Design and evaluation of the user interface for tractor air seeder systems

by

Aadesh Kumar Rakhra

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfillment of the requirements of the degree of

Doctor of Philosophy

Department of Biosystems Engineering

University of Manitoba

Winnipeg, Manitoba, Canada

ABSTRACT

User interface (UI) design for modern agricultural machines is crucial to the farmer's ability to accomplish their operational goals. A UI must address the information needs of the operator and adhere to the principles of human factors and UI design. This thesis describes the development and evaluation of a UI for an agricultural air seeder using a User-Centered Design (UCD) approach. Our primary focus was on the goals and information needs of the operators following situation awareness (SA) principles. We designed, tested and improved UI by conducting laboratory experiments using human participants in a two-phase study.

The first phase of the study involved the design and evaluation of eight elements of the air seeder interface on an individual basis. We compared the new UI elements with existing baseline (also referred as original or old) elements using the metrics of situation awareness (SW), mental workload (MWL), and subjective feedback (SF) of research participants. Comparison of the UCD and baseline UI designs showed improvements in SA, MWL, and SF of the users. The UCD interface elements invoked greater SA (the maximum improvement was 11% with a mean difference of 5.0 (4.8%), 95% CI (6.47, 3.60) and $P < 0.0001$) and lower MWL (the maximum reduction was 19.7% with a mean difference of -5.2 (-7.9%), 95% CI (0.11, -10.36), and $P = 0.0228$). Users significantly preferred the UCD elements over the baseline elements during SF. User's suggestions and the non-subjective outcomes (such as preference about colors and shapes, and performance consequences) led to further design changes.

The second phase of the study involved the design and evaluation of two versions of the UCD interface (UCD1 and UCD2; conceptually different in terms of details and

complexity) against the baseline UI using the metrics of SA, MWL, SF and user response time under different levels of automation (low and high) and steering type (manual and auto). UI design proved to be a significant main effect on the response time; UCD2 interface performed best (120 cs), followed by UCD1 (141 cs) and the baseline interface (174 cs). UI design did not prove to be a significant main effect on SA (level 1, 2, 3, trend or overall).

Regarding the MWL, UI design proved to a significant main; UCD2 interface performed significantly better than the UCD1 and baseline UI. Subjective ratings of the UCD interfaces are proved better than the baseline interface. Odds ratio for the UCD2 to baseline was 12.92, 95% CI (9.17, 18.21), and $P < 0.001$, and for UCD1 to baseline it was 2.84, 95% CI (2.08, 3.87), and $P <0.001$.

The presence of automation was also a significant main effect on the SA and MWL of the users. SA was better with the high level of automation with a mean difference of 1.34, 95% CI (0.21, 2.48), and $P <0.020$. A high level of automation also reduced the MWL with a mean difference of 0.61, 95% CI (0.26, 0.95), and $P < 0.001$. Moderate levels of automation combinations performed better than the extreme automation options (i.e., fully autonomous or manual).

ACKNOWLEDGMENTS

You are brighter, when someone enlightens you,

You are higher, when someone lifts you,

You are stronger, when someone backs up you,

And you are happier, when someone smiles along with you.

I am immensely grateful to my advisor Dr. Danny Mann for his all-around guidance, unconditional support, and natural compassion. You have not only given me the opportunity but also provided me with the means to see me grow further and do well in other areas of my life too. Also, I am truly thankful to my committee members; Dr. Jason Morrison for his constructive and critical feedback which encouraged me to explore more, and to improve further in my research endeavor, and Dr. Andrea Bunt for her prompt and expert counseling, and kind support. I am greatly thankful to my external examiner, Dr. Viacheslav Adamchuk for his time and contributions.

I am also thankful to Dr. Garry Crow for his helpful discussions and guidance. I am grateful to Dr. Jitendra Paliwal for his encouragement and support. I am also thankful to the other faculty members, and the staffs of the Biosystems Engineering for providing me the needed help. I am much thankful to all of my friends, fellow researchers, and the study participants. I am particularly grateful to Behzad Bashiri, Michele Berthelette, Uduak Edet, and Ivan Panfilov.

I would like to acknowledge the funding and support from NSERC, Mitacs-Accelerate, and Buhler Versatile Inc. during my Ph.D. program.

I am deeply grateful to my parents (Sh. Rama Nath and Smt. Tripta Devi) for their profound love and blessings. Thanks to my brother, my in-laws, relatives, and friends;

their constant support always strengthens me. I am truly thankful to Sh. Raghu Bhaji, and Mata Ji for their unparallel blessings. Finally, I would like to thank my wife Pooja; nothing would have been possible without her. I am immensely grateful to Pooja for her selfless love and constant commitment. I really appreciate the love of my kids Agam and Nishtha; they are the core of my existence. In the end, I am grateful to the Almighty for his mercy and eternal grace bestowed on me.

DEDICATION

I dedicate this work to Gurudev Maharishi Bhrigu Ji.

All the actions, I performed through my mind, body, and communication, towards the fulfillment of my Ph.D. education, with utmost devotion and complete surrender, I offer to Maharishi Bhrigu Ji.

TABLE OF CONTENTS

Abstract ii

Acknowledgments iv

Dedication vi

List of Tables xiii

List of Figures xviii

Chapter 1 Introduction 1

1.1 Background 1

1.2 Research Objectives 2

1.3 Research Execution 3

1.4 Dissertation Outline 5

Chapter 2 Literature review 6

2.1 Automation 6

2.2 Situation Awareness 8

2.2.1 Interface design guidelines from situation awareness perspective 13

2.3 Supplementary cognitive and perceptual psychology principles related to interface design 17

2.3.1 Gestalt principles of form perception 17

2.3.2 Role of colors in the presentation 20

2.3.3 How to attract a user's attention 23

2.4 Additional guidelines regarding interface design 24

2.5 Evaluation of the user-interfaces for SA 26

2.5.1 Situation Awareness 27

2.5.2 Mental Workload 31

2.5.3 Reaction time and subjective feedback 33

Chapter 3 Determining Situation Awareness requirements for an air seeder 35

3.1 Goal-Directed Task Analysis 35

3.2 Situation Awareness Requirements 44

3.3 Summary 50

Chapter 4 Conceptual designs for the user interface 51

4.1 Tractor speed 55

4.2 The desired path, travel direction, and location of the unit along with guidance bar 58

4.3 Blockage indication in seed and fertilizer distribution tubes 62

4.4 Seed and fertilizer application rates 64

4.5 Tool depth 67

4.6 Seed and fertilizer level status (tank level) 70

4.7 Fan RPM 73

4.8 Tool pressure 76

Chapter 5 Design and Evaluation of Individual Elements of the Interface for an Agricultural Machine[1] 80
5.1 Abstract 80
5.2 Introduction 81
5.3 Materials and Methods 84
5.3.1 Design of the Interface Elements 84
5.3.2 Evaluation of the Interface Elements 91
5.4 Results and Discussion 95
5.4.1 Situation Awareness 95
5.4.2 Mental Workload 98
5.4.3 Subjective Feedback 101
5.5 Limitations of the Study 104
5.6 Conclusions 105
Chapter 6 Design and evaluation of the User-Centered interface for studying situation awareness and mental workload of the operator of air seeder under varying levels of automation 107
6.1 Introduction 107
6.2 Automation and consequences 108
6.3 Interface design with a focus on situation awareness 110
6.4 Materials and methods 112

6.4.1 Designing of the user interface for tractor air seeder 112

6.4.2 Evaluations of the interfaces 119

6.5 Results and discussion 123

6.5.1 Situation awareness 124

6.5.2 Trend 135

6.5.3 Overall effectiveness 139

6.5.4 Mental workload 140

6.5.5 Response time analysis 147

6.5.6 Subjective data feedback 150

6.6 Summary and conclusion 152

Chapter 7 Conclusions 157

7.1 Specific conclusions 157

7.1.1 Phase 1 of the Study 157

7.1.2 Phase 2 of the study 157

7.2 General conclusions 160

7.3 Significance and implications 164

7.4 Limitations and future research prospects 165

References 169

Publications 178

Appendix A 180

A.1 Ethics Approval Certificate for the study 180

A.2 Consent form for the participants used during the First-Phase of the Study 181

A.3 Socio Demographic Questionnaire (Phase 1) 184

A.4 Subjective Feedback Questionnaire (Phase I) 185

A.5 Consent form for the participants used during the Second-Phase of the Study . 186

A.6 Socio demographic questionnaire (Phase II) 189

A.7 Subjective Feedback Questionnaire (Phase II) 190

Appendix B 191

B.1 Subjective feedback obtained during phase II of the study 191

Appendix C 194

C.1 Study design and data analysis considerations 194

Appendix D 196

D.1 Output of the oridinal logistics regression 196

Appendix E 197

E.1 Screenshots of the experiments during Phase 1 of the study 197

E.2 Screenshots of Phase 2 of the experiments 200

Appendix F 203

F.1 Screenshot of the coding environment (Microsoft Visual Studio Express), which was used to develop interface simulations for the tractor air seeder. 203

F.2 Example Program used to run interface simulation for designing, testing and evaluating the air seeder interface 204

LIST OF TABLES

Table 3-1. Determined SA requirements related to the tractor (power unit) based on GDTA mentioned in Figure 3-2 to Figure 3-7. 46

Table 3-2. Determined SA requirement related to the tractor and air seeder unit based on the GDTA as mentioned in Figure 3-2 to Figure 3-7.. 47

Table 3-3. Determined SA requirement mainly related to the air seeder unit based on GDTA mentioned in Figure 3-2 to Figure 3-7. 48

Table 3-4. Determined SA requirements related to both tractor and air seeder unit based on GDTA mentioned in Figure 3-2 to Figure 3-7.. 49

Table 4-1. Description of the application of SA and cognitive psychology principles in speed indicator design. 56

Table 4-2. Description of the application of SA and cognitive psychology principles for the design of spatial and temporal presentation tractor air seeder unit. 59

Table 4-3. Description of the application of SA and cognitive psychology principles for the design of blockage status inside the seed and fertilizer distribution tubes. 62

Table 4-4. Description of the application of SA and cognitive psychology principles for the design of seed application rate and fertilizer application rate. 65

Table 4-5. Description of the application of SA and cognitive psychology principles for the design of tool depth. 68

Table 4-6. Description of the application of SA and cognitive psychology principles for the design of tank levels. 71

Table 4-7. Description of the application of SA and cognitive psychology principles for the design of fan rpm. 74

Table 4-8. Description of the application of SA and cognitive psychology principles for the design of tool pressure. 77

Table 5-1. Situation awareness and human factors principles that informed the design of the new air seeder display elements. 88

Table 5-2. Questions asked of the participants at the end of the experiment for subjective feedback. 95

Table 5-3. Summary of statistical analysis (n = 30, α = 0.05) of single-tail t-test (matched pairs) comparing situation awareness (SA) responses for users of OD and UCD interfaces for air seeder display elements. 96

Table 5-4. Summary of statistical analysis (n = 30, α = 0.05) of single tail t-test (matched pairs) comparing mental workload (MWL) for users of OD and UCD interfaces for air seeder display elements. 99

Table 5-5. Summary of statistical analysis (n = 30, α = 0.01) of single-tail t-test (matched pairs) comparing the situation awareness (SA), preference, and overall effectiveness of OD and UCD interfaces for air seeder display elements based on the subjective feedback. 101

Table 6-1. Questions asked of the participants at the end of every experiment session to rate the three interfaces. 123

Table 6-2. Results of the Tukey's simultaneous tests for differences of means (N = 90) of Automation*Steer-Type interaction showing statistically significant differences. 128

Table 6-3. Results of the Tukey's simultaneous tests for differences of means (N = 60) of Automation*Interface-design interaction showing statistically significant differences. 129

Table 6-4. Results of the Tukey's simultaneous tests for differences of means (N = 30) of Automation-level*Steer-type*Interface-design interaction showing statistically significant differences. 130

Table 6-5. Results of the Tukey's simultaneous tests for differences of means (N = 90) of Automation*Steer-type interaction showing statistically significant differences. 132

Table 6-6. Results of the Tukey's simultaneous tests for differences of means (N = 60) of Automation*Interface-design interaction showing statistically significant differences. 133

Table 6-7. Results of the Tukey's simultaneous tests for differences of means (N = 30) of Automation*Steer-type*Interface-design interaction showing statistically significant differences. 134

Table 6-8. Results of the Tukey's simultaneous tests for differences of means (N = 60) of Automation*Interface-design interaction showing statistically significant differences. 136

Table 6-9. Results of the Tukey's simultaneous tests for differences of means (N = 90) of Automation*Steer-type interaction showing statistically significant differences. 137

Table 6-10. Results of the Tukey's simultaneous tests for differences of means (N = 30) of Automation*Steer-type*Interface-design interaction showing statistically significant differences. 138

Table 6-11. Results of the Tukey's simultaneous tests for differences of means (N = 90) of Automation*Steer-type interaction showing statistically significant differences. 140

Table 6-12. The differences of means (N = 180) for MWL in low and high automation scenario. 141

Table 6-13. Results of the Tukey's simultaneous tests for differences of means (N = 120) for MWL in Interface-design levels showing statistically significant differences. . 142

Table 6-14. Results of the Tukey's simultaneous tests for differences of means (N = 90) for MWL in Automation*Steer-type levels showing statistically significant differences. 143

Table 6-15. Results of the Tukey's simultaneous tests for differences of means (N = 60) for MWL in Automation*Interface-design levels showing statistically significant differences. 144

Table 6-16. Results of the Tukey's simultaneous tests for differences of means (N = 60) for MWL in Automation*Interface-design levels showing statistically significant differences. 145

Table 6-17. Results of the Tukey's simultaneous tests for differences of means (N = 30) for MWL in Automation*Interface-design*Interface-design levels showing statistically significant differences. 146

Table 6-18. Results of the Tukey's simultaneous tests for differences of means (N = 60) for MWL in Automation*Interface-design*Interface-design levels showing statistically significant differences. 148

Table 6-19. Results of the Tukey's simultaneous tests for differences of means (N = 60) for MWL in Automation*Interface-design*Interface-design levels showing statistically significant differences. 149

Table 6-20. Ordinal Logistics Regression output of Interface-Ranking (best, average, worst, with a total count of 900) as dependent (response) variable versus Interface-Design (A-Old, B-UCD1 & C-UCD2) as independent (predictor) variable. 151

LIST OF FIGURES

Figure 2-1. Gestalt principle of closure. 17

Figure 2-2. Gestalt principle of continuity. 18

Figure 2-3. Gestalt principle of proximity. 18

Figure 2-4. Gestalt principle of similarity. 19

Figure 2-5. Gestalt principle of symmetry. 19

Figure 2-6. Gestalt principle of common fate. 20

Figure 2-7. Gestalt principle of figure-ground. 20

Figure 2-8. Most distinct colors of the visible spectrum which cause strong signals on color perception. 21

Figure 2-9. Usage of colors along with other visual cues to enable quick recognition of a change. 22

Figure 2-10. Effect of paleness (left), size (middle) and separation (right) on the distinction among colors. 23

Figure 2-11. Various methods used for measuring Situation Awareness based on Endsley et al. (2003). 28

Figure 3-1. Preliminary goals and sub-goals hierarchy for tractor air seeder operation. 37

Figure 3-2. Delineation of decisions and SA requirements related to the goal of 'ensure proper functioning of the tractor. 38

Figure 3-3. Delineation of decisions and SA requirements related to the goal of 'ensure proper functioning of the air seeder........ 39

Figure 3-4. Delineation of decisions and SA requirements related to the goal of 'efficient and safe running of the tractor.'........ 40

Figure 3-5. Delineation of decisions and SA requirements related to the goal of 'efficient and safe running of the air seeder.'........ 41

Figure 3-6. Delineation of decisions and SA requirements related to the goal of 'ensure operator comfort and safety.'........ 42

Figure 3-7. Delineation of decisions and SA requirements related to the goal of 'timely maintenance and servicing.'........ 43

Figure 4-1. Earlier conceptual interface design for TAS-DS representing steering angle, speed, fan rpm, fertilizer application rate, speed application rate, tool depth, tool pressure, air seeder tanks status, blockage, seed spillage and plugging......... 52

Figure 4-2. Conceptual driver interface design for TAS-DS designed on the basis of SA and other human factors principles. 53

Figure 4-3. Greyscale presentation of the conceptual driver interface for TAS-DS designed based on SA and other human factors principles. 54

Figure 4-4. Speed indicator of the unit. 55

Figure 4-5. Spatial and temporal representation of the tractor air seeder unit inside the field........ 59

Figure 4-6. Blockage indication inside the seed and fertilizer distribution tubes of air seeder. 62

Figure 4-7. Seed application rate and fertilizer application rate of the tractor air seeder system. 65

Figure 4-8. Tool depth of the tillage tool inside the soil. 68

Figure 4-9. Status of the quantity of seed and fertilizer remaining inside the tanks....... 71

Figure 4-10. Fan RPM of the air seeder fan. 74

Figure 4-11. Tool pressure experienced by the tillage tools of air seeder. 77

Figure 5-1. Detailed goal-directed task analysis for a tractor and air seeder system. ... 85

Figure 5-2. Pictorials used to display air seeder elements during simulator studies in the Agricultural Ergonomics Laboratory at the University of Manitoba (based on Karimi et al., 2011). 86

Figure 5-4. Newly designed air seeder display elements based on user-centered design (UCD) principles to support the situation awareness of the user. 88

Figure 5-5. Questions asked of the participants after every simulation during the experiment. 94

Figure 5-6. Percentage improvement in situation awareness (SA) for user-centered design (UCD) over original design (OD) for air seeder display elements. 98

Figure 5-7. Percentage change in mental workload (MWL) for user-centered design (UCD) over original design (OD) for air seeder display elements. Negative values indicate a reduction in MWL. 100

Figure 5-8. Subjective feedback from participants on the user-centered design (UCD) and original design (OD) regarding situation awareness (SA), recall, preference, overall effectiveness, and individual display elements. Values are percentage differences. .. 104

Figure 6-1. Baseline (Old) interface based on the design of Karimi et al. 2011........... 114

Figure 6-2. UCD based interface elements from phase 1 of the study. 115

Figure 6-3. Version 1 of UCD (UCD1) based Interface elements. 116

Figure 6-4. Version 2 of the UCD (UCD2) based interface elements. 118

Figure 6-5. Depicting changes in guidance bar in auto-steering and manual steering configurations. In auto-steering mode (1), the system was automatically controlling the variation in the steering, and always displaying the corrected steering position (green-diamond) to the user. Whereas in manual-steering mode, the user was responsible for fixing the steering variation (during the lower automation level), change in steering movement was visible to the user (green, yellow or red diamonds depending upon the variation in the steering). 121

Figure 6-6. Snapshot of the form used to collect user responses after every simulation of the interfaces. 122

Figure 6-7. Mean values of level 1 type SA for automation-level*Steer-type interaction. Letters (A, B…) indicate grouping information based on Fisher's LSD pairwise comparison. Values that do not share a letter are significantly different. 125

Figure 6-8. Mean values of level 2 type SA for low and high automation levels. 126

Figure 6-9. Mean values of level 2 type SA for automation-level*Steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different. 127

Figure 6-10. Mean values of level 2 type SA for automation-level*interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different. 128

Figure 6-11. Mean values of level 2 type SA for automation-level, Steer-type, and Interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different. 129

Figure 6-12. Mean values of level 3 type SA for high and low automation levels. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different. 131

Figure 6-13. Mean values of level 3 type SA for automation-level and steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different...................... 132

Figure 6-14. Mean values of level 3 type SA for automation-level and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different...................... 133

Figure 6-15. Mean values of level 3 type SA for automation-level and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different...................... 134

Figure 6-16. Mean values of the Trend identification during the high and low automation scenario.. 135

Figure 6-17. Mean values of the Trend for automation-level and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different...................... 136

Figure 6-18. Mean values of the Trend identification for automation-level and steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different...................... 137

Figure 6-19. Mean values of the Trend identification for automation-level, steer-type, and Interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different. 138

Figure 6-20. Mean values of the overall situation awareness for high and low automation scenario. 139

Figure 6-21. Mean values of the overall situation awareness for automation-level and steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different. 140

Figure 6-22. Mean values of the mental workload for low and high automation scenario. 141

Figure 6-23. Mean values of the mental workload for three types of Interface designs viz., UCD1, Old and UCD2. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different. 142

Figure 6-24. Mean values of the mental workload for automation-level and steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different...................... 143

Figure 6-25. Mean values of the mental workload for automation-level and steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different...................... 144

Figure 6-26. Mean values of the mental workload for steer-type and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different...................... 145

Figure 6-27. Mean values of the mental workload for automation-level, steer-type and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different. 146

Figure 6-28. Mean values of the reaction time (centiseconds) for three types of interface-designs (Old, UCD1 and UCD2). Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different. 148

Figure 6-29. Mean values of the reaction time (centiseconds) for steer-type and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different. 149

Figure 7-1. Comparison of seed application rate for UCD1 (A) and UCD2 (B) design. 160

Figure 7-2. Comparison of the design of Phase 1 (original) and Phase 2 (modified) of the UCD1 interface elements. 162

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The user-interface (UI) is an integral part of today ’s automated and technology-dependent agricultural machines. The effectiveness of a UI is not only critical for achieving farmers operational and business goals but also crucial to machinery manufacturers for providing a better user experience for high value and complex agricultural machines. A typical UI helps the operator in monitoring, controlling and altering the target task environment (Liu 1997). Depending upon the level of automation (Parasuraman et al. 2000) and function allocation between man and machine (Bye et al. 1999), the machine operator depends upon the UI for fulfilling information needs to accomplish critical operational goals. The operator’s efficiency, workload, and safety are often negatively impacted when the UI does not meet the information needs of the operator.

Parasuraman and Riley (1997) mention that overreliance on automation, underutilization of automation and, implementation of a function’s automation without understanding the impact on human performance are the main contributors to automation-related problems. Weak interaction and feedback, insufficient monitoring and ineffective decision-making are some of the other drawbacks associated with the automation in the system (Norman 1990). These types of concerns are often related to the ‘lower level of situation awareness’ that mainly happens due to the ‘out of loop’ state of operators when the role of the operator shifts from the active-participant to passive-monitor of the system (Endsley and Kiris 1995).

Interface design with a user-centered approach could reduce the ‘out-of-the-loop’ problem by supporting the situation awareness (SA) of the users (Endsley 2003). Usage of the user-centered approach in designing interfaces for military and aviation applications is a common practice. Now, researchers and engineers from several other fields, such as “power, transportation, cyber, space, unmanned and autonomous systems, maritime, command and control, oil drilling, and health care,” are considering the user-centered and SA strategy for designing interfaces to handle automation and performance-related issues (Endsley 2015). However, this user-centered and SA-based approach for interface design is not common in the agricultural domain. This study focuses on understanding and applying user-centered and SA principles for designing a UI for an air seeder systems for a precision farming application.

A common air seeder system comprises an air cart and a seeding/tillage device. The air seeder distributes seeds and fertilizer in the air cart through a pneumatic conveying system to the seeding/tillage device for precise placement in the soil. Efficient and profitable operation of the air seeder is complex, and demands adequate situational awareness by the operator. This complexity makes the air seeder interface a preferred choice for designing and testing that incorporates situation awareness. The baseline interface used to compare the new UCD interface came from an earlier researcher (Karimi et al. 2011) at the University of the Manitoba’s Ergonomics lab.

1.2 RESEARCH OBJECTIVES

To apply a UCD process for developing a UI, understanding of the operational goals and SA information needs of the operator are essential. Hence, the content (number and type of interface elements), form, and functionality of the interface are determinable after specifying the SA information needs of the air seeder. While a complete interface consists of several individual elements; the design, testing, and improvements of individual elements were performed before the final interface design. This research set to address the following research objectives:

1. To design UCD oriented individual UI elements for the air seeder to support the SA of the operator. Then, to test and compare the UCD based UI elements with the baseline UI regarding SA, and mental workload (MWL) of the users. (Hypothesis: UCD based individual UI elements are better regarding SA and MWL.)
2. To design UCD oriented full UI (using the outcomes and feedback of the first objective) for the air seeder to support the SA of the user. Then, to evaluate and compare the UCD based UI with the baseline UI regarding SA, and mental workload (MWL) of the users. (Hypothesis: UCD based UI is better regarding SA and MWL of the user.)
3. To compare and test the effect of different levels of automation, steer-type and interface design on the SA, and MWL of the user. (Hypothesis: Different levels of automation and steer-type should influence the MWL and SA of the users, and UCD based design should enable better SA, and MWL of the users.)

1.3 RESEARCH EXECUTION

For detailed understanding of the SA requirements, while operating the tractor air seeder system, it was essential to understand the operator's goals, along with the functionality, features, and general working of the tractor air seeder system. The researcher itemized the SA requirements for the tractor air seeder system using a goal-directed task analysis approach while working as an intern for Buhler Versatile Inc. These SA requirements were done simultaneously to planning functional specifications for the interface of the other subsystems (hydraulic control system, three-point hitch, engine, transmission, power-take-off, differential lock, tractor performance parameters, and other optional features) of the tractor. This exposure has provided the background to the researcher to conceptualize the initial interface design to support the SA needs of the operator. After several preliminary attempts of conceptual UI design, the researcher realized the need for deeper understanding of the human factors and perceptual psychology principles related to UI design. Subsequently, the researcher further explored the scientific literature related to the human factors and perceptual psychology for designing UI.

After conceptual design, it was necessary to develop an interface to test and compare in a dynamic situation to achieve practical and reliable outcomes from the research. Development of the interface in-line with the research objectives was a complicated process. It required a thorough understanding of the programming language along with months of the rigorous efforts to code, debug and recode the thousands of lines of the program for a functional interface simulation. The programming language for the interfaces was Visual Basic (VB) because: 1) the baseline interface was developed in the VB, and 2), VB is easy to learn, and there was a lot of online help available. The first phase of the study addressed objective 1 by designing and testing individual air seeder interface elements. Then two versions of the complete air seeder interface were developed and compared with baseline conditions as per research objective 2 and 3 during the second phase of the study.

1.4 DISSERTATION OUTLINE

This dissertation is divided into seven chapters. Chapter 1 provides an introductory background justifying the research and presenting the research objectives. Chapter 2 provides a literature review related to; automation; user-centered and SA design; and other human factors and cognitive psychology information about interface design. Chapter 3 explains the goal-directed task analysis while illustrating the SA requirements of the operator of the tractor air seeder. Chapter 4 describes the conceptual interface design and explains the application of design principles in designing interface elements.

Chapters 5 and 6 are written in manuscript format. Chapter 5 describes the research methods and the outcomes of phase 1 of the study while comparing and testing the UCD oriented interface elements with baseline counterparts. Chapter 6 describes phase 2 of the experiments; two versions of the UCD (UCD1 and UCD2) oriented interfaces are tested and compared to the baseline conditions under different automation conditions as per research objectives 2 and 3.

Chapter 7 summarizes this research project by providing conclusions of the study and discussing future research possibilities. References and appendices are included to provide information about the sources and background work involved in the research.

CHAPTER 2

LITERATURE REVIEW

2.1 AUTOMATION

When automation is introduced to a process, it is typical to have outcome expectations that include: a decrease in physical and mental workload, lower production cost, reduction of errors, improvement of efficiencies, and minimizing the impact of human behavior and physiology on the quality and quantity of work performed. For example, computers often analyze vast amounts of information faster and with more accuracy than humans. They can project trends, provide decision making support and even implement the required action in most work domains. However, depending upon the relationship between the human and the system (i.e., interaction and division of tasks), the workload, SA, and outcome of the operation may all be different (Endsley 1996; Endsley and Kiris 1995; Endsley and Kaber 1999).

Parsuraman et al. (2000) explained automation as “full or partial replacement of the function previously carried out by the human operator,” and proposed four broad divisions of functions where automation applies. These divisions are: information acquisition; information analysis; decision and action selection, and action implementation. Automation of information acquisition involves “sensing and registering of the input data” (e.g., displaying of steering angle, tool depth, seed application rate are examples of sensing and registering input data in the context of the tractor air seeder operation). Automation of information analysis involves cognitive activities like memory, reasoning, judgment, logic, and inference. Judgment or recognition of higher or lower tool depth than the optimum tool depth is an example of information analysis that is automated when the machine performs these functions. Automation of decision and action selection further extends the essence of information analysis by executing the decision and action based on the analysis. For example, if the analysis infers that tool depth is higher than the optimum, then the decision would be to reduce the tool depth by selecting the tool depth lever. In action implementation mode, the system acts, by reducing tool depth to the optimum level.

Within each of these types, Parsuraman et al. (2000) proposed that the level of automation can vary from a scale of 1 to 10 (i.e., fully manual to fully automatic). Guidelines to system designers, suggest the evaluation of the type and level of automation based on: ‘human performance consequences’ (mental workload, situation awareness, complacency and skill degradation) and ‘automation reliability and the cost of decision/action consequences.’ After evaluation, system designers can adopt any suitable combination of automation (level and type) to meet their needs best. However, human performance problems are often about the automation in the system (Billings 1991; Skitka et al. 1999; Parasuraman and Riley 1997). Parasuraman and Riley (1997) mentioned that overreliance on automation, underutilization of the automation and, implementation of the function’s automation without understanding the impact on human performance are the main contributors in problems related to automation. Issues related to a decrement of the situation awareness (Endsley 1996; Dao et al. 2009); weak interaction and feedback (Norman 1990); and manual skill degradation are also observed about the automation in the system. These issues can be associated with the phenomenon of ‘out-of-the-loop performance’ (Endsley and Kiris 1995).

Several problems in the aviation domain occurred when pilots failed to obtain an adequate level of SA. Though sources of SA related errors are associable with all three levels of SA, the most errors fell under level 1 SA. About 76.3% of SA errors are related to the “failure to correctly perceive information” (which is level 1 SA), and 20.3% SA errors are related to the “failure to comprehend the situation” (level 2 SA), whereas 3.4% SA errors are related to “failure to project situation into the future” (level 3 SA) (Jones and Endsley 1996). Therefore, the loss in situation awareness of the operator can be associated with the loss in performance which poses the need for increased SA for the efficient operation of the machine (Endsley et al. 2003). These outcomes seem to imply that the overall effectiveness of an interface is influenced by the design that adheres to and supports the SA of the operator.

2.2 SITUATION AWARENESS

The paradigm of situation awareness is used in the design and evaluation of interfaces in many areas beyond the aviation and military domain. These areas include, but are not limited to, “power, transportation, cyber, space, unmanned and autonomous systems, maritime, command and control, oil drilling, and health care” (Endsley 2015). Golestan et al. (2016) used the concept of SA for interface design in the transportation domain proposing specific models and procedures for: perceiving the objects and their relationships; comprehending the “situation assessment” and “threat assessment,”; and projecting the “impact assessment.” In another study, Burns et al. (2007) compared traditional and improved interfaces related to a nuclear power plant using the premise of SA. The modified displays, mentioned as “ecological interfaces,” incorporate visuals and graphics into the traditional displays. These modified displays showed better outcomes than the traditional interfaces in specific conditions. Taylor et al. (2010) used a user-centered approach to improve the SA of the users during the design and evaluation of the interfaces for a regenerative life support system. The study inferred that the UIs which presented ‘situation-rich’ information helped users in better decision making. Findings from the scientific studies support the premise that interface designers should consider SA during the design process and to evaluate the adequacy of interface design.

Endsley (1988) defined situation awareness as “the perceptions of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future.” In level 1 (perception level), people perceive verbal or non-verbal information from the environment using various senses (i.e., auditory, visual, taste, smell, touch). At level 2 (comprehension level), the user processes and understands the information about the current situation by using available mental faculties. Lastly, the projection level (level 3), the user predicts the near future situation based on the current understating (level 2 SA) (Endsley et al. 2003). Creation of SA involves other processes including: perception, attention, working memory, mental model, schema and scripts (Endsley et al. 2003).

Mental Models, Schema and Scripts According to Rouse and Morris (1986) as cited by Endsley et al. (2003), “Mental models are the mechanisms whereby humans can generate descriptions of the system purpose and form, explanations of system functioning and observed system states and predictions of future system states.” Alternatively: “Why a system exists,” “How a system works,” “What a system is doing” and “What a system looks like” are the questions that the mental model addresses

(Rouse and Morris 1986). Information presented in-tune with the existing mental models of a person may tax less on cognitive resources and may enhance SA and reduce human errors. Improper design of control panels can have serious consequences. Casey (1993) described the near-fatal experience of a fighter plane pilot during World War II. An experienced pilot rushed into a plane during one of the air raids on Pearl Harbor, but could not get the plane into the air because the layout of the controls differed from the previous models he had flown. Design engineers neglected to consider that fighter pilots would have insufficient training time on a new instrument panel after its in the middle of the War.

The schema is the complex structure that plays an essential role in the attainment of SA of a person. As explained by Endsley et al. (2003), “These schemata or prototypical patterns are retained in the memory” and “people use a process called pattern matching to link the cues taken in from the current situation on schemata to pick the best match from those available.” Due to these pre-existing patterns in memory, people can quickly perceive and comprehend a situation. For example, an experienced person outperforms a novice because of the well-developed mental models and schema to tackle any situation in their environment despite a large amount of data, stress or complexity. Scripts are the mechanisms that trigger a sequence of actions associated with schema (Endsley et al. 2003). For example, an experienced surgeon has a well-developed script for performing heart surgery; which implies that the surgeon’s actions while performing a routine surgery would be spontaneous and natural with no stress of figuring out his/her next course of action during the operation.

Attention Humans cannot effectively perceive information from simultaneous multiple sources. To perceive information from some source, people need to attend to that source. However, one's ability to attend to multiple items simultaneously is limited (Endsley et al. 2003). In a dangerous or stressful environment, a person's attention becomes even narrower, and only focuses on 'those aspects of the situation he considers most important' (Baddeley 1972). The simultaneous focus of attention on several sources is difficult.

To get a rich understanding of one's surroundings, people need to share their attention with different sources of information. For example, while driving a car, it is essential to watch the traffic ahead, to watch the mirrors, to perform frequent shoulder checks, or to watch in-cab displays (i.e., the speed gauge, radio panel, turn signals). Likewise, pilots must scan several sources of data constantly to collect information from several instrument panels. A person's attention may focus on only one specific aspect of an environment, and the remaining elements are intentionally or unintentionally ignored. This narrowing or focusing of attention on certain aspects of the environment is described as 'attentional tunneling,' which severely affects the SA of the ignored part of the system (Baddeley 1972; Endsley et al., 2003).

Working Memory Working memory or short-term memory in users is limited. Miller (1956), in one of his papers in the cognitive research area "The Magical Number 7, Plus or Minus 2 - Some Limits on Our Capacity for Processing Information", described that human information processing capacity is limited. Researchers observed that the 'channel capacity of absolute judgment' of the participants was about 5 to 10 items for a 'unidimensional stimuli.' For example, listeners could identify a maximum of about 6 different tones making no mistakes (Pollack 1952, 1953) as cited by Miller (1956). However, for a ‘multidimensional stimuli,’ ‘channel capacity’ (‘upper limit on the extent to which observer can match his responses to the stimuli’) increased significantly. For example, when locating dots in a square from memory, observers could accurately identify about 24 positions (Klemmer and Frick, 1953) as cited by Miller (1956).

People need to scan their environment regularly to achieve good SA. Information obtained from one source may be needed later to assist the decision-making process. Due to the limitations of working memory, information may fade quickly (Baddeley 1984). This non-availability of information can negatively affect the SA and may cause fatal human errors. An example is a major aircraft accident (i.e., runway collision) that occurred at Los Angeles International Airport in 1991 when an air traffic controller forgot that an aircraft was moved to a runway and assigned another aircraft to land on the same runway (cited in Endsley et al. 2003).

Other SA considerations When data are present in quantities that exceed the human ability to process and comprehend them, these data hinder the achievement of good SA (Endsley et al. 2003). Instead, if these data are present in the form of graphs or trends (or in another similar visual form), better SA with lower stress on working memory is possible. (DeSanctis 1984; Endsley et al. 2003; Tullis 1981). Signs or objects that are salient by specific characteristics (such as colors, shapes, flashing) place less tax on cognitive resources (Christ 1975; Cummings 2005; Yuditsky et al. 2002). Complexity is another enemy of SA (Horrey 2011; Liu 2001). If a device, panel or display contains too many options, features or selections, it would be difficult for a person to perceive and comprehend the information (level 1 & level 2 SA), and to foresee the near future situation (level 3 SA) (Endsley et al. 2003). Researchers (Endsley et al. 2003; Fennell 2006; Stanton and Marsden 1996; Young et al. 2007) described several human errors and fatalities that happened because pilots had an inadequate understanding of automatic flight management systems. Displays that offer seamless shift between ‘goal driven’ (or ‘top-down’) processes to ‘data-driven’ (or ‘bottom-up’) processes provide the best support to the SA of a person (Endsley et al. 2003).

2.2.1 Interface design guidelines from situation awareness perspective

To design effective interfaces, researchers have recommended several interface design guidelines. Notably, from the perspective of improving SA, a leading researcher (Endsley et. 2003) described several principles for designing UI. The summary presented in this section is reproduced (from Rakhra and Mann 2014) for convenience.

Focus on the goals Instead of providing information generated from many sensors, technologies, and systems (e.g., temperature, pressure, speed, and rpm), provide and organize information related to the goals of the user.

Make it easy for the user Support level 2 SA (comprehension) by presenting usable information instead of lower-level data. This approach will put less load on the cognitive faculties (particularly on short-term memory and attention) of the user.

Assist in level 3 SA Help the user anticipate the near future based on the current information and data (i.e., trend lines, graphs, and visual presentations).

Maintain the global SA While the user is interacting with several sources and types of information, ‘attentional tunneling’ can occur if the user becomes too involved with one aspect of the information at the expense of the other. Providing ‘big-picture’ about the environment can ensure better SA of the user.

Trade-offs between goal-driven and data-driven processing The approach of focusing on the goals refers to a goal-driven strategy. Although this approach may be essential in certain situations, it may lead to ‘attentional narrowing.’ Displays designed to provide the ‘big-picture’ tend to use a data-driven approach which provides a wider range of information to the operator. The user’s attention is attracted to high priority goals using salient clues like a blinking light, sound or vibration. Both approaches are essential, and the displays which provide seamless shift between data-driven and goal-driven processing are best in supporting the SA of the user.

Make critical cues salient During the presentation of the information, the critical stages or intervals should be highlighted (i.e., if fuel level has dropped to 25%). This salient information would be helpful in the decision-making process.

Using parallel processing capabilities When using multiple displays (or interacting with several sources of information), the simultaneous use of visual, auditory or tactile modes may facilitate switching attention from one priority to another.

Handling information filtering Solutions to information overload (or extraneous information) are obtained by not presenting all the information to the operator. The summarized outcome is usually shown to the user, but the intermediate information processing steps formed by the computer are kept hidden. However, such filtering may deprive the user of the overall understanding of the information and system. The user may lack the SA required to cope with near future situations or handle emergencies if automation fails (i.e., out-of-the-loop syndrome).

Uncertainty and missing information When there is an ambiguity in the information provided, the user is forced to 'fill in the blank' based on experiences or expectations. Banbury et al. (1998) (cited by Endsley et al. 2003) described a situation where pilot responses differed when they were exposed to the information in two different ways. In the first scenario, pilots were provided with information stating that there was a 93% probability that it is an enemy aircraft. In the second scenario, they were provided with information that there was a 93% probability that it is an enemy aircraft and 7% probability that it is friendly aircraft. In the latter case, pilots were less likely to attack the aircraft even though both sets of information provided the same information.

Avoid excessive features Addition of features adds both cost and complexity. Only required features that are used most frequently and by most of the population should be added to displays. Other options that less frequently used can be placed in less prime locations.

Map system functions to the goals and mental models of a user When system functionality is displayed in line with the goals of the user, it makes it easy for the user to understand and operate the system.

Minimize display density but not at the cost of coherence Excessive data presented on display may overwhelm the user when searching for information. Too little information on a given display may result in the need for multiple levels of the display at the expense of increased complexity and cost. Designers must maintain the balance between display density and coherence.

People should not rely exclusively on alarms; provide projection support Alarms alert people to act immediately. They create a rapid response situation and cause stress. It is better to provide proactive information and projection of the near future so that the user may act even before the alarm sounds.

Provide SA support rather than decision Instead of providing a decision; it is more useful to provide situation awareness support and let the operator decide what to do. The system can provide several alternatives based on computer analysis. This approach also helps the operator be in control and in-the-loop.

Transparency in automation Providing transparency in the automation actions may reduce the probability of errors. Johnson and Pritchett (1995) (cited by Endsley et al. 2003) described a situation where the crash of an aircraft occurred due to the pilot's misunderstanding of the descent of the aircraft (3.3 degree versus 3300 feet per minute). In both cases, the display panel showed 33; only the mode indication provided the appropriate units as FPA or VS (FPA for Flight Path Angle, VS for Vertical Speed).

Besides the SA related principles mentioned above, researchers have also reported the effects of various colors, shapes, size, graphs and parallel processing capabilities (like audio or touch) on human performance (Liu 2001; Tullis 1981; Christ 1975). Liu (2001) reported that multimodal displays (visual and auditory) for advanced traveler information systems (ATIS) showed better human performance (response time, correct turns and subjective workload ratings) compared to a visual display alone. Also, multimodal displays caused fewer errors in other navigation and control-related tasks compared to visual displays. Tullis (1981) pointed out the improvement in the response time was observed using graphic displays compared to the narrative displays.

2.3 SUPPLEMENTARY COGNITIVE AND PERCEPTUAL PSYCHOLOGY PRINCIPLES RELATED TO INTERFACE DESIGN

In addition to the SA-related guidelines mentioned above, there are several other supplementary principles and guidelines based on cognitive and perceptual psychology which can be used for effective designing of driver interfaces for mobile agricultural machines (MAMs). These guidelines were mentioned in Rakhra and Mann 2014, and reprinted here for the information of the reader.

2.3.1 Gestalt principles of form perception

According to Gestalt theory (Todorovic 2008; Wikipedia 2014), the human mind is able to perceive meaningful shapes from a group of various objects, lines or figures. Gestalt principles illustrate the natural inclination of human perception to grasp the whole picture rather than the individual parts of it. These principles can be summarized as below.

The principle of closure People perceive objects (i.e., pictures, letters, and shapes) as whole even if the individual parts are missing (Figure 2-1).

Figure 2-1. Gestalt principle of closure.

The principle of continuity This principle states that the 'visual perception is biased to perceive continuous form rather than disconnected segments' (Johnson 2010). In Figure 2-2, four disconnected shapes are perceived as two intersecting arrows.

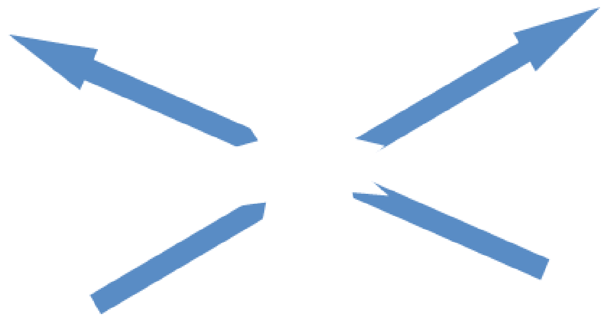


Figure 2-2. Gestalt principle of continuity.

The principle of proximity When people see a collection of objects, the objects placed closer to each other are perceived as being grouped. In Figure 2-3, instead of a single group of 32 stars, it is perceived as two groups of 16 stars each.

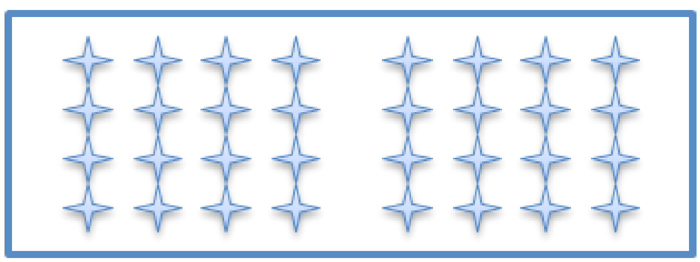


Figure 2-3. Gestalt principle of proximity.

The principle of similarity Objects which look similar to each other within a collection of objects are perceived as being together. This similarity can be by size, shape, color or any other visual property (Figure 2-4).

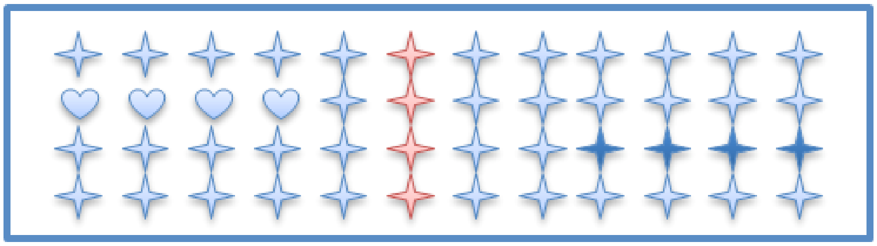


Figure 2-4. Gestalt principle of similarity.

The principle of Symmetry This principle states that our perception is naturally inclined towards simplicity and symmetry while looking at a complex object made of several symmetrical shapes. Figure 2-5 is perceived as five squares and four arrows, but not as several triangles, shapes, lines or parts.

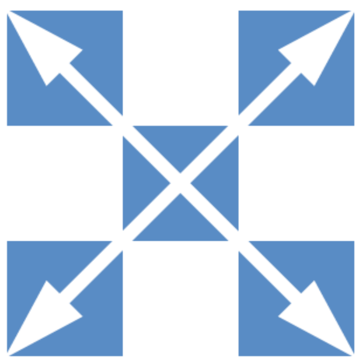


Figure 2-5. Gestalt principle of symmetry.

The principle of common fate In an array of objects, items are perceived as being grouped or related if they move together. For example, in an array of 48 stars (4 rows by 12 columns) in Figure 2-6, if the items of column 2 and 11 start to move simultaneously towards the right side, they will be perceived as being related or grouped.

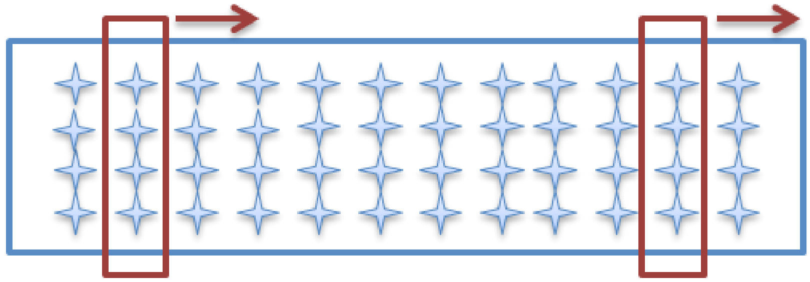


Figure 2-6. Gestalt principle of common fate.

The principle of Figure/Ground This principle states that we perceive visual field in the form of figure (salient or main attraction) and ground (background or lesser prominent) (Figure 2-7). However, this figure/ground paradigm is also influenced by 'characteristics of the scene' and how the viewer is focusing on the image (Johnson 2010).

Figure 2-7. Gestalt principle of figure-ground.

2.3.2 Role of colors in the presentation

Our ability to perceive information from an image is profoundly influenced by the color scheme being used. An understanding of the human's ability to interact and perceive colors can be crucial to the effectiveness of presenting information, particularly towards the improvement of the level 1 type of situation awareness: "perception of the elements of the environment" (Endsley 1988). Johnson (2010) has provided the following guidelines regarding the usage of colors during interface design.

Use of most distinct colors Human vision is not equally sensitive to all colors. Due to the complex internal mechanism of the human vision system, some shades of the visible color spectrum cause strong signals on our color perception and easily catch our eyes. These six distinct colors are black, white, red, green, yellow and blue (Figure 2-8). However, if these opponent colors (red, green, yellow, blue, black and white) are used adjacent to each other, they produce a 'shimmering effect'; this should be avoided (Johnson 2010).

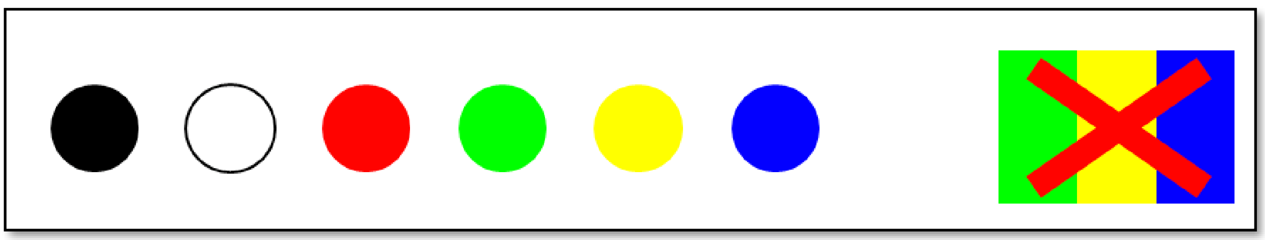


Figure 2-8. Most distinct colors of the visible spectrum which cause strong signals on color perception.

Using brightness and saturation to support distinction among colors Similar colors (hue) should be made different by applying a distinct level of brightness and saturation. On the standard HSB color scheme, colors can be understood by hue, saturation, and brightness. Hue is the property which distinguishes one color from other (e.g., green is different from red). Saturation represents the "pureness" of a color; high saturation color contains no or a little amount of an opponent or complementary colors (e.g., pure or high saturated red color has no traces of other colors like green or blue). Brightness can vary from total bright (i.e., white) to total dull (i.e., black). To ensure that the colors have enough difference, viewing that color image on the black and white or grayscale provides easy judgment regarding their distinction (Johnson 2010).

Colors for color–blind people Approximately 8% of men and less than 1% of women have difficulty in color perception (Wolfmaier 1999 as cited by Johnson 2010). The most common type of color-blindness is red/green which can prevent the person from being able to distinguish dark red from black, blue from purple and light green from white. Viewing the colors in grayscale can help in judging the distinction among colors. However, lighter color selections can be made from orange, yellow, green or blue-green while dark colors can be selected from violet, blue, purple or red (Johnson 2010).

For differentiation do not rely on colors alone Applying shape, size or any other cue along with the colors are more effective than using colors alone. Mainly when there are groups of items, differentiation based on color along with other cues can be more efficient rather than relying on the colors alone (Johnson 2010). Figure 2-9 is showing the application of other cues along with colors for differentiation.

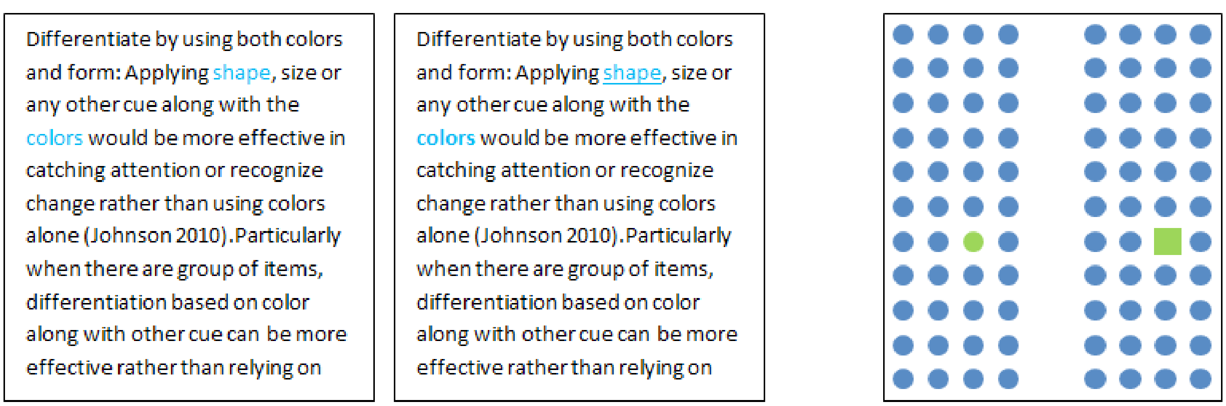


Figure 2-9. Usage of colors along with other visual cues to enable quick recognition of a change.

For example, underline or bold words along with the colors (left) or change in shape from a circle to square along with the change in color (right).

Paleness, size, and separation Johnson (2010) explained that, by paleness alone (lighter version of color with same hue or less saturated color), it is harder to distinguish among two colors (Figure 2-10 left); specifically, when comparable items are smaller in size (Figure 2-10 middle). Furthermore, the farther the comparable items are from each other, the more difficult it will it be to distinguish among them (Figure 2-10, right).

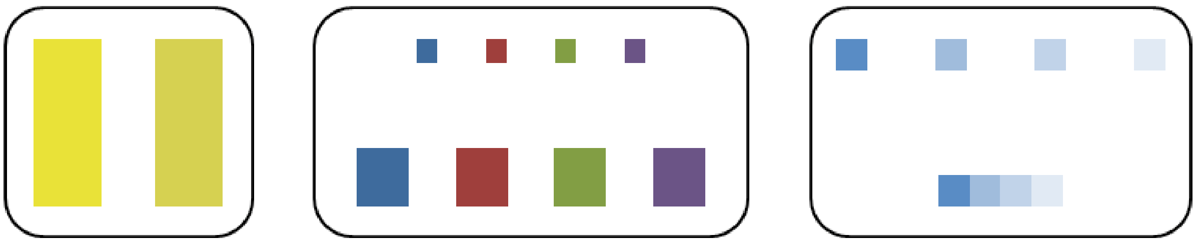


Figure 2-10. Effect of paleness (left), size (middle) and separation (right) on the distinction among colors.

2.3.3 How to attract a user's attention

Johnson (2010) described three common and effective ways to catch the user's attention: 1) using a pop-up message box, 2) using a beep sound and 3) making objects flash or wiggling. All methods possess a different level of influence. For example, using a pop-up dialog box may force the user to attend to the immediate priority by impeding other information sources. A beeping sound or wiggling objects may not directly interfere with the information being observed by the user, however, they are hard to miss by the user. All these methods should be used with caution; excessive or unnecessary use of these techniques may annoy the user. The user may even start to ignore these heavy cues due of the phenomenon of habituation ("decrement of response with repeated stimulation") (Sharpless and Jasper 1956).

2.4 ADDITIONAL GUIDELINES REGARDING INTERFACE DESIGN

Weber's Law Weber's law is a fundamental law in psychophysics which states that there is a threshold which must be exceeded for any change to be detected (Feldman 2008). This law can be applied to characteristics such as size, shape, color, luminosity, quantity, and sound. The minimum value (i.e., 'just noticeable difference') of the stimulus is a 'constant proportion' of the intensity of the initial value of the stimulus. This constant proportion is also called Weber's fraction and can vary for features such as sound, size, and weight (Feldman 2008).

Yerkes-Dodson law Yerkes-Dodson law describes the relationship between arousal and task performance. An optimal level of arousal is desirable for producing higher performance in a given task. Depending on the nature of the task, over- or under-arousal may negatively affect the performance of the task (Cohen 2011).

Consistency Google defines consistency as "Conformity in the application of something, typically that which is necessary for the sake of logic, accuracy, or fairness." Norman (1983) pointed out that non-consistent systems can lead to errors while consistent systems can minimize short-term memory problems related to operational performance. Several researchers (Nielsen and Molich 1990; Shneiderman 1998; Shneiderman et al. 2009; Stone et al. 2005) also emphasized the need of consistency, standardization, and predictability during the interface usage.

Feedback Appropriate feedback and interaction with the user is essential for the successful execution of automation mainly "when the situations exceed the capabilities of the automatic equipment; then the inadequate feedback leads to difficulties for the human controllers" (Norman 1990). Stone et al. (2005) mentioned that the operator should always be aware of what is happening in the immediate surroundings. Due to change in feedback or inadequate feedback, out-of-the-loop problems can arise that deprive the operator of a sufficient level of situation awareness to be effective when intervening should automation fail (Endsley and Kiris 1995).

Affordances and conventions Affordances can be understood as the perceptions of the individuals related to the constraints to their actions (Gibson 1977; Norman 1999). For example, a handle on a car door provides natural affordance to open the door by pulling the handle. Similarly, a “turnable” knob for volume control yields the meaning that volume can be raised or lowered by turning the knob. However, in the case of interface design and screen-based products, cultural conventions play a more important role than affordances (Norman 1999). Affordances deal with the physical constraints while cultural conventions (i.e., change in the color of the links after visited, or change in the mouse pointer from an arrow to hand) are of primary importance for interface designing which works as “symbolic communication” (Norman 1999).

Simplicity and structure Stone et al. (2005) mentioned that the UI should be perceived as familiar to the user, and “actions, icons, words, and UI controls” should be “natural” to the user. Structure or organization of the content on the interface is also of primary importance. This organization of the items should be based on the user’s mental model or perception. The features or tasks which the user perceives as being associated together should be co-located or grouped on the interface in some meaningful way (Stone et al. 2005).

2.5 EVALUATION OF THE USER-INTERFACES FOR SA

Designing of a UI as per the user's goals and information requirements is half the battle towards building an effective interface. Although designers apply many interface design guidelines based on human factors principles, it is still common that certain aspects of the design do not work as intended. Evaluation of a UI helps designers to identify ineffective features and other issues to improve the interface further. Commonly recommended interface evaluation methods by various researchers include: heuristic evaluation, cognitive walk-through, usability testing, and guidelines/standard inspection (Jeffries et al.1991; Nielsen1994). Heuristic evaluation and usability testing are considered the most effective methods for improving UIs (Jeffries et al. 1991). Heuristic evaluation method requires many UI experts to evaluate the interface. Usability testing provides data based on objective and subjective evaluations which makes it more suitable for research and scientific studies. However, usability testing is time consuming, costly, and needs expertise.

For this study, we adopted the usability testing strategy for evaluating the UI. Three types of evaluation criteria were considered for evaluation: 1) situation awareness, 2) mental workload, and 3) subjective feedback. Also, response time was used as an evaluation criterion where applicable during the experimentation. Situation awareness was considered as the primary means for evaluation, as SA is not only a widely accepted criterion of evaluation in many domains (Endsley 2015) but also provides necessary guidelines for designing the UI (as discussed in section 2.2). Mental workload is another widely used design and evaluation criteria in various human factors studies (Megaw 2005). Also, subjective evaluations were performed along with the SA and MWL mainly because they are easy to administer and analyze, and provide sufficient openness and flexibility to obtain detailed feedback from the user (Sinclair 2005).

2.5.1 Situation Awareness

There are several approaches used in the evaluation of interfaces from a SA perspective. Mainly related to the measurements of SA in the aviation domain, Endsley et al. (2003) classified these methods as ‘direct’ and ‘indirect’ approaches based on the inference of SA (Figure 2-11).

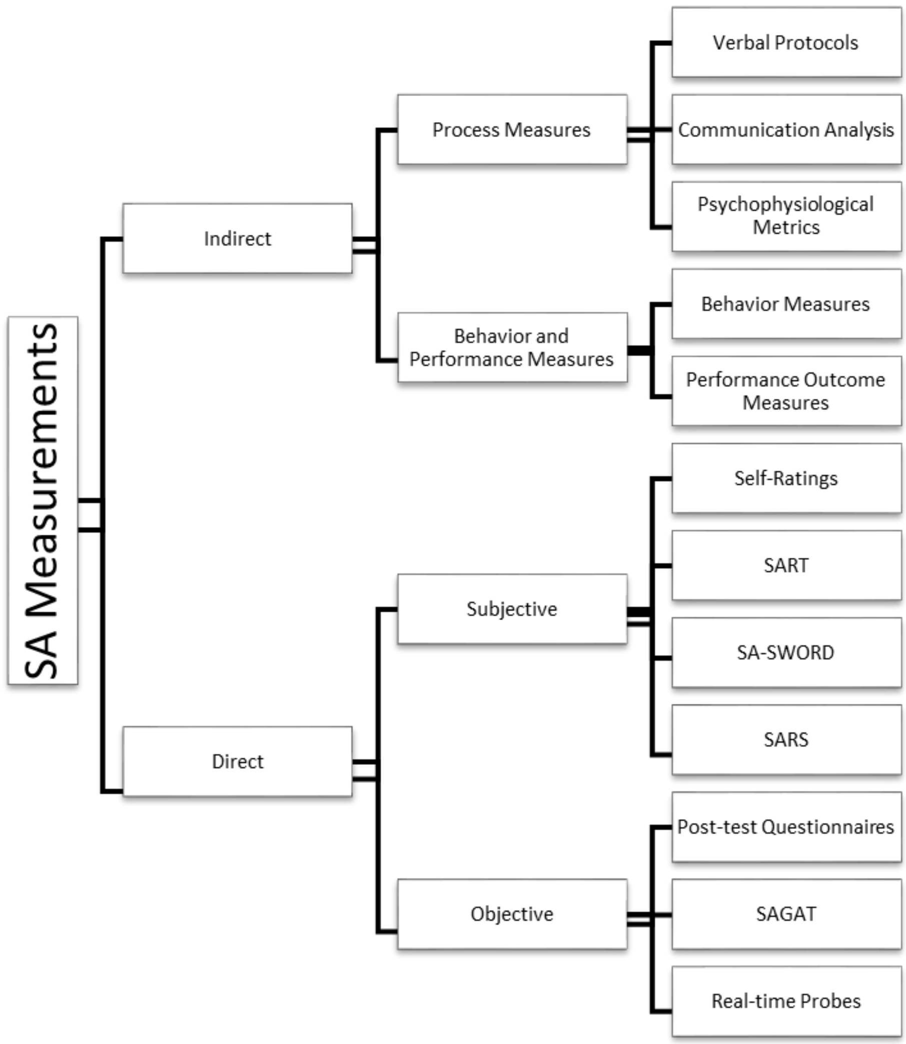


Figure 2-11. Various methods used for measuring Situation Awareness based on Endsley et al. (2003).

Using ‘indirect’ approaches, SA can be inferred by measuring various cognitive processes involved in the formation of SA (i.e., verbal protocols, communication analysis, and psychophysiological metrics or by analyzing behavior and performance outcomes). Verbal protocols method requires that the operator express his/her thoughts verbally while performing a task. Based on the analysis of the recorded verbal communication, an operator’s SA is determined. However verbal protocols cannot represent the exact picture of the SA because describing a situation may depend upon the operator's verbal communication ability and it may be different from his/her real perception, comprehension and projection of the situation. During the communication analysis technique, verbal communication/discussion/exchange of words among the team members is recorded and analyzed afterward to infer the SA of the operator. This strategy is mainly used in a team environment where members communicate with each other when performing various task-related activities (Endsley et al. 2003). Using psychophysiological metrics as a technique for measuring SA, the SA can be inferred from physical movements like eye glance behavior, brain electrical activity (electroencephalography (EEG), or heart electrical activity (electrocardiogram (ECG)). These methods have the advantage of providing continuous and unobtrusive data for later analysis, however, inferring SA from the data is not a proven strategy, and may require further research (Endsley et al. 2003).

Direct approaches for measuring SA involve both subjective and objective techniques. Subjective measures are usually based on the opinion of the operator (i.e., their impression of their SA during the scenario). An operator or observer may be asked to rate their SA on some scale of low to high or may be asked some questions related to the various aspects of the situation which were supposed to relate to SA of the scenario (Endsley et al. 2003). Subjective measures are usually easy to administer and analyze, and do not interfere with the working scenario. However, subjective measures may not help in interpreting the exact level of SA possessed by the operator. Subjective measures can be influenced by the performance of the operator, his/her judgment about the situation, limitations of recall while rating the SA, confidence level of the operator, experience of the operator in the domain (Endsley et al. 2003; Taylor and Selcon 1991;

Fracker and Vidulich 1991; Endsley et al. 1998). The Situation Awareness Rating Technique (SART) is one of the most used testing techniques for testing SA. Although this technique was developed for the evaluation of SA in the domain of aviation, “the construct is domain independent,” making this evaluation technique applicable in other similar environments (Endsley et al. 2003). As explained by Endsley et al. (2003), based on work done by Taylor (1990), the SART metric can be divided into three broad domains (i.e., attentional demand, attentional supply, and understanding). Attentional demand is influenced by “instability of the situation, the variability of the situation, the complexity of the situation”. Attentional supply depends on the “arousal, spare mental capacity, concentration, and division of attention of the operator”. Understanding is related to the “information quantity, information quality and familiarity” (Endsley et al. 2003). Overall SA rating of the SART metric can be calculated as SA (calculated) = Understanding - (Demand - Supply) (Endsley et al. 2003).

In contrast to subjective measures, during objective approaches, SA of the person is not entirely dependent on the self-rating of the person but compared with the facts (Endsley et al. 1998; Endsley et al. 2003). These techniques are quite useful in a simulated environment. For testing a person’s SA and comparing it with reality, displays are randomly blanked at irregular intervals for a duration of 2 to 5 min depending upon the number of queries to be asked (Endsley et al. 1998). This “asking of queries” during the testing scenario works to mitigate the effect of memory losses compared to asking queries at the end of each complete testing scenario (Endsley 1995a). However, several other challenges exist for evaluating a person’s SA using objective measures (Endsley et al. 2003). These challenges are: i) determining the appropriate questions which would reflect the true SA of the person related to the goals, (ii) questions should not interfere or amend the current SA of the person, (iii) question should be able to cover the entire range of SA but not be specific to certain aspects of the environment.

One of the widely accepted and reliable objective techniques for testing SA is the Situation Awareness Global Assessment Technique (SAGAT) (Endsley et al. 2003). This technique has been successfully used in various fields like military operations (Bolstad and Endsley 2003; Gorman et al. 2006; Strater et al. 2001), medical environment (Singh et al. 2006; Wright et al. 2004), nuclear power plant operation (Hogg et al. 1995), air traffic control (Durso et al. 1999; Endsley et al. 2000), human-robot interaction (Steinfeld et al. 2006) and driving (Gugerty 1997; Matthews et al. 2001; Ward 2000).

2.5.2 Mental Workload

The mental workload can be defined as "the amount of cognitive capacity required to perform a given task" (Di Stasi et al. 2013). Human cognition involves many processes such as perception, memory, attention, synthesis, and analysis which place a demand on our cognitive resources. Our cognitive resources are limited; under most cases, researchers observed that the 'channel capacity of absolute judgment ' is around 5 to 10 items (Pollack 1952, 1953) as cited by (Miller 1956). If the complexity and dynamics of the situation increase, demand on the human information processing system also increases (Di Stasi et al. 2013). Parasuraman and Squire (2010) mentioned that "an increase in task load led to lower situation awareness and higher mental workload, reduced mission success and increased mission times." Evaluation of the MWL provides critical insights into design considerations and operational outcomes. As described by Megaw (2005), most of the MWL evaluation techniques can be categorized as: analytical or empirical. The primary premise for this division is that analytical techniques (such as mathematical models and simulation models) do not require the operator to perform the task under investigation, while empirical techniques require the operator to perform the task under investigation. We can categorize empirical techniques into four divisions: primary task performance (e.g., time or error related), secondary task techniques (e.g., loading or subsidiary task), physiological or psychophysiological techniques (such as cardiac or brain activity, eye function,), and operator opinion or subjective techniques. Further details about all these techniques can be read in Megaw (2005).

According to Pickup et al. (2005) based on the work of (Hart and Staveland 1988; Jensen et al. 1994; Muckler and Factors 1992), subjective techniques (self-reports by the operators) are more effective in many ways. For example, self-reports “can reflect the actual effects on a performance better than measures of task demands.” Self-reports are more ‘sensitive’ and ‘accurate’ (as described by Pickup et al. (2005) based on Hart and Staveland (1988)), and operators show better judgment about their workload among varied task conditions (Muckler and Factors, 1992). Pickup et al. (2005) developed and tested a unidimensional MWL scale called the Integrated Workload Scale (IWS). This scale has shown advantages such as simplicity, ease of administration, the speed of use, and minimal obstruction with the task. In the current study, the IWS technique was used for MWL evaluation together with the SAGAT query set for SA evaluation.

2.5.3 Reaction time and subjective feedback

Measurement and analysis of the reaction time, to evaluate a UI, task performance, SA or MWL, is a common practice among human factors studies. For testing SA, using the *situation present assessment method* in Durso et al. (1999) as cited by M. Endsley (2016), the user's response time for answering the questions was considered as means for inferring the SA. Higher response time was associated with lower SA. In another detailed study related to workload, Hart and Staveland (1988) reported that "accuracy decreased and reaction time increased as the difficulty of information processing requirements was increased." Higher levels of subjective workload were associated with a lower level of performance and increased reaction time. In a study (Koch et al. 2013) comparing two types of interfaces in ICU (intensive care unit), lower response time and higher SA was observed for an 'integrated display' compared to the traditional display. From these studies, higher reaction (response) time can be associated with lower SA, higher MWL or lower performance.

For obtaining adequate feedback from the users, post-trial questionnaires and reports are also used to evaluate SA and MWL (Muckler and Factors 1992; Salmon et al. 2006). Asking open-ended questions or obtaining users ratings or preference for certain types of designs and features is common for subjective feedback and evaluation. Open-ended queries provide users with enough space and opportunity to describe and suggest any changes freely. Researchers may use this feedback to improve designs and features further. Closed queries such as obtaining ratings or preferences for a particular design variant or feature may help researchers to quantify the feedback to generate an overall opinion about the design or feature under investigation. Subjective feedback is easy to administer and analyze. Paper pencil based post-trial questionnaires used for subjective feedback are generally less intrusive, and could fit seamlessly as a supplementary procedure along the with the complex SA and MWL assessment strategies in a dynamic situation.

CHAPTER 3

DETERMINING SITUATION AWARENESS REQUIREMENTS FOR AN AIR SEEDER

3.1 GOAL-DIRECTED TASK ANALYSIS

To design an effective UI supporting SA, the operator’s dynamic information needs must be understood (Endsley et al. 2003). Dynamic information needs of the operator are the SA requirements that the operator must possess for effective decision making and goals accomplishment. SA requirements are gathered and illustrated by using the Goal-Directed Task Analysis (GDTA) methodology as explained by (Endsley et al. 2003).

The operator’s goals, decisions needed to accomplish the goals, and SA requirements of the operator vary for different domains. To determine SA requirements of an air seeder operation, designers must understand the operator’s job details and goals in detail. To accomplish this objective, the author studied operator manuals, product specifications and other technical literature relevant to the tractor and air seeder operation during his summer internship at a local agricultural machinery manufacturer (Buhler Versatile Inc.). Also, the researcher had informal discussions with operators, engineers, and farmers to understand the job details, goals, and the major decisions associated with operating an air seeder.

In line with the methodology recommended by Endsley et al. (2003), the researcher performed a detailed GDTA for an existing mobile agricultural machine. The GDTA is independent of any specific model of the machine or the available features and functions. For practical purposes, a Versatile 4WD 435 tractor model attached with a precision seeding drill ML 950 and an AC 400 air cart were considered for the task analysis. Considering the use of GDTA in future studies, a few additional goals and SA requirements of the operator were also considered. Currently, not all goals are supported by the machine under study. The outcome of the complete GDTA is provided in the following figures (Figure 3-1 to Figure 3-7).

Based on the information gathered (from operator's job details, studying user manuals, product information manuals, informal discussion with the operators, product experts, and farmers), a hierarchy of preliminary goals and sub-goals was formulated to deduce complete SA requirements.

Figure 3-1 shows the preliminary goal of the operator to 'operate tractor air seeder system efficiently and safely.' Based on this preliminary goal, six major goals and sub-goals are deduced.

In the subsequent figures (Figure 3-2 to Figure 3-7), each major goal and sub-goal is reviewed separately to delineate decisions and SA requirements associated with the goal under consideration. Underneath each goal in the GDTA hierarchy, decisions required to meet that goal effectively are listed. All the information that the operator requires (i.e., SA requirements) to make that decision are listed. These SA requirements are further broken down into level 1 (perception), level 2 (comprehension), and level 3 (projection) in Section 3.2.

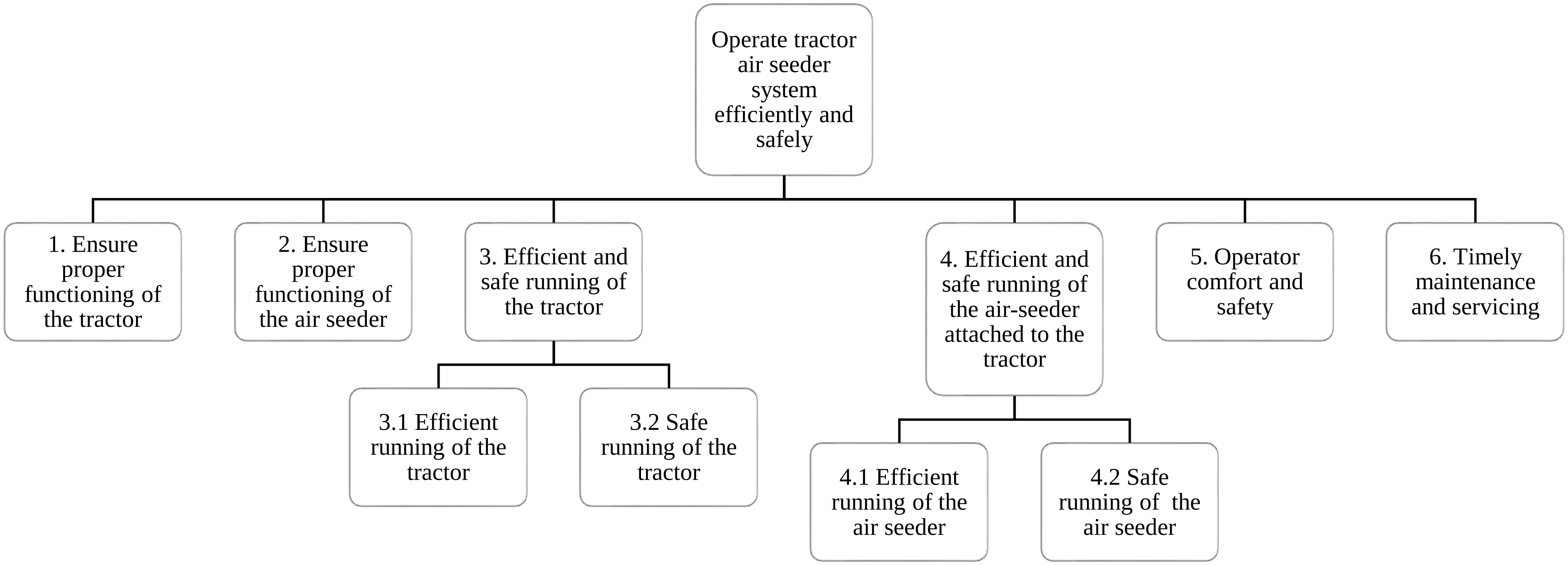


Figure 3-1. Preliminary goals and sub-goals hierarchy for tractor air seeder operation.

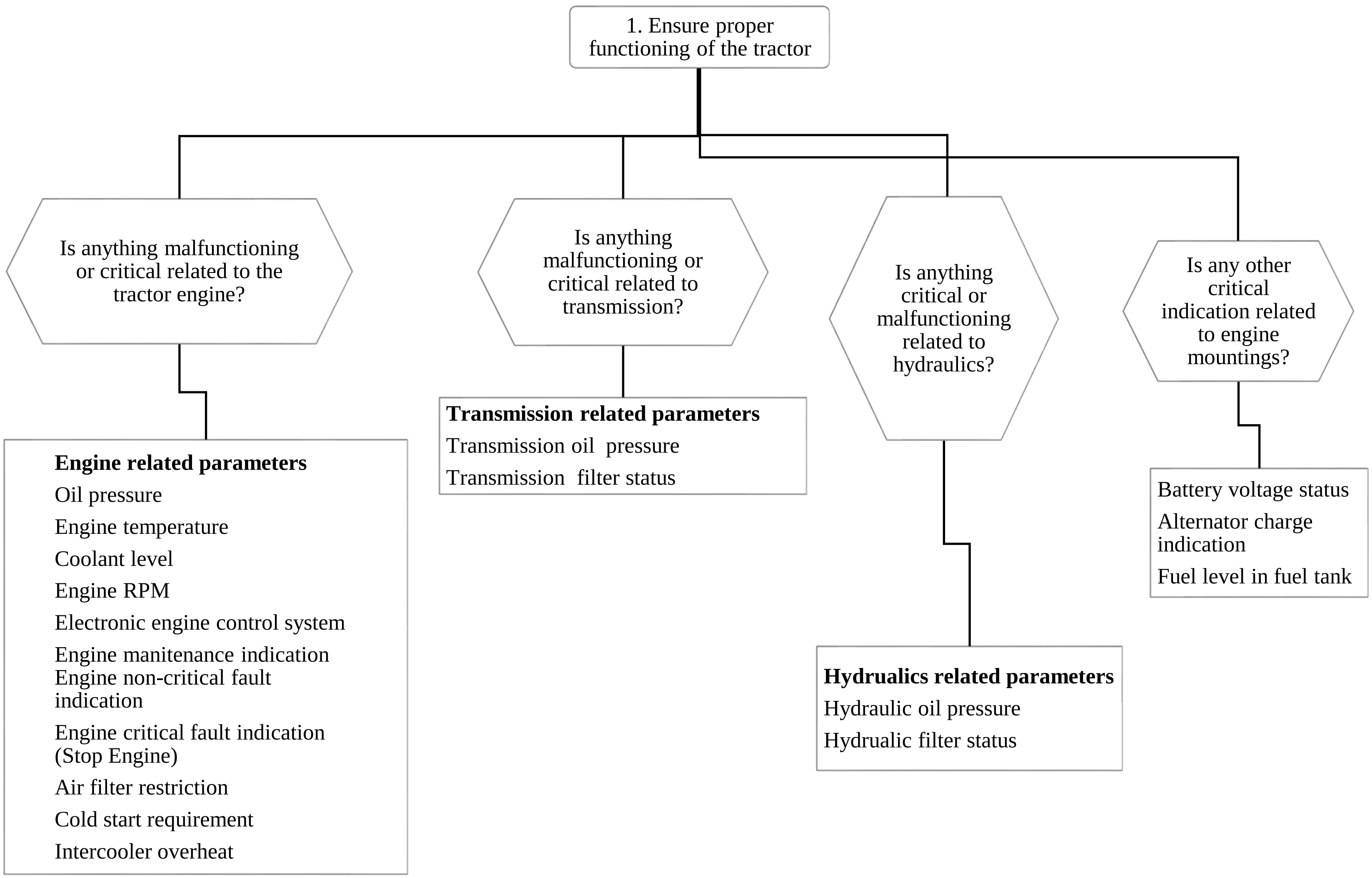


Figure 3-2. Delineation of decisions and SA requirements related to the goal of 'ensure proper functioning of the tractor.

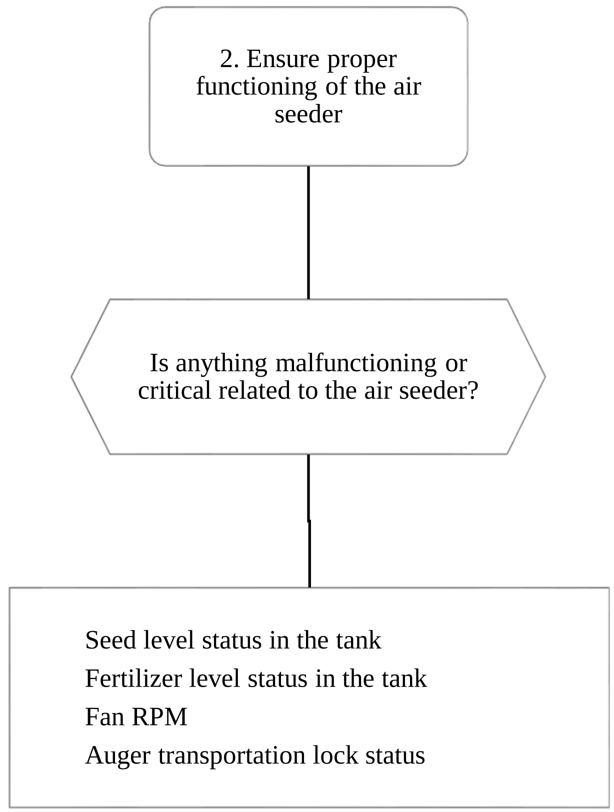


Figure 3-3. Delineation of decisions and SA requirements related to the goal of ‘ensure proper functioning of the air seeder.

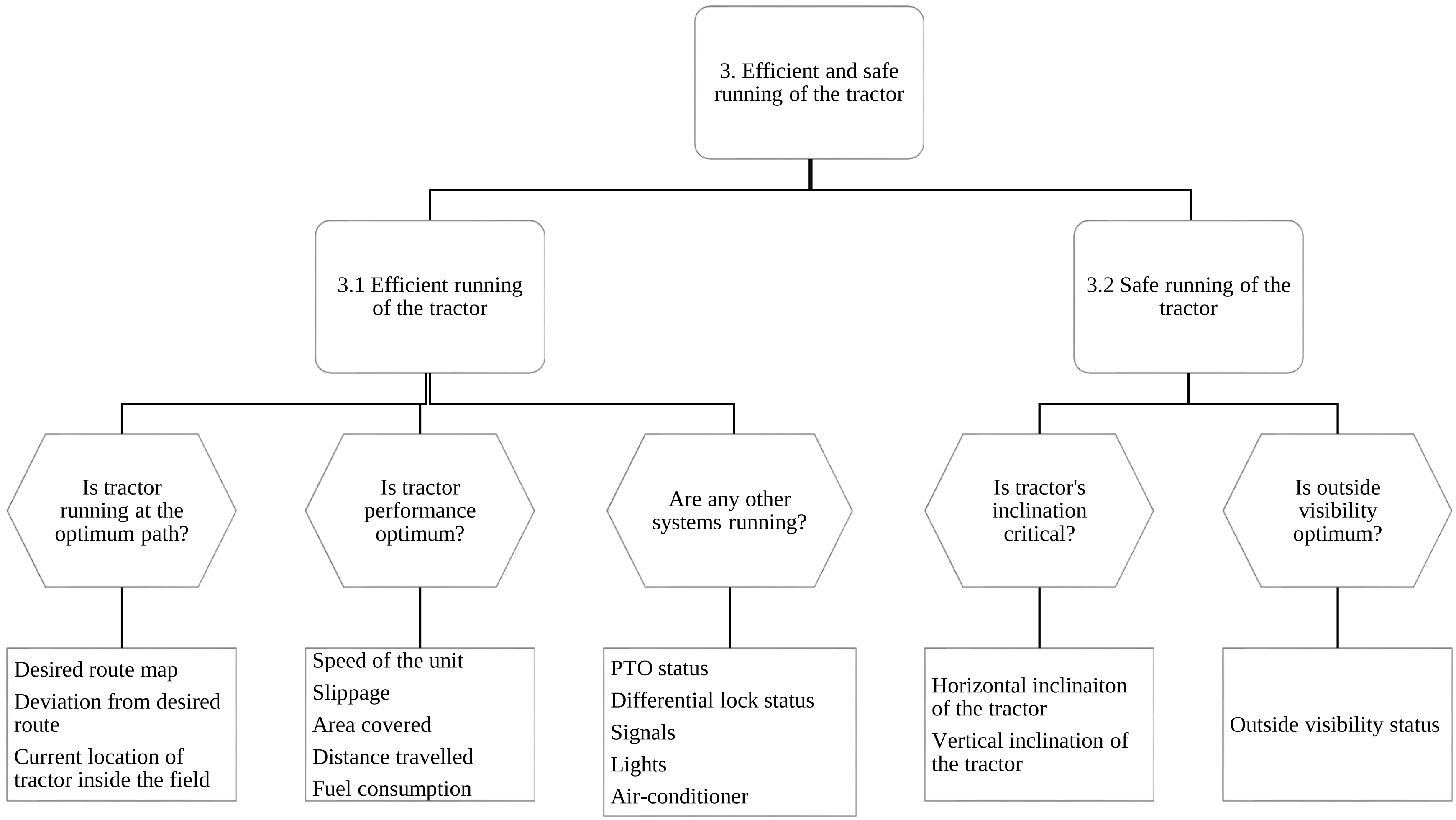


Figure 3-4. Delineation of decisions and SA requirements related to the goal of 'efficient and safe running of the tractor.'

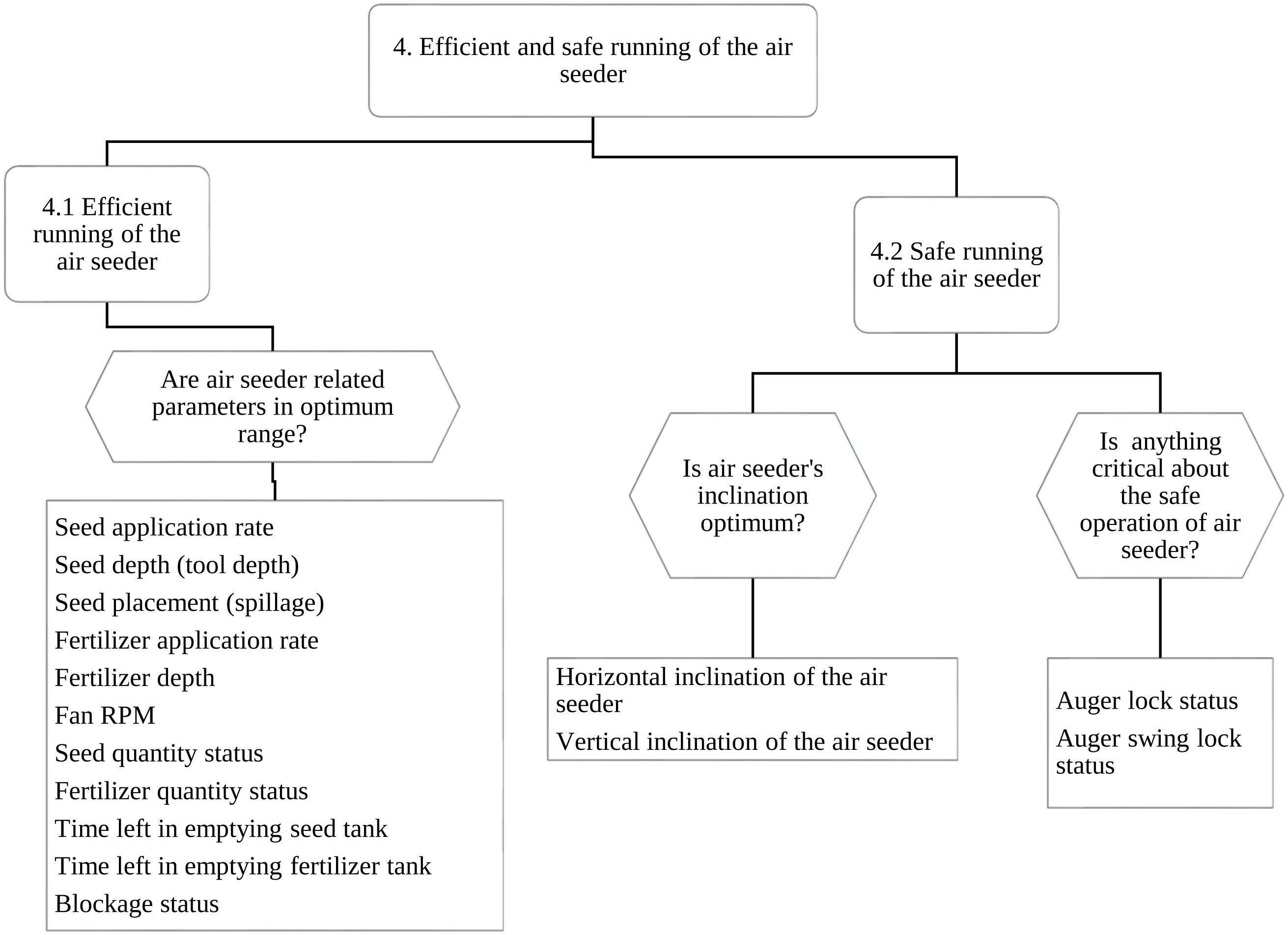


Figure 3-5. Delineation of decisions and SA requirements related to the goal of 'efficient and safe running of the air seeder.'

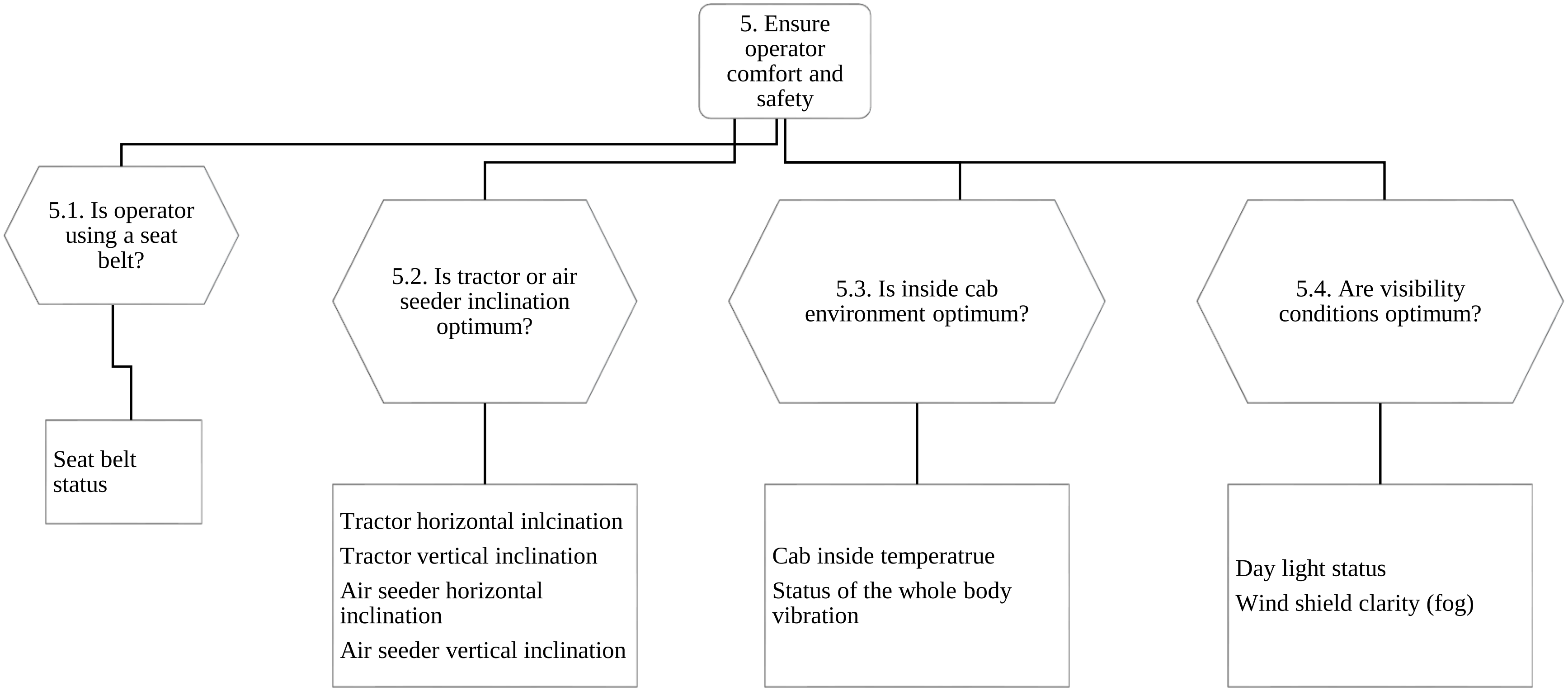


Figure 3-6. Delineation of decisions and SA requirements related to the goal of 'ensure operator comfort and safety.'

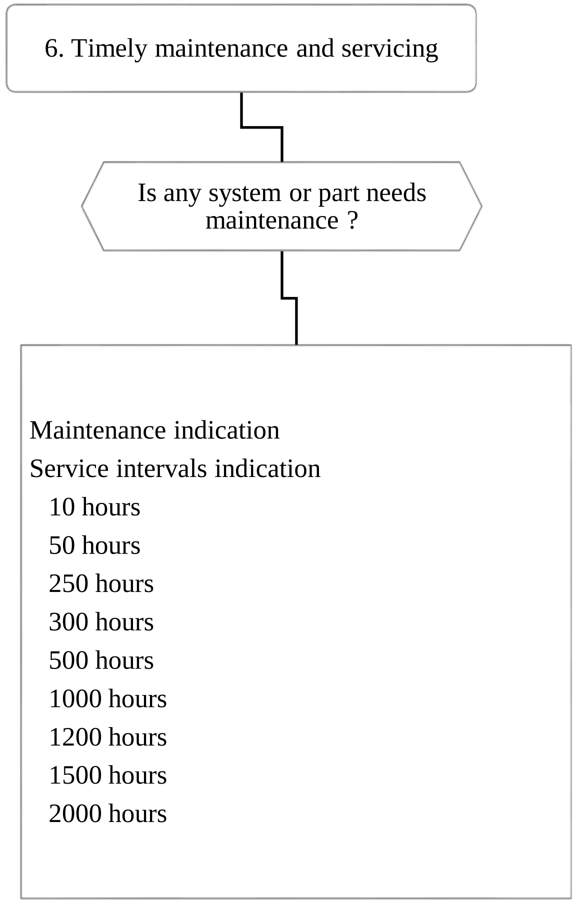


Figure 3-7. Delineation of decisions and SA requirements related to the goal of 'timely maintenance and servicing.'

3.2 SITUATION AWARENESS REQUIREMENTS

In Figure 3-1 to Figure 3-7, by using the GDTA process, various decision and associated SA requirements are delineated. Based on the fundamental concept of the SA, these SA requirements can be further categorized into three levels: perception, comprehension, and projection levels. Accordingly, in Table 3-1 to Table 3-4, SA requirements are tabulated into three columns in each table. A left, middle and right column in each table presents the SA requirements related to the perception, comprehension and projection levels of SA, respectively. Perception (level 1) related SA requirements are the information needs of the operator which he/she is required to perceive to make the associated decisions to accomplish the relevant goals. At level 2-comprehension level, the operator needs to understand the perceived information. For example, when the operator looks at the fuel gauge, he/she perceives the information (level 1 SA) regarding the amount of the fuel inside the fuel tank. Afterward, the operator recognizes that the fuel gauge (needle) is pointing towards the empty zone (or red mark). From this indication of the fuel gauge, the operator infers or comprehends (level 2 SA) that there is very little (or no) fuel in the fuel tank. At projection level, the operator sees the near future based on his/her current understanding of the situation. After the operator infers (level 2 SA) that there is low fuel in the fuel tank; the operator would like to foresee or project (level 3 SA) for how many miles (or acres or duration) the current low-level fuel would last. By all of this information (level 1, 2 and 3 SA), the operator can make a right decision (such as to continue the air seeder operation, or immediately move back for refueling, or to continue the operation for 15 minutes before moving back for refueling). In the following tables (Table 3-1 to Table 3-4) all the SA requirements based on the detailed GDTA (Figure 3-1 to Figure 3-7) are systematically tabulated.

Table 3-1. Determined SA requirements related to the tractor (power unit) based on GDTA mentioned in Figure 3-2 to Figure 3-7.

Level 1

- Tractor functionality
  - Engine oil pressure
  - Engine temperature
  - Coolant level
  - Engine maintenance indication
  - Engine non-critical fault indication
  - Engine critical fault indication (stop indication)
  - Air filter restriction
  - Cold start requirement
  - Intercooler overheat
  - Transmission oil pressure
  - Transmission filter status
  - Hydraulic oil pressure
  - Hydraulic filter status
  - Battery voltage status
  - Alternator not charging the battery
  - Fuel level in fuel tank
  - PTO status (engaged/disengaged)
  - Differential lock status (engaged/disengaged)
  - Turning signals
  - Air-conditioning/heat status
  - Service/routine maintenance indication

Level 2

- Tractor functionality
  - Deviation from required engine oil pressure
  - Deviation from required engine temperature
  - Deviation from required coolant level
  - Maintenance part name and due hours
  - Non-critical fault detail and repercussion
  - Engine critical fault detail and repercussion
  - Deviation from restricted air filter (e.g. 25% left to be fully restricted).
  - Deviation from required transmission oil pressure
  - Deviation from restricted transmission filter
  - Deviation from required hydraulic oil pressure
  - Deviation from restricted hydraulic filter status
- Deviation a from fully charged battery voltage
- Deviation from fully filled fuel tank
- Deviation of temperature from desirable

Level 3

- How much duration is needed to fully charge the battery
- How many hours/miles system can run with current fuel level

Table 3-2. Determined SA requirement related to the tractor and air seeder unit based on the GDTA as mentioned in Figure 3-2 to Figure 3-7.

Level 1	Level 2	Level 3
• Tractor/AirSeeder Performance • <u>Location of a unit inside the field</u> • <u>Desired path of the unit</u> • <u>Current speed of the unit</u> • Slippage • Distance travelled • Current fuel consumption (applies to tractor only) • Horizontal inclination of the tractor • Vertical inclination of the tractor	• Tractor/AirSeeder Performance • Area covered vs. area left • Deviation from desired path • Deviation from desired slippage • Total distance travelled vs. distance left • Deviation from desired fuel consumption (applies to tractor only) • Deviation from desired horizontal inclination • Deviation from desired vertical inclination	• Time required to cover the remaining area (based on average speed) • For how much duration can the system run with current fuel level • How many acres can the system work with remaining level of fuel

Table 3-3. Determined SA requirement mainly related to the air seeder unit based on GDTA mentioned in Figure 3-2 to Figure 3-7.

Level 1	Level 2	Level 3
• Air seeder • Seed level status • Fertilizer level status • Fan RPM • Auger transportation lock status (locked/unlocked) • Seed application rate • Seed depth (tool depth) • Fertilizer application rate • Fertilizer depth (tool depth) • Blockage status	• Air seeder • Deviation from fully filled seed tank (e.g. 55% filled) • Deviation from fully filled fertilizer tank • Deviation from desired fan RPM • Deviation from an optimum seed application rate • Deviation from an optimum seed depth • Deviation from optimum fertilizer application rate • Deviation from optimum fertilizer depth	• For how much acreage current seed level is sufficient • For how much duration current seed quantity is sufficient • For how much acreage current fertilizer quantity is sufficient • For how much duration current seed quantity is sufficient

Table 3-4. Determined SA requirements related to both tractor and air seeder unit based on GDTA mentioned in Figure 3-2 to Figure 3-7.

Level 1	Level 2	Level 3
• Operator comfort and safety • Seat belt indication • Tractor horizontal inclination • Tractor vertical inclination • Air seeder horizontal inclination • Air seeder vertical inclination • Cab inside temperature • Whole body vibration status • Day light status • Wind shield clarity	• Operator comfort and safety • Deviation from required horizontal inclination of the tractor • Deviation from required vertical inclination of the tractor • Deviation from required horizontal inclination of the air-seeder • Deviation from the required vertical inclination of the air-seeder • Deviation from the acceptable level of whole body vibration • Deviation from the optimum day light	

3.3 SUMMARY

The GDTA (section 3.1 and 3.2) provides detailed SA requirements for the power unit (i.e., the tractor) as well as for the auxiliary equipment (i.e., the air seeder). However, the scope of the current work is to study the SA information requirements of the auxiliary equipment only. Based on this GDTA, 38 SA information requirements (mentioned in Table 3-3 and 3-4) related to the air seeder operation at all three levels of SA are identified. However, for the currently available machinery (Versatile 4WD 435 tractor attached with precision seeding drill (ML 950) and air cart (AC 400)) which were considered for the GDTA, not all the goals were supported (e.g., there was no information provided to the operator regarding the unit's inclination). In line with the available features of the machine under study, 12 parameters related to the air seeder were identified to address the SA information requirements of the operator. These parameters are seed level status, fertilizer level status, fan RPM, seed application rate, seed depth (tool depth), fertilizer application rate, fertilizer depth (tool depth), blockage status, desired path of the unit, desired location of the unit and current speed of the unit (Table 3-2 and Table 3-3 – underlined). These parameters were also recommended by Karimi et al. (2011) as most frequently viewed parameters by the operators. We included one other most frequently viewed parameter (i.e., tool pressure) which was recommended by Karimi et al. (2011) in the conceptual UI for the air seeder.

CHAPTER 4

CONCEPTUAL DESIGNS FOR THE USER INTERFACE

During the beginning of the researcher's Ph.D. program, based on a preliminary understanding of the concepts of SA and UI design, the researcher made early attempts to give some shape and form to his initial concepts about the design of the air seeder elements. The first conceptual interface (Figure 4-1) was developed in 2013 (Rakhra et al. 2013). In this conceptual interface (Figure 4-1), 'Steering Angle' section represents the forward direction of travel of the tractor as well as relative position of the tractor inside the field. Speed section shows the forward traveling speed of the tractor and its visual presentation on an elliptical bar indicating the acceptable speed ranges by red markings. Fan RPM section represents the RPM of the air seeder fan. Fertilizer application rate and seed application rate are shown in their respective sections; red lines are showing the minimum and maximum acceptable values whereas current readings are mentioned below in black rectangular area (80 kg/ha for fertilizer application rate and 7.5 ka/ha for seed application rate). A moving arrow inside a column represents tool depth; 6 and 10 cm is the minimum and maximum acceptable values marked with the extended lines. Tool pressure is represented on the elliptical bar with a moving oval shape. The color of the oval will change from blue to yellow and then red depending upon the criticalness of the situation. Air seeder tank status represents the fill status of the tank in percentage. Blockage, seed spillage, and plugging's color will change from green to red if there will be any indication of any blockage, seed spillage or plugging's in the air seeder system.

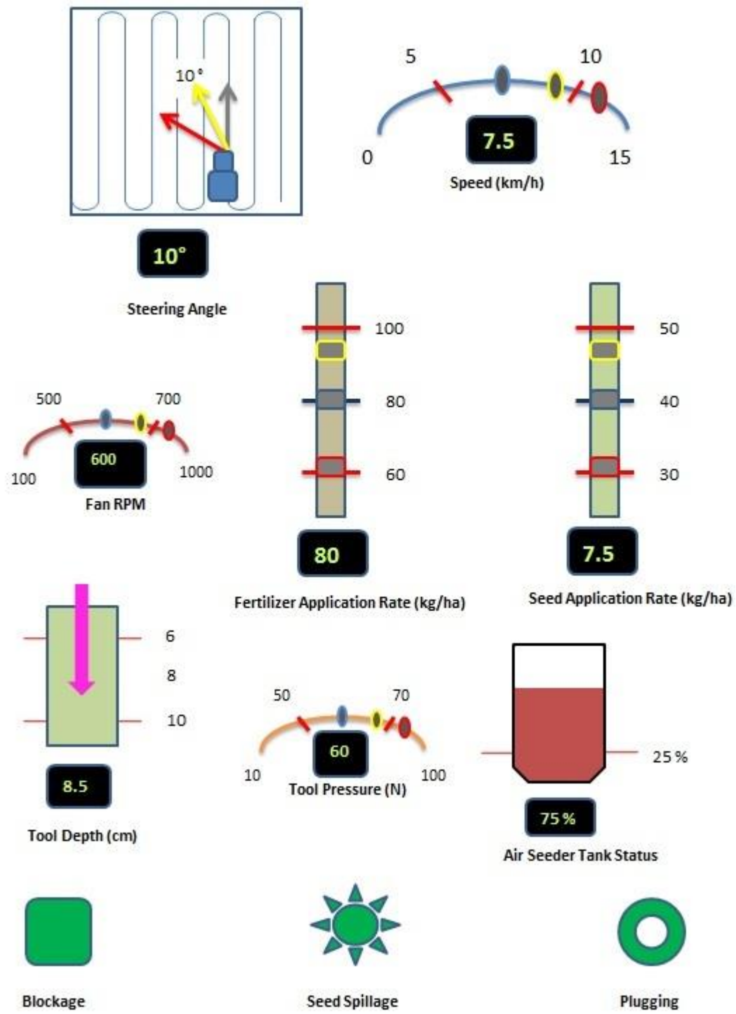


Figure 4-1. Earlier conceptual interface design for TAS-DS representing steering angle, speed, fan rpm, fertilizer application rate, speed application rate, tool depth, tool pressure, air seeder tanks status, blockage, seed spillage and plugging.

Cognitive and perceptual psychology principles (as explained in section 2.3) were further explored and applied to improve the initial interface design further. An air seeder driver interface was re-designed for 13 parameters: seed level status (tank level), fertilizer level status (tank level), fan RPM, seed application rate, fertilizer application rate, tool depth (represents seed and fertilizer depth), tool pressure, blockage status, desired path of the unit, desired location of the unit and current speed of the unit. Figure 4-2 shows the complete conceptual interface. In the following sections (4.1 to 4.8), the design of each interface element and the application of various design principles are discussed individually.

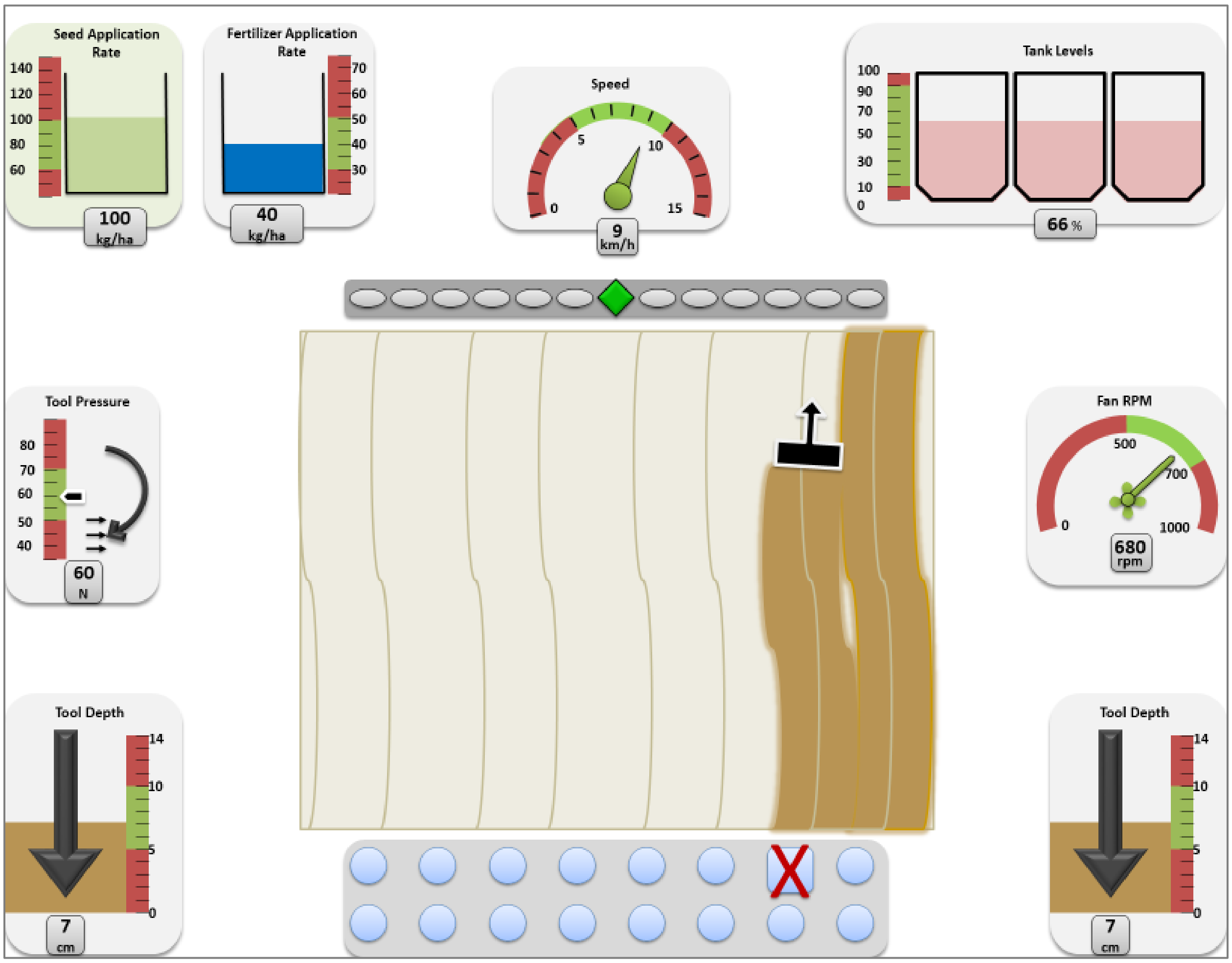


Figure 4-2. Conceptual driver interface design for TAS-DS designed on the basis of SA and other human factors principles.

In this display, items related to the movement of the unit such as travel speed, travel direction, the location of the machine and guidance bar are arranged near each other in the middle. Tool depth and blockage status are organized at the bottom, represented as being rear-mounted items. Other items such as seed application rate, fertilizer application rate, tank levels, tool pressure, and fan RPM are represented on the interface as per their size and symmetry. As much as possible, the effort has been made to use colors as recommended by human factors, cognitive and perceptual psychology principles to design air seeder driver interface as a universal design (i.e., equally usable for color vision deficient people). The grey scale version of this air seeder driver interface can be seen in Figure 4-3. On the grey scale, all the items and details appear as distinct and vivid.

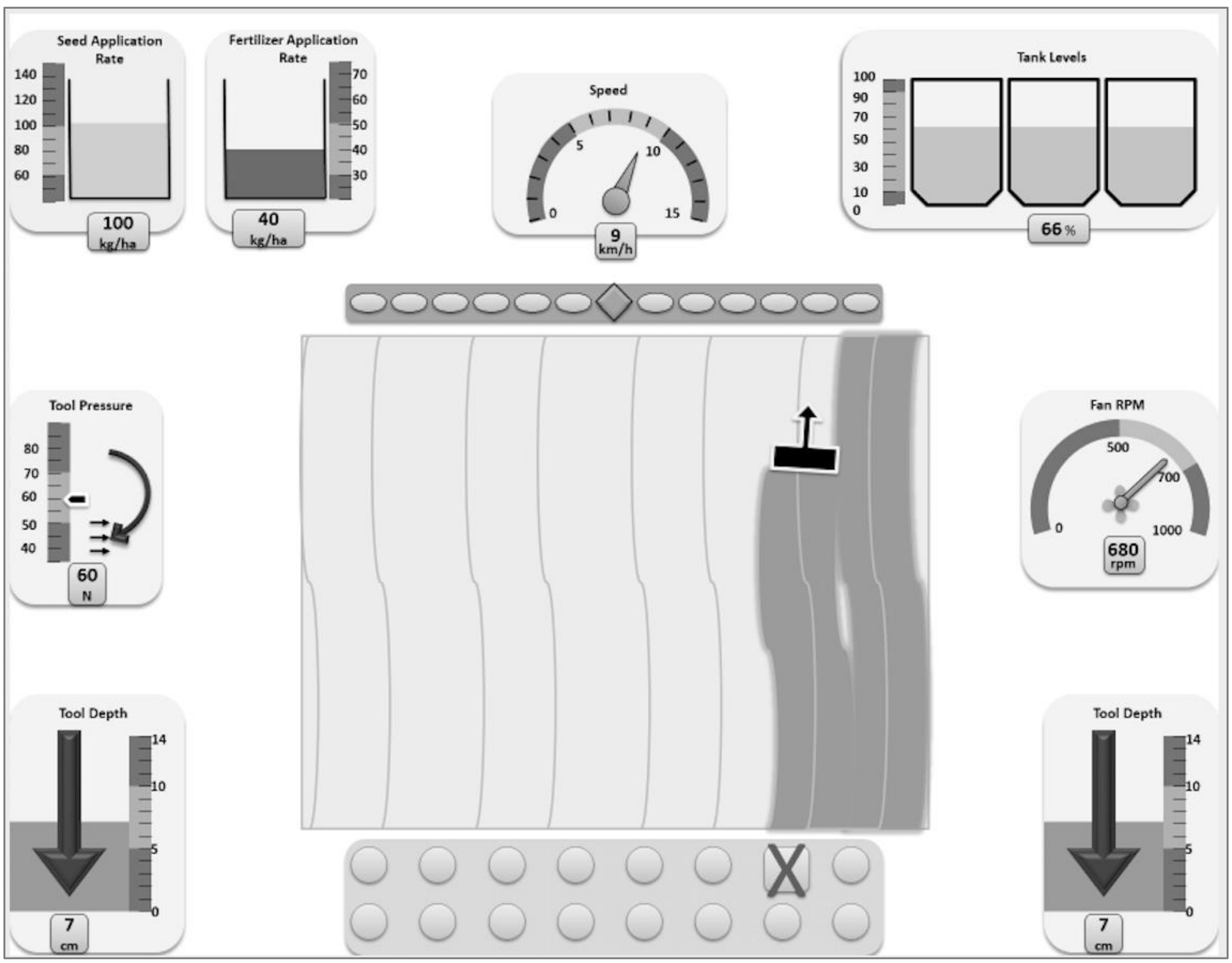


Figure 4-3. Greyscale presentation of the conceptual driver interface for TAS-DS designed based on SA and other human factors principles.

4.1 TRACTOR SPEED

This conventional and unambiguous speed indicator (Figure 4-4) provides tractor air seeder speed in both analog and digital form.

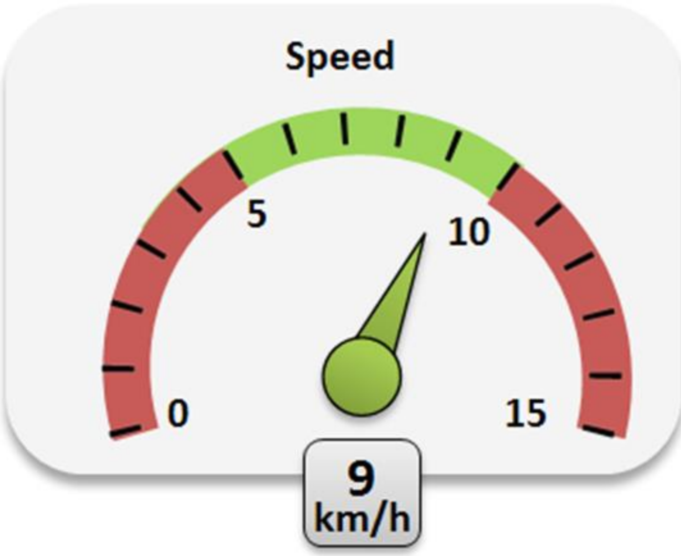


Figure 4-4. Speed indicator of the unit.

It helps the user to efficiently perceive (level 1 SA) the current traveling speed of the machine in the field. The speed indicator has been divided into red and green color zones based on the desirable (5 to 10 km/h) and non-desirable (0-5 and 10-15 km/h) speed ranges. This representation will help the user to comprehend (level 2 SA) the value of the current speed in relation to the operational goals, and project (level 3 SA) the near future state. For example, when the speed needle is in the red color zone, the operator can immediately perceive (level 1 SA) and comprehend (level 2 SA) that he/she is at a non-desirable speed. Furthermore, the movement of the needle along with the colored strip helps the user to project its near future state (level 3 SA) based on the direction/trend of speed (increasing or decreasing). In addition to the SA principles, guidelines related to cognitive and perceptual psychology are also used in the design of the speed presentation. For the convenience of the reader, the majority of the principles used in the design of the speed indicator are listed in Table 4-1.

Table 4-1. Description of the application of SA and cognitive psychology principles in speed indicator design.

Principles	Application of the principles
Focus on the goals	Knowing the speed of the unit (tractor and air seeder) is one of the critical SA information requirements for the operator and support by the GDTA. So, the presentation of the speed is related to the goals of the operator.
Make it easy for the user	Presentation of the speed is entirely conventional and easy to understand to help in level 2 type of SA.
Assist in level 3 SA	Movement of the speed pointer can help the operator to anticipate the trend of the speed, i.e., increment or decrement in the speed.
Support trade-offs between goal-driven and data-driven processing	Location of the speed pointer on green zone supports the goal-driven approach which indicates that the speed of the unit is in-line with the prescribed limits (speed goals). As soon as the speed pointer enters into the red zone, the user can comprehend that he/she is in the undesirable zone.
Make critical cues salient	The red and green coloring of the speed indicator supports this principle.
Avoid excessive features	The goals of the operator support the presence of the speed indicator. So this is an essential feature. Furthermore, the design of the speed indicator itself is also simple and does not have any unnecessary features.
Map system functions to the goals and mental models of user	The requirement of speed indicator is supported by the goals of the operator. Moreover, conventional look and feel of the indicator are also in-line with the mental model of the operator.
Minimize display density but not at the cost of coherence	There is adequate free space (white space) maintained in the speed indicator design as well as in the overall interface which is showing all the parameters.
People should not rely exclusively on alarms; provide projection support	Movement of the speed pointer will help the user to judge the trend/projection of the speed.
Provide SA support rather than a decision	During the simulation and evaluation of all parameters, no direct decision-making advice is provided to the user. The users were required to respond with their own free will based on their SA.

Principles	Application of the principles
Transparency in automation	Addition of digital indicator with units of speed (km/h) along with the analog indicator will keep no void in the perception of the information by the user.
Weber's Law	To indicate any critical changes to the user during the simulation, sufficient noticeable stimuli are provided for easy recognition of the changes, such as movement of a needle, movement of a progress bar, and change in the color and shape of a blockage indicator.
Yerkes-Dodson law	Movement of the needle or changing readings of speed indicator (digitally) can be considered as arousal for the operator along with any other noticeable stimulus like a change in the color or shape of the speed pointer.
Gestalt principle of continuity	Though needle pointer while moving is not touching the marks on the red-green scale, still it is perceived as representing the corresponding value on the scale.
Gestalt principle of symmetry	Overall speed indicator is made of several shapes, lines and colors, though perceived as symmetrical and meaningful graphic to represent the speed of the air seeder unit.
Gestalt principle of common fate	Movement of the speed pointer, as well as a change of the readings in digital form, act in synchronization, so they are perceived as related.
Gestalt principle of figure/ground	The color scheme of the scale, needle and digital scale has been made reasonably prominent than the background to provide a justified visible distinction between foreground and background.
Use of most distinct colors	Uses of most distinct and pure colors are avoided to reduce the shimmering and annoying appearance.
Using brightness and saturation to support distinction among colors	Colors used in the design are distinct enough regarding hue, saturation, and brightness. This distinction can be recognized by viewing the image on the grey scale.
Colors for color-blind people	To work this interface as 'Universal Design' high consideration has been made to use color-blind friendly color scheme. This can be judged by viewing the design on the grey scale.
For differentiation do not rely on colors alone	In addition to the presence of colored scale; change in color of the needle, and movement of the needle along with the scale helps the user to track the progress.
Paleness, size, and separation	Ample consideration has been made to use appropriate size, paleness, and separation for the speed indicator elements.
How to attract user's attention	Change of the color needle from green to red and vice versa, along with the movement of the needle helps in catching users attention.
Consistency	Design of speed indicator is conventional, logical and familiar.
Feedback	Constant feedback in the form of movement of the speed pointer in the green and red zone is provided to the operator.

Principles	Application of the principles
Affordances and convention	As indicated above the design of the speed indicator is conventional, consistent and logical.
Simplicity and structure	This speed indicator is simple in appearance and functionality. Organization of the speed indicator on the interface along with other parameters is also appropriate for the mental models of the operators. Speed indicator has been placed in the middle of the top row to give the operator a familiar feel of the driving scenario.

4.2 THE DESIRED PATH, TRAVEL DIRECTION, AND LOCATION OF THE UNIT ALONG WITH GUIDANCE BAR

A centrally-located field-overview showing the travel direction, desired path and location of the unit inside the field (Figure 4-5) is provided. This view is intended to facilitate global SA by providing both spatial and temporal information about the unit. This view can provide an overview of the area covered and uncovered, help in estimating (level 3 SA) the time required in completing the uncovered area. To maintain precise guidance (steering control), a typical ‘guidance bar’ is provided just above the field overview. A green diamond shape on the bar depicts the desired straight line traveling of the unit. If the unit moves either left or right (i.e., in a non-desirable direction), the green diamond will also move to help in perceiving (level 1 SA), comprehending (level 2 SA) and projecting (level 3 SA) the status of the unit. The further the green diamond appears from the central location, the higher the steering error would be (level 3 projection).

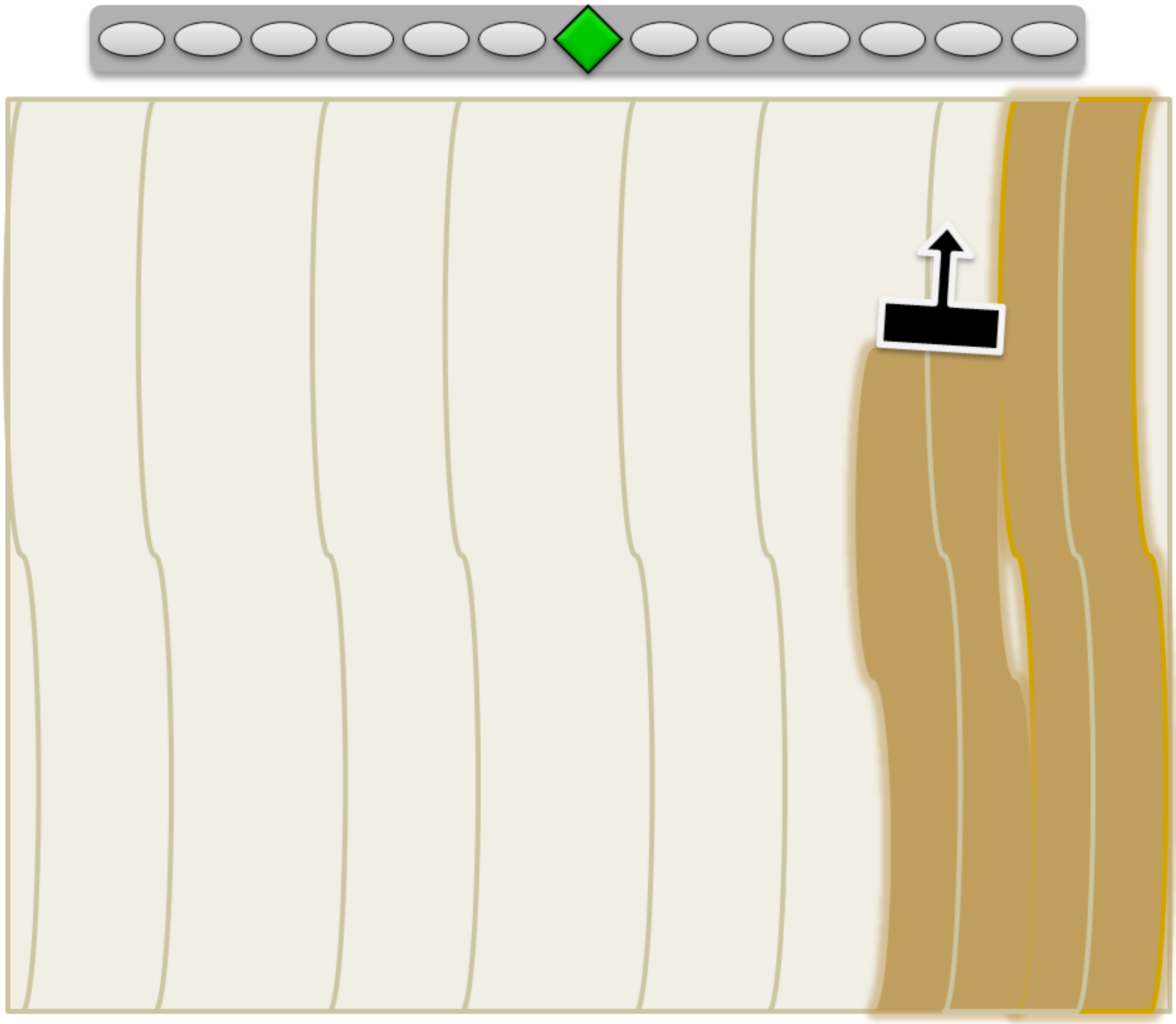


Figure 4-5. Spatial and temporal representation of the tractor air seeder unit inside the field.

For the convenience of the reader, the principles from section 4 and 5, which are used in the design of spatial and temporal representation of the tractor air seeder unit are listed in Table 4-2.

Table 4-2. Description of the application of SA and cognitive psychology principles for the design of spatial and temporal presentation tractor air seeder unit.

Principles	Application of the principles
Focus on the goals	Knowing the spatial and temporal position of the unit (tractor and air seeder) along with the guidance information, intended path of travel and area covered are critical SA information requirements

Principles	Application of the principles
	for the operator. These requirements are very well supported in the GDTA.
Make it easy for the user	Presentation of the unit, guidance information, traveling path and area covered/uncovered are simple in understanding to help in level 2 type of SA.
Assist in level 3 SA	Movement of the green diamond on the guidance bar can help the operator to anticipate the trend of turning of the steering wheel. Also, the movement and direction of the travel of the vehicle (black icon) can be very well anticipated.
Maintain the global SA	This overview of the movement of the tractor air seeder unit inside the field provides big-picture to the user and helps in maintain global SA.
Support trade-offs between goal-driven and data-driven processing	If the unit is moving on the intended path, then the diamond shape on the guidance bar will stay in the middle. If the unit deviates from the intended path, the green diamond will move left or right as well as a change in color from green to yellow or red depending upon the criticalness of the situation. So, the user will be well aware of both cases, i.e., when following the goals (green diamond) or when deviation from the desired performance (yellow or red diamond).
Make critical cues salient	Green, yellow and red coloring of the diamond shape on the guidance bar as well as its movement from the desired central location supports this principle.
Avoid excessive features	The GDTA supports the presence of all the mentioned parameters (guidance bar, the location of the unit, the area covered/uncovered). So, all these are essential features. Furthermore, the design of these all parameters is also simple enough and does not have any unnecessary features.
Map system functions to the goals and mental models of user	The goals of the operator support requirement of guidance bar, the location of unit, area covered/uncovered. Moreover, they have a simple and logical presentation to map with the mental model of the operator.
Minimize display density but not at the cost of coherence	There is adequate free space (white space) maintained during the design without compromising with the essential SA information needs of the operator.
People should not rely exclusively on alarms; provide projection support	Movement of the green diamond on the guidance bar can help the operator to anticipate the trend of turning of the steering wheel. Also, the movement and direction of the travel of the vehicle (black icon) can be very well anticipated.
Provide SA support rather than a decision	During the simulation and evaluation of all parameters, no direct decision-making advice is provided to the user. The users were required to respond with their own free will based on their SA.

Principles	Application of the principles
Transparency in automation	Parameters presented in this spatial and temporal overview are quite explicit, logical and straightforward.
Weber's Law	Green, yellow/ red (not shown in the present view) colors used in the diamond are different enough for the user to perceive the change.
Yerkes-Dodson law	Movement of the vehicle (black icon), movement on the guidance bar and appearance/change in the covered area represent reasonable arousal for the operator.
Gestalt principle of symmetry	Overall all the parameters (guidance bar, the location of the unit, the area covered/uncovered) are made of several shapes, lines, and colors; though perceived as symmetrical and meaningful graphics to represent the desired information.
Gestalt principle of common fate	Movement of the green diamond and vehicle arrow acts in synchronization, so they are perceived as related.
Gestalt principle of figure/ground	The color scheme of the guidance bar, the location of the unit, the area covered/uncovered has been made reasonably prominent than the background to provide a justified visible distinction between foreground and background.
Use of most distinct colors	Uses of most distinct and pure colors are avoided to reduce the shimmering and annoying appearance.
Using brightness and saturation to support distinction among colors	Colors used in the design are distinct enough regarding hue, saturation, and brightness. This distinction can be recognized by viewing the image on the grey scale.
Colors for color-blind people	To work this interface as 'Universal Design' high consideration has been made to use a color-blind friendly color scheme. This can be judged by viewing the design on the grey scale.
For differentiation do not rely on colors alone	For the guidance bar, change in shape (oval to diamond) and location (from center to left and right) are used to support this principle.
Paleness, size, and separation	Ample consideration has been made to use appropriate size, paleness, and separation for the various elements in the design.
How to attract user's attention	Change in the color of the guidance bar elements (green to yellow and red and vice versa) along with the change in the shape (oval to green and vice versa) have used to attract the adequate attention of the user.
Consistency	Design of overview and guidance parameter is conventional, logical and familiar.
Feedback	Constant feedback in the form of movement of the vehicle icon, the area covered, and guidance bar activity is provided.
Affordances and convention	Design of the guidance bar, the location of the unit, the area covered/uncovered is conventional, consistent and logical.
Simplicity and structure	Guidance bar, the location of the unit, the area covered/uncovered is simple in appearance and functionality.

Principles	Application of the principles
	Organization of the all these elements along with other interface elements (such as tool seed application and others) is also appropriate with the mental models of the operators. This global overview is placed in the middle of the interface.

4.3 BLOCKAGE INDICATION IN SEED AND FERTILIZER DISTRIBUTION TUBES

Two rows of 16 circles represent the blockage status of the seeding and fertilizer units (Figure 4-6). A red “X” appears inside a square to show that blockage has been detected. A change in status is represented by involving both a change in shape and the appearance of a new mark (X) with more eye-catching color to enable a better perception of the situation.

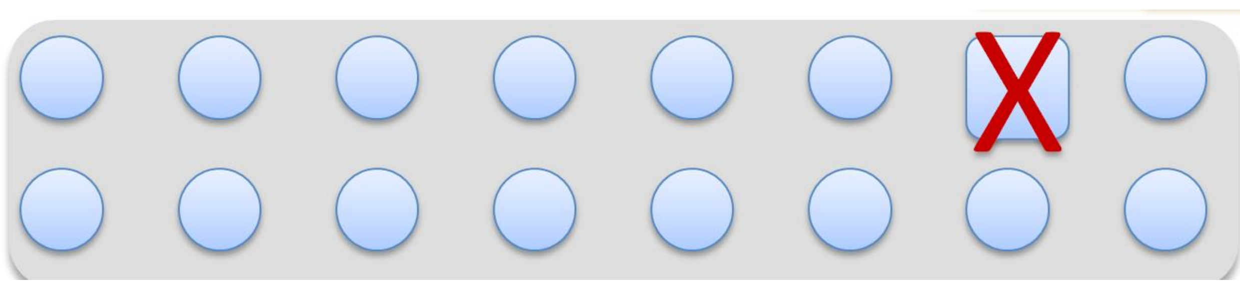


Figure 4-6. Blockage indication inside the seed and fertilizer distribution tubes of air seeder.

For the convenience of the reader, the principles used in the design of blockage status inside the seed and fertilizer distribution tubes of the tractor air seeder unit are listed in Table 4-3.

Table 4-3. Description of the application of SA and cognitive psychology principles for the design of blockage status inside the seed and fertilizer distribution tubes.

Principles	Application of the principles
Focus on the goals	Knowing the about the blockage in seed and fertilizer tubes is critical SA information requirements for the operator. This requirement is well supported in the GDTA.
Make it easy for the user	Presentation of the blockage is simple in understanding to help in level 2 type of SA.

Principles	Application of the principles
Maintain the global SA	This blockage status represents blockage in all seed and fertilizer tubes so helps in providing big-picture to the user and helps in maintain global SA.
Support trade-offs between goal-driven and data-driven processing	An indication of blockage (red X mark) helps the user to divert their attention from other goals to attend the blockage problem.
Make critical cues salient	Red X mark on the square-shaped background makes the changes salient to help in better perception.
Avoid excessive features	The GDTA supports the presence of the blockage indication. So, it is an essential feature. Furthermore, the design of the blockage indication parameters is also simple enough and does not possess any unnecessary features.
Map system functions to the goals and mental models of user	The requirement of the blockage indication is supported by the goals of the operator. Moreover, they have a simple and logical presentation to map with the mental model of the operator.
Minimize display density but not at the cost of coherence	There is adequate free space (white space) maintained during the design without compromising with essential SA information needs of the operator.
Provide SA support rather than a decision	During the simulation and evaluation of all parameters, no direct decision-making advice is provided to the user. The users were required to respond with their own free will based on their SA.
Transparency in automation	Presentation of blockage indication is quite explicit, logical and straightforward.
Weber's Law	The appearance of the Red X mark and change in background shape from circle acts as a noticeable stimulus to perceive the change in the situation.
Yerkes-Dodson law	The appearance of the Red X mark and change in background shape from circle also act as appropriate arousal for the task.
Gestalt principle of similarity	All similar blank circles are perceived as related which represent the non-blockage in the seed and fertilizer tubes.
Gestalt principle of figure/ground	The color scheme of the blockage indicator has been made reasonably prominent than the background to provide a justified visible distinction between foreground and background.
Use of most distinct colors	Uses of most distinct and pure colors are avoided to reduce the shimmering and annoying appearance.
Using brightness and saturation to support distinction among colors	Red and light blue colors used in the design are distinct enough regarding hue, saturation, and brightness. This can be recognized by viewing the image on the grey scale.

Principles	Application of the principles
Colors for color-blind people	To work this interface as 'Universal Design' high consideration has been made to use a color-blind friendly color scheme. This can be judged by viewing the design on the grey scale.
For differentiation do not rely on colors alone	The appearance of X mark and change in shape from a circle to the square while indicating blockage supports this principle.
Paleness, size, and separation	Ample consideration has been made to use appropriate size, paleness, and separation for the blockage indicator elements.
How to attract user's attention	Red X mark on square shaped background acts as a noticeable stimulus to attract user's attention.
Consistency	Design of blockage indicator is conventional, logical and familiar.
Feedback	Constant feedback in the form of blockage indication (red X mark) or non-blockage indication (i.e., when clear circles are present) is provided.
Affordances and convention	Design of the blockage indicator is conventional, consistent and logical.
Simplicity and structure	Blockage indicator is simple in appearance and functionality. Organization of this blockage indicator along with other parameters is also entirely appropriate with the mental models of the operators. This blockage indicator is placed beneath the bottom of the field overview which represents the tractor movement inside the field.

4.4 SEED AND FERTILIZER APPLICATION RATES

Representation of both seed and fertilizer application rates are also made according to SA and human factors principles (Figure 4-7). Presence of the scales along with a change of the levels of the application rates will help the user to maintain SA at all three levels (i.e., perception, comprehension and projection) without placing too much demand on cognitive resources.

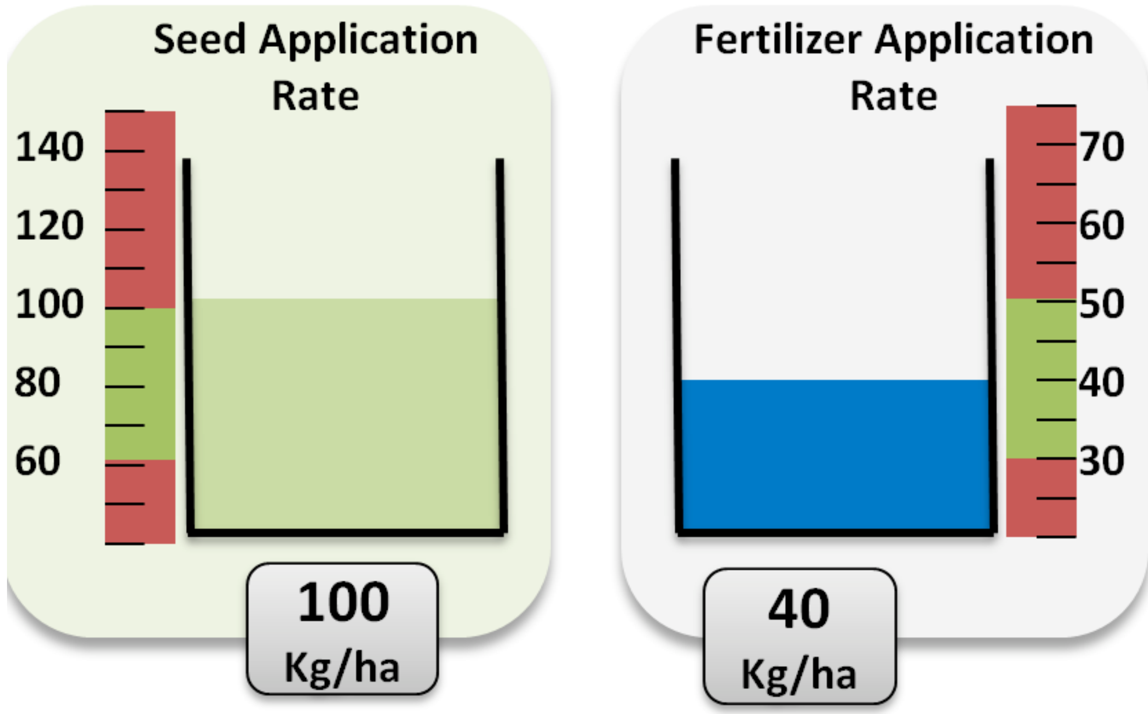


Figure 4-7. Seed application rate and fertilizer application rate of the tractor air seeder system.

For the convenience of the reader, the principles used in the design of seed and fertilizer application rate of the tractor air seeder unit are listed in Table 4-4.

Table 4-4. Description of the application of SA and cognitive psychology principles for the design of seed application rate and fertilizer application rate.

Principles	Application of the principles
Focus on the goals	Information about the seed and fertilizer application rate is one of the critical SA information requirements for the operator and support by the GDTA.
Make it easy for the user	Presentation of the items is simple and easy to understand to help in level 2 type of SA.
Assist in level 3 SA	Raising or lowering levels of seed and fertilizer application rate can help the operator to anticipate the trend, i.e., increment or decrement in the application rate.
Support trade-offs between goal-driven and data-driven processing	The positioning of the level in line with the green zone of the scale supports the goal-driven approach, which indicates that the application rate of the unit is within the prescribed limits. As soon as the application rate enters into the red zone of the scale, the user can comprehend that he/she is out of the prescribed zone, and should act accordingly.

Principles	Application of the principles
Make critical cues salient	The red and green coloring of the scale supports this principle.
Avoid excessive features	The goals of the operator support presence of the application rates, hence are essential. Furthermore, the design of the seed and fertilizer application rates is simple and does not have any unnecessary features.
Map system functions to the goals and mental models of user	The requirement of the application rates is supported by the goals of the operator. Moreover, logical and straightforward look and feel of the parameters can easily tune with the mental models of the operator.
Minimize display density but not at the cost of coherence	There is adequate free space (white space) maintained for the seed and fertilizer application rates design.
People should not rely exclusively on alarms; provide projection support	Raising or lowering levels of seed and fertilizer application rate can help the operator to anticipate the trend, i.e., increment or decrement in the application rate.
Provide SA support rather than a decision	During the simulation and evaluation of all parameters, no direct decision-making advice is provided to the user. The users were required to respond with their own free will based on their SA.
Transparency in automation	In addition to the digital reading along with units of the application rate (kg/ha), raising and lowering the levels would ensure the proper perception of the information by the user.
Weber's Law	Red and green colors used in the scale are different enough for the user to perceive the change. Also, a noticeable stimulus (such as a change in the color of the liquid level before entering into the critical red zone) may also be provided during the implementation of the design.
Yerkes-Dodson law	Movement of the levels or changing readings of the numerical display can be considered as reasonable arousal for the operator.
Gestalt principle of continuity	Though levels inside the container while moving are not touching the marks on the red-green scale, still they are perceived as representing the corresponding value on the scale.
Gestalt principle of symmetry	Overall application rate indicator is made of several shapes, lines, and colors, though perceived as symmetrical and meaningful graphic to represent the speed of the air seeder unit.
Gestalt principle of common fate	Movement of the levels, as well as a change of the readings, act in synchronization, so they are perceived as related.
Gestalt principle of figure/ground	The color scheme of the scale, levels inside the container and digital scale have been made reasonably prominent than the

Principles	Application of the principles
	background to provide a justified visible distinction between foreground and background.
Use of most distinct colors	Uses of most distinct and pure colors are avoided to reduce the shimmering and annoying appearance.
Using brightness and saturation to support distinction among colors	Colors used in the design are distinct enough regarding hue, saturation, and brightness. This can be recognized by viewing the image on the grey scale.
Colors for color-blind people	To work this interface as 'Universal Design' high consideration has been made to use a color-blind friendly color scheme. This can be judged by viewing the design on the grey scale.
For differentiation do not rely on colors alone	During the implementation of this design in experiment sessions, both changes in levels inside the container, and numerical readings beneath are provided to perceive any change in the situation.
Paleness, size, and separation	Ample consideration has been made to use appropriate size, paleness, and separation of the elements in the design.
How to attract user's attention	Movement of the levels and change in the colors while progressing through red or green area of the color scale would be used to catch the proper attention of the user.
Consistency	Design of application rate is simple, logical and familiar.
Feedback	Constant feedback in the form of movement of the levels inside the container and changing of numerical readings beneath provides constant feedback to the operator.
Affordances and convention	As indicated above the design of the application rate is logical and simple.
Simplicity and structure	Seed and fertilizer application rates are simple in appearance and functionality. Both seed and fertilizer application rates are organized near each other to appear as somewhat related and similar.

4.5 TOOL DEPTH

Tool depth represents the depth of the tillage tool inside the ground which ultimately represents the seed and fertilizer placement depth inside the soil (Figure 4-8). Two tool depth parameters are displayed on the air seeder driver interface to represent the tool depth of the left half and right half of the air seeder tools. The color-coded scale along with the graphic presentation of the tool depth will help the operator to maintain SA at all three levels.

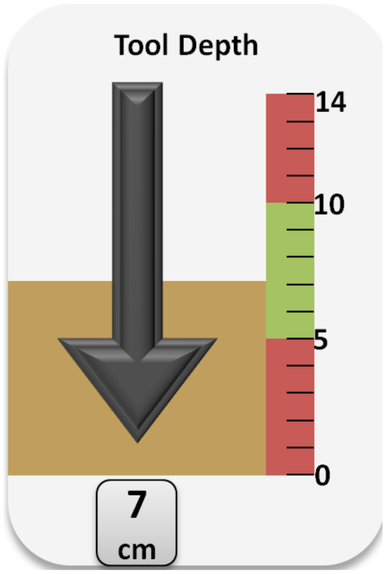


Figure 4-8. Tool depth of the tillage tool inside the soil.

For the convenience of the reader, the principles used in the design of tool depth of the tractor air seeder unit are listed in Table 4-5.

Table 4-5. Description of the application of SA and cognitive psychology principles for the design of tool depth.

Principles	Application of the principles
Focus on the goals	Information about the tool depth is one of the critical SA information requirements for the operator and is support by the GDTA. So, this parameter is related to the goals of the operator.
Make it easy for the user	Presentation of the item is simple and easy to understand to help in level 2 type of SA.
Assist in level 3 SA	Raising or lowering level of tool depth can help the operator to anticipate the trend, i.e., increment or decrement in the tool depth.
Support trade-offs between goal-driven and data-driven processing	Movement of the level indicator in line with the green zone of the scale supports the goal-driven approach, which indicates that the tool depth of the unit is within the prescribed limits. As soon as the tool depth will enter into the red zone of the scale user can comprehend that he/she is out of the prescribed limit and should act accordingly.
Make critical cues salient	The red and green coloring of the scale supports this principle.
Avoid excessive features	The need of tool depth is supported by the goals of the operator, so this is an essential feature. Furthermore, the design of the tool depth is simple and does not have any unnecessary features.

Principles	Application of the principles
Map system functions to the goals and mental models of user	The requirement of the tool depth is supported by the goals of the operator. Logical and straightforward look and feel of the parameters can easily tune with the mental model of the operator.
Minimize display density but not at the cost of coherence	There is adequate free space (white space) maintained for the design of tool depth as well as in the overall interface which is showing all the parameters.
People should not rely exclusively on alarms; provide projection support	Raising or lowering level of tool depth can help the operator to anticipate the trend, i.e., increment or decrement in the tool depth.
Provide SA support rather than a decision	During the simulation and evaluation of all parameters, no direct decision-making advice is provided to the user. The users were required to respond with their own free will based on their SA.
Transparency in automation	Addition of the digital reading with the units of the tool depth (cm), along with raising or lowering levels will keep no void in the perception of the information by the user.
Weber's Law	Red and green colors used in the scale are different enough for the user to perceive the change. Also, a noticeable stimulus (such as a change in the color of the level before entering into the critical red zone) may also be provided during the implementation of the design.
Yerkes-Dodson law	Movement of the tool depth level or change in readings of the numerical display can be considered as arousal for the operator along with any other noticeable stimulus like a change in the color of the levels.
Gestalt principle of continuity	Though levels while moving are not touching the marks on the red-green scale, still it is perceived as representing the corresponding value on the scale.
Gestalt principle of symmetry	Overall tool depth indicator is made of several shapes, lines, and colors, though perceived as symmetrical and meaningful graphic to represent the speed of the air seeder unit.
Gestalt principle of common fate	Movement of the level, as well as a change of the readings in digital form, act in synchronization, so they are perceived as related.
Gestalt principle of figure/ground	The color scheme of the scale, level, and digital scale has been made reasonably prominent than the background to provide a justified visible distinction between foreground and background.
Use of most distinct colors	Uses of most distinct and pure colors are avoided to reduce the shimmering and annoying appearance.

Principles	Application of the principles
Using brightness and saturation to support distinction among colors	Colors used in the design are distinct enough regarding hue, saturation, and brightness. This can be recognized by viewing the image on the grey scale.
Colors for color-blind people	To work this interface as 'Universal Design' high consideration has been made to use a color-blind friendly color scheme. This can be judged by viewing the design on the grey scale.
For differentiation do not rely on colors alone	During the implementation of this design in experiment sessions, both changes in level adjacent to the scale and numerical readings beneath are provided to perceive any change in the situation.
Paleness, size, and separation	Ample consideration has been made to use appropriate size, paleness, and separation of the elements in the design.
How to attract user's attention	To attract user's attention during some critical situations (e.g., if errors are lasting more than 1 minute), wiggling object or beep sound can be used.
Consistency	Design of application rate is simple, logical and familiar.
Feedback	Constant feedback in the form of movement of the level and changing of numerical readings beneath provides constant feedback to the operator.
Affordances and convention	As indicated above the design of the tool depth is logical and straightforward.
Simplicity and structure	Tool depth is simple in appearance and functionality. It is organized at the lower left and right corners of the interface to represent left-side and right-side tools depth.

4.6 SEED AND FERTILIZER LEVEL STATUS (TANK LEVEL)

Tank level represents the seed and fertilizer quantity inside the tank (Figure 4-9). Practically, seed and fertilizer quantity in the tanks may be different. However, in this illustration, both tank levels have been shown at the same level. A color-coded scale along with the graphical image of a tank will help the operator to maintain SA for all three levels.

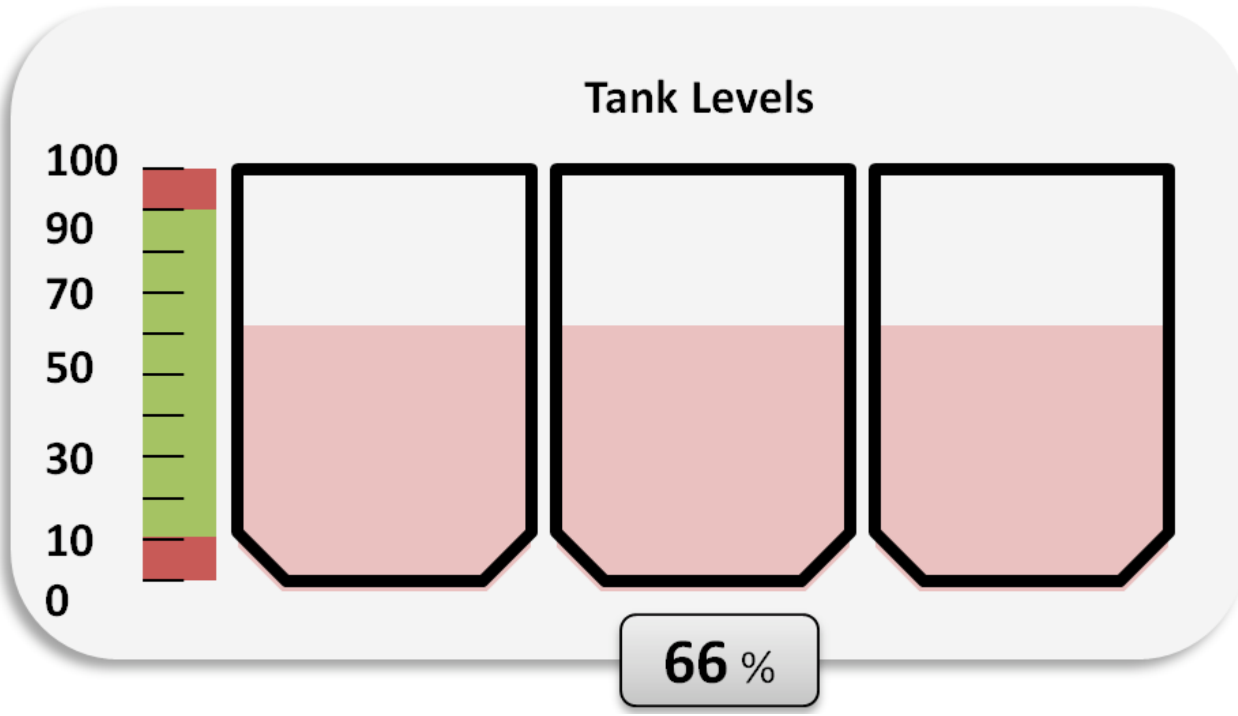


Figure 4-9. Status of the quantity of seed and fertilizer remaining inside the tanks.

For the convenience of the reader, the principles used in the design of tank levels for the status of seed and fertilizer quantity of the tractor air seeder unit are listed in Table 4-6.

Table 4-6. Description of the application of SA and cognitive psychology principles for the design of tank levels.

Principles	Application of the principles
Focus on the goals	Information about the quantity of seed and fertilizer remaining inside the tanks is one of the critical SA information requirements for the operator and support by the GDTA. So, these parameters are related to the goals of the operator.
Make it easy for the user	Presentation of the items is simple and easy to understand to help in level 2 type of SA.
Assist in level 3 SA	Lowering levels of seed and fertilizer can help the operator to anticipate the trend, i.e., decrement in quantity.
Support trade-offs between goal-driven and data-driven processing	The positioning of the level in line with the green zone of the scale supports the goal-driven approach, which indicates that the tank levels are within the prescribed limits. As soon as the tank-levels enter into the red zone of the scale, the user can

Principles	Application of the principles
	comprehend that he/she is out of the prescribed zone, and should act accordingly.
Make critical cues salient	The red and green coloring of the scale supports this principle.
Avoid excessive features	The goals of the operator support presence of the seed and fertilizer quantity indication inside the tanks. So, these are essential features. Furthermore, the design of the seed and fertilizer quantity indications is simple and does not have any unnecessary features.
Map system functions to the goals and mental models of user	The goals of the operator support the requirement of seed and fertilizer quantity indication inside the tanks. Moreover, a logical and straightforward illustration of the parameter can be easily tuned with the mental model of the operator.
Minimize display density but not at the cost of coherence	There is adequate free space (white space) maintained for the design of seed and fertilizer quantity indication inside the tanks as well as in the overall interface which is showing all the parameters.
People should not rely exclusively on alarms; provide projection support	Lowering levels of seed and fertilizer quantity indication inside the tanks can help the operator to anticipate the trend, i.e., decrement in the quantity of seed and fertilizer inside the tanks.
Provide SA support rather than a decision	During the simulation and evaluation of all parameters, no direct decision-making advice is provided to the user. The users were required to respond with their own free will based on their SA
Transparency in automation	Addition of the digital reading (%) of seed and fertilizer quantity inside the tanks along with the analog indication in the form of lowering level of seed and fertilizer quantity would ensure the proper perception of the information by the user.
Weber's Law	Red and green colors used in the scale are different enough for the user to perceive the change. Also, a noticeable stimulus (such as change in the color of the level before entering the critical red zone) may also be provided during the implementation of the design.
Yerkes-Dodson law	Movement of the levels or changing readings of the numerical display can be considered as arousal for the operator along with any other noticeable stimulus like a change in the color of the levels.
Gestalt principle of continuity	Though the seed and fertilizer levels inside the tanks while moving are not touching the marks on the red-green scale, still they are perceived as representing the corresponding value on the scale.
Gestalt principle of symmetry	Overall seed and fertilizer quantity indication inside the tanks is made of several shapes, lines and colors, though perceived as

Principles	Application of the principles
	symmetrical and meaningful graphic to represent the speed of the air seeder unit.
Gestalt principle of common fate	Movement of the levels, as well as a change of the numerical readings, act in synchronization, so they are perceived as related.
Gestalt principle of figure/ground	The color scheme of the scale, levels inside the tanks and digital scale has been made reasonably prominent than the background to provide a justified visible distinction between foreground and background.
Use of most distinct colors	Uses of most distinct and pure colors are avoided to reduce the shimmering and annoying appearance.
Using brightness and saturation to support distinction among colors	Colors used in the design are distinct enough regarding hue, saturation, and brightness. This distinction can be recognized by viewing the image on the grey scale.
Colors for color-blind people	To work this interface as ‘Universal Design’ high consideration has been made to use a color-blind friendly color scheme. This can be judged by viewing the design on the grey scale.
For differentiation do not rely on colors alone	During the implementation of this design in experiment sessions, both changes in levels inside the container and numerical readings beneath are provided to perceive any change in the situation vividly.
Paleness, size, and separation	Ample consideration has been made to use appropriate size, paleness, and separation of the elements in the design.
How to attract user’s attention	Movement of the levels, and change in the colors while progressing through red or green area of the color scale would be used to catch the adequate attention of the user.
Consistency	Design of the tank level elements is simple, logical and familiar.
Feedback	Constant feedback in the form of movement of the levels inside the container and changing of numerical readings beneath provides constant feedback to the operator.
Affordances and convention	As indicated above the design of the application rate is logical and straightforward.
Simplicity and structure	Seed and fertilizer quantity indication inside the tanks are simple in appearance and functionality. This parameter is organized in the top right corner due to its size and symmetry on the interface.

4.7 FAN RPM

Fan RPM represents the speed of the air seeder fan, in revolutions per minute, in both digital and analog form (Figure 4-10). A color-coded pattern helps the operator to perceive, comprehend and foresee the fan rpm in a user-friendly manner.

Figure 4-10. Fan RPM of the air seeder fan.

For the convenience of the reader, the principles used in the design of fan rpm of the tractor air seeder unit are listed in Table 4-7.

Table 4-7. Description of the application of SA and cognitive psychology principles for the design of fan rpm.

Principles	Application of the principles
Focus on the goals	Knowing about the fan rpm the unit (tractor and air seeder) is one of the critical SA information requirements for the operator and support by the GDTA. So the presentation of the fan rpm is related to the goals of the operator.
Make it easy for the user	Presentation of the fan rpm is entirely conventional and easy to understand to help in level 2 type of SA.
Assist in level 3 SA	Movement of the fan rpm pointer can help the operator to anticipate the trend of the fan rpm, i.e., increment or decrement in the fan rpm.
Support trade-offs between goal-driven and data-driven processing	The position of the fan rpm pointer inline with green zone supports the goal-driven approach which indicates that fan rpm of the unit is within the prescribed limits (fan rpm goals). As soon as the fan rpm pointer enters into the red zone, the user can comprehend that he/she is of out recommended limits. To further facilitate the perception of the user in the red zone, entry of the fan rpm pointer can be made salient in the actual design by implementing change in of the color the fan rpm pointer.

Make critical cues salient	The red and green coloring of the fan rpm indicator supports this principle.
Avoid excessive features	Presence of the fan rpm indicator is supported by the goals of the operator. So, this is an essential feature. Furthermore, the design of the fan rpm indicator itself is also simple and does not have any excessive features
Map system functions to the goals and mental models of user	The requirement of the fan rpm indicator is supported by the goals of the operator. Moreover, conventional look and feel of the indicator are also in-line with the mental model of the operator.
Minimize display density but not at the cost of coherence	There is adequate free space (white space) maintained in the fan rpm indicator design as well as in the overall interface which is showing all the parameters.
People should not rely exclusively on alarms; provide projection support	Movement of the fan rpm pointer will make continuously aware of the trend or projection of the fan rpm.
Provide SA support rather than a decision	During the simulation and evaluation of all parameters, no direct decision-making advice is provided to the user. The users were required to respond with their own free will based on their SA.
Transparency in automation	Addition of digital indicator with units of fan rpm (rpm) along with the analog movement of fan rpm indicator would ensure the proper perception of the information by the user.
Weber's Law	Red and green colors used in the indicator are different enough for the user to perceive the change. Also, a noticeable stimulus (such as a change in the color of the fan rpm pointer before entering into the critical red zone) may also be provided during the implementation of the design.
Yerkes-Dodson law	Movement of the needle or changing readings of fan rpm indicator (numerical values) can be considered as arousal for the operator along with any other noticeable stimulus like a change in the color or shape of the fan rpm pointer.
Gestalt principle of continuity	Though needle pointer while moving is not touching the marks on the red-green scale, still it is perceived as representing the corresponding value on the scale.
Gestalt principle of symmetry	Overall fan rpm indicator is made of several shapes, lines and colors, though perceived as symmetrical and meaningful graphic to represent the fan rpm of the air seeder unit.
Gestalt principle of common fate	Movement of the fan rpm pointer, as well as a change of the readings in digital form, act in synchronization, so they are perceived as related.

Gestalt principle of figure/ground	The color scheme of the scale, needle and digital scale has been made reasonably prominent than the background to provide a justified visible distinction between foreground and background.
Use of most distinct colors	Uses of most distinct and pure colors are avoided to reduce the shimmering and annoying appearance.
Using brightness and saturation to support distinction among colors	Colors used in the design are distinct enough regarding hue, saturation, and brightness. This can be recognized by viewing the image on the grey scale.
Colors for color-blind people	To work this interface as 'Universal Design' high consideration has been made to use a color-blind friendly color scheme. This can be judged by viewing the design on the grey scale.
For differentiation do not rely on colors alone	During the implementation of this design in experiment sessions, both changes in color and movement of the needle pointer would ensure that change would be effectively communicated to the user.
Paleness, size, and separation	Ample consideration has been made to use appropriate size, paleness, and separation for the fan rpm indicator elements.
How to attract user's attention	To attract user's attention during some critical situations (e.g., if fan rpm pointer is lasting more than 1 minute in error mode), wiggling object or beep sound can be used.
Consistency	Design of fan rpm indicator is conventional, logical and familiar.
Feedback	Constant feedback in the form of movement of the fan rpm pointer in the green and red zone is provided to the operator. Also, some salient cues like a change in color of the fan rpm pointer will be provided to the operator as feedback from the system.
Affordances and convention	As indicated above the design of the fan rpm indicator is conventional, consistent and logical.
Simplicity and structure	This fan rpm indicator is simple in appearance and functionality. Organization of the fan rpm indicator on the interface along with other parameters made based on the size, shape, and availability of the space on the interface.

4.8 TOOL PRESSURE

Tool pressure represents the draft force experienced by the tillage tools (Figure 4-11). The name 'tool pressure' is given by the Karimi et al. 2011, based on the common terminology used by the users. Although, the name 'tool pressure' includes the word 'pressure'; it represents the draft force experienced by the tillage tools, which is justified by the use of Newton (symbol: N) as units of measurement.

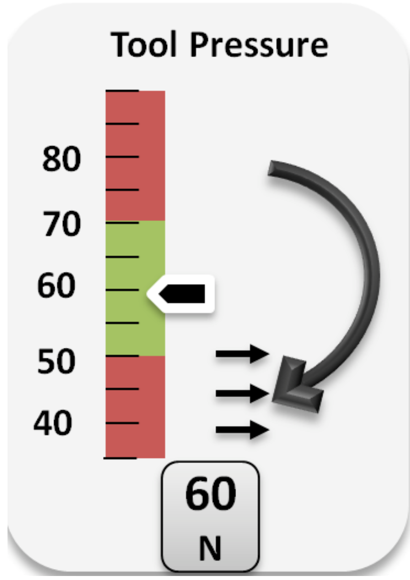


Figure 4-11. Tool pressure experienced by the tillage tools of air seeder.

During the study, for a fair comparison with the baseline conditions (based on Karimi et al. 2011), we neither changed the name or the units. However, we believe that depending upon the measurement methodology of draft force or tool pressure (which is not under the scope of the current study), appropriate unit, i.e. is Newton or Pascal can be used.

The principles used in the design of tool pressure of the tractor air seeder unit are listed in Table 4-8.

Table 4-8. Description of the application of SA and cognitive psychology principles for the design of tool pressure.

Principles	Application of the principles
Focus on the goals	Information about the tool pressure is one of the critical SA information requirements for the operator and support by previous research findings. So, this parameter is considered necessary from SA information requirement's point of view.
Make it easy for the user	Presentation of the tool pressure is simple and easy to understand to help in level 2 type of SA.

Assist in level 3 SA	Raising or lowering level of tool pressure can help the operator to anticipate the trend, i.e., increment or decrement in the tool pressure.
Support trade-offs between goal-driven and data-driven processing	Movement of the tool pressure indicator inside the green zone of the scale supports the goal-driven approach which shows that tool the pressure of the unit is within the recommended range. As soon as the tool pressure enters into the red zone of the scale, the user can comprehend that he/she is out of the prescribed range, and should act as needed.
Make critical cues salient	The red and green coloring of the scale supports this principle.
Avoid excessive features	The need of tool pressure is supported by the previous research finding. So, this is an essential feature. Furthermore, the design of the tool pressure is simple and does not have any unnecessary features.
Map system functions to the goals and mental models of user	The requirement of the tool pressure is supported by previous research finding. Logical and straightforward look and feel of the parameters can easily tune with the mental model of the operator.
Minimize display density but not at the cost of coherence	There is adequate free space (white space) maintained for the design of tool pressure as well as in the overall interface which is showing all the parameters.
People should not rely exclusively on alarms; provide projection support	Raising or lowering level of tool pressure can help the operator to anticipate the trend, i.e., increment or decrement in the tool pressure.
Provide SA support rather than a decision	During the simulation and evaluation of all parameters, no direct decision-making advice is provided to the user. The users were required to respond with their own free will based on their SA.
Transparency in automation	Presence of the numerical readings along with the analog indication in the form of raising or lowering of the tool pressure indicator would ensure the proper perception of the information by the user.
Weber’s Law	Red and green colors used in the scale are different enough for the user to perceive the change. Also, a noticeable stimulus (such as a change in the color of the pressure indicator before entering into the critical red zone or wiggling object) may also be provided during the implementation of the design.
Yerkes-Dodson law	Movement of the level or changing readings of the numerical display can be considered as arousal for the operator along with any other noticeable stimulus like a change in the color of the levels.

Gestalt principle of symmetry	Overall tool pressure indicator is made of several shapes, lines, and colors, though perceived as symmetrical and meaningful graphic to represent the speed of the air seeder unit.
Gestalt principle of common fate	Movement of the pressure indicator over the scale, as well as a change of the readings, act in synchronization, so they are perceived as related.
Gestalt principle of figure/ground	The color scheme of the scale, level, and digital scale has been made reasonably prominent than the background to provide a justified visible distinction between foreground and background.
Use of most distinct colors	Uses of most distinct and pure colors are avoided to reduce the shimmering and annoying appearance.
Using brightness and saturation to support distinction among colors	Colors used in the design are distinct enough regarding hue, saturation, and brightness. This can be recognized by viewing the image on the grey scale.
Colors for color-blind people	To work this interface as 'Universal Design' high consideration has been made to use a color-blind friendly color scheme. This can be judged by viewing the design on the grey scale.
For differentiation do not rely on colors alone	During the implementation of this design in experiment sessions, both changes in level adjacent to the scale and numerical readings beneath are provided to perceive any change in the situation.
Paleness, size, and separation	Ample consideration has been made to use appropriate size, paleness, and separation of the elements in the design.
How to attract user's attention	Movement of the tool pressure indicator, along with the change of color would help in catching the adequate attention of the user.
Consistency	Design of application rate is simple, logical and familiar.
Feedback	Constant feedback in the form of movement of the level and changing of numerical readings beneath provides constant feedback to the operator.
Affordances and convention	As indicated above the design of the tool pressure is logical and straightforward.
Simplicity and structure	Tool pressure is simple in appearance and functionality.

Based on the conceptual interface discussed in this section, individual interface elements, as well as the complete versions of the interfaces, were developed, tested and evaluated against the baseline condition in two phases. Details about the first and second phase of the study are discussed in Chapters 5 and 6.

CHAPTER 5

DESIGN AND EVALUATION OF INDIVIDUAL ELEMENTS OF THE INTERFACE FOR AN AGRICULTURAL MACHINE[1]

5.1 ABSTRACT

If a user-centered approach is not used to design information displays, the quantity, and quality of information presented to the user may not match the needs of the user, or it may exceed the capability of the human operator for processing and using that information. The result may be an excessive mental workload and reduced situation awareness of the operator, which can negatively affect the machine performance and operational outcomes. The increasing use of technology in agricultural machines may expose the human operator to excessive and undesirable information if the operator's information needs and information processing capabilities are ignored. In this study, a user-centered approach was used to design specific interface elements for an agricultural air seeder. Designs of the interface elements were evaluated in a laboratory environment by developing high-fidelity prototypes. Evaluations of the UI elements yielded significant improvement in situation awareness (up to 11%; overall mean difference = 5.0 (4.8%), 95% CI (6.4728, 3.5939), $P < 0.0001$). Mental workload was reduced by up to 19.7% (overall mean difference = -5.2 (-7.9%), $n = 30$, $\alpha = 0.05$). Study participants rated the overall performance of the newly designed user-centered interface elements higher in comparison to the previous designs (overall mean difference = 27.3 (189.8%), 99% CI (35.150, 19.384), $P < 0.0001$).

Keywords. *Agricultural machines, Interface design principles, Situation awareness, User-centered design, User interface design, User experience.*

5.2 INTRODUCTION

Modern mobile agricultural machines (MAMs) are becoming increasingly complex. The human operator is required to work with several mechatronic systems to operate the MAM. This work involves monitoring the input from systems and sensors, and controlling the output based on input information. If the information needs of the operator are ignored, or if the information presented to the operator (in terms of either quantity or quality) does not align with the human capabilities for processing and using the information, it may cause loss in situation awareness (SA) and excessive mental workload (MWL) (Endsley et al. 2003). Degraded SA can negatively affect both the productivity of the MAM and the safety of the operator.

When the quantity of information presented to operators exceeds the human ability to comprehend that information, it can be said that the operators lack adequate SA of the task in which they are involved. As explained by Endsley (1988), SA can be understood at three levels: perception, comprehension, and projection. Level 1 SA is the perception of information from the environment through the use of various senses (i.e., auditory, visual, touch, taste, and smell). At level 2, SA involves an understanding of the information that has been perceived at level 1. Finally, level 3 SA requires anticipation or projection of the near-future situation based on the prior perception and comprehension of the information. Large amounts of data should be presented using various visual elements (i.e., graphs, trends, shapes, colors, or other visual forms) so that SA can be enhanced and stress on working memory can be reduced (Atkinson and Shiffrin 1968; Baddeley 1997; DeSanctis 1984; Tullis 1981; Endsley et al. 2003). Complexity is another enemy of SA (Cummings 2005; Horrey 2011; Liu 2001). If a device, panel, or display contains too many features or too much complexity, it will be difficult for the user to perceive information (level 1 SA), to comprehend the information (level 2 SA), and to project into the future situation (level 3 SA) (Endsley et al., 2003).

Endsley et al. (2003) and others (Fennell et al. 2006; Lenorowitz 1988; Young et al. 2007) discussed several operational problems and accidents that occurred in aviation due to poorly designed interfaces. In most cases, the interface failed to provide sufficient SA to the pilot (Endsley et al., 2003). Key information may not have been detected by the pilot, even though it was present on the interface. In another example, the pilot may have known the exact altitude of the aircraft but failed to understand that the aircraft's altitude did not provide sufficient clearance for the terrain. In a different publication, Jones and Endsley (1996) reported that 76.3% of errors were related to the perception of information (i.e., level 1 SA), 20.3% of errors were related to the comprehension of information (i.e., level 2 SA), and 3.4% of errors were related to "failure to project the situation into the future" (i.e., level 3 SA). These results seem to imply that the overall effectiveness of an interface is influenced by consideration of the principles that support SA.

Apart from the aviation domain, the SA paradigm has also been applied to the design of interfaces for transportation and nuclear power plants. Golestan et al. (2016) highlighted the role of SA in the "internet of cars" concept. Based on the terminology of SA, Golestan et al. (2016) proposed different models and methods responsible for perceiving objects and relationships, comprehending "situation assessment" and "threat assessment," and using "impact assessment" and "decision making" as the projection component of SA. In a study related to nuclear power plants, traditional and improved interfaces were compared using the SA paradigm (Burns et al. 2007). Improved displays (referred to as “ecological interfaces”) were designed by incorporating visuals and graphics into the existing displays. The ecological interfaces showed performance advantages in comparison to the existing interfaces in specific scenarios. Evidence from the scientific literature supports the premise that interface designers should consider SA during the design process and as a means of evaluating the adequacy of a given interface design. This user-centered approach has never yet been applied to the design an air seeder interface.

Research in the Agricultural Ergonomics Laboratory at the University of Manitoba is dedicated to developing the knowledge necessary to design agricultural machines that minimize the impacts, both physical and mental, on the operator. The agricultural machine selected for this study was an air seeder because planting is a critical machine operation that requires careful attention by the operator. Furthermore, previous research provided knowledge of the task of operating an air seeder and enabled the design of a high-fidelity simulator that mimics the task of operating a tractor and air seeder system (Bashiri and Mann 2014, 2015). The specific objectives of this study were; (1) to design individual display elements, using various design principles intended to support the information needs of the operator, that would ultimately comprise the interface for monitoring the operation of an air seeder, and (2) to evaluate the newly developed display elements against display elements previously used in the high-fidelity simulator (Karimi et al. 2011) in terms of the SA enabled and MWL imposed.

5.3 MATERIALS AND METHODS

5.3.1 Design of the Interface Elements

To design the interface elements for an air seeder to support the operator's SA, it was necessary first to understand the goals of the operator, the decisions associated with these goals, and the information needs of the operator (Endsley et al., 2003). To gain this information for an entire system consisting of a tractor and an air seeder; operator's manuals, product specifications, and other technical literature were reviewed during an internship completed by the first author at the premises of a local manufacturer of agricultural machines. In addition to the review of documentation, informal discussions with machine operators and farmers were used to further understand the goals, tasks, and information requirements of the operator. Using the goal-directed task analysis (GDTA) approach (Endsley et al. 2003), the various goals, sub-goals, decisions, and information requirements related to the operation of an agricultural air seeder were summarized (Figure 5-1).

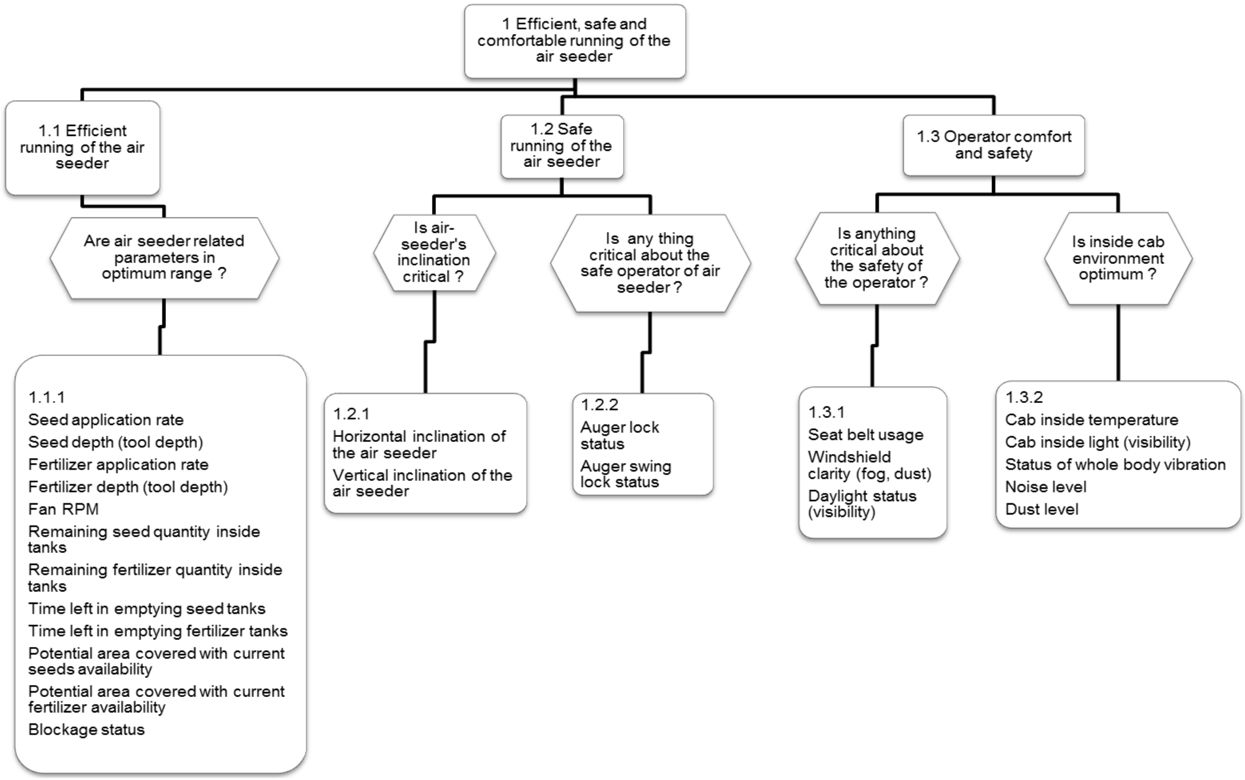


Figure 5-1. Detailed goal-directed task analysis for a tractor and air seeder system.

With reference to Figure 5-1, the major goal of “efficient, safe, and comfortable running of the air seeder” is divided into three sub-goals (i.e., efficient running of the air seeder, safe running of the air seeder, and operator comfort and safety). The decisions to be made by the operator, and the corresponding SA requirements for the operator to effectively meet each of these sub-goals, are listed under each sub-goal in sections 1.1.1, 1.2.1, 1.2.2, 1.3.1, and 1.3.2 in Figure 5-1. A total of 23 SA requirements were needed to meet all goals. However, not all the SA requirements were supported by the machines available at the local manufacturer (e.g., no information was provided to the operator regarding the inclination of the unit). A total of 12 parameters related to the air seeder were identified to address the SA requirements of the operator. These parameters were: seed level status (tank levels), fertilizer level status (tank levels), fan speed (RPM), seed application rate (SAR), seed depth (tool depth, TD), fertilizer application rate (FAR), fertilizer depth (tool depth, TD), tool pressure (TP), blockage status, desired path of the unit, desired location of the unit, and current speed of the unit. These parameters were also recommended by Karimi et al. (2011) as the most frequently viewed by air seeder operators. Figure 5-2 shows pictorials that have been

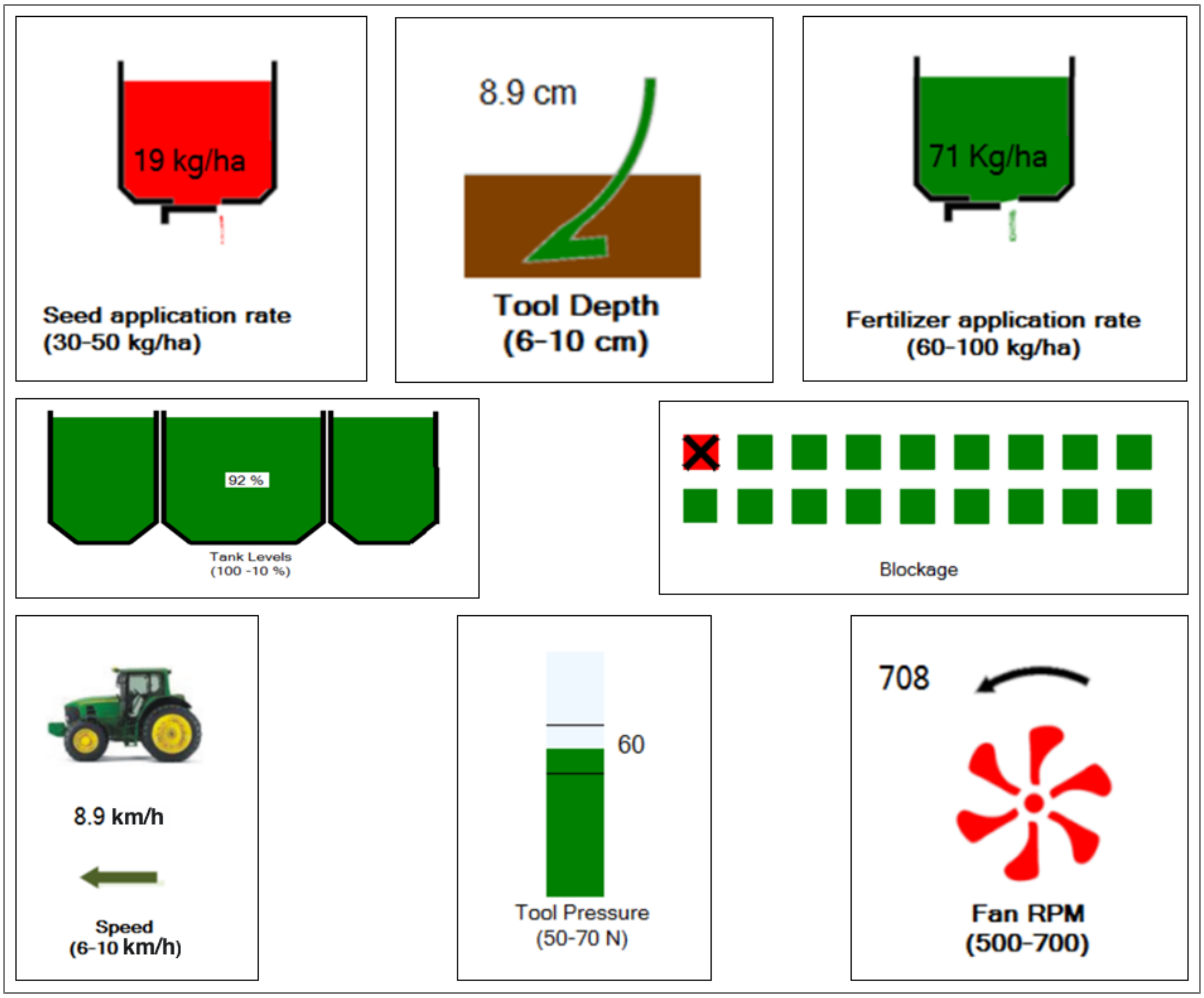


Figure 5-2. Pictorials used to display air seeder elements during simulator studies in the Agricultural Ergonomics Laboratory at the University of Manitoba (based on Karimi et al., 2011).

used to display these parameters in previous simulator studies conducted in the Agricultural Ergonomics Laboratory at the University of Manitoba (pictorials are based on a study completed by Karimi et al. 2011).

As mentioned previously, this study intended to improve upon the individual air seeder display elements that are currently used in the simulator in the Agricultural Ergonomics Laboratory at the University of Manitoba, with the ultimate goal of gaining knowledge to support interface design for agricultural machines. The eight original display elements shown in Figure 5-2 were re-designed following an in-depth review of the principles guiding display design. This in-depth review can be read in Rakhra and Mann (2014). The purpose was to design the display elements to support all three levels of SA (i.e., perception, comprehension, and projection). The newly designed display elements, which are referred to as the user-centered display (UCD) elements, are presented in Figure 5-3.

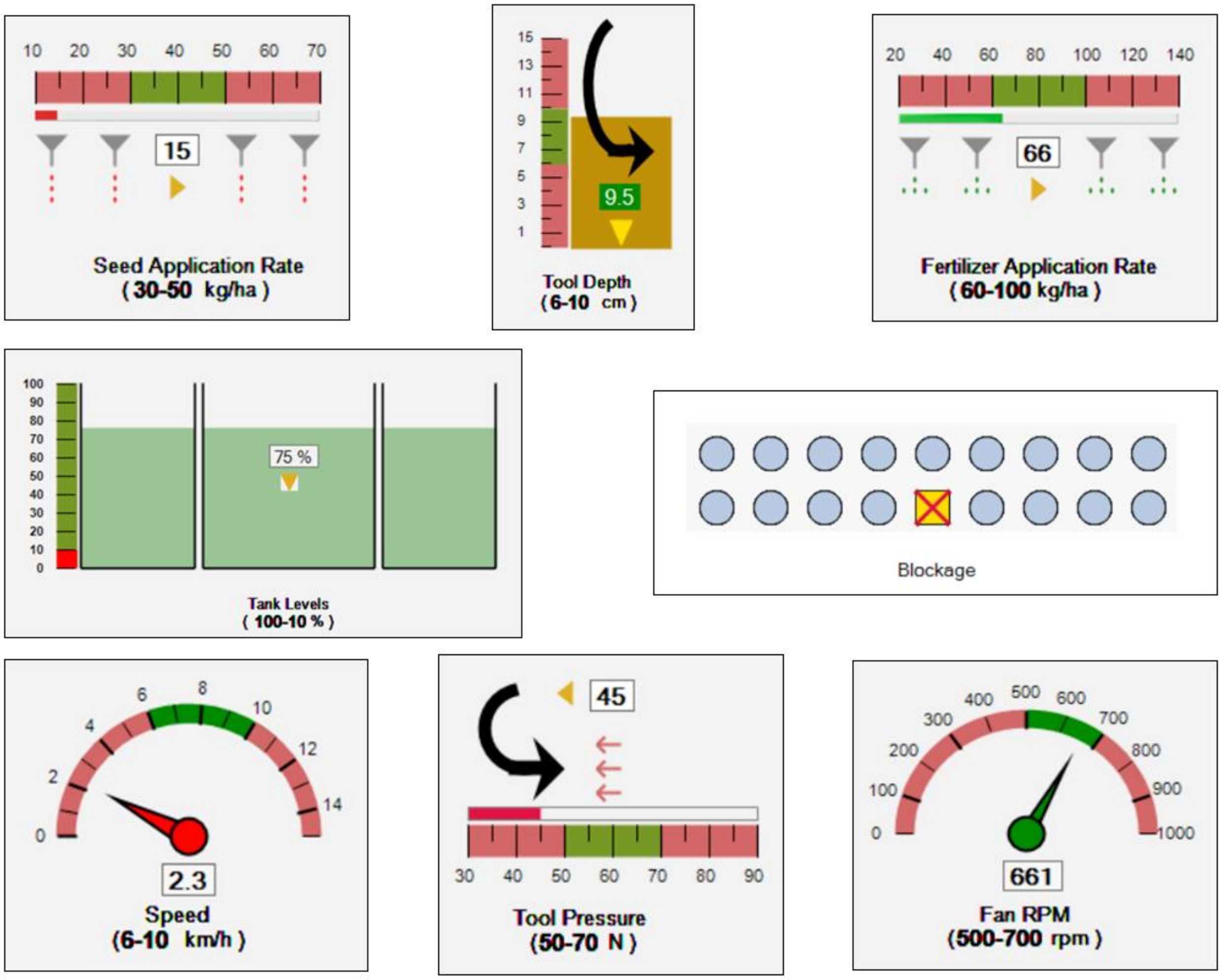


Figure 5-3. Newly designed air seeder display elements based on user-centered design (UCD) principles to support the situation awareness of the user.

Table 5-1 describes how various SA and human factors principles informed the designs.

Table 5-1. Situation awareness and human factors principles that informed the design of the new air seeder display elements.

Principle	Example Applications of the Principle for Interface Design
Focus on the goals (Endsley et al. 2003)	All eight parameters (i.e., seed application rate, fertilizer application rate, tool depth, tool pressure, speed, fan RPM, tank levels, and blockage) are critical for supporting the SA of the user, as indicated by the GDTA. This information is directly needed to make critical decisions associated with the operation of air seeder.

Principle	Example Applications of the Principle for Interface Design
Make user-assist easy in level 2 SA (Endsley et al. 2003)	Explicitly showing the desirable/undesirable ranges (red/green on the scale), increasing or decreasing trends (moving levels, arrows, needles, and dots), and blockage status make it easy for the user to comprehend the situation.
Assist in level 3 SA (Endsley et al. 2003)	Use of a colored scale and movement of a level or needle helps the user to project the future state based on the current situation. For example, if the seed application rate (SAR) is constant at 65 kg ha^{-1}, the user can anticipate that it should stay constant for the very near future in comparison to a situation in which the SAR is rapidly increasing.
Make critical cues salient (Endsley et al. 2003)	When a parameter exceeds a critical limit, such as less than 6 km h^{-1} or more than 10 km h^{-1} for speed, the color of the indicator, such as the speed needle, changes (from green to red, or vice versa) to catch the user's attention. Similar changes occur for the other parameters.
Avoid excessive features (Endsley et al. 2003; Miller 1956)	The design of all parameters is made simple without compromising the specific details. Based on the GDTA, only essential parameters were considered in the design.
Map system functions to user's goals and mental models (Endsley et al. 2003)	All eight functional parameters are related to the major goals of the user. The design of these parameters is simple, conventional, and associated with the various levels of SA so that it can be easily related to the mental models of the user.
Minimize display density but not at the cost of coherence (Endsley et al. 2003)	Adequate free space (white space) is maintained in the design of the parameters without compromising the details and coherence.
Users should not rely on exclusively on alarms; provide projection support (Endsley et al. 2003)	Scales and trends provide ample projection support to the user. Before entering into the "not acceptable" zones (e.g., red areas), trends and deviations help the user anticipate the criticalness of the situation and immediately correct the course.
Provide SA support rather than decision support (Endsley et al. 2003)	During the simulation and evaluation of all parameters, no direct decision-making advice is provided to the user. The users were required to respond with their own free will based on their SA.
Transparency in automation (Endsley et al. 2003)	Digital readings (numerical values), along with trends, ranges, and projection support, assist the user in avoiding ambiguity in the perception, comprehension, or foreseeing of the situation.

Principle	Example Applications of the Principle for Interface Design
Weber's law (Feldman 2008)	To indicate any critical changes to the user during the simulation, sufficient noticeable stimuli are provided for easy recognition of the changes, such as movement of a needle, movement of a progress bar, and change in the color and shape of a blockage indicator.
Yerkes-Dodson law (Cohen 2011; Hanoch et al. 2004)	Too little or too much arousal (in the form of information, changes, etc.) detracts from effective performance. The effort was made to provide an appropriate level of arousal to keep the user informed and alert. For example, the movement of a needle or the indication of a high, low, or correct quantity of seed or fertilizer can be considered an appropriate level of arousal.
Gestalt principle of continuity (Todorovic 2008)	This principle states that "visual perception is biased to perceive continuous form rather than disconnected segments" (Johnson 2010). Application of this principle can be found in the design of several parameters. For example, the speed needle does not touch the marks on the scale, but it is perceived as representing the corresponding value on the scale.
Gestalt principle of similarity (Todorovic 2008)	Objects that appear similar to each other, within a collection of objects, are perceived as related or grouped. For example, in the blockage indicators, all the round gray-colored circles, representing no blockage in the system, are perceived as related to each other.
Gestalt principle of figure and ground (Todorovic 2008)	The color schemes of the scales, animations, trend arrows, and other shapes have been made more prominent than the background to provide a visible distinction between foreground and background.
Use of colors (Christ, 1975; Johnson 2010)	Pure or highly saturated colors, with no traces of other colors, may cause strong signals in visual perception and should be used with caution. In all the design elements, the use of distinct and pure colors was avoided to reduce shimmering or an otherwise annoying appearance.
Color brightness and saturation (Johnson 2010)	The colors used in the design elements are distinct in hue, saturation, and brightness. This helps the user perceive the information as well as any changes during operation.
For differentiation, do not rely on colors alone (Johnson 2010)	In addition to changes in color, changes in shape or size are also provided to help the user perceive changes in operation. For example, each blockage indicator can change to a yellow square with a red cross to indicate a blockage in the system.
Paleness, size, and separation (Johnson 2010; Christ 1975)	Ample consideration was given to the appropriate size, paleness, and separation of the design elements to help the user perceive, comprehend, and foresee any situation.
Attracting user's attention (Johnson 2010)	Changes in color (red to green or vice versa), along with changes in the shape or size of the elements, were used to attract the user's attention without excessive cues, to avoid the phenomenon of habituation.

Principle	Example Applications of the Principle for Interface Design
Consistency (Norman 1983; Nielsen and Molich 1990; Shneiderman 1998; Shneiderman et al. 2009; Stone et al. 2005)	For all parameters, similar logic and appearances were adopted. For example, most parameters have a scale that shows the acceptable values and ranges. Alongside the scale, a progress indicator (bars or levels) helps users perceive, comprehend, and foresee their progress.
Feedback (Endsley and Kiris 1995; Norman 1990; Stone et al. 2005)	Constant feedback in the form of progress bars, readings, rising/falling levels, and other animation (e.g., movement of needles and high/low quantity indications) was provided to keep the user informed.
Affordances and convention (Gibson 1977; Gibson 2014; Norman 1999)	Most of the metaphors, representations, or conventions used in the design of the parameters are familiar, easy to understand, and in line with the mental models of the users.
Simplicity and structure (Gibson 2014; Stone et al. 2005)	The design of the individual elements and the interface as a whole is simple, vivid, and natural in appearance and function. For example, the speed needle is simple, familiar, and predictable.

5.3.2 Evaluation of the Interface Elements

To evaluate the eight newly designed UCD elements of the air seeder display regarding SA and MWL, simulations were developed in Visual Basic using Microsoft Visual Studio Express 2013. With eight original display (OD) elements and eight UCD elements, 16 different simulations were required. A brief introduction to the research and a 5 min training session were provided to the participants before the start of the experiment. An experimental session consisted of 48 simulations presented randomly. Each simulation was designed to run for 45 s, during which time the participant was expected to observe the changing information on the computer monitor. At the completion of each 45 s simulation, questions were presented to the participant. After completion of the 24th simulation, a 5 min break was provided to the participant to minimize fatigue. For additional details about the execution, please see the Appendix E and F. The experiments were conducted in the Agricultural Ergonomics Laboratory in the Department of Biosystems Engineering at the University of Manitoba. The experimental protocol for this study received human ethics approval from the Education/Nursing Research Ethics Board of the University of Manitoba (Appendix A.1).

The experimental design selected for this study was a randomized complete block design (RCBD). To reduce the variability due to individual differences among the participants, all participants were assigned both treatments (i.e., a repeated-measures, within-subjects design was used). To minimize the learning effect, both treatments (OD and UCD; 48 simulations) were randomly displayed to the participants. Thirty people were recruited to participate in this study, with the majority being University of Manitoba students. Twenty-five male and five female participants volunteered for the study. Ages ranged from 14 to 60, with a mean of 25.6 years. Only 13 of the 30 participants had prior farming experience. Participants received a small honorarium ($20) for their participation in the study.

As required for the Situation Awareness Global Assessment Technique (SAGAT) (Endsley et al. 2003), each participant was asked a series of questions after each simulation. Questions 1 and 2 (Figure 5-4) were related to level 1 SA, questions 3 and 4 were related to level 2 SA, and question 5 was related to level 3 SA. The participants answered the questions using the mouse and keyboard and then submitted their responses with a mouse click. To evaluate the MWL of the participants, a unidimensional, easy-to-administer, and less-intrusive subjective integrated workload scale (IWS) (Golightly et al. 2012; Pickup et al. 2005) was incorporated with the SAGAT questions (question 6 in Figure 5-4).

To measure the differences in SA and MWL of the participants for the two types of display elements, design type was considered an independent variable with two variants: original design (OD) and user-centered design (UCD). SA and MWL were considered dependent variables. A total of 7200 responses (i.e., 48 simulations $\times$ 5 questions (Figure 5-4, questions 1 to 5) $\times$ 30 participants) related to SA and 1440 responses (48 simulations $\times$ 1 question (Figure 5-4, question 6) $\times$ 30 participants) related to MWL were automatically recorded by the simulation program for later evaluation. Also, subjective feedback from the participants was collected using pen and paper format. Each subjective feedback form consisted of seven questions (Table 5-2) comparing the two types of display elements. All data were analyzed using JMP statistical software.

For analysis, the raw data collected during the experiment were processed and organized in a systematic way using Visual Basics for Applications (VBA) in Excel. To analyze and compare the OD and UCD interface designs for level 1, level 2, and level 3 SA (Table 5-3), as well as for MWL (Table 5-4) and subjective feedback (Table 5-5), the data were re-organized in Excel spreadsheets according to the comparisons needed. A total number of correct responses received for the OD and the UCD interface were analyzed. For level 1 SA (Table 5-3), correct responses for questions 1 and 2 (Figure 5-4) were compared. The total score was out of 48 (3 replicates of 8 elements for Q1 plus 3 replicates of 8 elements for Q2) for OD or UCD interface for level 1 SA. For statistical analysis, a paired t-test (n = 30, α = 0.05) was performed using JMP statistical software. The same approach was used for level 2 SA based on responses to Q3 and Q4. Level 3 SA was evaluated using only a single question (Q5), therefore, a perfect score was 24 (3 replicates of 8 elements). For overall SA, a perfect score was 120 (i.e., 48 for level 1 SA plus 48 for level 2 SA plus 24 for level 3 SA).

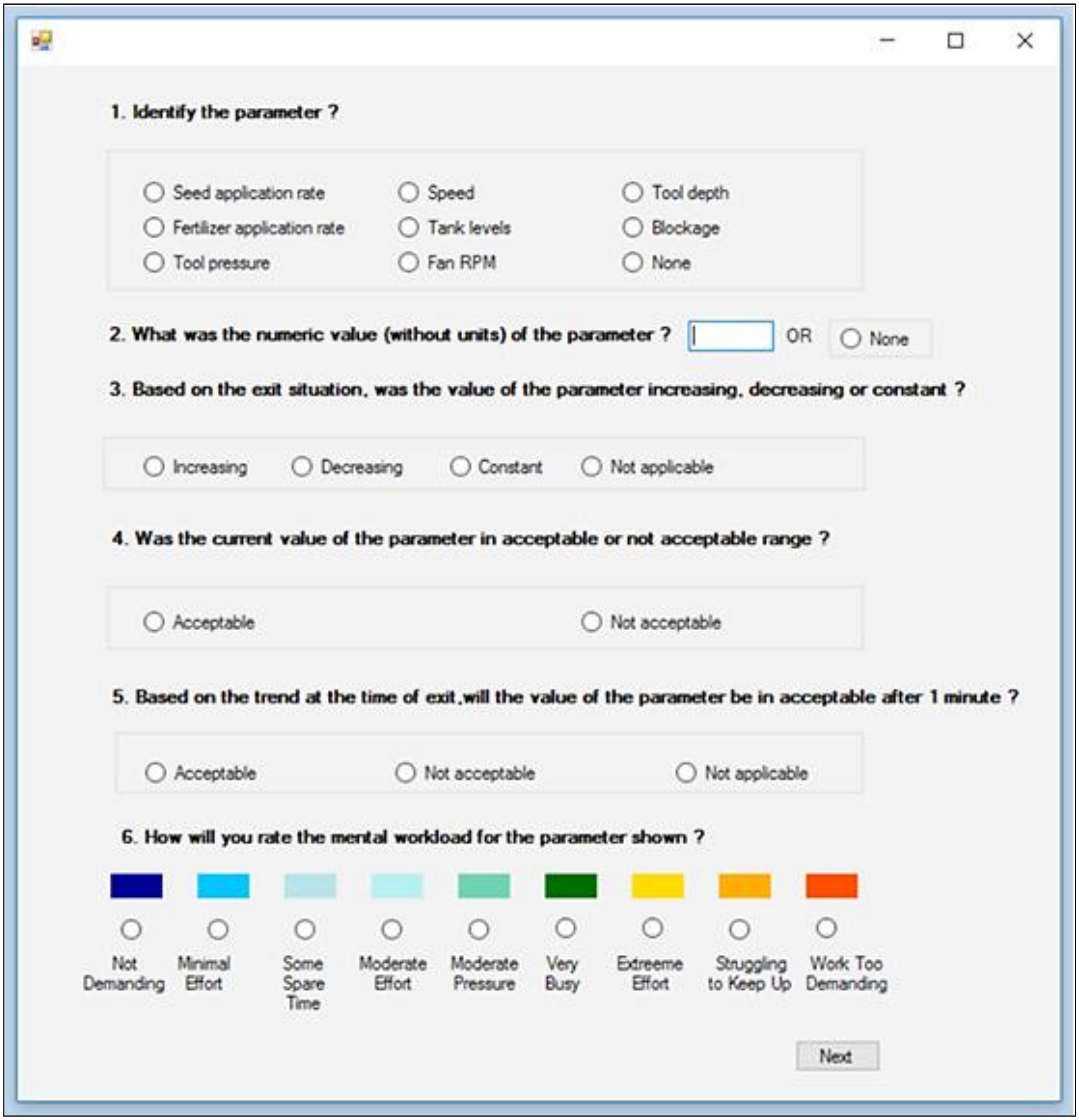


Figure 5-4. Questions asked of the participants after every simulation during the experiment.

Table 5-2 describes the questions asked to the participants at the end of the experiment for the collection of subjective feedback regarding two types of designs.

Table 5-2. Questions asked of the participants at the end of the experiment for subjective feedback.

No.	Question
1	Which display improves your ability to identify the parameter?
2	Which display improves your ability to recall (remember) the parameter value?
3	Which display improves your ability to understand or comprehend the situation?
4	Which display improves your ability to identify the trend?
5	Which display improves your ability to foresee and predict the situation?
6	If your goal is to control and maintain this parameter within the recommended range, which display do you prefer?
7	Which display do you like better?

5.4 RESULTS AND DISCUSSION

5.4.1 Situation Awareness

Two types of display elements (OD and UCD) were compared for eight air seeder elements. The null hypothesis (H_0) was that there would be no difference in the SA of the user or that the OD was better than the UCD (i.e., UCD mean – OD mean ≤ 0). The alternative hypothesis was that the newly designed UCD would be better than the OD regarding the SA of the user (i.e., UCD mean – OD mean > 0.)

Considering the consolidated responses to the SA questions (i.e., the sum of Q1 to Q5, OD vs. UCD) for all the display elements, highly significant ($p < 0.0001$) results were obtained; UCD significantly improved the SA of the users in comparison to OD (Table 5-3, ^{**4}). Regarding level 2 SA (^{**2}) and level 3 SA (^{**3}), significant improvements were found when using UCD in comparison to OD. For level 2 SA, the UCD mean was 3.2 (7.6%) higher than the OD mean. For level 3 SA, the UCD mean was 1.8 (8.3%)

higher than the OD mean. The superior performance of the UCD display elements can be explained by the presence of design attributes that better support the users' perceptual and cognitive characteristics to improve the users' SA (Endsley et al. 2003).

For level 1 SA, no statistically significant results were observed (**1). The OD elements did not demonstrate much inferior performance for level 1 SA (perception) in comparison to UCD. The comparable performance of OD can be attributed to the fact that the OD elements had adequate levels of visual and cognitive attributes (e.g., vivid colors, shapes, and sizes) that enabled a comparable level of SA as the UCD elements. However, considering the overall SA, as well as the level 2 and level 3 SA separately, the UCD elements showed better results.

Referring to the individual display elements (**5), statistically significant improvements in SA were observed when using UCD in comparison to OD for six of the eight display elements: SAR, FAR, speed, TP, fan RPM, and TD. However, for tank levels and blockage (**6), no statistically significant results were obtained. The comparable performance for tank levels and blockage among the OD and UCD display elements may have been due to the high degree of similarity between the two designs.

Table 5-3. Summary of statistical analysis (n = 30, α = 0.05) of single-tail t-test (matched pairs) comparing situation awareness (SA) responses for users of OD and UCD interfaces for air seeder display elements.

	OD Mean	UCD Mean	Mean Diff.	Standard Error	Upper 95%	Lower 95%	DF	t Ratio	Prob > t
All parameters									
Level 1 SA (Q1-2)**1	41.4	41.5	0.1	0.4597	1.0069	-0.8736	29	0.1450	0.4429
Level 2 SA (Q3-4)**2	42.1	45.3	3.2	0.4108	4.0401	2.3599	29	7.7902	<0.0001
Level 3 SA (Q5)**3	21.2	22.9	1.8	0.3413	2.4648	1.0686	29	5.1757	<0.0001

Overall SA (Q1-5)**4	104.7	109.7	5.0	0.7038	6.4728	3.5939	29	7.1514	<0.0001
Individual parameters**5 (Q1-5)									
SAR	13.1	13.7	0.6	0.3276	1.2367	-0.1033	29	1.7298	0.0472
FAR	13.5	14.1	0.7	0.2726	1.2243	0.1091	29	2.4453	0.0104
Speed	12.8	14.3	1.4	0.2568	1.9585	0.9082	29	5.5818	<0.0001
Tank levels**6	14.2	14.3	0.1	0.1300	0.3659	-0.1659	29	0.7693	0.2240
TP	13.1	13.9	0.8	0.2218	1.2536	0.3464	29	3.6068	0.0006
Fan RPM	13.7	14.2	0.4	0.2016	0.8457	0.0210	29	2.1492	0.0200
TD	12.9	13.7	0.8	0.2510	1.3133	0.2867	29	3.1876	0.0017
Blockage**6	11.3	11.5	0.2	0.2233	0.6900	-0.2233	29	1.0451	0.1002

Note: Maximum number of correct responses were 48 for level 1 SA (3 replicates x 8 elements x 2 questions), 48 for level 2 SA (3 replicates x 8 elements x 2 questions), 24 for level 3 SA (3 replicates x 8 elements x 1 question), and 15 for each of the individual parameters (3 replicates x 1 element x 5 questions).

The percentage improvement in SA using the UCD display elements compared to the OD display elements is shown in Figure 5-5. These percentage values are based on the differences in the UCD and OD means shown in table 3. Referring to Figure 5-5, a 3% to 11% improvement in SA was observed for the UCD display elements. Significant improvements in level 3 SA (8.3%) and level 2 SA (7.6%) were observed.

A large improvement (11.2%) was observed in the participants' SA for the speed element Figure 5-2 and Figure 5-3 show notable differences between the two designs for this element. The UCD design supports the perception level of SA with the movement of the needle along with numeric values. The vivid and changing color of the needle, along with its movement, helps the user perceive the information. Continuous availability of the semi-circular scale along with the movement of the needle helps the user with continuous evaluation and comprehension of the present situation (level 2 SA). This design further helps the user in anticipating or projecting the near-future situation (level 3 SA) based on the user's perception and comprehension of the current situation. This SA assistance is missing in the OD speed element.

However, for level 1 SA, and for tank levels and blockage, no statistically significant SA improvements were observed (Figure 5-5). Based on the other six display elements (SAR, FAR, speed, TP, fan RPM, and TD), an average SA improvement of 6.0% was observed.

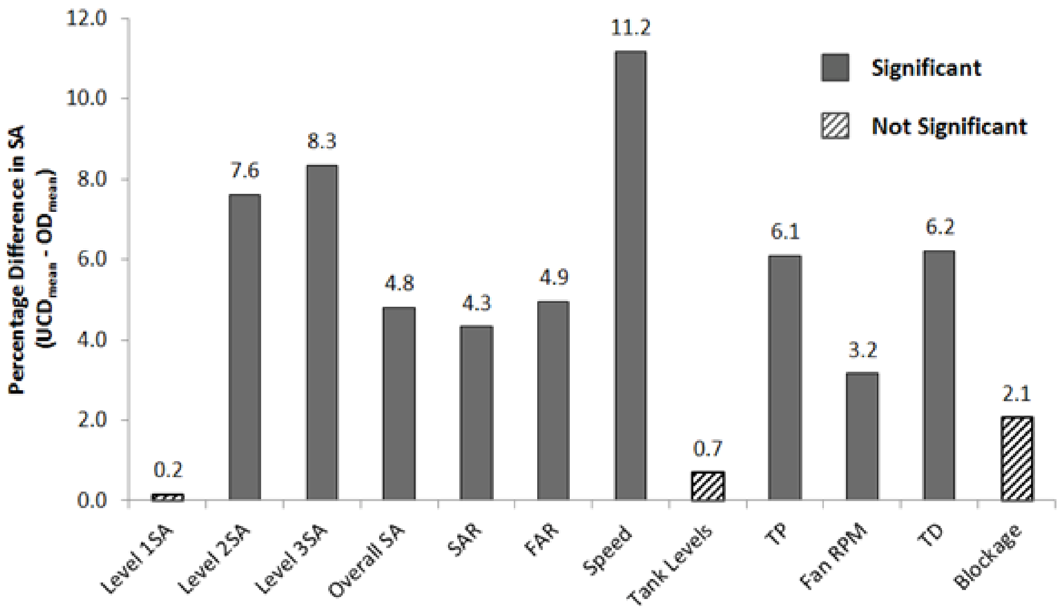


Figure 5-5. Percentage improvement in situation awareness (SA) for user-centered design (UCD) over original design (OD) for air seeder display elements.

5.4.2 Mental Workload

Based on the nine-point IWS, the subjective feedback from the participants about their perceived MWL is summarized in Table 5-4. Statistically significant reductions in the participants' MWL were observed for SAR, FAR, speed, and fan RPM. However, for tank levels, TD, and blockage, the differences were not statistically significant.

Table 5-4. Summary of statistical analysis (n = 30, α = 0.05) of single tail t-test (matched pairs) comparing mental workload (MWL) for users of OD and UCD interfaces for air seeder display elements.

	Mean OD	Mean UCD	Mean Diff.	Standard Error	Upper 95%	Lower 95%	DF	t Ratio	Prob < t
SAR	10.5	9.4	-1.1	0.5611	0.0808	-2.2142	29	-1.9012	0.0336
FAR	10.8	9.6	-1.3	0.6538	0.0705	-2.6038	29	-1.9374	0.0312
Speed	9.8	7.9	-1.9	0.6746	-0.5537	-3.3129	29	-2.8661	0.0038
Tank levels	7.7	8.1	0.4	0.4490	1.3516	-0.4849	29	0.9652	0.8288
TP*1	8.4	10.5	2.1	0.5338	3.1583	0.9750	29	3.8719	0.9997
Fan RPM	10.2	8.3	-1.9	0.7942	-0.2758	-3.5242	29	-2.3925	0.0117
TD	10.6	10.5	-0.1	0.6996	1.3643	-1.4976	29	-0.0953	0.4624
Blockage	6.4	7.0	0.6	0.4763	1.5408	-0.4075	29	1.1897	0.8781
Overall*2	66.1	60.9	-5.2	2.5041	-0.1119	-10.3550	29	-2.0899	0.0228

Note: The 9-point IWS was converted to a numerical score (1 through 9). There were 3 replicates for each element and the scores were summed, therefore, the maximum value for each element is 27.

Considering the overall MWL (*2) (i.e., based on the participants' ratings for seven of the eight parameters, excluding TP), a 7.9% reduction in MWL was observed. Based on the participants' ratings and subjective feedback about TP, substantial changes in the design of the TP element would be required, so the results for the TP ratings are not included in the overall MWL average calculation (*2). The observed percentage changes in MWL for all parameters are shown in Figure 5-6.

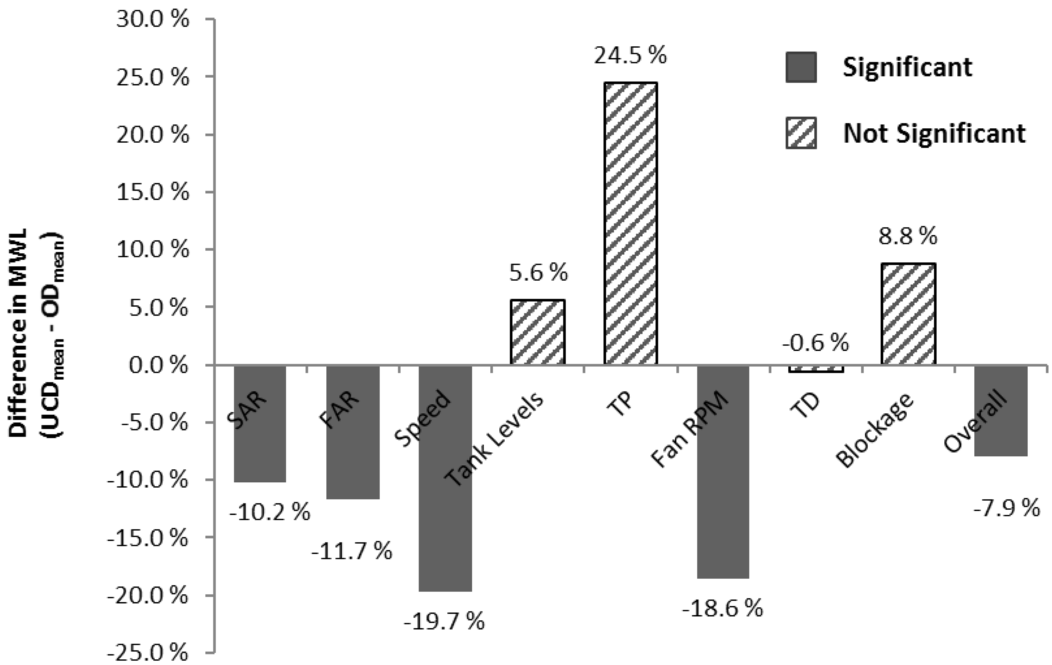


Figure 5-6. Percentage change in mental workload (MWL) for user-centered design (UCD) over original design (OD) for air seeder display elements. Negative values indicate a reduction in MWL.

Most of the display elements (SAR, FAR, speed, fan RPM, and TD) showed significant reductions in MWL; significant improvements in SA were also shown for these elements in the UCD in comparison to the OD. These improvements in both SA and MWL provide a clear indication that the UCD display elements are more effective than the OD display elements. However, for one of the parameters (TP), an increase in MWL of about 24.5% was observed, which suggests that the UCD design of the TP display element requires further modification to address this increase in MWL. Based on analysis of the TP display element, the complexity and the use of a higher number of shapes (i.e., a tool-shaped arrow and pressure represented by opposing arrows) can be considered major contributors to the increase in the MWL. Based on their subjective and verbal feedback, many participants indicated that the tool-shaped arrow along with the opposing arrows caused confusion and stress in interpreting the situation.

Nevertheless, this increase in MWL came with a 6.1% improvement in SA (Figure 5-5). Overall, for most of the display elements, a decrease in MWL was observed.

5.4.3 Subjective Feedback

For subjective opinion, ten questions were asked of the participants at the end of the experimental session. Of the ten questions, seven were related to comparison of the SA between the two designs (OD and UCD) (Table 5-2). The remaining three questions were descriptive to elicit feedback and suggestions to further improve the UCD for future study. Analysis of the seven questions is summarized in Table 5-5.

Table 5-5. Summary of statistical analysis (n = 30, α = 0.01) of single-tail t-test (matched pairs) comparing the situation awareness (SA), preference, and overall effectiveness of OD and UCD interfaces for air seeder display elements based on the subjective feedback.

	OD Mean	UCD Mean	Mean Diff.	Standard Error	Upper 99%	Lower 99%	DF	t Ratio	Prob > t
All parameters									
Level 1 SA (Q1)*1	2.8	5.2	2.3	0.5680	3.8989	0.7678	29	4.1081	0.0001
Level 2 SA (Q3)*2	2.3	5.7	3.4	0.5352	4.8752	1.9248	29	6.3528	<0.0001
Level 3 SA (Q4-5)*3	2.4	13.6	11.2	0.7086	13.1531	9.2469	29	15.8065	<0.0001
Overall SA (Q1-3-4-5)	7.5	24.5	16.9	1.5335	21.1602	12.7065	29	11.0424	<0.0001
Recall (Q2)*4	2.5	5.5	2.9	0.5732	4.5134	1.3533	29	5.1173	<0.0001
Preference (Q6-7)*5	4.3	11.7	7.4	1.0200	10.2116	4.5884	29	7.2547	<0.0001
Overall opinion (Q1-7)	14.4	41.6	27.3	2.8599	35.1497	19.3836	29	9.5341	<0.0001
Individual parameters (Q1-7)									
SAR	1.5	5.5	4.0	0.5734	5.5804	2.4196	29	6.9765	<0.0001
FAR	1.2	5.8	4.7	0.6786	6.5372	2.7961	29	6.8766	<0.0001
Speed	1.0	6.0	5.0	0.5985	6.6507	3.3493	29	8.3493	<0.0001
Tank levels	1.0	6.0	4.9	0.7270	6.9373	2.9293	29	6.7855	<0.0001
TP**6	2.9	4.1	1.1	0.8520	3.4816	-1.2150	29	1.3303	0.0969

Fan RPM	1.6	5.4	3.8	0.7942	5.9892	1.6108	29	4.7845	<0.0001
TD	1.2	5.8	4.7	0.6921	6.5742	2.7591	29	6.7433	<0.0001
Blockage[**6]	4.0	3.0	-0.9	1.2489	2.5077	-4.3744	29	-0.7476	0.7696

Note: Each participant answered each of 7 questions for 8 elements, selecting either the OD or UCD. For all parameters section, OD plus UCD sum to 8. The table shows the mean (based on n=30 participants) number of times OD and UCD were selected. In some instances, data from multiple questions were compiled yielding OD and UCD sums equivalent to 8 times the number of questions included in the compilation. For individual parameters section, OD plus UCD sum to 7, based on the participant's selection of OD or UCD for 8 elements while answered 7 questions. The table shows the mean (based on n=30 participants) number of times OD or UCD were selected.

Based on the subjective feedback regarding the effectiveness of the two designs (OD vs. UCD) in promoting the SA of the participants, significant (α = 0.01, P < 0.0001) differences were observed in the participant responses. The majority of the responses supported the UCD display elements over the OD display elements. At every level of SA, i.e., perception ([*1]), comprehension ([*2]), and projection ([*3]), the UCD means were higher than the OD means. In addition, regarding the recall of information ([*4]), the participants reported that the UCD display elements were better. In the other two questions ([*5]), the participants were asked which display elements they preferred; the majority chose UCD over OD. For six of the eight parameters, most of the participant responses supported the UCD display elements over the OD display elements.

However, for two of the parameters, i.e., TP (**[6]) and blockage (**[6]), no statistically significant results were observed.

The percentage differences in the participants' perceived effectiveness of the two designs (OD vs. UCD) concerning SA, recall, and overall preference are summarized in Figure 5-7. These percentage differences are based on the values in Table 5-5, and are calculated as: (UCD mean – OD mean) / OD mean $\times$ 100. The participants' responses supported the effectiveness of UCD over OD for almost every criterion. The OD display element was preferred only in the case of blockage; however, this difference was not statistically significant. Referring to the overall opinion (Q1-7) in Figure 5-7, a significant difference in effectiveness was observed; the UCD display elements were rated 190% higher than the OD display elements.

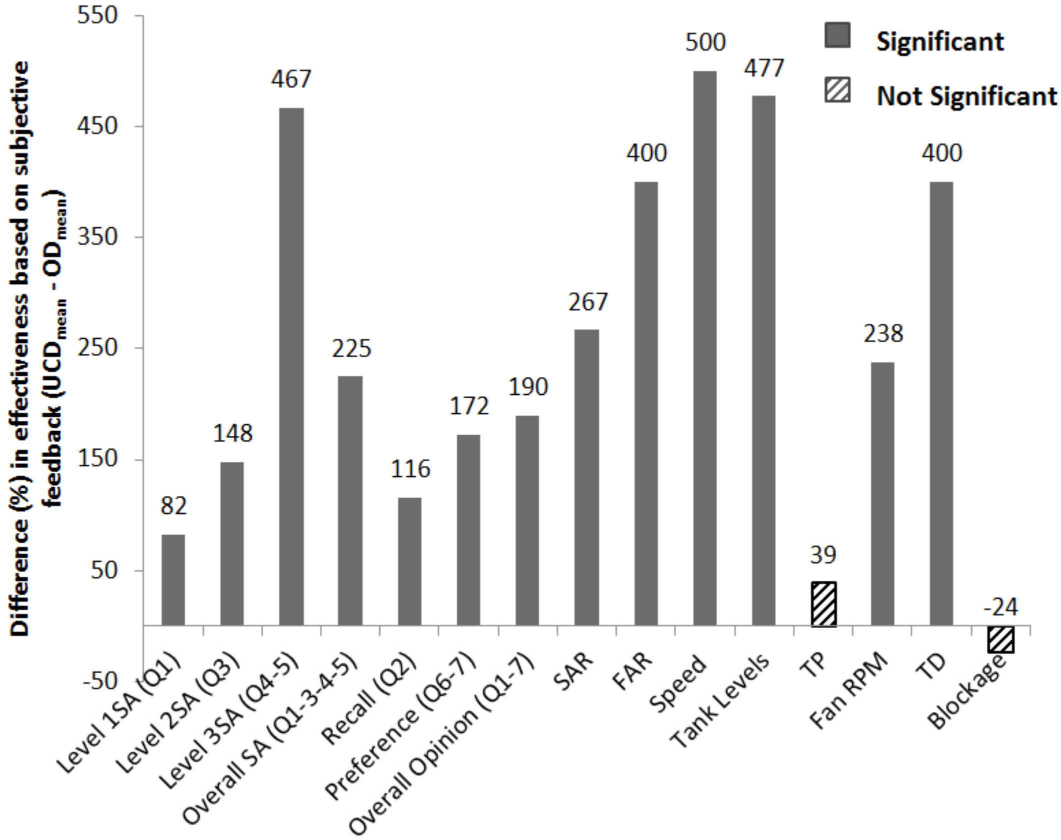


Figure 5-7. Subjective feedback from participants on the user-centered design (UCD) and original design (OD) regarding situation awareness (SA), recall, preference, overall effectiveness, and individual display elements. Values are percentage differences.

5.5 LIMITATIONS OF THE STUDY

We acknowledge two limitations associated with this study. The first limitation is that most of the participants were not experienced agricultural machinery operators. However, it is important to note that the UCD display elements generally enhanced the SA of the participants, even though most participants were not familiar with air seeder terminology. It is reasonable to assume that similar, if not better, SA enhancement would be reported by experienced agricultural machinery operators. The second limitation is that the study was conducted in a laboratory environment rather than in a field environment. The laboratory environment enabled us to minimize the influence of outside factors, although the downside is that context is lacking.

We also acknowledge the limitation in data analysis. For paired t-tests (Table 5-3 and Table 5-4) we compared the consolidated responses for three replications of OD vs. UCD (used n=30 for calculating the mean). Better precision and confidence in results could be obtained by not consolidating the responses for the replications (using n=90, instead of n=30), though the means, means differences and confidence intervals obtained would be approximately one third in values in comparison to the consolidated results. However, we have presented the mean differences as percentage differences which are independent of the scale. Furthermore, we tested only two variants of the design. Several additional design improvements were identified through this research. These changes will be implemented in a follow-up study that evaluates a complete UI.

5.6 CONCLUSIONS

Based on the objective and subjective results for participants who used both the UCD and OD display elements, we can confidently conclude that the UCD elements significantly improved the SA of the participants. For MWL, most of the UCD elements were rated higher than the corresponding OD elements. Evaluation of the results showed significant improvement in SA with the UCD elements (up to 11%; overall mean difference = 5.0 (4.8%), 95% CI (6.4728, 3.5939), $P < 0.0001$). For MWL, reductions of up to 19.7% (overall mean difference = -5.2 (-7.9%), $n = 30$, $\alpha = 0.05$) were observed with the UCD elements. The study participants also rated the overall performance of the newly designed UCD elements higher in comparison to the previous OD elements (overall mean difference = 27.3 (189.8%), 99% CI (35.150, 19.384), $P < 0.0001$).

CHAPTER 6

DESIGN AND EVALUATION OF THE USER-CENTERED INTERFACE FOR STUDYING SITUATION AWARENESS AND MENTAL WORKLOAD OF THE OPERATOR OF AIR SEEDER UNDER VARYING LEVELS OF AUTOMATION

6.1 INTRODUCTION

A UI is a means by which the user interacts with the target system. It is an important facet that helps the user to monitor, control and alter the target task environment (Liu 1997). Depending upon the level of automation (Parasuraman et al. 2000) and function allocation between man and machine (Bye et al. 1999), the role and goals of the operator may vary. However, efficiency and effectiveness of the operation, as well as workload and safety of the operator, depends on the information displayed to the user. Designers and researchers recommend that the typical UI should encapsulate and emphasize the “critical features” of the target environment (Liu 1997; Norman 1983; Sanchez and Duncan 2009; Vicente and Rasmussen 1992). Using the paradigm of situation awareness, Endsley et al. (2003) explained the goal-directed task analysis to determine the critical information needs of the user. Based on the situation awareness information needs of the operator, designers and engineers can design the UI by focusing on the essential interface elements to accomplish operational goals of the operator.

There are two important considerations for designing a UI: i) information requirements of the user, and ii) information presentation to the user. Information requirements of the user specify the quantity, type or variety of the information deemed necessary for the user to achieve his/her job-related goals. For example, for a car driver, knowledge of the current speed of the car is one of the critical information requirements - without the knowledge of the current speed of the vehicle, it would be difficult to drive safely and lawfully. Information presentation demonstrates the form, look, feel and mode of the information communicated to the user so that the processing and utilization of the information would be efficient. For example, the current speed of the car can be presented to the driver in many forms - using a numeric text, or by showing the movement of the animated needle on a dial gauge, or as a combination of both numeric text and animated needle, or by using a physical pointer moving on the dial gauge, or by any other design or means depending upon the imagination and resourcefulness of the designer.

In this study, we have designed and evaluated a driver interface for a tractor air seeder system. This work was accomplished in two phases. During the first phase, individual elements of the driver interface were designed and evaluated. Findings of the first phase of the study are discussed in chapter five (Rakhra and Mann 2018). During the second phase of the study, individual elements of the tractor air seeder interface were re-designed as a single interface based on the feedback and findings from the first phase of the study. Two versions of the newly designed interface have been evaluated and compared with the baseline interface (previously existing interface for tractor air seeder system). This manuscript discusses the findings of the second phase of the study.

6.2 AUTOMATION AND CONSEQUENCES

Automation has been introduced in almost every work domain with several expectations. Including the easing of the job (i.e., lower physical and mental work), decrement in the production cost, reduction in the errors, improvement in the efficiencies, and minimization of the impact of ineffective behavioral and physiological aspects of the humans on the quality and quantity of the work in general. Computers can be utilized to analyze vast amounts of information and data faster and more accurately and could project trends, provide decision making support or even implement the required action in almost any work domain. However, depending upon the allocation of the tasks and division of the work among the human and the system; the workload, the situation awareness and the outcome of the operation can be affected (Endsley 1996; Endsley and Kiris 1995; Endsley and Kaber 1999).

Parsuraman et al. (2000) explained automation as “full or partial replacement of the function previously carried out by the human operator.” They proposed four broad divisions of functions where automation can be applied: 1) information acquisition; 2) information analysis; 3) decision and action selection, and 4) action implementation. For example, in the agricultural domain, automation of information acquisition may involve “sensing and registering of the input data” (e.g., displaying of steering angle, tool depth, seed application rate are the examples of sensing and registering input data in the context of the tractor air seeder operation). Automation of the information analysis may include cognitive activities like memory, reasoning, judgment, logic, and inference. Understanding or recognition of higher or lower tool depth than the optimal tool depth is the example of information analysis. When the machine automatically performs this function, then it is termed as automation of information analysis. Automation of decision and action selection further extends the essence of information analysis by executing the appropriate decision and action based on the analysis. For example, if the analysis indicates that tool depth is higher than the optimum, then the decision would be to reduce the tool depth by selecting the tool depth lever or by other appropriate means. In action implementation mode, the system will automatically perform the appropriate action, by reducing tool depth to the optimum level.

Parasuraman and Riley (1997) mentioned that overreliance on automation, underutilization of the automation and, implementation of the function's automation without understanding the impact on human performance are the main contributors in the problems related to automation. Issues about out-of-the-loop performance (Endsley and Kiris 1995), weak interaction and feedback (Norman 1990), insufficient monitoring and ineffective decision making were a few of the several observed issues associated with the automation in the system. These types of concerns can be related to the 'lower level of situation awareness' of the operators of the automated systems (Endsley 1996).

6.3 INTERFACE DESIGN WITH A FOCUS ON SITUATION AWARENESS

The lower level of situation awareness (SA) of the operator could not only lead to ineffective, inadequate operational outcomes but also could be dangerous. Endsley (1996) described several catastrophic incidents of the airline crashes (Northwest Airlines MD-80 in 1987, US Air B-737 in 1989, or Korean Airlines Flight in 1983) which were directly/indirectly related to the lower level of the SA of the operators of the automated flight system.

The problem of the lower level SA of operators are usually caused due to the association of the automation in the system design by fundamentally shifting the role of the operator from 'active participant' to 'passive user/supervisor/monitor' (Byrne and Parasuraman 1996). The designing of an interface using a user-centered approach for improving the SA of the users can help in building more efficient interfaces.

Aviation is not the only area where a user-centered SA approach has been used extensively for analyzing and addressing the safety and operational concerns. In several other domains, like “power, transportation, cyber, space, unmanned and autonomous systems, maritime, command and control, oil drilling, and health care,” the concept of SA has been used to solve information and performance related issues (Endsley et al. 2015).

Taylor et al. (2010) developed and evaluated three interfaces for regenerative life support systems using the ‘ecological interface design’ which considers the user-centered approach to better support the situation awareness of the operators. Results of the study have indicated that the interfaces which presented ‘situation-rich’ information helped in better decision making.

In partially automated driving scenarios, adequate SA is essential for the safety of the drivers. During partially automated driving scenarios, human drivers are expected to take control of the situation whenever the situation demands attention (i.e., due to technology failure or technology limitation). However, as mentioned by Wulf et al. (2015), the ‘driver may not be able to overlook the whole situation, in the same way, he would do while driving manually,’ upkeep of adequate situation awareness is critical in partially automated driving scenarios. Wulf et al. (2015) recommend that to provide sufficient SA, the vehicle’s heads-up display in the peripheral field of view of the driver should display the secondary tasks. Also, controls for the secondary tasks should be located on the steering wheel to reduce the reaction time for steering corrections.

6.4 MATERIALS AND METHODS

6.4.1 Designing of the user interface for tractor air seeder

Understanding of the user's goals, decisions needed to achieve those goals, and information needs of the operator are the foundational elements in the design of UI to support the situation awareness of the operator (Endsley et al. 2003). For understanding the operation of the tractor air seeder system; operator's manuals, product specifications and other technical documenting had been reviewed during an internship completed by the first author. To further gain the understanding of the operational informational needs, decisions and goals of the operator of the tractor air seeder system, researcher informally discussed with the operators, and also performed ride-along with the farmers. By applying the GDTA approach, 23 informational needs were identified to address the various goals and sub-goals of the operator of the tractor air seeder system (Rakhra and Mann 2018).

Based on the identified information requirements and the baseline interface designed by Karimi et al. (2011),12 air seeder elements were shortlisted for the study. These elements were, seed level status (tank levels), fertilizer level status (tank levels), fan RPM, seed application rate, seed depth (tool depth), fertilizer application rate, fertilizer depth (tool depth), tool pressure, blockage status, desired path of the unit, desired location of the unit, and current speed of the unit. These shortlisted air seeder interface elements were re-designed by using User-Centered Design (UCD) and UI design principles (Endsley et al. 2003; Rakhra and Mann 2018). In-depth review of the different design principles and the details about the design and evaluation of the individual air seeder elements can be found in Rakhra and Mann (2014) and Rakhra and Mann (2018). Results of the assessment of these 12 air seeder elements demonstrated positive outcomes regarding the SA and MWL of the operators. However, these elements were compared with the baseline conditions on an individual basis, but not as a single interface showing all air seeder elements together. So, in the second phase of the study, all individually designed UCD based air seeder elements were re-designed and developed as a single interface computer application to compare it with the baseline conditions (interface based on the Karimi et al. 2011, Figure 6-1).

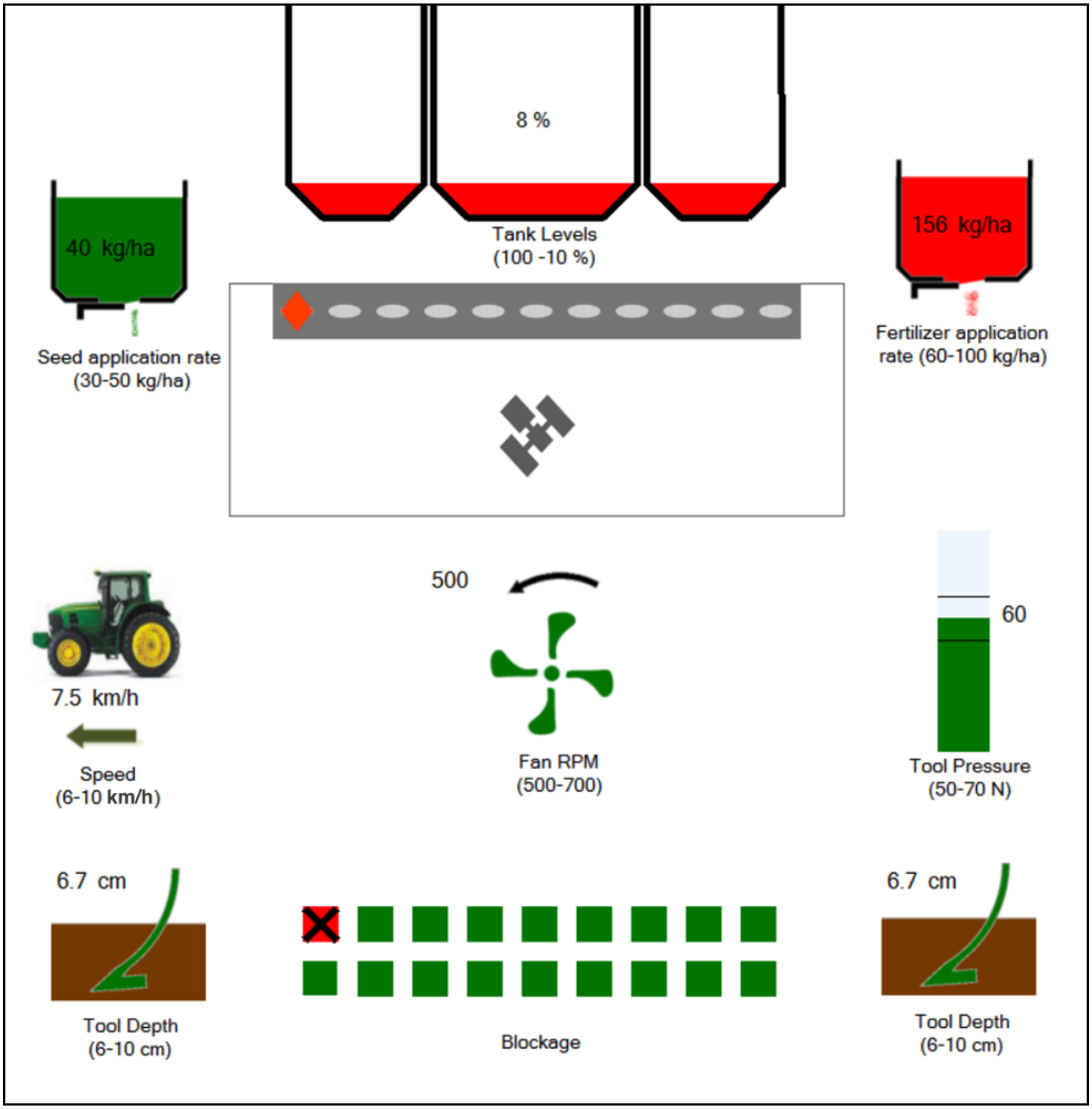


Figure 6-1. Baseline (Old) interface based on the design of Karimi et al. 2011.

Based on the feedback and outcomes from the first phase of the study, two variants of the UCD based interface were developed, viz., UCD1 (Figure 6-3) and UCD2 (Figure 6-4).

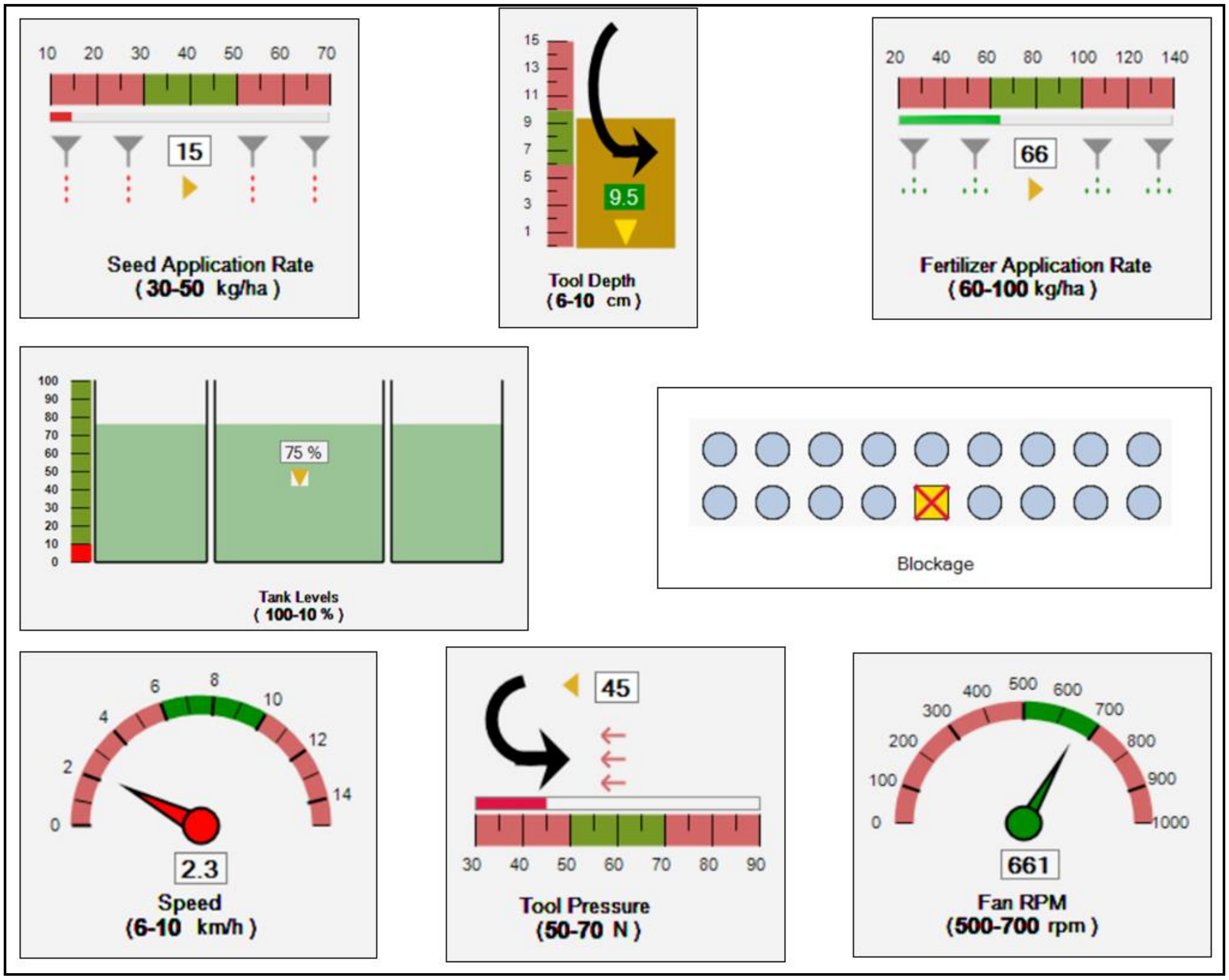


Figure 6-2. UCD based interface elements from phase 1 of the study.

UCD1(Figure 6-3) was developed by combing all individual UCD based elements on a single interface from phase one of the study (Figure 6-2).

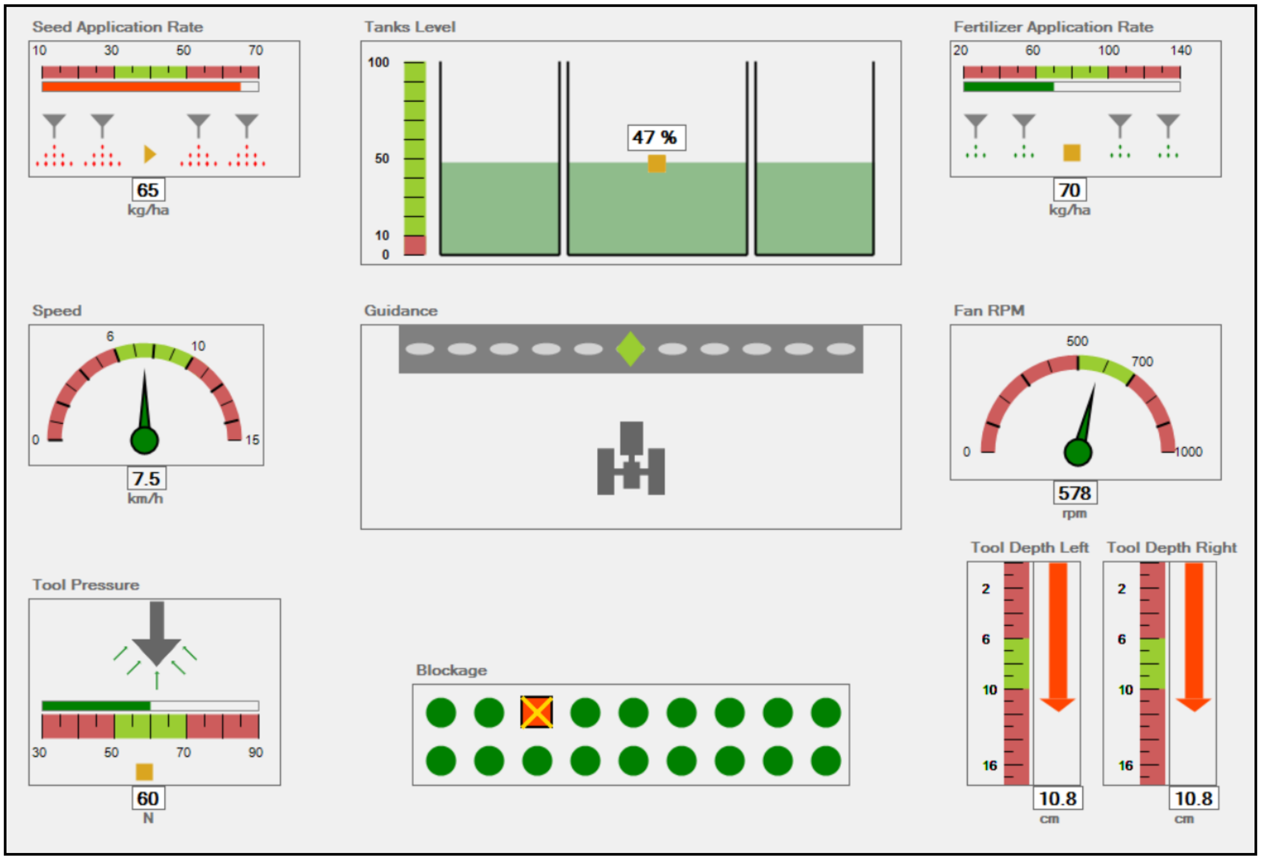


Figure 6-3. Version 1 of UCD (UCD1) based Interface elements.

Based on the results and feedback from the first phase of the study, additional modifications in the three UCD1 interface elements were made.

1. Referring to the Tool Depth element in Figure 6-2 and Figure 6-3, some participants showed difficulty in making sense of a stationary tool while soil levels were varying. Participants also indicated a preference for the scale starting from the top to the bottom for the tool depth element. Accordingly, in the UCD1 design (Figure 6-3), these changes were incorporated.

2. Referring to the Blockage element in Figure 6-2 and Figure 6-3, most of the participants indicated a preference for green color over light blue color as the indication of correct state (non-blockage state) in the blockage element of the air seeder unit. Therefore, the color of the Blockage elements is changed from blue to green (Figure 6-3).
3. Regarding Tool Pressure element (Figure 6-2 and Figure 6-3), results of the evaluations from the first phase had indicated that the UCD tool pressure element caused high mental workload - approximately 25% more than baseline conditions (Rakhra and Mann 2018). During the subjective feedback and discussions, many users indicated difficulty in inferring information from the Tool Pressure element. Accordingly, changes have been made in the Tool Pressure element design in the UCD1 interface (Figure 6-3).

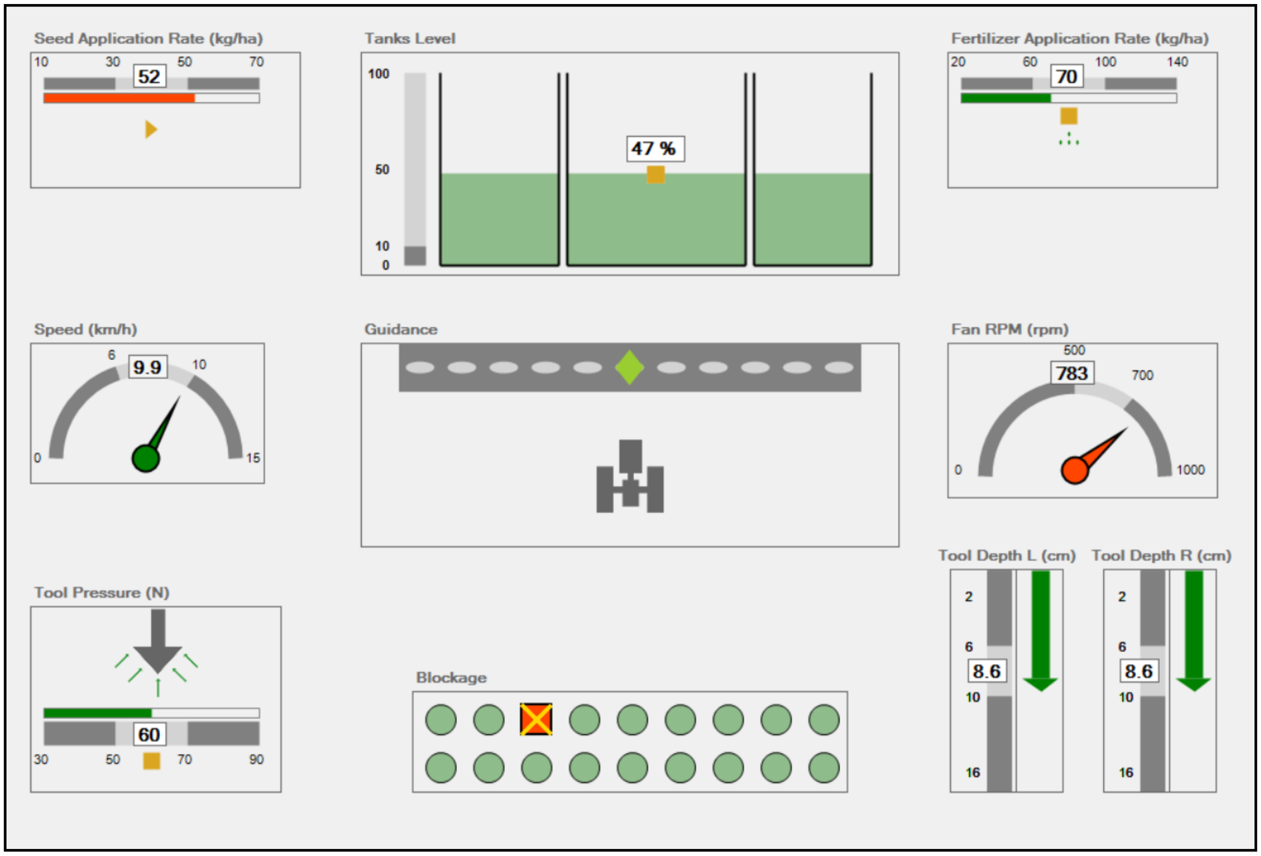


Figure 6-4. Version 2 of the UCD (UCD2) based interface elements.

Also, a second version of the user-centered design interface (UCD2, Figure 6-4) was developed to test and compare it with baseline, and UCD1 interface. In UCD2, basic design and functionality were kept the same as UCD1. However, a few other design alterations were made, such as:

1. The coloring of the scales was modified to grayscale from red/green.
2. Marks on the scales were removed.
3. In the seed application rate, an additional animation showing the falling of the seeds was removed.

4. In fertilizer application rate, only one animation representing the falling fertilizer was displayed, instead of the four animations as in UCD1.
5. Another significant change regarding the placement of numeric readings was made; readings were moved from the bottom of each element to the middle of the scale (e.g., see the seed application reading in Figure 6-3 and Figure 6-4).

6.4.2 Evaluations of the interfaces

This study was completed in the Agricultural Ergonomics Laboratory in the Department of Biosystems Engineering at the University of Manitoba. The experimental protocol for this study received human ethics approval from the Education/Nursing Research Ethics Board of the University of Manitoba (Appendix A.1).

6.4.2.1 Participants

Thirty individuals participated in the study. The majority of the participants were University of Manitoba students. Out of the 30 participants, 20 were male, and 10 were female. Ages ranged from 19 to 52 years with a mean age of 28.4 years. Only 10 participants indicated prior driving experience and most of the participants (29 out of 30) had no previous experience with agricultural machines or a driver interface for an agricultural machine.

6.4.2.2 Experiment Design and Procedure

Three types of driver interfaces (Baseline, UCD1, and UCD2) in two conditions (low-level automation and high-level automation) with different steering types (manual and auto steering) were compared for SA and MWL in the lab environment. Software simulation was developed in the Visual Basic programming language using Microsoft

Visual Studio Express 2013. For additional details about the experiment execution and simulation coding, please see Appendix E and F.

The full experiment was divided into two sessions, viz. low-level automation, and high-level automation sessions. In the low-level automation session, the user was responsible for observing as well as correcting the situation. As per the Parasuraman et al. (2000) analogy, during the low-level automation, automation support for information acquisition and information analysis functions were always provided. However, no support for decision and action selection, or action implementation was provided. For rectifying the situation, the user was required to click on the incorrect (out-of-range) parameter on the interface (e.g., see the seed application rate, blockage, and tool depth in Figure 6-3, and seed application rate, fan rpm and blockage in Figure 6-4).

During the high-level automation session, automation support for all four types of the functions (i.e., information acquisition, information analysis, decision and action selection, and action implementation) were provided (Parasuraman et al. 2000). The user was only responsible for observing the situation. The user was not allowed to correct the situation by clicking the mouse.

Regarding two types of steering configurations (Figure 6-5), the manual steering configuration involved the variation in the guidance bar; however, in auto-steering configuration, the guidance bar was programmed to be non-variable.

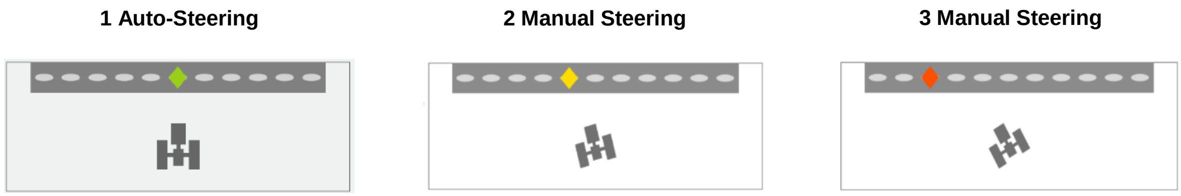


Figure 6-5. Depicting changes in guidance bar in auto-steering and manual steering configurations. In auto-steering mode (1), the system was automatically controlling the variation in the steering, and always displaying the corrected steering position (green-diamond) to the user. Whereas in manual-steering mode, the user was responsible for fixing the steering variation (during the lower automation level), change in steering movement was visible to the user (green, yellow or red diamonds depending upon the variation in the steering).

During the low-level automation session, three types of driver interfaces simulations in two configurations (i.e., manual steering and auto steering) were displayed two times on the computer monitor. Order of the appearance of 6 types of interface variants was programmed to be random. Also, the time interval for subsequent simulations was kept unequal. The actual duration of the random interfaces to stay on screen before ‘queries’ would appear was kept as 240 s, 120 s, 180 s, 120 s, 240 s, 180 s, 120 s, 180 s, 240 s, 180 s, 120 s, and 240 s. The same procedure was adopted for the high-level automation session.

After every simulation, queries related to the SA and MWL were asked to the user (Figure 6-6). During each session (low-level or high-level automation), in total, 12 times questions were asked to the user. As recommended in the Situation Awareness Global Assessment Technique (SAGAT) (Endsley et al. 2003), the questions asked were related to various levels of SA. Users responses received under ‘VALUE’, ‘STATUS’, and ‘FUTURE STATUS’ (Figure 6-6) were related to the level 1 (perception), level 2 (comprehension) and level 3 (projection) type of SA, respectively (Endsley et al. 2003). For the sake of simplicity, responses recorded under the ‘TREND’ section were not added to either perception or comprehension, but evaluated separately as trend identification by the users.

In addition to the SA, MWL responses of the users based on a unidimensional, easy-to-administer, and less-intrusive subjective Integrated Workload Scale (IWS) (Pickup et al. 2005) were also recorded by the simulation program during the experiment (Figure 6-6, last row-mental workload).

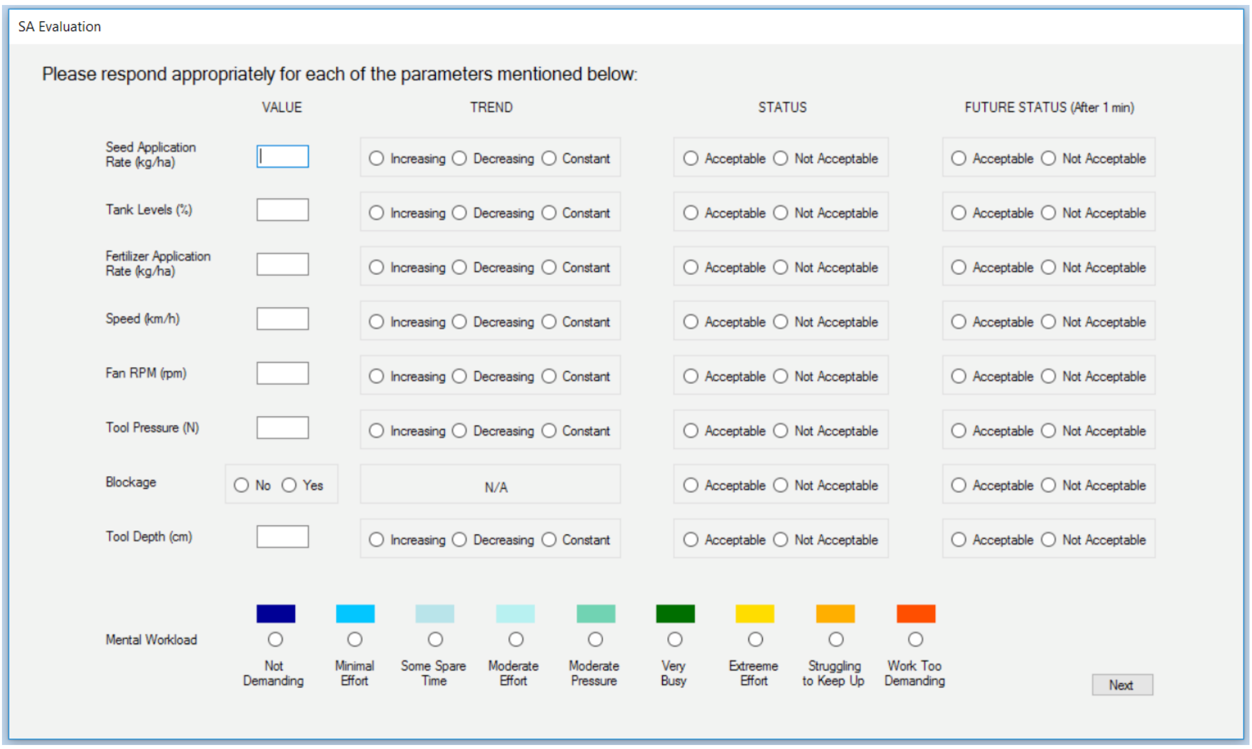


Figure 6-6. Snapshot of the form used to collect user responses after every simulation of the interfaces.

To minimize the learning effect among low-level and high-level automation, the sequence of the sessions varied (i.e., 15 participants had low-level automation session as their first session, and other 15 participants had high-level automation session as their first session).

In addition to the responses of the users related to SA, answers related to MWL were also recorded by the computer program. Subjective feedback regarding the three types of UIs was also collected after each experimental session using a paper-pencil format. After every low-level and high-level automation session, questionnaires were provided to the users along with the image of the three interfaces. Users were required to rate the three interfaces regarding the various criteria mentioned in the questionnaire (Table 6-1).

Table 6-1. Questions asked of the participants at the end of every experiment session to rate the three interfaces.

No.	Criteria
1	Rate the interfaces in terms of perception of the information.
2	Rate the interfaces in terms of recalling (or remembering) of the information.
3	Rate the interfaces in terms of understanding or comprehension of the situation.
4	Rate the interfaces in terms of trend identification.
5	Rate the interfaces in terms of prediction of the future situation.

6.5 RESULTS AND DISCUSSION

Experimental design can be described as 2 x 2 x 3 factors repeated measure (within subjects) design. Subjects acted as blocks, and each subject experienced all the treatments. For general considerations regarding the experiment design, please see the Appendix C.

The complete experiment involved two levels of automation (low & high), two levels of steering type (manual & auto) and three levels of interface-design type (Baseline, User-Centered 1 and User-Centered 2). Automation-levels, steering-type, and interface-design-type were considered as three independent variables, and situation awareness and mental workload were considered as two dependent variables.

Data were analyzed using General-Linear Model (GLM) involved mixed procedure in Mini-Tab. In the model, subjects were treated as random factors, whereas automation, steering type, and interface-design were treated as fixed factors.

6.5.1 Situation awareness

Based on the user's responses to the queries (Figure 6-6); level 1 (‘Value’), level 2 (‘Status’) and level 3 (‘Future Status’) SA were evaluated. Also, Overall-effectiveness was evaluated based on the responses related to the ‘Trend’ and SA (level 1, 2 and 3) together.

Level 1 SA Based on GLM involving mixed procedure in Mini-Tab, level 1 type SA (as response variable) versus automation (fixed factor), steer-type (fixed factor), interface-design (fixed factor), and subjects (random factor-block) were evaluated. We did not observe any significant main effect due to the automation, steer-type, and interface-design. However, for level 1 type SA, the interaction between automation type and steer type yielded significant results. We observed $F (1, 319) = 5.63$, $P = 0.018$ and $R^2 = 0.496$. Mean values for the four different combinations are shown in Figure 6-7.

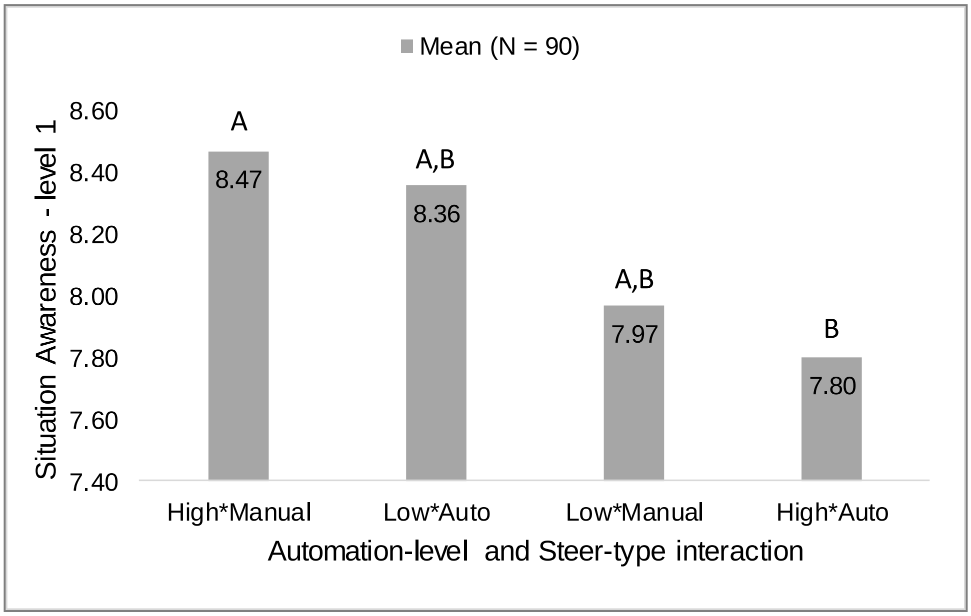


Figure 6-7. Mean values of level 1 type SA for automation-level*Steer-type interaction. Letters (A, B…) indicate grouping information based on Fisher's LSD pairwise comparison. Values that do not share a letter are significantly different.

Based on the results, users demonstrated higher level 1 SA (8.47 and 8.36) when they were interacting with the system, and their interaction level was intermediate. The lowest level of SA (7.80) was observed when users were not doing anything, but only watching the system (i.e., when the automation level was high, and steering was also auto). In this scenario, users were not responsible for making any corrections if things went wrong. Based on the Fisher's LSD pairwise comparison, the difference between the High*Manual and High*Auto is also significant ($p = 0.035$, 95% CI (0.05, 1.29)). However, Tukey's pairwise comparison does not consider this a significant difference.

We believe that the lowest level of SA in the highest automated scenario is due to the out of loop effect. Due to the fully automated conditions, the users did not find any need to be deeply and vigilantly monitoring the situation as the system was doing all the work for the users. Eventually, users became out of the loop. However, we did not find that the least automated scenario (low automation with manual steering) was the best. Though users demonstrated little better SA than the fully autonomous situation (7.97 vs. 7.80), the difference is not significant. We believe that during the least automated scenario, users were unable to perceive the desired information while they were busy in correcting the rapidly changing situation.

Level 2 SA For level 2 type SA (as response variable) versus automation (fixed factor), steer-type (fixed factor), interface-design type (fixed factor) and subjects (random factor-block), based on GLM involving mixed procedure in Mini-Tab, we did not find any significant effect due to steer-type (auto and manual) or interface-design type (Baseline, User-Centered 1 and User-Centered 2). However, we observed highly significant results due to the automation type (high and low). Obtained statistical values were; F (1,319) = 70.01, p= 0.000, and R^2 = 0.470. Mean values for high and low automation levels are mentioned in Figure 6-8.

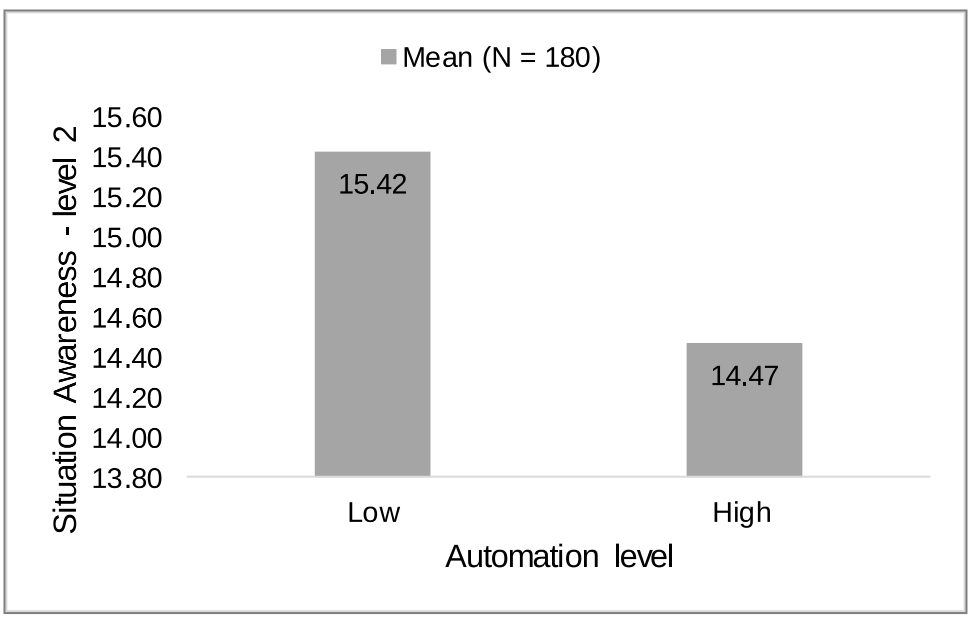


Figure 6-8. Mean values of level 2 type SA for low and high automation levels.

Users demonstrated higher level 2 SA in low automation scenario in comparison to the high automation (15.42 versus 14.47) with highly significant difference of means (difference = 0.96 (SE = 0.114), p= 0.000, and 95% CI is 0.73 to 1.18).

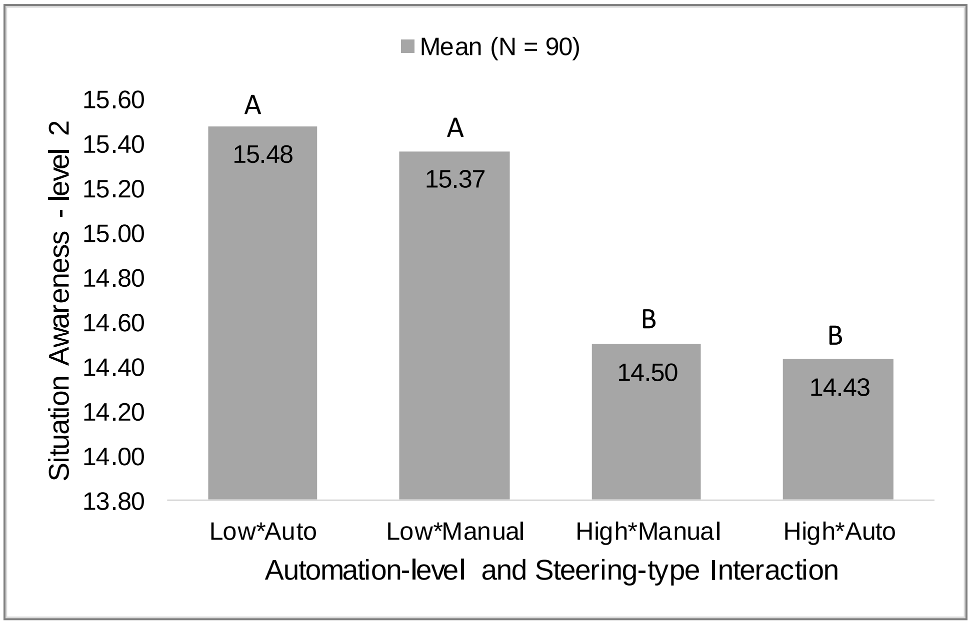


Figure 6-9. Mean values of level 2 type SA for automation-level*Steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

We did not observe any overall significant interaction effect for level 2 SA in the GLM; however, based on the Tukey's pairwise comparison, we observed a significant difference in some pairs due to the interaction of automation and steer-type (Figure 6-9); automation and interface design (Figure 6-10); and automation, steer-type and interface-design (Figure 6-11). Means of the low automation combinations were found lower than the means of the high automation combinations.

Results of the Tukey's simultaneous tests for differences of means showing statistically significant differences are listed below (Table 6-2).

Table 6-2. Results of the Tukey's simultaneous tests for differences of means (N = 90) of Automation*Steer-Type interaction showing statistically significant differences.

Difference of Automation*Steer-Type	Difference of Means	SE of Difference	Simultaneous 95% CI	T-Value	Adjusted P-Value
(Low*Auto) – (High*Auto)	1.04	0.16	(0.63, 1.46)	6.47	0.000
(Low*Manual) - (High*Auto)	0.93	0.16	(0.52, 1.35)	5.78	0.000
(Low*Auto) - (High*Manual)	0.98	0.16	(0.56, 1.39)	6.05	0.000
(Low*Manual) - (High*Manual)	0.87	0.16	(0.45, 1.28)	5.37	0.000
Individual confidence level = 98.93%					

Means of the interaction of automation-level and interface-design listed in Figure 6-10. A higher level of situation awareness was observed with the lower level of automation. Within each automation level (low or high), interaction with the interface-design type (Old, UCD1 or UCD2) did not yield any significant difference (as indicated by the letters A and B based on Tukey's pairwise comparison).

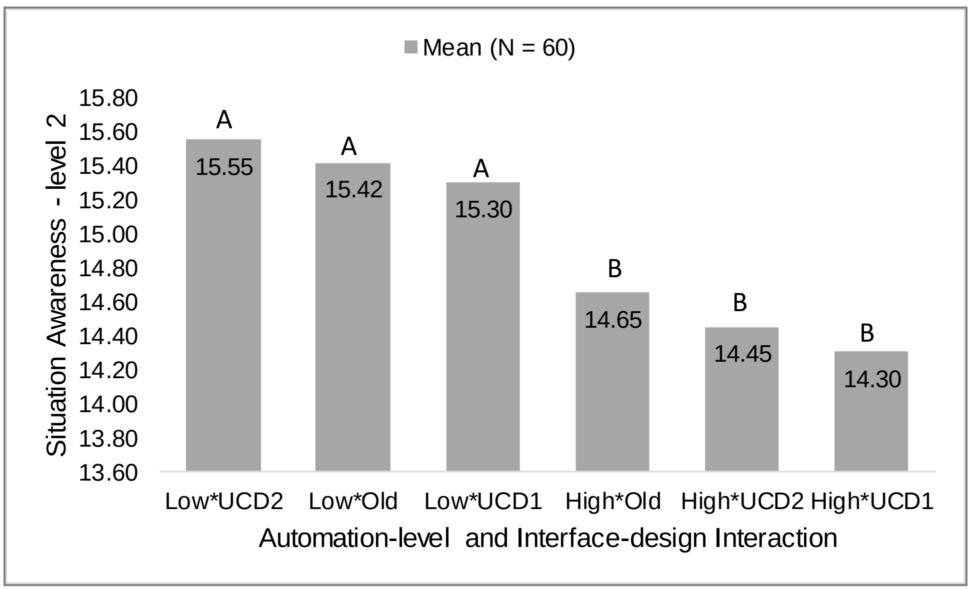


Figure 6-10. Mean values of level 2 type SA for automation-level*interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Results of the Tukey's simultaneous tests for differences of means showing statistically significant results are listed in Table 6-3.

Table 6-3. Results of the Tukey's simultaneous tests for differences of means (N = 60) of Automation*Interface-design interaction showing statistically significant differences.

Difference of Automation*Interface-design-Type	Difference of Means	SE of Difference	Simultaneous 95% CI	T-Value	Adjusted P-Value
(Low*Old) - (High*Old)	0.77	0.20	(0.20, 1.33)	3.88	0.001
(Low*UDD1) - (High*Old)	0.65	0.20	(0.09, 1.21)	3.29	0.013
(Low*UCD2) - (High*Old)	0.90	0.20	(0.34, 1.46)	4.55	0.000
(Low*Old)- (High*UCD1	1.12	0.20	(0.55, 1.68)	5.65	0.000
(Low*UCD1) - (High*UCD1	1.00	0.20	(0.44, 1.56)	5.06	0.000
(Low*UCD2) - (High* UCD1)	1.25	0.20	(0.69, 1.81)	6.32	0.000
(Low*Old) - (High* UCD2)	0.97	0.20	(0.40, 1.53)	4.89	0.000
(Low*UCD1) - (High* UCD2)	0.85	0.20	(0.29, 1.41)	4.30	0.000
(Low*UCD2) - (High* UCD2)	1.10	0.20	(0.54, 1.66)	5.56	0.000
Individual confidence level = 99.53%					

Based on the Tukey's pairwise comparison, we observed a significant difference in some pairs due to the interaction of automation-level, steer-type and interface-design. Means of the low automation combinations are found higher than the means of the high automation combination (Figure 6-11).

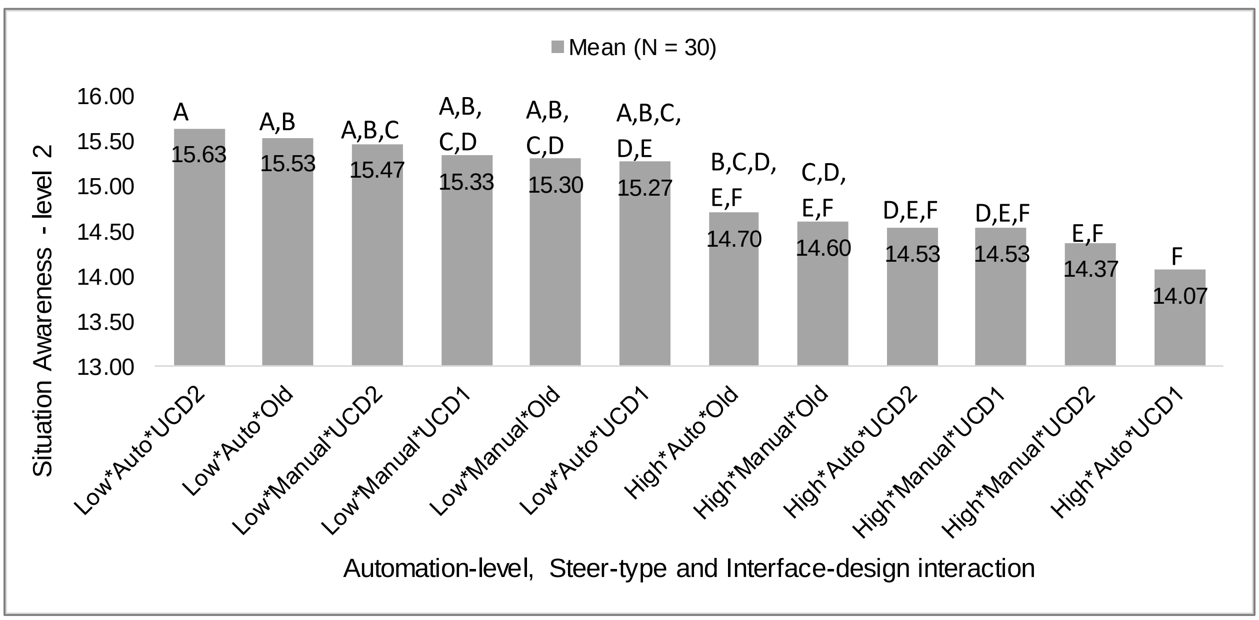


Figure 6-11. Mean values of level 2 type SA for automation-level, Steer-type, and Interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Results of the detailed Tukey's pairwise comparison showing significant differences are listed in Table 6-4.

Table 6-4. Results of the Tukey's simultaneous tests for differences of means (N = 30) of Automation-level*Steer-type*Interface-design interaction showing statistically significant differences.

Difference of Automation*Steer-Type*Interface-design-Type	Diff. of Means	SE of Diff.	Simultaneous 95% CI	T-Value	Adjusted P-Value
(Low*Auto*UCD2) -(High*Auto*Old)	0.93	0.28	(0.02, 1.85)	3.34	0.040
(Low*Auto*Old) -(High*Auto*UCD1)	1.47	0.28	(0.55, 2.38)	5.24	0.000
(Low*Auto*UCD1) -(High*Auto*UCD1)	1.20	0.28	(0.29, 2.11)	4.29	0.001
(Low*Auto*UCD2) -(High*Auto*UCD1)	1.57	0.28	(0.65, 2.48)	5.60	0.000
(Low*Manual*Old) -(High*Auto*UCD1)	1.23	0.28	(0.32, 2.15)	4.41	0.001
(Low*Manual*UCD1) -(High*Auto*UCD1)	1.27	0.28	(0.35, 2.18)	4.53	0.000
(Low*Manual*UCD2) -(High*Auto*UCD1)	1.40	0.28	(0.49, 2.31)	5.00	0.000
(Low*Auto*Old) -(High*Auto*UCD2)	1.00	0.28	(0.09, 1.91)	3.57	0.018
(Low*Auto*UCD2) -(High*Auto*UCD2)	1.10	0.28	(0.19, 2.01)	3.93	0.005
(Low*Manual*UCD2) -(High*Auto*UCD2)	0.93	0.28	(0.02, 1.85)	3.34	0.040
(Low*Auto*Old) -(High*Manual*Old)	0.93	0.28	(0.02, 1.85)	3.34	0.040
(Low*Auto*UCD2) -(High*Manual*Old)	1.03	0.28	(0.12, 1.95)	3.69	0.012
(Low*Auto*Old) -(High*Manual*UCD1)	1.00	0.28	(0.09, 1.91)	3.57	0.018
(Low*Auto*UCD2) -(High*Manual*UCD1)	1.10	0.28	(0.19, 2.01)	3.93	0.005
(Low*Manual*UCD2)-(High*Manual* UCD1)	0.93	0.28	(0.02, 1.85)	3.34	0.040
(Low*Auto*UCD1) -(High*Manual*UCD2)	0.90	0.28	(-0.01, 1.82)	3.22	0.058
(Low*Auto*UCD2) -(High*Manual*UCD2)	1.27	0.28	(0.35, 2.18)	4.53	0.000
(Low*Manual Old) -(High*Manual*UCD2)	0.93	0.28	(0.02, 1.85)	3.34	0.040
(Low*Manual*UCD1) -(High*Manual*UCD2)	0.97	0.28	(0.05, 1.88)	3.46	0.027
(Low*Manual*UCD2) -(High*Manual*UCD2)	1.10	0.28	(0.19, 2.01)	3.93	0.005

Individual confidence level = 99.88%

As we see in the above scenarios for level 2 type of SA, users have demonstrated a higher level of SA in lower automation levels with the various combinations of auto/manuals steering and UCD2/UCD1/Baseline interface-design. Furthermore, low automation in conjunction with auto/manual steering with the UCD2 design performed better than UCD1 or Baseline design (Figure 6-11). We believe that lower level 2 SA in the high automation scenario is due to the out of loop syndrome. In the high automation scenario, users were not responsible for making any corrections, which makes users less involved. This lower involvement made users out of the loop, which further caused a decrement in the level 2 SA of the users.

Level 3 SA Based on GLM involving mixed procedure in Mini-Tab; level 3 type SA (as response variable) versus automation (fixed factor), steer-type (fixed factor), interface-design (fixed factor) and subjects (random factor-block), highly significant main effect due to automation was observed. Resulting values are; F (1, 319) = 23.48, P = 0.000, and R^2 = 0.397. The difference between the means of low automation and high automation were observed to be significant (mean difference = -1.000, 95 % CI (-1.406, -0.594), P = 0.000). Mean values observed are listed in Figure 6-12.

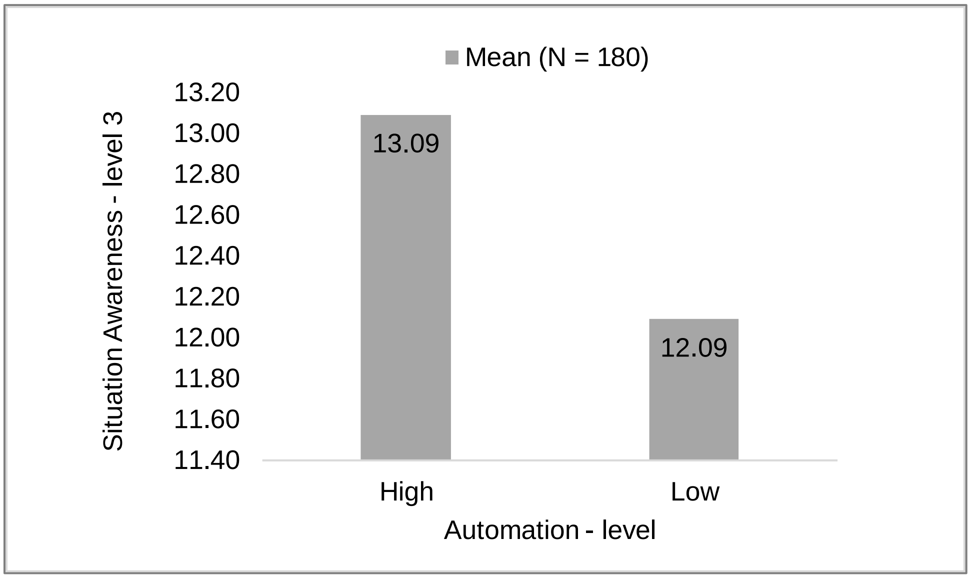


Figure 6-12. Mean values of level 3 type SA for high and low automation levels. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

We did not observe any overall significant interaction effect for level 3 SA in the GLM; however, based on the Tukey's pairwise comparison, we observed a significant difference in some pairs due to the interaction of automation and steer-type (Figure 6-13); automation and interface design (Figure 6-14); and automation, steer-type and interface-design (Figure 6-15). Users demonstrated higher level 3 SA awareness in the high-automation scenario in contrast to the low-automation scenario (Figure 6-13). The least SA was observed when no automation support was provided.

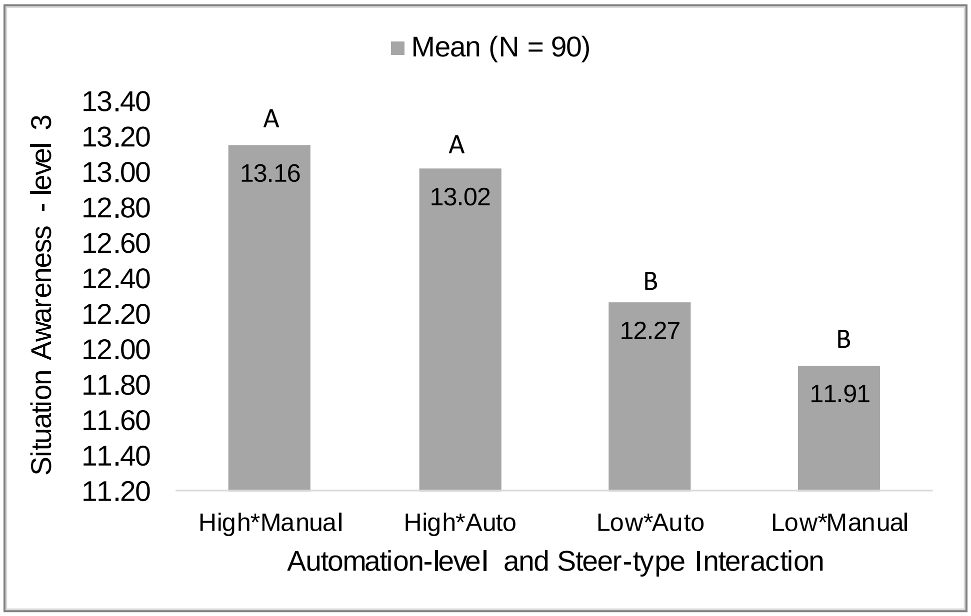


Figure 6-13. Mean values of level 3 type SA for automation-level and steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Based on Tukey's pairwise comparison, statistically different pairs are mentioned in Table 6-5.

Table 6-5. Results of the Tukey's simultaneous tests for differences of means (N = 90) of Automation*Steer-type interaction showing statistically significant differences.

Difference of Automation*Steer-type	Difference of Means	SE of Difference	Simultaneous 95% CI	T-Value	Adjusted P-Value
(Low*Auto) - (High*Auto)	-0.76	0.29	(-1.51, -0.01)	-2.59	0.047
(Low*Manual) - (High*Auto)	-1.11	0.29	(-1.86, -0.36)	-3.81	0.001
Low*Auto) - (High*Manual)	-0.89	0.29	(-1.64, -0.14)	-3.05	0.012
(Low*Manual) - (High*Manual)	-1.24	0.29	(-1.99, -0.50)	-4.26	0.000
Individual confidence level = 98.93%					

Based on the Tukey's pairwise comparison, a few significantly different pairs due to the automation*interface-design interaction are mentioned in Figure 6-14. Means that do not share a letter are significantly different.

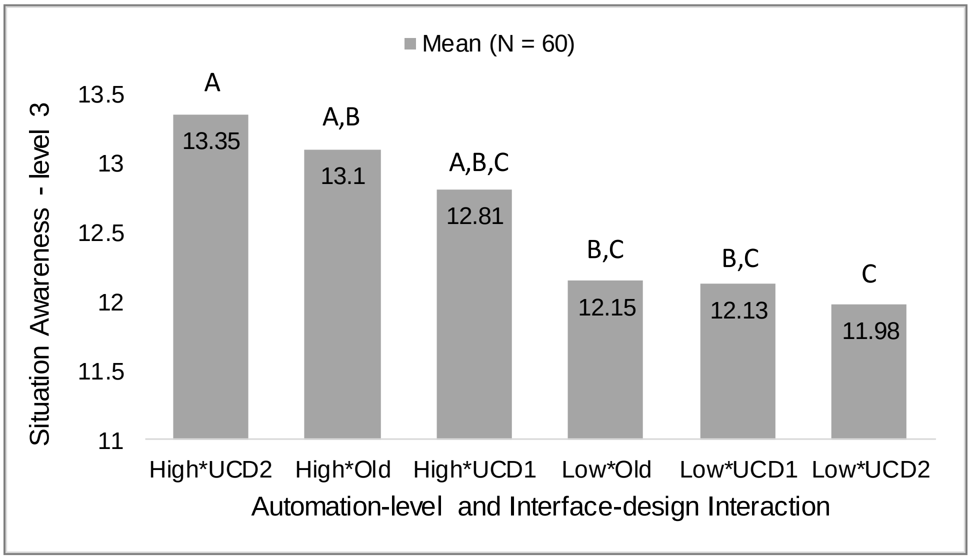


Figure 6-14. Mean values of level 3 type SA for automation-level and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Details of the Tukey's pairwise comparison for statistically different pairs are mentioned in Table 6-6.

Table 6-6. Results of the Tukey's simultaneous tests for differences of means (N = 60) of Automation*Interface-design interaction showing statistically significant differences.

Difference of Automation*Interface-design	Difference of Means	SE of Difference	Simultaneous 95% CI	T-value	Adjusted P-Value
(Low*UCD2) - (High*Old)	-1.12	0.36	(-2.14, -0.10)	-3.12	0.022
(Low*Old) - (High*UCD2)	-1.20	0.36	(-2.22, -0.18)	-3.36	0.010
(Low*UCD1) - (High*UCD2)	-1.22	0.36	(-2.24, -0.20)	-3.40	0.009
(Low*UCD2) - (High*UCD2)	-1.37	0.36	(-2.39, -0.35)	-3.82	0.002
Individual confidence level = 99.53%					

Also, based on the Tukey's pairwise comparison, a few significantly different pairs due to the automation*steer-type*interface-design interaction are mentioned in Figure 6-15. Means that do not share a letter are significantly different.

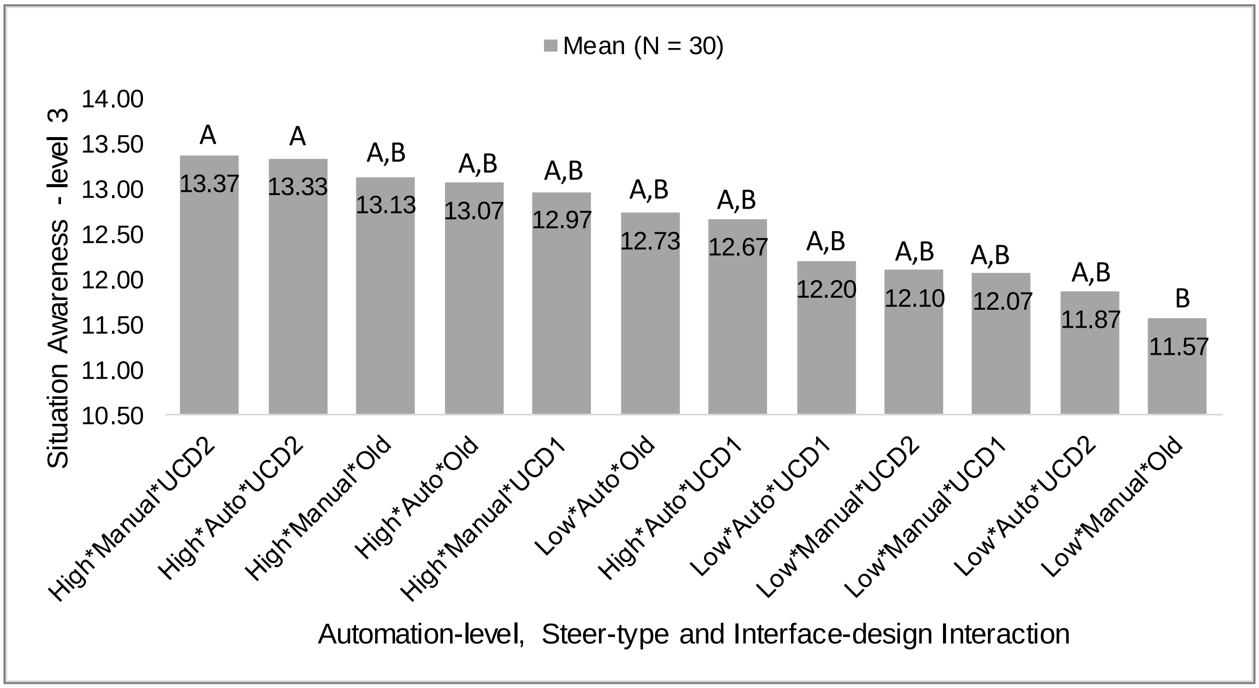


Figure 6-15. Mean values of level 3 type SA for automation-level and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Results of the detailed Tukey's pairwise comparison showing significant difference are listed in Table 6-7.

Table 6-7. Results of the Tukey's simultaneous tests for differences of means (N = 30) of Automation*Steer-type*Interface-design interaction showing statistically significant differences.

Difference of Automation*Steer-type*Interface-design	Diff. of Means	SE of Diff.	Simultaneous 95% CI	T-Value	Adjusted P-Value
(Low*Manual*Old) - (High*Auto*UCD2)	-1.77	0.51	(-3.42, -0.12)	-3.49	0.024
(Low*Manual*Old) - (High*Manual*UCD2)	-1.80	0.51	(-3.45, -0.15)	-3.56	0.019
Individual confidence level = 99.88%					

Based on the results of level 3 SA, we observed that users demonstrated higher level 3 SA in the high automation scenario (Figure 6-12), which is quite the opposite to the trend shown in level 2 SA (Figure 6-8). As far as the interaction effect of automation, steer-type, and interface-design are concerned, the UCD2 interface performed better than UCD1 and baseline conditions in most of the scenarios. However, the difference is not statistically significant for similar conditions.

6.5.2 Trend

Based on GLM involving mixed procedure in Mini-Tab, the trend (as response variable) versus automation (fixed factor), steer-type (fixed factor), interface-design (fixed factor) and Subjects (random factor-block), we did not observe any significant main effect due to the steer-type and interface-design. However, the main effect due the automation was found to be highly significant. We observed F (1,319) = 38.19, P = 0.000 and R^2 = 0.500. Mean values for high and low automation levels are mentioned in Figure 6-16. Difference between the means of low automation and high automation were observed to be significant (mean difference = -1.33 (SE = 0.215), 95 % CI (-1.75, -0.91), P = 0.000).

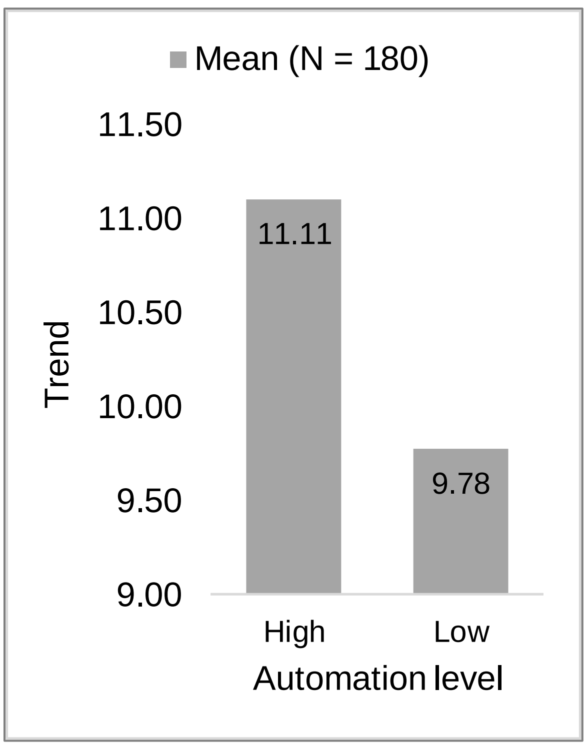


Figure 6-16. Mean values of the Trend identification during the high and low automation scenario.

Further, based on GLM using mixed procedure, significant interaction effect due to automation*interface-design was observed (F (2,319) = 3.06, P = 0.048), R^2 = 0.500).

Means of this interaction effect are mentioned in Figure 6-17. However, no other significant interaction effect has been observed in the GLM.

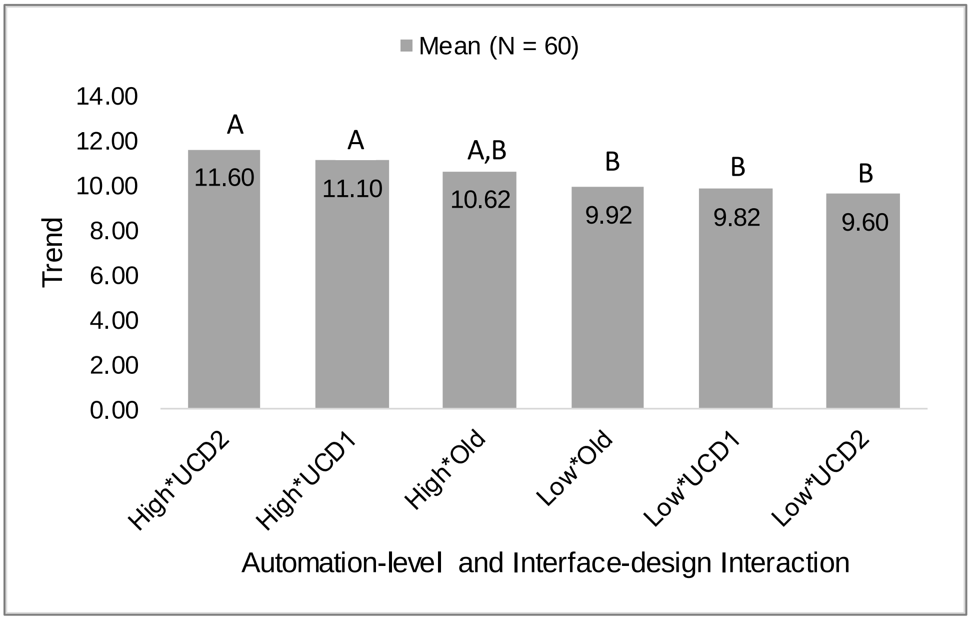


Figure 6-17. Mean values of the Trend for automation-level and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Results of the detailed Tukey's pairwise comparison showing significant difference are listed in Table 6-8.

Table 6-8. Results of the Tukey's simultaneous tests for differences of means (N = 60) of Automation*Interface-design interaction showing statistically significant differences.

Difference of Automation*Interface-design	Difference of Means	SE of Difference	Simultaneous 95% CI	T-value	Adjusted P-Value
(Low*Old) - (High*UCD1)	-1.18	0.37	(-2.24, -0.12)	-3.18	0.018
(Low*UCD1) - (High*UCD1)	-1.28	0.37	(-2.34, -0.22)	-3.45	0.007
(Low*UCD2) - (High*UCD1)	-1.50	0.37	(-2.56, -0.44)	-4.03	0.001
(Low*Old) - (High*UCD2)	-1.68	0.37	(-2.74, -0.62)	-4.52	0.000
(Low*UCD1) - (High*UCD2)	-1.78	0.37	(-2.84, -0.72)	-4.79	0.000
(Low*UCD2) - (High*UCD2)	-2.00	0.37	(-3.06, -0.94)	-5.37	0.000
Individual confidence level = 99.53%					

Based on the Tukey's pairwise comparison, in some pairs significant difference due to the interaction have been observed (i.e., automation*steer-type, and automation*steer-type*interface-design). Mean values of automation*steer-type are mentioned in Figure 6-18.

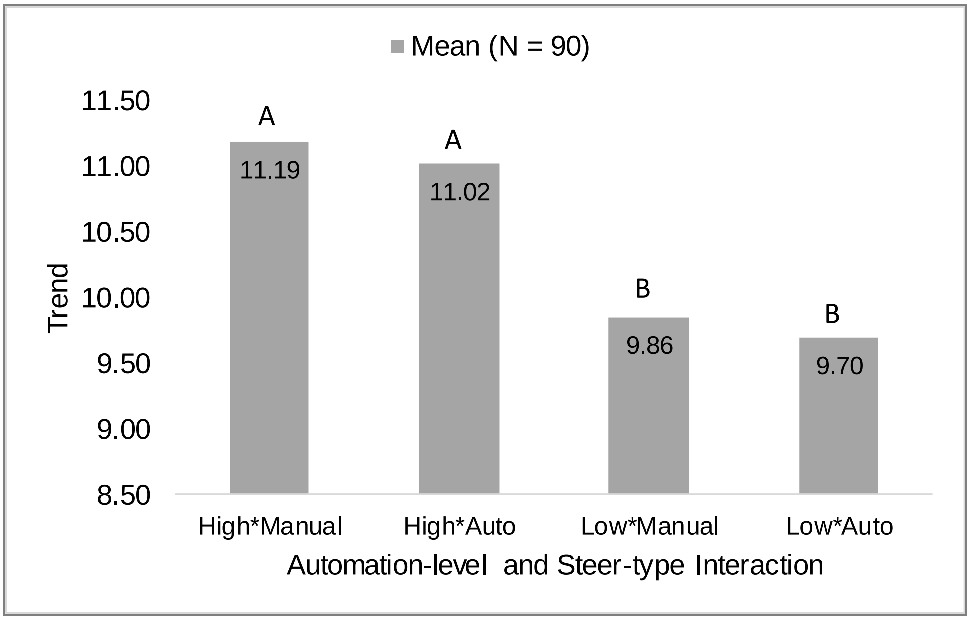


Figure 6-18. Mean values of the Trend identification for automation-level and steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Details of the significant different pairs based on the Tukey's pairwise comparison is mentioned in Table 6-9.

Table 6-9. Results of the Tukey's simultaneous tests for differences of means (N = 90) of Automation*Steer-type interaction showing statistically significant differences.

Difference of Automation*Steer-type	Diff. of Means	SE of Difference	Simultaneous 95% CI	T-Value	Adjusted P-Value
(Low auto) - (High auto)	-1.32	0.30	(-2.10, -0.54)	-4.35	0.000
(Low manual) - (High auto)	-1.17	0.30	(-1.95, -0.39)	-3.84	0.001
(Low auto) - (High manual)	-1.49	0.30	(-2.27, -0.72)	-4.90	0.000
(Low manual) - (High manual)	-1.33	0.30	(-2.11, -0.55)	-4.39	0.000
Individual confidence level = 98.93%					

Mean values of automation*steer-type*interface-design interaction are shown in Figure 6-19.

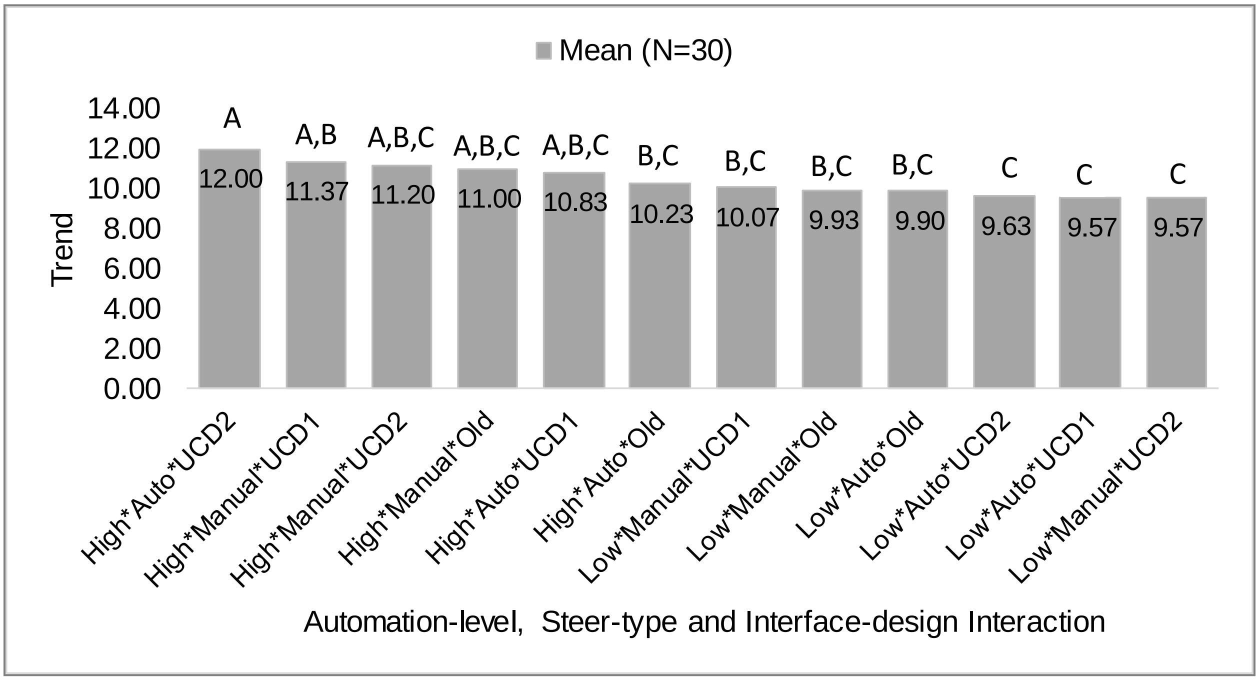


Figure 6-19. Mean values of the Trend identification for automation-level, steer-type, and Interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Results of the detailed Tukey's pairwise comparison showing significant difference are listed in Table 6-10.

Table 6-10. Results of the Tukey's simultaneous tests for differences of means (N = 30) of Automation*Steer-type*Interface-design interaction showing statistically significant differences.

Difference of Automation*Steer-type*Interface-design	Diff. of Means	SE of Diff.	Simultaneous 95% CI	T-Value	Adjusted P-Value
(High*Auto*UCD2) - (High*Auto*Old)	1.77	0.52	(0.05, 3.49)	3.36	0.038
Low*Auto*Old) - (High*Auto*UCD2)	-2.10	0.52	(-3.82, -0.38)	-3.99	0.004
(Low*Auto*UCD1) - (High*Auto*UCD2)	-2.43	0.52	(-4.15, -0.71)	-4.62	0.000
(Low*Auto*UCD2) - (High*Auto*UCD2	-2.37	0.52	(-4.09, -0.65)	-4.50	0.000
(Low*Manual*Old) - (High*Auto*UCD2)	-2.07	0.52	(-3.79, -0.35)	-3.93	0.005
(Low*Manual*UCD1) - (High*Auto*UCD2)	-1.93	0.52	(-3.65, -0.21)	-3.67	0.013
(Low*Manual*UCD2) - (High*Auto*UCD2)	-2.43	0.52	(-4.15, -0.71)	-4.62	0.000
(Low*Auto*UCD1) - (High*Manual*UCD1)	-1.80	0.52	(-3.52, -0.08)	-3.42	0.031
(Low*Auto*UCD2) - (High*Manual*UCD1)	-1.73	0.52	(-3.45, -0.01)	-3.29	0.046
(Low*Manual*UCD2) - (High*Manual*UCD1)	-1.80	0.52	(-3.52, -0.08)	-3.42	0.031
Individual confidence level = 99.88%					

6.5.3 Overall effectiveness

By applying GLM involving mixed procedure in Mini-Tab, overall effectiveness (based on the level 1,2,3 SA and Trend responses altogether) versus automation (fixed factor), steer-type (fixed factor), interface-design (fixed factor) and subjects (random factor-block), we observed significant main effect due to the automation. Observed values are; F (1, 319) = 5.45, P = 0.020 and R^2 = 0.509. Mean values of automation are shown in Figure 6-20.

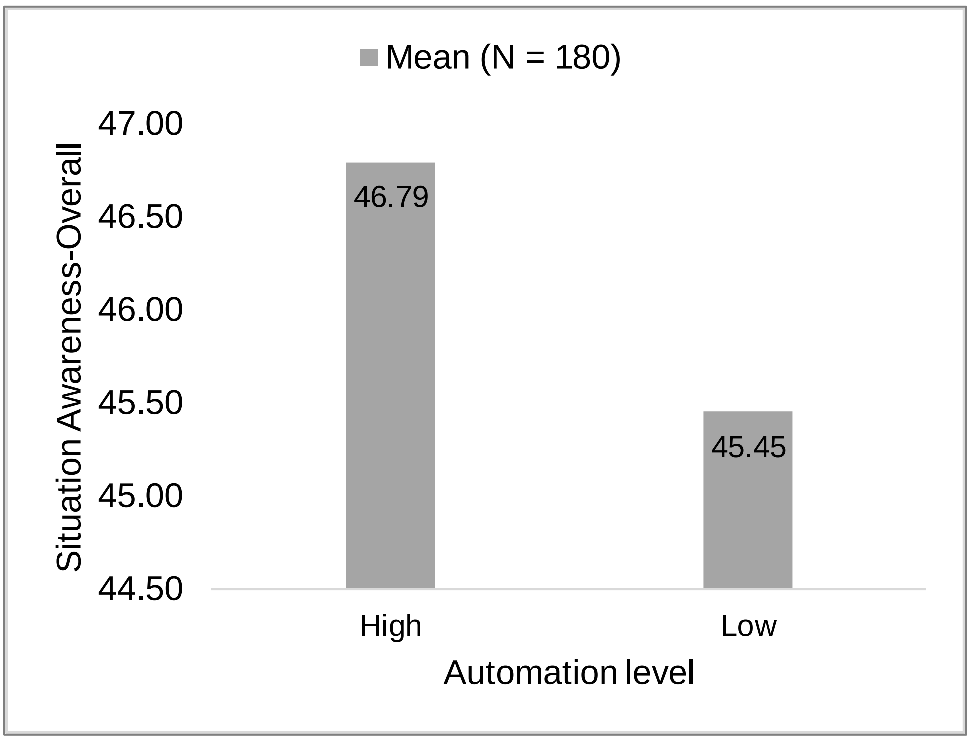


Figure 6-20. Mean values of the overall situation awareness for high and low automation scenario.

The difference between the means of low automation and high automation were observed to be significant (mean difference = -1.34 (SE = 0.576), 95 % CI (-2.48, -0.21), P = 0.020).

We did not observe any other main effect due to steer-type or interface-design based on GLM. Also, no significant interaction effect was observed.

Based on Tukey's pairwise comparison, in some pairs a significant difference due to the interaction (automation*steer-type) have been observed. Means are mentioned in Figure 6-21.

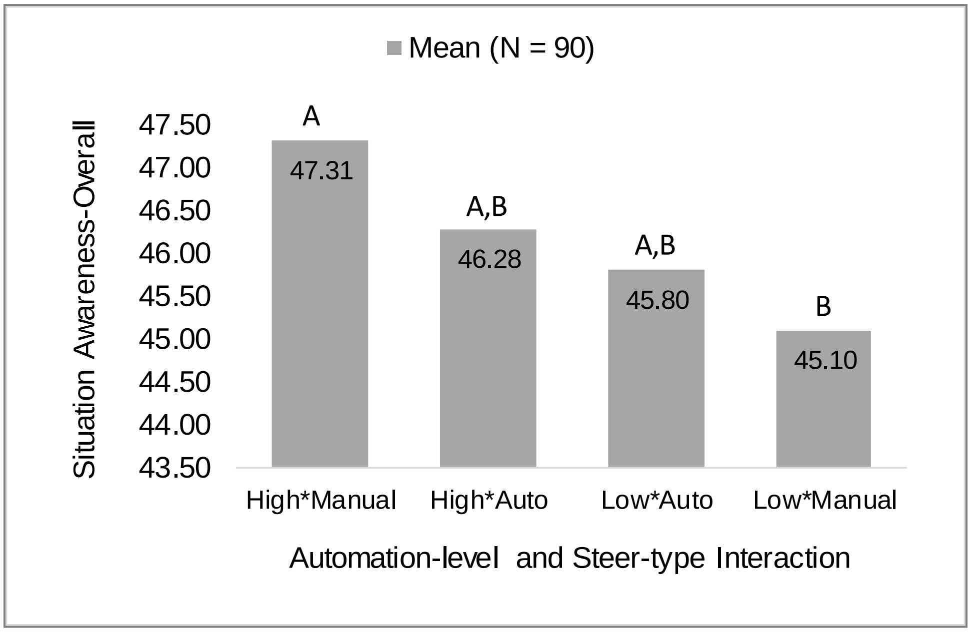


Figure 6-21. Mean values of the overall situation awareness for automation-level and steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Results of the detailed Tukey's pairwise comparison showing significant difference are listed in Table 6-11.

Table 6-11. Results of the Tukey's simultaneous tests for differences of means (N = 90) of Automation*Steer-type interaction showing statistically significant differences.

Difference of Automation*Steer-type	Difference of Means	SE of Difference	Simultaneous 95% CI	T-value	Adjusted P-Value
(Low*Manual) - (High*Manual)	-2.21	0.81	(-4.31, -0.12)	-2.72	0.033
Individual confidence level = 98.93%					

6.5.4 Mental workload

Based on GLM involving mixed procedure in Mini-Tab; MWL versus automation (fixed factor), steer-type (fixed factor), interface-design (fixed factor) and subjects (random factor-block), we observed significant main effect due to the automation and interface -design. For automation, observed values are: F (1, 319) = 11.93, P = 0.001 and R^2 = 0.609. Mean values of automation are shown in Figure 6-22.

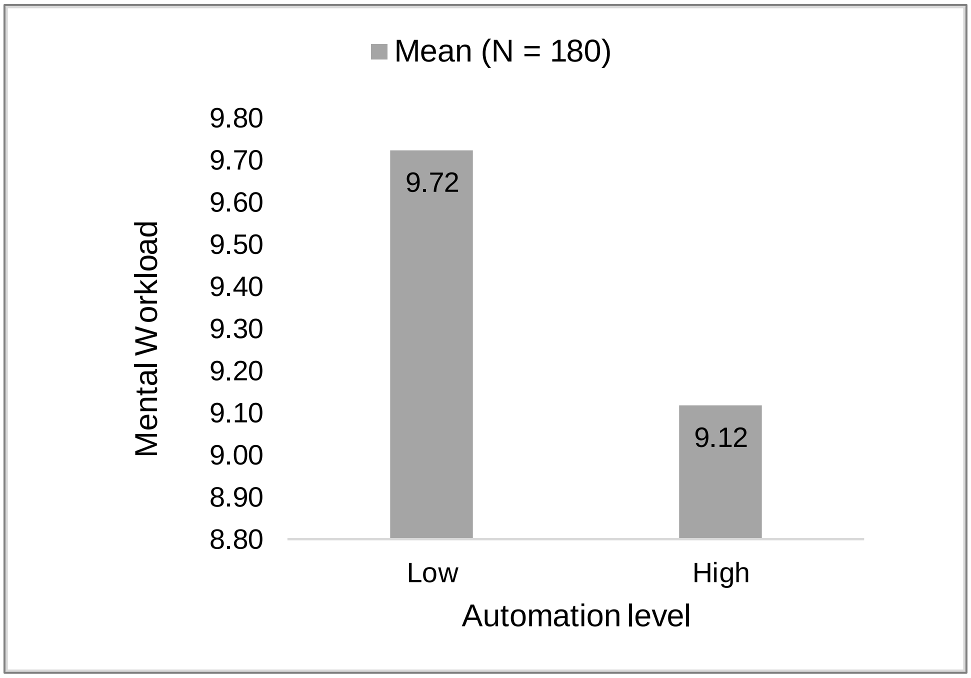


Figure 6-22. Mean values of the mental workload for low and high automation scenario.

Statistical difference between the means is shown in Table 6-12.

Table 6-12. The differences of means (N = 180) for MWL in low and high automation scenario.

Difference of MWL in Automation Levels	Difference of Means	SE of Difference	Simultaneous 95% CI	T-value	Adjusted P-Value
Low - High	0.61	0.18	(0.26, 0.95)	3.45	0.001
Individual confidence level = 95.00%					

For interface-design, observed values of significant main effect are: F (2, 319) = 11.99, P = 0.000 and R^2 = 0.609. Mean values of interface-design are shown in Figure 6-23.

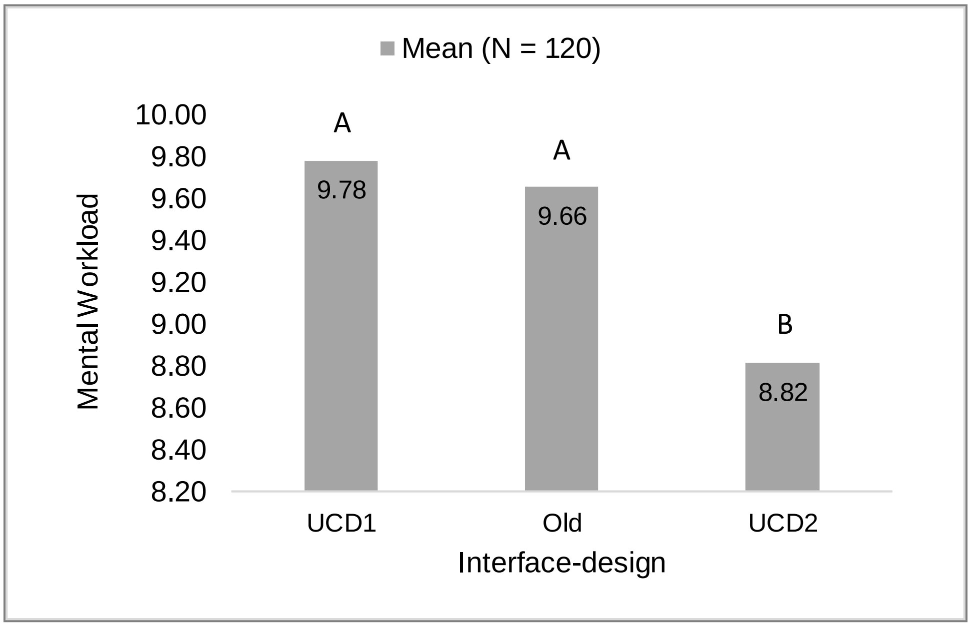


Figure 6-23. Mean values of the mental workload for three types of Interface designs viz., UCD1, Old and UCD2. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Results of the detailed Tukey's pairwise comparison showing significant difference are listed in Table 6-13.

Table 6-13. Results of the Tukey's simultaneous tests for differences of means (N = 120) for MWL in Interface-design levels showing statistically significant differences.

Difference of MWL in Interface-design Levels	Difference of Means	SE of Difference	Simultaneous 95% CI	T-value	Adjusted P-Value
UCD2 - Old	-0.84	0.22	(-1.34, -0.34)	-3.92	0.000
UCD2 - UCD1	-0.97	0.22	(-1.47, -0.46)	-4.50	0.000
Individual confidence level = 98.01%					

We did not observe any overall significant interaction effect for MWL in the GLM; however, based on the Tukey's pairwise comparison, we observed a significant difference in some pairs due to the interaction of automation and steer-type (Figure 6-24); automation and interface-design (Figure 6-25); steer-type and interface-design (Figure 6-26), and automation, steer-type and interface-design (Figure 6-27).

Mean values for all these significant interaction effects as well as their detailed Tukey's pairwise comparison are mentioned in the following paragraphs.

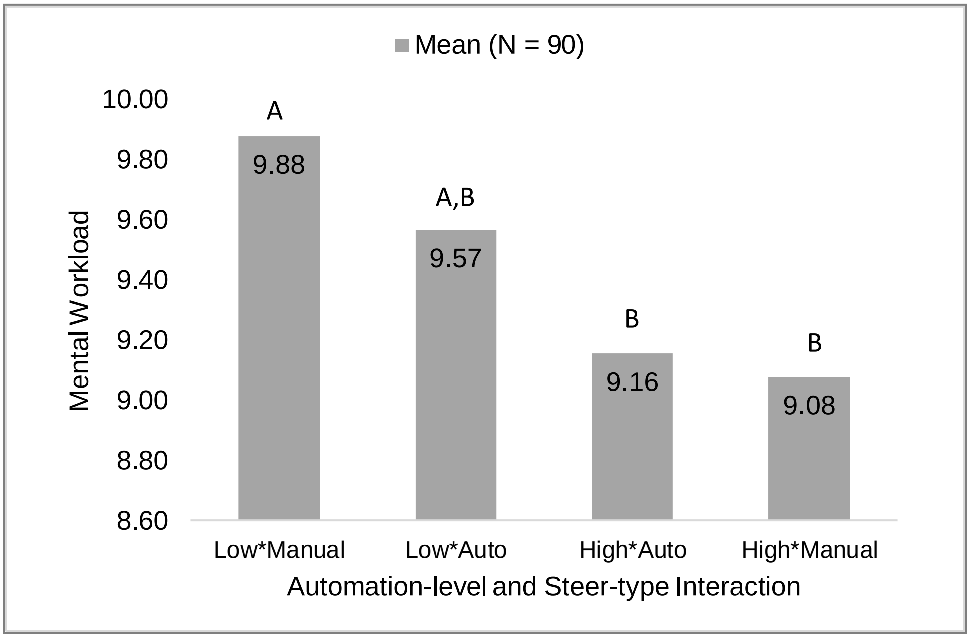


Figure 6-24. Mean values of the mental workload for automation-level and steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Results of the detailed Tukey's pairwise comparison showing significant difference are listed in Table 6-14.

Table 6-14. Results of the Tukey's simultaneous tests for differences of means (N = 90) for MWL in Automation*Steer-type levels showing statistically significant differences.

Difference of MWL in Automation*Steer-type Levels	Difference of Means	SE of Difference	Simultaneous 95% CI	T-value	Adjusted P-Value
(Low manual) - (High auto)	0.72	0.25	(0.25, 1.36)	2.91	0.019
(Low manual) - (High manual)	0.80	0.25	(0.16, 1.44)	3.23	0.007
Individual confidence level = 98.93%					

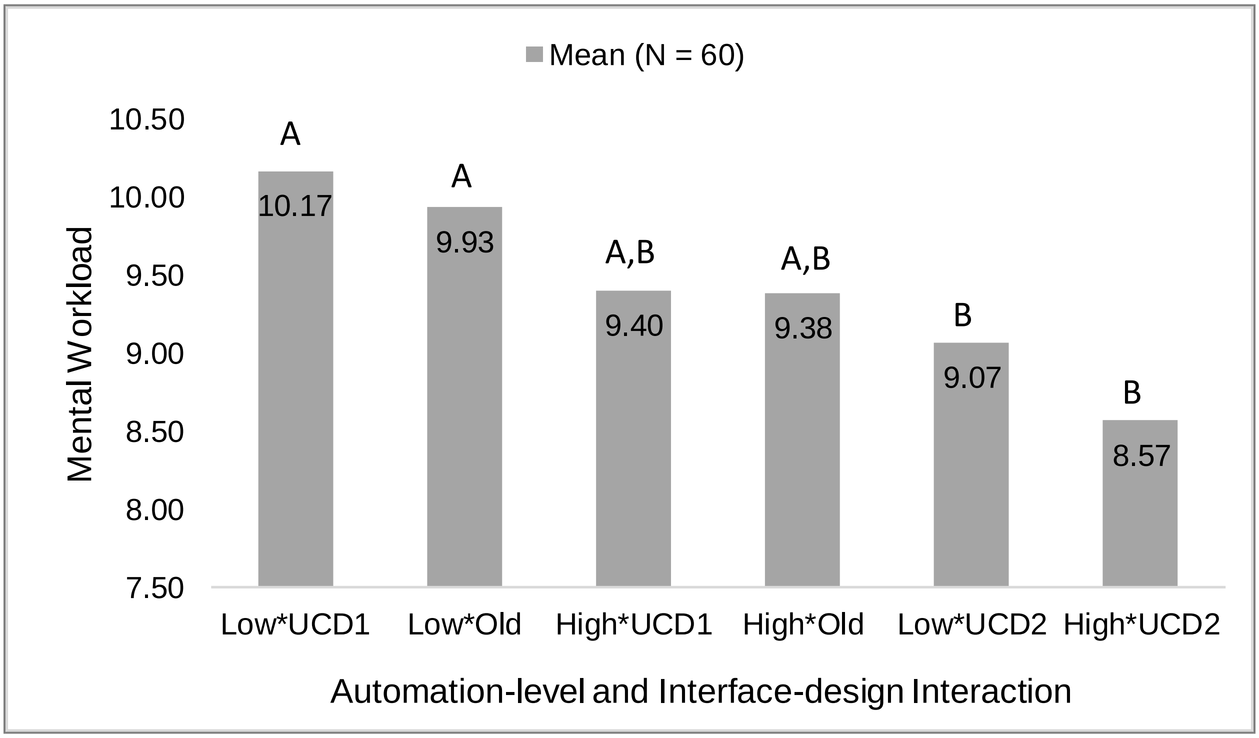


Figure 6-25. Mean values of the mental workload for automation-level and steer-type interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Results of the detailed Tukey's pairwise comparison showing significant difference are listed in Table 6-15.

Table 6-15. Results of the Tukey's simultaneous tests for differences of means (N = 60) for MWL in Automation*Interface-design levels showing statistically significant differences.

Difference of MWL in Automation*Interface-design Levels	Difference of Means	SE of Difference	Simultaneous 95% CI	T-value	Adjusted P-Value
(Low*Old) - (High*UCD2)	1.37	0.30	(0.50, 2.23)	4.50	0.000
(Low*UCD1) - (High*UCD2)	1.60	0.30	(0.74, 2.47)	5.27	0.000
(Low*UCD2) - (Low*Old)	-0.87	0.30	(-1.73, -0.00)	-2.85	0.049
(Low*UCD2) - (Low*UCD1)	-1.10	0.30	(-1.97, -0.24)	-3.62	0.004
Individual confidence level = 99.53%					

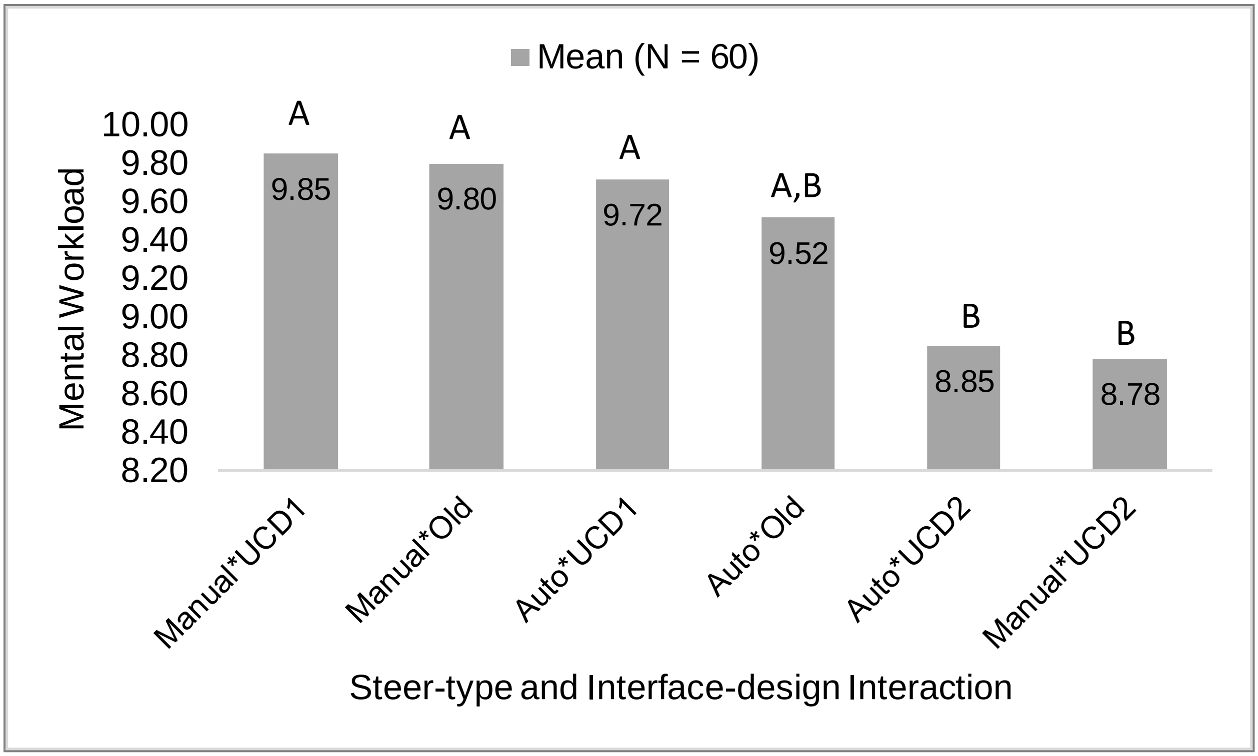


Figure 6-26. Mean values of the mental workload for steer-type and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Results of the detailed Tukey's pairwise comparison showing significant difference are listed in Table 6-16.

Table 6-16. Results of the Tukey's simultaneous tests for differences of means (N = 60) for MWL in Automation*Interface-design levels showing statistically significant differences.

Difference of MWL in Steer-type*Interface-design Levels	Difference of Means	SE of Difference	Simultaneous 95% CI	T-Value	Adjusted P-Value
(Auto*UCD2) - (Auto*UCD1)	-0.87	0.30	(-1.73, -0.00)	-2.85	0.049
(Manual*UCD2) - (Auto*UCD1)	-0.93	0.30	(-1.80, -0.07)	-3.07	0.026
(Manual*Old) - (Auto*UCD2)	0.95	0.30	(0.09, 1.81)	3.13	0.022
(Manual*UCD1) - (Auto*UCD2)	1.00	0.30	(0.14, 1.87)	3.29	0.013
(Manual*UCD2) - (Manual Old)	-1.02	0.30	(-1.88, -0.15)	-3.35	0.011
(Manual*UCD2) - (Manual*UCD1)	-1.07	0.30	(-1.93, -0.20)	-3.51	0.006
Individual confidence level = 99.53%					

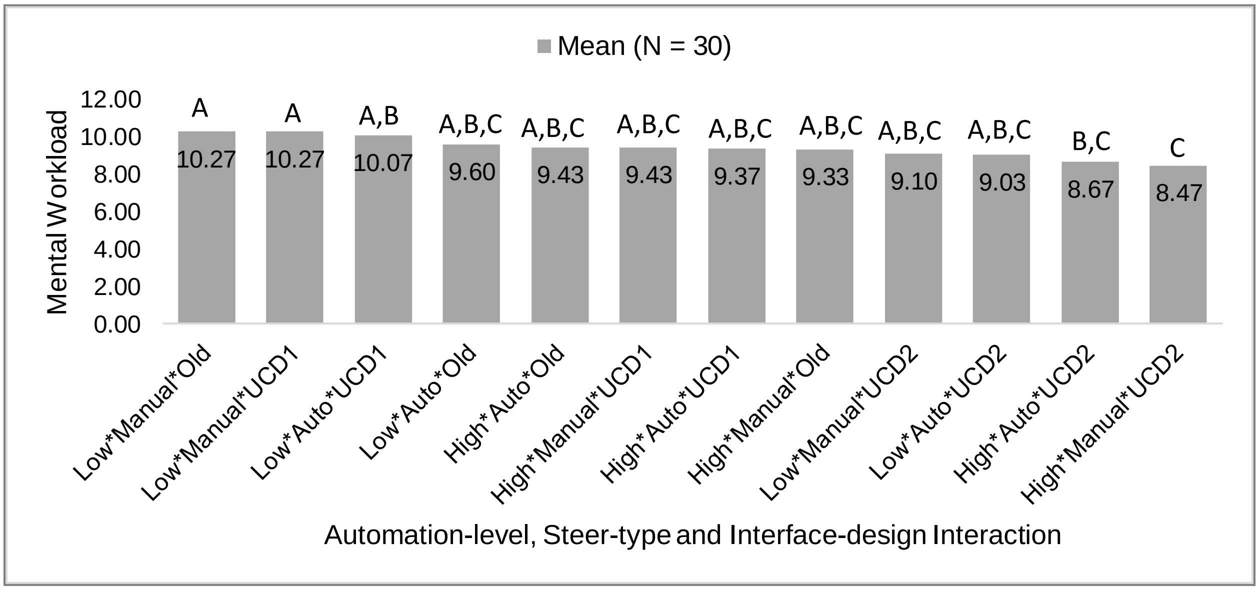


Figure 6-27. Mean values of the mental workload for automation-level, steer-type and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Results of the detailed Tukey's pairwise comparison showing significant difference are listed in Table 6-17.

Table 6-17. Results of the Tukey's simultaneous tests for differences of means (N = 30) for MWL in Automation*Interface-design*Interface-design levels showing statistically significant differences.

Difference of MWL in Automation*Steer-type*Interface-design Levels	Diff. of Means	SE of Diff.	Simultaneous 95% CI	T-value	Adjusted P-Value
(Low auto UCD1) - (High auto UCD2)	1.40	0.43	(-0.00, 2.80)	3.26	0.051
(Low manual Old) - (High auto UCD2)	1.60	0.43	(0.20, 3.00)	3.73	0.011
(Low manual UCD1) - (High auto UCD2)	1.60	0.43	(0.20, 3.00)	3.73	0.011
(Low auto UCD1) - (High manual UCD2)	1.60	0.43	(0.20, 3.00)	3.73	0.011
(Low manual Old) - (High manual UCD2)	1.80	0.43	(0.40, 3.20)	4.19	0.002
(Low manual UCD1) - (High manual UCD2)	1.80	0.43	(0.40, 3.20)	4.19	0.002
Individual confidence level = 99.88%					

We obtained many significant results in MWL of the users. Users demonstrated low MWL in the High automation scenario in comparison to the low automation scenario (Figure 6-22). We believe that support of automation (in the high automation scenario) helped in reducing the demand on the cognitive faculties of the users, which caused users to feel less MWL during the high automation scenario in comparison to the lower automation scenario.

Among three types of UIs, users demonstrated significantly lower MWL in the UCD2 interface in comparison to the UCD1 and Baseline (Old) interface (Figure 6-23). We believe that the effectiveness of the UCD2 interface for supporting all three levels of SA (i.e., perception, comprehension, and projection) without excessively stressing the cognitive resources in comparison to the Baseline and UCD1 interface, is the main reason for the reduced MWL. Similarly, based on the other combinations of UDC2 interface with manual/auto steering or low/high automation levels, the UCD2 interface caused significantly lower MWL to the users than the Baseline (Old) and UCD 1 interface (Figure 6-23, Figure 6-24,Figure 6-25 Figure 6-26, and Figure 6-27).

6.5.5 Response time analysis

During low automation scenario (when users were responsible for correcting the situation by clicking on the erroneous element on the interface), the reaction times of the users were also measured and evaluated. This reaction time is the amount of time between the appearance of the error and correction of the error by the user.

Based on GLM involving mixed procedure in Mini-Tab, reaction time (as response variable) versus steer-type (fixed factor), interface-design (fixed factor) and subjects (random factor-block) was evaluated. We observed significant main effect due to the interface-design. We obtained $F(2, 145) = 29.34$, $P = 0.000$, and $R^2 = 0.611$ for interface-design. No significant main effect was observed for steer-type. Mean values of reaction-time for three types of interfaces are provided in Figure 6-28.

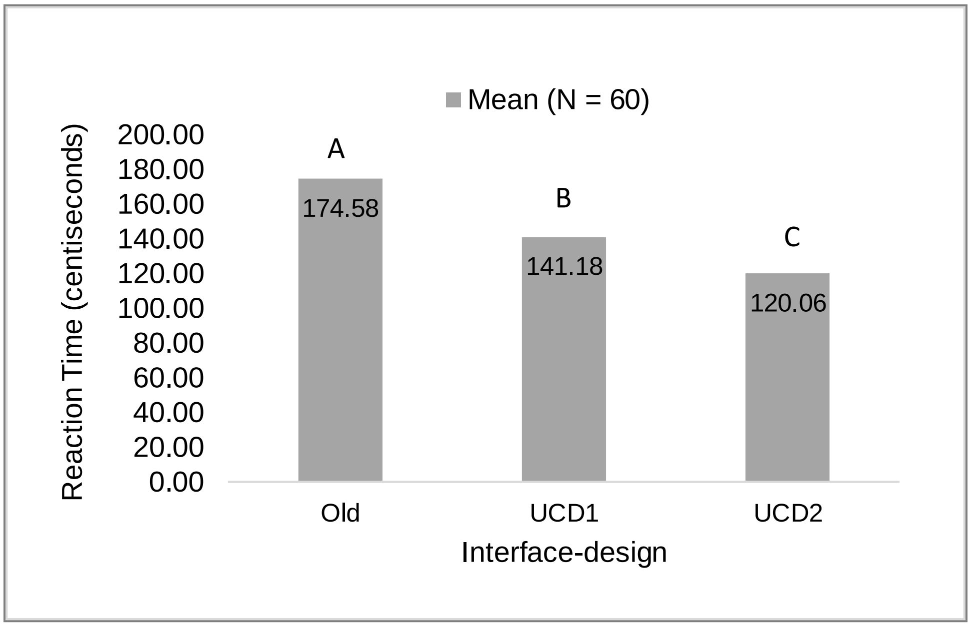


Figure 6-28. Mean values of the reaction time (centiseconds) for three types of interface-designs (Old, UCD1 and UCD2). Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Based on Tukey's pairwise test, a difference of the reaction-time means (of the interface-design), along with the standard error and 95% CI are mentioned in Table 6-18.

Table 6-18. Results of the Tukey's simultaneous tests for differences of means (N = 60) for MWL in Automation*Interface-design*Interface-design levels showing statistically significant differences.

Difference of reaction-time in Interface-design Levels	Difference of Means	SE of Difference	Simultaneous 95% CI	T-value	Adjusted P-Value
UCD1 - Old	-33.40	7.18	(-50.40, -16.40)	-4.65	0.000
UCD2 - Old	-54.52	7.18	(-71.53, -37.52)	-7.60	0.000
UCD2 - UCD1	-21.12	7.18	(-38.13, -4.12)	-2.94	0.011
Individual confidence level = 98.08%					

We did not observe any significant interaction effect of interface-design and steer-type in the GLM; however, based on the Tukey's pairwise comparison, we observed a significant difference in some pairs due to the interface-design and steer-Type (Figure 6-29).

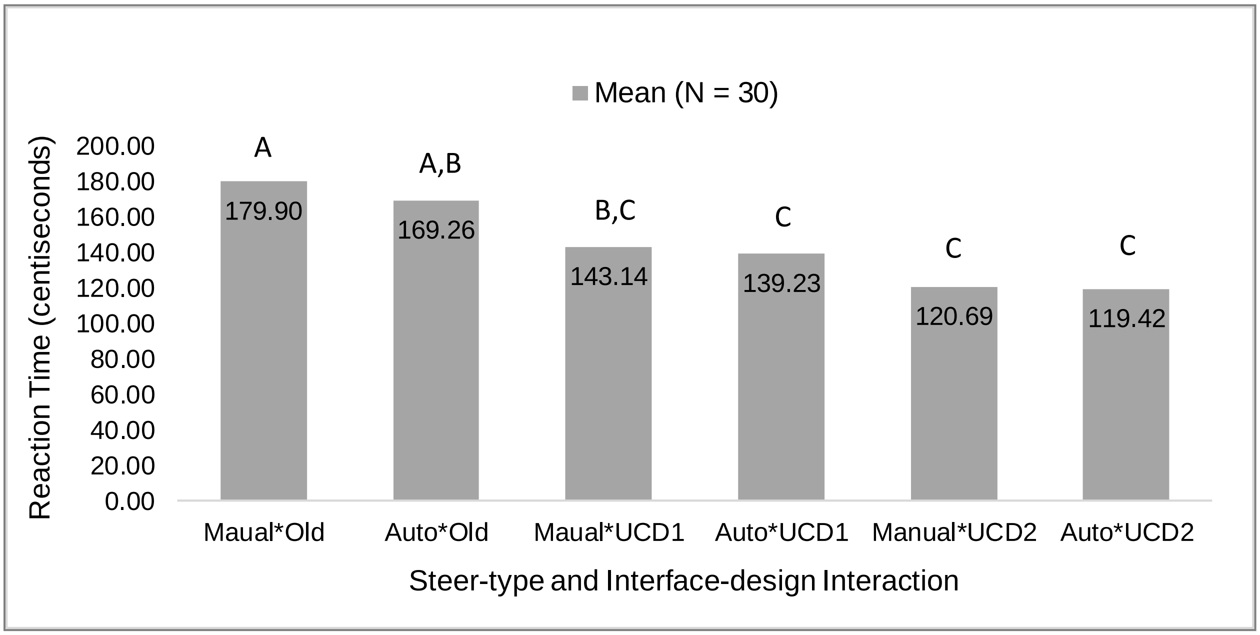


Figure 6-29. Mean values of the reaction time (centiseconds) for steer-type and interface-design interaction. Letters (A, B…) indicate grouping information based on Tukey's pairwise comparison. Values that do not share a letter are significantly different.

Details of the statistically different pairs based on the Tukey's pairwise comparison are mentioned in Table 6-19.

Table 6-19. Results of the Tukey's simultaneous tests for differences of means (N = 60) for MWL in Automation*Interface-design*Interface-design levels showing statistically significant differences.

Difference of response-time in Steer-type*Interface-design Levels	Difference of Means	SE of Difference	Simultaneous 95% CI	T-Value	Adjusted P-Value
(Auto*UCD1) - (Auto Old)	-30.0	10.2	(-59.3, -0.7)	-2.96	0.041
(Auto*UCD2) - (Auto Old)	-49.8	10.2	(-79.1, -20.6)	-4.91	0.000
(Manual*UCD2) - (Auto Old)	-48.6	10.2	(-77.9, -19.3)	-4.78	0.000
(Manual*Old) - (Auto*UCD2)	60.5	10.2	(31.2, 89.8)	5.96	0.000
(Manual*UCD1) - (Manual*Old)	-36.8	10.2	(-66.1, -7.5)	-3.62	0.005
(Manual*UCD2) - (Manual*Old)	-59.2	10.2	(-88.5, -29.9)	-5.83	0.000
Individual confidence level = 99.55%					

We observed significantly different response times for three interface-designs (Figure 6-28). Mean response time for UCD2 was 120.06 cs in comparison to 174.58 cs for the baseline interfaces, which was approximately 31% lower than the baseline interface. The response time for the UCD1 interface was 19% lower than the baseline interface (141.18 vs. 174.58 cs). These results indicate that the UCD based interfaces are much more effective in helping users to respond to the situation. This quicker response time of the UCD interfaces in comparison to the baseline may be due to the better perception of the situation (level 1 SA) which is further augmented by the better level 2 and level 3 SA support of the UCD interface in comparison to the baseline interface.

In the case of interaction between steer-type and interface-design (Figure 6-29), we observed that auto steering improved the response time in every interface-design type. However, the difference in response time within each interface-design is not statistically significant. We believe that a reduction in the response time in the auto steering scenario is due to a reduction in task load of the user. In the auto steering scenario, the task of controlling the steering was automatically performed by the system, so the user had comparably more free cognitive faculties to handle air seeder related tasks. This reduction in the task load and escalation in free user-faculties contributed to a reduction in the response time. This trend in the reduction of the response time was consistently observed in all three types of interface designs.

6.5.6 Subjective data feedback

At the end of each session (high-automation or low-automation), subjective feedback was collected. During the subjective feedback, users were asked to rate the three interfaces as best, average and worst based on their experience during the experiment. Users were required to evaluate three interface designs by 5 questions related to the perception of the information, recall, comprehension, trend, and prediction of the future state (Appendix A.7).

Data were evaluated by performing Ordinal Logistics Regression of Interface-Ranking versus Interface-Design in Minitab (Table 6-20) to understand and quantify the user's responses to the three types of interfaces. For detailed output, please refer to Appendix D.1. Interface-Ranking was a categorical response variable with three outcomes having an order (best, average and worst). Interface-Design was a categorical predictor variable having three levels (A, B, C). Level A represents the Baseline design (Old, Figure 6-1), whereas level B represents new design version 1 (UCD1, Figure 6-3) and C represents new design version 2 (UCD2, Figure 6-4).

Table 6-20. Ordinal Logistics Regression output of Interface-Ranking (best, average, worst, with a total count of 900) as dependent (response) variable versus Interface-Design (A-Old, B-UCD1 & C-UCD2) as independent (predictor) variable.

Predictor variable – Interface-Design (3 levels- A, B, C)	**Coefficient**	**SE of Coefficient**	**Z- value**	**P - Value**	**Odds Ratio**	**95% CI**	
						Lower	Upper
B (UCD1)	1.043	0.159	6.57	0.000	2.84	2.08	3.87
C (UCD2)	2.559	0.175	14.61	0.000	12.92	9.17	18.21

*[1]Results of the test that all slopes are zero: G = 244.062, DF = 2, P-Value = 0.000

Goodness-of-Fit Tests

Method	**Chi-Square**	**DF**	**P - Value**
Pearson	2.069	2	0.355
Deviance	2.050	2	0.359

Based on Logistics regression (Table 6-20), P-Value less than 0.05, from the results (*[1]) of the test that all slopes are zero, indicates that the relationship between predictor and response variables are significant (G = 244.062, P = 0.000). From the Goodness-of-Fit Tests, we have observed p-values as 0.355 and 0.359 based on the Pearson and

Deviance method. This high p-value does not provide evidence that the model is inadequate.

Furthermore, we observed that both Design B and C are significantly different from Design A. Referring to Table 6-20, for design B in comparison to A, we found P = 0.000, Odds ratio = 2.84 and Coefficient = 1.043. Positive Coefficient and greater than 1 odd ratio indicate that B has been rated significantly higher than A. Odds of being rated higher for B are 2.84 than that of A. Similar, odds for being ranked high for C are 12.92 times than A, which is a high value. The high odd ratio of C indicates that subjects have placed very high confidence in the UCD2 interface. We believe that this high confidence is due to the further improvements made in the UCD2 interface design over the baseline and UCD1 designs. Nevertheless, users have provided a few descriptive suggestions to further improve the designs which are listed in Appendix B.

6.6 SUMMARY AND CONCLUSION

For level 1 type of SA, we observed that users showed a higher level of SA when the intermediate level of automation was applied (i.e., when users were interacting with the systems either by correcting the steering or any other air seeder related elements) (Figure 6-7). In the fully autonomous scenario (where users were not responsible for remedying but only observing the situation) or fully manual conditions (where users were responsible for correcting everything whether it is steering or any other air seeder elements), SA of the users was lower than for the intermediate levels of automation.

For level 2 type of SA, users demonstrated a higher level of SA in low automation (Figure 6-8). Further, when lower automation was combined with the autonomous steering, users showed the highest level of SA (Figure 6-9). As far as automation and

Interface-design interaction is concerned, the UCD2 design performed best followed by Old and UCD1 (Figure 6-10) in a low level of automation scenario. However, the combination is not statistically significant. For automation, steer-type and interface-design interaction in the low automation scenario, UCD2 performed better than Old or UCD1. However, the difference between the interfaces is not significant (Figure 6-11).

For level 3 SA, we observed higher SA in a high level of automation (Figure 6-12). Furthermore, other combinations of manual/auto steering and interface-design (UCD2/UCD1/Old) in high automation mode produced better results than lower automation levels (Figure 6-13, Figure 6-14, and Figure 6-15). Moreover, in most of the scenarios, the UCD2 in various combinations (high/low automation and manual/auto steering) performed better than UCD1 and Baseline (Old) interface designs (Figure 6-15). However, the difference between the three interfaces when compared for similar conditions is not statistically significant (Figure 6-15). Also, during the Trend identification (Figure 6-16, Figure 6-17, Figure 6-18, and Figure 6-19), similar results (as of level 3 SA) were observed.

Regarding overall effectiveness (based on the level 1, 2, 3 SA, and Trend responses altogether), we observed significant results while comparing high and low automation levels. Users demonstrated a higher level of SA in the high automation scenario (Figure 6-20) in comparison to the low automation scenario. Furthermore, the highest level of SA was observed in the combination of high automation with manual steering, which is an intermediate level of automation demanding users to be involved and interactive with the systems (Figure 6-21).

Regarding MWL, users have reported lower MWL in the high automation scenario in comparison to the low automation scenario (Figure 6-22). We believe that the availability of the better automation assistance during the high-automation scenario in comparison to the low-automation scenario caused less MWL in the high-automation scenario.

Considering the outcomes of the response time for three types of interface-designs; the UCD2 interface demonstrated the least response time (120.06 cs) followed by UCD1 (141.18 cs) and baseline (174.58 cs) (Figure 6-28). We believe that better response time of the users for UCD based designs is due to the better support for all three levels of SA, particularly for level 1 type SA. Considering the interaction effect of steer-type with interface-design (Figure 6-29), we observed that each interface-design performed better when steering was autonomous. We believe that better response time in the auto steering scenario is due to the reduction in the task load in the auto steering scenario, as the system automatically controlled the steering. The system relieved the user of one additional task of steering control.

Based on the subjective feedback regarding 3 types of interface design (i.e., Baseline (Old), UCD1 and UCD2), users rated the UCD2 interface highest followed by the UCD1 and the Baseline Interface. Odds ratios of UCD2 and UCD1 were 12.92 and 2.84 in comparison to the baseline interface (Table 6-20).

Regarding interface design, users have demonstrated the lowest MWL with the UCD2 in comparison to the UCD1 and the Baseline Interface (Figure 6-23). We believe that due to the superior capabilities of the UCD2 interface towards achieving and maintaining better SA without causing too much demand on the cognitive faculties of the users is the primary cause for the reduced MWL for UCD2. Based on the other combinations of the UDC2 interface with manual/auto steering or low/high automation levels, the UCD2 interface performed significantly better than the UCD1 and the Baseline (Old) interface (Figure 6-25, and Figure 6-26).

The common denominator, on the basis of the analysis of SA, response time, MWL and subjective feedback is that the intermediate level of automation is better than either the fully autonomous or manual scenario, and the UCD2 interface-design is more competent than the UCD1 and the baseline design under most of the conditions.

During the fully autonomous scenario, users were not responsible for making any correction, whether the errors were related to the guidance or the air seeder elements. Whereas, during the fully manual conditions (low automation and manual steering), users were responsible for correcting all errors related to air seeder elements and guidance bar. Intuitively, it seems that during the fully autonomous scenario, due to the greater availability of cognitive resources, users should possess better SA and less MWL. However, results have indicated a different outcome. Instead, intermediate levels of automation produced better levels of SA and MWL. We believe that better SA was due to keeping users in-the-loop; being inside the loop was felt less demanding and more comfortable to the users, as indicated by the MWL evaluation.

Regarding three types of interface-designs, the UCD2 design performed better than the UCD1 and the Baseline (Old) design for SA, response time, MWL and subjective feedback under most of the conditions. Regarding SA, though the UCD2 interface design in combination of low/high automation and manual/auto steering performed better than other two designs, however, in most of the cases while comparing similar conditions, the difference among the three types of interfaces was not statistically significant. However, for response time, the USD2 interface performed significantly better than the USD1 and the baseline interface. Similarly, for MWL (Figure 6-23) and subjective feedback (Table 6-20), the UCD2 was significantly better than the UCD1 and the Baseline (Old) interface. We believe that the primary reason for the better performance of the UCD2 is its effectiveness in providing higher SA without causing an excess demand on cognitive faculties of the users.

CHAPTER 7

CONCLUSIONS

7.1 SPECIFIC CONCLUSIONS

7.1.1 Phase 1 of the Study

Based on the individual comparison of the air seeder elements in phase 1 of the study, I observed that UCD oriented air seeder elements are more effective regarding the SA and MWL of the operator than the baseline elements. Evaluation of the results showed significant improvement in SA with the UCD elements (up to 11%; overall mean difference = 5.0 (4.8%), 95% CI (6.47, 3.60), $P < 0.0001$). For MWL, reductions of up to 19.7% (overall mean difference = -5.2 (-7.9%), $n = 30$, $\alpha = 0.05$) were observed with the UCD elements. During the subjective feedback, the study participants rated the overall performance of the newly designed UCD elements higher in comparison to the previous baseline elements (overall mean difference = 27.3 (189.8%), 99% CI (35.15, 19.38), $P < 0.0001$). This higher ratings of the users about UCD oriented elements further indicate the higher perceived trust in the UCD oriented air seeder elements.

7.1.2 Phase 2 of the study

7.1.2.1 Interface Design Effect

Situation Awareness Comparisons of three interface designs (UCD2, UCD1, and Baseline) under different levels of automation during the second phase of the study yielded many significant results. For level 1, level 2 and level 3 SA, or for overall effectiveness (based on all levels of SA and trend factor altogether), I did not observe any significant main effect due to the interface-design. However, for level 2 SA (Figure

6-10), some pairs due to automation*interface-design interaction yielded significant results. The combination of low-automation*UCD2 is significantly better than the high-automation*UCD1 (difference = 1.25, 95% CI (0.67, 1.81), P = 0.000), or high-automation*Baseline (difference = 0.90, 95% CI (0.34, 1.46, P = 0.000). Similarly (Figure 6-14), for level 3 SA, high-automation*UCD2 interface performed significantly better than the low-automation*Baseline (difference = 1.20, 95 % CI (0.18, 2.22), P = 0.010), or low-automation*UCD1 (difference = 1.22, 99 % CI (0.20, 2.24), P = 0.009) (Table 6-6).

Mental Workload Evaluations of the MWL indicated significant main effect due to interface-design. The UCD2 interface performed significantly better than the UCD1 (difference = 0.97, 95 % CI (0.46, 1.47), P = 0.000) or the Baseline interface (difference = 0.84, 95 % CI (0.34, 1.34), P = 0.000) (Figure 6-23, Table 6-13).

Response Time Analysis of the response time of the users yielded significant main effect due to the interface-design. The UCD2 interface performed best regarding the response time (120.06 cs), followed by the UCD1 (141.18 cs) and the Baseline interface (174.58 cs) (Figure 6-28). Difference of the means between UCD2 and UCD1 is -21.12, with 95% CI as -38.13 to -4.12, and p-value = 0.011. Whereas the difference of means between UCD2 and Baseline is -54.52, with 95% CI as -71.53 to -37.52, and p-value = 0.000 (Table 6-18).

Subjective Feedback Based on the evaluation of the subjective feedback of users while rating three interface-designs (as best, average or worst), I observed that UCD oriented interfaces were rated higher than the baseline interface. Odds ratio for UCD2 was 12.92 (95% CI (9.17, 18.21), P = 0.000), and for UCD1 was 2.84 (95% CI (2.08, 3.87), P = 0.000), in comparison to the baseline interface (Table 6-20).

7.1.2.2 Automation Effect

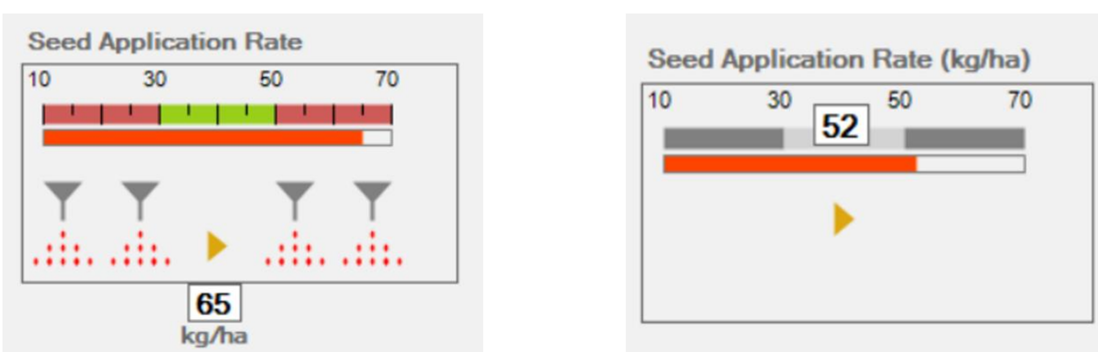
While testing the three interface-designs under two levels of automation (low automation, high automation) and two levels of steering-type (auto steering and manual steering), I observed several significant effects due to automation and steering-type on the SA and MWL of the operators. On level 1 SA, automation or steering-type did not prove any main effect. However, automation*steering-type interactions showed quite interesting results. Intuitively, it was expected that the fully autonomous scenario (when automation is high, and steering is also autonomous) would yield better SA. However, I observed least SA (level 1) in the fully autonomous scenario. Better SA was observed when either automation level was high and steering was manual; or automation was low and steering was autonomous. In other words, extreme cases of automation (fully manual or fully autonomous) were not found best for SA. Instead, intermediate levels (HighAutomation*ManualSteering or LowAutomation*AutoSteering) of the automation design were found to be better in level 1 SA. Similar results were obtained for level 2, level 3, Trend and Overall Effectiveness. For overall effectiveness (overall SA), I observed significant main effect due to the automation. In the high level of automation, better SA was observed (mean difference = 1.34, 95% CI (0.21, 2.48), P = 0.020) (Figure 6-20). I believe better SA during the intermediate levels of automation is due to keeping the user inside-the-loop. Detailed discussion and results of various automation and steering-type interaction combinations can be found in chapter 6 (section 6.5).

Regarding the effect of automation on MWL, I observed significant main effect of automation on MWL ($F(1, 319) = 11.93$, $P = 0.001$ and $R^2 = 0.609$). Higher levels of automation helped in the reduction of MWL (mean difference = 0.61, 95% CI (0.26, 0.95), $P = 0.001$) (Figure 6-22, Table 6-12). Similarly, interaction of highAutomation*manualSteering yielded lowest MWL, and is significantly different from lowAutomation*manualSteering (mean difference = 0.80, 95% CI (0.16, 1.44), $P = 0.007$) (Figure 6-24,Table 6-14).

7.2 GENERAL CONCLUSIONS

Based on the outcomes of phase 1 and phase 2 of the study, few general conclusions are also made from the study.

Less is more Based on the evaluations during phase 2 of the study, UCD2 design performed best; followed by the UCD1 and the baseline interface design. Whether it was SA, MWL, response time or subjective feedback, UCD2 outperformed even the UCD1 interface-design. A broader comparison of UCD2 with UCD1 shows that there is a significantly fewer number of elements in UCD2 than UCD1 interface.



A B

Figure 7-1. Comparison of seed application rate for UCD1 (A) and UCD2 (B) design.

For example, referring to the Seed Application Rate Design of UCD1 (A) with UCD2 (B) (Figure 7-1), there are about 60 elements in the UCD1 design (3 words mentioning seed application rate, 13 lines on bar, 1 progress bar, 4 numbers on scale, 4 funnels and 8x4 = 32 droplets from funnel, 1 indicator arrow, 1 reading number and 1 word mentioning a unit). Although these 60 elements are laid out in approximately seven groups, still the quantity of the total elements in UCD1 is much more than the UDC2 interface (UCD2 has 14 elements only). Considering the complete UCD1 interface which includes other air seeder elements like fan RPM, speed, and tank levels, the total number of UCD1 interface elements become much more than the UCD2 interface elements. I believe that this crowding of the UCD1 interface makes it cognitively demanding and less effective than the UCD2 interface as evident from the evaluation outcomes based on the SA, MWL, and subjective feedback.

Simple is sharp Based on the feedback from phase 1 of the study, the design of some interface elements (e.g., Tool Pressure and Tool Depth) was made simpler (Figure 7-2).

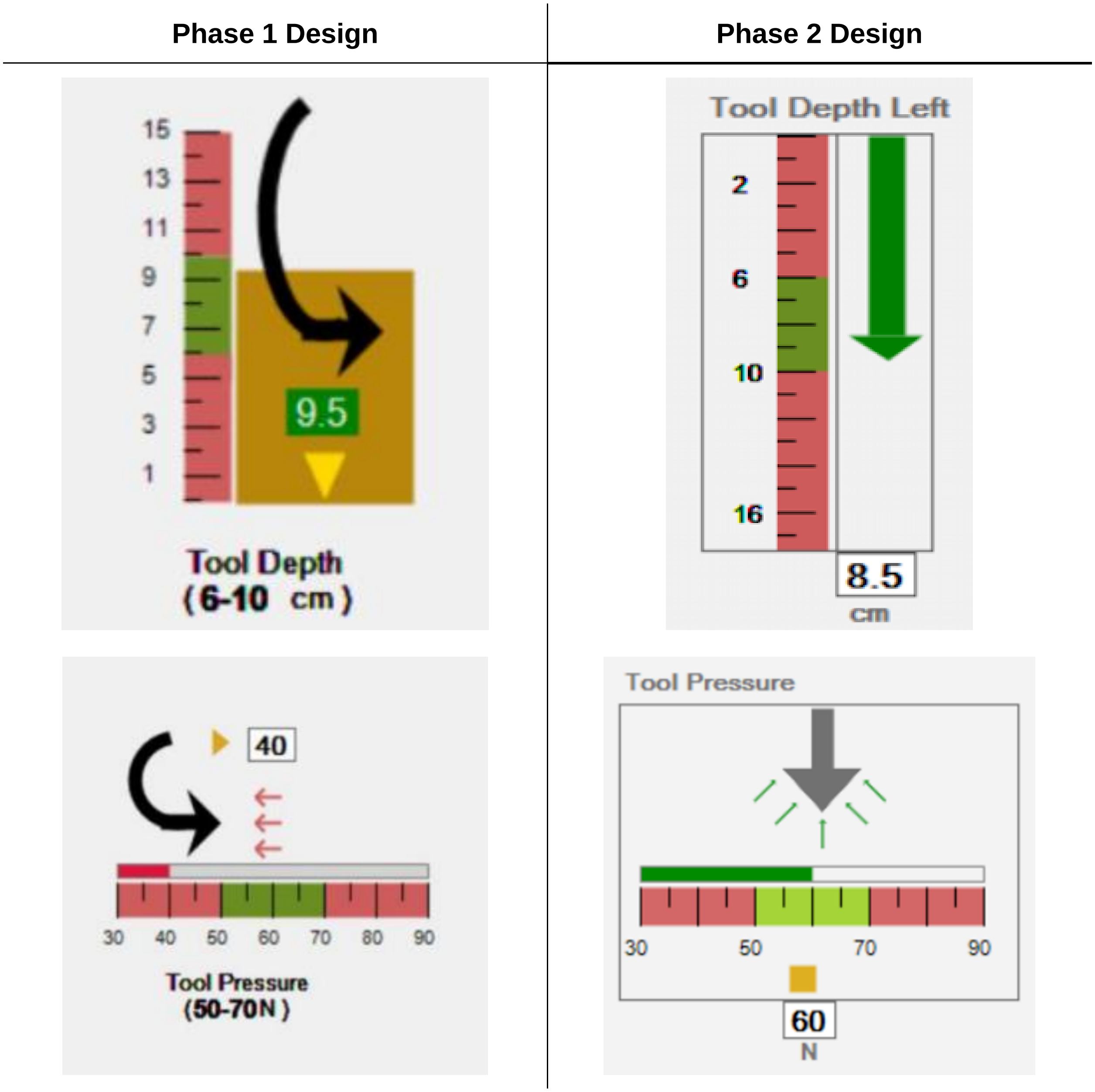


Figure 7-2. Comparison of the design of Phase 1 (original) and Phase 2 (modified) of the UCD1 interface elements.

Both for Tool Pressure and Tool Depth during phase 1, users find difficulty in perceiving and understanding the situation due to the complexity and symbols used in the Tool Pressure and Tool Depth. During evaluations, a substantial amount of increment (24.5%) in the MWL for Tool Pressure was observed. Subsequently, in

UCD1, I simplified the Tool Pressure and Tool Depth element. Furthermore, in UCD2, I even simplified the design of almost all interface elements by minimizing individual symbols, shapes, and colors. Evaluations of phase 2 of the study demonstrated that the simplified UCD2 interface is not only better regarding SA, MWL, and response time, but also highly trusted by the users based on their subjective opinion.

Moderate involvement results in better outcome In addition to the above two general conclusions related to the design of the interfaces, this inference is on the basis of the effect of automation on SA and MWL. Based on the outcomes, I observed that automation helps in improving the SA and reducing the MWL of the users. I observed better overall SA in high (vs. low) automation scenario (46.79 vs. 45.45, Figure 6-20). However, the highest level of overall SA (47.31) was observed when automation was high, but the steering was manual (Figure 6-21). The lowest level of SA (45.10) was observed when the automation was low, and steering was also manual (Figure 6-21). Similarly, in the case of MWL, users reported lower MWL in high automation than low automation (9.12 vs. 9.72) (Figure 6-22). However, least MWL was observed when automation was high and steering was manual, in comparison to the low automation and manual steering condition (9.08 vs. 9.88) (Figure 6-24). Both for SA and MWL, the best results were obtained in high automation and manual steering combination; this scenario demands the user to be interactive with the situation to decide and implement the action. Fully autonomous scenario (high automation and auto steering), or fully manual scenario (low automation and manual steering) performed worse than the intermediate level of automation (high-automation*manual-steering, or low-automation*auto-steering) mostly.

7.3 SIGNIFICANCE AND IMPLICATIONS

Estimated size of the agriculture machinery industry will be USD 200 billion by 2024 (Global Market Insights 2018). Growing application of information technology in the design and advancement of agricultural machines is posing a pressing need on the effective design of the information and data for the operators of agricultural machines. If the human factors and ergonomics principles are ignored during the user-interface design; engineers and designers are quite vulnerable to come-up with the technology-centered user-interfaces, which may lead to the lower situation awareness, higher mental workload, poor performance and safety concerns of the operator.

This study resulted in multiple contributions and implications. Scholarly implications include the delineation of SA requirements for the operator of the air seeder, generated through the Goal-Directed Task Analysis, which covered the air seeder as well as the power unit's (tractor's) dynamic information needs. This knowledge base may further help the research and scientific community to apply a similar approach for other agricultural machines or in other domains of interest. Furthermore, these combined SA requirements (both for air seeder and tractor) will act as the basis for developing an integrated UI which will cover the information needs related to the air seeder and tractor simultaneously.

Study outcomes further extended the existing knowledge by applying the various principles and methodologies, both for UI design and UI evaluation. Based on the literature review, SA principles and other cognitive and perceptual psychology guidelines related to the UI design are jointly described in the context of the design of effective UI. Design principles are carefully itemized and elaborated in relation to application domain for UI design. The research community can use this knowledge base for further studies, and UI designers can use this information as insights and guidelines for designing UIs.

During the evaluation process of the interfaces, multiple evaluation metrics are used (SA, MWL, Subjective Feedback and Reaction Time) which considerably increased the confidence in the results; other researchers can also use a similar strategy of using multiple evaluation criteria.

Outcomes of the evaluation indicate that UCD based design performed better on the multiple evaluation criteria than the baseline; this outcome will help and encourage industry stakeholders to use UCD approach in designing UIs. Furthermore, our second UI design (UCD2) is proved to be effective on multiple measures (SA, MWL, Reaction Time); also, UCD2 design was rated very high by the subjects (odds ratio of 12.9 against the baseline). We believe that this UCD2 design can be adopted by the industry stakeholders for further customization, testing, and subsequent implementation.

7.4 LIMITATIONS AND FUTURE RESEARCH PROSPECTS

This study has been completed in the lab environment using convenient samples mainly from the University of the Manitoba's student population. Using the lab environment helped me in minimizing the impact of external factors, and in faster development and implementing changes. However, the lab environment lacked in the actual agricultural context. Similarly, university students did not have experience with agricultural machinery, and do not represent actual machinery operators. Still, the UCD oriented interfaces yielded better outcomes. A similar study could be completed in an actual field environment using actual machine operators.

After designing, developing and evaluating several prototype interfaces designs, I learned that Interface design is a highly iterative process - no version of the interface could be perfect. Effective interface design requires not only an in-depth analysis of the user's information needs but also a thorough understanding of the perceptual and cognitive psychology principles related to the UI design. This scenario automatically leads to possibilities of several variants of the interface design to address the same information needs of the users; only limited by the imagination of the engineers and the designers. However, due to the involvement of significant skill set and efforts in building functional prototypes to collect research-related data and feedback; I only managed to develop and test 16 individual interface elements (8 UCD and 8 Baseline air seeder elements). Also, few other interactive forms/interfaces were added into the complete application to manage and collect the SA and MWL data during phase 1 and 2 of the experiments. During the second phase of the study, I managed to build three complete prototypes interfaces (two version of UCD and one version of baseline) under four automation conditions (low automation, high automation, manual steering, auto steering).

However, I believe that there are possibilities to build and evaluate other interface design variants for different combinations of the automation. Neither in the baseline or UCD based design, no concept of prewarning has been used. Progress situation of the air seeder elements abruptly enters into the non-desirable region (red zone) from the desirable region (green zone), both in UCD and baseline designs. Although, UCD based designs did a better job regarding the anticipation of the future situation because of the presence of progress indicators (such as bars and needles) along with the scale.

However, I believe that if some prewarning (e.g., yellow lights) could be provided to the user before entering into the red zone, it would be more useful regarding allocating user's attention. Instead of continually scanning all interface elements, it could be more effective for the user to respond quickly, efficiently and cognitively less demanding, because of the prior awareness of the most critical targets on the interface.

Furthermore, I also see the opportunity to develop other versions of the interface by changing visual appearance and interchanging the location of the air seeder elements. For example, appearance of seed application and fertilizer application elements, or speed and fan rpm look similar. Creating distinct designs for similar looking elements may further affect the SA, MWL, response time or subjective opinion of the participants. Similarly, altering the location and the size of various interface elements may also affect the outcome, which can be tested in the future.

Besides, I have tested three types of UIs (Baseline, UCD1 and UCD2) for four variations of automation (high-automation, low-automation, auto-steering, and manual-steering), which broadly falls under the upper three categories of the automation (information analysis, decision and action selection, and action implementation) as described by Parsuraman et al. (2000). Although the information acquisition part was involved in the interface design (which is mainly related to the sensing and registering of the input data), I did not make any significant change in the information acquisition part for the fair comparison of the Baseline interface with the UCD interfaces. I developed and compared the same eight air seeder elements (e.g., seed application rate, fertilizer application rate) in the UCD interface as there were in the Baseline interface. However, anticipating future technological advancements (mainly related to the sensing and input technology), additional interface elements, related to the air seeder and auxiliary power unit, can also be included in future interface designs.

REFERENCES

Atkinson, R., & Shiffrin, R. 1968. Human memory: A proposed system and its control processes. *Psychology of Learning and Motivation*. Retrieved from http://www.sciencedirect.com/science/article/pii/S0079742108604223

Baddeley, A. 1997. *Human memory: Theory and practice*. Retrieved from https://books.google.com/books?hl=en&lr=&id=fMgm-2NXAXYC&oi=fnd&pg=PR5&dq=Human+Memory&ots=jN2c5LkBLo&sig=QJh-tz3BFuaAFXJuJx9ISzIoSsY

Baddeley, A. D. 1972. Selective Attention and Performance in Dangerous Environments. *British Journal of Psychology*, *63*(4): 537–546.

Baddeley, A., & Hitch, G. J. 1984. Working memory. *Psychological Medicine*, *14*(2): 265–271. http://doi.org/10.1017/S0033291700003548

Banbury, S., Selcon, S., Endsley, M., Gorton, T., & Tatlock, K. 1998. Being certain about uncertainty: How the representation of system reliability affects pilot decision making. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 42, pp. 36–39). Retrieved from http://journals.sagepub.com/doi/abs/10.1177/154193129804200109

Bashiri, B., & Mann, D. D. 2014. Automation and the situation awareness of drivers in agricultural semi-autonomous vehicles. *Biosystems Engineering*, *124*, 8–15. http://doi.org/10.1016/j.biosystemseng.2014.06.002

Bashiri, B., & Mann, D. D. 2015. Impact of Automation on Drivers' Performance in Agricultural Semi-Autonomous Vehicles. *Journal of Agricultural Safety and Health*, *21*(2): 129–139. http://doi.org/10.13031/jash.21.10977

Billings, C. E. 1991. Toward a Human-Centered Aircraft Automation Philosophy. *International Journal of Aviation Psychology*, *1*(4): 261. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=7369150&site=ehost-live

Bolstad, C. A., Endsley, M. R., & Technologies, S. A. 2003. Measuring Shared and Team Situational Awareness in the Army's Future Objective Force. In *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting* (Vol. 47, pp. 369–373). http://doi.org/10.1177/154193120304700325

Burns, C., Jamieson, G., Skraaning, G., Lau, N., & Kwok, J. 2007. Supporting Situation Awareness Through Ecological Interface Design. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *51*(4): 205–209. http://doi.org/10.1177/154193120705100413

Bye, A., Hollnagel, E., & Brendeford, T. S. 1999. Human – machine function allocation : a functional modelling approach, *64*(November 1997): 291–300.

Byrne, E. a, & Parasuraman, R. 1996. Psychophysiology and adaptive automation. *Biological Psychology*, *42*(3): 249–68. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/8652747

Christ, R. E. 1975. Review and analysis of color coding research for visual displays. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *17*(6): 542–570. http://doi.org/10.1177/001872087501700602

Cohen, R. A. 2011. Yerkes-Dodson Law (pp. 2737–2738). New York City;NY:Springer.

Cummings, M. L., & Tsonis, C. 2005. Deconstructing Complexity in Air Traffic Control. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *49*(1): 25–29. http://doi.org/10.1177/154193120504900107

Dao, A.-Q. V. Q. V., Brandt, S. L., Battiste, V., Vu, K.-P. P. L., Strybel, T., & Johnson, W. W. 2009. The impact of automation assisted aircraft separation on situation awareness (Vol. 5618 LNCS, pp. 738–747). http://doi.org/10.1007/978-3-642-02559-4_80

DeSanctis, G. 1984. Computer Graphics As Decision Aids: Directions for Research. *Decision Sciences*, *15*(4): 463–487. http://doi.org/10.1111/j.1540-5915.1984.tb01236.x

Di Stasi, L. L., Adoracion, A., & Canas Jose J. 2013. Evaluating mental workload while interacting with computer-generated artificial environments. *Entertainment Computing*, *4*(1): 63–69. http://doi.org/10.1016/j.entcom.2011.03.005

Durso, F. T., Hackworth, C. A., & Truitt, T. R. 1998. Situation Awareness As a Predictor of Performance in En Route Air Traffic Controllers. *Air Traffic Control Quarterly*, *6*(1): 1–20. http://doi.org/10.1518/001872007X230235

Endsley, M. 2016. *Designing for situation awareness: An approach to user-centered design*. CRC Press. Retrieved from https://books.google.com/books?hl=en&lr=&id=eRPBkapAsggC&oi=fnd&pg=PP1&ots=dJIDDbrVlI&sig=JUoCXkSF7MIGhT-0uPgSnIwnN4o

Endsley, M. R. 1988. Design and Evaluation for Situation Awareness Enhancement. *Proceedings of the Human Factors Society Annual Meeting*, *32*(2): 97–101. http://doi.org/10.1177/154193128803200221

Endsley, M. R. 1995. Measurement of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1): 65–84. http://doi.org/10.1518/001872095779049499

Endsley, M. R. 1995. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1): 32–64. http://doi.org/10.1518/001872095779049543

Endsley, M. R. 1996. Automation and situation awareness. *Automation and Human Performance: Theory and Applications*, 163–181.

Endsley, M. R. 2015. Situation awareness : operationally necessary and scientifically grounded. *Cognition, Technology & Work*, 163–167. http://doi.org/10.1007/s10111-015-0323-5

Endsley, M. R., & Kaber, D. B. 1999. *Level of automation effects on performance, situation awareness and workload in a dynamic control task. Ergonomics* (Vol. 42). http://doi.org/10.1080/001401399185595

Endsley, M. R., & Kiris, E. O. 1995. The Out-of-the-Loop Performance Problem and Level of Control in Automation. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(2): 381–394. http://doi.org/10.1518/001872095779064555

Endsley, M. R., Bolte, B., & Jones, D. G. 2003. *Designing for Situation Awareness: An Approach to User-Centered Design*. CRC Press.

Endsley, M. R., Selcon, S. J., Hardiman, T. D., & Croft, D. G. 1998. A Comparative Analysis of Sagat and Sart for Evaluations of Situation Awareness. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 42, pp. 82–86). SAGE Publications. http://doi.org/10.1177/154193129804200119

Endsley, M., Sollenberger, R. L., Nakata, A., & Stein, E. S. 2000. *Situation awareness in air traffic control: Enhanced displays for advanced operations*.

Endsley, M., Sollenberger, R. L., Nakata, A., & Stein, E. S. 2000. *Situation awareness in air traffic control : Enhanced displays for advances operations*.

Endsley, M., Sollenberger, R., Nakata, A., & Stein, E. S. 2000. *Situation awareness in air traffic control: Enhanced displays for advanced operations*.

Feldman, R. 2008. *Essentials of Understanding Psychology* (7th ed.). New York City;New York: McGraw-Hill Education. Retrieved from http://books.google.ca/books?id=XM5gPwAACAAJ

Fennell, K., Sherry, L., Roberts, R. J., & Feary, M. 2006. Difficult access: The impact of recall steps on flight management system errors. *International Journal of Aviation Psychology*, *16*(2): 175–196. http://doi.org/10.1207/s15327108ijap1602_4

Fracker, M. L., & Vidulich, M. A. 1991. Measurement of situation awareness: A brief review. In *Designing for Everyone, Proceedings of the 11th Congress of the International Ergnomics Association* (pp. 795–797).

Gibson, J. J. 1977. The theory of affordances (RE Shaw &). Hillsdale, NJ: Lawrence Erlbaum Associates.

Gibson, J. J. 2014. *The ecological approach to visual perception: classic edition*. Psychology Press.

Global Market Insights. 2018. Agriculture Equipment Market size to cross $200bn by 2024. https://globenewswire.com/news-release/2018/08/30/1563162/0/en/Agriculture-Equipment-Market-size-to-cross-200bn-by-2024-Global-Market-Insights-Inc.html (2018/12/19).

Golestan, K., Soua, R., Karray, F., & Kamel, M. S. 2016. Situation awareness within the context of connected cars: A comprehensive review and recent trends. *Information Fusion*, *29*, 68–83. http://doi.org/10.1016/j.inffus.2015.08.001

Gorman, J. C., Cooke, N. J., & Winner, J. L. 2006. Measuring team situation awareness in decentralized command and control environments. *Ergonomics*, *49*(12–13): 1312–1325.

Gugerty, L. 1997. Situation awareness during driving: Explicit and implicit knowledge in dynamic spatial memory. *Journal of Experimental Psychology: Applied*, *3*(1): 42–66. Retrieved from https://www.researchgate.net/profile/Leo_Gugerty/publication/230898154_Situation_Awareness_During_Driving_Explicit_and_Implicit_Knowledge_in_Dynamic_Spatial_Memory/links/09e41505e72cc3f556000000.pdf

Gugerty, L. L. J. 1997. Situation awareness during driving: Explicit and implicit knowledge in dynamic spatial memory. *Journal of Experimental Psychology: Applied*, *3*(1): 42–66. Retrieved from https://www.researchgate.net/profile/Leo_Gugerty/publication/230898154_Situation_Awareness_During_Driving_Explicit_and_Implicit_Knowledge_in_Dynamic_Spatial_Memory/links/09e41505e72cc3f556000000.pdf

Hanoch, Y. 2004. When less is more: Information, Emotional Arousal and the Ecological Reframing of the Yerkes-Dodson Law. *Theory & Psychology*, *14*(4): 427–452. http://doi.org/10.1177/0959354304044918

Hart, S. G., & Staveland, L. E. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology* (Hancock, P, Vol. 52, pp. 139–183). North-Holland, New York. http://doi.org/10.1016/S0166-4115(08)62386-9

Hart, S. G., & Staveland, L. E. 1988. Development of NASA-TLX: Results of empirical and theoretical research. In *Human Mental Workload* (Vol. 52, pp. 139–183). Elsevier. http://doi.org/10.1016/S0166-4115(08)62386-9

Hogg, D. N., FOLLES?, K., Strand-Volden, F., & Torralba, B. 1995. Development of a situation awareness measure to evaluate advanced alarm systems in nuclear power plant control rooms. *Ergonomics*, *38*(11): 2394–2413.

Horrey, W. J. 2011. Assessing the Effects of In-Vehicle Tasks on Driving Performance. *Ergonomics in Design: The Quarterly of Human Factors Applications*, *19*(4): 4–7.

Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. 1991. User interface evaluation in the real world: a comparison of four techniques. *System*, *91*(c): 119–124. http://doi.org/10.1145/108844.108862

Jensen, S.E. Tudor, S.G., Adams, M. I. 1994. *Pilot workload in single- seat TIALD operation.*

Johnson, E. N., & Pritchett, A. R. 1995. Experimental study of vertical flight path mode awareness. In *Proceedings of the 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Man-Machine Systems* (pp. 185–190). International Center for Air Transportation. Retrieved from https://www.sciencedirect.com/science/article/pii/S1474667017452256

Johnson, J. 2010. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. *Morgan Kaufmann*. http://doi.org/10.1016/B978-0-240-81176-5.X0001-9

Jones, D. G., & Endsley, M. R. 1996. Sources of situation awareness errors in aviation. *Aviation, Space, and Environmental Medicine*, *67*(6): 507–512.

Karimi, D., Mann, D. D., & Yan, J. 2011. A comparison of textual , symbolic , and pictorial presentation of information on an air-seeder display, *2*(4): 90–95.

Klemmer, E. T., & Frick, F. C. 1953. Assimilation of information from dot and matrix patterns. *Journal of Experimental Psychology*, *45*(1): 15–19. http://doi.org/10.1037/h0060868

Koch, S. H., Weir, C., Westenskow, D., Gondan, M., Agutter, J., Haar, M., … Staggers, N. 2013. Evaluation of the effect of information integration in displays for ICU nurses on situation awareness and task completion time: A prospective randomized controlled study. *International Journal of Medical Informatics*, *82*(8): 665–675. http://doi.org/10.1016/j.ijmedinf.2012.10.002

Lenorowitz, J. M. 1988. A320 Crash Investigation Centers on Crew's Judgement during Flyby. *Aviation Weekly & Space Technology*, *4*(7): 88.

Liu, Y. 1997. Software-user interface design. In G. Salvendy (Ed.), (Second, pp. 1689–1724). New York City, NY:John Wiley & Sons.

Liu, Y. C. 2001. Comparative study of the effects of auditory, visual and multimodality displays on drivers' performance in advanced traveller information systems. *Ergonomics*, *44*(4): 425–42. http://doi.org/10.1080/00140130010011369

Matthews, M. L., Bryant, D. J., Webb, R. D. G., & Harbluk, J. L. 2001. Model for situation awareness and driving: Application to analysis and research for intelligent transportation systems. *Transportation Research Record: Journal of the Transportation Research Board*, *1779*(1): 26–32.

Megaw, T. 2005. The definition and measurement of mental workload. In *Evaluation of Human Work, 3rd Edition* (pp. 525–551). http://doi.org/10.1201/9781420055948.ch18

Miller, G. 1956. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*. http://doi.org/10.1037/h0043158

Muckler, F., & Factors, S. S. 1992. Selecting performance measures '"objective"' versus '"subjective"' measurement. *Human Factors*, *34*(4): 441–455. Retrieved from http://journals.sagepub.com/doi/abs/10.1177/001872089203400406

Nielsen, J. 1994. Usability inspection methods. *Conference Companion on Human Factors in Computing Systems - CHI '94*, 413–414. http://doi.org/10.1145/259963.260531

Nielsen, J., & Molich, R. 1990. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 249–256). ACM.

Norman, D. A. 1983. Design Rules Based on Analyses of Human Error. *Communications of the ACM*, *26*(4): 254–258.

Norman, D. A. 1990. The "problem" with automation: inappropriate feedback and interaction, not "over-automation". *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *327*(1241): 585–93. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/1970904

Norman, D. A. 1999. Affordance, convention, and design. *Interactions, 6*(3): 38–42.

Parasuraman, R., & Riley, V. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *39*(2): 230–253.

Parasuraman, R., & Riley, V. 1997. Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *39*(2): 230–253. http://doi.org/10.1518/001872097778543886

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. 2000. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans : A Publication of the IEEE Systems, Man, and Cybernetics Society*, *30*(3): 286–97. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/11760769

Pickup, L., Wilson, J. R., Norris, B. J., Mitchell, L., & Morrisroe, G. 2005. The Integrated Workload Scale (IWS): a new self-report tool to assess railway signaller workload. *Applied Ergonomics*, *36*(6): 681–93. http://doi.org/10.1016/j.apergo.2005.05.004

Pollack, I. 1952. The Information of Elementary Auditory Displays. *The Journal of the Acoustical Society of America*, *24*(6): 745–749. http://doi.org/10.1121/1.1906969

Pollack, I. 1953. The Information of Elementary Auditory Displays. II. *The Journal of the Acoustical Society of America*, *25*(4): 765–769. http://doi.org/10.1121/1.1907173

Rakhra, A. K., & Mann, D. D. 2014. Design guidelines review and conceptual design of an user-centered information display for mobile agricultural machines. In *American Society of Agricultural and Biological Engineers Annual International Meeting 2014, ASABE 2014* (Vol. 4, pp. 2917–2932). Retrieved from https://elibrary.asabe.org/abstract.asp?aid=44456

Rakhra, A. K., & Mann, D. D. 2018. Design and evaluation of individual elements of the interface for an agricultural machine. *Journal of Agricultural Safety and Health*, *24*(1): 27–42.

Rakhra, A., Mann, D., & Bashiri, B. 2013. User-centered information display design for tractor air seeder systems. In *CSBE/SCGAB 2013 Annual Conference*. University of Saskatchewan, Saskatoon, Saskatchewan. Retrieved from http://www.csbe-scgab.ca/docs/meetings/2013/CSBE13010.pdf

Salmon, P., Stanton, N., Walker, G., & Green, D. 2006. Situation awareness measurement: A review of applicability for C4i environments. *Applied Ergonomics*, *37*(2): 225–238. http://doi.org/10.1016/j.apergo.2005.02.001

Sanchez, J., & Duncan, J. R. 2009. Operator-Automation Interaction in Agricultural Vehicles. *Ergonomics in Design: The Quarterly of Human Factors Applications*, *17*(1): 14–19. http://doi.org/10.1518/106480409X415161

Selcon, S. J., Taylor, R. M., & Koritsas, E. 1991. Workload or Situation Awareness?: TLX vs. SART for Aerospace Stystems Design Evalution. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, *35*(2): 62–66. http://doi.org/10.1518/107118191786755706

Sharpless, S., & Jasper, H. 1956. Habituation of the arousal reaction. *Brain*, *79*(4): 655–680. http://doi.org/10.1093/brain/79.4.655

Shneiderman, B. 1998. *Designing the user interface: strategies for effective human-computer interaction* (3rd ed.). Reading, MA: Addison Wesley Longman.

Shneiderman, B., Plaisant, C., Cohen, M., & Jacobs, S. 2009. *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (5th ed.). Reading, MA: Addison Wesley Longman.

Sinclair, M. A. 2005. Participative assessment. In *Evaluation of Human Work* (pp. 83–111). Retrieved from https://books.google.com/books?hl=en&lr=&id=dSmKYLp82b4C&oi=fnd&pg=PA83&dq=participative+assessment&ots=JHWqZqJSpu&sig=DONlZwhn7FQ9hlq-8vd8DXOZkQ0

Singh, H., Petersen, L. A., & Thomas, E. J. 2006. Understanding diagnostic errors in medicine: a lesson from aviation. *Quality and Safety in Health Care*, *15*(3): 159–164.

Skitka, L. J., Mosier, K. L., & Burdick, M. 1999. Does automation bias decision-making? *International Journal of Human Computer Studies*, *51*(5): 991–1006. http://doi.org/10.1006/ijhc.1999.0252

Squire, P. N., & Parasuraman, R. 2010. Effects of automation and task load on task switching during human supervision of multiple semi-autonomous robots in a dynamic environment. *Ergonomics*, *53*(8): 951–61. http://doi.org/10.1080/00140139.2010.489969

Stanton, N. A., & Marsden, P. 1996. From fly-by-wire to drive-by-wire: Safety implications of automation in vehicles. *Safety Science*, *24*(1): 35–49. http://doi.org/10.1016/S0925-7535(96)00067-7

Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., & Goodrich, M. 2006. Common metrics for human-robot interaction. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction* (pp. 33–40). ACM.

Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. 2005. *User interface design and evaluation*. Burlington;MA:Morgan Kaufmann.

Strater, L. D., Endsley, M. R., Pleban, R. J., & Matthews, M. D. 2001. *Measures of platoon leader situation awareness in virtual decision-making exercises*.

Taylor, H., Lee, B., Jhingory, J., Drayer, G. E., & Howard, A. M. 2010. Development and Evaluation of User Interfaces for Situation Observability in Life Support Systems, 1–8.

Todorovic, D. 2008. Gestalt principles. *Scholarpedia*, *3*(12): 5345.

Tullis, T. T. S. 1981. An evaluation of alphanumeric, graphic, and color information displays. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *23*(5): 541–550. Retrieved from http://journals.sagepub.com/doi/abs/10.1177/001872088102300504

Vicente, K. J., Rasmussen, J., & Member, S. 1992. Ecological Interface Design : Theoretical Foundations, *22*(4).

Ward, N. J. 2000. Automation of task processes: An example of intelligent transportation systems. *Human Factors and Ergonomics in Manufacturing & Service Industries*, *10*(4): 395–408.

Wikipedia contributors. 2014. Gestalt Pyschology. Retrieved from https://en.wikipedia.org/w/index.php?title=Gestalt_psychology&oldid=844691826

Wikipedia contributors. 2018. Gestalt psychology. Retrieved from https://en.wikipedia.org/w/index.php?title=Gestalt_psychology&oldid=844691826

Wolfmaier, T. G. 1999. Designing for the color-challenged: A challenge. Retrieved September 19, 2003, from http://www.internettg.org/newsletter/mar99/accessibility_color_challenged.html

Wulf, F., Rimini-Doring, M., Arnon, M., & Gauterin, F. 2015. Recommendations Supporting Situation Awareness in Partially Automated Driver Assistance Systems. *IEEE Transactions on Intelligent Transportation Systems*, *16*(4): 2290–2296. http://doi.org/10.1109/TITS.2014.2376572

Young, M. S., Stanton, N. A., & Harris, D. 2007. Driving automation: learning from aviation about design philosophies. *International Journal of Vehicle Design*, *45*(3): 323–338.

Yuditsky, T., Sollenberger, R. L., Rocco, P. S. Della, Friedman-Berg, F. and Manning, C. A. (2002) *Application of color to reduce complexity in air traffic control*, *National Technical Information Service, Springfield, Virginia, 22161*

PUBLICATIONS

We have shared or intend to share the research work from this study with the scientific community in the form of journal articles, posters, presentations or conference proceedings. The list of the publications is as follows:

Rakhra, A.K., & Mann, D.D. 2019. Design and testing of the User-Centered interface for situation awareness and mental workload of the operator of air seeder under varying levels of automation. Journal: To be determined (under progress).

Rakhra, A. K., & Mann, D. D. 2018. Design and evaluation of individual elements of the interface for an agricultural machine. Journal of Agricultural Safety and Health, 24(1): 27–42.

Rakhra, A.K., D. D. Mann. 2017. Evaluation of implement monitoring systems. In: Canadian Society for Bioengineering (CSBE) Annual Conference, 10-13 July 2011, Winnipeg, MB.

Rakhra, A.K., D. D. Mann. 2016. Design & evaluation of pictorials used to display parameter information for an agricultural machine. In: International Conference on Agricultural Engineering, CIGR-AgEng 2016, 26-29 June, Aarhus, Denmark.

Rakhra, A.K., D. D. Mann. 2014. Design guidelines review and conceptual design of a user-centered information display for mobile agricultural machines. In: ASABE and CSBE/SCGAB Annual International Meeting, 13-16 July, Montreal, QC.

Rakhra, A.K., D. D. Mann, B. Bashiri. 2013. A user-centered information display design for tractor air-seeder systems. In: Conference for Interdisciplinary Engineering, 7-10 July 2013, Saskatoon, SK.

Appendix A

A.1 ETHICS APPROVAL CERTIFICATE FOR THE STUDY

UNIVERSITY OF MANITOBA | Research Ethics and Compliance
Office of the Vice-President (Research and International)

Human Ethics
208-194 Dafoe Road
Winnipeg, MB
Canada R3T 2N2
Phone +204-474-7122
Fax +204-269-7173

APPROVAL CERTIFICATE

March 16, 2015

NSERC

TO: **Aadesh Rakhra** (Advisor D. Mann)
Principal Investigator

FROM: **Lorna Guse, Chair**
Education/Nursing Research Ethics Board (ENREB)

Re: **Protocol #E2014:162**
"User-centered driver interface design for tractor air-seeder system"

Please be advised that your above-referenced protocol has received human ethics approval by the **Education/Nursing Research Ethics Board**, which is organized and operates according to the Tri-Council Policy Statement (2). **This approval is valid for one year only**.

Any significant changes of the protocol and/or informed consent form should be reported to the Human Ethics Secretariat in advance of implementation of such changes.

Please note:

- **If you have funds pending human ethics approval, please mail/e-mail/fax (261-0325) a copy of this Approval (identifying the related UM Project Number) to the Research Grants Officer in ORS in order to initiate fund setup. (How to find your UM Project Number: http://umanitoba.ca/research/ors/mrt-faq.html#pr0)**

- **if you have received multi-year funding for this research, responsibility lies with you to apply for and obtain Renewal Approval at the expiry of the initial one-year approval; otherwise the account will be locked.**

The Research Quality Management Office may request to review research documentation from this project to demonstrate compliance with this approved protocol and the University of Manitoba *Ethics of Research Involving Humans*.

The Research Ethics Board requests a final report for your study (available at: http://umanitoba.ca/research/orec/ethics/human_ethics_REB_forms_guidelines.html) **in order to be in compliance with Tri-Council Guidelines.**

umanitoba.ca/research

A.2 CONSENT FORM FOR THE PARTICIPANTS USED DURING THE FIRST-PHASE OF THE STUDY

UNIVERSITY OF MANITOBA | Faculty of Agricultural and Food Sciences

EST. 1877

Department of Biosystems Engineering
E2-376 EITC
Winnipeg MB R3T 5V6
CANADA
Tel: 204-474-6033
Fax: 204-474-7512

User-centered driver interface design for tractor air-seeder system.

Principal Investigator:
Aadesh Rakhra
Ph.D. Student, P.Eng.
E2-376 EITC
Department of Biosystems Engineering
University of Manitoba
Winnipeg, MB R3T 5V6
Phone: (204) 474-7966
E-mail: umrakhr2@myumanitoba.ca

Advisor:
Danny Mann, Ph.D., P.Eng.
Professor & Head
E2-376 EITC
Department of Biosystems Engineering
University of Manitoba
Winnipeg, MB R3T 5V6
Phone: (204) 474-7149
Fax: (204) 474-7512
E-mail: Danny.Mann@umanitoba.ca

This consent form, a copy of which will be left with you for your records and reference, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about anything mentioned here, or information not included here, you should feel free to ask. Please take the time to read this carefully and to understand any accompanying information.

PURPOSE: The purpose of the study is to determine the level of comprehension of the individual pictorials that have been designed for each of the relevant parameters of the air-seeder system, compared with level of comprehension of the individual pictorials that have previously been used to display the relevant parameters of the air-seeder system.
This research will be used as part of a doctoral dissertation study. Funding for this research has been provided by NSERC.

DESCRIPTION: In order to compare the individual newly designed pictorials with the previously used pictorials, one parameter (out of 16 parameters) at a time will be displayed on a computer screen. After the end of the parameter, a set of queries will be presented to you. You are expected to respond to the queries by selecting the appropriate response with the use of the mouse and/or keyboard. Once you have responded, a new parameter will appear. Each parameter out of 16 will appear in 3 different configurations, so in total you have to observe the parameters and respond to the queries 48 times in total during the experiment session. There will be 5 minutes rest time in the middle of the experiment session.

Before the start of the experimental session you are required to fill a questionnaire; a brief training session will also be provided. After the session you are required to fill another questionnaire. This whole experimental session should be completed in 90-120 minutes.

RISKS AND BENEFITS: You will not be subject to any risk as a result of the study. During the experimental session you are required to watch the computer screen and respond to the queries being asked by the computer program using mouse and/or keyboard.

COSTS AND PAYMENTS: There are no fees or charges to participate in this study.

CONFIDENTIALITY: Your personal information cannot be identified or included in any publication, conference or presentation based on this study. Only the Principal Investigator, Mr. Aadesh Rakhra, will know the names of the participants. The names of the participants will be encoded for the purpose of data analysis. During data analysis, all data will be grouped together. The participant's name will be removed from any materials that could associate his/her identity with the responses he/she provides. The identity of the participants will not be included or reported in any future publication, academic conferences or invited presentations based on the results of this study. All files containing identifying information (consent forms) will be stored in a cabinet in the Agricultural Ergonomics Laboratory (A115 AEB) or at the residence of the Principle Investigator. However participation of the students when they will come, leave or stay in the lab for experiment session cannot be kept secret.

All consent forms will be destroyed in December 2018.

DEBRIEFING: Feedback will not be provided immediately upon completion of participation in the study. Results will be sent to the interested participants in the form of a summary sheet. Interested participants can provide their mailing or email addresses at the end of the consent form.

DISSEMINATION: The outcome of the study will be used in the Thesis/Practicum required as partial fulfilment of the requirement of the degree Doctor of Philosophy in the Department of Biosystems Engineering for University of Manitoba. The results of the study will be also be presented in the form of oral presentation open to the public, as well as in written form to a scientific publication in an appropriate journal or conference proceedings.

VOLUNTARY CONSENT: If you *do not wish to participate* in the study, you are free to leave without consequence and we thank you for your consideration. You are free to withdraw from the study at any time, and/or refrain from answering any questions you prefer to omit, without prejudice or consequence by telling Aadesh Rakhra in person or email. In case of withdrawal from the study all data collected form you will be destroyed.

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the researchers, sponsors, or involved institutions from their legal and professional responsibilities. You are free to withdraw from the study at any time, and/or refrain from answering any questions you prefer to omit, without prejudice or consequence. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. The University of Manitoba may look at our research records to see that the research is being done in a safe and proper way.

This research has been approved by the Education/Nursing REB. If you have any concerns or complaints about this project you may contact any of the above-named persons i.e. Aadesh Rakhra or Danny Mann, or the Human Ethics Coordinator (HEC), Margaret Bowman at 474-7122 or email at margaret.bowman@ad.umanitoba.ca.

A copy of this consent form has been given to you to keep for your records and reference.

☐ Check this box if you would like to receive the summary of the research.
Please provide your email address or postal address to receive the summary of the report.

Email:	
Postal Address:	

Participant Signature ______________________________ Date ____________

Researcher Signature ______________________________ Date ____________

A.3 SOCIO DEMOGRAPHIC QUESTIONNAIRE (PHASE 1)

Subject ID:

Age:

Gender:

Ethnicity:

First Language:

Have you ever drove tractor or any other farming vehicle? If yes, which vehicle and for how many years?

Have you ever used interface or display for precision farming or machine control? If yes, which brand and for what application (e.g. AgLeader for planting operation)?

A.4 SUBJECTIVE FEEDBACK QUESTIONNAIRE (PHASE I)

Subjective feedback questionnaire

Subject ID: _____

Please answer the following questions based on the two images shown on the accompanying page; Page 1.

1. Which display improves your ability to **identify** the parameter? A B
2. Which display improves your ability to **recall** (remember) the parameter value?

   A B
3. Which display improves your ability to **understand/comprehend** the situation?

   A B
4. Which display improves your ability to identify the **trend**? A B
5. Which display improves your ability to **foresee** and **predict** the situation?

   A B
6. If your goal is to **control and maintain** this parameter within the recommended range, which display do you **prefer**? A B
7. Which display do you **like** better? A B
8. **Why** do you think this display is **better**?
9. What are the **weaknesses** of the other display?
10. Do you have any comments that might improve either one of the displays?

A.5 CONSENT FORM FOR THE PARTICIPANTS USED DURING THE SECOND-PHASE OF THE STUDY

UNIVERSITY OF MANITOBA | Faculty of Agricultural and Food Sciences

Department of Biosystems Engineering
E2-376 EITC
Winnipeg MB R3T 5V6
CANADA
Tel: 204-474-6033
Fax: 204-474-7512

User-centered driver interface design for tractor air-seeder system.

Principal Investigator:
Aadesh Rakhra
Ph.D. Student, P.Eng.
E2-376 EITC
Department of Biosystems Engineering
University of Manitoba
Winnipeg, MB R3T 5V6
Phone: (204) 474-7966
E-mail: umrakhr2@myumanitoba.ca

Advisor:
Danny Mann, Ph.D., P.Eng.
Professor & Head
E2-376 EITC
Department of Biosystems Engineering
University of Manitoba
Winnipeg, MB R3T 5V6
Phone: (204) 474-7149
Fax: (204) 474-7512
E-mail: Danny.Mann@umanitoba.ca

This consent form, a copy of which will be left with you for your records and reference, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about anything mentioned here, or information not included here, you should feel free to ask. Please take the time to read this carefully and to understand any accompanying information.

PURPOSE: The purpose of this study is to design and evaluate a tractor air- seeder driver interface using situation awareness, and other design principles based on cognitive and perceptual psychology.

This research will be used as part of a doctoral dissertation study. Funding for this research has been provided by NSERC.

DESCRIPTION: In order to evaluate and compare the newly designed tractor air-seeder driver interface with the previously used interface, six types of driver interface configurations (old interface in manual and automatic steering; new interface-version 1 in manual and automatic steering configuration, and new interface-version 2 in manual and automatic steering configuration) will be shown randomly to the user on a computer screen. You need to interact with the interface using mouse according to the situation. After every interface simulation questions will appear on the screen, and you need to answer the questions using mouse and keyboard. This same experiment will be repeated under second scenario where you are not allowed to interact with the interface; you only need to respond to questions asked after every simulation.

Before starting the experimental session, a practice session (of around 5 min) will be provided. You are also required to fill a demographic questionnaire.

After the end of the session, you are required to fill a post-trial questionnaire. This whole experimental session should be completed in approximately 2 hours.

RISKS AND BENEFITS: You will not be subject to any risk as a result of the study. During the experimental session you are required to watch the computer screen and respond to the queries being asked by the computer program using mouse and/or keyboard.

COSTS AND PAYMENTS: There are no fees or charges to participate in this study. A $20 gift card will be provided as honorarium.

CONFIDENTIALITY: Your personal information cannot be identified or included in any publication, conference or presentation based on this study. Only the Principal Investigator, Mr. Aadesh Rakhra, will know the names of the participants. The names of the participants will be encoded for the purpose of data analysis. During data analysis, all data will be grouped together. The participant's name will be removed from any materials that could associate his/her identity with the responses he/she provides. The identity of the participants will not be included or reported in any future publication, academic conferences or invited presentations based on the results of this study. All files containing identifying information (consent forms) will be stored in a cabinet in the Agricultural Ergonomics Laboratory (A115 AEB) or at the residence of the Principle Investigator. However participation of the students when they will come, leave or stay in the lab for experiment session cannot be kept secret.

All consent forms will be destroyed in December 2018.

DEBRIEFING: Feedback will not be provided immediately upon completion of participation in the study. Results will be sent to the interested participants in the form of a summary sheet. Interested participants can provide their email addresses at the end of the consent form.

DISSEMINATION: The outcome of the study will be used in the Thesis/Practicum required as partial fulfilment of the requirement of the degree Doctor of Philosophy in the Department of Biosystems Engineering for University of Manitoba. The results of the study will also be presented in the form of oral presentation open to the public, as well as in written form to a scientific publication in an appropriate journal or conference proceedings.

VOLUNTARY CONSENT: If you *do not wish to participate* in the study, you are free to leave without consequence and we thank you for your consideration. You are free to withdraw from the study at any time, and/or refrain from answering any questions you prefer to omit, without prejudice or consequence by telling Aadesh Rakhra in person or email. In case of withdrawal from the study all data collected form you will be destroyed.

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the researchers, sponsors, or involved institutions from their legal and professional responsibilities. You are free to withdraw from the study at any time, and/or refrain from answering any questions you prefer to omit, without prejudice or consequence. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. The University of Manitoba may look at our research records to see that the research is being done in a safe and proper way.

This research has been approved by the Education/Nursing REB. If you have any concerns or complaints about this project you may contact any of the above-named persons i.e. Aadesh Rakhra or Danny Mann, or the Human Ethics Coordinator (HEC), Ms Pinar Eskicioglu at humanethics@umanitoba.ca.

A copy of this consent form has been given to you to keep for your records and reference.

☐ Check this box if you would like to receive the summary of the research.
Please provide your email address or postal address to receive the summary of the report.

Email:	

Participant Signature ____ Date ____________

Researcher Signature ____Date ____________

A.6 SOCIO DEMOGRAPHIC QUESTIONNAIRE (PHASE II)

Subject ID:

Age:

Gender:

Ethnicity:

First Language:

Have you ever drove tractor or any other farming vehicle? If yes, which vehicle and for how many years?

Have you ever used interface or display for precision farming or machine control? If yes, which brand and for what application (e.g. AgLeader for planting operation)?

A.7 SUBJECTIVE FEEDBACK QUESTIONNAIRE (PHASE II)

Subjective feedback questionnaire

Subject ID: _____

Based on your experience during the experiment with 3 types of interfaces (A, B, & C), please answer the following questions;

1. Rate the interfaces in terms of **perception** of the information.

   Best: _____ Worst: _____ Average: _____

2. Rate the interfaces in terms of **recalling** (or remembering) of the information.

   Best: _____ Worst: _____ Average: _____

3. Rate the interfaces in terms of **understanding/comprehending** of the situation.

   Best: _____ Worst: _____ Average: _____

4. Rate the interfaces in terms of **trend** identification.

   Best: _____ Worst: _____ Average: _____

5. Rate the interfaces in terms of **prediction** of the future situation.

   Best: _____ Worst: _____ Average: _____

6. Which interface would you choose as your **preferable interface** for your agricultural machine?

   A B C

7. **Why** do you think this interface is **better**?

8. What are the **weaknesses** of the other two interfaces?

9. Do you have any suggestions that might further improve your **preferred** interface (Q6)?

Appendix B

B.1 SUBJECTIVE FEEDBACK OBTAINED DURING PHASE II OF THE STUDY

Table B.1. Subjective feedback collected during phase II of the study. Letter A, B, and C under column Q6 indicates a preference for a particular interface. Letter A indicates Old (baseline) User Interface, B – User-Centered Design Version 1, and C- User-Centered Design Version 2 interface. Two letters in column Q6 indicates two different choices during the feedback collected (during the middle break and at the end of the experiment).

Sr. No.	Q6 (Preferred Interface)	Q7 (Why your preferred interface is better?)	Q8 (Weakness of the others?)	Q9 (Further suggestion to improve your preferred interface)
1	C	Least distracting	A- need to pay close attention to the numerals increments B red/green is hard, too much data	Flashing prewarning could be added.
2	A/B	A-more graphics B- better prediction and trend	A- not easy to predict. B-too much colors. C- hard to read.	
3	C	Easy to understand, remember and better trend identification.	A-difficulty in understanding relative measurement. B-too much detail.	Better optimization of the dials to quickly detect when leaving from the safe margin.
4	B	Better trend	A – cannot estimate the next moment. C- missing graduations of the scale.	Readings could be shown in red color instead of the colored scale.
5	C	Easy, better to understand	A & B - confusing	Adding an alarming light to notify error.
6	A/C	A- simple design C- easy to predict the situation	A-no trend support B-busy and crowded C- busy	Trend symbol should be made bigger/longer.
7	B/C	B-clearly indicates permissible limits. C- giving enough information without confusing the user.	A- does not indicate a trend. B-too confusing, too much red. C- colors do not give a clear image about the out of range.	
8	C	Easy and fast to spot any change.	A-hard to follow trends. B-extra colors make reading hard.	No.
9	B/C	B- most visible. C- less busy.	A- not enough indicators for speed trend. B- busy compared to C. C-not visible enough.	Make trend icons bigger.
10	B	The color of the scale helps in the reading.	C- grayscale makes it hard to read.	Blinking should be used to pre-warn.
11	A/C	A-easy, clean. C- better in identifying the current situation.	B- too crowded	C- Speed and fan rpm should be made different.

12	C	Less red than B, more trend information than A.	A- not much trend information. B- too much red.	Use blinking. Use a bar graph for the tool depth instead of an arrow.
13	B/C	Easy to recall information.	Recalling information is hard with A and B.	Guidance bar should be separate than the other interface elements showing the readings.
14	C	Fewer colors.	A- Arrangement is not good. B- More colors.	
15	C	Simple and better.	B- overloaded and distracting details. A- Elements too diverse, no way to determine the trend.	Introduce progress bars which shows the trends themselves.
16	C	Shows better trend, not busy.	A- not showing the car moving (speed indicator). B -Very colorful.	Make the speed and fan different.
17	C	Less distracting.	B- red/green was slightly distracting. A-Photographs were not helpful. Harder to see everything at a glance.	Tool depth should be improved, may be horizontal arrow can be used, which could point out the exact reading.
18	B	Better graphics and appearance.	Need more detail. Not attractive.	Constant parameters can be made subtle (calm). The trend indicator could be made more prominent.
19	C	Easy to understand, and easier to catch attention.	Harder to find which element needs attention.	Enlarge the numbers (reading).
20	C	Better alert, easy to understand the trend	A- Poor alert and a trend presentation. B- Indication and values are not in one place for B.	Pre-alert can be provided.
21	C	C only has colors when there is an issue.	B-Too much red. A- has no gauges	
22	B	Because of the colors and the symbols.	A – Speed & tool depth; cannot tract the reduction and increment. C – Not good colors and symbols.	
23	C	Has enough indication for showing the trends and off-range values?	A – Does not help for Trend. B – Too complicated.	A prewarning can be added (like a yellow light in traffic control).
24	C	Interface C is best of all.	A – Poor job in trend/prediction. Both A and B make users eyes more tiring.	None.

25	B/C	B-Colors of the scale is clear C-Clear information identification.	A – limited information. B – look hard to follow change, busy. C- shades of grey will be hard to differentiate in dusty conditions.	Larger reading font should be used.
26	C	Clear to understand and predict the future situation.	B - Too much color A – Not easy to understand the trend	Looks fine.
27	A/C	A-Symbol/icon depicts the given information. C-Sings help in better understanding.		
28	C	Clearer, and easily identified than others	Not much at this stage but C is better	Add audio feature for alert.
29	C	Less crowded	B- too much color A- No gauges, not helpful	Improve tool depth.
30	C	Easy, tells trend, simplicity, clear.	B - Dizzy when watching long time. A – Not clear for speed change and tool depth.	May add arrow for speed and fan.

Appendix C

C.1 STUDY DESIGN AND DATA ANALYSIS CONSIDERATIONS

Both for phase 1 and phase 2 of the study, repeated measure design was considered. Repeated measure design (RMD) has advantages over the independent groups. RMD is robust against the individual differences due to the participants physical, cognitive, demographics or other characterizations affecting any specific treatment, as each subject is exposed to all treatments. For example, the individual difference due to the higher IQ, age or work experience would not influence differently to one particular treatment under consideration. Furthermore, RMD provides more statistical power in comparison to independent group design using the same number of subjects. However, RMD can cause some learning effect, which can be countered by randomizing the sequence of the treatments.

A total of 30 subjects were considered for the study. Based on the power and sample size calculations using Minitab, we obtained a sample of size of 27 for the target power of 80%, and significant level of 95% ($\alpha = 0.05$), to detect a minimum of 1 unit of difference between the paired means of two samples with an assumed standard deviation (SD) of paired differences as 2. For further exploration and estimation, we obtained a sample size of 30 and 25 for detecting the difference of 0.7 and 1.3 respectively, with an SD of paired differences as 1.5 and 2.5 respectively for a significant level of 95% and the target power of 80%. Similarly, for the phase 2 of study, for the 2 x 2 x 3 factorial design (two levels of automation x two levels of steering-type x three levels of interface design), for a target power of 80% and significant level of 95% ($\alpha = 0.05$), we would be able to detect the smallest difference of main effect means as

0.8052 for 30 subjects (360 runs) with an assumed SD as 2. Likewise, by keeping other parameters same, i.e., significance level 95%, power as 80%, number of subjects as 30, but by varying the SD of the sample to 5 (a large value), we would be able to detect the smallest difference of means of the main effect as 2.0131.

Before the analysis of the data, data were checked for its normality (visually by probability plot or using quantitative tests like Anderson-Darling), most of our data were observed normal or near normal. Although our data can also be considered as discrete, ordinal and had limited range (e.g., Likert scale or counts data), we analyzed our data using parametric tests (which relies on the assumption of continuous and normal data). However, it should be noted that non-normality of data does not affect the outcome much if the sample size is large enough like 20 or more. Particularly, the tests like t-tests, Analysis of Variance (ANOVA), and Regression are ‘robust’ to the normality assumptions. For further details, please refer to the Minitab help section (http://blog.minitab.com/blog/understanding-statistics-and-its-application/what-should-i-do-if-my-data-is-not-normal-v2, http://blog.minitab.com/blog/adventures-in-statistics-2/best-way-to-analyze-likert-item-data-two-sample-t-test-versus-mann-whitney).

Appendix D

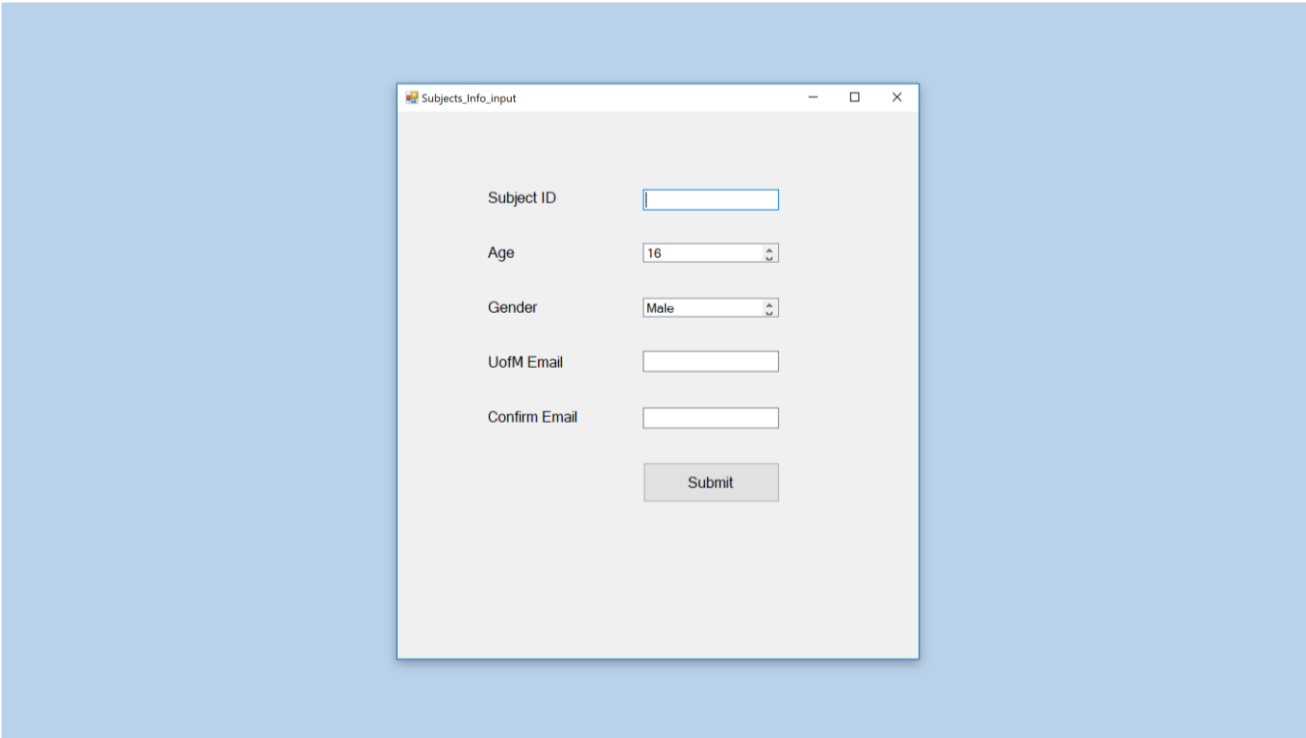
D.1 OUTPUT OF THE ORIDINAL LOGISTICS REGRESSION

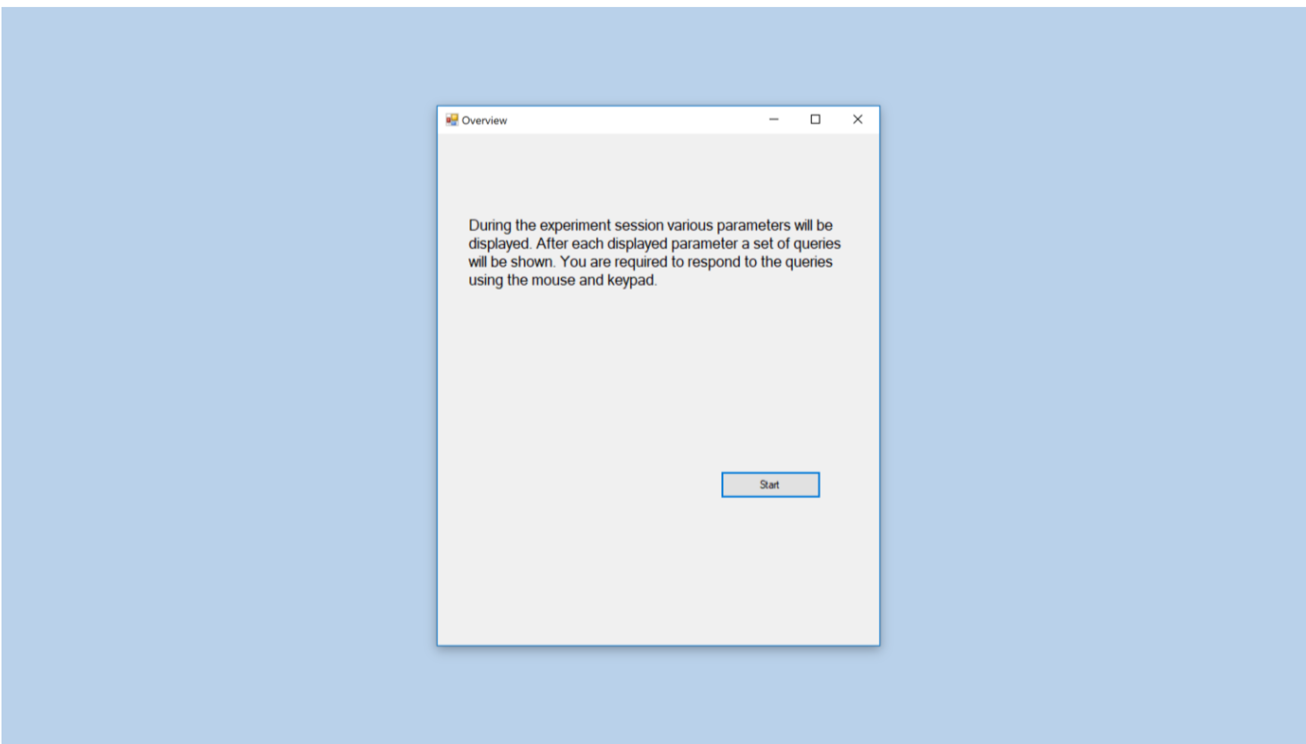
Table D.1. Ordinal Logistics Regression output (Minitab) of Interface-Ranking (best, average, worst, with a total count of 900) as dependent (response) variable versus Interface-Design (A-Old, B-UCD1 & C-UCD2) as independent (predictor) variable.

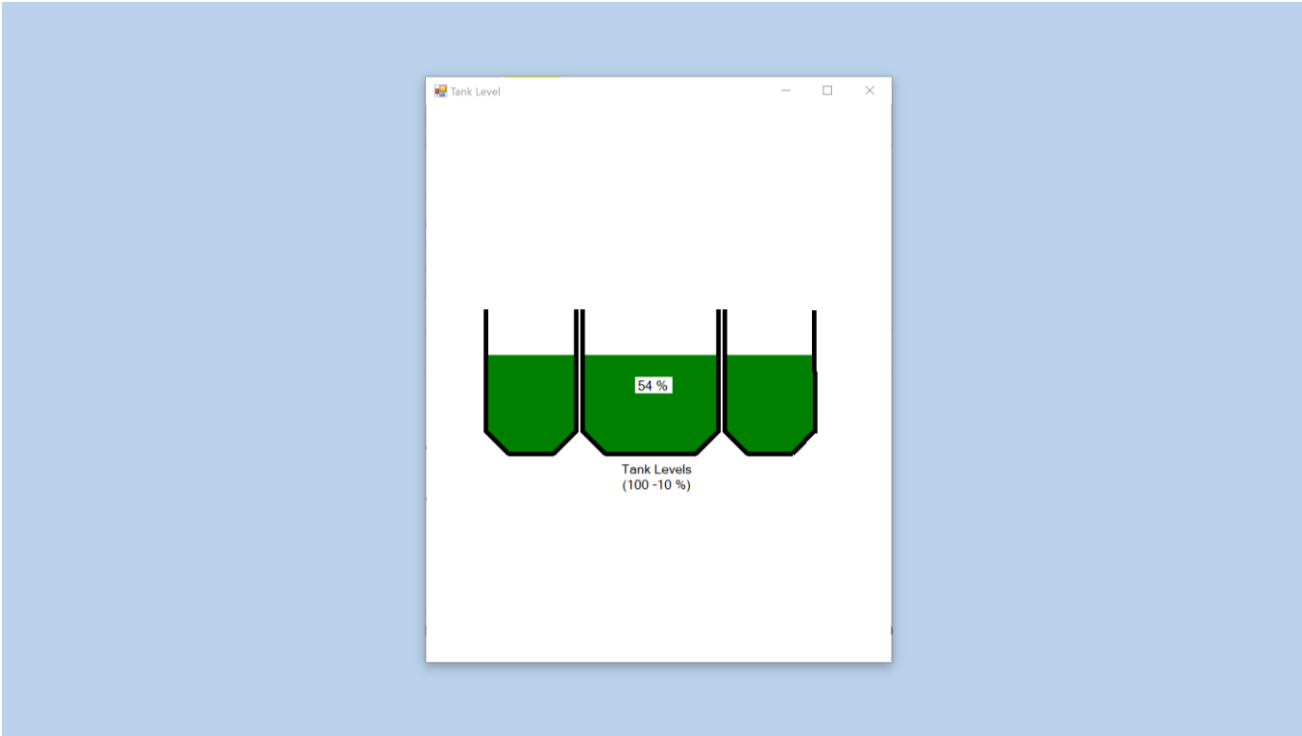
```
Link Function: Logit


Response Information

Variable           Value    Count
Interface-Ranking_  best       289
                    average    299
                    worst      312
                    Total      900


Factor Information

Factor            Levels  Values
Interface-Design       3  A, B, C


Logistic Regression Table

                                                          Odds     95% CI
Predictor              Coef   SE Coef       Z      P  Ratio  Lower  Upper
Const(1)           -2.13263  0.138911  -15.35  0.000
Const(2)          -0.393011  0.115561   -3.40  0.001
Interface-Design
 B                  1.04251  0.158723    6.57  0.000   2.84   2.08   3.87
 C                  2.55892  0.175113   14.61  0.000  12.92   9.17  18.21


Tests for terms with more than 1 degree of freedom

Term              Chi-Square  DF      P
Interface-Design     215.576   2  0.000


Log-Likelihood = -866.277
Test that all slopes are zero: G = 244.062, DF = 2, P-Value = 0.000


Goodness-of-Fit Tests

Method    Chi-Square  DF      P
Pearson      2.06868   2  0.355
Deviance     2.04950   2  0.359


Measures of Association:
(Between the Response Variable and Predicted Probabilities)

Pairs        Number  Percent  Summary Measures
Concordant  156080     57.8  Somers' D               0.45
Discordant   35903     13.3  Goodman-Kruskal Gamma  0.63
Ties         77884     28.9  Kendall's Tau-a         0.30
Total       269867    100.0
```

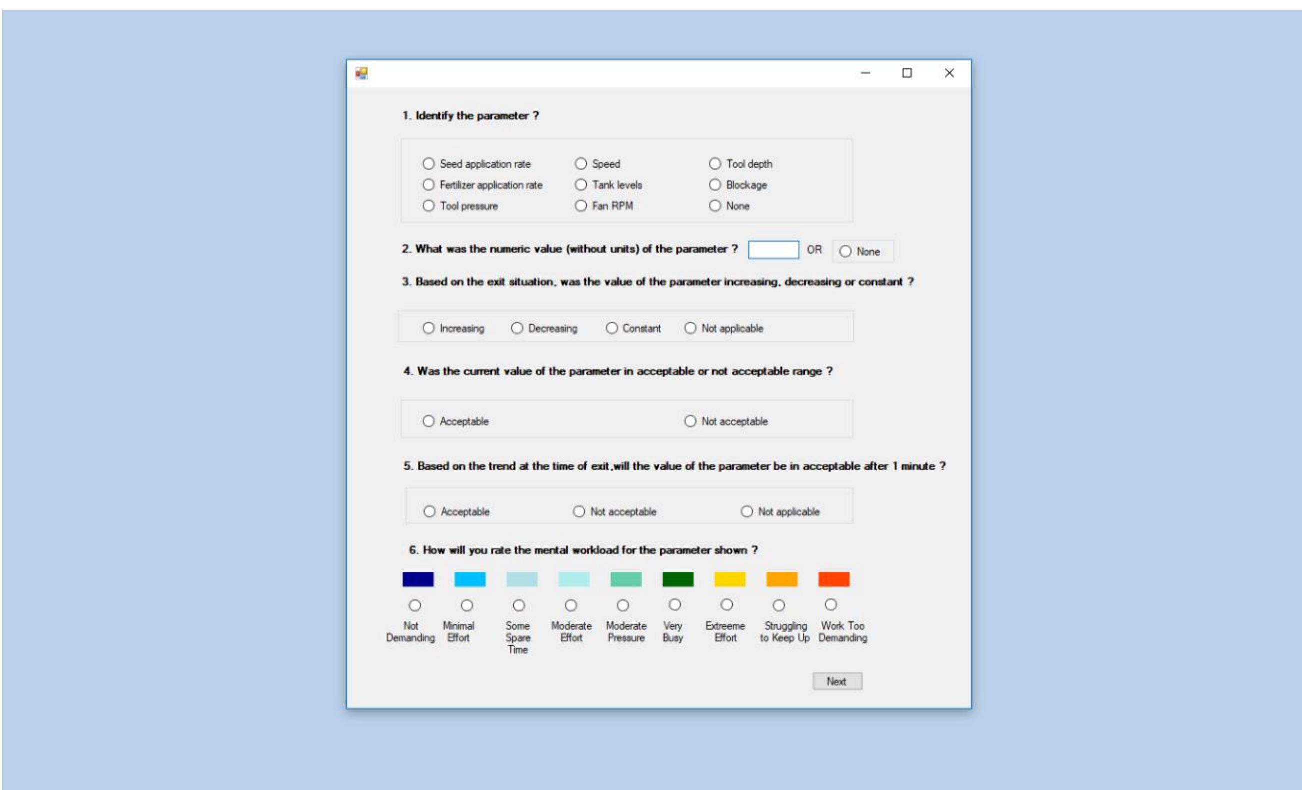
Appendix E

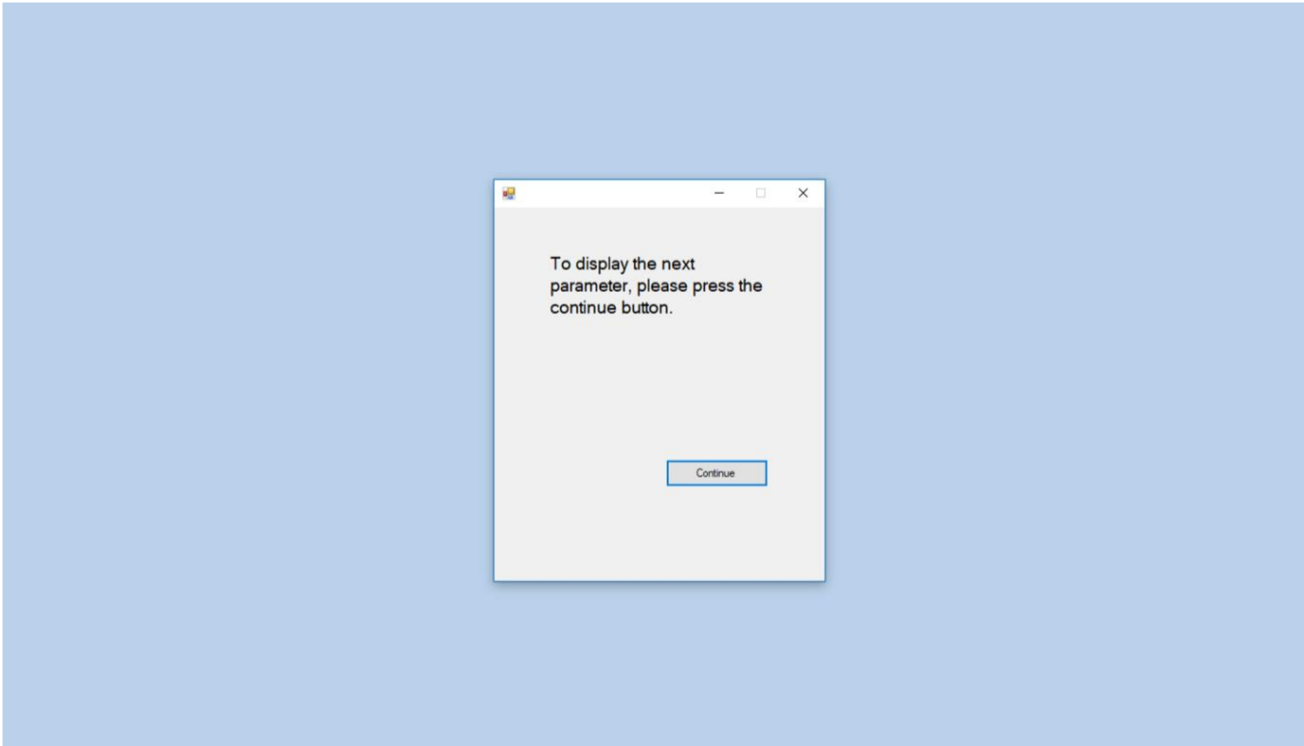
E.1 SCREENSHOTS OF THE EXPERIMENTS DURING PHASE 1 OF THE STUDY



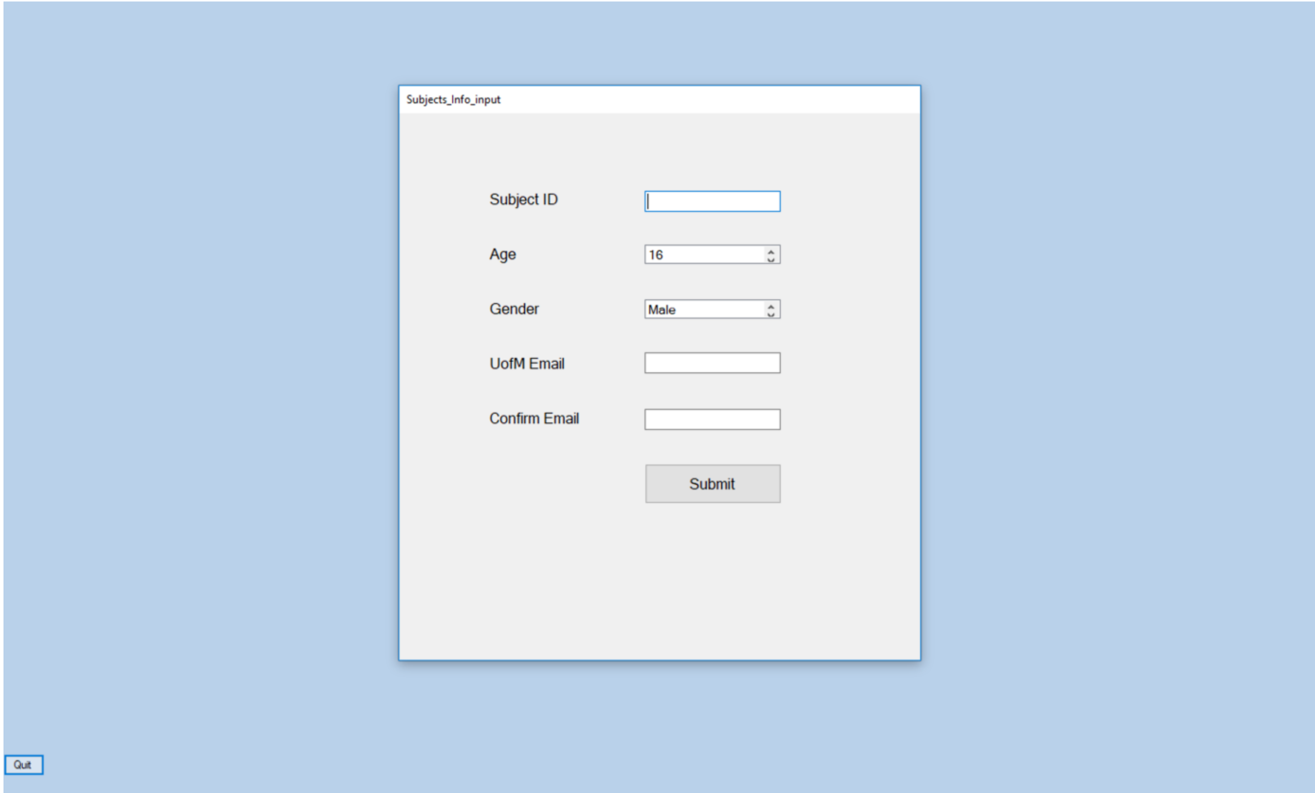


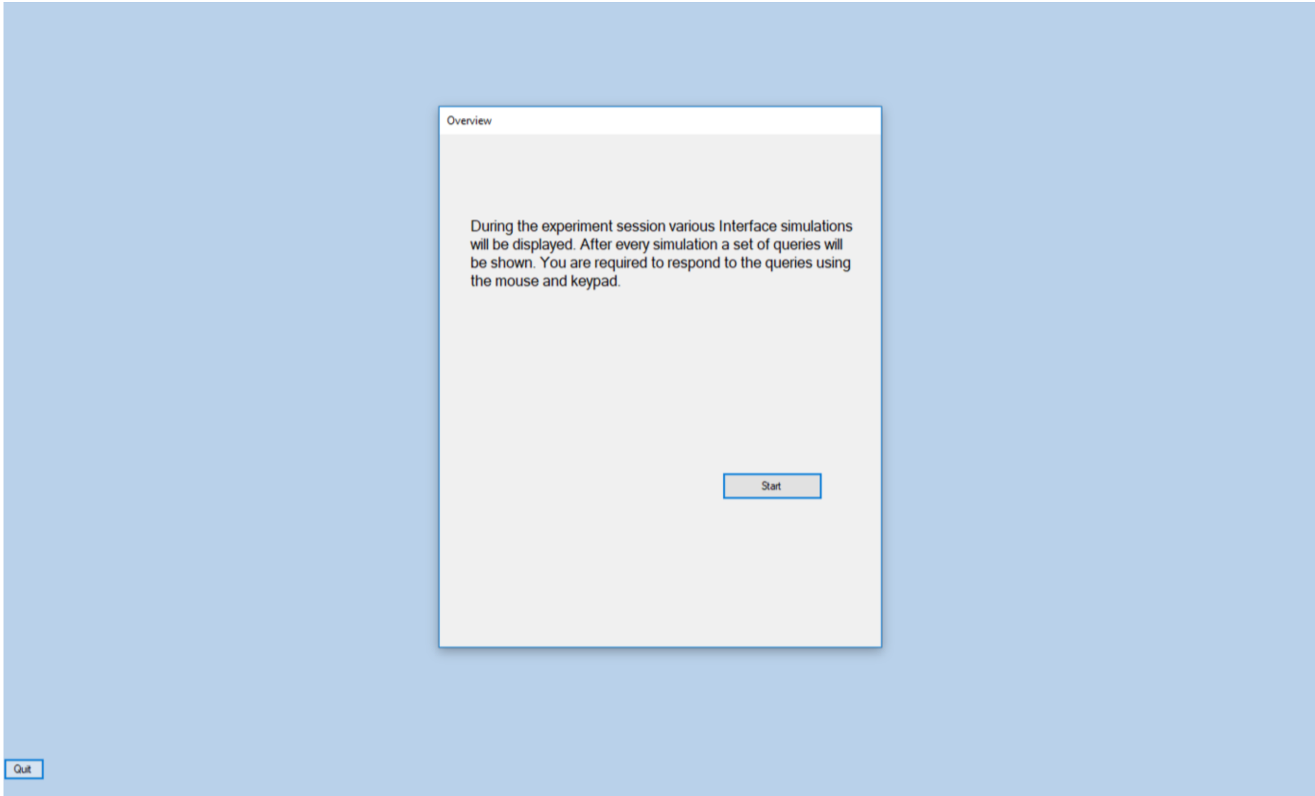

Tank Level
54 %
Tank Levels
(100 -10 %)

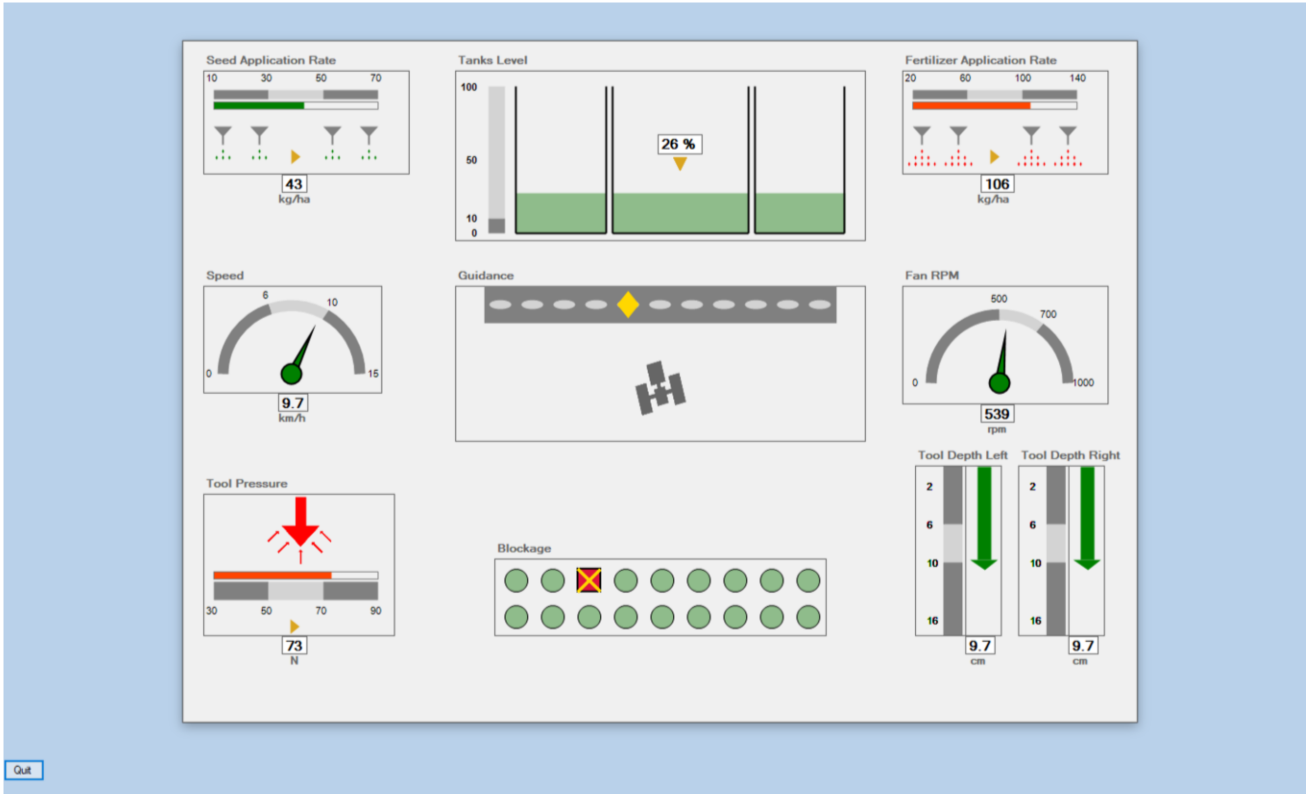

1. Identify the parameter ?
Seed application rate
Fertilizer application rate
Tool pressure
Speed
Tank levels
Fan RPM
Tool depth
Blockage
None
2. What was the numeric value (without units) of the parameter ?
OR
None
3. Based on the exit situation, was the value of the parameter increasing, decreasing or constant ?
Increasing
Decreasing
Constant
Not applicable
4. Was the current value of the parameter in acceptable or not acceptable range ?
Acceptable
Not acceptable
5. Based on the trend at the time of exit,will the value of the parameter be in acceptable after 1 minute ?
Acceptable
Not acceptable
Not applicable
6. How will you rate the mental workload for the parameter shown ?
Not Demanding
Minimal Effort
Some Spare Time
Moderate Effort
Moderate Pressure
Very Busy
Extreeme Effort
Struggling to Keep Up
Work Too Demanding
Next

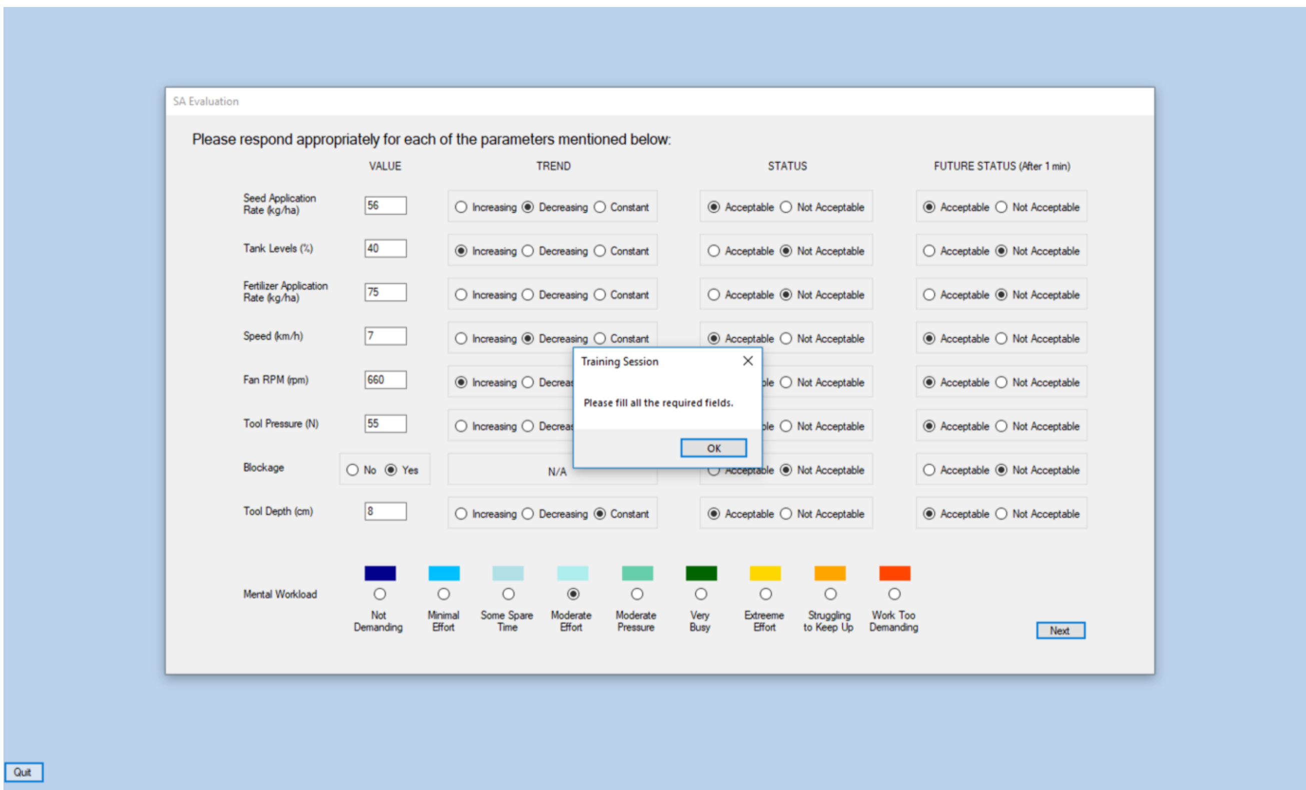

To display the next parameter, please press the continue button.
Continue

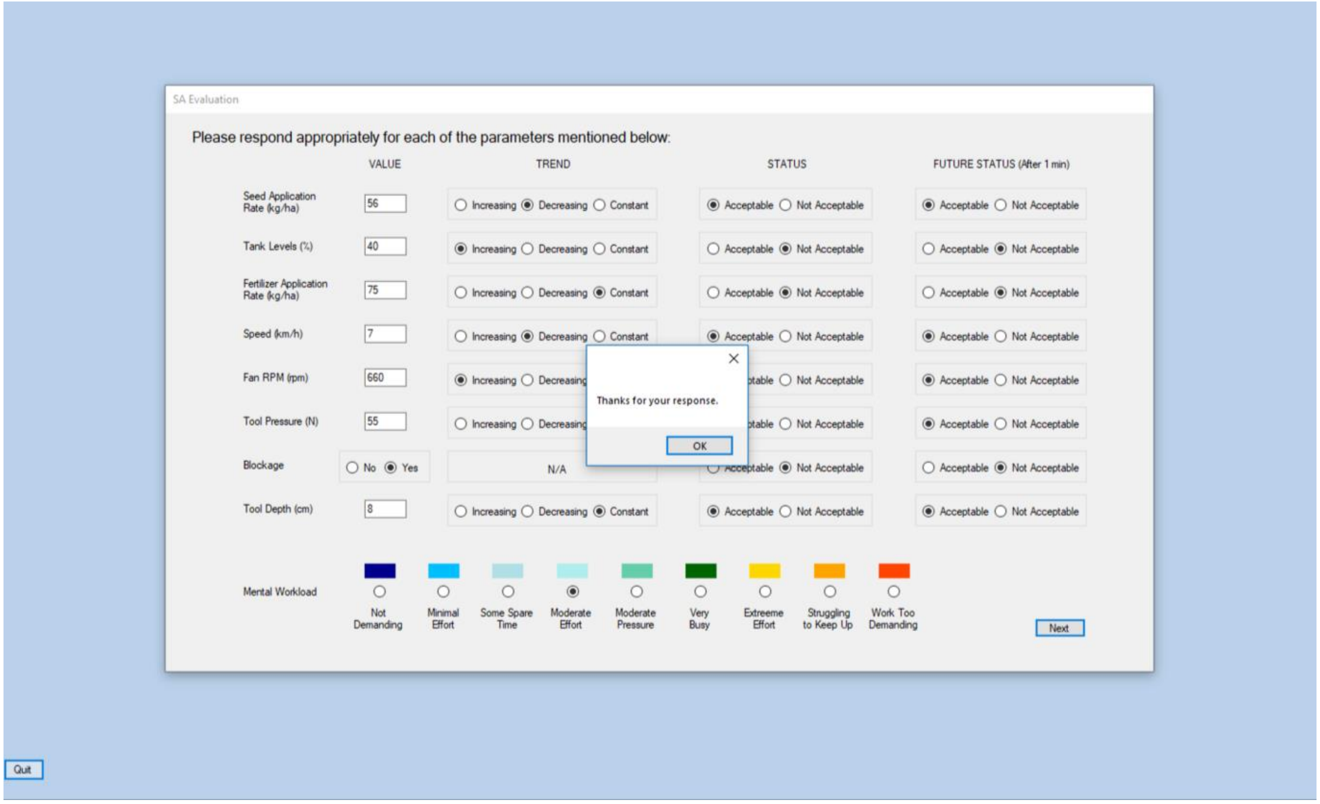
E.2 SCREENSHOTS OF PHASE 2 OF THE EXPERIMENTS

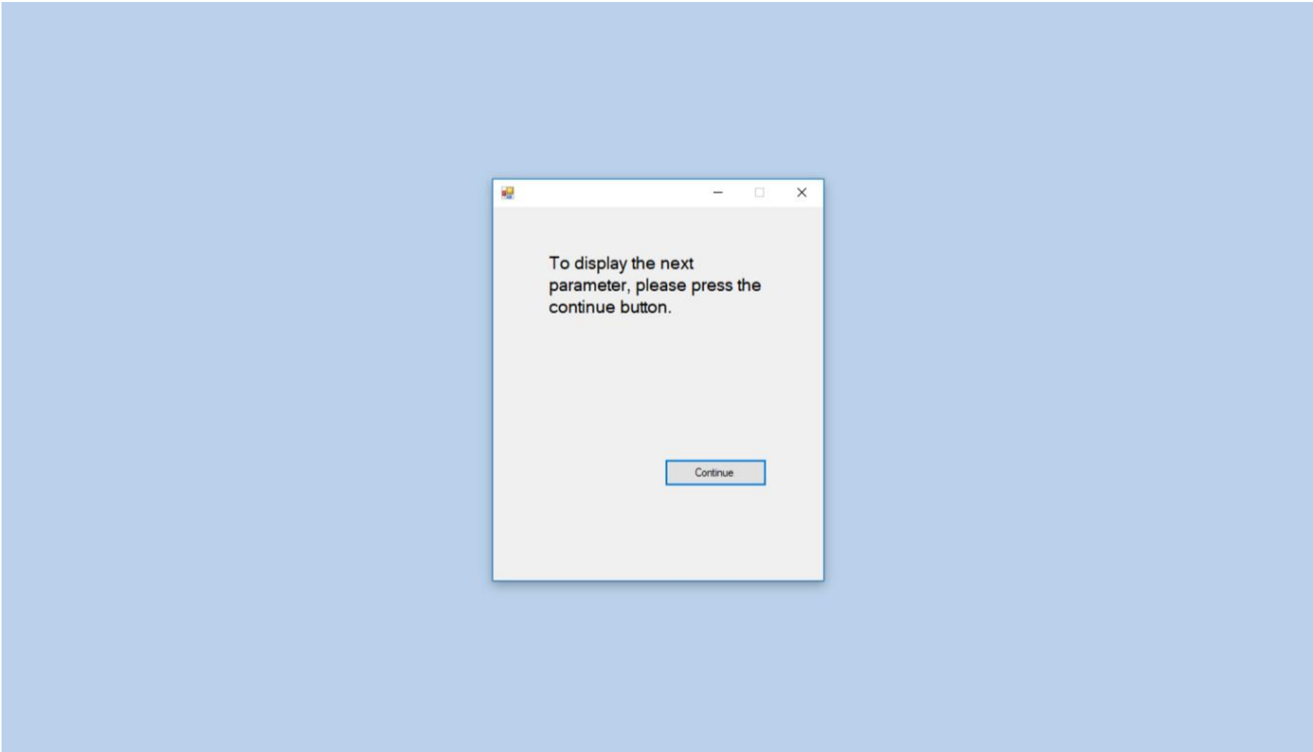





Seed Application Rate
43
kg/ha
Tanks Level
26 %
Fertilizer Application Rate
106
kg/ha
Speed
9.7
km/h
Guidance
Fan RPM
539
rpm
Tool Depth Left
Tool Depth Right
Tool Pressure
73
N
Blockage
9.7
cm
9.7
cm
Quit

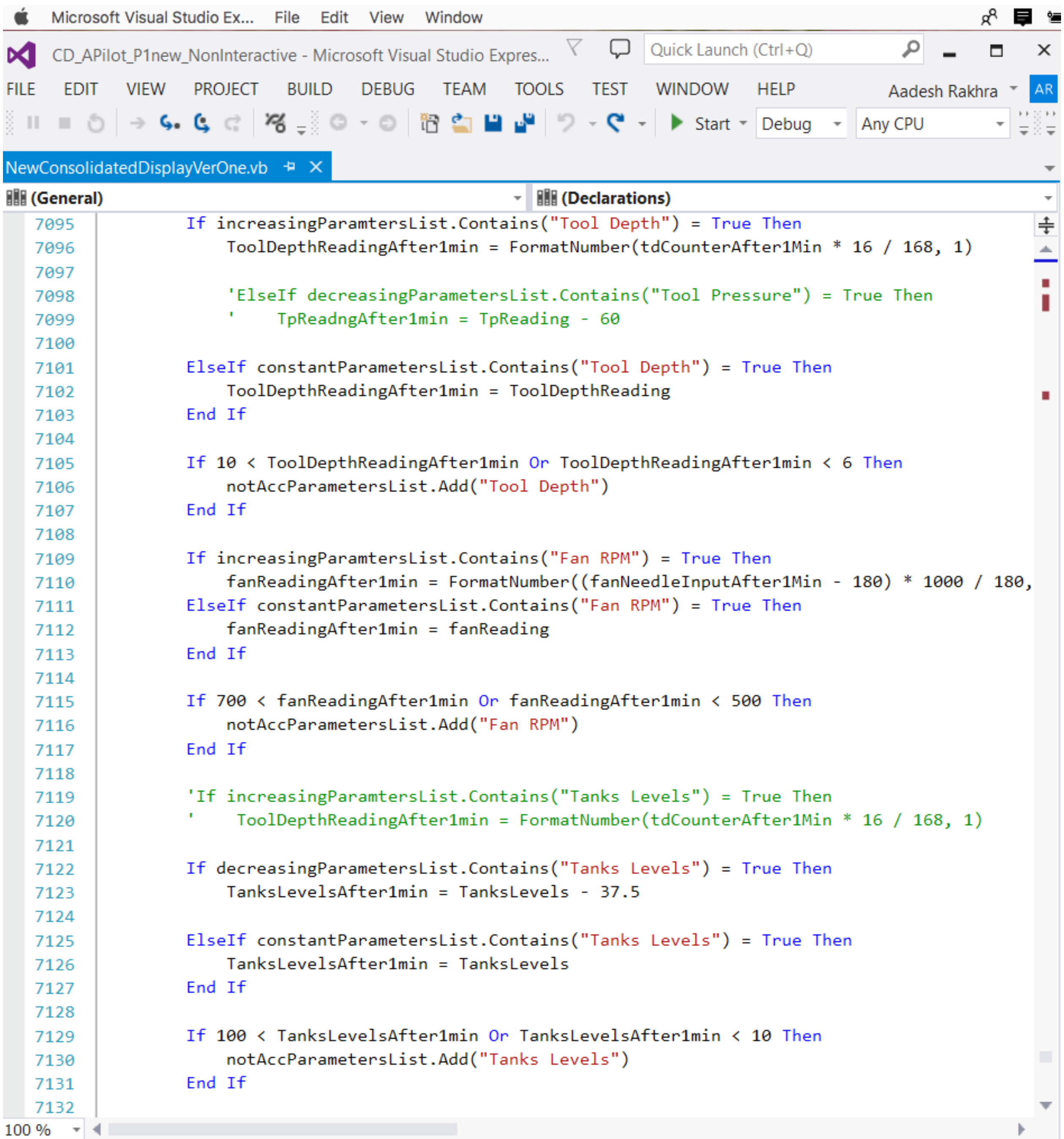

SA Evaluation
Please respond appropriately for each of the parameters mentioned below:
VALUE
TREND
STATUS
FUTURE STATUS (After 1 min)
Seed Application Rate (kg/ha)
Tank Levels (%)
Fertilizer Application Rate (kg/ha)
Speed (km/h)
Fan RPM (rpm)
Tool Pressure (N)
Blockage
Tool Depth (cm)
Increasing Decreasing Constant
Acceptable Not Acceptable
No Yes
N/A
Training Session
Please fill all the required fields.
OK
Mental Workload
Not Demanding
Minimal Effort
Some Spare Time
Moderate Effort
Moderate Pressure
Very Busy
Extreme Effort
Struggling to Keep Up
Work Too Demanding
Next
Quit


SA Evaluation
Please respond appropriately for each of the parameters mentioned below:
VALUE
TREND
STATUS
FUTURE STATUS (After 1 min)
Seed Application Rate (kg/ha)
Tank Levels (%)
Fertilizer Application Rate (kg/ha)
Speed (km/h)
Fan RPM (rpm)
Tool Pressure (N)
Blockage
Tool Depth (cm)
Increasing
Decreasing
Constant
Acceptable
Not Acceptable
No
Yes
N/A
Thanks for your response.
OK
Mental Workload
Not Demanding
Minimal Effort
Some Spare Time
Moderate Effort
Moderate Pressure
Very Busy
Extreeme Effort
Struggling to Keep Up
Work Too Demanding
Next
Quit


To display the next parameter, please press the continue button.
Continue

Appendix F

F.1 SCREENSHOT OF THE CODING ENVIRONMENT (MICROSOFT VISUAL STUDIO EXPRESS), WHICH WAS USED TO DEVELOP INTERFACE SIMULATIONS FOR THE TRACTOR AIR SEEDER.



F.2 EXAMPLE PROGRAM USED TO RUN INTERFACE SIMULATION FOR DESIGNING, TESTING AND EVALUATING THE AIR SEEDER INTERFACE

This program is used to run one of the interfaces (User-Centered Design Version 1) and is a part of the larger complete application used during the testing and evaluation of the user interface during the second phase of the experiments.

```
Imports System.Text

Imports System.IO

Public Class NewConsolidatedDisplayVerOne

    Public Property formClosetime As Integer

    Public Property getCounterTanksNew As Integer

    Dim RandomTanksNew As Integer

    Dim seedRateSequenceList As New List(Of Integer)

    Dim xStartTanksNew As Integer = 65

    '   Dim yStart As Integer = 149

    Dim yStartTanksNew As Integer = 160

    Dim counterTanksNew As Integer

    Dim counter2TanksNew As Integer 'for tankbase reduction

    Dim gTanksNew As Graphics

    ' responseTime related Variables

    Dim resTimeSAR As Integer

    Dim resTimeFAR As Integer

    Dim resTimeTanks As Integer

    Dim resTimeTp As Integer

    Dim resTimeSpeed As Integer

    Dim resTimeGuidance As Integer

    Dim resTimeFanRPM As Integer

    Dim resTimeToolDepth As Integer

    Dim resTimeBlockage As Integer

    ' newChanes
```

```
    Dim counterIdleSar As Integer

    Dim counterIdleTanks As Integer

    Dim counterIdleFar As Integer

    Dim counterIdleSpeed As Integer

    Dim counterIdleFan As Integer

    Dim counterIdleToolPressure As Integer

    Dim counterIdleToolDepth As Integer

    Dim counterIdleGlobal As Integer

    ' newChanges

    Dim intList3 As New List(Of Integer)

    Dim intList3RandomRate As New List(Of Integer)

    Dim Random As Integer

    Dim RandomRn As Integer

    Dim intListSingle As New List(Of Integer)

    Dim intListRandomRateSingle As New List(Of Integer)

    Dim RandomSingle As Integer

    Dim RandomRnSingle As Integer

    Private Sub NewConsolidatedDisplayForm_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

        ' newChanges

        timerIdleGlobal.Enabled = True

        timerIdleSar.Enabled = True

        timerIdleTanks.Enabled = True

        timerIdleFar.Enabled = True

        timerIdleSpeed.Enabled = True

        timerIdleFan.Enabled = True

        timerIdleTp.Enabled = True

        timerIdleToolDepth.Enabled = True

        '  start3randomParametersAfterFormLoad()

        '  randomNumber()

        Me.TopMost = True

        timerFormClose.Enabled = True
```

```
        timerResponseTime.Enabled = True
        timerTanksNew.Enabled = False
        elementsLocationSizeTanks()
        timerSARnew.Enabled = False
        elementsLocationSizeSAR()
        timerFAR.Enabled = False
        elementsLocationSizeFAR()
        timerSpeedNew.Enabled = False
        elementsLocationSizeSpeedNew()
        timerFanRPM.Enabled = False
        elementsLocationSizeFan()
        timerTp.Enabled = False
        elementsLocationSizeTp()
        timerToolDepth.Enabled = False
        elementsLocationSizeToolDepth()
        drawTpScale()
        blockageNewBackDots()
        timerGB.Enabled = True
        drawGuidanceBar()
        elementsLocationSizeGuidanceBar()
        timerBlockage.Enabled = True
        ' captionLocationBlocakge()
        elementsSizeLocaitonBlocakge()
        'storeSequeceOfRates()
        timerGlobalResetCall.Enabled = True
    End Sub
    Private Sub randomNumber()
        Dim upperbound As Integer = 6
        Dim lowerbound As Integer = 0
        Dim upperboundRn As Integer = 3
        Dim lowerboundRn As Integer = 1
```

```
        Do Until intList3.Count = 3
againGenerate: Randomize()
            Random = CInt(Math.Floor((upperbound - lowerbound + 1) * Rnd())) + lowerbound
            If intList3.Count = 0 Then
                intList3.Add(Random)
                GoTo againGenerate
            Else
                For i = 0 To intList3.Count - 1
                    If intList3(i) = Random Then
                        GoTo donothing
                    End If
                Next
                intList3.Add(Random)
donothing:
            End If
        Loop
        Do Until intList3RandomRate.Count = 3
againGenerateRn: Randomize()
            RandomRn = CInt(Math.Floor((upperboundRn - lowerboundRn + 1) * Rnd())) +
lowerboundRn
            If intList3RandomRate.Count = 0 Then
                intList3RandomRate.Add(RandomRn)
                GoTo againGenerateRn
            Else
                For i = 0 To intList3RandomRate.Count - 1
                    If intList3RandomRate(i) = RandomRn Then
                        GoTo donothingRn
                    End If
                Next
                intList3RandomRate.Add(RandomRn)
donothingRn:
            End If
```

```
        Loop
    End Sub
    Private Sub randomNumberSingle()
        If intListSingle.Count > 0 Then
            intListSingle.Clear()
        End If
        If intListRandomRateSingle.Count > 0 Then
            intListRandomRateSingle.Clear()
        End If
        Dim upperbound As Integer = 6
        Dim lowerbound As Integer = 0
        Dim upperboundRn As Integer = 3
        Dim lowerboundRn As Integer = 1
        Dim allParameterTimerNames() As Timer = {timerTanksNew, timerSARnew, timerFAR,
timerSpeedNew, _
                                                 timerFanRPM, timerTp, timerToolDepth}
randomizeAgain:
        Randomize()
        RandomSingle = CInt(Math.Floor((upperbound - lowerbound + 1) * Rnd())) + lowerbound
        For i = 0 To intList3.Count - 1
            If intList3(i) = RandomSingle Then
                GoTo randomizeAgain
            End If
        Next
        intList3.Add(RandomSingle)
        intListSingle.Add(RandomSingle)
        Randomize()
        RandomRnSingle = CInt(Math.Floor((upperboundRn - lowerboundRn + 1) * Rnd())) +
lowerboundRn
        intListRandomRateSingle.Add(RandomRnSingle)
    End Sub
    ' This program start 3 parameters randomly
```

```
Private Sub start3randomParametersAfterFormLoad()
    randomNumber()
    If intList3(0) = 0 Then
        timerTanksNew.Enabled = True
        rnTanks = intList3RandomRate(0)
    ElseIf intList3(0) = 1 Then
        timerSARnew.Enabled = True
        rnSAR = intList3RandomRate(0)
    ElseIf intList3(0) = 2 Then
        timerFAR.Enabled = True
        rnFAR = intList3RandomRate(0)
    ElseIf intList3(0) = 3 Then
        timerSpeedNew.Enabled = True
        rnSpeed = intList3RandomRate(0)
    ElseIf intList3(0) = 4 Then
        timerFanRPM.Enabled = True
        rnFanRpm = intList3RandomRate(0)
    ElseIf intList3(0) = 5 Then
        timerTp.Enabled = True
        rnToolPressure = intList3RandomRate(0)
    ElseIf intList3(0) = 6 Then
        timerToolDepth.Enabled = True
        rnToolDepth = intList3RandomRate(0)
    End If
    If intList3(1) = 0 Then
        timerTanksNew.Enabled = True
        rnTanks = intList3RandomRate(1)
    ElseIf intList3(1) = 1 Then
        timerSARnew.Enabled = True
        rnSAR = intList3RandomRate(1)
    ElseIf intList3(1) = 2 Then
```

```
        timerFAR.Enabled = True
        rnFAR = intList3RandomRate(1)
    ElseIf intList3(1) = 3 Then
        timerSpeedNew.Enabled = True
        rnSpeed = intList3RandomRate(1)
    ElseIf intList3(1) = 4 Then
        timerFanRPM.Enabled = True
        rnFanRpm = intList3RandomRate(1)
    ElseIf intList3(1) = 5 Then
        timerTp.Enabled = True
        rnToolPressure = intList3RandomRate(1)
    ElseIf intList3(1) = 6 Then
        timerToolDepth.Enabled = True
        rnToolDepth = intList3RandomRate(1)
    End If
    If intList3(2) = 0 Then
        timerTanksNew.Enabled = True
        rnTanks = intList3RandomRate(2)
    ElseIf intList3(2) = 1 Then
        timerSARnew.Enabled = True
        rnSAR = intList3RandomRate(2)
    ElseIf intList3(2) = 2 Then
        timerFAR.Enabled = True
        rnFAR = intList3RandomRate(2)
    ElseIf intList3(2) = 3 Then
        timerSpeedNew.Enabled = True
        rnSpeed = intList3RandomRate(2)
    ElseIf intList3(2) = 4 Then
        timerFanRPM.Enabled = True
        rnFanRpm = intList3RandomRate(2)
    ElseIf intList3(2) = 5 Then
```

```
            timerTp.Enabled = True

            rnToolPressure = intList3RandomRate(2)

        ElseIf intList3(2) = 6 Then

            timerToolDepth.Enabled = True

            rnToolDepth = intList3RandomRate(2)

        End If

        Dim allIdleTimers() As Timer = {timerIdleTanks, timerIdleSar, timerIdleFar,
timerIdleSpeed, timerIdleFan, _

                                        timerIdleTp, timerIdleToolDepth}

        allIdleTimers(intList3(0)).Enabled = False

        allIdleTimers(intList3(1)).Enabled = False

        allIdleTimers(intList3(2)).Enabled = False

    End Sub

    Private Sub start1randomParameterAfterResest()

        randomNumberSingle()

        Dim allIdleTimers() As Timer = {timerIdleTanks, timerIdleSar, timerIdleFar,
timerIdleSpeed, timerIdleFan, _

                                      timerIdleTp, timerIdleToolDepth}

        allIdleTimers(intListSingle(0)).Enabled = False

        Dim allParameterTimerNames() As Timer = {timerTanksNew, timerSARnew, timerFAR,
timerSpeedNew, _

                                                 timerFanRPM, timerTp, timerToolDepth}

        allParameterTimerNames(intListSingle(0)).Enabled = False

        If intListSingle(0) = 0 Then

            timerTanksNew.Enabled = True

            rnTanks = intListRandomRateSingle(0)

        ElseIf intListSingle(0) = 1 Then

            timerSARnew.Enabled = True

            rnSAR = intListRandomRateSingle(0)

        ElseIf intListSingle(0) = 2 Then

            timerFAR.Enabled = True

            rnFAR = intListRandomRateSingle(0)

        ElseIf intListSingle(0) = 3 Then
```

```
            timerSpeedNew.Enabled = True

            rnSpeed = intListRandomRateSingle(0)

        ElseIf intListSingle(0) = 4 Then

            timerFanRPM.Enabled = True

            rnFanRpm = intListRandomRateSingle(0)

        ElseIf intListSingle(0) = 5 Then

            timerTp.Enabled = True

            rnToolPressure = intListRandomRateSingle(0)

        ElseIf intListSingle(0) = 6 Then

            timerToolDepth.Enabled = True

            rnToolDepth = intListRandomRateSingle(0)

        End If

        Label2.Text = "timer:" + intListSingle(0).ToString + "Rate:" +
intListRandomRateSingle(0).ToString

        If enabledTimersCount() > 3 Then

            If intListSingle(0) = 0 Then

                timerTanksNew.Enabled = False

                timerIdleTanks.Enabled = True

            ElseIf intListSingle(0) = 1 Then

                timerSARnew.Enabled = False

                timerIdleSar.Enabled = True

            ElseIf intListSingle(0) = 2 Then

                timerFAR.Enabled = False

                timerIdleFar.Enabled = True

            ElseIf intListSingle(0) = 3 Then

                timerSpeedNew.Enabled = False

                timerIdleSpeed.Enabled = True

            ElseIf intListSingle(0) = 4 Then

                timerFanRPM.Enabled = False

                timerIdleFan.Enabled = True

            ElseIf intListSingle(0) = 5 Then

                timerTp.Enabled = False
```

```
                timerIdleTp.Enabled = True
            ElseIf intListSingle(0) = 6 Then
                timerToolDepth.Enabled = False
                timerIdleToolDepth.Enabled = True
            End If
        End If

    End Sub
    Function enabledTimersCount() As Integer
        Dim enabledTimersCounter As Integer = 0
        If timerSARnew.Enabled = True Then
            enabledTimersCounter = enabledTimersCounter + 1
        End If
        If timerFAR.Enabled = True Then
            enabledTimersCounter = enabledTimersCounter + 1
        End If
        If timerTanksNew.Enabled = True Then
            enabledTimersCounter = enabledTimersCounter + 1
        End If
        If timerSpeedNew.Enabled = True Then
            enabledTimersCounter = enabledTimersCounter + 1
        End If
        If timerFanRPM.Enabled = True Then
            enabledTimersCounter = enabledTimersCounter + 1
        End If
        If timerTp.Enabled = True Then
            enabledTimersCounter = enabledTimersCounter + 1
        End If
        If timerToolDepth.Enabled = True Then
            enabledTimersCounter = enabledTimersCounter + 1
        End If
```

```
        Return enabledTimersCounter
    End Function
    Private Sub idleSar()
        counterIdleSar = counterIdleSar + 1
        stopTrendSAR()
        okaySeeds()
        progressBarAllInOneSAR(35)
        lbSARreading.Text = FormatNumber(35, 0)
    End Sub
    Private Sub idleTanks()
        '  counterIdleTanks = counterIdleTanks + 1
        gTanksNew = Me.pbTanksNew.CreateGraphics
        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        stopTrendTanks()
        tankProgress(0, 47)
        tanksOutline()
    End Sub
    Private Sub idleFar()
        counterIdleFar = counterIdleFar + 1
        stopTrendFAR()
        okaySeedsFAR()
        progressBarAllInOneFAR(35)
        lbFARnewReading.Text = FormatNumber(70, 0)
    End Sub
    Private Sub idleSpeed()
        counterIdleSpeed = counterIdleSpeed + 1
        captionSpeed()
        drawDialNumberingSpeedNew()
        needleSpeedNew(270)
        gSpeedNew.Dispose()
```

```
End Sub
Private Sub idleFan()
    captionFan()
    needleFan(270)
End Sub
Private Sub idleToolPressure()
    captionTp()
    drawTpScale()
    stopTrendTp()
    okayPressureArrowsTp()
    progressBarAllInOneTp(60)
    lbTPreading.Text = Format(60, 0)
End Sub
Private Sub idleToolDepth()
    labeltoolDepth()
    toolDepthMovingTool(90)
    drawTanksScaleToolDepth()
End Sub
''' <summary>
''' newChanges
''' '''
''' </summary>
''' <remarks></remarks>
'''
Dim counterGlobal As Integer
Private Sub resetAllToIdle()
    pbSeedAppRateNew_Click(Nothing, Nothing)
    pbFAR_Click(Nothing, Nothing)
    pbTanksNew_Click(Nothing, Nothing)
    pbSpeedNew_Click(Nothing, Nothing)
    pbFanRPM_Click(Nothing, Nothing)
```

```
        pbTp_Click(Nothing, Nothing)

        pbToolDepth_Click(Nothing, Nothing)

        pbField_Click(Nothing, Nothing)

        pbBlockage_Click(Nothing, Nothing)

    End Sub

    Private Sub timerGlobalResetCall_Tick(sender As Object, e As EventArgs) Handles
timerGlobalResetCall.Tick

        counterGlobal = counterGlobal + 1

        'If counterGlobal > 1 Then

        '    Cursor.Hide()

        'End If

        Label1.Text = counterGlobal

        If counterGlobal = 1 Then

            Cursor.Hide()

        ElseIf counterGlobal = formClosetime - 1 Then

            Me.Cursor = New Cursor(Cursor.Current.Handle)

            Cursor.Position = New Point(Me.Top - 25, Me.Top - 25)

            Cursor.Show()

            Me.Cursor = Cursors.Default

        End If

        Label3.Text = counterGlobal

        If counterGlobal = 50 Then

            resetAllToIdle()

        ElseIf counterGlobal = 100 Then

            resetAllToIdle()

        ElseIf counterGlobal = 150 Then

            resetAllToIdle()

        ElseIf counterGlobal = 200 Then

            resetAllToIdle()

        ElseIf counterGlobal = 250 Then

            resetAllToIdle()

        End If
```

```
    End Sub

#Region "GuidanceBar"

    Private Sub elementsLocationSizeGuidanceBar()

        pbGBField.Left = pbTanksNew.Left + pbTanksNew.Width / 2 - pbGBField.Width / 2

        pbGuidanceBar.Left = pbGBField.Left + pbGBField.Width / 2 - pbGuidanceBar.Width / 2

        pbGuidanceBar.Top = pbGBField.Top

        pbTractorGuidance.Left = pbGuidanceBar.Left + pbGuidanceBar.Width / 2 -
pbTractorGuidance.Width / 2

        pbTractorGuidance.Top = pbGuidanceBar.Top + pbGuidanceBar.Height + 40

        lbGuidanceName.Top = pbGBField.Top - lbGuidanceName.Height - 5

        lbGuidanceName.Left = pbGBField.Left

    End Sub

    Dim counterGb As Integer

    Private Sub timerGB_Tick(sender As Object, e As EventArgs) Handles timerGB.Tick

        counterGb = counterGb + 1

        If counterGb = 1 Then

            drawGuidanceBar()

            gbAniGreenMiddle()

            pbTractorGuidance.Image = My.Resources.TractorfrontLight2

        ElseIf counterGb = 8 Then

            gbAniYellowLeft1()

            pbTractorGuidance.Image = My.Resources.frontLight15

        ElseIf counterGb = 16 Then

            gbAniYellowLeft2()

            pbTractorGuidance.Image = My.Resources.frontLight30

        ElseIf counterGb = 24 Then

            gbAniRedLeft1()

            pbTractorGuidance.Image = My.Resources.frontLight30

        ElseIf counterGb = 32 Then

            gbAniRedLeft2()

            pbTractorGuidance.Image = My.Resources.frontLight45
```

```
        ElseIf counterGb >= 40 Then

            gbAniRedLeft3()

            pbTractorGuidance.Image = My.Resources.frontLight45

            '  pbGuidanceBar.BackColor = Color.Red

        End If

    End Sub

    Private Sub guidanceBarReset()

        drawGuidanceBar()

        gbAniGreenMiddle()

        pbTractorGuidance.Image = My.Resources.TractorfrontLight2

    End Sub

    Private Sub drawGuidanceBar()

        Dim gGuidaceBar As Graphics

        gGuidaceBar = Me.pbGuidanceBar.CreateGraphics

        gGuidaceBar.Clear(Color.Gray)

        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        For i = 5 To 500 Step 35

            gGuidaceBar.FillEllipse(Brushes.LightGray, i, 15, 24, 10)

        Next

    End Sub

    Private Sub gbAniGreenMiddle()

        Dim gGuidaceBar As Graphics

        gGuidaceBar = Me.pbGuidanceBar.CreateGraphics

        Dim x, y As Integer

        x = 180

        y = 20

        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        Dim myDiamond() As Point = {New Point(x, y), New Point(x + 12, y - 15), New Point(x +
24, y), _

                                    New Point(x + 12, y + 15)}
```

```
        gGuidaceBar.FillPolygon(Brushes.YellowGreen, myDiamond)
    End Sub
    Private Sub gbAniYellowLeft1()
        Dim gGuidaceBar As Graphics
        gGuidaceBar = Me.pbGuidanceBar.CreateGraphics
        'clear and drew basic bar before yellow diamond
        gGuidaceBar.Clear(Color.Gray)
        drawGuidanceBar()
        Dim x, y As Integer
        x = 145
        y = 20
        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        Dim myDiamond() As Point = {New Point(x, y), New Point(x + 12, y - 15), New Point(x +
24, y), _
                                New Point(x + 12, y + 15)}
        gGuidaceBar.FillPolygon(Brushes.Gold, myDiamond)
    End Sub
    Private Sub gbAniYellowLeft2()
        Dim gGuidaceBar As Graphics
        gGuidaceBar = Me.pbGuidanceBar.CreateGraphics
        'clear and drew basic bar before yellow diamond
        gGuidaceBar.Clear(Color.Gray)
        drawGuidanceBar()
        Dim x, y As Integer
        x = 110
        y = 20
        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        Dim myDiamond() As Point = {New Point(x, y), New Point(x + 12, y - 15), New Point(x +
24, y), _
                                New Point(x + 12, y + 15)}
```

```
        gGuidaceBar.FillPolygon(Brushes.Gold, myDiamond)
    End Sub
    Private Sub gbAniRedLeft1()
        Dim gGuidaceBar As Graphics
        gGuidaceBar = Me.pbGuidanceBar.CreateGraphics
        'clear and drew basic bar before yellow diamond
        gGuidaceBar.Clear(Color.Gray)
        drawGuidanceBar()
        Dim x, y As Integer
        x = 75
        y = 20
        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        Dim myDiamond() As Point = {New Point(x, y), New Point(x + 12, y - 15), New Point(x +
24, y), _
                                   New Point(x + 12, y + 15)}
        gGuidaceBar.FillPolygon(Brushes.OrangeRed, myDiamond)
    End Sub
    Private Sub gbAniRedLeft2()
        Dim gGuidaceBar As Graphics
        gGuidaceBar = Me.pbGuidanceBar.CreateGraphics
        'clear and drew basic bar before yellow diamond
        gGuidaceBar.Clear(Color.Gray)
        drawGuidanceBar()
        Dim x, y As Integer
        x = 40
        y = 20
        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        Dim myDiamond() As Point = {New Point(x, y), New Point(x + 12, y - 15), New Point(x +
24, y), _
                                   New Point(x + 12, y + 15)}
```

```
        gGuidaceBar.FillPolygon(Brushes.OrangeRed, myDiamond)

    End Sub

    Private Sub gbAniRedLeft3()

        Dim gGuidaceBar As Graphics

        gGuidaceBar = Me.pbGuidanceBar.CreateGraphics

        'clear and drew basic bar before yellow diamond

        gGuidaceBar.Clear(Color.Gray)

        drawGuidanceBar()

        Dim x, y As Integer

        x = 5

        y = 20

        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

        gGuidaceBar.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        Dim myDiamond() As Point = {New Point(x, y), New Point(x + 12, y - 15), New Point(x +
24, y), _

                                    New Point(x + 12, y + 15)}

        gGuidaceBar.FillPolygon(Brushes.OrangeRed, myDiamond)

    End Sub

#End Region

#Region "Tanks"

    Dim tankPen As New Pen(Brushes.Black, 2)

    Dim wallsAllTanks() As Point = {New Point(xStartTanksNew, yStartTanksNew - 144), _

                                    New Point(xStartTanksNew, yStartTanksNew + 17), _

                                    New Point(xStartTanksNew + 99, yStartTanksNew + 17), _

                                    New Point(xStartTanksNew + 99, yStartTanksNew - 144), _

                                    New Point(xStartTanksNew + 106, yStartTanksNew - 144), _

                                    New Point(xStartTanksNew + 106, yStartTanksNew + 17), _

                                    New Point(xStartTanksNew + 255, yStartTanksNew - 144), _

                                    New Point(xStartTanksNew + 255, yStartTanksNew + 17), _

                                    New Point(xStartTanksNew + 262, yStartTanksNew - 144), _

                                    New Point(xStartTanksNew + 262, yStartTanksNew + 17), _

                                    New Point(xStartTanksNew + 360, yStartTanksNew - 144), _
```

```
                                        New Point(xStartTanksNew + 360, yStartTanksNew + 17), _

                                        New Point(xStartTanksNew, yStartTanksNew + 17), _

                                        New Point(xStartTanksNew + 99, yStartTanksNew + 17), _

                                        New Point(xStartTanksNew + 106, yStartTanksNew + 17), _

                                        New Point(xStartTanksNew + 255, yStartTanksNew + 17), _

                                        New Point(xStartTanksNew + 262, yStartTanksNew + 17), _

                                        New Point(xStartTanksNew + 360, yStartTanksNew + 17)}

    Private Sub elementsLocationSizeTanks()

        pbTanksNew.Left = Me.Width / 2 - pbTanksNew.Width / 2

        ' lbTanksNewReading.Left = xStartTanksNew + 106 + 115

        lbTanksNewReading.Top = pbTanksNew.Top + 70

        lbTanksNewReading.Left = pbTanksNew.Left + xStartTanksNew + 106 + 75 -
lbTanksNewReading.Width / 2

        'pbTanksNew.Width / 2

        'pbTanksNewName.Top = pbTanksNew.Top

        'pbTanksNewName.Left = pbTanksNew.Left + pbTanksNew.Width / 2

        pbTanksNewName.Top = pbTanksNew.Top + 200

        pbTanksNewName.Left = pbTanksNew.Left + pbTanksNew.Width / 2

        pbTrendTanks.Top = lbTanksNewReading.Top + 25

        pbTrendTanks.Left = lbTanksNewReading.Left + lbTanksNewReading.Width / 2 -
pbTrendTanks.Width / 2

        lbTanksNewCaption.Top = pbTanksNew.Top - lbTanksNewCaption.Height - 5

        lbTanksNewCaption.Left = pbTanksNew.Left

    End Sub

    ' change1

    Dim rnTanks As Integer

    ' Dim rnTanks As Integer = 1

    Private Sub TimerTanks_Tick(sender As Object, e As EventArgs) Handles timerTanksNew.Tick

        counterTanksNew = counterTanksNew + 1

        If rnTanks = 1 Then

            TanksRate1()

        ElseIf rnTanks = 2 Then
```

```
        TanksRate2()
    ElseIf rnTanks = 3 Then
        TanksRate3()
    End If
End Sub
Private Sub nameLabelTanks()
    Dim formGraphics As System.Drawing.Graphics = Me.pbTanksNewName.CreateGraphics()
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim drawString As String = "Tank Levels"
    Dim drawString2a As String = "( "
    Dim drawString2 As String = "100-10"
    Dim drawString3 As String = " % )"
    Dim drawFont As New System.Drawing.Font("Arial", 10)
    Dim drawFont2 As New System.Drawing.Font("Arial", 11, FontStyle.Bold)
    Dim drawBrush As New  _
        System.Drawing.SolidBrush(System.Drawing.Color.Black)
    Dim drawFormat As New System.Drawing.StringFormat
    formGraphics.DrawString(drawString, drawFont, drawBrush, _
    5, 0, drawFormat)
    formGraphics.DrawString(drawString2a, drawFont, drawBrush, _
    0, 15, drawFormat)
    formGraphics.DrawString(drawString2, drawFont2, drawBrush, _
    10, 15, drawFormat)
    formGraphics.DrawString(drawString3, drawFont, drawBrush, _
    55, 15, drawFormat)
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
End Sub
Private Sub tanksOutline()
    gTanksNew = Me.pbTanksNew.CreateGraphics
```

```
        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        scaleDrawTanks()
        nameLabelTanks()
        For index = 0 To 16 Step 2
            gTanksNew.DrawLine(tankPen, wallsAllTanks(index), wallsAllTanks(index + 1))
        Next
    End Sub
    Private Sub scaleDrawTanks()
        Dim xScale As Integer = 35
        '  g = Me.PictureBox1.CreateGraphics
        gTanksNew.FillRectangle(Brushes.YellowGreen, xScale, yStartTanksNew - 144, 18, 160)
        gTanksNew.FillRectangle(Brushes.IndianRed, xScale, yStartTanksNew - 144 + 145, 18, 16)
        For y = yStartTanksNew - 144 + 1 To yStartTanksNew - 144 + 176 Step 16
            gTanksNew.DrawLine(Pens.Black, xScale, y, xScale + 16, y)
        Next
        scaleNumberingTanks()
    End Sub
    Private Sub scaleNumberingTanks()
        Dim formGraphics As System.Drawing.Graphics = Me.pbTanksNew.CreateGraphics()
        Dim drawString As String = ""
        Dim drawFont As New System.Drawing.Font("Arial", 8, FontStyle.Bold)
        Dim drawBrush As New  _
            System.Drawing.SolidBrush(System.Drawing.Color.Black)
        Dim x As Single = 25
        Dim y As Single
        Dim drawFormat As New System.Drawing.StringFormat
        drawFormat.FormatFlags = StringFormatFlags.LineLimit
        drawFormat.LineAlignment = StringAlignment.Center
        drawFormat.Alignment = StringAlignment.Far
        Dim arrayDrawStirng() As String = {"100", "", "", "", "", "50", "", "", "", "10", "0"}
```

```
        Dim i As Integer = 0

        For y = 17 To 177 Step 16

            drawString = arrayDrawStirng(i)

            formGraphics.DrawString(drawString, drawFont, drawBrush, _

            x, y, drawFormat)

            i = i + 1

        Next

    End Sub


    Private Sub tankProgress(ByVal newCounter As Integer, ByVal startPercentage As Integer)

        counter2TanksNew = counter2TanksNew + 1.6

        Dim yProgresStart As Integer = (100 - startPercentage) * 1.6 + 15 + newCounter

        Dim yHeight As Integer = yStartTanksNew + 16 - yProgresStart + 1

        gTanksNew = Me.pbTanksNew.CreateGraphics

        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        gTanksNew.Clear(SystemColors.Control)

        Dim lbReading As Integer = startPercentage - newCounter * 10 / 16

        lbTanksNewReading.Text = FormatNumber(startPercentage - newCounter * 10 / 16, 0) & "
%"

        If lbReading >= 10 Then

            'g.FillPolygon(Brushes.DarkSeaGreen, tankBaseLeft)

            'g.FillPolygon(Brushes.DarkSeaGreen, tankBaseMiddle)

            'g.FillPolygon(Brushes.DarkSeaGreen, tankBaseRight)

            gTanksNew.FillRectangle(Brushes.DarkSeaGreen, xStartTanksNew, yProgresStart, 99,
yHeight)

            gTanksNew.FillRectangle(Brushes.DarkSeaGreen, xStartTanksNew + 106, yProgresStart,
149, yHeight)

            gTanksNew.FillRectangle(Brushes.DarkSeaGreen, xStartTanksNew + 262, yProgresStart,
99, yHeight)

        Else

            'g.FillPolygon(Brushes.DarkSeaGreen, tankBaseLeft)

            'g.FillPolygon(Brushes.DarkSeaGreen, tankBaseMiddle)
```

```
            'g.FillPolygon(Brushes.DarkSeaGreen, tankBaseRight)
            gTanksNew.FillRectangle(Brushes.Crimson, xStartTanksNew, yProgresStart, 99,
yHeight)
            gTanksNew.FillRectangle(Brushes.Crimson, xStartTanksNew + 106, yProgresStart, 149,
yHeight)
            gTanksNew.FillRectangle(Brushes.Crimson, xStartTanksNew + 262, yProgresStart, 99,
yHeight)
        End If
        'lbTpReading.Text = FormatNumber(startPercentage - counter * 10 / 16, 0) & " %"
    End Sub
    Private Sub incTriTanks()
        'PicBoxArrowIndicator.Left = PictureBox1.Right - (30)
        'PicBoxArrowIndicator.Top = PictureBox1.Top + (60)
        'PicBoxArrowIndicator.Width = 18
        'PicBoxArrowIndicator.Height = 20
        gTanksNew = Me.pbTanksNew.CreateGraphics
        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        '   g.Clear(SystemColors.Control)
        ' PicBoxArrowIndicator.BackColor = Color.Silver
        Dim x As Integer = 95
        Dim y As Integer = 85
        Dim incTri() As Point = {New Point(x, y), _
                                 New Point(x, y + 16), _
                                 New Point(x + 10, y + 8)}
        gTanksNew.FillPolygon(Brushes.Goldenrod, incTri)
    End Sub
    ' Decreasing indicating triangle
    Private Sub decTriTanks()
        gTanksNew = Me.pbTrendTanks.CreateGraphics
        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        gTanksNew.Clear(SystemColors.Control)
```

```
        '  pbTrendTanks.BackColor = Color.White

        Dim x As Integer = 0

        Dim y As Integer = 0

        'g.FillEllipse(Brushes.Silver, x, y, w, h)

        Dim incTri() As Point = {New Point(x, y), _

                                 New Point(x + 7, y + 14), _

                                 New Point(x + 14, y)}

        gTanksNew.FillPolygon(Brushes.Goldenrod, incTri)

        pbTrendTanks.Visible = True

    End Sub

    ' stop of the trend , reading is not decreasing or increasing

    Private Sub stopTrendTanks()

        gTanksNew = Me.pbTrendTanks.CreateGraphics

        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

        gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        Dim x As Integer = 0

        Dim y As Integer = 0

        Dim w As Integer = 14

        Dim h As Integer = 14

        'g.FillEllipse(Brushes.Silver, x, y, w, h)

        gTanksNew.FillRectangle(Brushes.Goldenrod, x, y, w, h)

        pbTrendTanks.Visible = True

    End Sub

    Private Sub TanksReset()

        timerIdleTanks.Enabled = True

        counterIdleTanks = 0

        timerTanksNew.Enabled = False

        timerTanksReset.Enabled = False

        intList3.Remove(0) ' 0 represents the tankstimer, so should be removed to make the
list dynamic

        start1randomParameterAfterResest()

    End Sub
```

```
Private Sub TanksRate1()
    '  tanksOutline()
    gTanksNew = Me.pbTanksNew.CreateGraphics
    gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    If counterTanksNew <= 65 Then
        decTriTanks()
        tankProgress(counterTanksNew, 47)
    ElseIf counterTanksNew <= 300 Then
        stopTrendTanks()
        tankProgress(65, 47)
    Else

    End If
    tanksOutline()
End Sub
Private Sub TanksRate2()
    '    tanksOutline()
    gTanksNew = Me.pbTanksNew.CreateGraphics
    gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    If counterTanksNew <= 35 Then
        decTriTanks()
        tankProgress(counterTanksNew, 29)
    ElseIf counterTanksNew <= 300 Then
        stopTrendTanks()
        tankProgress(35, 29)
        '  lbTpReading.Text = "31 %"
    Else
        'gTanksNew.Dispose()
        '' Me.Dispose()
```

```
        '' Me.Visible = False

        '_3Queries.Visible = True

        'Me.Close()

        'Me.Dispose()

        'Exit Sub

    End If

    tanksOutline()

End Sub

Private Sub TanksRate3()

    ' tanksOutline()

    gTanksNew = Me.pbTanksNew.CreateGraphics

    gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

    gTanksNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

    If counterTanksNew <= 22 Then

        decTriTanks()

        tankProgress(counterTanksNew, 20)

    ElseIf counterTanksNew <= 300 Then

        stopTrendTanks()

        tankProgress(22, 20)

        '  lbTpReading.Text = "31 %"

    Else

        'gTanksNew.Dispose()

        '' Me.Dispose()

        ''  Me.Visible = False

        '_3Queries.Visible = True

        'Me.Close()

        'Me.Dispose()

        'Exit Sub

    End If

    tanksOutline()

End Sub
```

```
#End Region

#Region "Seed Application Rate"

    '######### seedAppRateNew

    Public Property getCounterSeedAppRateNew As Integer

    ' Dim Counter As Single

    Dim RandomSARnew As Integer

    Dim counterSARnew As Integer

    Dim gSARnew As Graphics

    '= Me.pbSeedAppRateNew.CreateGraphics

    Dim gSARtrend As Graphics

    Dim gTpTrend As Graphics

    '= Me.pbTrendSARnew.CreateGraphics

    Dim hSARnew As Integer

    ' Function for changing the color of the progress bar:

    Declare Function SendMessageSAR Lib "user32" Alias "SendMessageA" _

        (ByVal hwnd As Integer, ByVal wMsg As Integer, ByVal wParam As Integer, ByVal lParam
As Integer) As Integer

    Dim rnSAR As Integer

    ' Dim rnSAR As Integer = 1

    Private Sub TimerSAR_Tick(sender As Object, e As EventArgs) Handles timerSARnew.Tick

        captionSeedAppRate()

        drawSARScale()

        If rnSAR = 1 Then

            seedAppRateNew1()

        ElseIf rnSAR = 2 Then

            seedAppRateNew2()

        ElseIf rnSAR = 3 Then

            seedAppRateNew3()

        End If

    End Sub

    Private Sub elementsLocationSizeSAR()

        lbSARreading.Width = 36
```

```
        ' lblSeedAppRate.Left = PictureBox1.Left + (190 - lblSeedAppRate.Width) / 2 + 10
        lbSARreading.Left = pbSeedAppRateNew.Left + 100 - (lbSARreading.Width / 2)
        ' lbSARreading.Top = pbSeedAppRateNew.Top + 60
        lbSARreading.Top = pbSeedAppRateNew.Top + pbSeedAppRateNew.Height
        pb2SARnew.Left = pbSeedAppRateNew.Left + pbSeedAppRateNew.Width / 2 - pb2SARnew.Width
/ 2
        pb2SARnew.Top = pbSeedAppRateNew.Top + pbSeedAppRateNew.Height + 5
        lbSARname.Left = pbSeedAppRateNew.Left
        lbSARname.Top = pbSeedAppRateNew.Top - lbSARname.Height - 5
        lbSARunits.Left = lbSARreading.Left + lbSARreading.Width / 2 - lbSARunits.Width / 2
        lbSARunits.Top = lbSARreading.Top + lbSARreading.Height
    End Sub
    Private Sub draw1Funnel()
        Dim xFunnel As Integer = 40
        Dim yFunnel As Integer = 60
        gSARnew = Me.pbSeedAppRateNew.CreateGraphics
        gSARnew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gSARnew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        ' draw a top bar

        For xFunnel = 10 To 180 Step 40
            If xFunnel = 90 Then Continue For
            Dim funnel1() As Point = {New Point(xFunnel, yFunnel), _
                                        New Point(xFunnel + 9, yFunnel + 10), _
                                      New Point(xFunnel + 9, yFunnel + 20), _
                                      New Point(xFunnel + 11, yFunnel + 20), _
                                      New Point(xFunnel + 11, yFunnel + 10), _
                                      New Point(xFunnel + 20, yFunnel)}
            gSARnew.FillPolygon(Brushes.Gray, funnel1)
        Next
    End Sub
    'Draw seeds
```

```
Private Sub lowSeeds()
    Dim xSeed As Integer = 30
    Dim ySeed As Integer = 85
    gSARnew = Me.pbSeedAppRateNew.CreateGraphics
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    For xSeed = 19 To 180 Step 40
        If xSeed = 99 Then Continue For
        For ySeed = 85 To 99 Step 7
            gSARnew.FillEllipse(Brushes.Red, xSeed, ySeed, 2, 4)
            'g.DrawEllipse(Pens.Black, xSeed, ySeed, 2, 4)
        Next
    Next
End Sub
Private Sub okaySeeds()
    Dim xSeed As Integer = 30
    Dim ySeed As Integer = 85
    gSARnew = Me.pbSeedAppRateNew.CreateGraphics
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    For xSeed = 19 To 180 Step 40
        If xSeed = 99 Then Continue For
        For ySeed = 85 To 97 Step 7
            gSARnew.FillEllipse(Brushes.Green, xSeed, ySeed, 2, 4)
            If ySeed = 85 Then Continue For
            gSARnew.FillEllipse(Brushes.Green, xSeed - 7, ySeed, 2, 4)
            gSARnew.FillEllipse(Brushes.Green, xSeed + 7, ySeed, 2, 4)
        Next
    Next
End Sub
Private Sub highSeeds()
```

```
    Dim xSeed As Integer = 30
    Dim ySeed As Integer = 85
    gSARnew = Me.pbSeedAppRateNew.CreateGraphics
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    For xSeed = 19 To 180 Step 40
        If xSeed = 99 Then Continue For
        For ySeed = 85 To 99 Step 7
            gSARnew.FillEllipse(Brushes.Red, xSeed, ySeed, 2, 4)
            If ySeed = 85 Then Continue For
            gSARnew.FillEllipse(Brushes.Red, xSeed - 7, ySeed, 2, 4)
            gSARnew.FillEllipse(Brushes.Red, xSeed + 7, ySeed, 2, 4)
            If ySeed = 85 Or ySeed = 92 Then Continue For
            gSARnew.FillEllipse(Brushes.Red, xSeed - 7, ySeed, 2, 4)
            gSARnew.FillEllipse(Brushes.Red, xSeed + 7, ySeed, 2, 4)
            gSARnew.FillEllipse(Brushes.Red, xSeed - 14, ySeed, 2, 4)
            gSARnew.FillEllipse(Brushes.Red, xSeed + 14, ySeed, 2, 4)
        Next
    Next
End Sub
'draw scale
Private Sub drawSARScale()
    gSARnew = Me.pbSeedAppRateNew.CreateGraphics
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    gSARnew.Clear(SystemColors.Control)
    Dim myBlackPen As New Pen(Brushes.Black, 2)
    ''bottom label text
    'Scale
    Dim xScale As Integer = 10
    Dim yScale As Integer = 20
```

```
        Dim wScale As Integer = 60
        Dim hScale As Integer = 10
        gSARnew.FillRectangle(Brushes.IndianRed, xScale, yScale, wScale, hScale)
        gSARnew.FillRectangle(Brushes.YellowGreen, xScale + 60, yScale, wScale, hScale)
        gSARnew.FillRectangle(Brushes.IndianRed, xScale + 60 + 60, yScale, wScale, hScale)
        For x = 10 To 210 Step 30
            gSARnew.DrawLine(Pens.Black, x, yScale, x, yScale + 10)
        Next
        For x = 25 To 200 Step 30
            gSARnew.DrawLine(Pens.Black, x, yScale, x, yScale + 5)
        Next
        DrawNumberingSAR()
        draw1Funnel()
    End Sub
    Public Sub DrawNumberingSAR()
        Dim formGraphics As System.Drawing.Graphics = Me.pbSeedAppRateNew.CreateGraphics()
        Dim drawString As String = ""
        Dim drawFont As New System.Drawing.Font("Arial", 8)
        Dim drawFont2 As New System.Drawing.Font("Arial", 10, FontStyle.Bold)
        Dim drawBrush As New  _
            System.Drawing.SolidBrush(System.Drawing.Color.Black)
        'Dim x As Single = 10
        ' Dim y As Single
        Dim drawFormat As New System.Drawing.StringFormat
        drawFormat.FormatFlags = StringFormatFlags.LineLimit
        Dim i As Integer = 1
        Dim arrayDrawStirng() As String = {"10", "", "30", "", "50", "", "70"}
        For x = 0 To 180 Step 30
            drawString = arrayDrawStirng(i - 1)
            formGraphics.DrawString(drawString, drawFont, drawBrush, _
            x, 0, drawFormat)
```

```
increment:
            i = i + 1
        Next
    End Sub
    '
    Private Sub captionSeedAppRate()
        Dim formGraphics As System.Drawing.Graphics = Me.pb2SARnew.CreateGraphics()
        Dim drawString As String = "Seed Application Rate"
        Dim drawString2a As String = "("
        Dim drawString2 As String = "30-50"
        Dim drawString3 As String = "kg/ha )"
        Dim drawFont As New System.Drawing.Font("Arial", 10)
        Dim drawFont2 As New System.Drawing.Font("Arial", 11, FontStyle.Bold)
        Dim drawBrush As New  _
            System.Drawing.SolidBrush(System.Drawing.Color.Black)
        Dim drawFormat As New System.Drawing.StringFormat
        formGraphics.DrawString(drawString, drawFont, drawBrush, _
        0, 0, drawFormat)
        formGraphics.DrawString(drawString2a, drawFont, drawBrush, _
        18, 15, drawFormat)
        formGraphics.DrawString(drawString2, drawFont2, drawBrush, _
        25, 15, drawFormat)
        formGraphics.DrawString(drawString3, drawFont, drawBrush, _
        70, 15, drawFormat)
        formGraphics.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        formGraphics.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        drawFont.Dispose()
        drawBrush.Dispose()
        formGraphics.Dispose()
    End Sub
    Private Sub incTriSAR()
```

```
    gSARnew = Me.pbSeedAppRateNew.CreateGraphics
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim x As Integer = 95
    Dim y As Integer = 85
    Dim incTri() As Point = {New Point(x, y), _
                             New Point(x, y + 16), _
                             New Point(x + 10, y + 8)}
    gSARnew.FillPolygon(Brushes.Goldenrod, incTri)
End Sub
' Decreasing indicating triangle
Private Sub decTriSAR()
    gSARnew = Me.pbSeedAppRateNew.CreateGraphics
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim x As Integer = 95
    Dim y As Integer = 85
    'g.FillEllipse(Brushes.Silver, x, y, w, h)
    Dim incTri() As Point = {New Point(x + 10, y), _
                             New Point(x, y + 8), _
                             New Point(x + 10, y + 16)}
    gSARnew.FillPolygon(Brushes.Goldenrod, incTri)
End Sub
' stop of the trend , reading is not decreasing or increasing
Private Sub stopTrendSAR()
    gSARnew = Me.pbSeedAppRateNew.CreateGraphics
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gSARnew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim x As Integer = 93
    Dim y As Integer = 85
    Dim w As Integer = 14
```

```
        Dim h As Integer = 14
        'g.FillEllipse(Brushes.Silver, x, y, w, h)
        gSARnew.FillRectangle(Brushes.Goldenrod, x, y, w, h)
    End Sub
    Private Sub progressBarAllInOneSAR(ByVal AppRate As Integer)
        Dim wBar As Integer = (AppRate - 10) * 3
        gSARnew = Me.pbSeedAppRateNew.CreateGraphics
        '  gSARnew = Me.pbTp.CreateGraphics
        gSARnew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gSARnew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        If wBar <= 59 Then
            gSARnew.FillRectangle(Brushes.OrangeRed, 10, 33, wBar, 8)
            gSARnew.DrawRectangle(Pens.Gray, 10, 33, 180, 8)
        ElseIf wBar <= 120 Then
            gSARnew.FillRectangle(Brushes.Green, 10, 33, wBar, 8)
            gSARnew.DrawRectangle(Pens.Gray, 10, 33, 180, 8)
        ElseIf wBar <= 180 Then
            gSARnew.FillRectangle(Brushes.OrangeRed, 10, 33, wBar, 8)
            gSARnew.DrawRectangle(Pens.Gray, 10, 33, 180, 8)
        ElseIf wBar > 180 Then
            gSARnew.FillRectangle(Brushes.OrangeRed, 10, 33, 180, 8)
            gSARnew.DrawRectangle(Pens.Gray, 10, 33, 180, 8)
        End If
        ' gSARnew.Dispose()
    End Sub
    Private Sub seedAppRateReset()
        'newChanges
        '        gSARnew.Clear(SystemColors.Control)
        timerIdleSar.Enabled = True
        counterIdleSar = 0
        timerSARnew.Enabled = False
```

```
        timerSARreset.Enabled = False
        intList3.Remove(1)
        start1randomParameterAfterResest()
    End Sub
    Private Sub seedAppRateNew1()
        counterSARnew = counterSARnew + 1
        If counterSARnew <= 25 Then
            'incTriSAR()
            stopTrendSAR()
            okaySeeds()
            progressBarAllInOneSAR(35)
            lbSARreading.Text = FormatNumber(35, 0)
        ElseIf counterSARnew <= 40 Then
            incTriSAR()
            okaySeeds()
            progressBarAllInOneSAR(35 + counterSARnew - 25)
            lbSARreading.Text = FormatNumber(35 + counterSARnew - 25, 0)
        ElseIf counterSARnew <= 300 Then
            incTriSAR()
            highSeeds()
            progressBarAllInOneSAR(35 + counterSARnew - 25)
            lbSARreading.Text = FormatNumber(35 + counterSARnew - 25, 0)
        Else
        End If
    End Sub
    ' Decreasing rate
    Private Sub seedAppRateNew2()
        counterSARnew = counterSARnew + 1
        If counterSARnew <= 10 Then
            incTriSAR()
            okaySeeds()
```

```
            progressBarAllInOneSAR(40 + counterSARnew)
            lbSARreading.Text = FormatNumber(40 + counterSARnew, 0)
        ElseIf counterSARnew <= 300 Then
            incTriSAR()
            highSeeds()
            progressBarAllInOneSAR(40 + counterSARnew)
            lbSARreading.Text = FormatNumber(40 + counterSARnew, 0)
        Else
        End If
    End Sub
    Private Sub seedAppRateNew3()
        counterSARnew = counterSARnew + 1
        If counterSARnew <= 15 Then
            incTriSAR()
            okaySeeds()
            progressBarAllInOneSAR(35 + counterSARnew)
            lbSARreading.Text = FormatNumber(35 + counterSARnew, 0)
        ElseIf counterSARnew <= 300 Then
            incTriSAR()
            highSeeds()
            progressBarAllInOneSAR(35 + counterSARnew)
            lbSARreading.Text = FormatNumber(35 + counterSARnew, 0)
        Else
        End If
    End Sub
#End Region
#Region "Fertilizer"
    '######### FAR
    Public Property getCounterfertilizerAppRateNew As Integer
    Dim RandomFAR As Integer
    ' Dim seedRateSequenceList As New List(Of Integer)
```

```
    Dim counterFAR As Integer

    Dim gFAR As Graphics

    Dim hFAR As Integer

    ' Function for changing the color of the progress bar:

    Declare Function SendMessageFAR Lib "user32" Alias "SendMessageA" _

        (ByVal hwnd As Integer, ByVal wMsg As Integer, ByVal wParam As Integer, ByVal lParam
As Integer) As Integer

    Private Sub FARNewReset()

        timerIdleFar.Enabled = True

        counterIdleFar = 0

        timerFAR.Enabled = False

        timerFARreset.Enabled = False

        intList3.Remove(2)

        start1randomParameterAfterResest()

    End Sub

    'change3

    Dim rnFAR As Integer

    '  Dim rnFAR As Integer = 1

    Private Sub TimerFAR_Tick(sender As Object, e As EventArgs) Handles timerFAR.Tick

        captionFAR()

        drawFARscale()

        If rnFAR = 1 Then

            FARNew1()

        ElseIf rnFAR = 2 Then

            FARNew2()

        ElseIf rnFAR = 3 Then

            FARNew3()

        End If

    End Sub

    Private Sub elementsLocationSizeFAR()

        pBarFAR.Left = pbFAR.Left + 10

        pBarFAR.Top = pbFAR.Top + 45
```

```
        pBarFAR.Width = 180

        pBarFAR.Height = 7

        lbFARnewReading.Width = 36

        ' lblSeedAppRate.Left = PictureBox1.Left + (190 - lblSeedAppRate.Width) / 2 + 10

        lbFARnewReading.Left = pbFAR.Left + 100 - (lbFARnewReading.Width / 2)

        '  lbFARnewReading.Top = pbFAR.Top + 60

        lbFARnewReading.Top = pbFAR.Top + pbFAR.Height

        pbFARCaption.Left = pbFAR.Left + pbFAR.Width / 2 - pbFARCaption.Width / 2

        pbFARCaption.Top = pbFAR.Top + pbFAR.Height + 5

        lbNameFAR.Left = pbFAR.Left

        lbNameFAR.Top = pbFAR.Top - lbNameFAR.Height - 5

        lbFARunits.Top = lbFARnewReading.Top + lbFARnewReading.Height

        lbFARunits.Left = lbFARnewReading.Left + lbFARnewReading.Width / 2 - lbFARunits.Width
/ 2

        'lbNameFAR.Left = pbFAR.Left + pbFAR.Width / 2 - lbNameFAR.Width / 2

        'lbNameFAR.Top = pbFAR.Top + pbFAR.Height + 5

    End Sub

    Private Sub draw1FunnelFAR()

        Dim xFunnel As Integer = 40

        Dim yFunnel As Integer = 60

        gFAR = Me.pbFAR.CreateGraphics

        gFAR.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

        gFAR.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        ' draw a top bar

        ' g.FillRectangle(Brushes.Black, 10, 55, 180, 2)

        ' g.FillPolygon(Brushes.SteelBlue, funnel1)

        For xFunnel = 10 To 180 Step 40

            If xFunnel = 90 Then Continue For

            Dim funnel1() As Point = {New Point(xFunnel, yFunnel), _

                                        New Point(xFunnel + 9, yFunnel + 10), _

                                      New Point(xFunnel + 9, yFunnel + 20), _

                                      New Point(xFunnel + 11, yFunnel + 20), _
```

```
                                New Point(xFunnel + 11, yFunnel + 10), _
                                New Point(xFunnel + 20, yFunnel)}
        gFAR.FillPolygon(Brushes.Gray, funnel1)
    Next
    '  g.Dispose()
End Sub
'Draw seeds
Private Sub lowSeedsFAR()
    Dim xSeed As Integer = 30
    Dim ySeed As Integer = 85
    gFAR = Me.pbFAR.CreateGraphics
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    For xSeed = 19 To 180 Step 40
        If xSeed = 99 Then Continue For
        For ySeed = 85 To 99 Step 7
            gFAR.FillEllipse(Brushes.Red, xSeed, ySeed, 2, 4)
            'g.DrawEllipse(Pens.Black, xSeed, ySeed, 2, 4)
        Next
    Next
End Sub
Private Sub okaySeedsFAR()
    Dim xSeed As Integer = 30
    Dim ySeed As Integer = 85
    gFAR = Me.pbFAR.CreateGraphics
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    For xSeed = 19 To 180 Step 40
        If xSeed = 99 Then Continue For
        For ySeed = 85 To 97 Step 7
            gFAR.FillEllipse(Brushes.Green, xSeed, ySeed, 2, 4)
```

```
            If ySeed = 85 Then Continue For

            gFAR.FillEllipse(Brushes.Green, xSeed - 7, ySeed, 2, 4)

            gFAR.FillEllipse(Brushes.Green, xSeed + 7, ySeed, 2, 4)

        Next

    Next

End Sub

Private Sub highSeedsFAR()

    Dim xSeed As Integer = 30

    Dim ySeed As Integer = 85

    gFAR = Me.pbFAR.CreateGraphics

    gFAR.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

    gFAR.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

    For xSeed = 19 To 180 Step 40

        If xSeed = 99 Then Continue For

        For ySeed = 85 To 99 Step 7

            gFAR.FillEllipse(Brushes.Red, xSeed, ySeed, 2, 4)

            If ySeed = 85 Then Continue For

            gFAR.FillEllipse(Brushes.Red, xSeed - 7, ySeed, 2, 4)

            gFAR.FillEllipse(Brushes.Red, xSeed + 7, ySeed, 2, 4)

            If ySeed = 85 Or ySeed = 92 Then Continue For

            gFAR.FillEllipse(Brushes.Red, xSeed - 7, ySeed, 2, 4)

            gFAR.FillEllipse(Brushes.Red, xSeed + 7, ySeed, 2, 4)

            gFAR.FillEllipse(Brushes.Red, xSeed - 14, ySeed, 2, 4)

            gFAR.FillEllipse(Brushes.Red, xSeed + 14, ySeed, 2, 4)

            'g.DrawEllipse(Pens.Black, xSeed, ySeed, 2, 4)

        Next

    Next

End Sub

'draw scale

Private Sub drawFARscale()

    gFAR = Me.pbFAR.CreateGraphics
```

```
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    gFAR.Clear(SystemColors.Control)
    Dim myBlackPen As New Pen(Brushes.Black, 2)
    ''bottom label text
    'Scale
    Dim xScale As Integer = 10
    Dim yScale As Integer = 20
    Dim wScale As Integer = 60
    Dim hScale As Integer = 10
    gFAR.FillRectangle(Brushes.IndianRed, xScale, yScale, wScale, hScale)
    gFAR.FillRectangle(Brushes.YellowGreen, xScale + 60, yScale, wScale, hScale)
    gFAR.FillRectangle(Brushes.IndianRed, xScale + 60 + 60, yScale, wScale, hScale)
    For x = 10 To 210 Step 30
        gFAR.DrawLine(Pens.Black, x, yScale, x, yScale + 10)
    Next
    For x = 25 To 200 Step 30
        gFAR.DrawLine(Pens.Black, x, yScale, x, yScale + 5)
    Next
    DrawNumberingFAR()
    draw1FunnelFAR()
End Sub
' numbering of the scale
Public Sub DrawNumberingFAR()
    Dim formGraphics As System.Drawing.Graphics = Me.pbFAR.CreateGraphics()
    Dim drawString As String = ""
    Dim drawFont As New System.Drawing.Font("Arial", 8)
    Dim drawBrush As New  _
        System.Drawing.SolidBrush(System.Drawing.Color.Black)
    'Dim x As Single = 10
    ' Dim y As Single
```

```
    Dim drawFormat As New System.Drawing.StringFormat
    drawFormat.FormatFlags = StringFormatFlags.LineLimit
    Dim i As Integer = 1
    Dim arrayDrawStirng() As String = {"20", "", "60", "", "100", "", "140"}
    For x = 0 To 180 Step 30
        drawString = arrayDrawStirng(i - 1)
        formGraphics.DrawString(drawString, drawFont, drawBrush, _
        x, 0, drawFormat)
        i = i + 1
    Next
End Sub
'
Private Sub captionFAR()
    Dim formGraphics As System.Drawing.Graphics = Me.pbFARCaption.CreateGraphics()
    Dim drawString As String = "Fertilizer Application Rate"
    Dim drawString2a As String = "("
    Dim drawString2 As String = "60-100"
    Dim drawString3 As String = " kg/ha )"
    Dim drawFont As New System.Drawing.Font("Arial", 10)
    Dim drawFont2 As New System.Drawing.Font("Arial", 11, FontStyle.Bold)
    Dim drawBrush As New  _
        System.Drawing.SolidBrush(System.Drawing.Color.Black)
    Dim drawFormat As New System.Drawing.StringFormat
    formGraphics.DrawString(drawString, drawFont, drawBrush, _
    0, 0, drawFormat)
    formGraphics.DrawString(drawString2a, drawFont, drawBrush, _
    18, 15, drawFormat)
    formGraphics.DrawString(drawString2, drawFont2, drawBrush, _
    25, 15, drawFormat)
    formGraphics.DrawString(drawString3, drawFont, drawBrush, _
    70, 15, drawFormat)
```

```
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
End Sub
Private Sub incTriFAR()
    gFAR = Me.pbFAR.CreateGraphics
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim x As Integer = 95
    Dim y As Integer = 85
    'g.FillEllipse(Brushes.Silver, x, y, w, h)
    Dim incTri() As Point = {New Point(x, y), _
                             New Point(x, y + 16), _
                             New Point(x + 10, y + 8)}
    gFAR.FillPolygon(Brushes.Goldenrod, incTri)
End Sub
' Decreasing indicating triangle
Private Sub decTriFAR()
    gFAR = Me.pbFAR.CreateGraphics
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim x As Integer = 95
    Dim y As Integer = 85
    'g.FillEllipse(Brushes.Silver, x, y, w, h)
    Dim incTri() As Point = {New Point(x + 10, y), _
                             New Point(x, y + 8), _
                             New Point(x + 10, y + 16)}
    gFAR.FillPolygon(Brushes.Goldenrod, incTri)
End Sub
' stop of the trend , reading is not decreasing or increasing
Private Sub stopTrendFAR()
    gFAR = Me.pbFAR.CreateGraphics
```

```
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim x As Integer = 93
    Dim y As Integer = 85
    Dim w As Integer = 14
    Dim h As Integer = 14
    'g.FillEllipse(Brushes.Silver, x, y, w, h)
    gFAR.FillRectangle(Brushes.Goldenrod, x, y, w, h)
End Sub
Private Sub progressBarAllInOneFAR(ByVal AppRate As Integer)
    Dim wBar As Integer = (AppRate - 10) * 3
    gFAR = Me.pbFAR.CreateGraphics
    '  gFAR = Me.pbTp.CreateGraphics
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gFAR.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    If wBar <= 59 Then
        gFAR.FillRectangle(Brushes.OrangeRed, 10, 33, wBar, 8)
        gFAR.DrawRectangle(Pens.Gray, 10, 33, 180, 8)
    ElseIf wBar <= 120 Then
        gFAR.FillRectangle(Brushes.Green, 10, 33, wBar, 8)
        gFAR.DrawRectangle(Pens.Gray, 10, 33, 180, 8)
    ElseIf wBar <= 180 Then
        gFAR.FillRectangle(Brushes.OrangeRed, 10, 33, wBar, 8)
        gFAR.DrawRectangle(Pens.Gray, 10, 33, 180, 8)
    ElseIf wBar > 180 Then
        gFAR.FillRectangle(Brushes.OrangeRed, 10, 33, 180, 8)
        gFAR.DrawRectangle(Pens.Gray, 10, 33, 180, 8)
    End If
    ' gSARnew.Dispose()
End Sub
Private Sub FARNew1()
```

```
    counterFAR = counterFAR + 1
    Dim reading As Integer = 70
    If counterFAR <= 15 Then
        'incTriSAR()
        stopTrendFAR()
        okaySeedsFAR()
        progressBarAllInOneFAR(reading / 2)
        lbFARnewReading.Text = FormatNumber(reading, 0)
    ElseIf counterFAR <= 30 Then
        incTriFAR()
        okaySeedsFAR()
        progressBarAllInOneFAR(reading / 2 + counterFAR - 15)
        lbFARnewReading.Text = FormatNumber(reading + (counterFAR - 15) * 2, 0)
    ElseIf counterFAR <= 300 Then
        incTriFAR()
        highSeedsFAR()
        progressBarAllInOneFAR(reading / 2 + counterFAR - 15)
        lbFARnewReading.Text = FormatNumber(reading + (counterFAR - 15) * 2, 0)
    Else
    End If
End Sub
' Decreasing rate
Private Sub FARNew2()
    counterFAR = counterFAR + 1
    Dim reading As Integer = 80
    If counterFAR <= 20 Then
        'incTriSAR()
        stopTrendFAR()
        okaySeedsFAR()
        progressBarAllInOneFAR(reading / 2)
        lbFARnewReading.Text = FormatNumber(reading, 0)
```

```
        ElseIf counterFAR <= 30 Then
            incTriFAR()
            okaySeedsFAR()
            progressBarAllInOneFAR(reading / 2 + counterFAR - 20)
            lbFARnewReading.Text = FormatNumber(reading + (counterFAR - 20) * 2, 0)
        ElseIf counterFAR <= 300 Then
            incTriFAR()
            highSeedsFAR()
            progressBarAllInOneFAR(reading / 2 + counterFAR - 20)
            lbFARnewReading.Text = FormatNumber(reading + (counterFAR - 20) * 2, 0)
        Else
        End If
    End Sub
    Private Sub FARNew3()
        counterFAR = counterFAR + 1
        Dim reading As Integer = 70
        If counterFAR <= 15 Then
            incTriFAR()
            okaySeedsFAR()
            progressBarAllInOneFAR(reading / 2 + counterFAR)
            lbFARnewReading.Text = FormatNumber(reading + (counterFAR) * 2, 0)
        ElseIf counterFAR <= 300 Then
            incTriFAR()
            highSeedsFAR()
            progressBarAllInOneFAR(reading / 2 + counterFAR)
            lbFARnewReading.Text = FormatNumber(reading + (counterFAR) * 2, 0)
        Else
        End If
    End Sub
#End Region
#Region "Speed"
```

```vb
    ''######## Speed new

    Public Property getCounterSpeedNew As Integer

    Dim RandomSpeedNew As Integer

    Dim incrementPBarSpeedNew As Integer

    Dim counterSpeedNew As Integer

    Dim gSpeedNew As Graphics

    Dim hSpeedNew As Integer

    ' Function for changing the color of the progress bar:

    Declare Function SendMessageSpeedNew Lib "user32" Alias "SendMessageA" _

        (ByVal hwnd As Integer, ByVal wMsg As Integer, ByVal wParam As Integer, ByVal lParam
As Integer) As Integer

    Dim rnSpeed As Integer

    ' Dim rnSpeed As Integer = 1

    Private Sub TimerSpeedNew_Tick(sender As Object, e As EventArgs) Handles
timerSpeedNew.Tick

        captionSpeed()

        drawDialNumberingSpeedNew()

        If rnSpeed = 1 Then

            speedNew1()

        ElseIf rnSpeed = 2 Then

            speedNew2()

        ElseIf rnSpeed = 3 Then

            speedNew3()

        End If

    End Sub

    Private Sub speedNewReset()

        ' new Changes

        timerIdleSpeed.Enabled = True

        counterIdleSpeed = 0

        timerSpeedNew.Enabled = False

        timerSpeedReset.Enabled = False

        intList3.Remove(3)
```

```
        start1randomParameterAfterResest()

    End Sub

    Private Sub elementsLocationSizeSpeedNew()

        lbSpeedNewReading.BackColor = Color.WhiteSmoke

        lbSpeedCaption.Top = pbSpeedNew.Top - lbSpeedCaption.Height - 5

        lbSpeedCaption.Left = pbSpeedNew.Left

        lbSpeedNewReading.Top = pbSpeedNew.Top + pbSpeedNew.Height

        lbSpeedNewReading.Left = pbSpeedNew.Left + pbSpeedNew.Width / 2 -
lbSpeedNewReading.Width / 2

        lbSpeedUnits.Top = lbSpeedNewReading.Top + lbSpeedNewReading.Height

        lbSpeedUnits.Left = lbSpeedNewReading.Left + lbSpeedNewReading.Width / 2 -
lbSpeedUnits.Width / 2

    End Sub

    Private Sub progressBarAllInOne(ByVal seedAppRate As Integer)

        Dim wBar As Integer = (seedAppRate - 30) * 3

        gSpeedNew = Me.pbSpeedNew.CreateGraphics

        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        If wBar <= 59 Then

            gSpeedNew.FillRectangle(Brushes.OrangeRed, 10, 84, wBar, 8)

            gSpeedNew.DrawRectangle(Pens.Gray, 10, 84, 180, 8)

        ElseIf wBar <= 119 Then

            gSpeedNew.FillRectangle(Brushes.Green, 10, 84, wBar, 8)

            gSpeedNew.DrawRectangle(Pens.Gray, 10, 84, 180, 8)

        ElseIf wBar <= 180 Then

            gSpeedNew.FillRectangle(Brushes.OrangeRed, 10, 84, wBar, 8)

            gSpeedNew.DrawRectangle(Pens.Gray, 10, 84, 180, 8)

        End If

        gSpeedNew.Dispose()

    End Sub

    'draw scale

    Private Sub drawDialNumberingSpeedNew()
```

```
Dim g As System.Drawing.Graphics = Me.pbSpeedNew.CreateGraphics()
Dim drawString As String = ""
Dim drawFont As New System.Drawing.Font("Arial", 8)
Dim drawBrush As New  _
    System.Drawing.SolidBrush(System.Drawing.Color.Black)
Dim drawFormat As New System.Drawing.StringFormat
drawFormat.Alignment = StringAlignment.Center
drawFormat.LineAlignment = StringAlignment.Center
drawFormat.FormatFlags = StringFormatFlags.LineLimit
'   g = Me.PictureBox1.CreateGraphics
g.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
g.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
'background color of the picturebox
'   g.Clear(SystemColors.Control)
Dim myRedPen As New Pen(Brushes.IndianRed, 12)
Dim myYellowGreenPen As New Pen(Brushes.YellowGreen, 12)
Dim myBlackPen As New Pen(Brushes.Black, 2)
Dim myRectangle As New Rectangle(20, 20, 150, 150)
'Draw Red and YellowGreen dial
g.DrawArc(myRedPen, myRectangle, 180, 72)
g.DrawArc(myYellowGreenPen, myRectangle, 252, 60)
g.DrawArc(myRedPen, myRectangle, 300, 60)
'Dim a As Double
Dim Xc As Integer = 95
Dim Yc As Integer = 95
Dim dialNumbering As Integer = 0
Dim angle() As Double = {3.14, 4.396, 5.233, 6.28}
Dim speedNums() As String = {"0", "6", "10", "15"}
For i = 0 To 3
    Dim XnewNumbering As Integer = Xc + 90 * Math.Cos(angle(i))
    Dim YnewNumbering As Integer = Yc + 90 * Math.Sin(angle(i))
```

```
            g.DrawString(speedNums(i), drawFont, drawBrush, _
            XnewNumbering, YnewNumbering, drawFormat)
        Next
        drawThickLines()
        drawTHINlines()
    End Sub
    Private Sub needleSpeedNew(ByVal aDegrees As Double)
        gSpeedNew = Me.pbSpeedNew.CreateGraphics
        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        gSpeedNew.Clear(SystemColors.Control)
        Dim aRadians As Double = 0.01745 * aDegrees
        Dim reading As Double = 15 / 180 * (aDegrees - 180)
        lbSpeedNewReading.Text = FormatNumber(reading, 1)
        Dim xNeedleTip As Integer = 95 + 50 * Math.Cos(aRadians)
        Dim yNeeleTip As Integer = 95 + 50 * Math.Sin(aRadians)
        'lbSpeedNewReading.Left = pbSpeedNew.Left + 95 - lbSpeedNewReading.Width / 2
        'lbSpeedNewReading.Top = pbSpeedNew.Top + 112
        Dim myNeedle1() As Point = {New Point(xNeedleTip, yNeeleTip), _
                                  New Point(90, 100), New Point(100, 90)}
        Dim myNeedle2() As Point = {New Point(xNeedleTip, yNeeleTip), _
                                   New Point(90, 95), New Point(100, 95)}
        Dim myNeedle3() As Point = {New Point(xNeedleTip, yNeeleTip), _
                                   New Point(90, 90), New Point(100, 100)}
        Dim myNeedle4() As Point = {New Point(144.699, 100.471), _
                                   New Point(90, 90), New Point(100, 100)}
        Dim myBlackPen As New Pen(Brushes.Black, 2)
        Dim myWhitePen As New Pen(Brushes.White, 2)
        If reading <= 5.9 Then
            gSpeedNew.FillPolygon(Brushes.OrangeRed, myNeedle1)
            gSpeedNew.DrawPolygon(myBlackPen, myNeedle1)
```

```
            gSpeedNew.FillEllipse(Brushes.OrangeRed, 84, 84, 22, 22)
            gSpeedNew.DrawEllipse(myBlackPen, 84, 84, 22, 22)
        ElseIf reading >= 6.0 And reading <= 10.0 Then
            gSpeedNew.FillPolygon(Brushes.Green, myNeedle2)
            gSpeedNew.DrawPolygon(myBlackPen, myNeedle2)
            gSpeedNew.FillEllipse(Brushes.Green, 84, 84, 22, 22)
            gSpeedNew.DrawEllipse(myBlackPen, 84, 84, 22, 22)
        ElseIf reading > 10.0 And reading <= 14.9 Then
            gSpeedNew.FillPolygon(Brushes.OrangeRed, myNeedle3)
            gSpeedNew.DrawPolygon(myBlackPen, myNeedle3)
            gSpeedNew.FillEllipse(Brushes.OrangeRed, 84, 84, 22, 22)
            gSpeedNew.DrawEllipse(myBlackPen, 84, 84, 22, 22)
        ElseIf reading > 14.9 Then
            gSpeedNew.FillPolygon(Brushes.OrangeRed, myNeedle4)
            gSpeedNew.DrawPolygon(myBlackPen, myNeedle4)
            gSpeedNew.FillEllipse(Brushes.OrangeRed, 84, 84, 22, 22)
            gSpeedNew.DrawEllipse(myBlackPen, 84, 84, 22, 22)
        End If
        drawDialNumberingSpeedNew()
    End Sub
    Private Sub drawThickLines()
        gSpeedNew = Me.pbSpeedNew.CreateGraphics
        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        Dim myBlackPen As New Pen(Brushes.Black, 2)
        Dim a As Double
        Dim dialNumbering As Integer = 0
        For a = 3.14 To 2 * 3.14 Step (3.14 / 15) * 2
            dialNumbering = dialNumbering + 1
            Dim Xc As Integer = 95
            Dim Yc As Integer = 95
```

```
            Dim Xp As Integer = 20

            Dim Yp As Integer = 95

            'Dim Xnew As Integer = Xc + 75 * Math.Cos(a)

            'Dim Ynew As Integer = Yc + 75 * Math.Sin(a)

            Dim XnewNumbering As Integer = Xc + 80 * Math.Cos(a)

            Dim YnewNumbering As Integer = Yc + 80 * Math.Sin(a)

            Dim XlineBottom As Integer = Xc + 68 * Math.Cos(a)

            Dim ylineBottom As Integer = Yc + 68 * Math.Sin(a)

            ' points for the markings

            gSpeedNew.DrawLine(myBlackPen, XnewNumbering, YnewNumbering, XlineBottom,
ylineBottom)

            ' g.DrawEllipse(myBlackPen, Xnew, Ynew, 2, 2)

        Next

    End Sub

    Private Sub drawTHINlines()

        gSpeedNew = Me.pbSpeedNew.CreateGraphics

        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        ' Dim myBlackPen As New Pen(Brushes.DarkGray, 2)

        Dim a As Double

        Dim dialNumbering As Integer = 0

        For a = 3.14 + 3.14 / 15 To 2 * 3.14 Step (3.14 / 15) * 2

            dialNumbering = dialNumbering + 1

            Dim Xc As Integer = 95

            Dim Yc As Integer = 95

            Dim Xp As Integer = 20

            Dim Yp As Integer = 95

            'Dim Xnew As Integer = Xc + 75 * Math.Cos(a)

            'Dim Ynew As Integer = Yc + 75 * Math.Sin(a)

            Dim XnewNumbering As Integer = Xc + 80 * Math.Cos(a)

            Dim YnewNumbering As Integer = Yc + 80 * Math.Sin(a)

            Dim XlineBottom As Integer = Xc + 70 * Math.Cos(a)
```

```
            Dim ylineBottom As Integer = Yc + 70 * Math.Sin(a)
            ' points for the markings
            gSpeedNew.DrawLine(Pens.Black, XnewNumbering, YnewNumbering, XlineBottom,
ylineBottom)
            ' g.DrawEllipse(myBlackPen, Xnew, Ynew, 2, 2)
        Next
    End Sub
    ' numbering of the scale
    '
    Private Sub captionSpeed()
        Dim formGraphics As System.Drawing.Graphics = Me.pbSpeedNCaption.CreateGraphics()
        Dim drawString As String = "Speed"
        Dim drawString2a As String = "("
        Dim drawString2 As String = "6-10"
        Dim drawString3 As String = " km/h )"
        Dim drawFont As New System.Drawing.Font("Arial", 10)
        Dim drawFont2 As New System.Drawing.Font("Arial", 11, FontStyle.Bold)
        Dim drawBrush As New  _
            System.Drawing.SolidBrush(System.Drawing.Color.Black)
        Dim drawFormat As New System.Drawing.StringFormat
        'drawFormat.LineAlignment = StringAlignment.Center
        'drawFormat.Alignment = StringAlignment.Center
        formGraphics.DrawString(drawString, drawFont, drawBrush, _
        30, 0, drawFormat)
        formGraphics.DrawString(drawString2a, drawFont, drawBrush, _
        10, 15, drawFormat)
        formGraphics.DrawString(drawString2, drawFont2, drawBrush, _
        15, 15, drawFormat)
        formGraphics.DrawString(drawString3, drawFont, drawBrush, _
        50, 15, drawFormat)
        formGraphics.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        formGraphics.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
```

```
        drawFont.Dispose()
        drawBrush.Dispose()
        formGraphics.Dispose()
    End Sub
    Private Sub incTriSpeedNew()
        gSpeedNew = Me.pbSpeedNew.CreateGraphics
        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        '   g.Clear(SystemColors.Control)
        ' PicBoxArrowIndicator.BackColor = Color.Silver
        Dim x As Integer = 65
        Dim y As Integer = 5
        'g.FillEllipse(Brushes.Silver, x, y, w, h)
        Dim incTri() As Point = {New Point(x, y), _
                                 New Point(x, y + 16), _
                                 New Point(x + 10, y + 8)}
        gSpeedNew.FillPolygon(Brushes.Goldenrod, incTri)
        gSpeedNew.Dispose()
    End Sub
    ' Decreasing indicating triangle
    Private Sub decTriSpeedNew()
        gSpeedNew = Me.pbSpeedNew.CreateGraphics
        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        '   g.Clear(SystemColors.Control)
        ' PicBoxArrowIndicator.BackColor = Color.Silver
        Dim x As Integer = 65
        Dim y As Integer = 5
        'g.FillEllipse(Brushes.Silver, x, y, w, h)
        Dim incTri() As Point = {New Point(x + 10, y), _
                                 New Point(x, y + 8), _
```

```
                                New Point(x + 10, y + 16)}
    gSpeedNew.FillPolygon(Brushes.Goldenrod, incTri)
    gSpeedNew.Dispose()
End Sub
' stop of the trend , reading is not decreasing or increasing
Private Sub stopTrendSpeedNew()
    gSpeedNew = Me.pbSpeedNew.CreateGraphics
    gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gSpeedNew.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    '  g.Clear(SystemColors.Control)
    ' PicBoxArrowIndicator.BackColor = Color.Silver
    Dim x As Integer = 65
    Dim y As Integer = 5
    Dim w As Integer = 14
    Dim h As Integer = 14
    'g.FillEllipse(Brushes.Silver, x, y, w, h)
    gSpeedNew.FillRectangle(Brushes.Goldenrod, x, y, w, h)
    gSpeedNew.Dispose()
End Sub
Private Sub speedNew1()
    counterSpeedNew = counterSpeedNew + 1
    ' value of the counter/progressbar1.value = (scaleReading-10)*10/6
    If counterSpeedNew <= 18 Then
        needleSpeedNew(270)
        ' needleSpeedNew(needleInput)
        gSpeedNew.Dispose()
    ElseIf counterSpeedNew <= 300 Then
        Dim needleInput As Integer = 270 + (counterSpeedNew - 18) * 1.7
        needleSpeedNew(needleInput)
        gSpeedNew.Dispose()
    Else
```

```
    End If
End Sub
Private Sub speedNew2()
    counterSpeedNew = counterSpeedNew + 1
    ' value of the counter/progressbar1.value = (scaleReading-10)*10/6
    If counterSpeedNew <= 25 Then
        '  drawDialNumbering()
        '  Dim needleInput As Integer = 270 + counterSpeedNew * 1.7
        needleSpeedNew(270)
        ' needleSpeedNew(needleInput)
        gSpeedNew.Dispose()
    ElseIf counterSpeedNew <= 300 Then
        Dim needleInput As Integer = 270 + (counterSpeedNew - 25) * 1.7
        needleSpeedNew(needleInput)
        gSpeedNew.Dispose()
    Else
        'Me.Visible = False
        '_3Queries.Visible = True
        'Me.Close()
    End If
End Sub
Private Sub speedNew3()
    counterSpeedNew = counterSpeedNew + 1
    If counterSpeedNew <= 300 Then
        Dim needleInput As Integer = 258 + (counterSpeedNew) * 1.7
        needleSpeedNew(needleInput)
        gSpeedNew.Dispose()
    Else
        'Me.Visible = False
        '_3Queries.Visible = True
        'Me.Close()
```

```
        End If
    End Sub
#End Region
#Region "Fan RPM"
    ' Decreasing rate
    '''' Fan RPM ###############
    Public Property getCounterFanRPMnew As Integer
    Dim RandomFanRPM As Integer
    ' Dim seedRateSequenceList As New List(Of Integer)
    Dim counterFanRPM As Integer
    Dim gFanRPM As Graphics
    Dim hFanRPM As Integer
    Dim XcFan As Integer = 105
    Dim YcFan As Integer = 105
    'change5
    Dim rnFanRpm As Integer
    ' Dim rnFanRpm As Integer = 1
    Private Sub TimerFanRPM_Tick(sender As Object, e As EventArgs) Handles timerFanRPM.Tick
        captionFan()
        '  drawDialNumbering()
        '  h = h + 1
        If rnFanRpm = 1 Then
            fanRpmNew1()
        ElseIf rnFanRpm = 2 Then
            fanRpmNew2()
        ElseIf rnFanRpm = 3 Then
            fanRpmNew3()
        End If
    End Sub
    Private Sub fanRpmNewReset()
        timerIdleFan.Enabled = True
```

```
        counterIdleFan = 0

        timerFanRPM.Enabled = False

        timerFanRPMreset.Enabled = False

        intList3.Remove(4)

        start1randomParameterAfterResest()

    End Sub

    Private Sub elementsLocationSizeFan()

        lbFanRPMreading.BackColor = Color.WhiteSmoke

        lbFanRPMreading.Width = 36

        'lbFanRPMreading.Left = pbFanRPM.Left + (XcFan - lbFanRPMreading.Width / 2)

        'lbFanRPMreading.Top = pbFanRPM.Top + YcFan + 18

        lbFanRPMreading.Left = pbFanRPM.Left + (XcFan - lbFanRPMreading.Width / 2)

        lbFanRPMreading.Top = pbFanRPM.Top + pbFanRPM.Height

        lbFanRpmUnits.Left = lbFanRPMreading.Left + lbFanRPMreading.Width / 2 -
lbFanRpmUnits.Width / 2

        lbFanRpmUnits.Top = lbFanRPMreading.Top + lbFanRPMreading.Height

        pbFanRPMCaption.Left = pbFanRPM.Left + pbFanRPM.Width / 2 - pbFanRPMCaption.Width / 2

        pbFanRPMCaption.Top = pbFanRPM.Top + pbFanRPM.Height + 20

        'lbFanRPMname.Left = pbFanRPM.Left + pbFanRPM.Width / 2 - lbFanRPMname.Width / 2

        'lbFanRPMname.Top = pbFanRPM.Top + pbFanRPM.Height + 20

        lbFanRPMname.Left = pbFanRPM.Left

        lbFanRPMname.Top = pbFanRPM.Top - lbFanRPMname.Height - 5

    End Sub

    'draw scale

    Private Sub drawDialNumberingFan()

        Dim g As System.Drawing.Graphics = Me.pbFanRPM.CreateGraphics()

        Dim drawString As String = ""

        Dim drawFont As New System.Drawing.Font("Arial", 8)

        Dim drawBrush As New  _

            System.Drawing.SolidBrush(System.Drawing.Color.Black)

        Dim drawFormat As New System.Drawing.StringFormat

        drawFormat.Alignment = StringAlignment.Center
```

```
    drawFormat.LineAlignment = StringAlignment.Center
    '   drawFormat.FormatFlags = StringFormatFlags.
    drawFormat.FormatFlags = StringFormatFlags.LineLimit
    '  g = Me.PictureBox1.CreateGraphics
    g.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    g.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    'background color of the picturebox
    ' g.Clear(SystemColors.Highlight)
    Dim myRedPen As New Pen(Brushes.IndianRed, 12)
    Dim myGreenPen As New Pen(Brushes.YellowGreen, 12)
    Dim myBlackPen As New Pen(Brushes.Black, 2)
    '  Dim myRectangle As New Rectangle(20, 20, 150, 150)
    Dim myRectangle As New Rectangle(30, 30, 150, 150)
    'Draw Red and Green dial
    g.DrawArc(myRedPen, myRectangle, 180, 90)
    g.DrawArc(myGreenPen, myRectangle, 270, 36)
    g.DrawArc(myRedPen, myRectangle, 306, 54)
    'Dim a As Double
    Dim dN As Integer = 0
    Dim angle() As Double = {3.14, 4.71, 5.338, 6.28}
    Dim number() As String = {"0", "500", "700", "1000"}
    For i = 1 To 4
        Dim XnewNumbering As Integer = XcFan + 92 * Math.Cos(angle(i - 1))
        Dim YnewNumbering As Integer = YcFan + 92 * Math.Sin(angle(i - 1))
        g.DrawString(number(i - 1), drawFont, drawBrush, _
       XnewNumbering, YnewNumbering, drawFormat)
        '  i = i + 1
    Next
    drawThickLinesFan()
    drawTHINlinesFan()
End Sub
```

```
    Private Sub needleFan(ByVal aDegrees As Double)

        gFanRPM = Me.pbFanRPM.CreateGraphics

        gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

        gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        gFanRPM.Clear(SystemColors.Control)

        Dim aRadians As Double = 0.01745 * aDegrees

        lbFanRPMreading.Text = FormatNumber(1000 / 180 * (aDegrees - 180), 0)

        Dim xNeedleTip As Integer = XcFan + 50 * Math.Cos(aRadians)

        Dim yNeeleTip As Integer = YcFan + 50 * Math.Sin(aRadians)

        'lblSeedAppRate.Left = PictureBox1.Left + Xc - lblSeedAppRate.Width / 2

        'lblSeedAppRate.Top = PictureBox1.Top + Yc + 15

        Dim myNeedle1() As Point = {New Point(xNeedleTip, yNeeleTip), _

                                    New Point(XcFan - 5, YcFan + 5), New Point(XcFan + 5, YcFan
- 5)}

        Dim myNeedle2() As Point = {New Point(xNeedleTip, yNeeleTip), _

                                     New Point(XcFan - 5, YcFan), New Point(XcFan + 5, YcFan)}

        Dim myNeedle3() As Point = {New Point(xNeedleTip, yNeeleTip), _

                                     New Point(XcFan - 5, YcFan - 5), New Point(XcFan + 5, YcFan
+ 5)}

        Dim myNeedle4() As Point = {New Point(154.7, 110.47), _

                                    New Point(XcFan - 5, YcFan - 5), New Point(XcFan + 5, YcFan
+ 5)}

        Dim myBlackPen As New Pen(Brushes.Black, 2)

        Dim myWhitePen As New Pen(Brushes.White, 2)

        If aDegrees <= 269 Then

            gFanRPM.FillPolygon(Brushes.OrangeRed, myNeedle1)

            gFanRPM.DrawPolygon(myBlackPen, myNeedle1)

            gFanRPM.FillEllipse(Brushes.OrangeRed, XcFan - 11, YcFan - 11, 22, 22)

            gFanRPM.DrawEllipse(myBlackPen, XcFan - 11, YcFan - 11, 22, 22)

        ElseIf aDegrees <= 306 Then

            gFanRPM.FillPolygon(Brushes.Green, myNeedle2)

            gFanRPM.DrawPolygon(myBlackPen, myNeedle2)

            gFanRPM.FillEllipse(Brushes.Green, XcFan - 11, YcFan - 11, 22, 22)
```

```
            gFanRPM.DrawEllipse(myBlackPen, XcFan - 11, YcFan - 11, 22, 22)
        ElseIf aDegrees <= 360 Then
            gFanRPM.FillPolygon(Brushes.OrangeRed, myNeedle3)
            gFanRPM.DrawPolygon(myBlackPen, myNeedle3)
            gFanRPM.FillEllipse(Brushes.OrangeRed, XcFan - 11, YcFan - 11, 22, 22)
            gFanRPM.DrawEllipse(myBlackPen, XcFan - 11, YcFan - 11, 22, 22)
        ElseIf aDegrees > 360 Then
            gFanRPM.FillPolygon(Brushes.OrangeRed, myNeedle4)
            gFanRPM.DrawPolygon(myBlackPen, myNeedle4)
            gFanRPM.FillEllipse(Brushes.OrangeRed, XcFan - 11, YcFan - 11, 22, 22)
            gFanRPM.DrawEllipse(myBlackPen, XcFan - 11, YcFan - 11, 22, 22)
        End If
        drawDialNumberingFan()
    End Sub
    Private Sub drawThickLinesFan()
        gFanRPM = Me.pbFanRPM.CreateGraphics
        gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        Dim myBlackPen As New Pen(Brushes.Black, 2)
        Dim a As Double
        Dim dialNumbering As Integer = 0
        For a = 3.14 + 0.314 To 2 * 3.14 Step (3.14 / 10) * 2
            dialNumbering = dialNumbering + 1
            Dim XnewNumbering As Integer = XcFan + 80 * Math.Cos(a)
            Dim YnewNumbering As Integer = YcFan + 80 * Math.Sin(a)
            Dim XlineBottom As Integer = XcFan + 68 * Math.Cos(a)
            Dim ylineBottom As Integer = YcFan + 68 * Math.Sin(a)
            ' points for the markings
            gFanRPM.DrawLine(myBlackPen, XnewNumbering, YnewNumbering, XlineBottom,
ylineBottom)
            ' g.DrawEllipse(myBlackPen, Xnew, Ynew, 2, 2)
        Next
```

```
    End Sub

    Private Sub drawTHINlinesFan()

        gFanRPM = Me.pbFanRPM.CreateGraphics

        gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

        gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        ' Dim myBlackPen As New Pen(Brushes.DarkGray, 2)

        Dim a As Double

        Dim dialNumbering As Integer = 0

        For a = 3.14 To 2 * 3.14 Step (3.14 / 10) * 2

            dialNumbering = dialNumbering + 1

            Dim XnewNumbering As Integer = XcFan + 80 * Math.Cos(a)

            Dim YnewNumbering As Integer = YcFan + 80 * Math.Sin(a)

            Dim XlineBottom As Integer = XcFan + 70 * Math.Cos(a)

            Dim ylineBottom As Integer = YcFan + 70 * Math.Sin(a)

            ' points for the markings

            gFanRPM.DrawLine(Pens.Black, XnewNumbering, YnewNumbering, XlineBottom,
ylineBottom)

            ' g.DrawEllipse(myBlackPen, Xnew, Ynew, 2, 2)

        Next

    End Sub

    ' numbering of the scale

    Public Sub DrawNumberingFan()

        Dim formGraphics As System.Drawing.Graphics = Me.pbFanRPM.CreateGraphics()

        Dim drawString As String = ""

        Dim drawFont As New System.Drawing.Font("Arial", 8)

        Dim drawBrush As New  _

            System.Drawing.SolidBrush(System.Drawing.Color.Black)

        'Dim x As Single = 10

        ' Dim y As Single

        Dim drawFormat As New System.Drawing.StringFormat

        drawFormat.FormatFlags = StringFormatFlags.LineLimit

        Dim i As Integer = 1
```

```
    Dim arrayDrawStirng() As String = {"30", "40", "50", "60", "70", "80", "90"}
    For x = 0 To 180 Step 30
        drawString = arrayDrawStirng(i - 1)
        formGraphics.DrawString(drawString, drawFont, drawBrush, _
        x, 120, drawFormat)
        i = i + 1
    Next
    drawFont.Dispose()
    drawBrush.Dispose()
    formGraphics.Dispose()
End Sub
'
Private Sub captionFan()
    Dim formGraphics As System.Drawing.Graphics = Me.pbFanRPMCaption.CreateGraphics()
    Dim drawString As String = "Fan RPM"
    Dim drawString2a As String = "("
    Dim drawString2 As String = "500-700"
    Dim drawString3 As String = " rpm )"
    Dim drawFont As New System.Drawing.Font("Arial", 10)
    Dim drawFont2 As New System.Drawing.Font("Arial", 11, FontStyle.Bold)
    Dim drawBrush As New  _
        System.Drawing.SolidBrush(System.Drawing.Color.Black)
    Dim drawFormat As New System.Drawing.StringFormat
    'drawFormat.LineAlignment = StringAlignment.Center
    'drawFormat.Alignment = StringAlignment.Center
    formGraphics.DrawString(drawString, drawFont, drawBrush, _
    30, 0, drawFormat)
    formGraphics.DrawString(drawString2a, drawFont, drawBrush, _
    10, 15, drawFormat)
    formGraphics.DrawString(drawString2, drawFont2, drawBrush, _
    15, 15, drawFormat)
```

```
    formGraphics.DrawString(drawString3, drawFont, drawBrush, _
    70, 15, drawFormat)
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    drawFont.Dispose()
    drawBrush.Dispose()
    formGraphics.Dispose()
End Sub
Private Sub incTriFan()
    gFanRPM = Me.pbFanRPM.CreateGraphics
    gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim x As Integer = 65
    Dim y As Integer = 5
    'g.FillEllipse(Brushes.Silver, x, y, w, h)
    Dim incTri() As Point = {New Point(x, y), _
                             New Point(x, y + 16), _
                             New Point(x + 10, y + 8)}
    gFanRPM.FillPolygon(Brushes.Goldenrod, incTri)
    gFanRPM.Dispose()
End Sub
' Decreasing indicating triangle
Private Sub decTriFan()
    gFanRPM = Me.pbFanRPM.CreateGraphics
    gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim x As Integer = 65
    Dim y As Integer = 5
    Dim incTri() As Point = {New Point(x + 10, y), _
                             New Point(x, y + 8), _
                             New Point(x + 10, y + 16)}
```

```
        gFanRPM.FillPolygon(Brushes.Goldenrod, incTri)

        gFanRPM.Dispose()

    End Sub

    Private Sub stopTrendFan()

        gFanRPM = Me.pbFanRPM.CreateGraphics

        gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

        gFanRPM.SmoothingMode = Drawing2D.SmoothingMode.HighQuality

        Dim x As Integer = 65

        Dim y As Integer = 5

        Dim w As Integer = 14

        Dim h As Integer = 14

        gFanRPM.FillRectangle(Brushes.Goldenrod, x, y, w, h)

        gFanRPM.Dispose()

    End Sub

    Private Sub fanRpmNew1()

        counterFanRPM = counterFanRPM + 1

        If counterFanRPM <= 29 Then

            needleFan(270)

            gFanRPM.Dispose()

        ElseIf counterFanRPM <= 300 Then

            Dim needleInput As Integer = 270 + (counterFanRPM - 29) * 1.7

            needleFan(needleInput)

            gFanRPM.Dispose()

        Else

        End If

    End Sub

    Private Sub fanRpmNew2()

        counterFanRPM = counterFanRPM + 1

        ' value of the counter/progressbar1.value = (scaleReading-10)*10/6

        If counterFanRPM <= 300 Then

            '  drawDialNumbering()
```

```vb
            Dim needleInput As Integer = 270 + counterFanRPM * 1.7

            needleFan(needleInput)

            gFanRPM.Dispose()

        Else


        End If

    End Sub

    Private Sub fanRpmNew3()

        counterFanRPM = counterFanRPM + 1

        If counterFanRPM <= 12 Then

            needleFan(270)

            gFanRPM.Dispose()

        ElseIf counterFanRPM <= 300 Then

            Dim needleInput As Integer = 270 + (counterFanRPM - 12) * 1.7

            needleFan(needleInput)

            gFanRPM.Dispose()

        End If

    End Sub

#End Region

#Region "Tool Pressure"

    ''''''' Tool Pressure

    Public Property getCounterToolPressureNew As Integer

    Dim RandomTp As Integer

    Dim incrementPBarTp As Integer

    Dim counterTp As Integer

    Dim gTp As Graphics

    Dim hTp As Integer

    ' Function for changing the color of the progress bar:

    Declare Function SendMessageTp Lib "user32" Alias "SendMessageA" _

        (ByVal hwnd As Integer, ByVal wMsg As Integer, ByVal wParam As Integer, ByVal lParam
As Integer) As Integer
```

```
'change6
Dim rnToolPressure As Integer
' Dim rnToolPressure As Integer = 1
Private Sub TimerTp_Tick(sender As Object, e As EventArgs) Handles timerTp.Tick
    captionTp()
    drawTpScale()
    If rnToolPressure = 1 Then
        toolPressureNew1()
    ElseIf rnToolPressure = 2 Then
        toolPressureNew2()
    ElseIf rnToolPressure = 3 Then
        toolPressureNew3()
    End If
End Sub
Private Sub toolPressureNewReset()
    timerIdleTp.Enabled = True
    counterIdleToolPressure = 0
    timerTp.Enabled = False
    timerTpReset.Enabled = False
    intList3.Remove(5)
    start1randomParameterAfterResest()
End Sub
Private Sub elementsLocationSizeTp()
    'ProgressBar1.Left = PictureBox1.Left + 10
    'ProgressBar1.Top = PictureBox1.Top + 85
    'ProgressBar1.Width = 180
    'ProgressBar1.Height = 7
    lbTPreading.Width = 36
    ' lblSeedAppRate.Left = PictureBox1.Left + (190 - lblSeedAppRate.Width) / 2 + 10
    'lbTPreading.Left = pbTp.Left + 100 - (lbTPreading.Width / 2)
    'lbTPreading.Top = pbTp.Top + 5
```

```vb
    lbTPreading.Left = pbTp.Left + 100 - (lbTPreading.Width / 2)
    lbTPreading.Top = pbTp.Top + pbTp.Height
    pbTpTrend.Left = pbTp.Left ' + pbTp.Width / 2 - pbTpTrend.Width / 2
    pbTpTrend.Top = pbTp.Top + pbTp.Height - pbTpTrend.Height
    lbTpUnits.Top = lbTPreading.Top + lbTPreading.Height
    lbTpUnits.Left = lbTPreading.Left + lbTPreading.Width / 2 - lbTpUnits.Width / 2
    pbToolpCaption.Left = pbTp.Left + pbTp.Width / 2 - pbToolpCaption.Width / 2
    pbToolpCaption.Top = pbTp.Top + pbTp.Height + 5
    'lbTpName.Left = pbTp.Left + pbTp.Width / 2 - lbTpName.Width / 2
    'lbTpName.Top = pbTp.Top + pbTp.Height + 5
    lbTpName.Left = pbTp.Left
    lbTpName.Top = pbTp.Top - lbTpName.Height - 5
    pbTpTyne.Left = pbTp.Left + pbTp.Width / 2 - pbTpTyne.Width / 2
    pbTpTyne.Top = pbTp.Top + 1
End Sub
Private Sub drawTyneTp()
End Sub
'Draw seeds
Private Sub lowPressureArrowsTp()
    pbTpTyne.Image = My.Resources.PressureLow
End Sub
Private Sub okayPressureArrowsTp()
    pbTpTyne.Image = My.Resources.PressureOkay
End Sub
Private Sub highPressureArrowsTp()
    pbTpTyne.Image = My.Resources.PressureHigh
End Sub
' progressBar Increasing
Private Sub progressBarIncDecTp(ByVal steps As Integer)
    incrementPBarTp = incrementPBarTp + steps
    gTp = Me.pbTp.CreateGraphics
```

```
        gTp.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gTp.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        'g.Clear(SystemColors.ControlLight)
        Dim xScale As Integer = 10
        Dim yScale As Integer = 84
        Dim wScale As Integer = 0 + incrementPBarTp
        Dim hScale As Integer = 8
        'Gray background of the progress bar
        gTp.FillRectangle(Brushes.LightGray, 10, 84, 180, 8)
        If wScale <= 59 Then
            gTp.FillRectangle(Brushes.OrangeRed, xScale, yScale, wScale, hScale)
            gTp.DrawRectangle(Pens.Gray, 10, 84, 180, 8)
        ElseIf wScale <= 129 Then
            gTp.FillRectangle(Brushes.Green, xScale, yScale, wScale, hScale)
            gTp.DrawRectangle(Pens.Gray, 10, 84, 180, 8)
        ElseIf wScale <= 190 Then
            gTp.FillRectangle(Brushes.OrangeRed, xScale, yScale, wScale, hScale)
            gTp.DrawRectangle(Pens.Gray, 10, 84, 180, 8)
        End If
        gTp.Dispose()
    End Sub
    ' progress bar STOP
    Private Sub progressBarAllInOneTp(ByVal AppRate As Integer)
        Dim wBar As Integer = (AppRate - 30) * 3
        gTp = Me.pbTp.CreateGraphics
        gTp.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gTp.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        If wBar < 0 Then
            gTp.FillRectangle(Brushes.OrangeRed, 10, 84, 0, 8)
            gTp.DrawRectangle(Pens.OrangeRed, 10, 84, 180, 8)
        ElseIf wBar <= 59 Then
```

```
        gTp.FillRectangle(Brushes.OrangeRed, 10, 84, wBar, 8)
        gTp.DrawRectangle(Pens.Gray, 10, 84, 180, 8)
    ElseIf wBar <= 120 Then
        gTp.FillRectangle(Brushes.Green, 10, 84, wBar, 8)
        gTp.DrawRectangle(Pens.Gray, 10, 84, 180, 8)
    ElseIf wBar <= 180 Then
        gTp.FillRectangle(Brushes.OrangeRed, 10, 84, wBar, 8)
        gTp.DrawRectangle(Pens.Gray, 10, 84, 180, 8)
    ElseIf wBar > 180 Then
        gTp.FillRectangle(Brushes.OrangeRed, 10, 84, 180, 8)
        gTp.DrawRectangle(Pens.Gray, 10, 84, 180, 8)
    End If
    '  gTp.Dispose()
End Sub
'draw scale
Private Sub drawTpScale()
    gTp = Me.pbTp.CreateGraphics
    gTp.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gTp.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    'background color of the picturebox
    gTp.Clear(SystemColors.Control)
    Dim myBlackPen As New Pen(Brushes.Black, 2)
    ''bottom label text
    'Scale
    Dim xScale As Integer = 10
    Dim yScale As Integer = 95
    Dim wScale As Integer = 60
    Dim hScale As Integer = 20
    gTp.FillRectangle(Brushes.IndianRed, xScale, yScale, wScale, hScale)
    gTp.FillRectangle(Brushes.YellowGreen, xScale + 60, yScale, wScale, hScale)
    gTp.FillRectangle(Brushes.IndianRed, xScale + 60 + 60, yScale, wScale, hScale)
```

```vb
    For x = 10 To 210 Step 30
        gTp.DrawLine(Pens.Black, x, yScale, x, yScale + 20)
    Next
    For x = 25 To 200 Step 30
        gTp.DrawLine(Pens.Black, x, yScale, x, yScale + 10)
    Next
    DrawNumberingTp()
    drawTyneTp()
    '  draw1Funnel()
End Sub
' numbering of the scale
Public Sub DrawNumberingTp()
    Dim formGraphics As System.Drawing.Graphics = Me.pbTp.CreateGraphics()
    Dim drawString As String = ""
    Dim drawFont As New System.Drawing.Font("Arial", 8)
    Dim drawBrush As New  _
        System.Drawing.SolidBrush(System.Drawing.Color.Black)
    'Dim x As Single = 10
    ' Dim y As Single
    Dim drawFormat As New System.Drawing.StringFormat
    drawFormat.FormatFlags = StringFormatFlags.LineLimit
    Dim i As Integer = 1
    Dim arrayDrawStirng() As String = {"30", "", "50", "", "70", "", "90"}
    For x = 0 To 180 Step 30
        drawString = arrayDrawStirng(i - 1)
        formGraphics.DrawString(drawString, drawFont, drawBrush, _
        x, 120, drawFormat)
        i = i + 1
    Next
    drawFont.Dispose()
    drawBrush.Dispose()
```

```
    formGraphics.Dispose()
End Sub
'
Private Sub captionTp()
    Dim formGraphics As System.Drawing.Graphics = Me.pbToolpCaption.CreateGraphics()
    Dim drawString As String = "Tool Pressure"
    Dim drawString2a As String = "("
    Dim drawString2 As String = "50-70 "
    Dim drawString3 As String = " N )"
    Dim drawFont As New System.Drawing.Font("Arial", 10)
    Dim drawFont2 As New System.Drawing.Font("Arial", 11, FontStyle.Bold)
    Dim drawBrush As New  _
        System.Drawing.SolidBrush(System.Drawing.Color.Black)
    Dim drawFormat As New System.Drawing.StringFormat
    formGraphics.DrawString(drawString, drawFont, drawBrush, _
    0, 0, drawFormat)
    formGraphics.DrawString(drawString2a, drawFont, drawBrush, _
    10, 15, drawFormat)
    formGraphics.DrawString(drawString2, drawFont2, drawBrush, _
    15, 15, drawFormat)
    formGraphics.DrawString(drawString3, drawFont, drawBrush, _
    55, 15, drawFormat)
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    drawFont.Dispose()
    drawBrush.Dispose()
    formGraphics.Dispose()
End Sub
Private Sub incTriTp()
    gTp = Me.pbTp.CreateGraphics
    gTp.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
```

```vb
    gTp.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    '   g.Clear(SystemColors.Control)
    ' PicBoxArrowIndicator.BackColor = Color.Silver
    Dim x As Integer = pbTp.Left + pbTp.Width / 2 - 8 - 25
    Dim y As Integer = pbTp.Height - 20
    'g.FillEllipse(Brushes.Silver, x, y, w, h)
    Dim incTri() As Point = {New Point(x, y), _
                             New Point(x, y + 16), _
                             New Point(x + 10, y + 8)}
    gTp.FillPolygon(Brushes.Goldenrod, incTri)
    gTp.Dispose()
End Sub
' Decreasing indicating triangle
Private Sub decTriTp()
    gTp = Me.pbTp.CreateGraphics
    gTp.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gTp.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim x As Integer = pbTp.Left + pbTp.Width / 2 - 8 - 25
    Dim y As Integer = pbTp.Height - 20
    Dim incTri() As Point = {New Point(x + 10, y), _
                             New Point(x, y + 8), _
                             New Point(x + 10, y + 16)}
    gTp.FillPolygon(Brushes.Goldenrod, incTri)
    gTp.Dispose()
End Sub
Private Sub stopTrendTp()
    gTp = Me.pbTp.CreateGraphics
    gTp.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gTp.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim w As Integer = 14
    Dim h As Integer = 14
```

```
        Dim x As Integer = pbTp.Left + pbTp.Width / 2 - w - 25
        Dim y As Integer = pbTp.Height - 20
        gTp.FillRectangle(Brushes.Goldenrod, x, y, w, h)
        gTp.Dispose()
    End Sub
    Private Sub toolPressureNew1()
        counterTp = counterTp + 1
        'drawTpScale()
        If counterTp <= 20 Then
            stopTrendTp()
            okayPressureArrowsTp()
            progressBarAllInOneTp(60)
            lbTPreading.Text = Format(60, 0)
        ElseIf counterTp <= 30 Then
            incTriTp()
            okayPressureArrowsTp()
            progressBarAllInOneTp(60 + counterTp - 20)
            lbTPreading.Text = Format(60 + counterTp - 20)
        Else
            incTriTp()
            highPressureArrowsTp()
            ' okayPressureArrowsTp()
            progressBarAllInOneTp(60 + counterTp - 20)
            lbTPreading.Text = Format(60 + counterTp - 20)
        End If
    End Sub
    ' Decreasing rate
    Private Sub toolPressureNew2()
        counterTp = counterTp + 1
        If counterTp <= 10 Then
            incTriTp()
```

```
        ' highPressureArrowsTp()
        okayPressureArrowsTp()
        progressBarAllInOneTp(60 + counterTp)
        lbTPreading.Text = Format(60 + counterTp)
    ElseIf counterTp <= 300 Then
        incTriTp()
        highPressureArrowsTp()
        'okayPressureArrowsTp()
        progressBarAllInOneTp(60 + counterTp)
        lbTPreading.Text = Format(60 + counterTp)
    Else
    End If
End Sub
Private Sub toolPressureNew3()
    counterTp = counterTp + 1
    'drawTpScale()
    If counterTp <= 10 Then
        stopTrendTp()
        okayPressureArrowsTp()
        progressBarAllInOneTp(60)
        lbTPreading.Text = Format(60, 0)
    ElseIf counterTp <= 20 Then
        decTriTp()
        okayPressureArrowsTp()
        progressBarAllInOneTp(60 - counterTp + 10)
        lbTPreading.Text = Format(60 - counterTp + 10)
    ElseIf counterTp <= 70 Then  ' corrected/ changed
        ' decTriTp()
        stopTrendTp()
        lowPressureArrowsTp()
        ' okayPressureArrowsTp()
```

```
            progressBarAllInOneTp(60 - counterTp + 10)

            lbTPreading.Text = Format(60 - counterTp + 10)

        ElseIf counterTp <= 300 Then

            decTriTp()

            lowPressureArrowsTp()

            ' okayPressureArrowsTp()

            progressBarAllInOneTp(60 - counterTp + 10)

            lbTPreading.Text = Format(0, 0)

        End If

    End Sub

#End Region

#Region "Tool Depth"

    '''''''' ToolDepth

    Public Property getCounterToolDepthNew As Integer

    Dim RandomToolDepth As Integer

    'Dim seedRateSequenceList As New List(Of Integer)

    Dim gToolDepth As Graphics

    Dim gMovingTool As Graphics

    Dim gToolDepthRight As Graphics

    Dim gMovingToolRight As Graphics

    Dim hToolDepth As Integer

    Dim counterToolDepth As Integer

    'change7

    Dim rnToolDepth As Integer

    ' Dim rnToolDepth As Integer = 1

    Private Sub TimerToolDepth_Tick(sender As Object, e As EventArgs) Handles
timerToolDepth.Tick

        If rnToolDepth = 1 Then

            toolDepthNew1()

        ElseIf rnToolDepth = 2 Then

            toolDepthNew2()

        ElseIf rnToolDepth = 3 Then
```

```
        toolDepthNew3()
    End If
    hToolDepth = hToolDepth + 1
End Sub
Private Sub toolDepthNewReset()
    timerIdleToolDepth.Enabled = True
    counterIdleToolDepth = 0
    timerToolDepth.Enabled = False
    timerToolDreset.Enabled = False
    intList3.Remove(6)
    start1randomParameterAfterResest()
End Sub
Private Sub drawTanksScaleToolDepth()
    Try
        gToolDepth = Me.pbToolDepth.CreateGraphics()
        gToolDepthRight = Me.pbToolDepthRight.CreateGraphics()

        gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gToolDepthRight.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        '  g.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        Dim myBlackPen As New Pen(Brushes.Black, 2)
        'lbToolDReading.Left = pbToolDepth.Left + 90 - lbToolDReading.Width / 2
        'lbToolDReading.Top = pbToolDepth.Top + (125)
        ''bottom label te
        'Scale
        gToolDepth.FillRectangle(Brushes.IndianRed, 30, 0, 20, 63)
        gToolDepth.FillRectangle(Brushes.YellowGreen, 30, 63, 20, 42)
        gToolDepth.FillRectangle(Brushes.IndianRed, 30, 63 + 42, 20, 63 + 21)
        gToolDepthRight.FillRectangle(Brushes.IndianRed, 30, 0, 20, 63)
        gToolDepthRight.FillRectangle(Brushes.YellowGreen, 30, 63, 20, 42)
        gToolDepthRight.FillRectangle(Brushes.IndianRed, 30, 63 + 42, 20, 63 + 21)
```

```
        For y = 0 To 168 Step 21
            gToolDepth.DrawLine(Pens.Black, 30, y, 45, y)
            gToolDepthRight.DrawLine(Pens.Black, 30, y, 45, y)
        Next
        For y = 0 To 177 Step 21
            gToolDepth.DrawLine(Pens.Black, 30, y + 10, 37, y + 10)
            gToolDepthRight.DrawLine(Pens.Black, 30, y + 10, 37, y + 10)
        Next

        DrawNumberingToolDepth()
        gToolDepth.Dispose()
        gToolDepthRight.Dispose()
    Catch fileException As Exception
        Throw fileException
    End Try
End Sub
' numbering of the scale
Public Sub DrawNumberingToolDepth()
    Dim formGraphics As System.Drawing.Graphics = Me.pbToolDepth.CreateGraphics()
    Dim formGraphicsRight = Me.pbToolDepthRight.CreateGraphics
    Dim drawString As String = ""
    Dim drawFont As New System.Drawing.Font("Arial", 8)
    Dim drawBrush As New  _
        System.Drawing.SolidBrush(System.Drawing.Color.Black)
    Dim x As Single = 10
    Dim y As Single
    Dim drawFormat As New System.Drawing.StringFormat
    drawFormat.FormatFlags = StringFormatFlags.LineLimit
    Dim i As Integer = 1
    Dim arrayDrawStirng() As String = {"2", "", "6", "", "10", "", "", "16"}
    For y = 15 To 170 Step 21 'upto 1 only
```

```
        drawString = arrayDrawStirng(i - 1)
        formGraphics.DrawString(drawString, drawFont, drawBrush, _
        10, y, drawFormat)
        formGraphicsRight.DrawString(drawString, drawFont, drawBrush, _
       10, y, drawFormat)
        i = i + 1
    Next
End Sub
Private Sub drawTyneToolDepth()
    Dim xTip = 115
    Dim yTip = 100
    Dim myTip() As Point = {New Point(xTip, yTip), _
                     New Point(xTip - 20, yTip - 20), _
                     New Point(xTip - 15, yTip - 5), _
                      New Point(xTip - 15, yTip + 5), _
                      New Point(xTip - 20, yTip + 20)}
    Dim myBlackPen As New Pen(Brushes.Black, 8)
    gToolDepth = Me.pbToolDepth.CreateGraphics
    gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    gToolDepth.FillRectangle(Brushes.Black, xTip - 35, yTip - 5, 25, 9)
    gToolDepth.DrawArc(myBlackPen, xTip - 50, yTip - 100, 50, 100, 260, -180)
    gToolDepth.FillPolygon(Brushes.Black, myTip)

End Sub
'
Private Sub labeltoolDepth()
    Dim formGraphics, formGraphicsRight As System.Drawing.Graphics
    formGraphics = Me.pbToolDepthCaption.CreateGraphics()
    formGraphicsRight = Me.pbToolDepthCaptionRight.CreateGraphics
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
```

```
    formGraphics.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    formGraphicsRight.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    formGraphicsRight.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    Dim drawString As String = "Tool Depth"
    Dim drawString2a As String = "("
    Dim drawString2 As String = "6-10"
    Dim drawString3 As String = " cm )"
    Dim drawFont As New System.Drawing.Font("Arial", 10)
    Dim drawFont2 As New System.Drawing.Font("Arial", 11, FontStyle.Bold)
    Dim drawBrush As New  _
        System.Drawing.SolidBrush(System.Drawing.Color.Black)
    Dim drawFormat As New System.Drawing.StringFormat
    formGraphics.DrawString(drawString, drawFont, drawBrush, _
    30, 0, drawFormat)
    formGraphics.DrawString(drawString2a, drawFont, drawBrush, _
    28, 15, drawFormat)
    formGraphics.DrawString(drawString2, drawFont2, drawBrush, _
    35, 15, drawFormat)
    formGraphics.DrawString(drawString3, drawFont, drawBrush, _
    70, 15, drawFormat)
    formGraphicsRight.DrawString(drawString, drawFont, drawBrush, _
   30, 0, drawFormat)
    formGraphicsRight.DrawString(drawString2a, drawFont, drawBrush, _
    28, 15, drawFormat)
    formGraphicsRight.DrawString(drawString2, drawFont2, drawBrush, _
    35, 15, drawFormat)
    formGraphicsRight.DrawString(drawString3, drawFont, drawBrush, _
    70, 15, drawFormat)
End Sub
' draw of tool depth
Private Sub toolDepth(ByVal toolDepthStart As Integer, ByVal newCounter As Integer)
```

```
    Dim yStart As Integer = (15 - toolDepthStart) * 159 / 15 + newCounter * 1.5 + 15
    Dim recHeight As Integer = 174 - yStart
    gToolDepth = Me.pbToolDepth.CreateGraphics
    gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    gToolDepth.Clear(SystemColors.Control)
    gToolDepth.FillRectangle(Brushes.DarkGoldenrod, 52, yStart, 76, recHeight)
    Dim depthReading As Double = toolDepthStart - newCounter * 1.5 * 15 / 159
    If depthReading < 5.9 Then
        lbToolDReading.ForeColor = Color.White
        lbToolDReading.BackColor = Color.IndianRed
    ElseIf depthReading < 10.1 Then
        lbToolDReading.ForeColor = Color.White
        lbToolDReading.BackColor = Color.Green
    ElseIf depthReading <= 15 Then
        lbToolDReading.ForeColor = Color.White
        lbToolDReading.BackColor = Color.Red
    End If
    lbToolDReading.Text = FormatNumber(toolDepthStart - newCounter * 1.5 * 15 / 159, 1)
End Sub
Private Sub toolDepthMovingTool(ByVal toolDepth As Integer)
    Dim tdReading = FormatNumber(toolDepth * 16 / 168, 1)
    pbToolDepthTip.Width = 40
    '  pbToolDepthTip.Height = 230
    pbToolDepthTipRight.Width = 40
    '  pbToolDepthTipRight.Height = 230
    Dim width = 15
    Dim xStart As Integer
    xStart = pbToolDepthTip.Width / 2 - width / 2
    Dim yStart = 0
    Dim myTip() As Point = {New Point(xStart - width / 2, yStart + toolDepth), _
```

```
                                New Point(xStart + width + width / 2, yStart + toolDepth), _
                                New Point(xStart + width / 2, yStart + 11 + toolDepth)}
        gMovingTool = Me.pbToolDepthTip.CreateGraphics
        gMovingToolRight = Me.pbToolDepthTipRight.CreateGraphics
        gMovingTool.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gMovingToolRight.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        'gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        gMovingTool.Clear(SystemColors.Control)
        gMovingToolRight.Clear(SystemColors.Control)
        If toolDepth <= 63 Then
            gMovingTool.FillRectangle(Brushes.OrangeRed, xStart, yStart, width, toolDepth)
            gMovingTool.FillPolygon(Brushes.OrangeRed, myTip)
            gMovingToolRight.FillRectangle(Brushes.OrangeRed, xStart, yStart, width,
toolDepth)
            gMovingToolRight.FillPolygon(Brushes.OrangeRed, myTip)
        ElseIf toolDepth <= 105 Then
            gMovingTool.FillRectangle(Brushes.Green, xStart, yStart, width, toolDepth)
            gMovingTool.FillPolygon(Brushes.Green, myTip)
            gMovingToolRight.FillRectangle(Brushes.Green, xStart, yStart, width, toolDepth)
            gMovingToolRight.FillPolygon(Brushes.Green, myTip)
        Else
            gMovingTool.FillRectangle(Brushes.OrangeRed, xStart, yStart, width, toolDepth)
            gMovingTool.FillPolygon(Brushes.OrangeRed, myTip)
            gMovingToolRight.FillRectangle(Brushes.OrangeRed, xStart, yStart, width,
toolDepth)
            gMovingToolRight.FillPolygon(Brushes.OrangeRed, myTip)
        End If

        lbToolDReading.Text = tdReading
        lbToolDReadingRight.Text = tdReading
    End Sub
    Private Sub elementsLocationSizeToolDepth()
```

```
        pbToolDepthTip.Left = pbToolDepth.Left + 55

        pbToolDepthTip.Top = pbToolDepth.Top

        pbToolDepthTipRight.Left = pbToolDepthRight.Left + 55

        pbToolDepthTipRight.Top = pbToolDepthRight.Top

        lbToolDReading.Left = pbToolDepthTip.Left + pbToolDepthTip.Width / 2 -
lbToolDReading.Width / 2 - 5

        lbToolDReading.Top = pbToolDepthTip.Top + pbToolDepthTip.Height

        lbToolDReadingRight.Left = pbToolDepthTipRight.Left + pbToolDepthTipRight.Width / 2 -
lbToolDReadingRight.Width / 2 - 5

        lbToolDReadingRight.Top = pbToolDepthTipRight.Top + pbToolDepthTipRight.Height

        lbTdLeftUnits.Left = lbToolDReading.Left + lbToolDReading.Width / 2 -
lbTdLeftUnits.Width / 2

        lbTdLeftUnits.Top = lbToolDReading.Top + lbToolDReading.Height

        lbTdRightUnits.Left = lbToolDReadingRight.Left + lbToolDReadingRight.Width / 2 -
lbTdRightUnits.Width / 2

        lbTdRightUnits.Top = lbToolDReadingRight.Top + lbToolDReadingRight.Height

        pbToolDepthTrend.Left = lbToolDReading.Left - lbToolDReading.Width - 3

        pbToolDepthTrend.Top = lbToolDReading.Top

        pbToolDepthTrendRight.Left = lbToolDReadingRight.Left - lbToolDReadingRight.Width - 3

        pbToolDepthTrendRight.Top = lbToolDReadingRight.Top


        lbTDleftName.Left = pbToolDepth.Left

        lbTDleftName.Top = pbToolDepth.Top - lbTDleftName.Height - 5

        lbTdNameRight.Left = pbToolDepthRight.Left

        lbTdNameRight.Top = pbToolDepthRight.Top - lbTdNameRight.Height - 5


    End Sub

    Private Sub incTriToolDepth()

        Dim readings As Integer = CInt(lbToolDReading.Text)

        pbToolDepthTrend.Width = 18

        pbToolDepthTrend.Height = 20

        pbToolDepthTrendRight.Width = 18

        pbToolDepthTrendRight.Height = 20
```

```
    gToolDepth = Me.pbToolDepthTrend.CreateGraphics
    gToolDepthRight = Me.pbToolDepthTrendRight.CreateGraphics
    gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    gToolDepthRight.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gToolDepthRight.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    'gToolDepth.Clear(SystemColors.Control)
    'gToolDepth.Clear(Color.Gray)
    'gToolDepthRight.Clear(Color.Gray)
    gToolDepth.Clear(SystemColors.Control)
    gToolDepthRight.Clear(SystemColors.Control)
    ' PicBoxArrowIndicator.BackColor = Color.Silver
    Dim w As Integer = 18
    Dim h As Integer = 20
    'g.FillEllipse(Brushes.Silver, x, y, w, h)
    Dim incTri() As Point = {New Point(w / 2, 0), _
                             New Point(0, h), _
                             New Point(w, h)}
    gToolDepth.FillPolygon(Brushes.Gold, incTri)
    gToolDepthRight.FillPolygon(Brushes.Gold, incTri)
End Sub
' Decreasing indicating triangle
Private Sub decTriToolDepth()
    Dim readings As Integer = CInt(lbToolDReading.Text)
    'pbToolDepthTrend.Left = pbToolDepth.Left + 90 - pbToolDepthTrend.Width / 2
    'pbToolDepthTrend.Top = pbToolDepth.Top + (152)
    pbToolDepthTrend.Width = 20
    pbToolDepthTrend.Height = 20
    pbToolDepthTrendRight.Width = 20
    pbToolDepthTrendRight.Height = 20
    gToolDepth = Me.pbToolDepthTrend.CreateGraphics
```

```
    gToolDepthRight = Me.pbToolDepthTrendRight.CreateGraphics
    gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    gToolDepthRight.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
    gToolDepthRight.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
    'gToolDepth.Clear(Color.White)
    'gToolDepthRight.Clear(Color.White)
    gToolDepth.Clear(SystemColors.Control)
    gToolDepthRight.Clear(SystemColors.Control)
    ' gToolDepth.Clear(SystemColors.Control)
    '   g.Clear(Color.Silver)
    Dim w As Integer = 18
    Dim h As Integer = 20
    'g.FillEllipse(Brushes.Silver, x, y, w, h)
    Dim incTri() As Point = {New Point(0, 0), _
                             New Point(w, 0), _
                             New Point(w / 2, h)}
    gToolDepth.FillPolygon(Brushes.Gold, incTri)
    gToolDepthRight.FillPolygon(Brushes.Gold, incTri)
End Sub
' stop of the trend , reading is not decreasing or increasing
Private Sub stopTrendToolDepth()
    Dim readings As Integer = CInt(lbToolDReading.Text)
    'pbToolDepthTrend.Left = pbToolDepth.Left + 90 - pbToolDepthTrend.Width / 2
    'pbToolDepthTrend.Top = pbToolDepth.Top + (152)
    pbToolDepthTrend.Width = 20
    pbToolDepthTrend.Height = 20
    pbToolDepthTrendRight.Width = 20
    pbToolDepthTrendRight.Height = 20
    gToolDepth = Me.pbToolDepthTrend.CreateGraphics
    gToolDepthRight = Me.pbToolDepthTrendRight.CreateGraphics
```

```
        gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gToolDepth.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        gToolDepthRight.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gToolDepthRight.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        '  gToolDepth.Clear(Color.DarkGoldenrod)
        ' g.Clear(Color.Silver)
        gToolDepth.Clear(SystemColors.Control)
        gToolDepthRight.Clear(SystemColors.Control)
        Dim wBig As Integer = 20
        Dim hBig As Integer = 20
        Dim w As Integer = wBig / 2
        Dim h As Integer = hBig / 2
        Dim x As Integer = (wBig - w) / 2
        Dim y As Integer = (hBig - h) / 2
        gToolDepth.FillRectangle(Brushes.Gold, x, y, w, h)
        gToolDepthRight.FillRectangle(Brushes.Gold, x, y, w, h)
    End Sub
    ' Draw tanks and scale
    Private Sub toolDepthNew1()
        labeltoolDepth()
        counterToolDepth = counterToolDepth + 1
        If counterToolDepth <= 300 Then
            ' toolDepth(15, counterToolDepth)
            '   decTriToolDepth()
        Else
        End If
        drawTanksScaleToolDepth()
        'drawTyneToolDepth()
        toolDepthMovingTool(counterToolDepth + 69)
    End Sub
    Private Sub toolDepthNew2()
```

```
    '  incTri()
    ' drawTanksScale()
    labeltoolDepth()
    counterToolDepth = counterToolDepth + 1
    If counterToolDepth <= 8 Then
        ' toolDepth(15, counterToolDepth)
        toolDepthMovingTool(90)
        ' decTriToolDepth()
        '  stopTrendToolDepth()
    ElseIf counterToolDepth <= 300 Then
        toolDepthMovingTool(counterToolDepth - 8 + 90)
        ' decTriToolDepth()
    Else
    End If
    drawTanksScaleToolDepth()
    'drawTyneToolDepth()
End Sub
Private Sub toolDepthNew3()
    '  incTri()
    ' drawTanksScale()
    labeltoolDepth()
    counterToolDepth = counterToolDepth + 1
    If counterToolDepth <= 300 Then
        ' toolDepth(15, counterToolDepth)
        toolDepthMovingTool(counterToolDepth + 82)
        '  decTriToolDepth()
        '  stopTrendToolDepth()
    Else
    End If
    drawTanksScaleToolDepth()
    'drawTyneToolDepth()
```

```
    End Sub

#End Region

#Region "Blockage"

    ''''''''' Blockage]]]]]]]]

    Public Property getCounterBlockageNew As Integer

    Dim RandomBlockage As Integer

    'Dim seedRateSequenceList As New List(Of Integer)

    Dim gBlockage As Graphics

    Dim counterBlockage As Integer

    Private Sub elementsSizeLocaitonBlocakge()

        pbBlockage.Left = pbGBField.Left + pbGBField.Width / 2 - pbBlockage.Width / 2

        lbBlocCaption.Top = pbBlockage.Top - lbBlocCaption.Height - 5

        lbBlocCaption.Left = pbBlockage.Left

    End Sub

    Dim rnBlockage As Integer = 1

    Private Sub TimerBlockage_Tick(sender As Object, e As EventArgs) Handles
timerBlockage.Tick

        If counterBlockage = 1 Then

            blockageNewBackDots()

        End If

        If rnBlockage = 1 Then

            ' blockageNewBackDots()

            blockageNew1()

        ElseIf rnBlockage = 2 Then

            blockageNew2()

        ElseIf rnBlockage = 3 Then

            blockageNew3()

        End If

        counterBlockage = counterBlockage + 1

    End Sub

    Private Sub blockageNewReset()

        counterBlockage = counterBlockage + 1
```

```
        If counterBlockage = 1 Then
            blockageNewBackDots()
        End If
        ' value of the counter/progressbar1.value = (scaleReading-10)*10/6
        If counterBlockage > 30 Then
            timerBlockage.Enabled = True
            '  counterBlockage = 0
            Randomize()
            rnBlockage = CInt(Math.Floor((2) * Rnd())) + 2
            timerBlockageReset.Enabled = False
        End If
    End Sub
    Private Sub blockageNewBackDots()
        gBlockage = Me.pbBlockage.CreateGraphics
        gBlockage.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias
        gBlockage.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        For x = 10 To 360 Step 40
            gBlockage.FillEllipse(Brushes.Green, x, 10, 25, 25)
            ' gBlockage.DrawEllipse(Pens.Black, x, 10, 25, 25)
            gBlockage.FillEllipse(Brushes.Green, x, 50, 25, 25)
        Next
    End Sub
    Private Sub blockageNew1()
        Dim myBlackPen As New Pen(Color.Black, 2)
        Dim myCrimpsonPen As New Pen(Brushes.Crimson, 3)
        Dim myYellowPen As New Pen(Brushes.Gold, 3)
        gBlockage = Me.pbBlockage.CreateGraphics
        If counterBlockage > 30 Then
            gBlockage.FillRectangle(Brushes.OrangeRed, 90, 10, 25, 25)
            gBlockage.DrawRectangle(myBlackPen, 90, 10, 25, 25)
            gBlockage.DrawLine(myYellowPen, 90, 10, 115, 35)
```

```
            gBlockage.DrawLine(myYellowPen, 90, 35, 115, 10)
        End If
    End Sub
    Private Sub blockageNew2()
        Dim myBlackPen As New Pen(Color.Black, 2)
        Dim myCrimpsonPen As New Pen(Brushes.Crimson, 3)
        Dim myYellowPen As New Pen(Brushes.Gold, 3)
        gBlockage = Me.pbBlockage.CreateGraphics
        gBlockage.FillRectangle(Brushes.OrangeRed, 210, 10, 25, 25)
        gBlockage.DrawRectangle(myBlackPen, 210, 10, 25, 25)
        gBlockage.DrawLine(myYellowPen, 210, 10, 235, 35)
        gBlockage.DrawLine(myYellowPen, 210, 35, 235, 10)
    End Sub
    Private Sub blockageNew3()
        Dim myBlackPen As New Pen(Color.Black, 2)
        Dim myCrimpsonPen As New Pen(Brushes.Crimson, 3)
        Dim myYellowPen As New Pen(Brushes.Gold, 3)
        gBlockage = Me.pbBlockage.CreateGraphics
        gBlockage.FillRectangle(Brushes.OrangeRed, 290, 10, 25, 25)
        gBlockage.DrawRectangle(myBlackPen, 290, 10, 25, 25)
        gBlockage.DrawLine(myYellowPen, 290, 10, 315, 35)
        gBlockage.DrawLine(myYellowPen, 290, 35, 315, 10)
    End Sub
#End Region
#Region "ResponseTime"
    Dim SARreading As Integer
    Dim FARreading As Integer
    Dim TanksReading As Integer
    Dim TpReading As Integer
    Dim speedReading As Integer
    Dim guidanceReading As Integer
```

```
    Dim fanRPMreading As Integer

    Dim toolDepthReading As Integer

    Dim blockageReading As Integer

    Private Sub timerResponseTime_Tick(sender As Object, e As EventArgs) Handles
timerResponseTime.Tick

        SARreading = CInt(lbSARreading.Text)

        FARreading = CInt(lbFARnewReading.Text)

        TanksReading = CInt(lbTanksNewReading.Text.Substring(0, 2))

        TpReading = CInt(lbTPreading.Text)

        speedReading = CInt(lbSpeedNewReading.Text)

        fanRPMreading = CInt(lbFanRPMreading.Text)

        toolDepthReading = CInt(lbToolDReading.Text)

        guidanceReading = counterGb

        blockageReading = counterBlockage

        If SARreading > 50 Then

            resTimeSAR = resTimeSAR + 1

        ElseIf SARreading < 30 Then

            resTimeSAR = resTimeSAR + 1

        End If

        If FARreading > 100 Then

            resTimeFAR = resTimeFAR + 1

        ElseIf FARreading < 60 Then

            resTimeFAR = resTimeFAR + 1

        End If

        If TanksReading < 10 Then

            resTimeTanks = resTimeTanks + 1

        End If

        If TpReading > 70 Then

            resTimeTp = resTimeTp + 1

        ElseIf TpReading < 50 Then

            resTimeTp = resTimeTp + 1

        End If
```

```
        If speedReading > 10 Then

            resTimeSpeed = resTimeSpeed + 1

        ElseIf speedReading < 6 Then

            resTimeSpeed = resTimeSpeed + 1

        End If

        If fanRPMreading > 700 Then

            resTimeFanRPM = resTimeFanRPM + 1

        ElseIf fanRPMreading < 500 Then

            resTimeFanRPM = resTimeFanRPM + 1

        End If

        If toolDepthReading > 10 Then

            resTimeToolDepth = resTimeToolDepth + 1

        ElseIf toolDepthReading < 6 Then

            resTimeToolDepth = resTimeToolDepth + 1

        End If

        If guidanceReading > 7 Then

            resTimeGuidance = resTimeGuidance + 1

        End If

        If blockageReading > 30 Then

            resTimeBlockage = resTimeBlockage + 1

        End If

    End Sub

#End Region

#Region "Click Events"

    Private Sub pbTanksNew_MouseEnter(sender As Object, e As EventArgs) Handles
pbTanksNew.MouseEnter

        Cursor = Cursors.Hand

    End Sub

    Private Sub pbTanksNew_MouseLeave(sender As Object, e As EventArgs) Handles
pbTanksNew.MouseLeave

        Cursor = Cursors.Default

    End Sub
```

```
    Private Sub pbTanksNew_Click(sender As Object, e As EventArgs) Handles pbTanksNew.Click

        Cursor = Cursors.AppStarting

        timerTanksNew.Enabled = False

        timerTanksReset.Enabled = True

        counterTanksNew = 0

        Try

            Dim filePath As String

            filePath =
System.IO.Path.Combine(My.Computer.FileSystem.SpecialDirectories.MyDocuments,

                                                  "ConsolidatedDisplayFile.txt")

            My.Computer.FileSystem.WriteAllText(filePath, vbCrLf & "SystemTime: " & Now() &
vbCrLf & "Tanks responseTime(cs): " & resTimeTanks & vbCrLf, True)

            ' My.Computer.FileSystem.WriteAllText(filePath, "#" & num1 & " ", True)

        Catch fileException As Exception

            Throw fileException

        End Try

        resTimeTanks = 0

    End Sub

    Private Sub timerTanksReset_Tick(sender As Object, e As EventArgs) Handles
timerTanksReset.Tick

        counterTanksNew = counterTanksNew + 1

        TanksReset()

    End Sub

    Private Sub pbSeedAppRateNew_MouseEnter(sender As Object, e As EventArgs) Handles
pbSeedAppRateNew.MouseEnter

        Cursor = Cursors.Hand

    End Sub

    Private Sub pbSeedAppRateNew_MouseLeave(sender As Object, e As EventArgs) Handles
pbSeedAppRateNew.MouseLeave

        Cursor = Cursors.Default

    End Sub

    Private Sub pbSeedAppRateNew_Click(sender As Object, e As EventArgs) Handles
pbSeedAppRateNew.Click

        Cursor = Cursors.AppStarting
```

```
        timerSARnew.Enabled = False

        timerSARreset.Enabled = True

        counterSARnew = 0

        gSARnew = Me.pbSeedAppRateNew.CreateGraphics

        gSARnew.Clear(SystemColors.Control)

        Try

            Dim filePath As String

            filePath =
System.IO.Path.Combine(My.Computer.FileSystem.SpecialDirectories.MyDocuments,

                                                    "ConsolidatedDisplayFile.txt")

            My.Computer.FileSystem.WriteAllText(filePath, vbCrLf & "SystemTime: " & Now() &
vbCrLf & "SAR responseTime(cs): " & resTimeSAR & vbCrLf, True)

            ' My.Computer.FileSystem.WriteAllText(filePath, "#" & num1 & " ", True)

        Catch fileException As Exception

            Throw fileException

        End Try

        resTimeSAR = 0

    End Sub

    Private Sub timerSARreset_Tick(sender As Object, e As EventArgs) Handles
timerSARreset.Tick

        seedAppRateReset()

    End Sub

    Private Sub pbFAR_MouseEnter(sender As Object, e As EventArgs) Handles pbFAR.MouseEnter

        Cursor = Cursors.Hand

    End Sub

    Private Sub pbFAR_MouseLeave(sender As Object, e As EventArgs) Handles pbFAR.MouseLeave

        Cursor = Cursors.Default

    End Sub

    Private Sub pbFAR_Click(sender As Object, e As EventArgs) Handles pbFAR.Click

        Cursor = Cursors.AppStarting

        timerFAR.Enabled = False

        timerFARreset.Enabled = True

        counterFAR = 0
```

```
        gFAR = Me.pbFAR.CreateGraphics

        gFAR.Clear(SystemColors.Control)

        Try

            Dim filePath As String

            filePath =
System.IO.Path.Combine(My.Computer.FileSystem.SpecialDirectories.MyDocuments,

                                              "ConsolidatedDisplayFile.txt")

            My.Computer.FileSystem.WriteAllText(filePath, vbCrLf & "SystemTime: " & Now() &
vbCrLf & "FAR responseTime(cs): " & resTimeFAR & vbCrLf, True)

            ' My.Computer.FileSystem.WriteAllText(filePath, "#" & num1 & " ", True)

        Catch fileException As Exception

            Throw fileException

        End Try

        resTimeFAR = 0

    End Sub

    Private Sub timerFARreset_Tick(sender As Object, e As EventArgs) Handles
timerFARreset.Tick

        FARNewReset()

    End Sub

    ''' <summary> speed

    ''' ''''''''''''''''''

    ''' </summary>

    ''' <param name="sender"></param>

    ''' <param name="e"></param>

    ''' <remarks></remarks>

    Private Sub pbSpeedNew_MouseEnter(sender As Object, e As EventArgs) Handles
pbSpeedNew.MouseEnter

        Cursor = Cursors.Hand

    End Sub

    Private Sub pbSpeedNew_MouseLeave(sender As Object, e As EventArgs) Handles
pbSpeedNew.MouseLeave

        Cursor = Cursors.Default

    End Sub

    Private Sub pbSpeedNew_Click(sender As Object, e As EventArgs) Handles pbSpeedNew.Click
```

```
        Cursor = Cursors.AppStarting

        timerSpeedNew.Enabled = False

        timerSpeedReset.Enabled = True

        counterSpeedNew = 0

        Try

            Dim filePath As String

            filePath =
System.IO.Path.Combine(My.Computer.FileSystem.SpecialDirectories.MyDocuments,

                                              "ConsolidatedDisplayFile.txt")

            My.Computer.FileSystem.WriteAllText(filePath, vbCrLf & "SystemTime: " & Now() &
vbCrLf & "Speed responseTime(cs): " & resTimeSpeed & vbCrLf, True)

            ' My.Computer.FileSystem.WriteAllText(filePath, "#" & num1 & " ", True)

        Catch fileException As Exception

            Throw fileException

        End Try

        resTimeSpeed = 0

    End Sub

    Private Sub timerSpeedReset_Tick(sender As Object, e As EventArgs) Handles
timerSpeedReset.Tick

        speedNewReset()

    End Sub

    Private Sub pbTp_Click(sender As Object, e As EventArgs) Handles pbTp.Click

        Cursor = Cursors.AppStarting

        timerTp.Enabled = False

        timerTpReset.Enabled = True

        counterTp = 0

        gTp = Me.pbTp.CreateGraphics

        gTp.Clear(SystemColors.Control)

        Try

            Dim filePath As String

            filePath =
System.IO.Path.Combine(My.Computer.FileSystem.SpecialDirectories.MyDocuments,

                                              "ConsolidatedDisplayFile.txt")
```

```
            My.Computer.FileSystem.WriteAllText(filePath, vbCrLf & "SystemTime: " & Now() &
vbCrLf & "ToolPressure responseTime(cs): " & resTimeTp & vbCrLf, True)

            ' My.Computer.FileSystem.WriteAllText(filePath, "#" & num1 & " ", True)

        Catch fileException As Exception

            Throw fileException

        End Try

        resTimeTp = 0

    End Sub

    Private Sub pbTp_MouseEnter(sender As Object, e As EventArgs) Handles pbTp.MouseEnter

        Cursor = Cursors.Hand

    End Sub

    Private Sub pbTp_MouseLeave(sender As Object, e As EventArgs) Handles pbTp.MouseLeave

        Cursor = Cursors.Default

    End Sub

    Private Sub timerTpReset_Tick(sender As Object, e As EventArgs) Handles timerTpReset.Tick

        captionTp()

        drawTpScale()

        toolPressureNewReset()

    End Sub

    Private Sub pbToolDepth_Click(sender As Object, e As EventArgs) Handles pbToolDepth.Click

        Cursor = Cursors.AppStarting

        timerToolDepth.Enabled = False

        timerToolDreset.Enabled = True

        counterToolDepth = 0

        gToolDepth = Me.pbToolDepth.CreateGraphics

        gToolDepth.Clear(SystemColors.Control)

        gToolDepthRight = Me.pbToolDepthRight.CreateGraphics

        gToolDepthRight.Clear(SystemColors.Control)

        drawTanksScaleToolDepth()

        Try

            Dim filePath As String
```

```
            filePath =
System.IO.Path.Combine(My.Computer.FileSystem.SpecialDirectories.MyDocuments,

                                              "ConsolidatedDisplayFile.txt")

            My.Computer.FileSystem.WriteAllText(filePath, vbCrLf & "SystemTime: " & Now() &
vbCrLf & "ToolDepthLeft responseTime(cs): " & resTimeToolDepth & vbCrLf, True)

            ' My.Computer.FileSystem.WriteAllText(filePath, "#" & num1 & " ", True)

        Catch fileException As Exception

            Throw fileException

        End Try

        resTimeToolDepth = 0

    End Sub

    Private Sub pbToolDepth_MouseEnter(sender As Object, e As EventArgs) Handles
pbToolDepth.MouseEnter

        Cursor = Cursors.Hand

    End Sub

    Private Sub pbToolDepth_MouseLeave(sender As Object, e As EventArgs) Handles
pbToolDepth.MouseLeave

        Cursor = Cursors.Default

    End Sub

    Private Sub timerToolDreset_Tick(sender As Object, e As EventArgs) Handles
timerToolDreset.Tick

        toolDepthNewReset()

    End Sub

    ''''''''''''''''Fan RPM

    Private Sub pbFanRPM_Click(sender As Object, e As EventArgs) Handles pbFanRPM.Click

        Cursor = Cursors.AppStarting

        timerFanRPM.Enabled = False

        timerFanRPMreset.Enabled = True

        counterFanRPM = 0

        Try

            Dim filePath As String

            filePath =
System.IO.Path.Combine(My.Computer.FileSystem.SpecialDirectories.MyDocuments,

                                              "ConsolidatedDisplayFile.txt")
```

```
            My.Computer.FileSystem.WriteAllText(filePath, vbCrLf & "SystemTime: " & Now() & 
vbCrLf & "FanRPM responseTime(cs): " & resTimeFanRPM & vbCrLf, True)

            ' My.Computer.FileSystem.WriteAllText(filePath, "#" & num1 & " ", True)

        Catch fileException As Exception

            Throw fileException

        End Try

        resTimeFanRPM = 0

        ' gFAR = Me.pbFAR.CreateGraphics

        ' gFanRPM.Clear(SystemColors.Control)

    End Sub

    Private Sub pbFanRPM_MouseEnter(sender As Object, e As EventArgs) Handles 
pbFanRPM.MouseEnter

        Cursor = Cursors.Hand

    End Sub

    Private Sub pbFanRPM_MouseLeave(sender As Object, e As EventArgs) Handles 
pbFanRPM.MouseLeave

        Cursor = Cursors.Default

    End Sub

    Private Sub timerFanRPMreset_Tick(sender As Object, e As EventArgs) Handles 
timerFanRPMreset.Tick

        fanRpmNewReset()

    End Sub

    Private Sub pbBlockage_Click(sender As Object, e As EventArgs) Handles pbBlockage.Click

        Cursor = Cursors.AppStarting

        timerBlockage.Enabled = False

        timerBlockageReset.Enabled = True

        counterBlockage = 0

        gBlockage = Me.pbBlockage.CreateGraphics

        gBlockage.Clear(SystemColors.Control)

        Try

            Dim filePath As String

            filePath = 
System.IO.Path.Combine(My.Computer.FileSystem.SpecialDirectories.MyDocuments, 

                                              "ConsolidatedDisplayFile.txt")
```

```
            My.Computer.FileSystem.WriteAllText(filePath, vbCrLf & "SystemTime: " & Now() &
vbCrLf & "Blockage responseTime(cs): " & resTimeBlockage & vbCrLf, True)

            ' My.Computer.FileSystem.WriteAllText(filePath, "#" & num1 & " ", True)

        Catch fileException As Exception

            Throw fileException

        End Try

        resTimeBlockage = 0

    End Sub

    Private Sub pbBlockage_MouseEnter(sender As Object, e As EventArgs) Handles
pbBlockage.MouseEnter

        Cursor = Cursors.Hand

    End Sub

    Private Sub pbBlockage_MouseLeave(sender As Object, e As EventArgs) Handles
pbBlockage.MouseLeave

        Cursor = Cursors.Default

    End Sub

    Private Sub timerBlockageReset_Tick(sender As Object, e As EventArgs) Handles
timerBlockageReset.Tick

        blockageNewReset()

    End Sub



    Private Sub pbToolDepthRight_Click(sender As Object, e As EventArgs) Handles
pbToolDepthRight.Click

        Cursor = Cursors.AppStarting

        timerToolDepth.Enabled = False

        timerToolDreset.Enabled = True

        counterToolDepth = 0

        gToolDepth = Me.pbToolDepth.CreateGraphics

        gToolDepth.Clear(SystemColors.Control)

        gToolDepthRight = Me.pbToolDepthRight.CreateGraphics

        gToolDepthRight.Clear(SystemColors.Control)

        drawTanksScaleToolDepth()

        Try

            Dim filePath As String
```

```vb
            filePath =
System.IO.Path.Combine(My.Computer.FileSystem.SpecialDirectories.MyDocuments,

                                                "ConsolidatedDisplayFile.txt")

            My.Computer.FileSystem.WriteAllText(filePath, vbCrLf & "SystemTime: " & Now() &
vbCrLf & "ToolDepthRight responseTime(cs): " & resTimeTp & vbCrLf, True)

            ' My.Computer.FileSystem.WriteAllText(filePath, "#" & num1 & " ", True)

        Catch fileException As Exception

            Throw fileException

        End Try

        resTimeToolDepth = 0

    End Sub

    Private Sub pbToolDepthRight_MouseEnter(sender As Object, e As EventArgs) Handles
pbToolDepthRight.MouseEnter

        Cursor = Cursors.Hand

    End Sub

    Private Sub pbToolDepthRight_MouseLeave(sender As Object, e As EventArgs) Handles
pbToolDepthRight.MouseLeave

        Cursor = Cursors.Default

    End Sub

    Private Sub pbField_Click(sender As Object, e As EventArgs) Handles pbGBField.Click

        Cursor = Cursors.AppStarting

        counterGb = 0

        Try

            Dim filePath As String

            filePath =
System.IO.Path.Combine(My.Computer.FileSystem.SpecialDirectories.MyDocuments,

                                                "ConsolidatedDisplayFile.txt")

            My.Computer.FileSystem.WriteAllText(filePath, vbCrLf & "SystemTime: " & Now() &
vbCrLf & "GuidanceBar responseTime(cs): " & resTimeGuidance & vbCrLf, True)

            ' My.Computer.FileSystem.WriteAllText(filePath, "#" & num1 & " ", True)

        Catch fileException As Exception

            Throw fileException

        End Try

        resTimeGuidance = 0
```

```
    End Sub

    Private Sub pbField_MouseEnter(sender As Object, e As EventArgs) Handles
pbGBField.MouseEnter

        Cursor = Cursors.Hand

    End Sub

    Private Sub pbField_MouseLeave(sender As Object, e As EventArgs) Handles
pbGBField.MouseLeave

        Cursor = Cursors.Default

    End Sub

#End Region

#Region "Correct Responses"

    Private Function blockageRespose() As String

        Dim blockageResponse = "No"

        If timerBlockage.Enabled = True And rnBlockage = 1 And counterBlockage > 30 Then

            blockageResponse = "Yes"

        ElseIf timerBlockage.Enabled = True And (rnBlockage = 2 Or rnBlockage = 3) Then

            blockageResponse = "Yes"

            'Else

            '    blockageResponse = "No"

        End If

        'If timerBlockage.Enabled = True Then

        '    If rnBlockage = 1 And counterBlockage > 30 Then

        '        blockageResponse = "Yes"

        '    ElseIf rnBlockage = 2 Or 3 Then

        '        blockageResponse = "Yes"

        '    End If

        'End If

        Return blockageResponse

    End Function

    Private Function increasing()

        Dim increasingParameters As String = " "

        Dim incParametersList As New List(Of String)
```

```
' no case for timerReset.enabled because during reset there is stop trend
If timerSARnew.Enabled = True Then
    If rnSAR = 1 And counterSARnew > 25 Then
        incParametersList.Add("Seed Application Rate")
    ElseIf rnSAR = 2 Or rnSAR = 3 Then
        incParametersList.Add("Seed Application Rate")
    End If
End If
If timerFAR.Enabled = True Then
    If rnFAR = 1 And counterFAR > 15 Then
        incParametersList.Add("Fertilizer Application Rate")
    ElseIf rnFAR = 2 And counterFAR > 20 Then
        incParametersList.Add("Fertilizer Application Rate")
    ElseIf rnFAR = 3 Then
        incParametersList.Add("Fertilizer Application Rate")
    End If
End If
If timerSpeedNew.Enabled = True Then
    If rnSpeed = 1 And counterSpeedNew > 18 Then
        incParametersList.Add("Speed")
    ElseIf rnSpeed = 2 And counterSpeedNew > 25 Then
        incParametersList.Add("Speed")
    ElseIf rnSpeed = 3 Then
        incParametersList.Add("Speed")
    End If
End If
If timerTp.Enabled = True Then
    If rnToolPressure = 1 And counterTp > 20 Then
        incParametersList.Add("Tool Pressure")
    ElseIf rnToolPressure = 2 Then
        incParametersList.Add("Tool Pressure")
```

```
            End If
        End If
        If timerToolDepth.Enabled = True Then
            If rnToolDepth = 1 Then
                incParametersList.Add("Tool Depth")
            ElseIf rnToolDepth = 2 And counterToolDepth > 8 Then
                incParametersList.Add("Tool Depth")
            ElseIf rnToolDepth = 3 Then
                incParametersList.Add("Tool Depth")
            End If
        End If
        If timerFanRPM.Enabled = True Then
            If rnFanRpm = 1 And counterFanRPM > 29 Then
                incParametersList.Add("Fan RPM")
            ElseIf rnFanRpm = 2 Then
                incParametersList.Add("Fan RPM")
            ElseIf rnFanRpm = 3 And counterFanRPM > 12 Then
                incParametersList.Add("Fan RPM")
            End If
        End If
        increasingParameters = String.Join(", ", incParametersList)
        Return increasingParameters
    End Function
    Private Function decreasing()
        Dim decreasingParameters As String = " "
        Dim decParametersList As New List(Of String)
        ' no case for timerReset.enabled because during reset there is stop trend
        ' rn = 3 tool pressure decreases, no  inc.
        If timerTp.Enabled = True Then
            If rnToolPressure = 3 And counterTp > 10 And counterTp <= 70 Then  ' corrrected
/changed
                decParametersList.Add("Tool Pressure")
```

```
        End If
    End If
    If timerTanksNew.Enabled = True Then
        If rnTanks = 1 And counterTanksNew <= 65 Then
            decParametersList.Add("Tanks Levels")
        ElseIf rnTanks = 2 And counterTanksNew <= 35 Then
            decParametersList.Add("Tanks Levels")
        ElseIf rnTanks = 3 And counterTanksNew <= 22 Then
            decParametersList.Add("Tanks Levels")
        End If
    End If
    decreasingParameters = String.Join(", ", decParametersList)
    Return decreasingParameters
End Function
Private Function constant()
    Dim constantParameters As String = " "
    Dim contParametersList As New List(Of String)
    If timerSARnew.Enabled = True Then
        If rnSAR = 1 And counterSARnew <= 25 Then
            contParametersList.Add("Seed Application Rate")
        End If
    End If
    'newChanges
    If timerSARnew.Enabled = False And timerIdleSar.Enabled = True Then
        contParametersList.Add("Seed Application Rate")
    End If
    ''''''''''
    If timerFAR.Enabled = True Then
        If rnFAR = 1 And counterFAR <= 15 Then
            contParametersList.Add("Fertilizer Application Rate")
        ElseIf rnFAR = 2 And counterFAR <= 20 Then
```

```
            contPararametersList.Add("Fertilizer Application Rate")
        End If
    End If
    'newChanges
    If timerFAR.Enabled = False And timerIdleFar.Enabled = True Then
        contParametersList.Add("Fertilizer Application Rate")
    End If
    ''''''''''''
    'If timerSpeedReset.Enabled = True And counterSpeedNew <= 10 Then
    '    contParametersList.Add("Speed")
    'End If
    If timerSpeedNew.Enabled = True Then
        If rnSpeed = 1 And counterSpeedNew <= 18 Then
            contParametersList.Add("Speed")
        ElseIf rnSpeed = 2 And counterSpeedNew <= 25 Then
            contParametersList.Add("Speed")
            'ElseIf rnSpeed = 3 And counterSpeedNew <= 27 Then    ' corrected @3 speed is
always increasing,
            '    contParametersList.Add("Speed")
        End If
    End If
    ' newChanges
    If timerSpeedNew.Enabled = False And timerIdleSpeed.Enabled = True Then
        contParametersList.Add("Speed")
    End If
    '''''''''''
    ' newchanges
    If timerTp.Enabled = False And timerIdleTp.Enabled = True Then
        contParametersList.Add("Tool Pressure")
    End If
    ' rn = 3 tool pressure decreases, no  inc.
    If timerTp.Enabled = True Then
```

```
            If rnToolPressure = 1 And counterTp <= 20 Then

                contParametersList.Add("Tool Pressure")

            ElseIf rnToolPressure = 3 And (counterTp <= 10 Or counterTp > 70) Then ' corrected
/changed

                contParametersList.Add("Tool Pressure")

            End If

        End If

        '''''''''''

        'If timerToolDreset.Enabled = True And counterToolDepth <= 30 Then   ' Corrected added
this section

        '    contParametersList.Add("Tool Depth")

        'End If

        If timerToolDepth.Enabled = True Then

            If rnToolDepth = 2 And counterToolDepth <= 8 Then

                contParametersList.Add("Tool Depth")

            End If

        End If

        ' new changes

        If timerToolDepth.Enabled = False And timerIdleToolDepth.Enabled = True Then

            contParametersList.Add("Tool Depth")

        End If

        '''''''''

        If timerFanRPM.Enabled = True Then

            If rnFanRpm = 1 And counterFanRPM <= 29 Then

                contParametersList.Add("Fan RPM")

            ElseIf rnFanRpm = 3 And counterFanRPM <= 12 Then

                contParametersList.Add("Fan RPM")

            End If

        End If

        ' new changes

        If timerFanRPM.Enabled = False And timerIdleFan.Enabled = True Then

            contParametersList.Add("Fan RPM")
```

```
    End If
    '''''
    'If timerTanksReset.Enabled = True And counterTanksNew <= 25 Then
    '    contParametersList.Add("Tanks Levels")
    'End If
    If timerTanksNew.Enabled = True Then
        If rnTanks = 1 And counterTanksNew > 65 Then
            contParametersList.Add("Tanks Levels")
        ElseIf rnTanks = 2 And counterTanksNew > 35 Then
            contParametersList.Add("Tanks Levels")
        ElseIf rnTanks = 3 And counterTanksNew > 22 Then ' corrected rnTanks made 3 from 2
            contParametersList.Add("Tanks Levels")
        End If
    End If
    'new changes
    If timerTanksNew.Enabled = False And timerIdleTanks.Enabled = True Then
        contParametersList.Add("Tanks Levels")
    End If
    constantParameters = String.Join(", ", contParametersList)
    Return constantParameters
End Function
Private Function acceptable()
    Dim acceptableParameters As String = " "
    Dim accParametersList As New List(Of String)
    ' no case for timerReset.enabled because during reset there is stop trend
    Dim seedReading As Integer = CInt(lbSARreading.Text)
    Dim fertilizerReading As Integer = CInt(lbFARnewReading.Text)
    Dim speedReading As Double = CDbl(lbSpeedNewReading.Text)
    Dim TpReading As Integer = CInt(lbTPreading.Text)
    Dim ToolDepthReading As Double = CDbl(lbToolDReading.Text)
    Dim fanReading As Integer = CInt(lbFanRPMreading.Text)
```

```
    Dim TanksLevels As Integer = CInt(lbTanksNewReading.Text.Substring(0, 2))
    Dim blockage As String = blockageRespose()
    If 30 <= seedReading And seedReading <= 50 Then
        accParametersList.Add("Seed Application Rate")
    End If
    If 60 <= fertilizerReading And fertilizerReading <= 100 Then
        accParametersList.Add("Fertilizer Application Rate")
    End If
    If 6 <= speedReading And speedReading <= 10 Then
        accParametersList.Add("Speed")
    End If
    If 50 <= TpReading And TpReading <= 70 Then
        accParametersList.Add("Tool Pressure")
    End If
    If 6 <= ToolDepthReading And ToolDepthReading <= 10 Then
        accParametersList.Add("Tool Depth")
    End If
    If 500 <= fanReading And fanReading <= 700 Then
        accParametersList.Add("Fan RPM")
    End If
    If 10 <= TanksLevels And TanksLevels <= 100 Then
        accParametersList.Add("Tanks Levels")
    End If
    If blockage = "No" Then
        accParametersList.Add("Blockage")
    End If
    acceptableParameters = String.Join(", ", accParametersList)
    Return acceptableParameters
End Function
Private Function notAcceptable()
    Dim notAccParameters As String = " "
```

```vb
Dim notAccParametersList As New List(Of String)
' no case for timerReset.enabled because during reset there is stop trend
Dim seedReading As Integer = CInt(lbSARreading.Text)
Dim fertilizerReading As Integer = CInt(lbFARnewReading.Text)
Dim speedReading As Double = CDbl(lbSpeedNewReading.Text)
Dim TpReading As Integer = CInt(lbTPreading.Text)
Dim ToolDepthReading As Double = CDbl(lbToolDReading.Text)
Dim fanReading As Integer = CInt(lbFanRPMreading.Text)
Dim TanksLevels As Integer = CInt(lbTanksNewReading.Text.Substring(0, 2))
Dim blockage As String = blockageRespose()
If 50 < seedReading Or seedReading < 30 Then
    notAccParametersList.Add("Seed Application Rate")
End If
If 100 < fertilizerReading Or fertilizerReading < 60 Then
    notAccParametersList.Add("Fertilizer Application Rate")
End If
If 10 < speedReading Or speedReading < 6 Then
    notAccParametersList.Add("Speed")
End If
If 70 < TpReading Or TpReading < 50 Then
    notAccParametersList.Add("Tool Pressure")
End If
If 10 < ToolDepthReading Or ToolDepthReading < 6 Then
    notAccParametersList.Add("Tool Depth")
End If
If 700 < fanReading Or fanReading < 500 Then
    notAccParametersList.Add("Fan RPM")
End If
If 100 < TanksLevels Or TanksLevels < 10 Then
    notAccParametersList.Add("Tanks Levels")
End If
```

```
        If blockage = "Yes" Then
            notAccParametersList.Add("Blockage")
        End If
        notAccParameters = String.Join(", ", notAccParametersList)
        Return notAccParameters
    End Function
    Private Function acceptableAfter1min()
        Dim acceptableParameters As String = " "
        Dim accParametersList As New List(Of String)
        ' no case for timerReset.enabled because during reset there is stop trend
        Dim seedReading As Integer = CInt(lbSARreading.Text)
        Dim fertilizerReading As Integer = CInt(lbFARnewReading.Text)
        Dim speedReading As Double = CDbl(lbSpeedNewReading.Text)
        Dim TpReading As Integer = CInt(lbTPreading.Text)
        Dim ToolDepthReading As Double = CDbl(lbToolDReading.Text)
        Dim fanReading As Integer = CInt(lbFanRPMreading.Text)
        Dim TanksLevels As Integer = CInt(lbTanksNewReading.Text.Substring(0, 2))
        Dim seedReadingAfter1min As Integer
        Dim fertilizerReadingAfter1min As Integer
        Dim speedReadingAfter1min As Double
        Dim TpReadngAfter1min As Integer
        Dim ToolDepthReadingAfter1min As Double
        Dim fanReadingAfter1min As Integer
        Dim TanksLevelsAfter1min As Integer
        Dim blockageAfter1min As String = blockageRespose()  ' corrected/ added
        Dim needleInputafter1min As Double = speedReading * 12 + 180 + 60 * 1.7
        Dim tdCounterAfter1Min As Double = ToolDepthReading * 168 / 16 + 60
        Dim fanNeedleInputAfter1Min As Double = fanReading * 0.18 + 180 + 60 * 1.7
        Dim increasingParamtersList As String = increasing()
        Dim decreasingParametersList As String = decreasing()
        Dim constantParametersList As String = constant()
```

```
If increasingParamtersList.Contains("Seed Application Rate") = True Then
    seedReadingAfter1min = seedReading + 60
ElseIf constantParametersList.Contains("Seed Application Rate") = True Then
    seedReadingAfter1min = seedReading
End If
If 30 <= seedReadingAfter1min And seedReadingAfter1min <= 50 Then
    accParametersList.Add("Seed Application Rate")
End If
If increasingParamtersList.Contains("Fertilizer Application Rate") = True Then
    fertilizerReadingAfter1min = fertilizerReading + 120
ElseIf constantParametersList.Contains("Fertilizer Application Rate") = True Then
    fertilizerReadingAfter1min = fertilizerReading
End If
If 60 <= fertilizerReadingAfter1min And fertilizerReadingAfter1min <= 100 Then
    accParametersList.Add("Fertilizer Application Rate")
End If
If increasingParamtersList.Contains("Speed") = True Then
    speedReadingAfter1min = FormatNumber((needleInputafter1min - 180) * 15 / 180, 1)
ElseIf constantParametersList.Contains("Speed") = True Then
    speedReadingAfter1min = speedReading
End If
If 6 <= speedReadingAfter1min And speedReadingAfter1min <= 10 Then
    accParametersList.Add("Speed")
End If
If increasingParamtersList.Contains("Tool Pressure") = True Then
    TpReadngAfter1min = TpReading + 60
ElseIf decreasingParametersList.Contains("Tool Pressure") = True Then
    TpReadngAfter1min = TpReading - 60
ElseIf constantParametersList.Contains("Tool Pressure") = True Then
    TpReadngAfter1min = TpReading
```

```
        End If
        If 50 <= TpReadngAfter1min And TpReadngAfter1min <= 70 Then
            accParametersList.Add("Tool Pressure")
        End If
        If increasingParamtersList.Contains("Tool Depth") = True Then
            ToolDepthReadingAfter1min = FormatNumber(tdCounterAfter1Min * 16 / 168, 1)
        ElseIf constantParametersList.Contains("Tool Depth") = True Then
            ToolDepthReadingAfter1min = ToolDepthReading
        End If
        If 6 <= ToolDepthReadingAfter1min And ToolDepthReadingAfter1min <= 10 Then
            accParametersList.Add("Tool Depth")
        End If
        If increasingParamtersList.Contains("Fan RPM") = True Then
            fanReadingAfter1min = FormatNumber((fanNeedleInputAfter1Min - 180) * 1000 / 180,
1)
        ElseIf constantParametersList.Contains("Fan RPM") = True Then
            fanReadingAfter1min = fanReading
        End If
        If 500 <= fanReadingAfter1min And fanReadingAfter1min <= 700 Then
            accParametersList.Add("Fan RPM")
        End If


        If decreasingParametersList.Contains("Tanks Levels") = True Then
            TanksLevelsAfter1min = TanksLevels - 37.5
        ElseIf constantParametersList.Contains("Tanks Levels") = True Then
            TanksLevelsAfter1min = TanksLevels
        End If
        If 10 <= TanksLevelsAfter1min And TanksLevelsAfter1min <= 100 Then
            accParametersList.Add("Tanks Levels")
        End If
        If blockageAfter1min = "No" Then       ' corrected /added
            accParametersList.Add("Blockage")
```

```
        End If

        acceptableParameters = String.Join(", ", accParametersList)

        Return acceptableParameters

    End Function

    Private Function notAcceptableAfter1min()

        Dim notAcceptableParameters As String = " "

        Dim notAccParametersList As New List(Of String)

        ' no case for timerReset.enabled because during reset there is stop trend

        Dim seedReading As Integer = CInt(lbSARreading.Text)

        Dim fertilizerReading As Integer = CInt(lbFARnewReading.Text)

        Dim speedReading As Double = CDbl(lbSpeedNewReading.Text)

        Dim TpReading As Integer = CInt(lbTPreading.Text)

        Dim ToolDepthReading As Double = CDbl(lbToolDReading.Text)

        Dim fanReading As Integer = CInt(lbFanRPMreading.Text)

        Dim TanksLevels As Integer = CInt(lbTanksNewReading.Text.Substring(0, 2))

        Dim seedReadingAfter1min As Integer

        Dim fertilizerReadingAfter1min As Integer

        Dim speedReadingAfter1min As Double

        Dim TpReadngAfter1min As Integer

        Dim ToolDepthReadingAfter1min As Double

        Dim fanReadingAfter1min As Integer

        Dim TanksLevelsAfter1min As Integer

        Dim blockageAfter1min As String = blockageRespose()  ' corrected/ added

        Dim needleInputafter1min As Double = speedReading * 12 + 180 + 60 * 1.7

        Dim tdCounterAfter1Min As Double = ToolDepthReading * 168 / 16 + 60

        Dim fanNeedleInputAfter1Min As Double = fanReading * 0.18 + 180 + 60 * 1.7

        Dim increasingParamtersList As String = increasing()

        Dim decreasingParametersList As String = decreasing()

        Dim constantParametersList As String = constant()


        If increasingParamtersList.Contains("Seed Application Rate") = True Then
```

```
    seedReadingAfter1min = seedReading + 60
ElseIf constantParametersList.Contains("Seed Application Rate") = True Then
    seedReadingAfter1min = seedReading
End If
If 50 < seedReadingAfter1min Or seedReadingAfter1min < 30 Then
    notAccParametersList.Add("Seed Application Rate")
End If
If increasingParamtersList.Contains("Fertilizer Application Rate") = True Then
    fertilizerReadingAfter1min = fertilizerReading + 120
ElseIf constantParametersList.Contains("Fertilizer Application Rate") = True Then
    fertilizerReadingAfter1min = fertilizerReading
End If
If 100 < fertilizerReadingAfter1min Or fertilizerReadingAfter1min < 60 Then
    notAccParametersList.Add("Fertilizer Application Rate")
End If
If increasingParamtersList.Contains("Speed") = True Then
    speedReadingAfter1min = FormatNumber((needleInputafter1min - 180) * 15 / 180, 1)
ElseIf constantParametersList.Contains("Speed") = True Then
    speedReadingAfter1min = speedReading
End If
If 10 < speedReadingAfter1min Or speedReadingAfter1min < 6 Then
    notAccParametersList.Add("Speed")
End If
If increasingParamtersList.Contains("Tool Pressure") = True Then
    TpReadngAfter1min = TpReading + 60
ElseIf decreasingParametersList.Contains("Tool Pressure") = True Then
    TpReadngAfter1min = TpReading - 60
ElseIf constantParametersList.Contains("Tool Pressure") = True Then
    TpReadngAfter1min = TpReading
End If
If 70 < TpReadngAfter1min Or TpReadngAfter1min < 50 Then
```

```
        notAccParametersList.Add("Tool Pressure")
    End If
    If increasingParamtersList.Contains("Tool Depth") = True Then
        ToolDepthReadingAfter1min = FormatNumber(tdCounterAfter1Min * 16 / 168, 1)
        'ElseIf decreasingParametersList.Contains("Tool Pressure") = True Then
        '    TpReadngAfter1min = TpReading - 60
    ElseIf constantParametersList.Contains("Tool Depth") = True Then
        ToolDepthReadingAfter1min = ToolDepthReading
    End If
    If 10 < ToolDepthReadingAfter1min Or ToolDepthReadingAfter1min < 6 Then
        notAccParametersList.Add("Tool Depth")
    End If
    If increasingParamtersList.Contains("Fan RPM") = True Then
        fanReadingAfter1min = FormatNumber((fanNeedleInputAfter1Min - 180) * 1000 / 180,
1)
    ElseIf constantParametersList.Contains("Fan RPM") = True Then
        fanReadingAfter1min = fanReading
    End If
    If 700 < fanReadingAfter1min Or fanReadingAfter1min < 500 Then
        notAccParametersList.Add("Fan RPM")
    End If
    'If increasingParamtersList.Contains("Tanks Levels") = True Then
    '    ToolDepthReadingAfter1min = FormatNumber(tdCounterAfter1Min * 16 / 168, 1)
    If decreasingParametersList.Contains("Tanks Levels") = True Then
        TanksLevelsAfter1min = TanksLevels - 37.5
    ElseIf constantParametersList.Contains("Tanks Levels") = True Then
        TanksLevelsAfter1min = TanksLevels
    End If
    If 100 < TanksLevelsAfter1min Or TanksLevelsAfter1min < 10 Then
        notAccParametersList.Add("Tanks Levels")
    End If
    If blockageAfter1min = "Yes" Then      ' corrected / added
```

```
            notAccParametersList.Add("Blockage")
        End If
        notAcceptableParameters = String.Join(", ", notAccParametersList)
        Return notAcceptableParameters
    End Function
    Private Async Function printResponses() As Task
        Dim mydocpath As String =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments)
        ' Create a string builder and write the user input from the textbox to it.
        Dim sb As StringBuilder = New StringBuilder()
        'sb.AppendLine("New User Input")
        'sb.AppendLine("= = = = = =")
        ' sb.AppendLine()
        sb.AppendLine("End of New-Ver.1 : " & Now())
        sb.AppendLine()
        sb.AppendLine("Correct Responses New-Ver.1 : ")
        sb.AppendLine()
        sb.AppendLine("SAR:" & lbSARreading.Text)
        sb.AppendLine("FAR:" & lbFARnewReading.Text)
        sb.AppendLine("Speed:" & lbSpeedNewReading.Text)
        sb.AppendLine("Tool Pressure:" & lbTPreading.Text)
        sb.AppendLine("Fan RPM:" & lbFanRPMreading.Text)
        sb.AppendLine("Tool Depth:" & lbToolDReading.Text)
        sb.AppendLine("Tanks:" & lbTanksNewReading.Text)
        sb.AppendLine("Blockage:" & blockageRespose())
        sb.AppendLine("Increasing Parameters:" & increasing())
        sb.AppendLine("Decreasing Parameters:" & decreasing())
        sb.AppendLine("Constant Parameters:" & constant())
        sb.AppendLine("Acceptable Parameters:" & acceptable())
        sb.AppendLine("Not Acceptable Parameters:" & notAcceptable())
        sb.AppendLine("Acceptable after 1 min:" & acceptableAfter1min())
        sb.AppendLine("Not Acceptable after 1 min:" & notAcceptableAfter1min())
```

```vb
        sb.AppendLine()

        ' sb.AppendLine()

        Using outfile As StreamWriter = New StreamWriter(mydocpath +
"\ConsolidatedDisplayFile.txt", True)

            Await outfile.WriteAsync(sb.ToString())

        End Using

        SAEvaluationForm.Visible = True

        Me.Close()

        '  Me.Visible = False

        Me.Dispose()

    End Function

    Dim counterFormClose As Integer = 0

    Private Async Sub timerFormClose_Tick(sender As Object, e As EventArgs) Handles
timerFormClose.Tick

        counterFormClose = counterFormClose + 1

        If counterFormClose = formClosetime Then

            'If counterFormClose = 65 Then

            Await printResponses()

            ' Cursor.Show()

        End If

    End Sub

#End Region

    Private Sub timerIdleSar_Tick(sender As Object, e As EventArgs) Handles timerIdleSar.Tick

        counterIdleSar = counterIdleSar + 1

        If counterIdleSar < 3 Then

            captionSeedAppRate()

            drawSARScale()

            idleSar()

        End If

    End Sub

    Private Sub timerIdleTanks_Tick(sender As Object, e As EventArgs) Handles
timerIdleTanks.Tick

        counterIdleTanks = counterIdleTanks + 1
```

```
        If counterIdleTanks < 3 Then

            idleTanks()

        End If

    End Sub

    Private Sub timerIdleFar_Tick(sender As Object, e As EventArgs) Handles timerIdleFar.Tick

        counterIdleFar = counterIdleFar + 1

        If counterIdleFar < 3 Then

            captionFAR()

            drawFARscale()

            idleFar()

        End If

    End Sub

    Private Sub timerIdleSpeed_Tick(sender As Object, e As EventArgs) Handles
timerIdleSpeed.Tick

        counterIdleSpeed = counterIdleSpeed + 1

        If counterIdleSpeed < 3 Then

            captionSpeed()

            drawDialNumberingSpeedNew()

            idleSpeed()

        End If

    End Sub

    Private Sub timerIdleFan_Tick(sender As Object, e As EventArgs) Handles timerIdleFan.Tick

        counterIdleFan = counterIdleFan + 1

        If counterIdleFan < 3 Then

            idleFan()

        End If

    End Sub

    Private Sub timerIdleTp_Tick(sender As Object, e As EventArgs) Handles timerIdleTp.Tick

        counterIdleToolPressure = counterIdleToolPressure + 1

        If counterIdleToolPressure < 3 Then

            idleToolPressure()

        End If
```

```
    End Sub

    Private Sub timerIdleToolDepth_Tick(sender As Object, e As EventArgs) Handles
timerIdleToolDepth.Tick

        counterIdleToolDepth = counterIdleToolDepth + 1

        If counterIdleToolDepth < 3 Then

            idleToolDepth()

        End If

    End Sub

    Private Sub timerIdleGlobal_Tick(sender As Object, e As EventArgs) Handles
timerIdleGlobal.Tick

        counterIdleGlobal = counterIdleGlobal + 1

        If counterIdleGlobal = 10 Then

            start3randomParametersAfterFormLoad()

        End If

    End Sub

    Private Sub NewConsolidatedDisplayVerOne_MouseEnter(sender As Object, e As EventArgs)
Handles Me.MouseEnter

    End Sub

    Private Sub NewConsolidatedDisplayVerOne_MouseLeave(sender As Object, e As EventArgs)
Handles Me.MouseLeave

    End Sub



End Class
```