018 and then received her Master of Science in Systems Engineering in 2019. After receiving her Doctor of Philosophy in Systems Engineering and Operations Research from George Mason University in 2024, she will join the Industrial Engineering and Systems Department at Princess Nourah University in Riyadh, Saudi Arabia, as a faculty member.