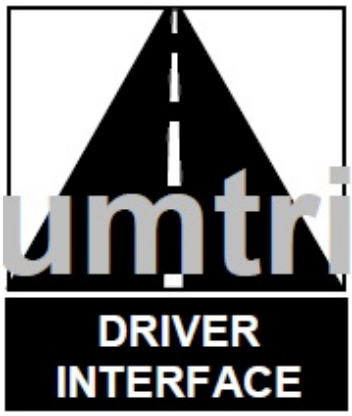
UMTRI Design Guidelines and Supporting Information for Automated Driving Systems

version 2.0

Paul Green



University of Michigan Transportation Research Institute

revised January 16, 2019

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
UMTRI-2018-**		

4. Title and Subtitle	5. Report Date
UMTRI HMI Design Guidelines and Supporting Information for Automated Driving Systems, version 2.0	January 2019
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Paul Green	U. of Michigan. project grant

9. Performing Organization Name and Address	10. Work Unit no. (TRAIS)
University of Michigan Transportation Research Institute 2901 Baxter Road Ann Arbor, Michigan 48109-2150 USA	11. Contract or Grant No. U. of Michigan

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Changan US Research and Development Center, Inc. 47799 Halyard Drive, Plymouth, Michigan 48170 USA	September-December, 2018
	14. Sponsoring Agency Code

15. Supplementary Notes

Attention: Claudia Escobar, Intelligent and Connected Vehicle Group

16. Abstract

This report lists and explains design principles, guidelines, and related information pertaining to the design of the human-machine interface (HMI) for vehicles providing SAE Levels 1-3 automated driving systems.

The report begins with automation-related definitions (primarily from SAE J3016, e.g., operational design domain) and concepts (e.g., takeover taxonomies including a new proposal), level of automation schemes (e.g., SAE, Sheridan, Parasuraman), and a list of relevant SAE standards.

The next sections identify the most serious problems (1-the driver does not know what the vehicle knows or how confident it is, 2-the driver does not know what mode they are in and how that depends on the operational design domain). Guidelines to solve these problems are provided, guidelines that do not appear elsewhere. In addition, several methods (that utilize matrices and production rules) are described to identify inconsistencies and complexity in how setting tasks and in mode changes are performed. Also described are measures and statistics from software science (e.g., McCabe's Cyclomatic Complexity, Halstead's difficulty measure,) that can predict software complexity as well as those from cognitive complexity research (e.g., number of rules needed to perform a task). These measures and statistics could be used to assess the merits of alternative system architectures and HMI designs.

The last portion of the report lists traditional HMI guidelines (for text size, colors, abbreviations, icons, navigation screen, menus, etc.) that are relevant to automated vehicle HMIs. To help identify problems that need to be addressed, the find, perceive, decide, execute (FPDE) stage-based method is described as well as associated design principles.

17. Key Words	18. Distribution Statement
human factors, ergonomics, safety, usability, automation, automated driving, situation awareness, task analysis, ADAS	

19. Security Classify. (of this report)	20. Security Classify. (of this page)	21. No. of pages	22. Price
(None)	(None)	159	

Form DOT F 1700 7 (8-72) **Reproduction of completed page authorized**

TABLE OF CONTENTS

INTRODUCTION 1
- **Purpose:** 1

Q1: WHAT ARE SOME KEY TERMS AND CONCEPTS RELATED TO AUTOMATED DRIVING SYSTEMS AND THEIR FUNCTIONALITY? 3

Q2: HOW IS THE LEVEL OF/DEGREE OF AUTOMATION SPECIFIED? 9
- **SAE 6-Level Scheme** 9
- **Sheridan's 10 Levels** 10
- **Parasuraman Decision-Making Perspective** 12
- **Endsley-Kaber Matrix** 13
- **Wickens' Refinements** 14
- **Relevant Guidelines** 15

Q3: WHAT ARE THE MOST IMPORTANT EXISTING STANDARDS AND GUIDELINES FOR AUTOMATION? 17

Q4: WHAT ARE THE MAJOR PROBLEMS DRIVERS ENCOUNTER WHEN USING AUTOMATED DRIVING SYSTEMS? 21
- **Dikmen and Burns (2016)** 21
- **Brown and Laurier (2017)** 21
- **Endsley (2017)** 22
- **Disengagements** 24

Q5: WHAT ARE SOME GUIDELINES THAT CAN HELP DESCRIBE AND SIMPLY THE INFORMATION ARCHITECTURE? 31
- **High-Level Guidelines** 31
- **Examples of Applications of the High-Level Guidelines** 33

Q6: WHAT MAKES AN INTERFACE COMPLEX (AND NOT EASY TO USE)? 35
- **How Can System Operation Be Represented (in General)?** 35
- **How Has Automation Architecture Been Represented?** 37
- **How Can the Complexity of System Logic Be Determined from Those Representations?** 46
- **What Make Makes System Logic Complex for People? - Cognitive Complexity Theory** 50
- **How Can the Complexity of the Representation Be Determined?** 53
- **Synthesis of This Section** 53

Q7: WHAT ARE SOME GUIDELINES THAT CAN REDUCE INTERFACE COMPLEXITY? 55

Q8: WHAT ARE TASKS THAT PEOPLE DO OR DO COULD DO WITH AUTOMATION INTERFACES? 57

Q9: WHAT ARE SOME TRADITIONAL HMI GUIDELINES THAT CONCERN INFORMATION PRESENTATION? 63
- **General HMI Guidelines** 63
- **Automotive Specific Guidelines** 64
- **Automotive HMI Checklists** 66

Q10: WHAT ARE SOME STAGE-BASED TASK DESIGN PRINCIPLES? 69
Find 69
Perceive 74
Decide 75
Execute 78
Q11: HOW SHOULD SPECIFIC INTERFACE ELEMENTS BE DESIGNED TO SUPPORT DRIVER TASKS? 83
Displays 83
Controls 92
HMI Information Systems 98
NEXT STEPS 107
REFERENCES 109
APPENDIX A – GUIDELINES IN THIS REPORT 127
APPENDIX B – LEVEL OF AUTOMATION TECHNOLOGY 133
APPENDIX C – BILLINGS GUIDELINES FOR HUMAN-CENTERED AVIATON AUTOMATION 137
APPENDIX D – ENDSLEY SITUATION AWARENESS DESIGN PRINCIPLES 139
APPENDIX E – CAMP GUIDELINES 141
APPENDIX F – EUROPEAN STANDARD OF PRACTICE FOR HMI – ABRIDGED 145
APPENDIX G – NAUJOKS (BMW) EXTENDED CHECKLIST 151

INTRODUCTION

Purpose:

The purpose of this guidance is to help Changan and others develop safe, easy to use, and useful human-machine interfaces (HMIs) for SAE Level 1-3 passenger vehicles, light trucks, and SUVs.

To clarify the purpose, ISO 9241:210:2010, definition 2.13 (see also ISO 9241-11:2018, definition 3.1.1) defines usability as the, "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." Thus, usability, as defined by ISO, has 3 aspects to it, (1) effectiveness ("accuracy and completeness with which users achieve specified goals"), ISO 9241-11, definition 3.1.12), (2) efficiency **"**resources used in relation to the results achieved," ISO 9241-11, definition 3.1.13), and (3) satisfaction **("**extent to which the user's physical, cognitive and emotional responses that result from the use of a system, product or service meet the user's needs and expectations,"

ISO 9241-11, definition 3.1.14). ISO 5840-3:2013, definition 3.44 defines safe as "freedom from unacceptable risk." ISO does not define useful, but the dictionary of record, the 3rd International Dictionary does, "capable of being put to use *especially***:** serviceable for an end or purpose e.g., *useful* tools."

To achieve those purposes, this report addresses 11 questions:

Q1: What are some key terms and concepts related to automated driving systems and their functionality?

Q2: How is the level of/degree of automation specified?

Q3: What are the most important existing standards and guidelines for automation?

Q4: What are the major problems drivers encounter when using automated driving systems?

Q5: What are some guidelines that can help describe and simplify the information architecture?

Q6: What makes an interface complicated (and not easy to use)?

Q7: What are some guidelines that can reduce interface complexity?

Q8: What are the tasks that people do or could do with automation interfaces?

Q9: What are some traditional HMI guidelines that support information representation and therefore, task completion?

Q10: What are some stage-based task design principles?

Q11: How should specific interface elements be designed to support driver tasks?

The full set of guidelines, numbered to correspond to these questions, and distributed with the associated question, also appear in Appendix A.

Q1: WHAT ARE SOME KEY TERMS AND CONCEPTS RELATED TO AUTOMATED DRIVING SYSTEMS AND THEIR FUNCTIONALITY?

There are a number of terms critical to understanding the literature on automated driving systems, most of which appear in SAE Recommended Practice J3016 (Society of Automotive Engineers, 2018). Those terms and their definitions are repeated here for convenience. The words in square brackets are not in the original definition but have been added for clarity.

J3016 recommends the use of the term *automated driving system*. The use of other following terms is *deprecated*, which means they are not approved/discouraged from use (Table 1-1).

Table 1-1. Terms Related to Automated Driving Systems Not Recommended by SAE J3016

Term	Misleading Impression Conveyed
autonomous	decision making is independent and self-sufficient and ignores communication and cooperation with other entities
self-driving	there is no human driver present, which is not always true
driverless and unmanned	there is no human driver present, which is not always true
robotic	there is no human involved
automated or autonomous vehicle	emphasis on the vehicle rather than the driving task (a fine distinction) and also may indicate a human driver is not present.

Source: SAE J3016. pages 26 - 27.

Equivalent terms in Mandarin, German, Japanese, Korean, Italian, and the native languages of non-English speaking automakers are to be determined.

However, for convenience in this report the author sometimes uses the terms automated and partially-automated vehicle or simplicity. Partially-automated is specifically intended to refer to SAE Levels 1-3 (defined in the next section). Automated is intended to refer to any SAE Level greater than 0 (not completely manual, defined later).

Widely used in the literature and this report are the abbreviations DDT, ODD, and OEDR, defined in Table 1-2.

Table 1-2. Commonly Abbreviated Terms Related to Automated Driving Systems

Abbrev	Term	Definition	J3016 page
DDT	Dynamic Driving Task	"real time … functions required to operate a vehicle excluding strategic functions" [such as trip planning]	5
ODD	Operational Design Domain	"conditions [environment, roadway, speed, traffic, etc.] under which an automated driving system is designed to function"	12
OEDR	Object and Event Detection and Response	"monitoring the driving environment (detecting, recognizing, and classifying objects [such as road marking, signs, other vehicles, traffic signals]) and executing an appropriate response [to them]"	12

Although not explicitly defined in SAE J3016, J3016 refers to 3 regimes of control identified by Michon (1993) – operational, tactical, and strategic.

Operational refers to actions that are typically below 1 second, such as a steering wheel action to correct a lane departure.
Tactical refers to maneuvers, such as changing lanes, that transpire over several seconds.
Strategic refers to decision that takes many seconds at the least, such as selecting a route.

Identifying the regime of control, and who is engaged in that regime (human, automation, or both), and if the other party is actively engaged in backup are important considerations for automation system design.

Another important aspect of partially automated systems is that transfer of control, traditionally thought of either from the human operator (the driver) to the automation (the vehicle) or from the automation to the human operator. In the literature, the predominant focus has been on the automation-human transfer, with takeover (often unexpectedly) being the label for that transfer (Merat, Jamson, Lai, Daly, and Carsten, 2014; Zeeb, Buchner, and Schrauf, 2015). (See also Lu, Happee, Cabrall, Kyriakidis, and deWinter 2016; Wintersberger, Green, and Riener, 2017; McCall, McGee, Mirnig, Meschtscherjakov, Louveton, Engel, and Tscheligi, 2018). Another term that is sometimes used is handover, which could mean the driver relinquishes control to the automation, but there is no reason why it could not refer to the transfer of control in the opposite direction. When used with the term handback, the usual interpretation is the driver is handing over control to the automation and the automation hands it back to the driver.

Why does this ambiguity exist? The dictionary definition of takeover is "to assume control or possession" (https://www.merriam-webster.com/dictionary/takeover). From a driver-centric perspective, takeover would mean the driver assumes control, but can hand it back to the automation. From an automation-centric perspective, the use of the terms is reversed. For example, in some systems, it is envisioned that if the driver is no longer capable of driving, for example, because they are drunk, then the automation would assume control; it would take over.

Thus, although these terms takeover, handover, handback, others are widely used and there is some consensus as to what they mean, there are opportunities for misunderstanding.

In SAE J3015, only the term *fallback* is used and the term takeover never appears, although there is 1 instance in J3016 where the word pair "take over" appears (section 4, page 18). *("If the driving automation system performs the entire DDT, the user does not do so. However, if a DDT fallback-ready user is expected to take over the DDT when a DDT performance-relevant system failure occurs or when the driving automation system is about to leave its operational design domain (ODD), then that user is expected to be receptive and able to resume DDT performance when alerted to the need to do so."*

Fallback, is defined in section 3.14 (page 7) of J3016 as "the response by the *user* to either perform the *DDT* or achieve a *minimal risk condition* after occurrence of a *DDT performance-relevant system failure(s)* or upon *operational design domain* (*ODD*) exit, or the response by an *ADS* to achieve *minimal risk condition*, given the same circumstances." Thus, fallback is the transfer of control from the automation to a human-being. No specific information is provided about when the sequence begins (when the automation informs the driver of a fall back (now, 5 seconds from now, etc.) or when the driver begins to anticipate when it could occur, and when it ends (e.g., when the driver begins to exert control or when the driver has total control).

Interestingly, J3016 does not include the term fallback request, but uses the term *request to intervene* (definition 3.24, page 15). Note that the term fallback is automation-centered (what the system does) where takeover is human-centered (what the person does).

Those interested in definitions in J3016 should make sure they examine the latest version (2018) when this report was written, as J3016 has evolved over time.

Taxonomies for Transfer of Control

As a starting point, a taxonomy is needed is to identify all of the situations in which transfer of control can occur. What follows is a scheme developed by the author that is more complete that scheme previously described in the literature, but could use additional development. The initial consideration is who are the parties are in the transfer: (1) human to automation (HA), (2) automation to human (AH), (3) human to human (HH, from or to a remote driver), and (4) automation to automation (AA, onboard to off board, or a level change). Currently, the taxonomy does not reflect the specific nature of the responsibility of each party (for example for object and event detection, strategic decisions) or the degree of automation (such as Sheridan's levels, described later). See Figure 1-1. However, given confusion about what takeover, handover, and handback mean, and that SAE J3016 does not cover all of the transfers of interest, this report will use the taxonomy proposed (HA, AH, AA, HH transfer, with traditional terms as provided as well for clarity.

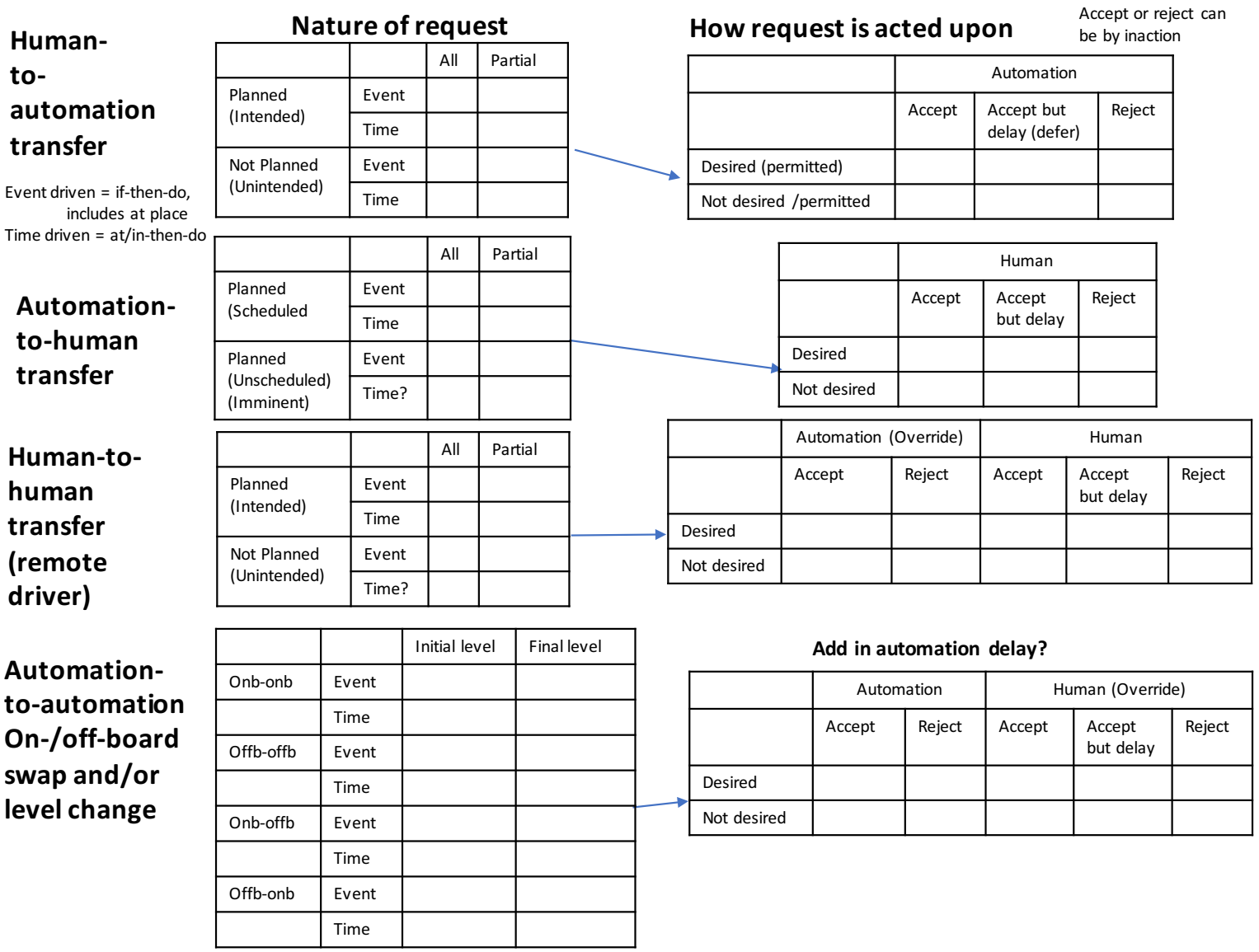


Figure 1-1. Handover Taxonomy

The second characteristic is the nature of the request, if the request is planned (intended) or not planned (unintended). Both types of requests can be triggered by an event (if occurred-then do, if at place, then do) or by time (if at time x, then do). Time driven events can be scheduled for the future. One also could treat the trigger as interrupts, with events thought of as unscheduled interrupts, and place and time interrupts as be scheduled. This idea could lead to further refinement of the taxonomy.

Furthermore, the nature of the request can be for full or partial transfer of control. For example, a system could transfer steering control back to a driver but retain speed control (using ACC). The final aspects of the taxonomy are if that task is desired/not desired or permitted/not permitted, how the request is acted upon, (if a human or automation accepts the request), and if it is accepted, is there a delay in the acceptance. The permission idea is in SAE J3114. One important idea that will be added in a future version of this taxonomy is if the takeover is temporary or not. Temporary take overs are well known to occur for cruise control systems, where the driver wants to go faster than the set speed, and presses the accelerator to do so, overriding the cruise control.

When designing an HMI for an automated driving system, all of the use cases this taxonomy identifies must be examined.

The literature on control transition is best summarized by Mirnig, Gärtner, Laminger, Meschtscherjakov, Trösterer, Tscheligi, McCall and McGee (2017). Their scheme identifies the factors in Table 1-3. To help organize these factors, they have been grouped into 3 broad categories, structure or nature of a request, executing a request, and feedback for the process (some of which is needed for planning).

Table 1-3. Control Transition Factors from Mirnig et al (2017).

Factor Category	Factor	Description and /or Options and Coding
Structure or Nature of Request What is to be done and by who	Direction of transfer	• Handover (H) – human to automation (HA) • Takeover (T) – automation to human (AH)
	Initiator of request	• Driver=user (U) • system (S)
	Driver involvement (in controlling transitions)	• Active (A) – press button, turn steering wheel, push pedals • Semiactive (SA) – confirm or deny transitions that are system initiated • Passive (P) – no input to initiate or confirm (example = ESC application)
	Completeness (all or some functions are transferred)	• Full (F) • Partial (P) • Incremental (I)
Execution of Request	Fallback (if transition is not successful)	• Own lane (O) • slow down and stop outside one's own lane (U) • System (s) =no specification of what system does

How it is to be done	Input (How to initiate or respond to transition requests)	• Physical (P) • Touch (T) • Speech (S) • Gestures (G)
Output Related to Request How the process is monitored	Output	• Available Visual (V) • Haptic (H) • Acoustic (A) • Speech (S)
	Visual Output Position	• Dashboard (D) • Console (C) • Head-Up Display (H) • Windshield (W) or combinations of these
	Haptic Output Position	• Steering wheel (S) • seat (T) • Pedals (P) • Wristband (W) or combinations of these
	Temporal Output Mode	• Once (O) • Several (S) • Increment (I) – Incremental is a special case of S where the output increases or decreases incrementally to signal a transition
	Mode Awareness (inform driver if in manual or autonomous mode)	• Symbol (S) • Color (C) • Text (T) • Audio (A) • Other (O)

The value of these taxonomies is that they help identify the use cases to be explored, the specific tasks, and the devices needed to support those tasks.

Q2: HOW IS THE LEVEL OF/DEGREE OF AUTOMATION SPECIFIED?

SAE 6-Level Scheme

The 6-levels of driving automation defined in SAE Recommended Practice J3016 is the most commonly used scheme now (Table 2-1). Historically, one can find similar schemes from NHTSA and from BASt that are no longer used (Gasser, 2014, Winkle, 2016). As described by Blain (2017), for Level 0, the driver is in full control (acceleration, braking, steering) at all times, though warning systems may be present.

For Level 1, the automation can either control steering or the pedals, but not both. So, adaptive cruise control is Level 1 as is park assist (where the car steers but the driver still operates the pedals).

For Level 2, the automation sometimes controls both steering and the pedals, but the driver must be prepared to take over (an AH transfer). (Note: SAE does not consider AA transfers.) Thus, a vehicle with both adaptive cruise control and lane-keeping assist (or lane centering) would be considered Level 2 if both are operating at the same time. If only 1 is operating, then the vehicle is Level 1. The current version of the Tesla Autopilot and the GM Super Cruise (for expressways only) are examples of Level 2 systems.

For Level 3, the vehicle has greater control authority, including the ability to change lanes (in recent Tesla updates), but most importantly, the ability to deal with a full range of hazards that affect steering and braking/acceleration. However, the driver still needs to be able to resume control. Because of concerns that drivers will be sufficiently in the loop to takeover if needed, some manufacturers have decided, for now, not to offer Level 3 systems.

For Level 4, the vehicle can drive itself, and if it encounters a problem, the vehicle can pull over or park itself. However, the operational domain is constrained, for example to only paved roads or to a certain area. The Navya shuttles in Ann Arbor are Level 4 vehicles (though for now, there is still an operator in the vehicle available to take over.)

For Level 5, the vehicle can drive anywhere, and no human input is ever expected. (See also, Godsmark, 2017.)

Table 2-1. SAE 6 Levels of Automated Driving Systems

Level	Name	Narrative definition	DDT		DDT fallback	ODD
			Sustained lateral and longitudinal vehicle motion control	OEDR		
Driver* performs part or all of the *DDT						
0	**No Driving Automation**	The performance by the *driver* of the entire *DDT*, even when enhanced by *active safety systems*.	*Driver*	*Driver*	*Driver*	n/a
1	**Driver Assistance**	The *sustained* and *ODD*-specific execution by a *driving automation system* of either the *lateral* or the *longitudinal vehicle motion control* subtask of the DDT (but not both simultaneously) with the expectation that the *driver* performs the remainder of the *DDT*.	*Driver* and *System*	*Driver*	*Driver*	Limited
2	**Partial Driving Automation**	The *sustained* and *ODD*-specific execution by a *driving automation system* of both the *lateral* and *longitudinal vehicle motion control* subtasks of the *DDT* with the expectation that the *driver* completes the *OEDR* subtask and *supervises* the *driving automation system*.	***System***	*Driver*	*Driver*	Limited
***ADS* ("*System*") performs the entire *DDT* (while engaged)**			*System*	***System***	*Fallback-ready user (becomes the driver during fallback)*	Limited
3	**Conditional Driving Automation**	The *sustained* and *ODD*-specific performance by an *ADS* of the entire DDT with the expectation that the *DDT fallback-ready user* is *receptive* to *ADS*-issued *requests to intervene*, as well as to *DDT performance-relevant system failures* in other *vehicle* systems, and will respond appropriately.				
4	**High Driving Automation**	The *sustained* and *ODD*-specific performance by an *ADS* of the entire *DDT* and *DDT fallback* without any expectation that a *user* will respond to a *request to intervene*.	*System*	*System*	***System***	Limited
5	**Full Driving Automation**	The *sustained* and unconditional (i.e., not *ODD*-specific) performance by an *ADS* of the entire *DDT* and *DDT fallback* without any expectation that a *user* will respond to a *request to intervene*.	*System*	*System*	*System*	**Unlimited**

Source: SAE J3016, p. 19

Sheridan's 10 Levels

When characterizing automated driving system functions, Sheridan's (or Sheridan and Verplank's) 10 Levels of Automation ("Man-Computer Decision-Making), can be informative. That scheme was developed for application to human and computer control of underwater teleoperators, so some changes in language are needed for automotive applications. Although not commonly cited in automotive research documents that have been widely published, this scheme has been used by internally by system developers from at least 1 automaker when implementing features because the scheme makes useful distinctions about who is in control (what is automated) and to what degree that the SAE levels do not distinguish. For example,

should an automatic lane change system operate without any human interaction (Level 10), tell the driver when lanes were changed if it deems necessary (Level 9), tell the driver only if they ask (Level 8), always tell the driver (Level 7), plan the maneuver and change lanes only if the driver does not object within some time window (Level 6), or suggest a lane change, but only execute it if the driver approves (Level 5). Thus, when an automated driving system is being designed, an important architectural decision is the Sheridan Automation Level for each feature of the automation. Note that difference features of an automated driving system, can have difference Sheridan Levels (e.g., one feature of adaptive cruise control system vs. another feature of that ACC system).

Table 2-2 lists those levels, the text for which has been slightly edited for clarity. The original wording of levels 9 and 10 is confusing, and the author has relied on the interpretation in Save, Feuerberg, and Avia (2012).

Table 2-2. Sheridan's 10 Levels of Automation

Level	Description
1	Human does the whole job up to the point of turning it over to the computer to implement
2	The computer helps by determining the options
3	The computer helps to determine the options and suggests 1, which the human need not follow
4	The computer selects an action and the human may or may not do it.
5	The computer selects an action and implements it if the human approves
6	The computer selects an action, informs the human in plenty of time to stop it.
7	The computer does the whole job and necessarily tells the human what it did
8	The computer does the whole job and necessarily tells the human what it did only if the human explicitly asks.
9	The computer does the whole job if it decides it should be done, and if so, tells the human if it decides he should be told.
10	The computer does the whole job if it decides it should be done, and ignores the human.

Source: Sheridan and Verplank (1979), p. 8-17 to 8-19

That scheme has been repeated using somewhat clearer wording (Table 2-3) in Parasuraman, Sheridan, and Wickens (2000) (Table 2), although there are other variations as well (e.g., Moray, Rodriguez and Clegg, 2000). Thus, these descriptions of the levels of automation could be used as alternatives to the prior Sheridan list.

Table 2-3. Parasuraman's 10 Levels of Automation

Amount of Automation	Level	The computer …
Low	1	offers no assistance: human must take all decisions and actions
	2	offers a complete set of decision/action alternatives,
	3	narrows the selection down to a few
	4	suggests 1 alternative
	5	executes that suggestion if the human approves
	6	allow the human a restricted time to veto before automatic execution
	7	executes automatically, then necessarily informs the human
	8	informs the human only if asked
	9	informs the human only if it, the computer, decides to
High	10	decides everything, acts autonomously, ignoring the human.

Source: Parasuraman, Sheridan, and Wickens (2000), p. 287

Parasuraman Decision-Making Perspective

Parasuraman, Sheridan, and Wickens (2000) make even finer distinctions, partitioning tasks into 4 broad classes of functions: (1) information acquisition, (2) information analysis, (3) decision and action selection, and (4) action implementation, and then for each class of functions, determining the degree of automation provided (Figure 2-1). These functions are parallel to a simple 4-stage model of human information processing (sensory procession, perception/working memory, decision making, response execution). The author does not know of any automotive systems for which this perspective has been applied. However, if it was, most likely it would be applied for at least as a system level (for example, the degree to which action implementation is automated), not on a feature by feature basis.

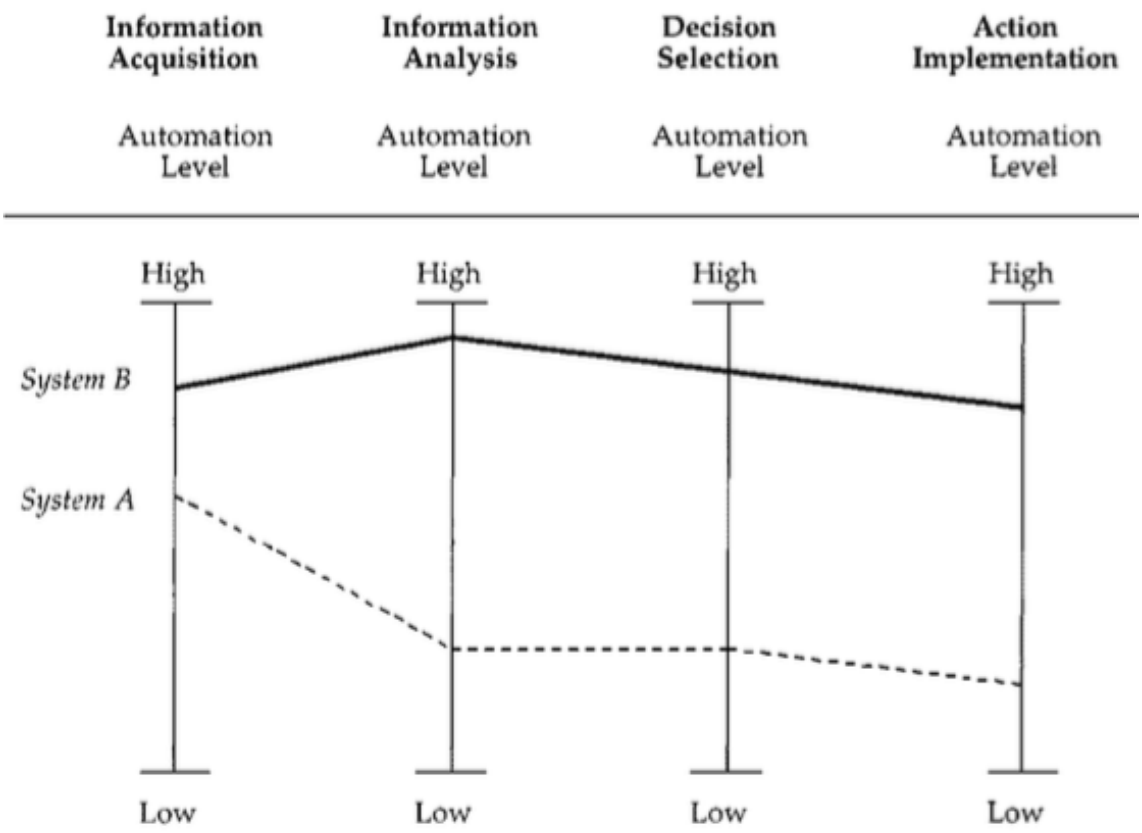


Figure 2-1. Levels of Automation by Function
Source: Parasuraman, Sheridan, and Wickens (2000), p. 288

Thus, what is automated depends on who (the person, the machine/computer) is best at what. This idea harkens back to an idea common in the early days of human factors (Fitts, 1951), the so-called MABA-MABA lists (men are better at-machines are better at).

Endsley-Kaber Matrix

An extension of these ideas is the Endsley-Kaber Matrix, which describes to whom information processing activities are assigned (human or computer) as the level of automation increases, (Endsley and Kaber, 1999). See Table 2-4. In this scheme, they refer to information processing activities as roles, which are similar but not identical to the term function used by Parasuraman, Sheridan, and Wickens (2000).

Table 2-4. Endsley-Kaber Matrix

	Roles			
Level of automation	Monitoring	Generating	Selecting	Implementing
(1) Manual control (MC)	Human	Human	Human	Human
(2) Action support (AS)	Human/Computer	Human	Human	Human/Computer
(3) Batch processing (BP)	Human/Computer	Human	Human	Computer
(4) Shared control (SHC)	Human/Computer	Human/Computer	Human	Human/Computer
(5) Decision support (DS)	Human/Computer	Human/Computer	Human	Computer
(6) Blended decision making (BDM)	Human/Computer	Human/Computer	Human/Computer	Computer
(7) Rigid system (RS)	Human/Computer	Computer	Human	Computer
(8) Automated decision making (ADM)	Human/Computer	Human/Computer	Computer	Computer
(9) Supervisory control (SC)	Human/Computer	Computer	Computer	Computer
(10) Full automation (FA)	Computer	Computer	Computer	Computer

Source: Endsley and Kaber (1999) p. 466

In reviewing the automation classification schemes to date, Save and Feuerberg (2012) note that (1) these schemes tend to view automation as all or none (only human involvement), when sometimes both the human and the computer are involved, (2) there is not enough detail about information analysis, and (3) decision and action selection may be difficult to distinguish from action implementation. This led to the proposal of the 8x4 (with many empty cells) Level of Automation Technology (LOAT). The lengthy table for this scheme appears in Appendix B. This relatively new scheme has yet to be widely cited in the literature.

For unknown reasons, the Endsley-Kaber Matrix has not as much impact on automation research as has Parasuraman, Sheridan, and Wickens (2000), whose impact is less than that of Sheridan's 10 Levels. The LOAT scheme is too new to assess its impact.

One challenge to all that has been presented so far is that level of automation is assumed to be fixed, which in many cases could be true, but automation can be adaptive (Kaber and Endsley, 2004), which complicates system design and analysis.

Wickens' Refinements

Wickens, Mavor, Parasuraman, and McGee (1988) take the idea of information functions 1 step further by identifying 6 attributes of information acquisition (filtering, information distribution, transformations, confidence estimates, integrity checks, and user request enabling). One could also identify the attributes of the other 3 information processing activities as well to help describe the nature of the automation being provided. The author has not found research applying these refinements, but given what has been found so far, it seems likely to exist.

All of this leads the author to propose the first automation guideline.

Relevant Guidelines

Guideline 2-1: For each automation system feature, a first step in design should be selecting a proposed level of automation, either using Sheridan's 10-Levels of Automation or Parasuraman's 10-Levels of Automation.

Note 1: As a design evolves, it could become apparent that there are better alternatives to the initial proposal for the intended level, in which case it should be changed.

Note: There are more detailed alternatives and in the future, they could be recommended over these 2 schemes in the future.

As evidence for the need for this guideline, consider the follow example. Suppose a vehicle will have an automatic lane change feature as was mentioned previously. How much authority should the automation have? Should the vehicle just change lanes whenever it wants to change (Sheridan Level 10)? Should do and allow the driver to override (Sheridan Level 6)? Should it as for permission each time before it changes lanes (Sheridan Level 5)? Should it change lanes only when asked (Sheridan Level 1)?

Q3: WHAT ARE THE MOST IMPORTANT EXISTING STANDARDS AND GUIDELINES FOR AUTOMATION?

The most comprehensive list of standards pertaining to automated vehicle appears in the U.S. DOT roadmap on that topic (U.S. Department of Transportation, 2018). Of the many standards listed in Appendix C in the DOT Roadmap, most concern technology and system performance. In spite of its implied completeness, there are actually several relevant standards not listed in that appendix, standards that are listed later in this section.

- In terms of the human interface, standards relating to this topic are being developed primarily by the Society of Automotive Engineers but there are some related efforts in the International Standards Organization (ISO TC 22/SC 39 –Road Vehicles, Ergonomics). The SAE effort is difficult for the outsider to understand, because the activities are distributed among several committees. To get as sense of the SAE efforts, go to SAE.org and from the menu select standards and then standards works. Accessing this page can be confusing, because the page thereby selected is titled "SAE Standards Development," an alternative menu entry on the main SAE page. Scrolling down towards the bottom of the information provided, under the heading Motor Vehicle Council, Vehicle Safety Systems Group, the information in Table 3-1 appears. There should also be activities for trucks and buses, but they are less well known. Thus, to summarize what has been said so far, the guideline is as follows.
  Guideline 3-1: Make sure the HMI complies with standards of the SAE Safety and Human Factors Committee and committees reporting to it (Adaptive Cruise Control and Forward Collision Warning Committee, Adaptive Cruise Control and Forward Collision Warning Committee, the Driver Metrics, Performance, Behaviors and States Committee, the Driver Vehicle Interface (DVI) Committee, the Advanced Driver Assistance Systems (ADAS) Committee, and the Driving Automation Systems Committee). Also important is the On-Road Automated Driving (ORAD) committee.

Table 3-1. Passenger Groups within SAE Developing Standards for Automated Driving Systems

Vehicle Safety Systems Group

Safety and Human Factors Standards Steering Committee o Vehicle Sound for Pedestrians o J2831 In-Vehicle Text Messaging Task Force o J2395 ITS In-Vehicle Message Priority Task Force o **Adaptive Cruise Control and Forward Collision Warning** o **Driver Metrics, Performance, Behaviors and States Committee** o **Driver Vehicle Interface (DVI) Committee** o **Advanced Driver Assistance Systems (ADAS) Committee** o **Driving Automation Systems Committee**
o Driver Vision Standards Committee
Occupant Protection and Biomechanics Steering Committee
Driver Assistance Systems Steering Committee

- **On-Road Automated Driving (ORAD) committee**
  - Infrastructure Needs Related to Automated Driving ORAD
  - On-Road Automated Driving (ORAD) Definitions Task Force
  - ISO_SAE Automation Taxonomy and Definitions Joint WG
  - On-Road Automated Driving (ORAD) Planning Task Force
  - On-Road Automated Driving (ORAD) Verification and Validation
  - ORAD Reference Architecture and Interfaces (RAI)
  - Identifying ADS-DV User Issues for Persons with Disabilities
  - On Road Automated Driving Maneuver Task Force
  - ORAD ADS Testing
  - J3018 Task force
- **Active Safety Systems Standards Committee**
  - Active Safety Test Target Validation_Correlation Task Force
  - Active Safety Bicycle Test Targets Task Force
  - Vehicle Lane Deviation Warning Task Force
  - Hardware-in-the-Loop Working Group
- Active Safety Road Markings for Machine Vision Systems WG
- Crash Data Collection and Analysis Steering Committee

Within the Vehicle Safety Systems Group, the active committees are several reporting to them are the Safety and Human Factors Steering Committee, the On-Road Automated Driving (ORAD) committee, the Active Safety Systems Standards Committee, and others. Those who want additional information on these committees should find the URLs in Table 3-2 to be useful. Additional URLs can be identified by going to the SAE committee page and clicking on the committee name (https://www.sae.org/servlets/works/),

Table 3-2. Useful SAE URLs

Committee	URL
Safety and HF Steering Committee	https://www.sae.org/servlets/works/committeeHome.do?comtID=TEITSSHF
ORAD	https://www.sae.org/servlets/works/committeeHome.do?comtID=TEVAVS
Active Safety Systems Standards Committee	https://www.sae.org/servlets/works/committeeHome.do?comtID=TEVSSAS
Advanced Driver Assistance Systems (ADAS) Committee	https://www.sae.org/servlets/works/committeeHome.do?comtID=TEVSHF3
Driver Vehicle Interface (DVI) Committee	https://www.sae.org/servlets/works/committeeHome.do?comtID=TEVSHF2

There are a number of documents that specifically concern the subject matter of this report (Table 3-3). Given time and response, the specific requirements in them will be merged into these guidelines in a future version, if funded. The belong in the section on specific interface requirements, not here as stand-alone requirements. SAE J3016 has already be summarized.

Table 3-3. Most Relevant Standards and Related Documents

Category	Document	Title	Committee
primary	SAE J2399:2014	Adaptive Cruise Control (ACC) Operating Characteristics and User Interface	ADAS committee
	SAE J2400:2003	Human Factors in Forward Collision Warning Systems: Operating Characteristics and User Interface Requirements	ADAS committee
	SAE J2802:2015	Blind Spot Monitoring System (BSMS): Operating Characteristics and User Interface	ADAS committee
	SAE J2808:2017	Lane Departure Warning Systems: Information for the Human Interface	ADAS committee
	SAE J3016:2018	Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems	ORAD
	SAE J3018:2015	Guidelines for Safe On- Road Testing of SAE Level 3, 4, and 5 Prototype Automated Driving Systems	ORAD
	SAE J3048:2016	Driver-Vehicle Interface Considerations for Lane Keeping Assistance Systems	ADAS committee
	SAE J3077:2015	Definitions and Data Sources for the DVI	DVI committee
	SAE J3114:2016	Human Factors Definitions for Automated Driving and Related Research Topics	Driving Automation Systems Committee
secondary	SAE Recommended Practice J2944:2015	Operational Definitions of Driving Performance Measures and Statistics	Driver Metrics, Performance, Behaviors and States Committee
	ISO standard 2575:2010	Road vehicles — Symbols for controls, indicators, and tell-tales	ISO TC22/SC39/WG8
	SAE J3063:2015	Active Safety Systems Terms and Definitions	Active Safety Systems Standards Committee

In addition to these specific standards, there are 2 more general standards of particular note that the author largely wrote -- SAE J2944 (driving performance definitions, Society of Automotive Engineers, 2015) and SAE J2365 (task time predictions, Society of Automotive Engineers, 2016). Those assessing driving performance should careful read J2944 and use it to guide their measurements. J2365 allows one to predict task times, though some aspects of contemporary HMIs are difficult to predict. Research at UMTRI is underway to improve the predictions of touch-screen applications and predict occlusion times from the NHTSA guidelines.

Finally, for those unfamiliar with them, the NHTSA guidelines provide restrictions of what can be presented (e.g., no album art, no social media browsing) and provide 2 methods to determine if tasks are excessively distracting. The most widely used method involves visual occlusion, in which subjects wear a pair occlusion goggles that allows them to see and not see the interface for 1.5 s intervals, simulating looking between the road (and not seeing the interface, the occluded period) and seeing the interface (the unoccluded period). The test sample is required to include 24 subjects equally distributed among 4 groups ranging in age from 18 to 55+. (See U.S. Department of Transportation, 2013 for details.)

In addition to all of this material, the author is preparing YouTube and PowerPoint summaries of the major human factors guidelines and the SAE standards relevant to HMIs for the National Institute of Standards and Technology (NIST). They should be available in the first quarter of 2019. Again, see also Green (2018a, b).

Q4: WHAT ARE THE MAJOR PROBLEMS DRIVERS ENCOUNTER WHEN USING AUTOMATED DRIVING SYSTEMS?

The principles in the human factors literature would lead one to believe that designing human-automation interfaces is simply another application of the knowledge and existing guidelines from the past. Green and Rudin-Brown (2017), in reviewing the literature, suggest that is not true. They reviewed 4 recent articles on the topic of current problems with automated vehicles, summarized here.

Dikmen and Burns (2016)

Dikmen and Burns (2016) conducted an online survey of 162 Tesla owners, 121 of whom completed the full survey. Some 91 automation events, approximately ¾ were related to lane detection, including cases where (1) the car unexpectedly attempted to take an exit ramp, (2) "swerving and veering due to failure to detect the lane," and (3) the vehicle was "trying to cross lanes for no apparent reason," even into traffic in the opposite direction (page 226). The underlying problem the *HMI did not communicate its confidence in recognition of the lane markings*.

Other problems were related to "sudden braking or uncomfortable acceleration and deceleration" adaptive cruise control (ACC), most often occurred in heavy traffic, apparently because drivers were unaware that ACC is engaged. The underlying problem is a *failure to communicate the automation mode.*

Brown and Laurier (2017)

Very cleverly, Brown and Laurier (2017) collected 10.5 hours of uncut from YouTube consisting of 69 clips from 63 uploaders. Except for a few clips concerning the Volvo XC90 and Honda Civic, almost all of the clips concerned Tesla and its Autopilot system. Table 4-1 lists some of the issues identified by drivers, most of which were technology related, but all of which have implications for the HMI design. Most problems arose because the interface did not effectively communicate confidence in its ability to identify where the lane markings were, did not indicate when sensors (and associated functions) were lost, when challenging driving situations were ahead (blind corners), and did not adjust speed for challenging driving situations in the same way as drivers do.

Table 4-1. Some Issues with Automated Driving Systems from Brown and Laurier (2017)

Problem	Possible Driver Interface Solution
Vehicle lurches at highway exit and drives past it as it is unsure where the exit is; Vehicle unintentionally exits highway when lane lines disappear	Show lane marking confidence and where markings are lost
Sunshine overwhelms camera	Indicate when sensor is lost and implications

Blind summits and corners lead to sudden maneuvers of automation failures	Using navigation information, indicate to driver that upcoming segment is difficult for automation
Vehicle needs to slow down more when approaching corners, tunnels, bridges, tight curves	Algorithm issue; change to make driving more human-like
Stop and go feature is ok for intersection approach, but not for driving through intersection	Forewarn driver about the need to take over

Endsley (2017)

Endsley (2017) describes her personal experiences from driving a Tesla Model S (version 7 software) over a 6-month period (Table 4-2). Endsley is probably the best-known expert on situation awareness (SA) and has published extensively on that topic (e.g., Endsley, 1995; Endsley and Garland, 2000; Endsley, 2016). The score indicates how well the Tesla she drove supported situation awareness. Notice the characteristics that were very poor or poor –no confidence information, uncertainty about when to intervene, too many modes, system states (modes) are not salient, and queuing to tasks, which has to do with set up.

Table 4-2. Sorted Scores on Automation Guidelines for Situation Awareness from Endsley (2017)

Score	**Guideline** (from Endsley and Jones, 2012))	**Comment**
very poor	19 – Support assessments of confidence in composite data	No confidence information associated with system was provided.
	20 – Support system reliability assessments	Trust and effective judgments on when to intervene depend on an accurate assessment of system reliability for performing the task at hand. Interfaces should make explicit how well the automation is currently performing and its ability to handle upcoming or contemplated tasks.
Poor	5 – Avoid the proliferation of automated modes	Multiple modes and mode interactions create complexity and unexpected behaviors. Better integration of mode operations and deconfliction of mode activation/deactivation methods are needed to improve mode operation and awareness.
	6 – Make modes and system states salient.	Good display of current modes. Audio cues of unanticipated transitions to manual control lack needed salience. Unique audio cues are also needed to alert driver to partial mode changes (e.g., autosteer off but ACC still on).
	8 – Avoid advanced queuing of tasks	Autosteer, ACC, and navigation all set up tasks to be carried out in advance, which create the lowest levels of SA. Approaches that maintain operator involvement in the

		execution of tasks should be considered.
Average	1 – Automate only if necessary—avoid out-of-the-loop problems if possible	Combined ACC/autosteer creates an out-of-the-loop issue and added complexity in predicting system issue and added complexity in predicting system behavior.
	3 – Provide SA support rather than decisions	Most decisions are left to driver. Speed limit and lane departure warnings are beneficial. Exceedance of speed limit needs continued display with more salient representation (e.g., red outline). Side collision warnings are not salient.
	4 – Keep the operator in control and in the loop	Driver is in control of each automated feature. Driver engagement during autosteer/ACC is low. New strategies are needed to incorporate the driver and improve engagement.
	11 – Human-system symbiosis	Displays of the road and vehicles sensed by the system good for supporting shared SA with driver and provide projection of system actions. Improvements in supporting understandability of system actions and predictability of braking and speed changes are needed.
	12 – Provide automation transparency	Some unexpected behaviors arose from mode interactions, particularly as situations changed. Mode behaviors should be reviewed and modified to consider such interactions in context to conform to driver expectations.
	13 – Ensure logical consistency across features and modes	System complexity increased as new rules were added to the system. Complexity can be minimized by reducing the linkages and conditional operations contained in the autonomy, thereby avoiding modes with their multiple-branch logic as much as possible.
	14 – Minimize logic branches	A clear mapping between user goals and system functions should be present—for instance, merging with traffic or exiting highways could be single-step actions, rather than requiring multiple interactions with different modes and an understanding of how those interact.
	16 – Minimize task complexity	Display of lane and vehicles in front is good. Improved display of objects on the sides, blind spot, and rear of vehicle is needed. Improved display of exceedance of speed limits, obstacles below bumper, and stationary vs moving objects on sides is needed.
	17 – Integrate information to support comprehension of information (Level 2 SA)	Power and range projections are good. Improved display to predict actions of autonomy is needed, particularly those that it cannot handle. Improved display of future road hazards, traffic, and side collision warnings is needed.
	2 – Use automated assistance for carrying out routine	Key decisions regarding route selection and route following are left to driver. Automation of many routine tasks is useful, such as headlight and brights on/off, garage door operation,

	tasks rather than higher-level cognitive functions	backing camera activation, and locking/unlocking. Autobraking is inconsistent.
	7 – Enforce automation consistency	Consistency in the terminology and information placement among modes was good. Some unexpected behaviors arose from mode interactions.
Good	18 – Use information filtering carefully	Automatic replacement of information displays (e.g., presenting phone call information) covered only low priority data.
Very good	9 – Avoid the use of information cueing	No information cueing provided.
	15 – Map system functions to the goals and mental models of users	Actions to interact with the automation were simple and intuitive.
N/A	10 – Decision support should create human-system symbiosis	No decision support systems provided

To support this report, the entire set of the 50 guidelines Endsley (2017), except for those pertaining to training, appear in Appendix D. Of the 50 guidelines, those concerning multiple operators (guidelines 45-50) are unlikely to be applicable to motor vehicle design, but they are provided for the sake of completeness. For this application, the guidelines would be more useful if examples were provided.

Disengagements

One of the major concerns with autonomous vehicles is that there will be a problem with the automation and the driver will need to take over, potentially unexpectedly and immediately. Currently, for autonomous vehicles driving in California, disengagement reports are required. These reports describe incidents where either the automation requested a takeover (passive disengagement) or the driver monitoring the automation decided to take over (active disengagement), both of which are instances of AH transfer. Three recent articles have summarized the data so far, apparently up to January of 2017. Depending on which data is used, there is information from 7-8 manufacturers and anywhere from 2500 to 5000 plus disengagement over 500,000 to 1 million miles and as many as 42 crashes (Lv et all, 2018; Favaor, Eurich, and Nader, 2018 (see also Favarò, Nader, Eurich, 2017; and Banerjee et al, 2018). As the raw data are not coded, each study does that slightly differently, and accordingly, draws slightly different conclusions about the causes of disengagements. In analyzing the data, it can be done pooled or with splitting Google out as the largest share of the data is from their vehicles.

The problems differ with the manufacturer and if the intervention was automation or driver initiated. However, they are still valuable in identifying use cases that deserve emphasis and in

providing insight to drivers concerning the conditions under which automation failures are most likely.

The Favaor et al (2018) analysis is the most complete (Table 4-3). Note that about half of the failures are due to hardware and software, though studies suggest that value is close to 80% due to that cause. Notice the situations in which drivers want to take over (human factors category) involve objects maneuvering erratically. Although no systems do this now, this research suggests that drivers could find it useful to be alerted of these situations, when automation is controlling the vehicle.

Table 4-3. Causes of Disengagements

Category	Subcategory	%
Human factors (30.1 %) test driver decides to intervene	recklessly behaving agent	2.8
	precautionary spacing; cyclist	0.5
	driver discomfort	26.8
System failure (52.2) Automation hardware or software fails	software discrepancy	13.1
	perception discrepancy	6.0
	planner not ready	5.7
	traffic light detection	4.8
	lane change	4.3
	unwanted maneuver of vehicle	3.6
	other failures	14.8
External (10.8 %) e.g., road debris	poorly marked lanes	4.6
	construction zone	2.2
	heavy pedestrian traffic	2.0
	weather condition	0.7
	other factors	1.2
Other (6.9 %)	planner output invalid	3.1
	follower output invalid	3.6
	ACC cancel	0.1
	health monitor	<0.1

Adapted from: Favaor et al (2018), p. 142-143.

In terms of where disengagement occurred, the percentages are 49% street, 38% interstate, highway 7%, freeway 5%, city/highway 1%, and arterial road <1%. Unfortunately, these terms are not consistent with FHWA definitions, and are not defined. Furthermore, the percentages vary by manufacturer and depend upon where the vehicles were tested (exposed). Of these events, 82% occurred in clear weather, 16% when it was cloudy, 1% in rain, and 1% in snow.

Additional detail appears in Favaro et al (2017), which provides a summary table of each crash. Also provided is information showing the mode associated with the crash (mostly autonomous, Table 4-4), various factors (most intersection, Table 4-5), and the relative speed (10 mi/hr or less Table 4-6). Bear in mind that many of these crashes were associated with Google vehicles driving in Mountain View, CA.

Table 4-4. Operating Mode Associated with Crash

Mode	%
autonomous	62
manual disengagement before collision	19
manual	15
manual disengagement after collision	5
autonomous disengagement	0

Table 4-5. Factors Associated with the 26 Crashes

Factor	# Crashes
Intersection	23
Traffic light	14
Right turn	8
Turn lane	6
Slip lane; stop sign	4
Yield sign; lane change	2
Shoulder; left turn; bike lane	1

Table 4-6. Relative Speed Associated with Crash

Speed (mi/hr)	%
0-5	28
5-10	32
10-15	4
15-20	8
20-25	4
25-30	4
n/a	20

Also provided is a figure showing each crash scenario (Figure 4-1). Notice that a significant number of the crashes involve the autonomous vehicle being struck in the rear by other vehicles.

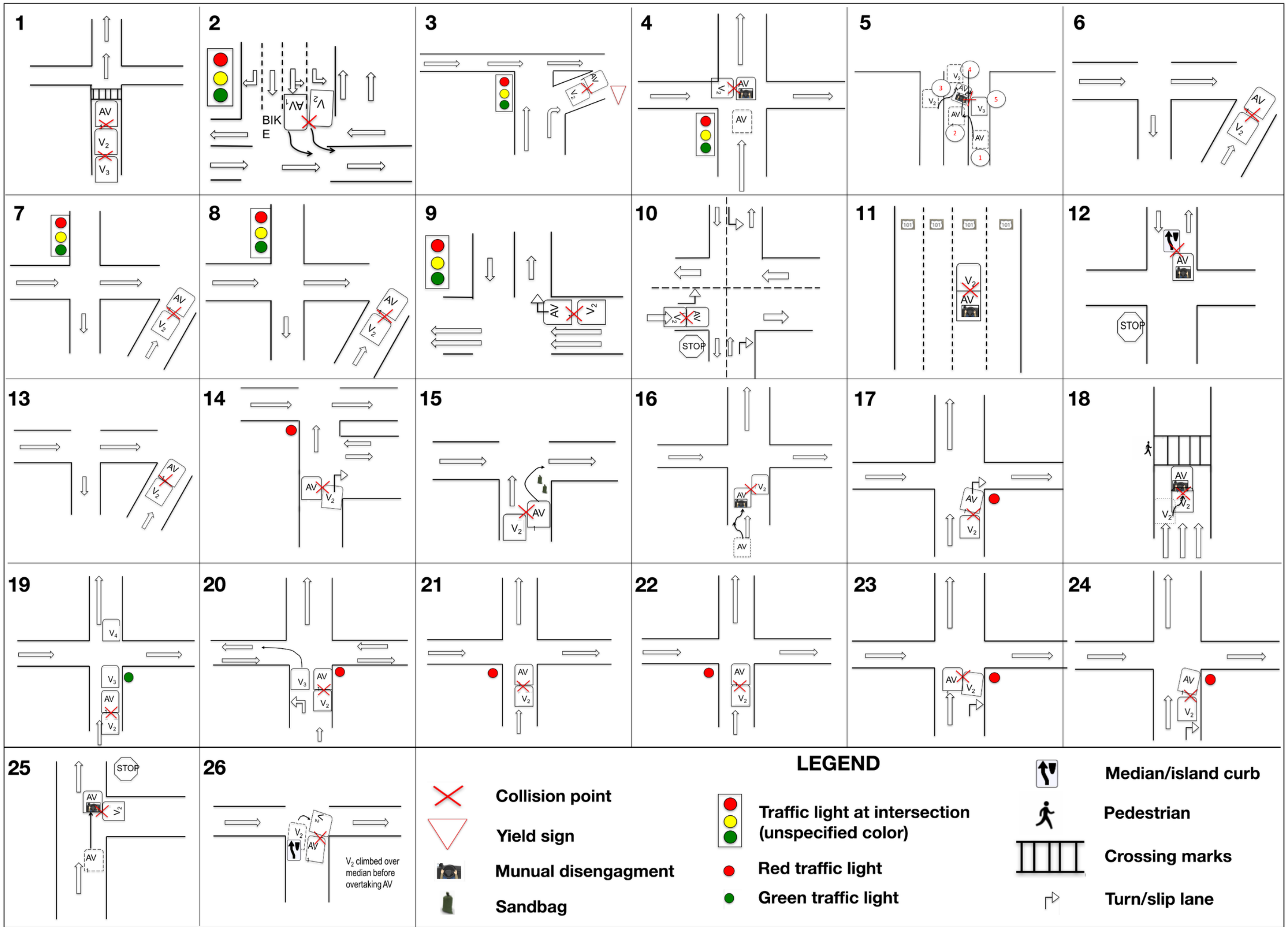
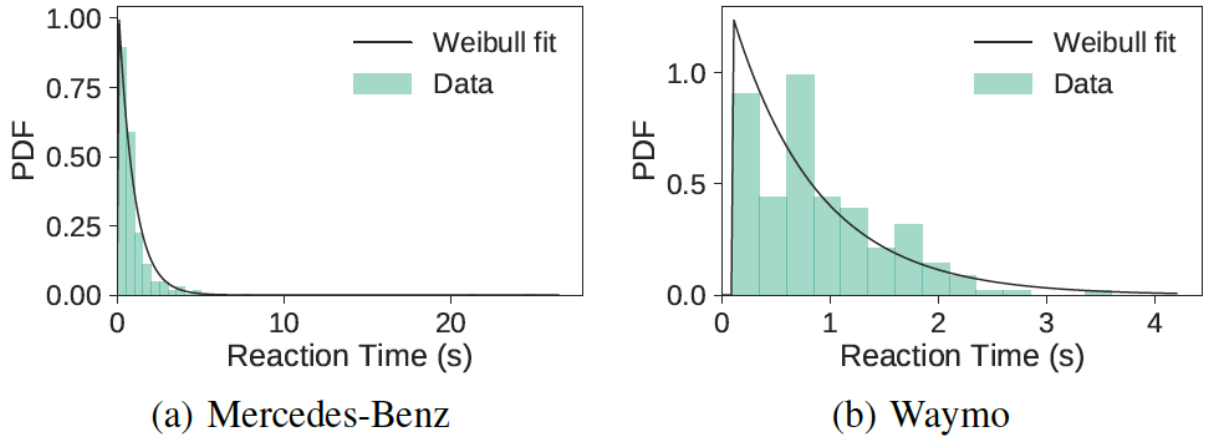


Figure 4-1. Disengagement Crash Scenarios Reported by Favaro et al (2017)

One other insight from these studies are data on driver reaction times (Banerjee et al., 2018), defined as "the period of time elapsed from when the autonomous vehicle test driver was alerted of the technology failure, and the driver assumed manual control of the vehicle". What constitutes resuming control is undefined.

Figure 4-2 shows the driver reaction times for 2 manufacturers, with a mean time of 1.09 s. The data are easy to misinterpret. First, the time required to be confident these drivers can react in time is at least 4 seconds. Second, these drivers are trained test drivers, who should respond more quickly that the driving public on average.



(a) Mercedes-Benz (b) Waymo

Figure 4-2. Reaction Times to Take Over (AH Transfer) Alerts.
Note: Even though the x-axis scaling is different, the 2 distributions are quite similar.
Source: Banerjee et al, (2018), p. 895.

Q5: WHAT ARE SOME GUIDELINES THAT CAN HELP DESCRIBE AND SIMPLY THE INFORMATION ARCHITECTURE?

High-Level Guidelines

Based on the previous section, there are consistent problems with these automation systems. The common problems from the literature lead to guidelines 1, 3, and 4 in Table 5-1. The remaining guidelines are form personal experience in driving a few partially automated vehicles and the clinics.

Table 5-1. High-Level Guidelines

Category	#	Problem	Guideline
Roadway Situation	1	Data Confidence – The driver does not know how confident the vehicle is in detecting lane markings. This could be due to poor lane marking quality, sunlight on the sensor, or other reasons.	**5-1: The vehicle shall show how confident the vehicle is in the data it is using to drive, in particular lane markings.** See Note 1 below.
	2	What the Vehicle Does See – There could also be issues with object detection.	See Note 2 below.
	3	Path Uncertainty –The driver in unsure about the path the vehicle is going to take, a byproduct of uncertainty about lane markings. This can also occur due to a rapidly changing road geometry (e.g., sharp curves) of which the vehicle should know from GPS and map matching data.	**5-2: Where there are alternative paths or the vehicle is changing paths, the vehicle shall show how confident the vehicle is in the path it intends to take/is taking (for example when under navigational control or using automatic lane change features).**
Mode and Mode Changes	4	Uncertain current mode – The driver does not know or is not sure what mode the vehicle is in. In addition, personal experience with driving partially automated vehicles suggests the following as concerns.	**5-3: The vehicle shall show who is steering (driver, automation, shared), who is controlling speed (same), who is responsible for object and event detection (same), and if a transition is occurring.**
	5	Operational Design Domain for Mode Is Uncertain – The operational design domain in which a mode is available is uncertain.	**5-4: The operational design domain for a mode shall be shown. For speed control this includes the range of speeds within which it operates. For gap control, the range of gaps shall be indicated.**

	6	Mode Settings – How settings are adjusted can be inconsistent, which can make them difficult to do.	**5-5: Except for the control used, the sequence of actions to change speed and gap settings should be the same.**
	7	Takeover Clock – When asked to take over (AH Transfer), the driver does not know the time frame in which that needs to occur.	**5-6: Where the driver is to take over control of the vehicle (AH transfer), the time period over which that shall occur and the time remaining shall be indicated.**
	8	Temporary Take Over (AH Transfer)– Drivers grab or hold on the steering wheel or press on the accelerator, not realizing they override the automation.	**5-7: When a temporary takeover (AH transfer) occurs, what the driver did for that to occur shall be indicated. In addition, guideline 3 will be followed. Of particular concern is where maneuvers (e.g., lane change) are interrupted.**

Notes: 1: In some vehicles, vehicle position is determined not just by the lane markings, but by the position of the lead vehicle and GPS data. Drivers may not be aware of this because lane tracking problems often occur when the lane markings are degraded.

2: A guideline is not provided here, because it is uncertain how to show what the vehicle has not detected.

3. It is not known how much drivers know about how lane tracking systems actually work. In some systems, there can be complete reliance on the lane markings, and given that is how drivers control their lane position, they are likely to think vehicles operate in the same manner. However, if the lane markings are lost, some automated driving systems track the lead vehicle if it is in the same lane, which drivers could do, but only when the forward scene is highly degraded. Furthermore, some vehicles may use highly accurate GPS data, which of course people cannot use by themselves to drive.

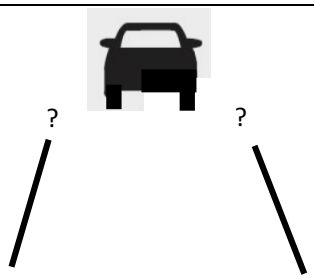
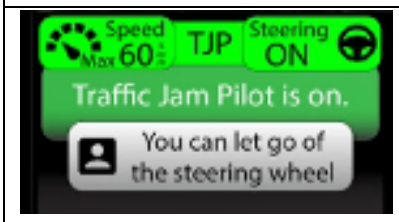
Thus, one could summarize most of these guidelines as answering the following 8 questions.

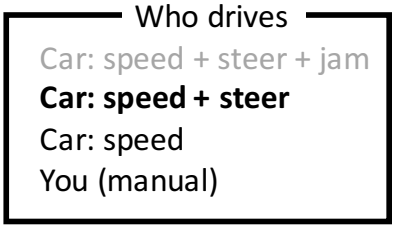
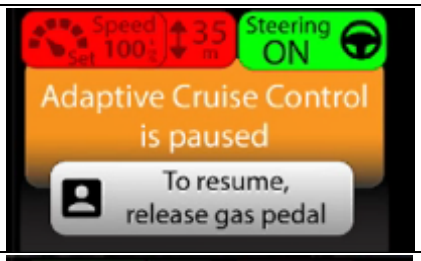
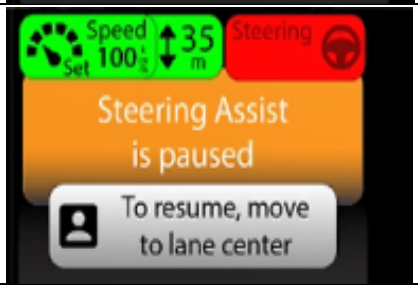
1. What does the automation know about the driving situation?
2. How well is what the automation knows about driving situation communicated to the driver?
3. What is the automation doing now?
4. What can the automation do now?
5. What does the automation intend to do and when?
6. What is the driver doing to the automation now?
7. What can the driver do now?
8. What does the driver need to do and when?

To reduce these concepts further, it is what the driver knows that the vehicle automation, what the automation is/can/ intends to do and what the driver is/can/intends to do.

Examples of Applications of the High-Level Guidelines

Table 5-2. Examples of Applications of High-Level Guidelines

Guideline	Examples	
5-1: The vehicle shall show how confident the vehicle is in the data it is using to driving, in particular lane markings.		This was not done but should be. Possibly the length of the lane marking of question marks could indicate confidence. Because the driver needs to see this while looking at the road, the display should be on a HUD.
5-2: Where there are alternative paths or the vehicle is changing paths, the vehicle shall show how confident the vehicle is in the path it intends to take/is taking (for example when under navigational control or using automatic lane change features).	None, but needed in next update	
5-3: The vehicle shall show who is steering (driver, automation, shared), who is controlling speed (same), who is responsible for object and event detection (same), and if a transition is occurring.		Example HUD: In this case, who is doing what (steering, speed control, object detection) is shown, but the format is inconsistent.
		Example HUD: Part 1 of the 2-part message. Drivers needed to know they could let go of the steering wheel.
		Example HUD: Part 2 - In this case, it is assumed the vehicle is in full control because not instructions otherwise are provided.

5-4: The operational design domain for a mode shall be shown. For speed control this includes the range of speeds within which it operates. For gap control, the range of gaps shall be indicated.		This was not done. Some have proposed to put marks or rings on analog speedometers to show the range of ACC. One step in that direction would be a display that shows the modes that are available and engaged.
5-5: Except for the control used, the sequence of actions to change speed and gap settings should be the same.	Guideline was not followed because the controls used for clinic 2 were to be as in the current production vehicle.	
5-6: Where the driver is to take over control of the vehicle (an AH transfer), the time period over which that shall occur and the time remaining shall be indicated.		The ball rotates around the digit, analogous to clock hands (which can be detected peripherally) but without obscuring the digit. With 1 rotation/second, one does not to look for an entire second to know the time is changing.
5-7: When a temporary takeover (an AH transfer) occurs, what the driver did for that to occur shall be indicated. In addition, guideline 3 will be followed. Of particular concern is where maneuvers (e.g., lane change) are interrupted.		Example HUD: In this case, it is implied the gas pedal is being pressed, but ACC being paused is clearly indicated.
		Example HUD: In this case, steering assist has been overridden by steering.

Q6: WHAT MAKES AN INTERFACE COMPLEX (AND NOT EASY TO USE)?

One of the problems associated with many automated driving systems is that (1) they are complex and (2) their operation is not apparent to drivers. When developing an automated driving system that is easy for drivers to use, one must consider both the underlying logic and how the HMI represents that logic. Most HMI design focuses on the second topic, but in this section, the emphasis is on the first topic as it appears to be the more significant in view of the prior section that described problems drivers have with existing partially automated vehicles. Again, these problems are often due to architectural choices. If there are too many modes and too many exceptions due to the operational design domain, then the system is likely to be difficult to operate, not matter how much effort is put into developing and testing the HMI.

How Can System Operation Be Represented (in General)?

There are at least 3 ways to represent the how a system functions (Olsson, 2016). One representation is as a finite-state machine. A finite-state machine consists of a number of predefined states, with all the computation occurring within a state (Figure 6-1). Finite-state machines are often used to represent systems because they are simple and require the least explanation to those not familiar with computer science or systems engineering. However, as systems become large and functionality grows, maintaining the keeping the finite state machine current, in particular the links, becomes difficult. HMIs for automated driving systems, in particular, Levels 1-3, probably have not yet reached that point. See Zhang, Girard, and Kolmanovsky (2017).

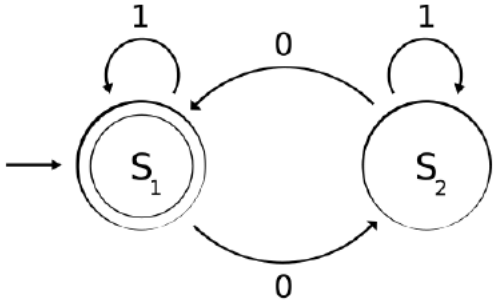


Figure 6-1. Example of a Finite-State Machine
Source: Olsson, 2016, p. 7

An evolution of finite state machines is hierarchical finite state machines (or hierarchical state machine), sometime referred to as state charts (Harel, 1987; Harel and Naamda, 1996; Figure 6-2), which are supported by MATLAB/Simulink and other software. The key difference from finite-state machines is that states can be grouped into superstates and properties are inherited, simplifying the description.

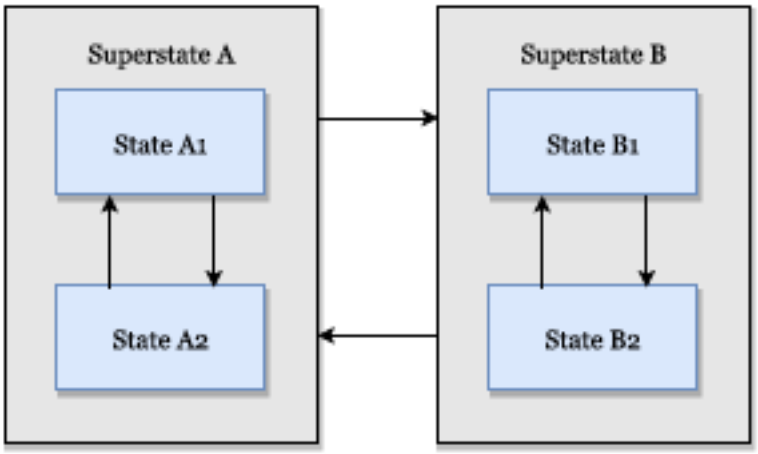


Figure 6-2. Example of Hierarchical State Machine
Source: Olsson, 2016, p. 8

A third, but much less well known alternative, is behavior trees, which represent the system as a hierarchical tree structure (Figure 6-3). There are 6 types of nodes that vary depending upon if the child leafs are acting in parallel or not, and the number of successes of the node children. They have advantages of making the diagram representing the system easier to modify because elements are more independent, and are popular for creating video game architectures (and potentially, driving simulators), and used for control systems and in robotics. For additional information, see https://en.wikipedia.org/wiki/Behavior_tree. For an example application, see Kurt and Ozguner (2013).

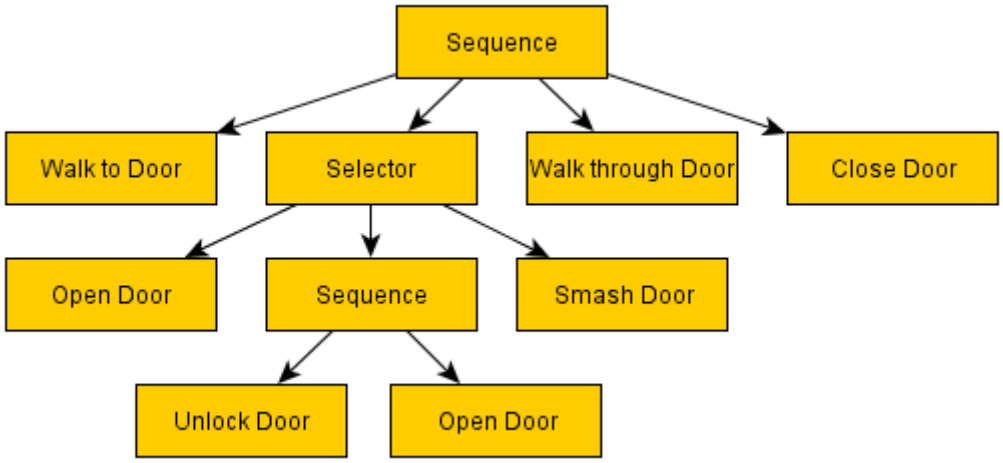


Figure 6-3. Example of Behavior Tree
Source: http://outforafight.files.wordpress.com/2014/07/selector1.png

Alternatives to these 3 representations have yet to be examined. However, Olsson (2016) provides an interesting examination of them from strictly a software perspective. An extension

of that work to explore the human logic associated with the software could provide some insights in human perspectives of system complexity. For related information on this topic, see Ledang and Souquières (2002) and Moody (2009).

How Has Automation Architecture Been Represented?

Human use of automation often concerns the transitions, and thus representations of those activities, in particular inconsistencies often are shown as matrices of various types. The inconsistencies are important, because as noted later, they indicate the driver needs to learn more rules about how to operate the system, and the greater the number of rules required, the more complex the system is.

Finite State Machine Representations

Asaf Degani has written a number of papers related to understanding complex devices, initially aircraft automation (Degani, Heymann, and Shafto, 1999). For a number of years has been working at GM and applying the ideas that follow there. As an example, Heymann and Degani (2013) present an example of a hierarchical state chart for an adaptive cruise control system (Figure 6-4).

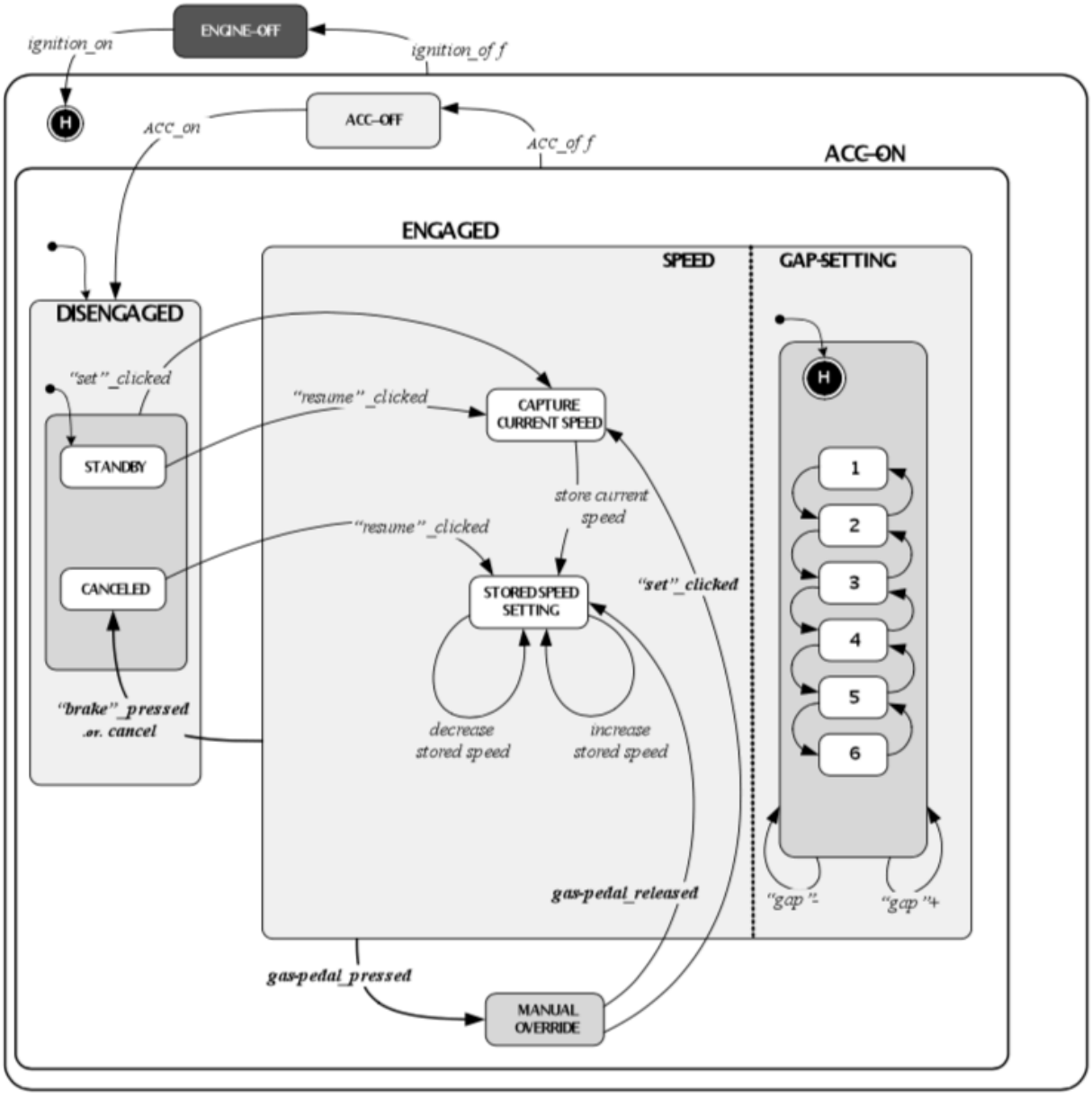


Figure 6-4. Hierarchical Finite State Machine Representation of ACC
Source: Heymann and Degani (2013), p. 145.

Degani and Heymann (2002) describe how a state machine can be used to represent how a device operates (including automated modes) and what a person can do to operate the device, resulting in a composite description. They provide some insights into how the model can be used to formally evaluate a user interface, in their example, an aircraft altitude hold system. Readers may find Heymann and Degani (2007) more helpful because it provides more specific advice about creating user interfaces. Providing those details is appropriate in a further revision of this guidelines report. As an example, they present the image in Figure 6-5 to represent a transmission. To explain, the transmission has 3 basic ranges, low, medium, and high that the user can manually select, but within them are 2 or 3 levels the system automatically switches

between depending upon torque, engine RPM, speed, etc. These states (modes) are nested within the high, medium, and low states.

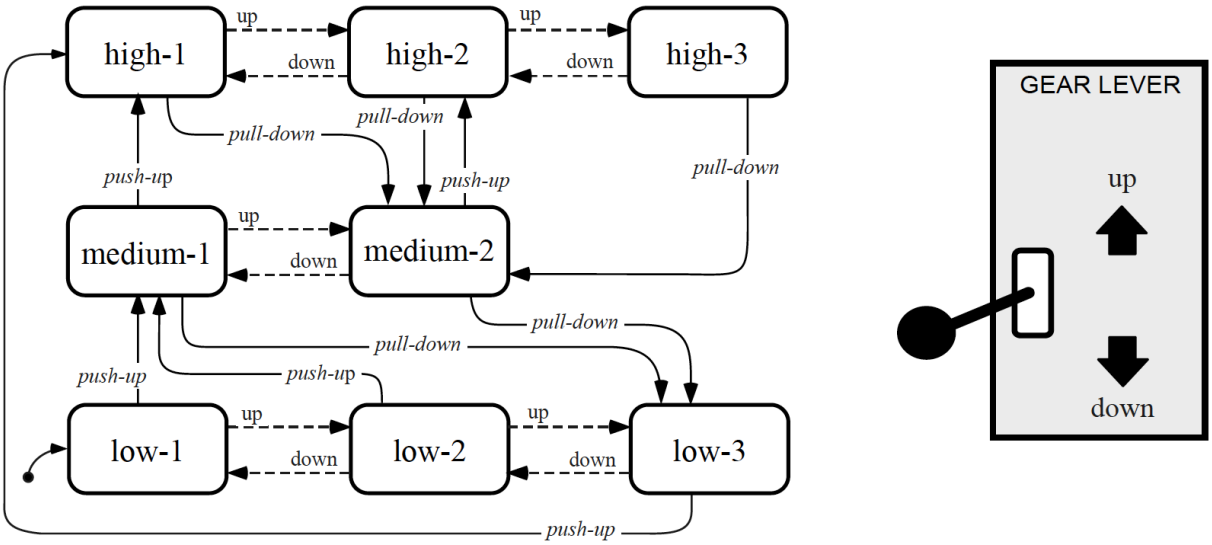


Figure 6-5. Vehicle Transmission State Diagram and Shift Lever
Source Heymann and Degani (2007). p. 314.
Note: User-triggered actions are solid lines, automation triggered are dashed.

ETIT-like representations

When HMI evaluators think about display design, they commonly think about how easy to use the controls and displays are, independent of the underlying operation logic. How can the representation of the logic be improved? Moran (1983 describes an external task-internal task analysis, a mapping between what the user is to do and how they do it. The goal of the mapping is to highlight the consistencies and inconsistencies between the 2. Figure 6-6 shows an ETIT map for the DISPED editor. Each cell represents a task the user is to perform, which is comprised of the object on which the user is to act and the action to be performed. The entry in the cell concerns the method by which the task is to be performed and related information. (See Moran, 1983 for additional details). Ignoring the details of which rules the user follows to complete a task, notice that regardless of the object (character, word, etc.), what the user needs to do (the method), for example to insert something, is always the same. This consistency makes tasks easier to do because there are fewer rules to learn and because the rules are consistent (and thus, there are fewer exceptions). Thus, what this diagram allows one to do is to spot inconsistencies in task methods at a glance, inconsistencies that make the interface difficult to use.

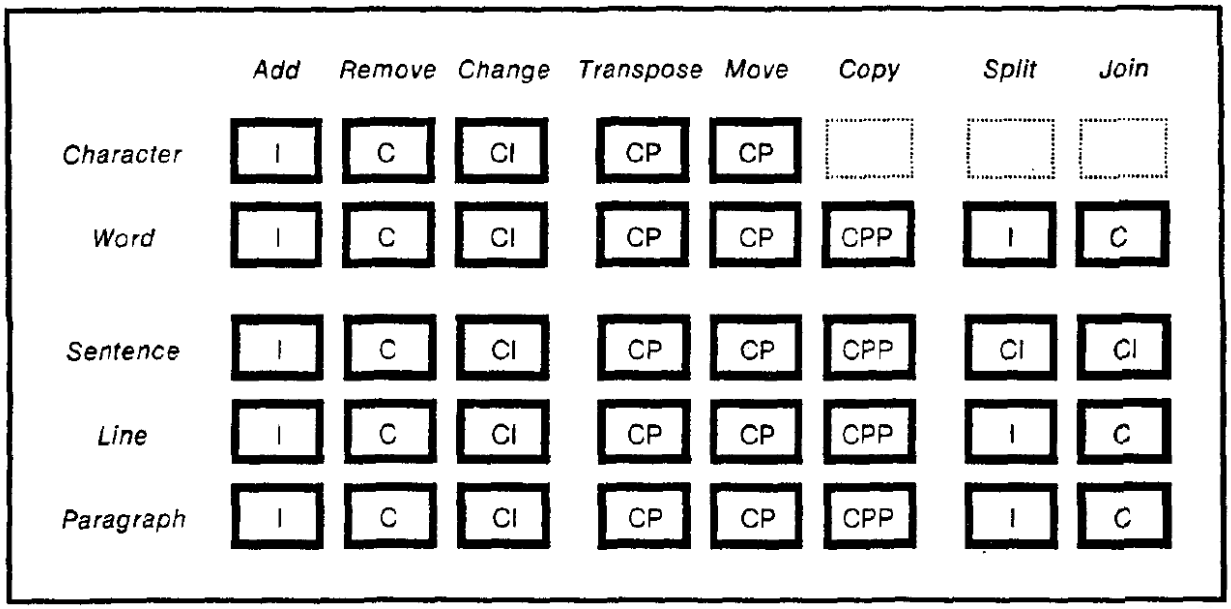


Figure 6-6. ETIT Mapping for DISPED Editor
Source: Moran (1983), p. 47
I = insert string, C = cut string, P = paste string

This diagram does not indicate the details of the method to accomplish a task. Where there are variations in the method or where the number of step to methods vary could be indicated in the table. Of course, this runs the risk of making the table difficult to understand and less useful.

Sometimes, seeing the method in detail is revealing. For example, suppose for example, to insert a character, the process was:

1. move the cursor to the insertion point using a mouse
2. click the mouse button
3. type in the desired character

However, for a word, the process was:

1. move the cursor to the insertion point using a mouse
2. click the mouse button
3. press the insert word function key
4. a box appears with a window to enter text
5. enter the text
6. hit return when done.

Although these methods share the first 2 steps, beyond that, they are quite different. Obviously, having a different method for inserting a character and inserting a word would not be a good design because this added complexity requires the user to learn more rules to perform both tasks, making the interface more complex and more difficult to use. An ETIT diagram can make those differences readily apparent, but it does not show the extent of the difference.

Another use of these diagrams is to show inconsistencies between systems, what the user needs to learn. As an example, Figure 6-7 shows the ETIT map for the LINED editor. Notice the large number of differences in cell entries, reflecting the number of rules the user needs to learn.

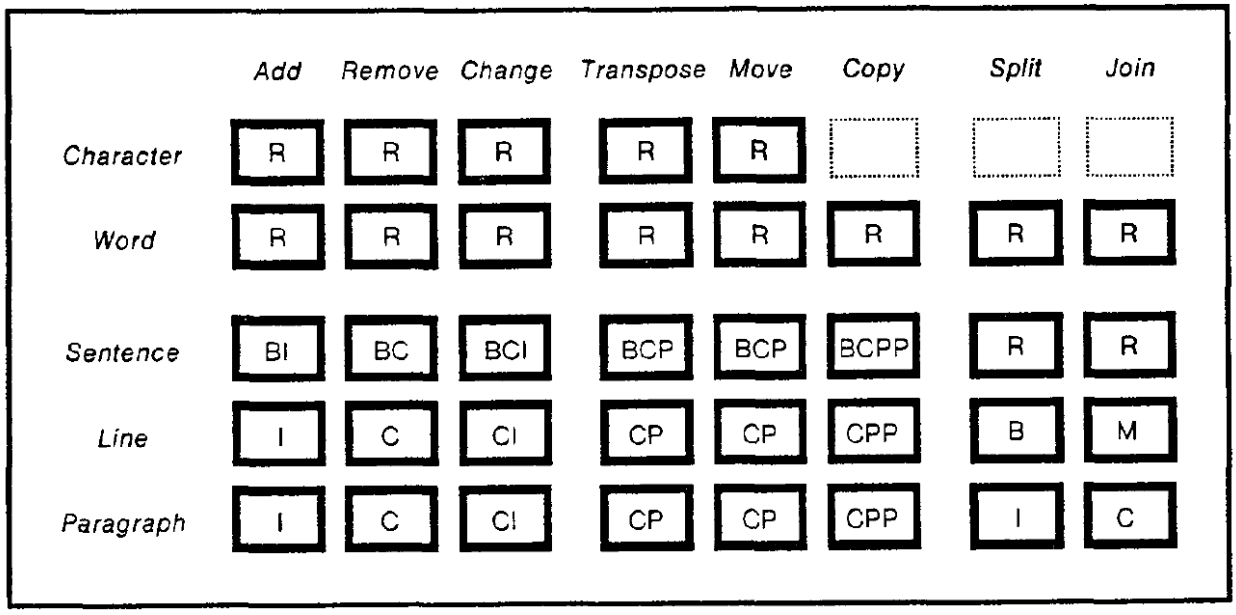


Figure 6-7. ETIT Mapping for LINED Editor
Source: Moran (1983), p. 48
I = insert line, c = cut line, p = paste line, b = break line, m= merge line, r= replace string

For motor vehicles, drivers are likely to have some prior knowledge, for example, how manual cruise control works. Thus, one not only needs to consider the total number of rules and their complexity when operating a system, but also the number of new rules for the user to learn. (More on this later.) In those cases, a diagram or some other representation of current user knowledge is desired.

Furthermore, from a manufacturer's perspective, there can be value in creating such diagrams for a range of vehicles to support the goal of common methods of operation across a product line.

To reiterate, the level of detail in a cell will vary with the goal for the analysis. If detail is desired, then one should probably use production rules as described else. Here what makes sense is a higher-level summary, for example, the name for the rule sequence.

To make this discussion more concrete, consider a hypothetical setting task. Suppose the automation system has (1) lane keeping (and the driver can set how stringently the lane center is maintained), (2) adaptive cruise control (for which speed and gap can be set), and (3) automatic lane change (whose aggressiveness could be set). These systems could have the settings in Table 6-1, examples provided only for the sake of discussion.

Table 6-1. Automation Settings

Lane keeping	Adaptive cruise control		Lane change
	Speed	Gap	
strict	faster	more	quick
moderate	current speed	current gap	normal
loose	slower	less	slow
weave			gradual

Furthermore, suppose the 3-button interface in Figure 6-8 was used, and the operating logic was each push of a button would lead to 1 operational cycle. (Thus, there would be no hidden logic such as push and hold or some repetition function that would depend on how long the button is depressed.) At this point, ignore the question if there were 3 different buttons for each system or 1 set of buttons and an automation mode setting switch.

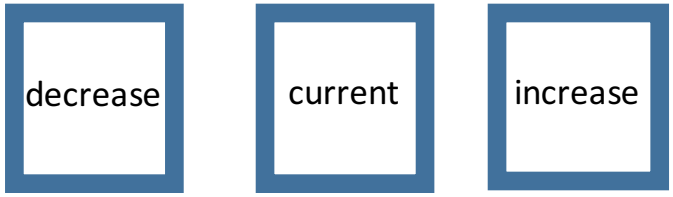


Figure 6-8. Hypothetical 3-Button Interface

How could the setting tasks associated with the 3 automation systems be accomplished? Those buttons could be implement in at least 3 ways (methods), (1) limit freeze, (2) rollover, and (3) reverse. Do not assume that because they are listed as examples, that these 3 methods are recommended. In fact, the reverse logic seems a bit odd, but if the goal was to discourage users from selecting minimum or maximum values, this method could accomplish that goal.

See Table 6-2 for the logic for these 3 methods. As a starting point, in all 3 cases, pushing the current button selects the current value.

Table 6-2. Code to Implement the Selection Options

Method 1 - Limit freeze

If (the current button is pressed) then (value = current value). Exit.
Else
If (the increase button is pressed) and if (the current value is not the maximum)
Then (increase the current value 1 step). Exit.
Else (the current value *does not change*). Exit. ** It is at maximum **
Else
If (the decrease button is pressed) and if (the current value is not the minimum)
Then (decrease the current value 1 step). Exit.
Else (the current value *does not change*). Exit . ** It is at minimum **

Method 2 - Rollover (circular linked list)

If (the current button is pressed) then (value = current value). Exit.
Else
If (the increase button is pressed) and if (the current value is not the maximum)
Then (increase the current value 1 step). Exit.
Else (the current value is the *minimum* value). Exit. ** rollover **
If (the decrease button is pressed) and if (the current value is not the minimum)
Then (decrease the current value 1 step). Exit.
Else (the current value is the *maximum* value). Exit. ** rollover **

Method 3 - Reverse

If (the current button is pressed) then (value = current value). Exit.
Else
If (the increase button is pressed) and if (the current value is not the maximum)
Then (increase the current value 1 step). Exit.
Else (*decrease* the current value 1 step). Exit. **reverse direction at maximum **
If (the decrease button is pressed) and if (the current value is not the minimum)
Then (decrease the current value 1 step). Exit.
Else (*increase* the current value 1 step). Exit. **reverse direction at minimum **

Given the system settings, the 3-button interface, and the methods for how the buttons operate, how could this setting interface be implemented.? Table 6-3 presents some examples.

Table 6-3. Example of Representing Operational Alternatives

Option 1: Inconsistencies are apparent and they are both within and across systems. – poor design

User Intention	**Automation System**			
	Lane keeping	**Adaptive cruise control**		**Lane change**
		Speed	**Gap**	
current	limit freeze	rollover	reverse	rollover
increase	limit freeze	rollover	reverse	rollover
decrease	rollover	limit freeze	rollover	reverse

Option 2: Consistent logic, but the logic chosen is unexpected. – poor design

User Intention	**Automation System**			
	Lane keeping	**Adaptive cruise control**		**Lane change**
		Speed	**Gap**	
current	reverse	reverse	reverse	reverse
increase	reverse	reverse	reverse	reverse
decrease	reverse	reverse	reverse	reverse

Option 3: Consistent logic, logic is expected. – ok design

User Intention	Automation System			
	Lane keeping	Adaptive cruise control		Lane change
		Speed	Gap	
current	limit freeze	limit freeze	limit freeze	limit freeze
increase	limit freeze	limit freeze	limit freeze	limit freeze
decrease	limit freeze	limit freeze	limit freeze	limit freeze

Keep in mind that what needs to be considered is just not the method of operation for the automation, but all the other systems as well. Representations for that purpose are desired as well.

The key point here is that the more inconsistent the method of operation is for a user interface, the more difficult it is to use. By example, Option 1 will always be difficult to use, no matter how the controls are labeled, and no matter what feedback is provided (tones, speech, vibration, etc.), or how much training the users are given. The phrase to remember, which may not translate well, is that *a dead fish with wheels doesn't move any faster than a dead fish without them*.

The representation shown can be useful for those instances where consistent methods are used and there are a variety of objects and actions. However, for automation, the goal is to change from 1 state (or mode) to another, and in that case, an alternative representation can be useful to highlight inconsistencies. In some sense, setting tasks can be thought of as state changes as well, but representing them in the form described in the section that follows is unwieldy.

Upper and Lower Triangular Matrix Representation

Suppose for the sake of discussion, a vehicle had the automation modes in Table 6-4.

Table 6-4. Example Modes

Mode	Abbreviation	Function
Manual	M	no automation
Adaptive Cruise Control	ACC	adjusts speed based on lead vehicle gap, does not detect other vehicles
AutoSteer + ACC	AS + ACC	automatically steers
Enhanced driving mode	AS + ACC+	also adjusts gaps based on vehicles in adjacent lanes

How could transition between modes be represented? Table 6-5, partially complete, provides an option. In this example, no matter what mode what was is in, stepping on the brake puts the vehicle in the manual model. Thus, the rule the subject needs to learn is simple.

If (the brake is pressed)
  Then (go to manual mode)
However, suppose in the AS + ACC+ mode, pressing the brake cause the vehicle were to revert to the AS + ACC mode, then the rule would be more complicated.

If (the brake is pressed) and (the vehicle is not in AS + ACC+ mode)
  Then (go to manual mode)
  Else (go the AS + ACC mode)

Table 6-5. Transition Options

Option 1 – Braking Always Transitions to Manual Mode

Initial state	M	brake			
	ACC	brake			
	AS + ACC	brake			
	AS + ACC+	brake			
		M	ACC	AS + ACC	AS + ACC+
		Final state			

Option 2 – Braking Usually Transitions to Manual Mode

Initial state	M	brake			
	ACC	brake			
	AS + ACC	brake			
	AS + ACC+			brake	
		M	ACC	AS + ACC	AS + ACC+
		Final state			

Keep in mind that this representation is only for the brake, and that one needs to consider what happens when the steering wheel is operated to override the automation. Furthermore, the lower triangular half represents takeover, whereas the upper half represents transitions to greater automation, which often involves the actions of other controls.

Two things need further thought. First, how can the operational design domain (ODD) under which transitions can occur be represented in this table? The ODD is most critical for handover to automation, where preconditions about road types, speeds, weather, etc. need to be met. They make the interface complicated, and showing them is important. At this point, how to do that in a manner that simply but clearly represents the interface is to be determined.

Second, what values can be computed from this matrix to represent its complexity? One simple approach would be to count the number of rules and some measure of rule complexity (as described in the software section) to describe the matrix. Both that value and the matrix are useful. The value provides a quantitative assessment of the complexity of an interface that could be a design goal. The matrix visually represents the interface complexity in a compact form that those unfamiliar with production rules and human decision making can understand. Keep in

mind that this simplification does not consider prior human knowledge of the operating rules, which can make decision-making easier or more complex, depending upon if the known rules are in harmony or conflict with the intended principles of operation of the interface.

To reinforce what was said earlier, a huge advantage of these representations is that at the outset they provide an initial estimate of the complexity of the system before the HMI is implemented. If there was a proposal for an automation architecture that had many modes and their operation was highly dependent upon the ODD, one sensible decision could be to not offer them until the ODD was less constrained or to eliminate less useful modes, all with the goal of making those remaining less complex and easier to operate.

Thus, this sections leads to the following guideline.

Guideline 6-1: For setting tasks and for mode change tasks, create matrices or some other visual representation that shows the method to accomplish each task and highlights inconsistencies in methods. The representations for setting and mode change tasks can be different.

How Can the Complexity of System Logic Be Determined from Those Representations?

McCabe Cyclomatic Complexity

The issue of software complexity was a major research topic from the last 1970's until about 1990, although research continues on that topic today. Of interest was making software easier to understand, more reliable, and easier to maintain. There is a strong parallel between accomplishing those goals and making an interface easy to use, as is described later where production rules are discussed. Thus, if the logic a person needs to use to make decisions can be expressed using the same language used by software, then the measures of complexity for software should also be appropriate for human decision making.

One of the first papers on this topic is McCabe (1976), a paper cited more than 6000 times. Using ideas from graph theory, he proposed the Cyclomatic Number as being an indicator of software complexity. The Cyclomatic Number is the number of possible paths between the beginning and end point of a computer program that returns to the origin. That value is expressed as $V(G) = e - n + p$, where graph G has e edges (paths connecting node pairs), n nodes (vertices), and p connected components (the number of nodes that have exit points). The Cyclomatic Complexity, a similar value and most commonly used, is defined as $e - n + 2p$. Example calculations are shown in Figure 6-9. Numerous other examples are presented in his paper, with the target application being FORTRAN.

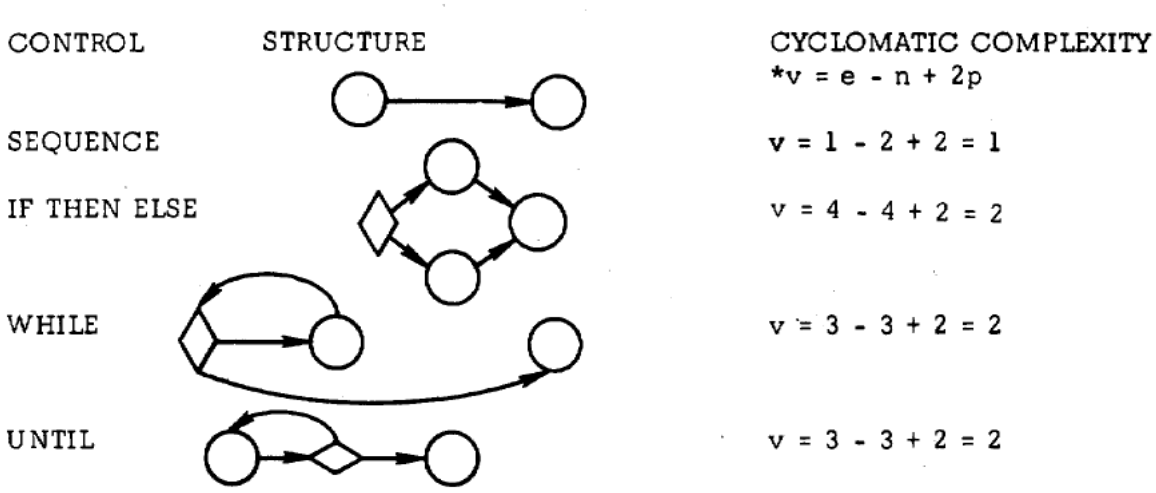


Figure 6-9. Examples of Cyclomatic Complexity
McCabe (1976), p. 308

Note that the simplest programming structure, one of unit complexity, has 1 input, 1 output, and 1 path between them. In programming, one way to simplify code is to make use of subroutines, which usually have unit complexity. By analogy, if one has a variety of tasks, but they all require the same method and that method is well learned, then in some sense that method is like a subroutine, and as a consequence, the complexity of the method is reduced.

Note that McCabe (1976) does not provide an empiric validation for his proposal, only a theory, but his book does (McCabe (1977). However, others have, and the cyclomatic complexity is one of many statistics used to assess software complexity. For additional information on this topic, see McCabe (1977), Shepperd (1988), and McCabe and Butler (1989).

Halstead Lines of Code and Other Measures

In Halstead's classic textbook (Halstead, 1977), he introduced a number of statistics pertaining to software complexity. He thought of software from an atomic perspective, considering the number of distinct tokens (n1), the number of distinct operands (n2), the total number of operators (N1) and the total number of operands (N2). Operators include (), int, +, print, and so forth. Operands are the objects manipulated, the terms after an equals sign or parameters of a loop. Based on these basic measures, e proposed "*software science metrics"* including:

Proposed vocabulary $n = n_1 + n_2$
Program length = $N = N_1 + N_2$
Calculated program length $N_h = n_1 \log_2 n_1 + n_2 \log_2 n_2$
Volume = $N * \log_2 n$
Difficulty = $D = n_1/2 * N_2/ n_2$
Effort = D x V

At least 1 study suggests that many of the McCabe and Halstead statistics are highly correlated (Henry, Kafura, and Harris, 1981).

In fact, quite important and often cited more than these measures is the number of lines of code, which includes declaration, definitions, directives, and code. (https://www.verifysoft.com/en_halstead_metrics.html, retrieved December 12, 2018). Related measures include the total number of physical lines, the number of blank lines, and the number of comment lines. See https://en.wikipedia.org/wiki/Halstead_complexity_measures. For summaries of this topic, see Zuse (1991), Chhillar and Gahlot (2017)

Chidamber and Kemerer Object Oriented Metrics and Others

Much of the initial research on program complexity was concerned with FORTRAN, the predominant language at the time. However, object oriented languages have some fundamentally different organizational logic, and hence, it makes sense that the complexity could be determined in a different manner (Table 6-6). It is uncertain how these metrics relate to human decision-making, and but they are included for the sake of completeness. In addition to proposing these measures, Chidamber and Kemerer provide some validation data for their use from C++ and Smalltalk. See Chidamber and Kemerer (1994) for details. See also Lake and Cook (1994) for additional information on this topic.

Table 6-6. Metrics from Chamber and Kemerer

Metric	Definition
Weighted Methods Per Class (WMC)	sum of the complexities of all methods of a class.
Depth of Inheritance Tree (DIT)	how many ancestor classes can potentially affect this class
Number of Children (NOC)	number of immediate subclasses subordinated to a class in the class hierarchy
Coupling between object classes (CBO)	a count of the number of other classes to which it is coupled.
Response For a Class (RFC)	set of methods that can potentially be executed in response to a message received by an object of that class
Lack of Cohesion in Methods (LCOM)	How often instances cannot be executed by methods

Also mentioned with some regularity are the MOOD's object oriented metrics (Harrison, Counsell and Nithi, 1998), Component-Oriented Metrics (Haines, Carney, and Foreman, 1997; Aloysius and Maheswaran, 2015; Tobias, Mwani, and Michael, 2015), and Aspect-Based Metrics (Guyomarc'h and Guéhéneuc, 2005). Again, at this point, they are worth noting, but unknown in how useful they are in characterizing human decision-making.

Thus, many of these measures are potentially useful for examining the complexity of decision making with automated driving systems. Note that the limited data available suggests that traditional measures of complexity (Halsteads's E, McCabe's Cyclomatic Complexity, lines of code were correlated with programmer performance in code maintenance tasks, with those measures being more predictive for less experience programmers (Curtis, Sheppard, Milliman, Borst, and Love, 1979). In one study examining multiple measures cited here, with 1 exception,

the correlations of these measures with ratings were 0.6 to 0.8 except for simple readability (Posnett, Hindle, and Devanbu, 2011), which was much less. Ratings were developed from pairwise comparisons of various method in which one or the other was rated as more complex, or they were rated as equal. The number of lines of code was in the middle of the range of statistics. Readers should see Katzmarski and Koschke (2012) for details.

As a footnote, for assessment of the driving automation software, Olsson, whose research was mentioned earlier, uses cyclomatic complexity (E-N+2P) and the Maintainability Index (Oman and Hagemeister, 1992; Coleman, Ash, Lowther, and Oman, 1994).

Maintainability Index = MI = $171 - 5.2\ln(HV) - 0.32CC - 16.2\ln(LOC) + 50 \sin(2.46COM)^{0.5}$.

where:

HV = Halstead Volume=

total # of operators and operands x $\log_2$(total # distinct operators and operands)

CC = cyclomatic complexity

LOC = # of Lines Of Code

COM = % of COMments in code

Those seeking additional information on this topic should see Weyuker (1988), Zuse (1991), Shao and Wang (2003), Yu and Zhou (2010), and Chhillar and Gahlot (2017).

In selecting measures and statistics for further evaluation, it is important that they be relevant to human decision-making, well correlated with the performance statistics of interest, readily implemented, and well understood, among other characteristics. Table 6-7 provides some insights into these characteristics, but for a situation some different from that here.

Table 6-7. Comparison of Some Metrics

Metric	Usability	Effect	Popularity	Theory Basis
Lines of code (LOC)	easy	good	wide	no
Halstead Complexity Metric (HCM)	medium	good	wide	no
Weighted Halstead Complexity Metric (WHCM)	medium	bad	narrow	no
Data Stream and Control Stream Model (DCM)	hard	bad	narrow	no
McCabe Cyclomatic Complexity Metric (CCM)	medium	good	wide	yes
Pseudo Path Metric Model (PPMM)	medium	bad	narrow	no

Pseudo Path Metric Model resembles the Cyclomatic Complexity Metric but paths are weighted
Weighted Halstead Complexity Metric gives different operators different weights
Data Stream and Control Stream Model combines data flow and control into a single measure of complexity.
Source: Yu and Zhou (2010), p. 355.

What Make Makes System Logic Complex for People? - Cognitive Complexity Theory

Cognitive complexity is a topic of continuing interest, with the focus here being on the research of Dave Kieras, although many others have contributed to understanding this topic. One of the early studies to consider task complexity was Kieras and Polson (1985). They noted that "the complexity of a device, from the point of view of a user, depends on the amount, content, and structure of the knowledge required to operate the device successfully" (p. 366). Furthermore, they note that the complexity of a device depends upon:

(1) "The complexity of the user's task representation, and the learning, memory, and processing capacity demands implied by the task representation;
(2) The number of device-dependent functions, which are not part of the user's initial task representation, and the difficulty of learning them;
(3) The ease with which a user can acquire how-it-works knowledge" (p. 367).

They then describe how users performing a task can be described as a set of if-then-else rules and how such rules function, referring to finite state machines, recognize-act cycles, the GOMS model, the idea of subroutines, and other concepts from computing. They do not propose a single expression to compute complexity, but propose that it depends upon the number of productions to be learned, then number of productions triggered to accomplish a task, the number of keystrokes required, the number of cycles to complete a task, the mean and maximum number of items in memory, as well as the total number of items required to be remembered to complete a task.

Thus, the essence of this approach is that one can get a sense of the complexity of a task by describing it using production rules, and then determining the statistics just listed. For an example of such, see SAE J2365. Note, the author is in the process of completing at literature review to support the eventual updating of SAE J2365 to support contemporary interfaces.

There are several languages that could be used for this purpose (e.g., ACT-R, EPIC, Soar) but they are not easy to learn. As a practical matter, the author would suggest considering NGOMSL (Kieras, 1968) as it is reasonably well documented, simpler, and easier to learn, though there are likely to be viable alternatives. Keep in mind that the purpose here is not to examine the nuances of human performance, but to rapidly create a model of what people are doing and identify weaknesses in the interface architecture and the HMI.

The key elements of NGOMSL are a structure for each method used to complete each task, flow control operators, a set of operators for human actions, and rules for the operation of memory. Set Table 6-8.

Table 6-8. Elements of NGOMSL

Element	Examples
Method structure	Method for goal: <goal description> Step 1. <operator> ... Step 2. <operator> ... ... Step n. Return with goal accomplished.
Flow control operators	Accomplish goal: <goal description> (equivalent to CALL statement) Return with goal accomplished (equivalent to RETURN statement) Decide: If <operator...> Then <operator> Goto Step <number>
Operators for human actions	Home hand to mouse (H) Press <key name> (K) Type a string of characters <string> (T) Press or release mouse button (B) Click mouse button (BB) Type <string of characters> (T(n)) Move cursor to <target coordinates> or Point to <target coordinates> (P) Locate object on screen <object description> (M) Verify that <description> (M) Wait for <description> (W(t))
Rules for memory operation	Recall that <WM-object-description> Retain that <WM-object-description> Forget that <WM-object-description> Retrieve-from-LTM that <LTM-object-description>

Several studies have followed on this approach. In Polson and Kieras (1985), 2 studies of learning to use a computer editor are described. The research centers around general production rules of the form:

```
(label
        (IF (AND (TEST-GOAL description of goal)
                 (TEST-NOTE description of note)
                 (TEST-SCREEN description information on screen)
                 (TEST-MSS description of editing task)
                 (NOT (condition)) )
        THEN
                 ((ADD-GOAL description of goal)
                 (DELETE-GOAL description of goal)
                 (ADD-NOTE description of note)
                 (DELETE-NOTE description of note)
                 (DO-KEYSTROKE depress single key or control)
                 (DO-TYPE-IN enter string stored in working memory)
                 (LOOK-MSS find string in manuscript)) )
```

Goals are what the user intends to do. Note refers to information in working memory.

In the first experiment, which 15 subjects learned various editing features. A regression equation to predict the time to learn each feature that considered the time per production rule (0.54 minutes) and an intercept of 9.02 minutes accounted for 83% of the variance. The time to learn a new production rule was about 30 seconds. In a second experiment concerning editing, involving 8 subjects, the equation that follows to predict editing time accounted for 72% of the variance.

Editing time (s) = 0.17 (# recognize act cycles) + 0.56 (# goals and notes in working memory).

Adding in the number of keystrokes did not improve the model because that value is highly correlated with the number of recognize act cycles.

For those interested in using NGOMSL, Bovair, Kieras, and Polson (1990) provide a very useful description of the notation related to production rules. They note that in assessing what is to be learned, a proposed new rule can be (1) identical to an existing one, in which no learning is needed, (2) a trivial generalization, in which almost no learning is needed, (3) subsumed under a previous generalization, in which almost no learning is needed, or (4) completely new, in which a rule is added to long-term memory.

To gain a further understanding of the relationship between production rules and learning, they had 90 subjects learn to use an editor with various features. In the first experiment, they found the equation that follows to account for 88% of the variance. Note that training time includes study time and practice time.

Training time (s) = 683.2 + 34.7 (# of new rules)

A second experiment examined other characteristics as well in greater detail, but many were editing-specific (# times manuscript to edited was examined) and not covered here for the sake of brevity.

In another study, Vossen, Evans, and Ziegler (1988) had 62 people use an editor with graphical capabilities. Their basic model, which accounted for 80% of the variance in learning times was

Learning time (s) = 214.3 + 12.1(# of new production rules)

However, some 88% of the variance was accounted for by:

Learning time (s) = -477. + 12.7 (# of new production rules) + 0.4 (# of recognize act cycles).

Actual training time and the constants in these regression equations can be context specific. However, these studies convincingly show that learning time, a measure related to ease of use, is a function primarily of the # of new production rules to be learned, and secondarily, the # of recognize-act cycles, which in keystroke intensive tasks, is close to the # of keystrokes.

How Can the Complexity of the Representation Be Determined?

This is a topic for a future update of this report and is what people traditionally think of as HMI design. For information on this topic, see Hugo and Gertman (2013), Yoon, Lim, and Ji (2015), and Jang, Kim, Jung, and Kim (2016).

Synthesis of This Section

This section describes several ways to visually represent how an interface works, in particular what a user needs to do to change modes or set parameters. Of the ways to represent those task methods described in this report -- finite state machines, hierarchical state machines, behavior trees – it is uncertain which of them is most effective in describing how a system functions to engineers familiar with those alternatives and those not so familiar. Again, this is important because if the system logic is extremely complex, no HMI, no matter how well constructed, will lead to an interface that is easy to use.

Thus, a major thrust of this section has been on determined how software complexity is assessed and what characteristics of a task method influence how easy it is to learn. Central to this line of thinking is describing the task using production rules, if-then-else and other statements that describe what the user needs to do to accomplish a task. The advantage of this approach is that the human logic and software logic are described in a similar manner, leading to a greater understand of the design implications by all of those involved in the design. The measures and statistics that are appropriate to assess interface complexity come from software science (e.g., number lines of code, number of operator, operands, number of nodes and paths, etc.) as well as from studies of human learning (the total number of rules, the number of new rules).

At this point, there are a number of items that need to be explored as suggested by the preceding paragraphs in this section. However, it is apparent that computational model based on production rules and some of the measures and statistics just mention should lead methods to compute the complexity of an interface. These methods will be useful for both making architectural design decisions about features to include and requirements for the operational design domain, as well as specific HMI implementations. Accordingly, the following guidelines are recommended.

Guideline 6-2: For each task associated with the automation, write out the production rules to accomplish them.

Guideline 6-3: Compute the complexity of each task and each collection of tasks. The formulas for those computations need to be determined. Candidate independent variables include Cyclomatic Complexity, lines of code, the number of production rules, the number of operators and operands, and other statistics.

Further development of these ideas is an appropriate topic for a follow-on project.

Q7: WHAT ARE SOME GUIDELINES THAT CAN REDUCE INTERFACE COMPLEXITY?

Complexity is a function of the number of rules about how the system operates that the driver needs to learn and how many then need to execute to complete a task. Compliance with the guidelines that follow can be determined by applications of the guidelines from the previous section.

Guideline	Explanation
7-1: Minimize the number of modes.	The more modes there are, the more likely users will be confused as to what the current mode is. From the marketing perspective, there is pressure to provide more modes because it increases the apparent feature count. One of the more confusing aspects for subjects are the distinctions between off, enabled, and on. One way to potentially reduce this confusion is to always never try have an off state, but to have off as no setting. If a vehicle has adaptive cruise control, it is highly unlikely that cruise control would be used without the adaptive feature (ACC). Accordingly, for ACC-equipped vehicles, manual cruise control should not be available.
7-2: Make the transitions between modes consistent.	For example, if stepping the brake causes a transition from one automation mode to manual mode, then that should occur of all automation modes. An ETIT or some other transition diagram can make the inconsistencies apparent.
7-2: Make the transitions between modes consistent.	For example, if stepping the brake causes a transition from one automation mode to manual mode, then that should occur of all automation modes. An ETIT or some other transition diagram can make the inconsistencies apparent. Note: There will be times when this guideline conflicts with what experience suggest drivers should do. In that instance, the challenge is to decide which guideline has precedence, both short term and long term.
7-3: Make tasks that have negative consequences not easy to do.	For the purposes of this project, that task could be turning off an automatic emergency braking system which has proven to be highly reliable in all of the conditions experienced by a particular driver, and when triggered in that past, avoided crashes that driver would have otherwise experienced.

7-4: Minimize the number of steps to set a mode.	Pressing a switch to turn cruise control on would be preferred to pressing a switch to powering all automation and then a second switch for cruise control. At this point, there is no definition of what constitutes a step. However, the time to complete a task is a useful surrogate measure. Task completion time can be determined using SAE J2365. SAE J2365 is in the process of being updated by the author, as in its current form, operators for many touch screen gestures (e.g., swipe) are not included. In some circumstances, guidelines 7-3 and 7-4 are in conflict.
7-5: Minimize the number of steps to set the automation.	Some vehicles have complex menu systems that require many steps to select a setting. How many steps are acceptable will depend upon the frequency of changing settings and the importance of the parameter set.

Q8: WHAT ARE TASKS THAT PEOPLE DO OR DO COULD DO WITH AUTOMATION INTERFACES?

This document provides a different perspective to guidelines than is typically the case, and reflects ideas from a series of reports funded by Transport Canada (Cassano-Piche and Green, 2017; Green, 2017; Green and Rudin-Brown, 2017; Rudin-Brown and Green, 2017; Smiley, Smahel, and Green, 2017; Green and Rudin-Brown, 2018a, b) to develop a method that can be used by experts to evaluate an HMI. The literature reviews conducted as a part of that project emphasized traditional HMI checklists and other assessment methods most appropriate for information systems. For the automated systems of interest, the assessment methods used in human-robot and human-automation systems are closer to what is needed. As these human-automation systems become more complex, the attributes of a traditional HMI can become more prominent, in part because menu depth and breadth have increased to support increased interface complexity.

Furthermore, a review of contemporary automation interfaces reveals that a significant problem with many HMIs is they lead to a lack of situation awareness, which reports have called "why did my Tesla do that?"

Given those differences and the recent research, these guidelines take a task-focused approach. Thus, the essential question is "how well does the HMI support what the driver wants to do?" There are 10 basic tasks (Table 8-1) a driver must do with every automation system, though in some systems they can be grouped together so that the number of tasks may appear to be less. These tasks are revised from those in Green and Rudin-Brown (2018a, b) (tasks 8, and 9, 10) and task 5 was added. In some systems, several of these tasks could be almost identical, for example turn on/off and set/change mode. Explanations of these tasks follow.

Table 8-1. Tasks of Interest

User Task or Task Category	Control System	Warning System
1. Determine operational domain	x	x
2. Turn it on/off	x	x
3. Determine mode	x	
4. Set/change mode	x	
5. Temporarily override	x	
6. Determine current settings	x	x
7. Set/change settings	x	x
8. Monitor for proper operation	x	x
9. Scheduled takeover	x	
10. Unscheduled takeover	x	x
Learn how to use from documentation (manual, YouTube videos, etc.)	x	x

1. *Determine the operational domain (ODD)*

The term operational domain, defined in SAE J3016 concerns the conditions under which a system is intended to operate – the types of roads, the traffic, the weather conditions, and so forth. Typically, this is only communicated in the owner's manual (for which there are reasons why people do not read them (Mehlenbacher, Wogalter, and Laughery, 2002; Novick and Ward, 2006)) or in videos, for example, on YouTube.

Guideline 8-1: ***The guiding principle is to design for reasonable and expected use and misuse.*** Often the term foreseeable misuse is used. (Thus, even if cautioned against in a manual, drivers will try to use automation under a wider range of conditions, and if the automation works, then they will use it under the those wider conditions. Again, most HMIs do not communicate their ODD, which encourages use other than intended. There are differences in product liability law between nations. However, most auto manufacturers intend to eventually sell all of their products globally, so a global perspective is needed even if that is not the initial market. Should further guidance be needed, contact a lawyer with expertise in product liability.

Key questions to address:

a. Can the driver determine the conditions under which the system will and will not operate (and operate properly) from the HMI alone? Conditions of interest include speeds, traffic, road types, weather conditions, and others.

   Guideline 8-2: The driver should be able to determine the ODD without looking away from the road scene. This is not easy to do. It can be accomplished using a Head-Up Display (HUD) or augmented reality.

b. Does the documentation distinguish between those conditions that are most likely and least likely to affect operation?

c. In the documentation, is the explanation of how the system works complete but also brief? People are not willing to read manuals (Mayer, 2002).

2. *Turn it on/off*

This could be treated as 2 tasks, but they are bundled together. In some cases, the on and off tasks are basically the same except for a last step, for example pushing an on or off button. In contrast for some systems, such as adaptive cruise control (ACC), there can be methods to turn a system off (stepping on the brake) for which there are not direct parallels for the opposite state.

Key questions to address:

a. Does the HMI make it apparent which systems default to be on and which default to be off?

   Guideline 8-3: The driver must be able to turn automation off without looking away from the road scene because sometimes it needs to be done quickly under challenging circumstances. Generally, this is accomplished by stepping on the brake and/or grabbing the steering wheel.

3. *Determine mode*

The mode is the state of an HMI, such as off, enabled, and on. In the enabled state, the system is powered but not exerting any control. This state is often used to alter the operational setting of a system. Mode errors are a major problem with automated systems (Levenson and Palmer, 1997; Sarter, Woods, and Billings, 1997). The inability to easily determine the mode (and know what it means) is a major weakness of many automated driving systems.

Key questions to address:

a. Does the HMI make it apparent at all times which mode options exist and the current mode for each system?
   Guideline 8-4: The driver must be able to determine the automation mode without looking away from the road scene because sometimes the automation exerts control (or fails to exert control) in unexpected ways that could require immediate action from the driver.
b. Does the user understand what the modes are? People can find it difficult to distinguish enabled from on and off from enabled modes.
c. Is the mode indication provided in a manner that it can be determined instantly, for example by looking at a HUD?

4. Set/change mode

Depending how a system is implemented, that task of determining the mode and setting/changing it could be identical. So, for example, if the only method to determine the mode of an automation system was to examine a menu on the center console, the set/change mode and determine model tasks would be essentially identical. That is not the case if there is a separate mode display (often an icon).

Key questions to address:

a. Is the mode easy to change?
   Guideline 8-5: As with determining the mode, drivers must be able to disengage automation instantly for the same reason. Often this can be done by stepping on the brake or steering.

5. Temporarily override

This turns out to be a fairly common and sometimes forgotten task. Typically, it involves driver operating the steering wheel or accelerator pedal, for example, pressing the accelerator to override the cruise control system and drive faster. Usually, ending the interruption causes the system to return to its prior state. Generally, brake pedal operation shuts the system off. For many systems, there is not salient indication that either the system is being overridden or that it still is on.

Key questions to address:

a. How does the driver know the system is being overridden?
   Guideline 8-6: The driver must be able to determine the override is occurring without looking away from the road scene. One way to accomplish this is to provide a HUD.

6. Determine current settings

The settings are the values by which a system operates in 1 or more modes. Settings include the following distance for ACC or the speed at which it is to drive.

Key questions to address:

a. Are major settings visible to the driver at all times?
   Guideline 8-7: The driver must be able to determine the current settings without looking away from the road scene.

7. Set/change settings

In much the same manner that determining the mode and setting/changing mode can be very similar if not identical tasks, the same is true for determining settings, and setting/changing them.

Key questions to address:

a. Are those settings that are likely to be changed often while driving able to be changed in 1 or 2 steps?

   Guideline 8-8: Which settings need to be able to be changed without looking away from the road will depend up the frequency with which a setting is changed and the extent of its connection with crash risk.

b. Do drivers understand what those settings represent?

8 Monitor

This task involves observing the operation of the automation. In some instances, there may be no overt action from the driver. For example, for an intervention system (automatic emergency braking), the driver observes its operation to determine if that system is within the desired bounds, in particular, that the vehicle is being operated safely and there are no threats the driver needs to address. Some partially automated vehicles do not support maintaining situation awareness to help the driver monitor the vehicle very well (Green and Rudin-Brown, 2018b).

Key questions to address:

a. Does the HMI show the lane markings, surrounding traffic, and other information it uses to make decisions in a manner that the driver does not need to look away from the road scene to know what the vehicle knows?

   Note: Depending on the system, other information the vehicle could know is speed limit or other signs, information about curves or exits, the existence of pedestrians or bicyclists, and so forth.

b. Does the system communicate its confidence in those items?

   Note: This is a major problem with many current systems. For example, a vehicle may be showing 1 path as the road and then suddenly change. Confidence data is a leading indicator of when a vehicle decision could change.

   Guideline 8-9: The driver must be able to determine what the vehicle knows and how confident the vehicle is in what it knows without looking away from the road scene.

   In addition to visual displays to indicate who is control, concepts involving moving the foot pedals and steering wheel (or other steering devices) out of the way have been explored (e.g., Mok, Johns, Yang, and Ju, 2017a, b). How who is in control and when the driver needs to take over (an AH transfer) is an important topic that needs further investigation.

9 Scheduled or Routine Takeover

In this case, the driver takes over either because (1) it is time for them to do so (for example, they approach a road class in which the automation does not function well), (2) the driver simply wants to be engaged, or (3) for other reasons. Most vehicles do not indicate when scheduled takeovers are to occur because the HMI is not strict in enforcing the intended ODD. This will change as drivers use automation in situations for which use is not intended.

Key questions to address:

a. When takeover could be scheduled, such as leaving a limited access road to a road class that is not supported by automation, how will the HMI provide enough notice

for the driver to take over safely in almost every situation?
Guideline 8-10: When a takeover (AH transfer) is scheduled, the driver shall be informed how much time they have to take over. This is true for unscheduled takeovers as well.
Note: At this point in time, what the driver is doing prior to taking over is to be determined. In some situations, they could be asleep.
Note: One likely scenario is a system that provides automation support on a expressway but not off the expressway. In that case, the driver should be told the time before which they need to take over (an AH Transfer), in seconds if it less than a minute, and minutes and seconds, if it is more than a minute.

b. For much more information on scheduled takeovers, see the special interface elements section.

10 Unscheduled Takeover

For this task, something has occurred where the driver needs to resume control, usually immediately. However, the reverse could also be true, where the automation takes control, such as for automatic emergency braking. Usually, the onset of the unscheduled takeover is immediate. Of all of the topics on automated vehicles, the unscheduled takeover situation has been the subject of more studies than any other. See Trosterer, Meschtscherjakov, Mirnig, Lupp, Gartner, McGee, McCall, Tscheligi, and Engel (2017) for a review.

Key questions to address:

a. As with scheduled takeover, what is the state of the driver before the take over (an AH transfer)?
Guideline 8-11: If a takeover (AH transfer) is to occur, and the driver is not capable of taking over, then the driver should be informed of such. What should occur next has yet to be determined. Some would argue that the driver is ultimately responsible and they should never be overruled by automation. Others would argue that driving is a public health problem, and that overrule should be allowed.
b. What is the demand, the workload of the driving situation?
c. Given a and b, how much time is required for the take over?
Note: The time required is not the mean, but some time on the tail of the distribution. What percentile that should be (90^{th}, 95^{th}, 98^{th}, etc.) is to be determined, as well as what distribution from which that percentile should be drawn.
For more information on unscheduled takeovers, see the special interface elements section.

In developing an HMI, it is important to consider how it should be designed to support each of these 10 tasks. In fact, if one were to score an interface, that score could consider the ease with which each task is performed. In developing a composite score across tasks, keep in mind that all tasks may not be equally important, and that for some tasks, a minimum level of performance may be required.

Q9: WHAT ARE SOME TRADITIONAL HMI GUIDELINES THAT CONCERN INFORMATION PRESENTATION?

This report needs to be viewed in the context of many years of research and development that has led to numerous human factors guidelines. A key point to be made is that most of the guidelines focus on traditional information-oriented HMIs, not on automation. However, the traditional guidelines contain considerable information that is useful, pertaining to menus, display and control design, text messages and so forth. Unfortunately, those guidelines do not address some of the central issues here, which is situation awareness and shared control, in particular, determining who is in control at any moment, the driver or the automation.

In addition, it is likely that people working on this project will later apply what was learned here to other applications when they change groups as part of their professional development. For those individuals, providing a more general context is important. Accordingly, those guidelines are identified and should be useful in supporting HMI design in a variety of contexts, with the focus on human-automation interaction. In the sections that follow, the guidelines have been grouped into 2 categories, general HMI guidelines (many of which are not automotive) and guidelines and standards specific to automotive applications. For a broader perspective of standards, readers may find Green (2018a, b) to be useful.

General HMI Guidelines

There are 9 primary collections of human factors/ergonomics standards (Table 9-1) that provide broad coverage of the topic.

Table 9-1. Primary Human Factors / Ergonomic Standard Collections

Collection	Description
Military Standard 1472G (U.S. Department of Defense, 2012).	The human factors "Bible." Although developed for military use, this standard is very broad in its coverage, especially for topics such as control and display design, and anthropometry. Topics such as control size and spacing are well covered in 1472G. Following the recommendations in this document is highly recommended, though users should realize that in many cases the requirements are minima, not designed design limits.

ISO Technical Committee 159 (Ergonomics) standards	ISO TC 159 consists of 4 subcommittees that collectively have published 132 some of which are relevant to automotive HMIs, in particular the 68 of subcommittee 4 (Ergonomics of Human-System Interaction). Examples include ISO 1503 (direction of movement), ISO TR 16982 (usability methods), 18152 (process for the assessment of human-system issues), 20282-2 (summative test method), and particular, ISO 9241 with its almost innumerable parts concerning the selection of input devices, tactile and haptic interactions, visual display requirements, and many, many other topics. Coverage of just that standard alone is well beyond the project scope.
ISO JTC1 / SC7 and SC 35, especially the ISO 2506x series	ISO/IEC JTC 1 has 2 relevant subcommittees, Software and Systems Engineering) and SC 35 (User Interfaces). The standards of SC 35 are more design-oriented (e.g., dialog interaction – ISO/IEC 10741) whereas those of SC 7 are more process oriented (e.g., ISO 25062 – Common Industry Format for Usability Test Reports).
AAMI standard HE 75 (American Association of Medical Instrumentation, 2009) and IEC 62366	These standards are mentioned primarily for the sake of completeness. Some topics are specific to medical devices such as hand tool design, packaging, connectors, and home health care. The coverage of human error is strong and useful. See also AAMI (2015).
NUREG 0700 (U.S. Nuclear Regulatory Commission, 2002)	The focus of this document is on process control. However, coverage of some topics, such as display elements, could be useful.
ANSI/HFES standard 100 (Human Factors and Ergonomic Society (2007)	This standard is primarily concerned with office workstations. In addition, there are other standards (200, 300) that either have been or will be phased out.
NASA Handbook National Aeronautics and Space Administration (2010).	Its emphasis is on space flight, so topics such as habitability and hatches are given emphasis, but some of the anthropometric data could be useful to automotive HMI design.
FAA HF Standard-100 (Federal Aviation Administration, 2012)	This standard is particular useful when issues relating to design for ease of maintenance arise.
OSHA standards (U.S. Department of Labor)	This collection of standards was written to deal with workplace hazards – ladders and scaffolds, nonionizing radiation, compressed gases, personal protective equipment, confined spaces, machinery guarding, and chemical hazards (benzene, coke oven emissions). It is mentioned because those now involved in HMI design could have the design of production facilities as a future assignment.

Automotive Specific Guidelines

More closely related to the topic of interest are automotive specific guidelines, with the first set being developed by the author of this document in 1993. (See Table 9-2.) Since then, there has

been a succession of guidelines and set of principles. The UMTRI, HARDIE, Battelle, and Campbell (aka NHTSA, new Battelle) guidelines are, as the table suggests, truly detailed sets of guidelines, anywhere from 100-500 pages long. The remaining documents, are all sets of principles, though the TRL guidelines have more explanation than the others listed. The CAMP principles are different from other guidelines listed in that they are specifically intended to assess automation. The other documents in this table are intended for traditional HMIs. The author would suggest that human factors practitioners developing HMIs should at some time at least skim all of these guidelines. If pressed for time, focus on the EU Statement of Principles, the CAMP 2016 Principles, and the Campbell Guidelines.

Table 9-2. Current General Automotive HMI guidelines

Document	Comments
UMTRI Guidelines (Green, Levison, Paelke, and Serafin, 1993).	The first set of comprehensive design guidelines, including principles, general guidelines, and specific design criteria, with an emphasis on navigation interfaces. (111 p).
HARDIE Guidelines (Ross, Midtland, Fuchs, Pauzie, Engert, Duncan, Vaughan, Vernet, Peters, Burnett, and May, 1996).	Early set of European guidelines containing less data than UMTRI or Battelle. (562 p)
Battelle Guidelines Campbell, Carney, and Kantowitz, (1997).	A voluminous document with references to interface design which is heavy on trucks. The user interface presented has been said to have a Windows flavor. The document includes physical ergonomics information (e.g., legibility, control sizes) which are not included in the UMTRI guidelines. (261 p)
TRL Guidelines (2004) Stevens, Quimby, Board, Kersloot and Burns (2004).	The guidelines in this document are intended to alert designers of IVIS to some legal and ergonomic issues relevant to safety. (56 p)
AMA Guidelines JAMA (Japan Automobile Manufacturers Association) (2004)	The first set of detailed design guidelines for interfaces from Japan. These guidelines are voluntary in Japan but followed by all OEMs and sometimes by aftermarket suppliers. Some aspects are particular to Japan. (5 p)
AAM Statement of Principles Alliance of Automobile Manufacturers (2006).	These are voluntary industry guidelines and best practices for telematics devices including cell phones, navigation systems, and internet links. These were developed by the auto industry in private with no outside input. (90 p)
EU Statement of Principles (ESOP) Commission of the European Communities (1999, 2007).	Mostly motherhood statements, but there have been efforts to develop assessment methods for these principles. (42 p)

Crash Avoidance Metrics Partnership (CAMP) 2016 Safety Principles (Tellis et al., 2016)	This most recent set of principles, and in contrast to the prior sets, are specifically intended for automation systems. They focus on how to identify hazards, not specific HMI characteristics. The approach is based on the STPA (System Theoretic Process Analysis) method derived from STAMP (System Theoretic Accident Model and Processes).
NHTSA/new Battelle (2016) Guidelines Campbell, et al. (2016) (See also Jerome, Monk, and Campbell, 2015.)	This is the most recent set of HMI guidelines. Topics covered include principles of safety messages, visual, auditory, and haptic interfaces, input devices, and other topics. The document is well done and well referenced, but it does not make as many numeric recommendations as the literature permits. The document is formatted consistently, with each facing 2 pages concerning a very specific topic, emphasizing the design goal, the relevant design principles, the supporting research, and a list of references.

Of the many guidelines listed, Military Standard 1472 is the most widely accepted. Accordingly, the following guideline is suggested**.**

Guideline 9-1: The design of the interface shall comply with the latest version of Military Standard 1472. There a several other standards listed in this section or referred to by reference, for which compliance could be required. Further investigation is appropriate.

Automotive HMI Checklists

To assess compliance with good human factors practice, a number of checklists have been developed (Table 9-3). For the most part, they are intended to be applied by experts. The difficulty with these lists is they require a fair amount of judgement to apply and anchors to aid that judgement is often not provided. However, they are useful. In a number of instances, the goal was to make these guidelines applicable to automation interfaces, not just traditional HMIs. To aid the reader in their retrieval, most of these checklists are reproduced in the appendices to Rudin-Brown and Green (2017).

Table 9-3. HMI Checklists

Advisors QuickCheck (2002) SWOV (2003a, b).	The Advisors QuickCheck had its origins in the ESOP and considers anthropometric, biomechanical and perceptual/cognitive aspects of the HMI, with specific applications to advanced driver assistance systems (ADAS). There are sections concerning general criteria, criteria related to control (input) functions, and criteria related to display (output) functions.
SafeTE (2007) (Engström and Mårdh, 2007)	This method includes both subjective and objective testing (i.e., real driving to assess visual and cognitive demand). The checklist (subjective) has 7 sections (equipment placement, visual information, system interaction, auditory information, mental workload, driver support, and driver behavior).

Code of Practice for the Design and Evaluation of ADAS (RESPONSE 3 project) (2009)	This ADAS-focused evaluation emphasized controllability, the ability of the driver to (1) perceive the criticality of a situation, (2) decide on appropriate countermeasures (e.g. override, system switch-off) and (3) perform the chosen countermeasure (dependent upon reaction time, accuracy). There is significant European interest in these guidelines.
TRL Checklist (Stevens, Quimby, Board, Kersloot, and Burns, 2002, Stevens and Cynk, 2011)	This checklist is intended primarily for the evaluation of in-vehicle information systems, in contrast to the other methods listed in this table. As examples, questions concern (1) compliance with ISO/BS/SAE recommendations for control locations (true/false/NA) and (2) the use of color conventions / compliance with standards for use of color (true/false/NA). (78 p)
Nuajoks (BMW) Guidelines (Naujoks, Wiedeann, Schomig, Hergeth and Keinath, 2019)	These guidelines were released after the first draft of this report and are contained in the Appendix. For the most part, they emphasize traditional HMI issues.

Given there are so many other guidelines and standards, why was this new guidance document needed?

1. Although the existence of these documents was known, there had been no detailed review of them, in particular, in light of the needs of developing and evaluating automated driving systems.
2. A review of those documents, independently funded by Transport Canada, occurred in parallel with this effort.
3. At the beginning of this project, the impression was, "there are many HMI standards, why do we need more?" In fact, some of the first reports completed for Transport Canada (e.g., Rudin-Brown and Green, 2017) described some of those standards. However, as part of that project literature was identified that revealed that the traditional HMI problems were not the ones that were most critical to automation HMIs, and therefore another report was written (unfunded and not called for by the Transport Canada project) to emphasize this (Green and Rudin-Brown, 2017). In the final reports in that project, 3 vehicles with Level 2 automation systems were examined (Green and Rudin-Brown 2018a, b) and those impressions led to reconsidering the need for more traditional HMI specifications, but particularly for computer menus. Thus, the current perspective is that traditional HMI guidelines do not emphasize what is important for automated driving systems, namely making it clear who is in control and what the system operational domain is. In addition, they do not address menu systems to the degree necessary. However, they effectively handle many basic issues of control and display design.

   Keep in mind that when this project began, many of the points just described were unknown and there was an urgent need to have guidelines to meet specific product needs. Thus, both from the perspectives of knowing what was known when this project was first proposed and what is known now, the guidance development effort was needed.

Q10: WHAT ARE SOME STAGE-BASED TASK DESIGN PRINCIPLES?

Thus, one could list a set of guidelines for each task, which is likely to be extremely repetitive because there are common elements to many tasks. Those elements are the stages or steps in completing a task as identified in the FPDE analysis method. For the most part, a task can be partitioned into 4 stages, find (the control or display of interest), perceive (the information), decide (what to do), and execute (the action). So far, this method has only been applied to visual-manual interfaces. This section describes each of those task stages in detail, the problems likely to occur, and the design solutions for those problems.

Find

The key phrase for the find stage is "some thing in some place." Thus, the issue is what is the driver looking for and where is it? The what part concerns the information the driver is seeking. Associated with that part of the find task is Table 10-1.

Table 10-1. What is the Driver Seeking?

function uncertain	The user does not know (1) what to look for or (2) that the system or function exists. This is can be a documentation issue (e.g., incomplete).
function wrong	The user is looking for the wrong control or display or information in the wrong format (e.g., seeking a beep instead of a flashing light). This could be due to prior experience (expectation) or weaknesses in the documentation.
function not provided	At the time the information is needed (e.g., a warning or control for some purpose), it is not provided.
system identification uncertain	The collection of controls/display is not labeled/identified.
unit identification uncertain	The individual control/display (1) is not or (2) is poorly labeled/identified. (Which switch is it?)
sources compete/conflict	There are competing or conflicting sources needing attention (multiple alarms, multiple controls that could be relevant).
information overload	There is too much for the user.

These problems associated with what the driver is seeking have to do with what can be further grouped into 3 categories, (1) not correctly knowing if the function is provided, (2) not being able to identify it, and (3) having information the driver cannot process properly, because there is too much or it is conflicting. With this in mind, several design principles emerge. Their priorities and insights are the products of the author's exposure to automated vehicles and his knowledge of the human factors research relevant to all types of vehicles.

Guideline 10-1: The HMI must communicate which systems are provided.

The driver must be able to determine which systems are provided by a vehicle without referring to the owner's manual, YouTube videos, or any off-board information. Typically, this is done by warning lights that illuminate when the vehicle is started and by the controls that are present. However, that often is not adequate. No other suggestions for communicating such are provided. Some creativity is required.

In part, this is because the functionality of systems for some purpose differ. For example, for automatic emergency braking systems, differ in terms of the speed range over which they operate, their ability to response to stationary objects, the types of objects they can detect (vehicles, pedestrians, bicycles, trailers, etc.), the weather conditions under which they operate, and so forth. They also differ in their abilities (crash prevention vs. crash mitigation) and the name used for them (automatic emergency braking, crash mitigation system, etc.). This does not make it easy for the driver to understand what is provided.

Guideline 10-2: Collections of controls and displays must be labeled explicitly.

This problem is fairly common in motor vehicles and is easily solved. For example, the vehicle may have a collection of controls to control the adaptive cruise control system, but there is no explicit label to indicate such. An example of how systems in other contexts are labeled is shown in Figure 10-1. Useful cues include surrounding groups of controls with a border, using a common color backdrop, and of course, including the name associated with the group of controls (or displays). The shape of the group does not need to be rectangular. This labeling practice is common in nuclear power plants and aircraft cockpits.

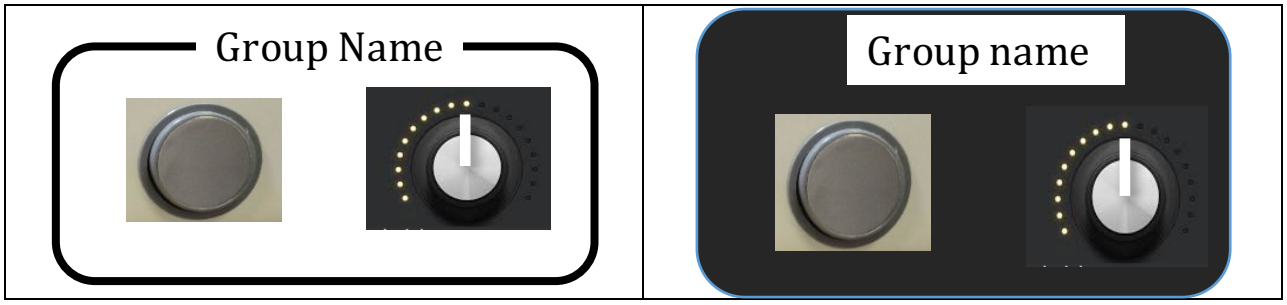


Figure 10-1. Labeling Groups of Controls

Guideline 10-3: Do not overload the user.

When multiple messages are to be provided at the same time, only present the most important message, with a few exceptions. For example, if there is a forward collision warning and then immediately thereafter, a lane departure warning (often triggered by swerving), only present the first warning as the driver is likely to be aware they have departed their lane without signaling. Guidance concerning warnings and determining their priority is presented in ISO 16951 (International Organization for Standardization, 2004) and SAE J2395 (Society of Automotive Engineers, 2002). One notable exception is backup parking warnings, whereas the driver backs

up and directs their attention to the rear, potential forward contracts could be unnoticed and require warnings.

One strategy to assure warnings are detected is to present them multimodally, for example to have both a tone and a visual warning (or maybe even a haptic warning such as a vibrating seat as well. However, if the warnings are not properly synchronized, or if the warning in 1 of the modalities is confusing, then it could take the driver longer to respond if fewer (or 1 modality) was used. Furthermore, keep in mind that the driver should process all of the modalities, and if they occur in series, that sometimes multimodal warnings are not desired. For those interested in the underlying theory, see the research on the psychological refractory period and related topics (e.g., Welford, 1952; Meyer and Kieras, 1997).

Guideline 10-4: Avoid conflicting messages.

These tend not to occur very often in automated driving systems. Examples will be added as they are identified.

The second set of issues related to finding HMI elements is where they are located. Table 12 shows the major issues.

Table 12. Control and display locations

location uncertain	The user does not know where to look.
location wrong	The control/display is not where the user expects it to be.
location inaccessible	The display requires the driver to look away from the road. The control is not within easy reach.
grouping poor	Controls and displays used together are not co-located, or unrelated controls and display are together.

Guideline 10-5a: Place displays where drivers expect to find them.

The challenge is that for many controls and display, there are no strong expectancies described in the literature for many controls and displays. About all that is standard is there are 2 pedals, with the go-pedal on the right and the stop pedal on the left. The direction is controlled by a steering wheel. Turning it clockwise moves the vehicle to the right and counterclockwise to the left.

There are very few expectancies for display location. The trend has been for speedometers to be in the speedometer tachometer cluster, but there are a number of vehicles (e.g., certain Minis, VW Beetle), where the speedometer was in the center stack. Within the speedometer/tachometer cluster, the speedometer can be found to left, to the right, or in the center.

Guideline 10-5b: Place controls where driver expect to find them.

For automated driving systems as an example, cruise controls can be found on the left or right side steering wheel yoke, and on stalks. Further information on this is desired. However, there are constraints, primarily for secondary controls on where they can be located per ISO 4040

(International Organization for Standardization, 2009), SAE J1138 (Society of Automotive Engineers, 2009, and FMVSS 101 (U.S. Department of Transportation, 2011).

In addition, there is some data on driver preferences for controls and how to determine them. The recommended approach is the Potato Head method (Green, Paelke, and Boreczky, 1992). In that method, subjects are shown a blank HMI and asked to place controls and displays where they would expect to find them. For example, in the original implementation, the interior of a car was covered with Velcro and subjects were given hundreds of Velcro backed switches and asked to place them where they belong. If done today, a computer display would be used with images of the controls and displays. There actually is much more to the method than is briefly described here, and those interested in conducting a Potato Head study should see the cited research.

Guideline 10-6a: Place displays where drivers can easily see them.

Automated driving system status displays (e.g., ACC is on, lead vehicle detected) should appear on a HUD or a similar display (e.g., augmented reality), or some other close to the forward field of view because they are regularly used while those systems are in operation. The driver visual field of view is described in SAE J1050 (Society of Automotive Engineers, 2009) with SAE J941 (Society of Automotive Engineers, 2010) being used to determine eye locations. What drivers can see is determined by their seat position, their upper body size, seat back angle, steering wheel rim location, and other factors. Keep in mind that the data in J1050 was collected in the U.S. and represents adults in the U.S. of some time ago.

For automated vehicles, this could be 1 of the most important principles. One of the most common tasks in driving with automation is maintaining situation awareness, namely determining what the vehicle knows and is doing at any given moment. To determine the status of the automation, in particular for some systems when systems are on and engaged, the driver could end up looking away from the road scene, distracting them from paying attention to the road scene. That status could be that adaptive cruise control is on and set, a lead vehicle or other vehicles have been detected, or for some other activity. Typically, this is indicated by symbols on the instrument panel display. Providing this information on a HUD would avoid drawing the driver's vision from the road scene. There could be other acceptable options as well. To get a sense of the time penalty of responding to displays as they deviate from the line of sight, see (Yoo, Tsimhoni, Watanabe, Green, and Shah, 1999; Tsimhoni, Green, and Watanabe, 2001; Lin, Kang, and Green, 2014)

Imminent crash warning displays should appear on a HUD or similar display.

A second application of this principle is for imminent crash warnings such as forward crash warnings, which typically involve some sort of auditory alert, such as series of beeps, and graphic and text indicating a crash warning. Unfortunately, that visual warning often appears on a display between the speedometer and tachometer, inducing the driver to look away from the road scene, not desired.

At this point, there is no accepted auditory signal for an imminent crash warning, in particular one that leads to an immediate and unambiguous response by itself. One interesting alternative being explored is to place a row of LEDs at the base of the windshield and either illuminating a single LED or sequence of LEDs to direct the driver's attention to the threat.

Guideline 10-6b: Place controls where drivers can easily reach them.

Reach limits for driver hand controls are described in SAE Recommended Practice J287 (Society of Automotive Engineers, 2007). Realize that the constraints in J287 (fixed lap belt with a 1 fist of slack between the driver and the belt) are different from the current situation (extendable belt), but in fact, it so happens that the reach distances end up being reasonable. However, realize those data are for a U.S. population of some time ago, and folks in the U.S. are now much heavier. For additional information on this topic, see ISO 3958 (International Organization for Standardization, 1977). Between nations, there can be large anthropometric differences.

In addition, ISO 4040 (International Organization for Standardization, 2009) specifies where controls should be placed. However, the placement is in terms of zones, and there are many zones in which controls can be placed, so the constraints are minimal.

On final caveat… Some control locations which seem desirable, such as the low on the center console or on the steering wheel yoke (or below) have disadvantages. The center console location, such as aft of the normal shift location, is quite far from the normal line of sight, so if the driver needs to use vision to identify labels or select 1 control from another, then there can be a substantial visual penalty. For controls on the steering wheel yolk, the close distance can be a challenge for older drivers wearing corrective lens to see at a distance. Those drivers may need to flip their glasses upward to be able to focus at that close distance.

Guideline 10-7: Put controls and displays together that are related or used together and separate those that are not.

This principle concerns grouping controls together, grouping displays together, and grouping controls with displays. This principle is sometimes in conflict with either the easy to see or easy to reach principles. For example, a control that is easy to reach could be located near the gear shift lever between the driver and passenger. However, a display placed near the shift lever (in on the transmission tunnel) is too far from the forward scene to be seen easily.

To aid the reader in understanding this principle examples include (1) placing the set +/resume – switch for the cruise control on the steering wheel yolk and the on /off switch on the instrument panel (which should not be separated), and (2) placing the adaptive cruise control functions and the entertainment source selection switch next to each other on the right side of the steering wheel yoke (which should not be together).

Perceive

Table 10-2 lists some of the characteristics that are important for perception. Factors related to both intensity and timing are listed together. For all modalities, the key issue is the signal-to-noise ratio.

Table 10-2. Factors related to Perception

intensity too low	The signal/display/control is difficult to perceive (low contrast, low intensity, tiny text or graphics), and thus in some sense is uncertain.
intensity too high	The signal is too intense (startling) or too large, leading to sensor overload. Making some information elements to be too large, can force others to be too small.
timing wrong	The signal is too brief, too long, or the timing of the pattern is undesired (e.g., too many beeps).

Guideline 10-8: Auditory warnings need to be loud enough so they can be readily heard. Text and graphics need to be large enough and have enough contrast so they can be easily seen. Vibrations need to be intense enough to be readily felt.

Values that are recommended to satisfy this principle are in the specific interface elements and task section later in this report.

Guideline 10-9: For visual, auditory, and haptic patterns, breaks between flashes, beeps, and pulses need to be long enough so that they are perceived, but not so long that the sequence takes too long to complete, delaying driver responses.

As a starting point, use the middleman value for a perceptual processor cycle in the Model Human Processor (Card, Moran, and Newell, 1983) of 100 ms for the off duration, although one can be certain the stimuli are distinct at 200 ms for the off duration. Likewise, breaks between stimuli should be of similar durations.

Guideline 10-10: Keep sequences of flashes, tones, and vibrations short.

A number of problems can arise. In many situations, drivers wait to respond until the sequence is completed, thus increasing response time to sequences used as warnings. Thus, if a pattern is presented, then there is a brief delay, and then there is a repeated presentation of the same warning, drivers may delay in responding until both presentations are complete. The repeated presentation was intended to make sure the warning was perceived and deemed urgent, but in some situations the repetition is not desired.

Secondly, in some instances, the number of stimuli (e.g., beeps) is meaningful. If they are paying attention, drivers can readily distinguish 1 vs. 2 beeps, and often 2 vs. 3, but greater values are problematic.

Decide

Characteristics related to making decisions fall into 2 categories, (1) the information and (2) the process. Information-related characteristics are shown in Table 10-3.

Table 10-3. Information-Related Characteristics

information uncertain	The user does not know if the information is correct.
information incomplete	All of the information needed to make a decision are not provided (e.g., missing switch position label, information from memory is forgotten).
information wrong	The information is wrong.
format wrong	The information is not in the format that is appropriate for the task (e.g., giving them information in a language they do not know).
meaning uncertain	The signal/image/label is uncertain. What does this mean? Color or other coding is absent.

Guideline 10-11: Provide information the user will know is correct.

This principle is not invoked very often. Often the situation is that the meaning of information provided is uncertain, which is a later principle.

Guideline 10-12: All of the information needed is not provided.

Most commonly, for the interfaces of interest, this will be controls that are not labeled or control or display positions that are not labeled.

Guideline 10-13: Provide correct information.

The wrong information is provided or entered. Usually the case is that the information entered by the driver is wrong, not that the wrong information is provided by the user. One of the more common situations is when users are to enter text, such as a destination, and they misspell the destination name.

Guideline 10-14: The format of the information should match the decision the driver needs to make and not require recoding or transformation.

For example, if the driver want to know the absolute distance to a turn far ahead, they should not be give a bar, but rather the distance in miles or kilometers, as appropriate for where they are driving.

Guideline 10-15: The information should be provided in a format that is meaningful to the driver.

For example, if warning the driver about a failure of the Blind Spot Warning System, flashing the text BSWS is unlikely to be meaningful. Even though it is in ISO 2575, the "talking tombstones" icon (Figure 10-2), when display away from the exterior mirrors, is unlikely to be informative as well.



Figure 10-2. ISO Blind Spot Icon

Table 10-4 shows process-related characteristics pertaining to decisions.

Table 10-4. Process-related characteristics pertaining to decisions

grouping poor	The information needed is provided, but not grouped together. Why is this over here? Also included is where elements that do not belong together are grouped.
process uncertain	The user does not know how to decide. Sometimes, the system says what is wrong but not what to do.
process wrong	The user misunderstands how to decide.
task too complex	The task requires the user to do more than they know, remember, or can process.

Guideline 10-16: To aid making decisions, the information relevant to that decision should be grouped together.

This principle is similar to one associated with funding information, but there is a subtle difference. For example, suppose the driver needs to make a decision about stopping for gas or for charging the vehicle. The range remaining display should be near the map that showing nearby gas or recharging stations.

Guideline 10-17: The interface should clearly communicate how to perform a task.

For the sake of simplicity, situations where the driver does not know what to do or misunderstands what to do have been grouped together in this version of these guidelines. An example would be where to operate a menu system, the driver needed to pull on a lever, but there was no indication the driver should do this to operate the control (instead of pushing it). Often when this occurs, what the driver should do violates expectancies.

Guideline 10-18: Minimize the number of steps and the time required to complete a task.

This is often a problem with menu systems. Menu system design is described in a separate section later in these guidelines. Consider the case of a menu system that only has 2 controls, 1 to select a choice and 1 to move to the next menu item. If a driver wanted to change some system setting for a feature that was 2 levels down, then they would need to select the menu function, then scroll through the list of options, and then repeat the tedious scroll-select process. Typically, one exits a menu system such as this with "go-back" as a menu choice.

Selection time can be predicted using SAE J2365 (Society of Automotive Engineers 2002), Extensions of it being developed, and possibly other methods. Details appear in the sections on menus.

The NHTSA guidelines (U.S. Department of Transportation (2013) provide limits on the number of glances for visual-manual tasks. Some of the visual-manual tasks that are not permitted include: (1) text messaging, (2) Internet browsing, (3) social media browsing, (4) navigation system destination entry by address, (5) 10-digit phone dialing, and (6) displaying to the driver more than 30 characters of text unrelated to the driving task.

In addition, NHTSA also identifies 2 methods to determine if a task has excessive visual demands, the most widely used of which is the occlusion method. In that method, subjects wear a pair of goggles (Figure 10-3) which contain LCD shutters that open and close, which each open duration and each closed duration being 1.5 s (for a total of 3 s). This cycling is intended to represent looking at the interface for 1.5, a typical glance (so the shutter is open) and then looking at the road (so the shutter is closed). The total time the interface can be seen is 12 s, or 8 open cycles.



Figure 10-3. Occlusion Goggles
Source: http://www.redscientific.com/index.php

If each step in a task involved looking at the display and then pressing a key, then in theory, the maximum number of glances to complete at task would be 8. However, often the driver looks at

the interface and then makes a second glance to guide their hand to the first button, so the first button press requires 2 glances. If they keep their hand near the display, then the remaining button presses can be completed with 1 glance each, for a total of 9 glances. However, it is often the case that the display provides visual feedback or some search of the display is required, in which case something more like 5 or 6 glances is the maximum possible. As was noted previously, extensions of SAE J2365 are being developed to provide a more precise determination.

To assure compliance with the NHTSA guidelines, the occlusion task time for each task should be calculated and those tasks whose duration is close to the NHTSA threshold should examined experimentally. The intent is that as the calculation is improved, that the experimental evaluation will not be required. Keep in mind that the experimental evaluation is an estimate, not the true time, just as the calculation is an estimate. Occlusion task experiments take time to complete, are costly, and can only be done when the interface is functional. Calculations can be made when wireframe drawings are available, much earlier in design when changes are easy to make.

Execute

Tables 10-5 (and 10-6 later) shows process-related characteristics pertaining to executing decisions, mostly using controls. The first table concerns what is to be moved. The second concerns how.

Table 10-5. Control-related Principles –What to Operate

affordance lacking	The labels and/or shape of the control do not suggest how to operate it (to turn, to push, to push and hold for some time). Where do I push?
manipulation concerns	The control is difficult (1) to grasp (usually too small) or (2) otherwise manipulate.

Guideline 10-19: Design controls so that their design suggests how they should be operated.

For example, for a knob, providing a surface to grasp suggests it should be grasped. If the surface should be serrated on the side or a bar depends upon which surface is expected or intended to be grasped.

One of the most common problems for touch screens occur when the images for controls and for displays look alike. In that situation, when not sure what to do, drivers try to press everything. The solution is to make virtual controls look like physical controls, for example to show the sides of a rocker switch, or for a push button, to show a drop shadow of the button elevated above a surface (Figure 10-4).

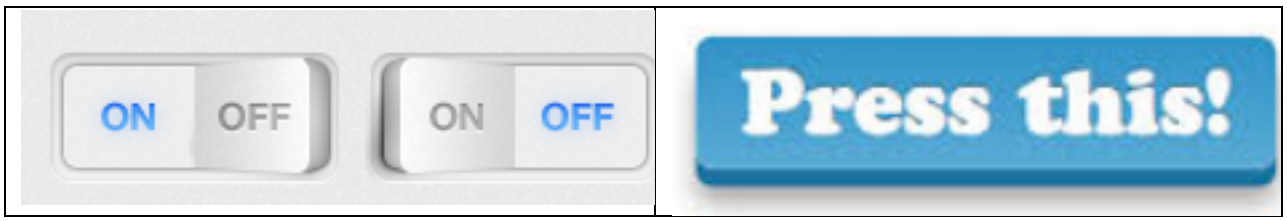


Figure 10-4. Switches with Drop Shadows
Source: http://ui-cloud.com/free-ui-toggle-switches/
https://blog.typekit.com/2011/02/10/type-study-an-all-css-button/

Guideline 10-20: Controls should be large enough so they can be easily grasped.

Table 10-6 lists the principles concerning control operation. Recommendations on minimum control sizes are well described in the literature. As a starting point, use the recommendations in Mil-Standard 1472G (U.S. Department of Defense, 2012). Although this standard is a design specification intended for application for military systems, it is considered to be the human factors bible and the bare minimum design requirements for systems for all applications. Keep in mind that in some regions, in the winter, controls are operated while wearing gloves.

Table 10-6. Principles Concerning Control Operation – How to Operate

logic inconsistent	The operation of a set of controls is not consistent (push this one, but push and hold this one).
stereotype inconsistency	The movement of the control is not consistent with stereotypes.
consequences unintended	Operation of the control has unintended consequences (bumping another control, shutting off a safety system). Controls may be too close together.
forces inappropriate	The control forces are not as desired (too high or too low). It is too hard to move (or too sensitive).
feedback insufficient	If a control has been operated is either unknown or uncertain (due to lack of tactile, visual, or auditory feedback).

Guideline 10-21: Controls should operate in a manner consistent with their appearance.

Controls that have different purposes should not look alike. For example, a button should look different from a slide switch or rocker switch, and buttons that are push to operate should look different from those that are push and hold to operate. These problems are most common with touch screen buttons, where efforts to create visual harmony lead to confusing the driver.

Guideline 10-22: The operation of controls and movement of displays should conform to population stereotypes.

The rule of thumb is forward, clockwise, to the right and up for on or increase as described in SAE Recommended Practice J1139. Additional information is in the Tasks section.

Guideline 10-23: The operation of a control should not have unintended consequences.

Most commonly this occurs when controls are located too close together, leading to inadvertent operation. For example, consider the situation of 2 pushbuttons located side by side. If the buttons are small, then pushing one could result in both of them being pushed. Separating or providing a raised bar between then can reduce the probably of multiple switch closures. Avoiding this is a challenge for small touch screens, where barriers are not feasible and there is a desire to provide significant functionality. When considering the amount of separation required, keep in mind that in many regions, in the winter, controls are operated while wearing gloves. This is a topic for which Mil-Standard 1472G (United States Department of Defense, 2012) is a good resource.

Inadequate separation problems can occur for slide switches, knobs, and many other types of controls, both virtual and real.

Guideline 10-24: Forces to operate controls should be not so excessive as to interfere with intended operation but not so low that inadvertent operation can occur.

The most common problem in contemporary vehicle is forces for touch screens, where sometimes the operational forces are high. In many cases, it is not a force problem, but one of the contact area. In other cases, it is a matter of drivers being off target because their hand obscures a switch label or because of inaccuracies in reaching far away. See for example Hall, Cunningham, Roache, and Cox (1988), U.S. Department of Defense (2012) and Colle and Hiszem (2004).

In some instances, especially for mechanical controls that are closely spaced, sometimes forces should be set a bit on the high side so when brushed against, they do not operate.

On interesting example of how challenging setting forces levels can be is the case of the push button ignition. Being able to have an electronic key in one's pocket and only push a button on the instrument panel to start a car is a convenience. However, drivers who are used to turning and removing a physical key can forget to shut down their vehicle, especially if it is quiet (e.g., an electric vehicle). Interestingly, in the case of the Tesla, leaving with an electronic key shuts it down. For many contemporary vehicles, manufacturers are concerned with a child in the front seat randomly pressing buttons on the instrument panel. With a mechanical key on the steering column, it was not easy for a passenger (child) to reach and turn the key. With a pushbutton control, the control is often located on the instrument panel closer to the front passenger seat, which is easier for the driver, and now the passenger to reach. To make it more difficult for inadvertent shut down (by a passenger), many vehicles have a press and hold shutoff, although the duration of the press, and if more than 1 is needed varies between vehicles.

Guideline 10-25: Provide feedback that a control has been operated.

This feedback can take many forms such as the sound of engine starting, a radio station changing, or a map scale changing.

The most common problems occur when the design try to uses 1 tone for everything, sometimes varying the number of tones to indicate different messages. Clearly, it is a mistake to use the same tone to indicate a switch has been operated or that the wrong switch (one not relevant at a moment) has been pressed or that some other error has occurred. Tones are also used to indicate warnings, especially crash warnings, typically using a sequence of tones, with the number indicating urgency. Unfortunately, that sequence does not indicate where the driver should look. This can be problematic for parking system and backup warnings.

Q11: HOW SHOULD SPECIFIC INTERFACE ELEMENTS BE DESIGNED TO SUPPORT DRIVER TASKS?

This section provides detailed guidelines on a number of topics of particular interest to automated vehicles.

Displays

Display Type

Guideline 11-1: The information in a display shall be in the format the driver needs to make a decision that is part of a task.

This is a simple idea and very important for display design. Graphically, that guidance is presented as shown in Figure 11-1. So, if a driver needs to make a decision that requires an exact value, then show an exact value (a number). If the driver needs to know if something is increasing or decreasing or is approximate, then a moving pointer display is recommended. The challenge is that sometimes there are multiple tasks that use the same display, so some weighing of the importance and frequency of each task, along with the merits of each display for each task is required.

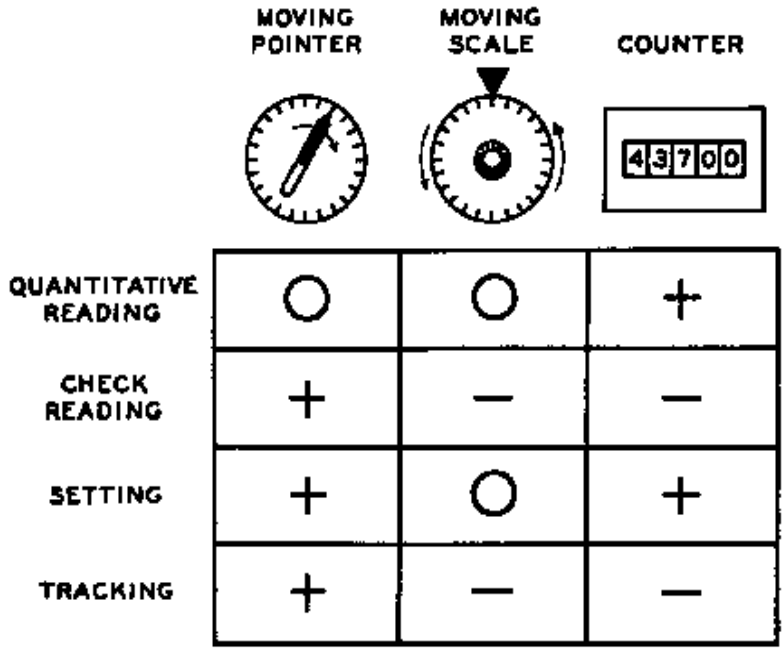


Figure 11-1. "Chapanis" Figure
+ = Satisfactory
o = Acceptable but not completely satisfactory
- = Usage should be avoided

Source: Chapanis (1960).

Much of the research on Head-Up Displays (HUDs) has been conducted in aviation, where the focus is on one's attitude in 3-dimensional space, not 2 dimensions. The classic reference is the Book of HUD (Weintraub, 1992) with Tonis, Klinker, and Plavsic providing a useful summary of design issues. See also Gish and Staplin (1995). For some interesting student concepts, see Bostrom and Ramstrom (2014). There is some basic information on this topic in the Campbell guidelines (page 6-14), but additional information is needed. The most important recommendation is for the focal distance of the HUD image to be beyond the front end of the vehicle, with farther distances minimizing the amount of refocusing from the road scene.

Aside: As Daniel Weintraub, the author of the Book of HUD (Weintraub, 1992) often said, the term is head-up display, not heads-up display, because people have only 1 head.

Text Legibility and Readability

Guideline 11-2: As a starting point, text height should be at least 0.007 times the viewing distance. Important displays should be double to quadruple that size.

How big does text need to be to be (a) legible and (b) easy to read? No topic has been the subject of more research than the legibility of text, with the central question being how tall do letters and numbers need to be (Cornog and Rose, 1967; Mourant and Langolf, 1976; Green, 1988; Cai, Green, and Kim, 2013; Reimer et al, 2014). For text, for roman characters, the simple rule of thumb is that the minimum height of characters should be 0.007*(viewing distance), commonly referred to as the Bond Rule (as in James Bond, Smith, 1979). The units of height and vehicle distance are the same. The Bond Rules ignores contrast, the type face, viewer visual acuity, and a host of other factors that influence the legibility of text, and lead to smaller or larger recommendations. The advantage of the Bond Rule is that the recommended size can be calculated in one's head.

However, the recommended size is not necessarily what the Bond Rule suggests. For text to be easy to read, the rule of thumb is that the character height should be double the threshold. In addition, for graphic characters, the legibility threshold is about double that of roman characters (Oda and Imahashi, 1995). (See also Cai, Chi, and You, 2001, 2003; Ayama, Ujike, Iawi, Funakawa, and Okajim, 2007 as well as Peters and Adams, 1959.)

For a more precise calculation, the NBS method is recommended (Howett, 1983). However, the NBS method requires much more detail, some of which is usually known (the mean character height to width ratio for a character set, the viewing distance). Other information may not be readily available, such the background and character luminance (determined using a precision spot photometer capable of measuring 0.1 degree targets) or the distribution of viewer visual acuities. Viewer visual acuity is specified by the Snellen ratio. In the U.S., 20/20 is considered to be normal visual acuity. For that ratio, a particular observer can see at 20 feet what a normal observer can see at 20 feet. If a person was worse than normal, then their acuity could be 20/40, which says that can see at 20 feet what a normal observer can see at 40 ft. In countries where the metric system predominates, the units of acuity are expressed in meters, with 6/6 considered to be normal.

The visual acuity used in the calculation should be a reasonable worst case. If the mean is used, then in theory the display is not legible to half of the population.

Guidance: To precisely calculate the character size is required for threshold legibility, use the NBS method.

The NBS equation is expressed as follows:

Contrast = C = ((Lb - Lt) / Lb) x 100
where Lb is the luminance of the background
Lt is the luminance of the target (the character)

Effective Snellen Acuity =S = Sd x $(85/Lb)^{0.213}$ x $(90/C)^{0.532}$
where Sd = the denominator in the Snellen ratio

Height = H = (H:Sw) x 1.45 x 10^{-5} x S x D
where H:Sw = Height to stroke width ratio (for 8:1, the value is 8)
D = viewing distance

Again, note that the NBS method provides a more precise estimate, but it requires more information, some of which may not be available. In many situations, the quick estimate from the Bond Rule is sufficient.

Another noteworthy equation is the estimated size for speedometer digits when the display is numeric (not analog, that is, having scale and pointer). Based on Boreczky, Green, Bos, and Kerst (1988), the time to respond to a speedometer and make decisions about the speed shown can be estimated using the equation that follows. Notice, this equation does not determine a recommended height. Rather, increasing the height decreases the response time in a manner inversely proportional to a weighted combination of the height and height squared. See Boreczky, Green, Bos, and Kerst (1988) for details.

RT (ms) = 1054 - 320(A) + 1050(1/H) + 2202(L)
+ 89.6(1/ln(C)) - 9.58(ln(I)) + 4538($1/H^2$)

where: A = Age Group (1 for old, 2 for young)
H = Digit Height (mm, for 5-19 mm)
L = Location (1 for center, 2 for sides)
C = Contrast Ratio (for 1.5:1 to 20:1)
I = Illumination (lux, for 1.08 to 915)

Text Warnings and Other Messages

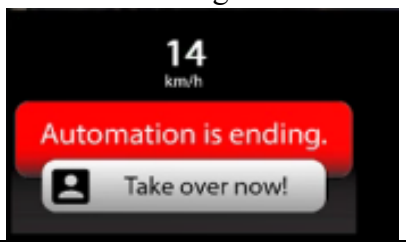
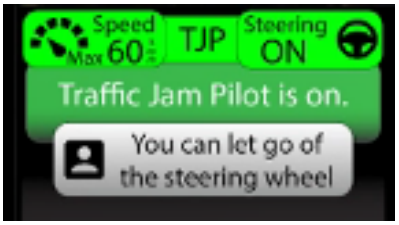
Motor vehicles present messages to drivers as text in roman characters. These messages often communicate to the driver that something is wrong with the vehicle, but the messages could be for other purposes, such as navigation instructions. Often the number of characters that can be presented is limited, (1) sometimes because the display is character based and the number of

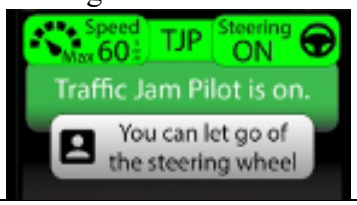
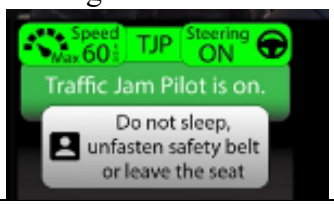
characters is limited (e.g., 2 lines of 24 characters), (2) sometimes because the overall size of the display is limited, and providing more characters would reduce their size below a legibility threshold, and (3) sometimes for other reasons. The important point is that the number of characters is limited and there is no easy work around that limit. For those situations, how should messages using roman characters be presented?

The literature on error messages for computer systems often refers to 5 criteria on which those messages should be assessed, (1) specificity, (2) constructive guidance, (3) positive tone, (4) user centered style, and (5) appropriate physical format. Damouth and Green (1997) expanded upon those ideas, creating more extensive list of criteria, shown in Table 11-1, though in a different order than originally presented. See also Green (1996.) The guidelines were used extensively in designing the HMI for the interface tested in clinic 2.

Table 11-1. Design Criteria for Short Text Messages and Warnings

#	Criteria	Question	Comment / Example
1	Understandability	Do users understand the message (as indicated by the comprehension data and/or their comments)?	Just ask users to read the message aloud. If there is hesitation in their voice, they do not understand it.
2	Error feedback	Does the message tell the user what is wrong?	Example: *Get service* (no error feedback) Without information about why, drivers do not know if it is necessary or how soon to do it.
3	Positive guidance	Does the message tell the user what to do?	Example 1: *Low oil pressure* *Stop safely with 1 minute* Most drivers do not know that when the oil pressure is low, one needs to stop very soon before the engine seizes. Example 2: *Engine* which gives no guidance
			Example 3: This message says what is wrong and what to do. 

4	Desired response	Do users carry out the wrong action or actions at the wrong time?	Example: *1st gear* Does this mean only to drive in first gear (because the other gears have failed) or not to drive in first gear (because it has failed).
5	Specificity	Could the user interpret a message in more than 1 way?	Example: *Fan fault* *Seek service* Has the engine cooling fan failed (which in warm weather usually requires attention in minutes) or the climate control fan failed (which is less urgent)?
6	Consistent format	Does the first line say what is wrong and the second line what to do? Is the same message phrasing always used?	Example 1: *Slow down* *Excessive drivetrain vibration* People expect to be told what first and then why. If the message is ambiguous, violating the expected order makes the message less comprehensible.
		Example 2: The interfaces of clinic 2 had a consistent 2-part format as the messages were more than 1 line long. 	
7	Temporal order	When there are multiple steps, are the listed in the order of occurrence	Example: *Engine overheating* *Find a safe place to stop.* *Open the hood.* - Not - *Engine overheating* *To be able to open the hood* *Find a safe place to stop.*

		Message 1 	Message 2 
8	Continuation	Is a space provided for continuation lines?	Indent 1 or 2 spaces for continuation lines Example: This *Check the oil pressure and temperature* - Not - *Check the oil pressure and temperature*
9	Common language	Are the terms in the user's language (e.g., for the US, common American English)?	Example: (not this) *System BDx 15 malfunction*
10	Minimum verbiage	Are the fewest characters used that maintain comprehension?	*Blind Spot Warning* could be shortened to *Blind Spot Warn* - Not - *Blndsptwrnng*
11	Aesthetics	Does the message make the product look cheap?	Example: *Blind Spot Warning* - Not - *BS warning*

Abbreviations and Acronyms

Guideline 11-3: Use the same combination of truncation and vowel deletion rules when creating all abbreviations.

Given limitations in space, it is often necessary to abbreviate text so the text is legible. However, how well drivers, not the engineers who design vehicles, comprehend abbreviations is not well documented, but it is believed to not be high, even for terms such a FCW and BSW, that are used with some regularity. Part of the problem is the TLAs used (Three Letter Acronyms) are not consistently created because the words "warning" or "system" may or may not be part of them, the root words are different, the manufacturer may include their name as part of the system, or because the system name appears to be a "made up" term (e.g., Distronic).

The most important guidance concerning constructing abbreviations is that all abbreviations related to a product, the entire HMI in this case, are created in a consistent manner. Research (e.g., Nawrocki, 1979; Ehrenriech, 1985) has shown that truncation (the first n letters, often 5) leads to the best comprehension, followed by vowel deletion, which is better than using existing

abbreviations or even populations stereotypes. The relatively poor performance of population stereotypes is unexpected as a key principle is to seek and rely upon user input when designing an interface. Upon explanation, this finding makes sense. When creating abbreviations, people do not use a single consistent method to create them. Hence, when others are presented with an abbreviation based on stereotypes, they do not know how to reconstruct the original term because no consistent rules were followed. Thus, the most important guideline is however abbreviations are created, that the same set of rules be applied to all abbreviations. Often those rules are some combination of truncation and vowel deletion because multiple words, when truncated, have the same first few letters.

One common phenomenon in text messaging is to use shortened terms (netlingo) to speed entry, and how they will be used in vehicles is unknown. Data on Netlingo understanding has been the topic of limited research and Netlingo should be avoided. Some terms are likely to be understood (DIY=do it yourself), but others are not (AAK or AATK = asleep at the keyboard).

For related information on acronyms, see McCrobie (1986.)

Icons/Symbols

Guideline 11-4: Use the symbols in ISO 2575 supplemented by text.

Symbols required to be used in motor vehicles are specified in (1) ISO 2575 (International Organization for Standardization, 2010), (2) Federal Motor Vehicle Safety Standard 101 (U.S. Department of Transportation, 2011, which contains a subset of those in ISO 2575), and (3) SAE 2402 (Society of Automotive Engineers, 2010, which also contains a subset of those in ISO 2575).

In response to ongoing requests for new symbols for new functions, ISO 2575 International Organization for Standardization (2010) has been amended almost every year over the last few years. Accordingly, designers should have both the base standard and all amendments. Each amendment only contains the symbols that were added that year, not those in previous amendments. The symbols in 2575 are not reproduced here because they are copyrighted.

There is no reason to believe that any particular symbol in ISO 2575 will be legible or comprehended by drivers, and in fact, the evidence is that many symbols that have been in vehicles for many years are not well understood by drivers (e.g., Green and Pew, 1978; Carney, Campbell, and Mitchell, 1998). This is because many of the symbols in 2575 have never been tested for legibility or comprehension. Those seeking to do so should see SAE Recommended Practice J2830 (Society of Automotive Engineers 2008) and all 3 parts of ISO 9186 (International Standards Organization, 2008, 2014a, b). It should be noted that poor comprehension of active safety symbols (Campbell, Hoffmeister, Kiefer, Selke, Green, and Richman, 2004) is of particular concern.

To overcome the lack of comprehension of symbols, some supplemental text (often an abbreviation) is desired to increase comprehension. In some sense, that defeats the purpose of

symbols, which is to effectively communicate an idea in a limited amount of space. In many situations, the use of symbols is mandated. For design guidance, see Chi and Dewi (2014).

Navigation (Navi) Screens

An ongoing implementation issue is how navigation information will be presented in the future, via an in-vehicle system, the driver's smartphone or tablet, or some combination of in-vehicle and carry-in devices. The personal devices tend to cost less, and because of turnover, are generally newer. Most automakers provide some means to interface phones and tablets to the vehicle HMI but discourage direct use while driving other than to play music. This is because the displays tend to be small and touch interaction is visually demanding. Mounting devices are invariably aftermarket products. Furthermore, the personal device will go through several life cycles over the life cycle of a vehicle, and predicting what features a personal device will have several life cycles into the future is difficult.

An important theme repeated in this document is that how information should be entered will depend upon what the task is, who is doing it, and what the context is. For navigation systems, the task goals are to enter the destination, understand the plan to get to the destination, follow the plan, and make adjustments as needed (for example, due to traffic).

Guideline 11-5: Allow for destination entry in many methods and in any order.

Methods can include street address, point of interest (POI), intersection, area, or location on a map. There is data on how often each method is used is limited (Lo, Green, and Franzblau, 2011). However, what is important is that the interface support entry of the fields in an order, and not force the entry of state first, followed by city, etc. This forced sequence makes the programmers' task easier, but not the drivers' task.

Furthermore, many automotive systems force entry of destinations using a proper postal address, but do not support entry using region or neighborhood names (e.g., Manhattan vs. New York City). The focus needs to be on how the driver knows the address, not the system. Another problem concerns users not knowing street name prefixes (North Main vs South Main) or suffixes (Peachtree Avenue, Lane, Parkway, etc.). There are 71 variants of Peachtree in Atlanta (https://en.wikipedia.org/wiki/Peachtree_Street, retrieved March 30, 2018)

Guideline 11-6: For planning, show the map north up. For guidance, show the map heading up.

When planning a route, people take a "god's eye" view of the world to help them understand how to get there. Typically, the route is highlighted, but the names of the roads travelled, which is needed information, are not, but should be. The tendency is to either implement to duplicate whatever Google maps does, as shown in Figure 11-2.

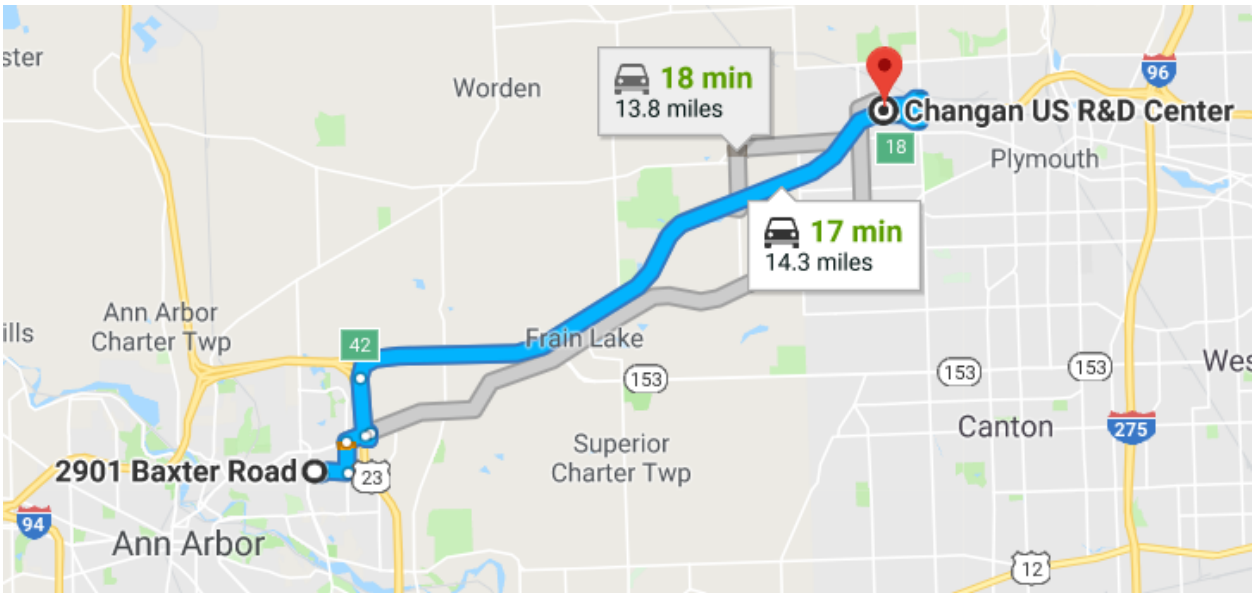


Figure 11-2. Google Maps Route - UMTRI to the Changan US R&D Center, Retrieved March 30, 2018

When driving, drivers need to coordinate what they see out the windshield with what they see on the map. If the 2 views are incompatible, then mental rotation is required of the map, which is a challenging task. "I turn right at the intersection ahead, but because the map is upside down, it appears I am going to the left."

One solution to this problem is to split the map screen and either show 2 maps or a map and guidance in some other format, such as a turn list. How effective this solution is could depend if the reduced size maps are large enough to easily read.

Guideline 11-7: For route following, provide turn guidance directly in front of the driver, for example on a HUD or augmented reality display.

When making "turn" decisions, there is often little time to decide available, especially when traffic is present. Thus, in those instances, driver do not have time to look away from the road scene to study a display.

Guideline 11-8: For navigation decisions, providing good voice guidance is a priority and is even more important than the visual display.

This is because the driver often cannot look away from the road. In fact, navigation can be effectively executed using voice guidance system alone, but drivers prefer to have a display for orientation.

Guideline 11-9: Provide a "say again" button for voice guidance that allows the driver to have the last spoken instruction repeated without their hands leaving the steering wheel.

Often, in intense situations, drivers may not be able to attend to the first presentation of a navigation instruction, and want to hear it again. This often occurs if one device is providing music and another is providing navigation instructions.

Guideline 11-10: Include landmarks in turn instructions.

If one asks a person for directions, they naturally include landmarks in their directions because providing landmarks make following the directions less error prone and easier (Burnette, 2000a, b, Burnett, Smith, and May, 2001; Elias and Brenner, 2005, Large, Burnett, Benford, and Oliver, 2016). ("Go 2 blocks to the stop sign. Turn right and go to the second traffic light. There is a gas station on the right….") The advantage of landmarks is that they can be seen from afar, so there is more time to plan maneuvers. Most navigation systems do not include landmarks, reportedly because of the effort to keep the data current and correct. Furthermore, some data bases do not support them.

Guideline 11-11: Provide countdowns to indicate the distance to turns.

Even though countdown displays have been in navigation systems for many years, it is uncertain if the countdown display should be a shrinking bar, a numeric distance, tones, or a multimodal combination. For timing information, see Green and George (1995), Green, George, and Fleming (1995).

Guideline 11-12: For closely spaced turns, present them together.

"Make a left turn and then a quick right." This is because after the first turn there is not enough time for the driver to hear the second instruction, plan what to do, and execute the maneuver in a nonaggressive manner.

Guideline 11-13: Developing solutions to the "last block" problem will lead to much more useful navigation systems.

Depending on the navigation system, the last instruction could be, "you have arrived at your destination," or "your destination is on the right." What the driver wants to know is which building is the destination (which requires knowing what the building looks like), where the building entrance is, and where to park?

Controls

Controls can be grouped into 2 categories, discrete and continuous depending upon the nature of the output, as series of discrete states, or a continuous value. Continuous controls include joysticks, cranks (for windows), steering wheels, foot pedals, and thumbwheels. For the most part, the types of controls used to provide continuous output unlikely to change for motor vehicles over the near term, so the focus of this section in on discrete controls. (For example, steering wheels are unlikely to be replaced by joysticks.)

Guideline 11-14: For 2 state selections, use rocker switches, with pushbuttons as alternatives. For 3 or more states, use slide switches or knobs with discrete positions.

Among traditional controls, the switch that is most appropriate depends upon the number of states from which selection occurs. The advantage of a rocker switch is that it makes it apparent that the selection is between 2 states. One can think of a slide switch as a stretched out knob. Knobs tend to be more compact. Slide switches tend to have stronger directional stereotypes and are more easily represented on touch screens.

Direction-of Motion-Stereotypes (Population Stereotypes)

How people expect displays to move in response to control actions is referred to direction-of-motion stereotypes (Figure 8). It must be emphasized that these stereotypes are trends, and there is not 100% agreement for any control-display movement combination, and with the strength of the stereotype depending upon human range of motion and learned relationships. Some learned relationships differ between countries. For example, in the U.S., to turn on a light switch mounted on a wall, the stereotype is to move it up. In the UK, the stereotype is to move it down. However, overall, there is agreement to some stereotypes. Unfortunately, at this time, those with interest in comparing alternative control-display combinations will need make decisions based only on these principles and the literature as there is no integrating computational model. However, the 7 general rules of stimulus-response compatibility that follow will be helpful in making design decisions. Also useful will be SAE J1139 (Society of Automotive Engineers, 2010).

Society of Automotive Engineers (2010). Direction-of-Motion Stereotypes for Automotive Hand Controls (SAE Recommended Practice J1139), Warrendale, PA: Society of Automotive Engineers.

Guideline 11-15: Stereotype 1: Numbers should increase from left to right **(except in Arabic and Hebrew).**

Guideline 11-16: Stereotype 2: Dimensional Expectancies - Clockwise, up, to the right, or forward for increase or on.

See Figure 11-3.

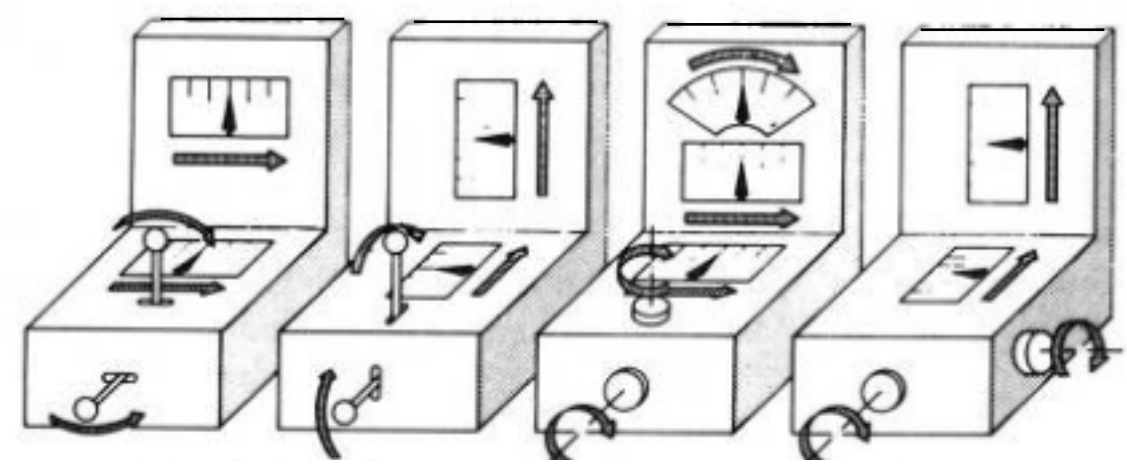


Figure 11-3. Example Population Stereotypes

Guideline 11-17: Stereotype 3: Clockwise-Clockwise Hypothesis -Clockwise control movements lead to CW display movements.

Guideline 11-18: Stereotype 4: Parallel Movement - Linear controls and linear scales move together.

See Figure 11-4.

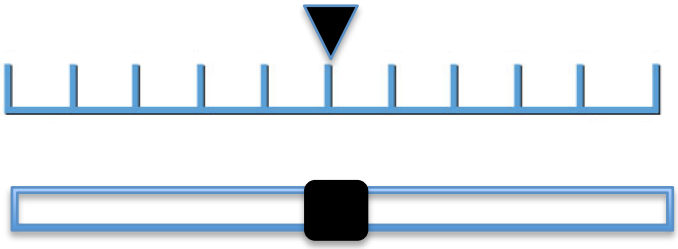


Figure 11-4. Parallel Control-Display Movements

Guideline 11-19: Stereotype 5: Warrick's Principle – An indicator moves in the same direction as the side of the control adjacent to it (scale side effect).

As shown in Figure 11-5, the knob and associated display are thought of as a rack and pinion.

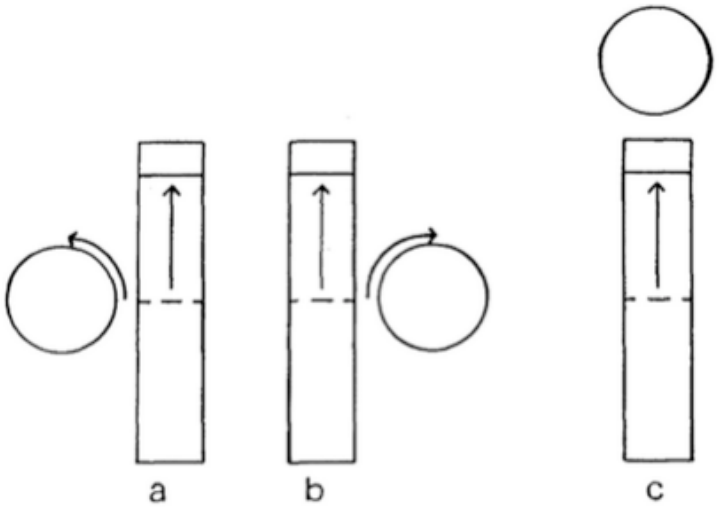


Figure 11-5. Scale Side Effects.

Guideline 11-20: Compatibility Principle 6: The initial position of the indicator affects the control action.

The tendency is to move towards midrange of the display.

Guideline 11-21: Compatibility Principle 7: Available Limb Motion - Clockwise with right hand, counterclockwise with left.

The hand or arm tends to move in the direction from which the greatest range of motion remain from the position in which the control is grasped.

Touch-Screen Controls

Surprisingly, there are few comprehensive studies comparing the usability of alternative touch screen controls for common tasks. For those interested in comparing alternatives, consider using the Model Human Processor to compare alternatives before conducting experiments. The best known study of touch-screen input is Johnsgard, Page, Wilson, and Zeno (1995). In their research, subjects were shown a request (e.g., select Wednesday) and after seeing it, pushed a start button. Subsequently, a specific input device appeared (Figure 11-6) that the subject used to select the correct response (e.g., selecting "Wednesday" from a list of menu options). A logical next step would be to develop models for each device and then compare devices (and the models for them) as the number of options is systematically varied.

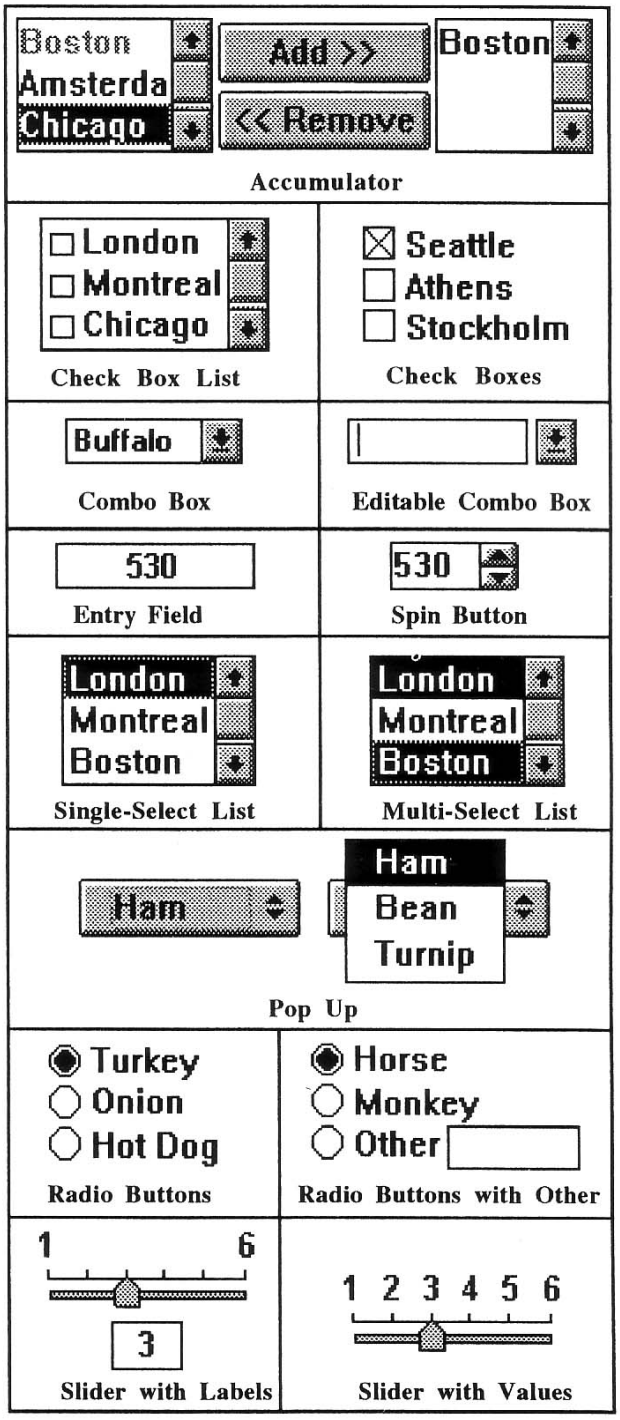


Figure 11-6. Widgets Examined

Table 11-2 shows the results with Table 11-3 providing recommendations. The full results are provided because sometimes it may not be feasible to select the first choice for a control.

Table 11-2. Results for Widget Performance and Preference

Task	Widget	Time (s)	Accuracy (%)	Preference (1=most)
Mutually exclusive	radio buttons	3.7	99	1.6
	slider (labels)	5.1	96	2.5
	pop up	5.3	99	1.8
	single select list box	5.3	96	2.3
	entry field	5.6	96	2.7
	slider (values)	7.7	97	2.7
	spin buttons/Spin box	8.0	99	3.5
	combo box	8.9	99	2.8
Non-Mutually Exclusive	check boxes	8.6	92	1.5
	multi-select list box	12.0	93	2.3
	check box list	12.5	94	2.1
	entry field	17.3	71	3.9
Select or Add	radio button with other	9.2	95	2.4
	edit combo box	18.4	91	4.2
Set a Value (in a Range)	entry field	9.3	94	2.3
	spin buttons	10.8	98	3.2
	slider (values)	13.3	95	4.2

Preference refers to mean preference score from subjects.
Source: Johnsgard, Page, Wilson, and Zeno (1995)

Guideline 11-22: For touch screen widgets, the following are recommended.

Table 11-3. Summary of Widget Recommendations

Task	Best Widget	If Space Is Tight
Mutually Exclusive	radio buttons	pop-up
Non- Mutually Exclusive	check boxes	check box list
Select or Add	radio buttons with other	editable combo box
Set a Value (in a Range)	spin button	entry field

Source: Johnsgard, Page, Wilson, and Zeno (1995)

Guideline 11-23: In general, for typing in English, a Qwerty keyboard arrangement is recommended.

The method for typing can be (1) 2-handed, as when touch typing on a desktop computer keyboard, (2) 2-finger typing (for those who have not memorized the keyboard arrangement), (3) 2-thumb typing (for smart phones), and (4) 1-finger typing (for a variety of devices). For touch typing, the Qwerty arrangement is fast because successive keystrokes are often opposite hand, so the 2 hands can work in parallel (Evey, 1980). The same is true for 2-thumb typing.

Beyond that, typing is a highly practiced task, and depending upon the age of the individual, they have probably spent thousands of hours on that task. Hence, they can more rapidly located keys to press, even for keyboards that may seem more logical.

For number pads, both the telephone (123) and calculator (789) arrangements are popular (Figure 11-7). The arrangement that leads to the most rapid entry and fewest errors depends upon the task. If the number is used for entering phone numbers, then the 123 arrangement is recommended. If the device is used as computational device, then the 789 arrangement is preferred.

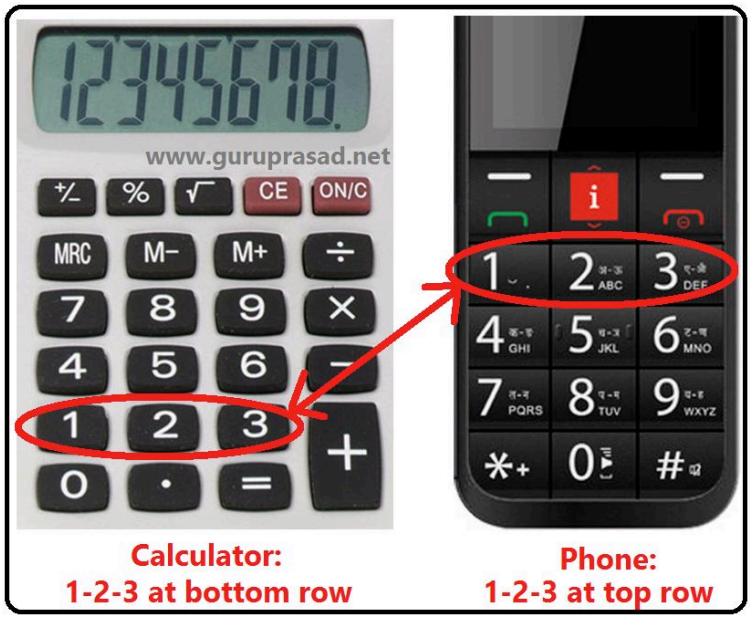


Figure 11-7. 123 and 789 keypads
Source: http://guruprasad.net/wp-content/uploads/2015/12/keypad_calc_phone.jpg

There are a number of methods that can be used for text entry such as Qwerty keyboards, Swype methods, handwriting (of single letters), and others. See MacKenzie and Soukoreff (2002). Although Swype has been found to be faster that 2-thumb typing for small devices, those evaluations have been conducted using handheld devices, not 1 outstretched hand as is the case in a motor vehicle Cuaresma and MacKenzie (2013). Data on single letter handwriting is limited.

HMI Information Systems

Menus

Menus for automotive systems are becoming increasing complex as the amount of information that needs to be presented increases. The fundamental issues are (1) how much information can be seen at any given time, (2) the depth vs. breadth tradeoff, (3) how items are assigned to menu categories and (4) how menu entries are named.

In motor vehicles, there is limited space for display and controls, so therefore both are often minimized. In theory, a menu system could consist of 2 buttons (select, next item) and a 1-line display. It would not be easy to use.

Guideline 11-24: Use computational models to select menu architectures.

As context, using a menu system involves 2 sets of sequential decisions, referred to as category matches and identity matches. A category match concerns if an item sought belongs to a certain category. For example, suppose a person was looking for information on spruce trees in a data base. The first level menu could have the options plant or animal. The user would select plant, because a spruce tree is plant. The next level could be trees, shrubs, or grasses, in which case the choice would be trees. The next category could be evergreen or deciduous. (Select evergreen.) The last choice could be pine, spruce, hemlock, cypress, or yew. The choice would be spruce, an identify match. Thus, selecting from a hierarchical menu usually involves as series of category matches, anywhere from none to several, followed by an identity match. (Note: Linnaeus' taxonomic rank (kingdom, phylum, etc.) was not used for simplicity.)

According to Miller (1981a, b), the time to make a category match is equal to a + bN, where N is the number of choices for each decision, and a and b are constants. Identify match time is c + dN, where c and d are constants and N is the number of choices. The constants a, b, c, and d are empirically determined for each HMI.

For other research on this topic, see Landauer and Nachbar (1986), Lee and MacGregor (1985), Paap and Roske-Hofstrand (1986), Norman and Chin (1988), and Norman (1991).

Guideline 11-25: The display needs to be large enough to show all of the menu options for that choice simultaneously (and sometimes more).

For example, on the Tesla Model X (version 8.1), if one wanted to turn the summons feature on or off, and one pressed the control button on the lower left of the screen, the main screen could show 1 of 2 tabs, one labeled "controls" and the other as "settings." One would need to know that summons is a setting, which may not be obvious. However, at a minimum, when first controls selection appears, the words "settings" and "controls," the next choices, should be visible.

The challenge is when the menu is broad and there are many choices. So, for the next level down, the setting choices are apps, driver profiles, units and format, vehicle, safety and security, displays, homelink, driver assistance, services and reset, at total of 9 categories. Thus, satisfying this requirement means the display needs to be large enough to show all 9 options.

Suppose the user had these 9 options, and there were 3 buttons: select, go back, and next. If the user was looking for summons, and was not sure if summons was as option or was in a category (exhaustive search), how would they respond? If only 1 option was visible, when they selected the settings category, and the first option would appear. If they did know what the options were, then they would need to review the 8 remaining options (8 button presses) and after they know what was present, move to the desired category (on average 4.5 button presses, self-terminating

search) and then select it (another button press). In these situations, reading all of the options and selecting 1 is much faster than a read select process for 1 option at a time.

Guideline 11-26: There is a tradeoff between menu breadth (how many items are shown at the same time) and depth (how many menu levels are provided), but there is not universal optimum.

For example, to select among 64 items, one could have 2 menu levels with 8 items each, 3 levels with 4 items each, or 8 levels with 2 items each. There is not simple rule for which combination is optimal as it depends on category groups that make sense, the size of the display available, and the control for selection. For motor vehicles, total eyes of road time is the primary selection factor, though mean glance time is also important. A large number of menu options at any point can lead to long mean glance times.

Guideline 11-27: Use card sorting and naming methods to develop menu grouping and category names.

In this method (Hudson, 2005), users are given the names of the items that will appear in menus and asked to sort them into groups. Often, the items are on Post-its (Figure 11-8). The users are not told how many groups are desired. After they are done, they are asked to name the groups, which become the menu categories. Usually there is no single set of groups that are used, so factor analysis is often used to identify a best set of groups.



In a project to design a new digital phone service, a card -sorting exercise enable potential users to influence the final menu structure and naming.

Figure 11-8. Photo of card sorting
Source: https://image.slidesharecdn.com/1-091103083111-phpapp01/95/ideo-method-card-14-728.jpg?cb=1257237126

Guideline 11-28: Menu category names should be distinct and indicate to what they refer.

User suggested names are a starting point, but often the names they suggest reflect the content of the category, which may not effectively distinguish between categories. This can lead to uncertainty about in which category a menu item belongs.

Guideline 11-29: Consider using examples of category content to help with category naming

One way to aid the user is to provide the name of at least 1 item in the category as an example after the category name. The evidence from desktop web site design (where screen space is considerable) suggests that up to 3 examples can be helpful, especially if there is uncertainty about the content of a category. Sufficient screen space for this purpose may not be available in a motor vehicle.

Guideline 11-30: The need to have a home page varies with the system.

For example, for center console display pages, the driver may wish to show a page for navigation or climate control or entertainment. For other systems in the center display, there may need to have a default home page.

Guideline 11-31: For now, use 11 s as a reasonable worst case for takeover (AH transfer) reaction time and 8 s as a reasonable worst case for handback reaction time.

Consistent with the prior table, there are 2 situations to consider, (1) takeover from the vehicle (when the automation fails or cannot deal with a situation, AH transfer) and (2) handback (when the driver transfers control to the vehicle (manual to automation, HA transfer). Ordinarily, there is no time pressure for handback, but there may be such for take over.

Of the many studies in the literature, Eriksson and Stanton (2017) provides the best summary of current knowledge and it will be the primary source for making HMI design recommendations.

But what is meant by takeover (AH) time and handback (HA transfer) time? For takeover time, the onset could be when an alert to takeover begins. What is the end point – when the driver looks at a visual warning, when they are in the seat to respond, when they first touch the steering wheel or foot pedals, when some to be specified level of speed control and lateral position control is achieved, or it is something else? Ericksson and Stanton define Take-Over-Request lead time (TORlt: the lead-time from a ‘take-over request’ (TOR) to a critical event, such as a stranded vehicle) and Take Over reaction time (TOrt: the time is takes the driver to take back control of the vehicle from the automated system when a TOR has been issued). Thus, it appears that TORlt is the time from the onset of a takeover request until some indication of driver response has begun. Take Over reaction time appears to be from when that initial indication of a response has begun (the end of TORlt) until the driver is in full control of the vehicle.

Figure 11-9 shows the distribution of mean takeover-request lead times from the 25 studies Erikson and Stanton reviewed. There are multiple values for several studies because multiple conditions were reported. For takeover lead time, the mean response time was 6.4 s (+/- 5.4 s), but as shown in the figure, the mean was strongly influenced by times of 30 s in 1 study.

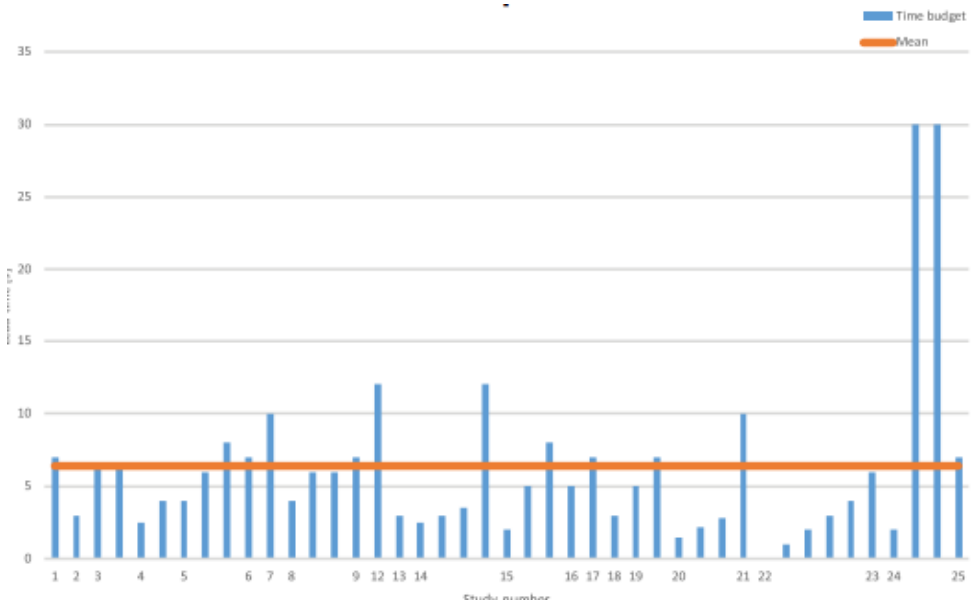


Figure 11-9. Takeover (AH Transfer) Lead Times from Erikson and Stanton (2017), page 692

Figure 11-10 shows the mean takeover reaction times from the same 25 studies, with at mean of 3.0 s (+/-2.0 s). Again, note that the mean was strongly influenced by very long times in 4 studies.

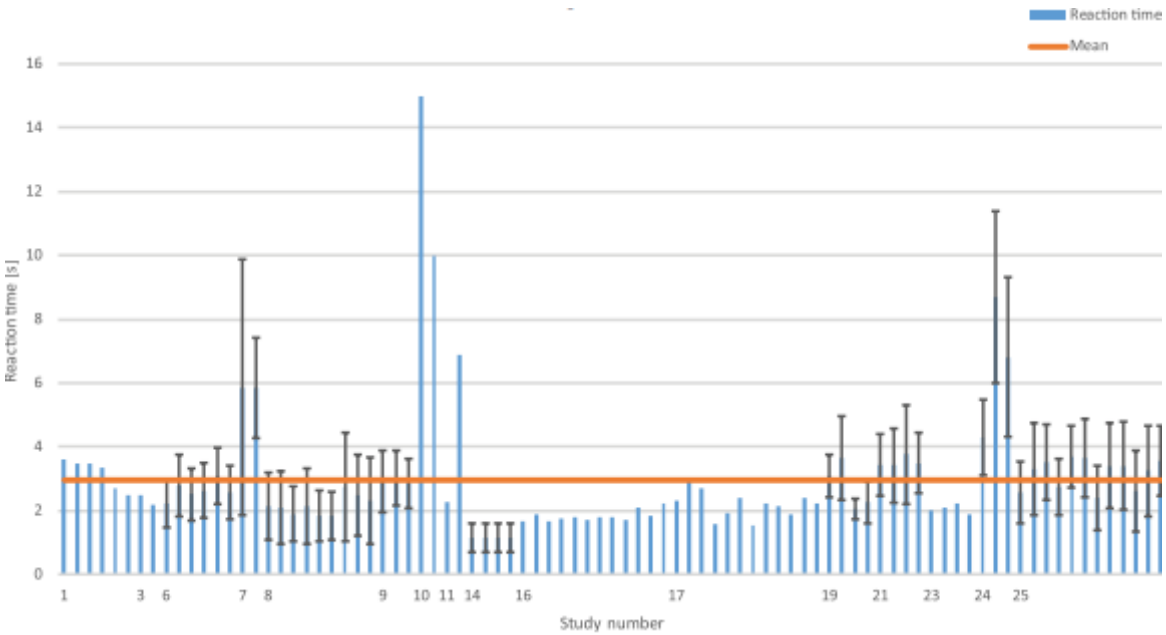


Figure 11-10. Takeover Reaction Times from from Erikson and Stanton (2017), page 692.

Readers are cautioned against using the mean value. If the data were normally distributed, then the mean and median would be the same, and half of the data points would be greater than the mean. Designing to satisfy only half to the situations of interest is not a good idea. Pending further review of the studies summarized by Eriksson and Stanton (2017), the author suggests using 11.8 s (=6.5 + 5.4) for a reasonable worst cast takeover lead time and 5.0 s (=3.0 + 2.0) for a takeover reaction time. Needing further examination are the specific conditions that lead to the times reported and the development of an expression to predict take over times as a function of test conditions.

In addition, Eriksson and Stanton (2017) experimentally examined both takeover and handback times for subjects driving on a highway in a driving simulator at 70 mi/hr, in some cases while they were reading an issue of the National Geographic (a secondary task). There was a visual cue along with a voice (either "Please resume control" or "Automation available" when transfer was to occur. Note that there was no urgency to either request. Subjects were ages 20-52.

Figure 11-11 shows the experimental results along with a distribution summarizing the prior literature. Takeover was determined by examining the value of the standard deviation of angular rate. Each of the 2 curves for the experiment represents the smoothed distribution of 312 data points. Notice that the experimental values are much larger than those reported in the literature, in part because there was not urgency to the takeover request. In addition, also notice that the times are log-normally distributed as one would expect. Finally, note the substantial effect of the particular secondary task examined in this experiment.

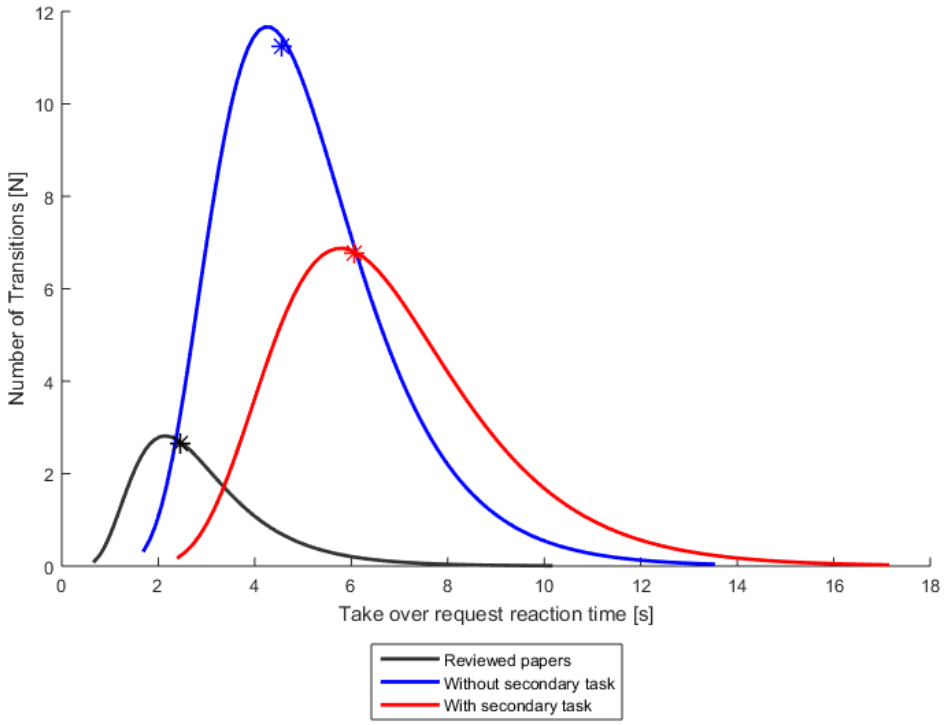


Figure 11-11. Takeover (AH Transfer) Reaction Times (s)
Note: * indicates the median value.

Figure 11-12 shows the experimental results for transfer in the opposite direction, from the driver to automation. Again, the data are log-normally distributed, but the effect of the secondary task is minor at best. For comparison, the x-axes of these 2 figures are similar.

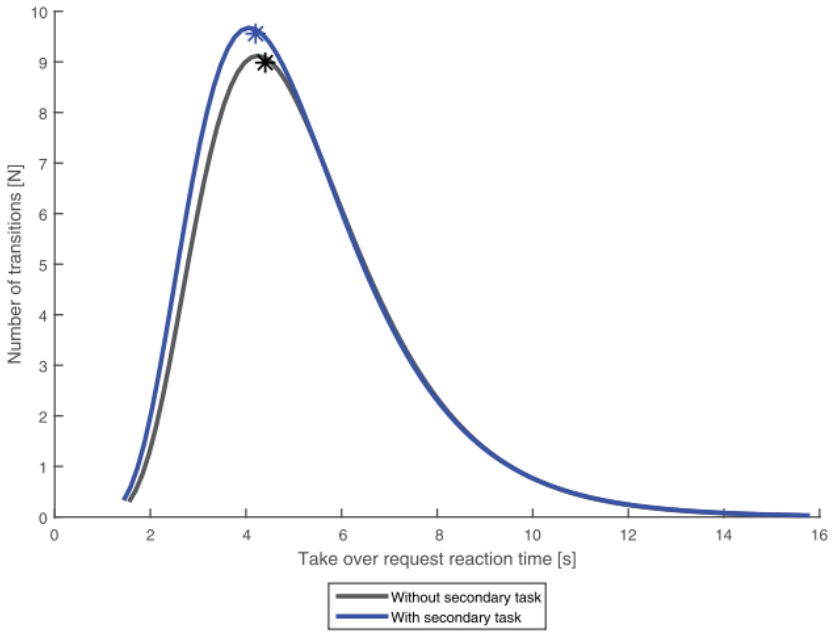


Figure 11-12. Distribution of Handover (HA Transfer) Reaction Times (s) (to automation)

To aid in comparison, Table 11-4 shows summary statistics for the data presented so far. Median values are used to partial resolve difficulties due to the log-normally distributed data. The original data are reported to the nearest millisecond. Given the nature of the tasks examined in the literature and the data collection process, and the need to focus on major differences, times are reported here to the nearest 0.1 s.

Table 11-4. Reaction Time Summary Statistics for Takeover and Handback

Statistic	**Metareview**	**Automation to Human (Takeover)**		**Human to Automation (Handback)**	
		No 2nd Task	**2nd Task**	**No 2nd Task**	**2nd Task**
median	2.5	4.6	6.1	4.2	4.4
IQR	1.4	1.6	2.4	2.0	1.8
minimum	1.1	2.0	3.2	2.8	2.9
maximum	15.0	25.7	21.0	23.9	23.2

Source: Eriksson and Stanton (2017), p. 697.

What values should be used for design? A simple approximation, ignoring the lognormal nature of the distributions, would be to set the threshold at the median + 2*IQC. For automated to manual (takeover), the reaction times would be 7.8 s (no secondary task) and 10.9 s (secondary task). For manual to automatic (handback), the reaction times are 8.2 (no secondary task) and 8.0 with a secondary task. These estimates are probably going well beyond what the data allows.

Notice that for takeover reaction times, the threshold with a secondary task is less than that without. In the previously shown figures, their times correspond to very few data points. Finally, they ignore log-normal nature of the distributions and many factors that affect these distributions that were identified by Eriksson and Stanton. Thus, even though there are many studies of takeover and handback times, the absence of experimental details (in a few cases) and distribution information makes it difficult to set exact times for design thresholds. Further examination of the existing data is warranted as well as new research (e.g., Lu, Coster, and deWinter, 2017).

Another set of estimates come from Gold, Happee, and Bengler (2017) who used regression modelling of 5 related experiments. In those experiments, subjects were reacting to a scenario in which they needed to change lanes to take over and change lanes from the middle lane (of 3 lanes) to avoid a blocking vehicle. The time available to respond varied from 5 to 7.8 s, if subjects were engaged in a secondary task (or not, if and so which one) varied, as well as traffic density. Using regression analysis, they modeled takeover time which here was defined as the time between the takeover request (probably the onset, but it is not certain) until the start of the maneuver (2 degree change in steering wheel angle or a 10% brake pedal actuation). To reflect the actual predictive capability of the model, the equation has been rounded off to 3 digits.

$$\text{Takeover time (s)} = 2.07 + 0.33TB - 0.15(\text{Lane} - 1.94)^2 - 0.006(TD - 15.7)^2 - 0.57*\ln(\text{Repetition}) + 2.12\times10^{-4}(\text{Age} - 46.2)^2$$

where:
TB = Time Budget (s, range of 5 to 7.8 s)
Lane = Lane number (1=left, 2=center, 3=right)
TD = Traffic Density (vehicles / kilometer, range of 0 to 30)
Repetition = number of time each scenario is experienced by each subject (range 2 to 20)
Age= Driver Age (years, range 19 to 79)

Note that a number of factors in the model are quadratic as plots of those factors versus takeover time revealed such.

Whether people can actually respond in time to the events simulated is an open question. For example, many on-board sensors available such as Lidar have a maximum range of about 120 meters (about 394 feet). At 70 mi/hr (about 102 feet/s), the posted speed on many expressways in the U.S., the driver has about 3.9 s to respond, somewhat less than the range of the model explored. A more modest speed of 55 mi/hr is 81 ft/s. In that situation, the time available is 4.9 s, just below the minimum time budget examined by Gold, Happee, and Bengler (2017). This calculation is important because if the vehicle relies on onboard sensors only, and the driver is truly not attentive, they may not be able to respond to an immediate and unexpected situation.

However, to provide an example, for a 5 s time budget, a driver in the center lane, a relatively moderate traffic density (25), 3 repetitions, and a 50-year-old driver, the takeover time is approximately 4.1 s.

Keep in mind that all of these calculations assume are reasonably alert driver sitting in the driver's seat and facing forward, which may not be the case in the future. There data suggest the

need to forewarn the driver well in advance to prepare to take over, such as when approaching a construction zone.

NEXT STEPS

Other than the CAMP project and the Naujoks guidelines, there are no published guidelines for Level 1-3 automated vehicle design. This report fills identifies important gaps in what is known about the use of automation, in particular problems of situation awareness and mode switching, and broadly describes methods for addressing those problems, though there are many unknowns about exactly how to proceed.

To take these guidelines to the next level is several years of work. Here is what is to be done.

1. One of the first next steps should be a joint review of the decisions made to date in the design of the Clinic 2 interface to identify how each decision was made. This should be done in the context of creating a paper to summarize the process. One of the goals is to determine which guidelines were used most often as well where decisions were made, but there is not a guideline in the current set (which should be added).

2. Many of the guidelines do not cite specific research support them. Although some guidelines are only based on principles, the extent to which literature bears on each principle, that literature should be cited, explained, and used to enhance the principles provided. For example, one of the principles recommends the use of QWERTY keyboards in general, but how does that recommendation vary as function of the space available? As the application of these guidelines is global, how does that recommendation change when non-Roman characters are to be entered? To provide a sense of the scope of work, just addressing that 1 point could be weeks of work

3. For many of the guidelines listed, there is either no research or minimal research to bear. For example, how should vehicle confidence (in lane marking, object detection, path selection, etc.) be shown? Each one of these topics could be the focus of 1 or more experiments.

4. There are a number of guidelines that concern how to represent inconsistencies in task methods. For this topic, it is not mere a matter of conducting experiments, by the development or identification of a theory that will lead to the creation of alternatives to be compared.

5. A major topic needing further thought was predicting the complexity of an interface. Clearly, the literature provides indications of what the potential predictors could be – the number of lines of code, Cyclometric Complexity, the number of operators and operands, the number of rules, etc. However, what is need is further thought about what representation makes the most sense, in particular some that is easy to implement. The complex tools that researchers use to explain cognitive processes are interesting tools for research, but not practical for day-to-day user interface engineering.

6. In several cases, topics are identified as being relevant, but other than providing a few references as a starting point, the topics are unexplored (e.g., the complexity of a visual representation). They should be explored and guidelines relevant to them added.

7. At some point in the process, there needs to be a review of the guidelines as a whole, to identify which guidelines overlap and/or are redundant, and where there are gaps, topics that are not covered by guidelines but should be covered.

8. Finally, the optimal process for enhancing the guidelines is one where the research team has continuing involvement in the development of new interfaces. In that manner, there is an ongoing focus of translating research into practice and identifying new issues as they arise. For example, at the current time, there are no specific guidelines for augmented reality displays, and yet manufacturers wanting to offer state-of-the-art vehicle will want to offer them. Those displays need to be easy to use, and accordingly, guidelines for them are needed.

What is suggested here is a set of substantial next steps that will take several years to complete. This should not be a quarter by quarter project as it can be most efficiently conducted by a team focused on it with a long-term vision that is stable over time. That stability requires a commitment from the sponsor. Considering the potential product implications and the investment in the technology to achieve automation, the relative cost is not high.

REFERENCES

Alliance of Automobile Manufacturers (June 26, 2006). *Statement of Principles, Criteria and Verification Procedures on Driver Interactions with Advanced In-Vehicle Information and Communication Systems*, Washington, D.C.: Alliance of Automobile Manufacturers.

Aloysius, A., and Maheswaran, K. (2015). A review on component based software metrics. *International Journal of Fuzzy Mathematical Archive, 7*(2), 185-194.

American Association of Medical Instrumentation (2009). *ANSI/AAMI HE75, Human Factors Engineering-Design of Medical Devices*, Arlington, VA: American Association of Medical Instrumentation.

American Association of Medical Instrumentation (2015). *ANSI/AAMI/IEC 62366-1:2015 -- Medical Devices - Application of Usability Engineering to Medical Devices*, Arlington, VA: American Association of Medical Instrumentation.

American Society for Testing and Materials (2013). *ASTM F1166 - 07 Standard Practice for Human Engineering Design for Marine Systems, Equipment, and Facilities*, West Conshohocken, PA: American Society for Testing and Materials.

American Society for Testing and Materials (2015). *ASTM F1337 - 10 Standard Practice for Human Systems Integration Requirements for Ships and Marine Systems, Equipment, and Facilities*, West Conshohocken, PA: American Society for Testing and Materials.

Ayama, M., Ujike, H., Iwai, W., Funakawa, M., and Okajima, K. (2007). Effects of contrast and character size upon legibility of Japanese text stimuli presented on visual display terminal, *Optical Review*, *14*(1), 48-56.

Banerjee, S. S., Jha, S., Cyriac, J., Kalbarczyk, Z. T., and Iyer, R. K. (2018, June). Hands off the wheel in autonomous vehicles?: A systems perspective on over a million miles of field data. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 586-597, New York, NY: IEEE.

Blain (2017, June 8). Self-driving vehicles: What are the six levels of autonomy? https://newatlas.com/sae-autonomous-levels-definition-self-driving/49947/, retrieved January 9, 2018.

Boreczky, J., Green, P., Bos, T., and Kerst, J. (1988). *Effects of Size, Location, Contrast, Illumination, and Color on the Legibility of Numeric Speedometers* (technical report UMTRI-88-36), Ann Arbor, Michigan: University of Michigan Transportation Research Institute.

Bostrom, A., and Ramstrom, F. (2014). *Head-Up Display for Enhanced User Experience: Infotainment Designed Around You* (master's thesis), Gothenburg, Sweden: Chalmers University of Technology, Department of Applied Information Technology.

Bovair, S., Kieras, D. E., and Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. *Human-Computer Interaction*, *5*(1), 1-48.

Brown, B., & Laurier, E. (2017, May). The trouble with autopilots: Assisted and autonomous driving on the social road, *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, New York, NY: Association for Computing Machinery, 416-429.

Burnett, G. (2000a). 'Turn Right at the Traffic Lights': The requirement for landmarks in vehicle navigation systems, *The Journal of Navigation*, *53*(3), 499-510.

Burnett, G. E. (2000b). Usable vehicle navigation systems: Are we there yet? paper presented at *Vehicle Electronic Systems 2000 Conference*, 3.1.1-3.1.11.

Burnett, G., Smith, D., and May, A. (2001). Supporting the Navigation Task: Characteristics of 'Good' Landmarks, *Contemporary Ergonomics*, *1*, 441-446.

Cai, D., Chi, C. F., and You, M. (2001). The legibility threshold of Chinese characters in three type styles, *International Journal of Industrial Ergonomics*, *27*(1), 9-17.

Cai, H., Green, P. A., and Kim, J. J. (2013). Estimating the legibility of a single letter E viewed at different display angles, *Applied Ergonomics*, *44*(4), 575-587.

Campbell, J. L., Brown. J. L., Graving, J. S., Richard, C. M., Lichty, M. G., Sanquist, T., Bacon, P., Woods, R., Li, H., Williams, D.N., and Morgan, J. L. (2016, December). *Human Factors Design Guidance for Driver-Vehicle Interfaces* (technical report DOT HS 812 360). Washington, DC: U.S. Department of Transportation, National Highway Traffic Safety Administration.

Campbell, J. L., Hoffmeister, D. H., Kiefer, R. J., Selke, D. J., Green, P., and Richman, J. B. (2004). *Comprehension Testing of Active Safety Symbols* (SAE paper 2004-01-0450), Warrendale, PA: Society of Automotive Engineers.

Campbell, J.L., Carney, C., and Kantowitz, B.H. (1997). *Human Factors Design Guidelines for Advanced Traveler Information Systems (ATIS) and Commercial Vehicle Operations (CVO)* (technical report FHWA-RD-98-057), Washington, D.C.: U.S. Department of Transportation, Federal Highway Administration.

Card, S.K, Moran, T.P. and Newell, A. (1983). *The Psychology of Human-Computer Interaction*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Carney, C., Campbell, J.L., and Mitchell, E.A (1998). *In-Vehicle Display Icons and Other Information Elements: Literature Review* (technical report FHWA-RD-98-164), Washington, D.C.: U.S. Department of Transportation, Federal Highway Administration.

Cassano-Piche, A. and Green, P. (2017). *Design Process and Automated Vehicle Driver Interfaces: An Outline Summary of Key Articles*, Toronto, CA: Human Factors North.

Chapanis, A. (1960). *On Some Relations Between Human Engineering, Operations Research and Systems Engineering* (technical Report 8), Baltimore, MD: Psychological Laboratory, The Johns Hopkins University.

Chhillar, R. S., and Gahlot, S. (2017, August). An evolution of software metrics: A review. In *Proceedings of the International Conference on Advances in Image Processing*, 139-143. ACM.

Chi, C. F., Cai, D., and You, M. (2003). Applying image descriptors to the assessment of legibility in chinese characters, *Ergonomics*, *46*(8), 825-841.

Chidamber, S. R., and Kemerer, C. F. (1994). A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, *20*(6), 476-493.

Coleman, D., Ash, D., Lowther, B., and Oman, P. (1994). Using metrics to evaluate software system maintainability. *Computer*, *27*(8), 44-49.

Colle, H. A., and Hiszem, K. J. (2004). Standing at a kiosk: Effects of key size and spacing on touch screen numeric keypad performance and user preference, *Ergonomics*, *47*(13), 1406-1423.

Commission of the European Communities (1999). *Statement of Principles on Human Machine Interface (HMI) for In-Vehicle Information and Communication Systems (EU Principles),* (Annex 1 to Commission Recommendation of 21 December 1999 on safe and efficient in-vehicle information and communication systems: A European statement of principles on human machine interface) Brussels, Belgium: European Union.

Commission of the European Communities (2007). *Commission Recommendation on Safe and Efficient In-Vehicle Information and Communication Systems: Update of the European Statement of Principles on Human Machine Interface.* Brussels, Belgium: European Union.

Cornog, D. Y., and Rose, F. C. (1967). *Legibility of Alphanumeric Characters and Other Symbols. 2. A Reference Handbook* (report NBS-MISC-262-2), Washington, D.C.: National Bureau of Standards, Institute for Applied Technology.

Cuaresma, J., and MacKenzie, I. S. (2013). A study of variations of Qwerty soft keyboards for mobile phones, *Proceedings of the International Conference on Multimedia and Human-Computer Interaction-MHCI*, 126.1-126.8.

Curtis, B., Sheppard, S. B., Milliman, P., Borst, M. A., and Love, T. (1979). Measuring the psychological complexity of software maintenance tasks with the Halstead and McCabe metrics. *IEEE Transactions on Software Engineering*, (2), 96-104.

Damouth, D. and Green, P. (1997). *Influence of Warning Message Content on Message Understandabilty and When Drivers Respond* (technical report UMTRI-97-22). Ann Arbor, Michigan: University of Michigan Transportation Research Institute.

Degani, A., and Heymann, M. (2002). Formal verification of human-automation interaction. *Human Factors*, *44*(1), 28-43.

Degani, A., Heymann, M., and Shafto, M. (1999, September). Formal aspects of procedures: The problem of sequential correctness. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, San Monica, CA: Human Factors and Ergonomics Society, 1113-1117.

Dikmen, M. and Burns, C.M., (2016). Autonomous driving in the real world: Experiences with tesla autopilot and summon. *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* New York, NY: Association for Computing Machinery, 225-228. ACM.

Ehrenreich, S. L. (1985). Computer abbreviations: Evidence and synthesis, *Human Factors*, *27*(2), 143-155.

Elias, B., and Brenner, C. (2005). Automatic generation and application of landmarks in navigation data sets, *Developments in Spatial Data Handling*, Springer, Berlin, Heidelberg, 469-480.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, *37*(1), 32-64.

Endsley, M. R. (2016). *Designing for Situation Awareness: An Approach to User-centered Design*. Boca Raton, FL: CRC press.

Endsley, M.R. (2017). Autonomous driving systems: a preliminary naturalistic study of the Tesla Model S. *Journal of Cognitive Engineering and Decision Making*, *11*(3), 225-238.

Endsley, M.R. and Garland, D.J. (eds.), (2000). *Situation Awareness Analysis and Measurement*. Boca Raton, FL: CRC Press.

Endsley, M.R. and Jones, D. (2012). Applying SA-Oriented Design to Complex Systems (chapter 15), in Endsley, M.R., (2016). *Designing for Situation Awareness: An Approach to User-Centered Design*. Boca Raton, FL: CRC Press.

Endsley, M. R. and Kaber, D.B. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task, *Ergonomics*, *42*(3), 462-492.

Engström, J., and Mårdh, S. (2007, January). *SafeTE Final Report* (technical report 2007:36), Borlänge, Sweden: Swedish Road Agency.

Eriksson, A., and Stanton, N. A. (2017). Takeover time in highly automated vehicles: Noncritical transitions to and from manual control, *Human Factors*, *59*(4), 689-705.

Evey, R.J. (1980). *How Typists Type* (IBM technical report HFC-25), San Jose, CA: IBM.

Favarò, F. M., Nader, N., Eurich, S. O., Tripp, M., and Varadaraju, N. (2017). Examining accident reports involving autonomous vehicles in California. *PloS one*, *12*(9), e0184952.

Favarò, F., Eurich, S., and Nader, N. (2018). Autonomous vehicles' disengagements: Trends, triggers, and regulatory limitations. *Accident Analysis & Prevention*, *110*, 136-148.

Federal Aviation Administration, (2012). *The Human Factors Design Standard* (HF-STD-001), Atlantic City, NJ: Federal Aviation Administration, http://hf.tc.faa.gov/hfds/download-hfds/.

Fitts, P.M. (1951). *Human Engineering for an Effective Air Navigation and Traffic Control System* (technical report), Washington, D.C.: National Research Council.

Gasser. T.M. (2014). Vehicle automation: Definitions, legal aspects, research needs (PowerPoint presentation*), UNECE -Workshop: German Federal Highway Research Institute Towards a new transportation culture: technology innovations for safe, efficient and sustainable mobility* 17 – 18 November 2014, Brussels, Belguim, https://www.unece.org/fileadmin/DAM/trans/events/2014/Joint_BELGIUM-UNECE_ITS/07_ITS_Nov2014_Tom_Gasser__BASt.pdf, retrieved December 2, 2018.

George, K., Green, P., and Fleming, J., (1995). *Timing of Auditory Route Guidance Instructions*, (technical report UMTRI-95-6), Ann Arbor, MI: The University of Michigan Transportation Research Institute (available from NTIS as PB-96-187190).

Gish, K.W. and Staplin, L. (1995). *Human Factors Aspects of Using Head Up Displays in Automobiles: A Review of the Literature* (technical report DOT HS 808 320), Washington, D.C: U.S. Department of Transportation, National Highway Traffic Safety Administration.

Godsmark, P. (2017, April 10). *The Definitive Guide to the Levels of Automation for Driverless Cars*, https://driverless.wonderhowto.com/news/definitive-guide-levels-automation-for-driverless-cars-0176009/, retrieved January 9, 2017.

Gold, C., Happee, R., and Bengler, K. (2017, in press). Modeling take-over performance in level 3 conditionally automated vehicles, *Accident Analysis & Prevention* (on line).

Gould, J. D., and Lewis, C. (1985). Designing for usability: Key principles and what designers think, *Communications of the ACM*, *28*(3), 300-311.

Green, P. (1996). Development of easy–to–Understand Vehicle Malfunction Warnings, *International Journal of Vehicle Design*, *17*(1), 27-39.

Green, P. (2017). *Human Factors for Connected and Automated Vehicles: Lessons Learned About Methods from Previous Field Operational Tests* (technical report UMTRI-2016-xx), Ann Arbor, MI: University of Michigan Transportation Research Institute.

Green, P. (2018a, in press). *Development of Human Factors and Automotive Standards Curricula Materials for the University of Michigan and Beyond*, (technical report UMTRI–2018–**), Ann Arbor, MI: University of Michigan Transportation Research Institute.

Green, P. (2018b, in press). *Getting Research into Practice via Standards: Standards 101* (technical report UMTRI-2018-**), Ann Arbor, MI: University of Michigan Transportation Research Institute.

Green, P. and George, K. (1995). When should auditory guidance systems tell drivers to turn? *Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting*, Santa Monica, CA: Human Factors and Ergonomics Society, 1072-1076.

Green, P., George, D., Cheydleur, G., and Jan, J. (2017). *Core Design and Evaluation Guidelines and Standards for Automotive HMIs* (technical report UMTRI 2017-xx), Ann Arbor, MI: University of Michigan Transportation Research Institute.

Green, P., Goldstein, S., Zeltner, K., and Adams, S (1988). *Legibility of Text on Instrument Panels: A Literature Review* (technical report UMTRI-88-34), Ann Arbor, MI: University of Michigan Transportation Research Institute.

Green, P., Levison, W., Paelke, G., and Serafin, C. (1993). *Preliminary Human Factors Guidelines for Driver Information Systems* (technical report UMTRI-93-21), Ann Arbor, MI: University of Michigan Transportation Research Institute (also published as FHWA-RD-94-087, McLean, VA: U.S. Department of Transportation, Federal Highway Administration, December, 1995).

Green, P., Paelke, G., and Boreczky, J. (1992). The "Potato Head" Method for identifying driver preferences for vehicle controls, *International Journal of Vehicle Design*, *13*(4), 352-364.

Green, P., and Pew, R. W. (1978). Evaluating pictographic symbols: An automotive application, *Human Factors*, *20*(1), 103-114.

Green, P. and Rudin-Brown, C.M. (2017). *Expert Review of Automated Vehicle Driver Interfaces: The Automation Literature* (technical report 2017-xx), Ann Arbor, Michigan: University of Michigan Transportation Research Institute.

Green, P. and Rudin-Brown (2018a). *The Find, Perceive, Decide, Execute (FPDE) Method for Assessing the Usability of Connected and Automated Vehicle HMIs* (technical report UMTRI-2018-**), Ann Arbor, MI: University of Michigan Transportation Research Institute.

Green, P. and Rudin-Brown (2018b). *The Find, Perceive, Decide, Execute (FPDE) Method: Examination of 3 Partially-Automated Vehicles* (technical report UMTRI-2018-**), Ann Arbor, MI: University of Michigan Transportation Research Institute.

Green, P. and Rudin-Brown, C.M. (2017). *Expert Review of Automated Vehicle Driver Interfaces: The Automation Literature* (technical report), Ann Arbor, MI: University of Michigan Transportation Research Institute.

Guyomarc'h, J. Y., and Guéhéneuc, Y. G. (2005, July). On the impact of aspect-oriented programming on object-oriented metrics. In *Proceedings of the 9th ECOOP Workshop on Quantitative Approaches to Object-Oriented Software Engineering (QAOOSE 2005),* Glasgow, UK.

Haines, G., Carney, D., and Foreman, J. (1997). Component-based software development/cots integration. *SEI Technology Description. Pittsburgh PA: Software Engineering Institute, Carnegie Mellon University*

Hall, A. D., Cunningham, J. B., Roache, R. P., and Cox, J. W. (1988). Factors affecting performance using touch-entry systems: Tactual recognition fields and system accuracy. *Journal of Applied Psychology*, *73*(4), 711-720.

Halstead, M. H. (1977). *Elements of Software Science*, New York, NY: Elsevier.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, *8*(3), 231-274.

Harel, D., and Naamad, A. (1996). The STATEMATE semantics of statecharts. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, *5*(4), 293-333.

Harrison, R., Counsell, S. J., and Nithi, R. V. (1998). An evaluation of the MOOD set of object-oriented software metrics. *IEEE Transactions on Software Engineering*, *24*(6), 491-496.

Henry, S., Kafura, D., and Harris, K. (1981). On the relationships among three software metrics. *ACM SIGMETRICS Performance Evaluation Review*, *10*(1), 81-88.

Heymann, M., and Degani, A. (2007). Formal analysis and automatic generation of user interfaces: Approach, methodology, and an algorithm. *Human Factors*, *49*(2), 311-330.

Heymann, M., and Degani, A. (2013, October). Automated driving aids: modeling, analysis, and interface design considerations. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, New York, NY: ACM, 142-149.

Howett, G. L. (1983). *Size of Letters Required for Visibility as a Function of Viewing Distance and Observer Visual Scuity* (report NBS TN 1180), Washington, D.C.: National Bureau of Standards.

Hudson, W. (2005). Playing your cards right: Getting the most from card sorting for navigation design, *Interactions*, *12*(5), 56-58.

Hugo, J., and Gertman, D. (2013). A qualitative method to estimate HSI display complexity. *Nuclear Engineering and Technology*, *45*(2), 141-150.

Human Factors and Ergonomic Society (2007). *ANSI/HFES 100-2007 Human Factors Engineering of Computer Workstations*, Santa Monica, CA: Human Factors and Ergonomics Society.

Human Factors and Ergonomic Society (2007). *ANSI/HFES 300 Guidelines for Using Anthropometric Data in Product Design*, Santa Monica, CA: Human Factors and Ergonomics Society.

International Electrotechnical Commission (2015). *Medical Devices Part 1: Application of Usability Engineering to Medical Device*s (IEC standard 62366), Geneva, Switzerland: International Electrotechnical Commission.

Human Factors and Ergonomic Society (2008). *ANSI/HFES 200 Human Factors Engineering of Software User Interfaces*, Santa Monica, CA: Human Factors and Ergonomics Society.

International Organization for Standardization (2002). *Road vehicles – Ergonomics Aspects of Transport Information and Control Systems – Dialogue Management Principles and Compliance Procedure* (ISO Standard 15005:2002). Geneva, Switzerland.

International Organization for Standardization (2003). *Road Vehicles – Ergonomics Aspects of Transport Information and Control Systems (TICS) – Procedures for Assessing Suitability for Use while Driving* (ISO Standard 17287:2003). Geneva, Switzerland.

International Organization for Standardization (2004). *Road Vehicles – Ergonomics Aspects of Transport Information and Control Systems (TICS) – Procedures for Determining Priority of On-board Messages Presented to Drivers* (ISO Technical Specification 16951:2004). Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization (2004). *Road Vehicles – Location of Hand Controls, Indicators, and Tell-tales in Motor Vehicles)* ISO standard 4040, Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization (2004). *Road Vehicles – Passenger Cars – Driver Hand Control Reach)* ISO standard 3858, Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization (2007). *Road Vehicles – Ergonomics Aspects of Transport Information and Control Systems – Occlusion Method to Assess Visual Demand Due to the Use of In-vehicle Systems* (ISO Standard 16673:2007). Geneva, Switzerland.

International Organization for Standardization (2009). *Road Vehicles – Ergonomics aspects of transport information and control systems – Specifications for In-vehicle Auditory Presentation* (committee draft standard 15006). Geneva, Switzerland.

International Organization for Standardization (2010). *Road vehicles – Symbols for Controls, Indicators and Tell-tales* (ISO Standard 2575). Geneva, Switzerland: International Organization for Standardization.

International Organization for Standardization (2011). *Road vehicles – Ergonomics Aspects of Transport Information and Control Systems – Integration of Safety Critical and Time Critical Warning Signals: Introduction of Avoiding Conflicts* (technical report 12204), Geneva, Switzerland.

International Organization for Standardization (2011). *Road Vehicles – Measurement of Driver Visual Behaviour with Respect to Transport Information and Control Systems – Part 1: Definitions and Parameters* (committee draft standard 15007-1:2011). Geneva, Switzerland.

International Organization for Standardization (2011). *Road Vehicles – Measurement of Driver Visual Behaviour with Respect to Transport Information and Control Systems – Part 2: Equipment and Procedures* (ISO Draft Technical Specification 15007-2:2011). Geneva, Switzerland.

International Organization for Standardization (2014). *Road Vehicles – Ergonomics Aspects of Transport Information and Control Systems – Specification and Test Procedure for In-vehicle Visual Presentation* (committee draft standard 15008:2014). Geneva, Switzerland.

International Organization for Standards (2008). *Graphical symbols -- Test methods -- Part 2: Method for Testing Perceptual Quality* (ISO standard 9186:2), Geneva, Switzerland: International Organization for Standardization.

International Standards Organization (2010). *Ergonomics of Human-System Interaction —* Part 210: Human-centred design for interactive systems (ISO 9241-210), Geneva, Switzerland, International Standards Organization.

International Standards Organization (2011). *Ergonomics of Human System Interaction -- Part 420: Selection of physical input devices* (ISO 9241-420), Geneva, Switzerland, International Standards Organization.

International Organization for Standards (2014a). *Graphical Symbols -- Test Methods -- Part 1: Method for Testing Comprehensibility* (ISO standard 9186:1), Geneva, Switzerland: International Organization for Standardization.

International Organization for Standards (2014b). *Graphical Symbols -- Test Methods -- Part 3: Part 3: Method for Testing Symbol Referent Association* (ISO standard 9186:3), Geneva, Switzerland: International Organization for Standardization.

International Standards Organization (2018). *Ergonomics of Human-System Interaction — Part 11: Usability: Definitions and Concepts* (ISO 9241-11), Geneva, Switzerland, International Standards Organization.

Jang, I., Kim, C. H., Jung, W., and Kim, H. J. (2016). Estimation of computer-based display complexity in digitalized MCR of NPPs, *Proceedings of the KNS 2016 Autumn Meeting.*

JAMA (Japan Automobile Manufacturers Association). *Guideline for In-Vehicle Display Systems*, version 2.1, version 3.0, Tokyo, Japan: Japan Automobile Manufacturers Association.

Jeong, H. and Green, P. (2013). *SAE and ISO Standards for Warnings and Other Driver Interfaces Elements: A Summary* (technical report UMTRI-2013-16), Ann Arbor, MI: University of Michigan Transportation Research Institute.

Jerome, C., Monk, C., and Campbell, J. (2015). Driver vehicle interface design assistance for vehicle-to-vehicle technology applications (paper 15-0452), *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV),* Washington, D.C.: U.S. Department of Transportation, National Highway Traffic Safety Administration.

Johnsgard, T. J., Page, S. R., Wilson, R. D., and Zeno, R. J. (1995, October). A comparison of graphical user interface widgets for various tasks, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Santa Monica, CA: Human Factors and Ergonomics Society, 287-291.

Kaber, D. B., and Endsley, M. R. (2004). The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task, *Theoretical Issues in Ergonomics Science*, *5*(2), 113-153.

Katzmarski, B., and Koschke, R. (2012, June). Program complexity metrics and programmer opinions. In *2012 IEEE 20th International Conference on Program Comprehension (ICPC),* 17-26. New York, NY: IEEE.

Kieras, D. (1988). Towards a practical GOMS model methodology for user interface design. In Martin Helander. *Handbook of Human-Computer Interaction*. Amsterdam, The Netherlands: Elsevier Science Publishers, 135–157.

Kieras, D. E., and Polson, P. G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*, *22*(4), 365-394.

Kiger, J. I. (1984). The depth/breadth trade-off in the design of menu-driven user interfaces, *International Journal of Man-Machine Studies*, *20*(2), 201-213.

Knapp, A., Neumann, M., Borckmann, M., Walz, R., and Winkle, T. (2009, August). *Code of Practice for the Design and Evaluation of ADAS* (2009, August). RESPONSE 3 project. Version 5.0. Available at: https://www.acea.be/uploads/publications/20090831_Code_of_Practice_ADAS.pdf.

Kurt, A. and Özgüner, Ü. (2013). Hierarchical finite state machines for autonomous mobile systems. *Control Engineering Practice*, *21*(2), 184-194.

Lake, A., and Cook, C. (1994, April). Use of factor analysis to develop OOP software complexity metrics. In *Proc. 6th Annual Oregon Workshop on Software Metrics*, Silver Falls, Oregon.

Landauer, T. K., and Nachbar, D. W. (1985). Selection from alphabetic and numeric menu trees using a touch screen: Breadth, depth, and width, *ACM SIGCHI Bulletin*, *16*(4), 73-78.

Large, D. R., Burnett, G., Benford, S., and Oliver, K. (2016). Crowdsourcing good landmarks for in-vehicle navigation systems, *Behaviour & Information Technology*, *35*(10), 807-816.

Ledang, H., and Souquières, J. (2002, May). Contributions for modelling UML state-charts in B. In *International Conference on Integrated Formal Methods*, Springer, Berlin, Heidelberg, 109-127.

Lee, E., and MacGregor, J. (1985). Minimizing user search time in menu retrieval systems, *Human Factors*, *27*(2), 157-162.

Leveson, N. G., and Palmer, E. (1997, October). Designing automation to reduce operator errors, *1997 IEEE International Conference on Computational Cybernetics and Simulation Systems, Man, and Cybernetics, 2*, New York, NY: IEEE, 1144-1150.

Li, G., Li, S.E., Cheng, B. and Green, P. (2017). Estimation of driving style in naturalistic highway traffic using maneuver transition probabilities, *Transportation Research Part C: Emerging Technologies*, *74*, 113-125.

Lin, B.T.W., Kang, T-P., and Green, P. (2014). *Driver Response Times to In-Vehicle Warning Signals* (technical report UMTRI-2014-37), Ann Arbor, MI: University of Michigan Transportation Research Institute.

Lo, V. E. W., Green, P. A., and Franzblau, A. (2011). Where do people drive? Navigation system use by typical drivers and auto experts, *The Journal of Navigation*, *64*(2), 357-373.

Lu, Z., Coster, X., and de Winter, J. (2017). How much time do drivers need to obtain situation awareness? A laboratory-based study of automated driving. *Applied Ergonomics*, *60*, 293-304.

Lu, Z., Happee, R., Cabrall, C. D., Kyriakidis, M., and de Winter, J. C. (2016). Human factors of transitions in automated driving: A general framework and literature survey. *Transportation Research Part F: Traffic Psychology and Behaviour*, *43*, 183-198.

Lv, C., Cao, D., Zhao, Y., Auger, D. J., Sullman, M., Wang, H., Dutka, L.M., Skrypchuk, L. and Mouzakitis, A. (2018). Analysis of autopilot disengagements occurring during autonomous vehicle testing. *IEEE/CAA Journal of Automatica Sinica*, *5*(1), 58-68.

MacKenzie, I. S., and Soukoreff, R. W. (2002). Text entry for mobile computing: Models and methods, theory and practice. *Human–Computer Interaction*, *17*(2-3), 147-198.

Mayer, C.E. (2002, May 26). Why Won't We Read the Manual, *Washington Post*, https://www.washingtonpost.com/archive/business/2002/05/26/why-wont-we-read-the-manual/b7f08098-1d08-4d67-9e3e-8f3814f4d90a/?utm_term=.3f2344c6f2a6, retrieved April 1, 2018.

McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on Software Engineering*, (4), 308-320.

McCabe, T.J., (1977). *Elements of Software Science*, New York, NY: North Holland.

McCabe, T. J., and Butler, C. W. (1989). Design complexity measurement and testing. *Communications of the ACM*, *32*(12), 1415-1425.

McCall, R., McGee, F., Mirnig, A., Meschtscherjakov, A., Louveton, N., Engel, T., and Tscheligi, M. (2018). A taxonomy of autonomous vehicle handover situations. *Transportation Research Part A: Policy and Practice*.

McCrobie, D. (1986). An analysis of acronyms in written text, *Proceedings of the Human Factors Society 30th Annual Meeting*, Santa Monica, CA: Human Factors and Ergonomics Society, 936-940.

Mehlenbacher, B., Wogalter, M. S., and Laughery, K. R. (2002). On the reading of product owner's manuals: Perceptions and product complexity, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Santa Monica, CA: Human Factors and Ergonomics Society, 730-734.

Merat, N., Jamson, A. H., Lai, F. C., Daly, M., and Carsten, O. M. (2014). Transition to manual: Driver behaviour when resuming control from a highly automated vehicle, *Transportation Research Part F: Traffic Psychology and Behaviour*, *27*, 274-282.

Meyer, D. E., and Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part 2. Accounts of psychological refractory-period phenomena, *Psychological Review*, *104*(4), 749.

Michon, J. A. (Ed.). (1993). *Generic Intelligent Driver Support*. Boca Raton, FL: CRC Press.

Miller, D. P. (1981a). *Factors Affecting Item Acquisition Performance in Hierarchical Systems: Depth vs. Breadth* (Doctoral dissertation), Columbus, Ohio: Ohio State University.

Miller, D. P. (1981b, October). The depth/breadth tradeoff in hierarchical computer menus. *Proceedings of the Human Factors Society Annual Meeting*, Santa Monica, CA: Human Factors and Ergonomics Society, 296-300.

Mirnig, A. G., Gärtner, M., Laminger, A., Meschtscherjakov, A., Trösterer, S., Tscheligi, M., McCall, R., and McGee, F. (2017, September). Control transition interfaces in semiautonomous vehicles: A categorization framework and literature analysis, *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, New York, NY: Association for Computing Machinery, 209-220.

Mok, B., Johns, M., Yang, S., and Ju, W. (2017, October). Reinventing the Wheel: Transforming Steering Wheel Systems for Autonomous Vehicles, *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, New York, NY: Association for Computing Machinery, 229-241.

Mok, B., Johns, M., Yang, S., and Ju, W. (2017). Actions Speak Louder: Effects of a Transforming Steering Wheel on Post-Transition Driver Performance, *IEEE 20th International Conference on Intelligent Transportation Systems*, 431-438.

Moody, D. (2009). The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, *35*(6), 756-779.

Moran, T. P. (1983, December). Getting into a system: External-internal task mapping analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* New York, NY: ACM, 45-49.

Moray, N., Rodriguez, D., and Clegg, B. (2000, July). Levels of automation in process control. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Santa Monica, CA: Human Factors and Ergonomics Society, 93-96.

Mourant, R. R., and Langolf, G. D. (1976). Luminance specifications for automobile instrument panels, *Human Factors*, *18*(1), 71-84.

National Aeronautics and Space Administration (2010). *NASA/SP-2010-3047 Human Integration Design Handbook*, Washington D.C.: National Aeronautics and Space Administration, https://ston.jsc.nasa.gov/collections/TRS/_techrep/SP-2010-3407.pdf.

Nawrocki, L. (1979). *Word Abbreviations in Man-Computer Communication Systems* (ARI working paper 79-034), Arlington, VA: U.S. Army Research Institute.

Norman, K. L. (1991). *The Psychology of Menu Selection: Designing Cognitive Control at the Human/Computer Interface*, Bristol, United Kingdom: Intellect Books.

Norman, K. L., and Chin, J. P. (1988). The effect of tree structure on search in a hierarchical menu selection system, *Behaviour & Information Technology*, *7*(1), 51-65.

Novick, D. G., and Ward, K. (2006, October). Why don't people read the manual? *Proceedings of the 24th annual ACM International Conference on Design of Communication*, New York, NY: Association for Computing Machinery, 11-18.

Oda, K. and Imahashi, M. (1995). Critical character size for reading Japanese, poster at 1995 ARVO (Association for Research on Vision and Ophthalmology) meeting (www.twcu.ac.jp/~k-oda/arvo/ARVO5Oda.html).

Olsson, M. (2016). *Behavior Trees for Decision-Making in Autonomous Driving* (master's thesis), Stockholm, Sweden: School of Computer Science and Communication, Kth Royal Institute of Technology.

Oman, P., and Hagemeister, J. (1992, November). Metrics for assessing a software system's maintainability. In *Conference on Software Maintenance, 1992. Proceedings.,* New York, NY: IEEE, 337-344.

Paap, K. R., and Roske-Hofstrand, R. J. (1986). The optimal number of menu options per panel, *Human Factors*, *28*(4), 377-385.

Parasuraman, R., Sheridan, T. B., and 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*, *30*(3), 286-297.

Peters, G.A. and Adams, B.B. (1959, May 25). These 3 criteria for readable Ppanel markings, *Product Engineering*, *30*(21), 55-57.

Polson, P. G., and Kieras, D. E. (1985). A quantitative model of the learning and performance of text editing knowledge. *ACM SIGCHI Bulletin*, *16*(4), 207-212.

Posnett, D., Hindle, A., and Devanbu, P.T. (2011). A simpler model of software readability, In *Working Conference on Mining Software Repositories*, 73–82.

Reimer, B., Mehler, B., Dobres, J., Coughlin, J. F., Matteson, S., Gould, D., Chahine, N, and Levantovsky, V. (2014). Assessing the impact of typeface design in a text-rich automotive user interface. *Ergonomics*, *57*(11), 1643-1658.

Ross, T., Midtland, K., Fuchs, M., Pauzie, A., Engert, A., Duncan, B., Vaughan, G., Vernet, M., Peters, H., Burnett, G., and May, A (1996). *HARDIE Design Guidelines Handbook: Human Factors Guidelines for Information Presentation by ATT Systems*, Luxembourg: Commission of the European Communities.

Rudin-Brown, C.M. and Green, P. (2017). *Expert Review of Automated Vehicle Driver Interfaces: The Automotive Human Factors and Human-Computer Interaction Literature* (technical report), Toronto, CA: Human Factors North.

Sarter, N. B., Woods, D. D., and Billings, C. E. (1997). Automation surprises, *Handbook of Human Factors and Ergonomics*, *2*, 1926-1943.

Save, L., and Feuerberg, B. (2012). Designing human-automation interaction: a new level of automation taxonomy, in deWaard, D., Brookhuis, K. Dehais, F., Weikert, C., Rottger, S., Manzey, D., Biede, S., Reuzeau, F., and Terrier, P. (eds.), *Proceedings of the Human Factors and Ergonomics Society Europe Chapter Annual Meeting (HFES Europe).*

Shao, J., and Wang, Y. (2003, May). A new measure of software complexity based on cognitive weights. In *IEEE Canadian Conference on Electrical and Computer Engineering, 2003, CCECE 2003*, 2, 1333-1338), New York, NY: IEEE.

Shepperd, M. (1988). A critique of Cyclomatic Complexity as a software metric. *Software Engineering Journal*, *3*(2), 30-36.

Sheridan, T.B., and Verplank, W. (1978). *Human and Computer Control of Undersea Teleoperators*, Cambridge, MA: Man-Machine Systems Laboratory, Department of Mechanical Engineering, Cambridge, MAP: MIT.

Smiley, A., Smahel, T., and Green, P. (2017). *Laboratory Testing of Level 2 and Level 3 Driver Interfaces: A Literature Review* (technical report), Toronto, CA: Human Factors North.

Smith, S. L. (1979). Letter size and legibility, *Human Factors*, *21*(6), 661-670.

Society of Automotive Engineers (2000, July). *Definition and Experimental Measures Related to the Specification of Driver Visual Behavior Using Video Based Techniques* (SAE Recommended Practice J2396, Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2002, February). *ITS In-Vehicle Message Priority* (SAE Recommended Practice J2395, Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2004, February 12). *SAE Recommended Practice Navigation and Route Guidance Function Accessibility While Drivin*g (SAE 2364), Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2007). *Driver Hand Control Reach* (SAE Recommended Practice J287), Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2008, July). *Process for Comprehension Testing of In-Vehicle Icons* (SAE Information Report J2830, Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2009). *Describing and Measuring the Driver's Field of View* (SAE Recommended Practice J1050), Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2009). *Design Criteria – Driver Hand Controls Location for Passenger Cars, Multipurpose Passenger Vehicles, and Trucks (10000 GVW and Under)* (SAE Recommended Practice J1138), Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2010). *Motor Vehicle Drivers' Eye Locations* (SAE Recommended Practice J941), Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2010*). Direction-of-Motion Stereotypes for Automotive Hand Controls* (SAE Recommended Practice J1139, Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2010). *Road Vehicles - Symbols for Controls, Indicators, and Tell-Tales* (SAE Standard J2402, Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2012, April). *Development of Design and Engineering Recommendations for In-Vehicle Alphanumeric Messages* (SAE Information Report J2831, Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2015). *Operational Definitions for Driving Performance Measures and Statistics* (SAE Recommended Practice J2944), Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2016). *Calculation of the Time to Complete In-Vehicle Navigation and Route Guidance Tasks* (SAE Recommended Practice J2365, Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2016). *Human Factors Definitions for Automated Driving and Related Research Topics* (SAE Information Report J3114), Warrendale, PA: Society of Automotive Engineers.

Society of Automotive Engineers (2018). *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles* (SAE Recommended Practice J3016), Warrendale, PA: Society of Automotive Engineers.

Stevens, A., and Cynk, S. (2011). *Checklist for the Assessment of in-Vehicle Information Systems*. Available at: http://umich.edu/~driving/documents/MIS005TRLChecklist2011.pdf.

Stevens, A., Quimby, A., Board, A., Kersloot, T. and Burns, P. (2004). *Design Guidelines for Safety of In-Vehicle Information Systems,* (TRL Research Report No. PA3721/01). Available at: http://www.transport-research.info/sites/default/files/project/documents/20060728_165141_88073_UG340_Final_Report.pdf.

SWOV (2003a). *ADVISORS Final Publishable Report.* EC PROJECT GRD1 2000 10047 (http://www.transport-research.info/sites/default/files/project/documents/20060622_165324_86183_ADVISORS_Final_Report.pdf).

SWOV (2003b). *ADVISORS Final Publishable Report: Annexes*.

Tellis, L., Engelman, G., Christensen, A., Cunningham, A., Debouk, R., Egawa, K., Green, C., Kawashima, C., Nicols, G., Prokhorov, D., Wendling, B., Yako, S., and Kiger, S. (2016, March). *Automated Vehicle Research for Enhanced Safety Final Report* (NHTSA contract no. DTFH6114H00002), Crash Avoidance Metrics Partnership on behalf of the Automated Vehicle Research Consortium.

Tobias, M., Mwangi, W., and Michael, K. (2015). Empirical Evaluation of Complexity Metrics for Component Based Systems, *Journal of Theoretical & Applied Information Technology*, *73*(2), 275-282.

Tönnis, M., Klinker, G., and Plavšic, M. (2009). Survey and classification of head-up display presentation principles, *Proceedings of the International Ergonomics Association (IEA)*.

Trimble, T.E., Bishop, R., Morgan, J.F., and Blanco, M. (2014). *Human Factors Evaluation of Level 2 and Level 3 Automated Driving Concepts: Past Research, State of Automation Technology, and Emerging Systems Concepts* (technical report DOT HS 812 043), Washington, D.C.: U.S. Department of Transportation, National Highway Traffic Safety Administration.

Trösterer, S., Meschtscherjakov, A., Mirnig, A.G., Lupp, A., Gärtner, M., McGee, F., McCall, R., Tscheligi, M., and Engel, T. (2017). What We Can Learn from Pilots for Handovers and (De) Skilling in Semi-Autonomous Driving: An Interview Study, In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, New York, NY: Association for Computing Machinery, 173-182.

Trombetta, W. L., and Wilson, T. L. (1975). Foreseeability of misuse and abnormal use of products by the consumer, *The Journal of Marketing*, 48-55.

Tsimhoni, O., Green, P., and Watanabe, H., (2001). Detecting and reading text on HUDS: Effects of driving workload and message location, paper presented at the ITS-A Annual Meeting (CD-ROM).

U.S. Department of Health and Human Services (2011). *Applying Human Factors and Usability Engineering to Optimize Medical Device Design*, Washington, DC: U.S. Department of Health and Human Services, Food and Drug Administration, Center for Devices and Radiological Health, Office of Device Evaluation, https://www.fda.gov/downloads/MedicalDevices/.../UCM259760.pdf.

U.S. Department of Transportation (2011). *Controls and Displays* (Federal Motor Vehicle Safety Standard 101, 49CFR 571.101), Washington, D.C.: U.S. Department of Transportation, National Highway Traffic Safety Administration.

U.S. Department of Transportation (2013). *Visual-Manual NHTSA Driver Distraction Guidelines for In-Vehicle Electronic Devices* (47 CFR 24890), Washington, D.C.: U.S. Department of Transportation, National Highway Traffic Safety Administration.

United States Army (2000). *Human Engineering Data Digest*, Redstone Arsenal, AL: US Army Missile Command, Human Engineering Laboratory Detachment, https://www.acq.osd.mil/rd/hptb/hfetag/products/documents/HE_Design_Data_Digest.pdf, retrieved April 17, 2018.

United States Department of Defense (2012). *Department of Defense Design Criteria Standard - Human Engineering* (Military Standard MIL-STD 1472G) Washington D.C. Department of Defense (Quicksearch.dla.mil - free download of 1472G (and older versions)).

United States Department of Transportation (2018). *Preparing to the Future of Transportation, Automated Vehicles 3.0*, Washington, D.C.: United States Department of Transportation.

United States Nuclear Regulatory Commission (2002). *Human-System Interface Design Review Guidelines* (NUREG-0700, Revision 2), Washington, D.C.: Nuclear Regulatory Commission, https://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0700/.

Vossen, P., Evans, S., and Ziegler, J. (1988). *Cognitive Complexity Theory's Approach to the Modelling of Human-Computer Interaction*, FhG iAO 1988, 1-19.

Weintraub, D.J. (1992). *Human Factors Issues in Head-Up Display Design: The Book of HUD*, Wright-Patterson Air Force Base, OH: CSERIAC.

Welford, A. T. (1952). The 'psychological refractory period' and the timing of high-speed performance—A review and a theory, *British Journal of Psychology*, *43*(1), 2-19.

Weyuker, E. J. (1988). Evaluating software complexity measures. *IEEE Transactions on Software Engineering*, *14*(9), 1357-1365.

Wickens, C.D., Mavor, A.S., Parasuraman, R., and McGee, J.P. (editors) (1998). *Panel on Human Factors in Air Traffic Control Automation, The Future of Air Traffic Control: Human Operators and Automation*, Washington, D.C.: National Research Council.

Winkle, T. (2016). *Safety Benefits of Automated Vehicles: Extended Findings from Accident Research for Development, Validation and Testing* (chapter 17), in Maurer, M., Gerdes, J.C., Lenz, B., and Winner, H. (eds.), Autonomous Driving, Springer Open.

Wintersberger, P., Green, P., and Riener, A. (2017). Am I driving or are you or are we both? A taxonomy for handover and handback in automated driving, *9th International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design.*

Yoo, H., Tsimhoni, O., Watanabe, H., Green, P. and Shah, R. (1999). *Display of HUD Warnings to Drivers: Determining an Optimal Location* (technical report UMTRI-99-9, ITS RCE report #939423), Ann Arbor, MI, The University of Michigan Transportation Research Institute.

Yoon, S. H., Lim, J., and Ji, Y. G. (2015). Assessment model for perceived visual complexity of automotive instrument cluster. *Applied Ergonomics*, *46*, 76-83.

Yu, S., and Zhou, S. (2010, April). A survey on metric of software complexity. In *The 2nd IEEE International Conference on Information Management and Engineering (ICIME), 2010*. New York, NY: IEEE, 352-356.

Zeeb, K., Buchner, A., and Schrauf, M. (2015). What determines the take-over time? An integrated model approach of driver take-over after automated driving, *Accident Analysis & Prevention*, *78*, 212-221.

Zhang, M., Li, N., Girard, A., and Kolmanovsky, I. (2017, October). A Finite State Machine Based Automated Driving Controller and its Stochastic Optimization. In *ASME 2017 Dynamic Systems and Control Conference,* American Society of Mechanical Engineers, V002T07A002-V002T07A002

Zhou, S., Jeong, H., and Green, P. A. (2017). How consistent are the best-known readability equations in estimating the readability of design standards? *IEEE Transactions on Professional Communication*, *60*(1), 97-111.

Zuse (1991). *Software Complexity*, Berlin, Germany: Walter de Gruyter.

APPENDIX A – GUIDELINES IN THIS REPORT

Guideline 2-1: For each automation system feature, a first step in design should be selecting a proposed level of automation, either using Sheridan's 10-Levels of Automation or Parasuraman's 10-Levels of Automation.

Guideline 3-1: Make sure the HMI complies with standards of the SAE Safety and Human Factors Committee and committees reporting to it (Adaptive Cruise Control and Forward Collision Warning Committee, Adaptive Cruise Control and Forward Collision Warning Committee, the Driver Metrics, Performance, Behaviors and States Committee, the Driver Vehicle Interface (DVI) Committee, the Advanced Driver Assistance Systems (ADAS) Committee, and the Driving Automation Systems Committee). Also important is the On-Road Automated Driving (ORAD) committee.

Guideline 5-1. The vehicle shall show how confident the vehicle is in the data it is using to drive, in particular lane markings.

Guideline 5-2. Where there are alternative paths or the vehicle is changing paths, the vehicle shall show how confident the vehicle is in the path it intends to take/is taking (for example when under navigational control or using automatic lane change features).

Guideline 5-3. The vehicle shall show who is steering (driver, automation, shared), who is controlling speed (same), who is responsible for object and event detection (same), and if a transition is occurring.

Guideline 5-4. The operational design domain for a mode shall be shown. For speed control this includes the range of speeds within which it operates. For gap control, the range of gaps shall be indicated.

Guideline 5-5. Except for the control used, the sequence of actions to change speed and gap settings should be the same.

Guideline 5-6. Where the driver is to take over control of the vehicle (AH transfer), the time period over which that shall occur and the time remaining shall be indicated.

Guideline 5-7. When a temporary takeover (AH transfer) occurs, what the driver did for that to occur shall be indicated. In addition, guideline 3 will be followed. Of particular concern is where maneuvers (e.g., lane change) are interrupted.

Guideline 6-1: For setting tasks and for mode change tasks, create matrices or some other visual representation that shows the method to accomplish each task and highlights inconsistencies in methods. The representations for setting and mode change tasks can be different.

Guideline 6-2: For each task associated with the automation, write out the production rules to accomplish them.

Guideline 6-3: Compute the complexity of each task and each collection of tasks. The formulas for those computations need to be determined. Candidate independent variables include Cyclomatic Complexity, lines of code, the number of production rules, the number of operators and operands, and other statistics.

Guideline 7-1: Minimize the number of modes.

Guideline 7-2: Make the transitions between modes consistent.

Guideline 7-3: Make tasks that have negative consequences not easy to do.

Guideline 7-4: Minimize the number of steps to set a mode.

Guideline 7-5: Minimize the number of steps to set the automation.

Guideline 8-1: *The guiding principle is to design for reasonable and expected use and misuse.*

Guideline 8-2: The driver should be able to determine the ODD without looking away from the road scene.

Guideline 8-3: The driver must be able to turn automation off without looking away from the road scene because sometimes it needs to be done quickly under challenging circumstances.

Guideline 8-4: The driver must be able to determine the automation mode without looking away from the road scene because sometimes the automation exerts control (or fails to exert control) in unexpected ways that could require immediate action from the driver.

Guideline 8-5: As with determining the mode, drivers must be able to disengage automation instantly for the same reason.

Guideline 8-6: The driver must be able to determine the override is occurring without looking away from the road scene.

Guideline 8-7: The driver must be able to determine the current settings without looking away from the road scene.

Guideline 8-8: Which settings need to be able to be changed without looking away from the road will depend up the frequency with which a setting is changed and the extent of its connection with crash risk.

Guideline 8-9: The driver must be able to determine what the vehicle knows and how confident the vehicle is in what it knows without looking away from the road scene.

Guideline 8-10: When a takeover is scheduled (AH transfer), the driver shall be informed how much time they have to take over. This is true for unscheduled takeovers as well.

Guideline 8-11: If a takeover (AH transfer) is to occur, and the driver is not capable of taking over, then the driver should be informed of such.

Guideline 9-1: The design of the interface shall comply with the latest version of Military Standard 1472.

Guideline 10-1: The HMI must communicate which systems are provided.

Guideline 10-2: Collections of controls and displays must be labeled explicitly.

Guideline 10-3: Do not overload the user.

Guideline 10-4: Avoid conflicting messages.

Guideline 10-5a: Place displays where drivers expect to find them.

Guideline 10-5b: Place controls where driver expect to find them.

Guideline 10-6a: Place displays where drivers can easily see them.

Guideline 10-6b: Place controls where drivers can easily reach them.

Guideline 10-7: Put controls and displays together that are related or used together and separate those that are not.

Guideline 10-8: Auditory warnings need to be loud enough so they can be readily heard. Text and graphics need to be large enough and have enough contrast so they can be easily seen. Vibrations need to be intense enough to be readily felt.

Guideline 10-9: For visual, auditory, and haptic patterns, breaks between flashes, beeps, and pulses need to be long enough so that they are perceived, but not so long that the sequence takes too long to complete, delaying driver responses.

Guideline 10-10: Keep sequences of flashes, tones, and vibrations short.

Guideline 10-11: Provide information the user will know is correct.

Guideline 10-12: All of the information needed is not provided.

Guideline 10-13: Provide correct information.

Guideline 10-14: The format of the information should match the decision the driver needs to make and not require recoding or transformation.

Guideline 10-15: The information should be provided in a format that is meaningful to the driver.

Guideline 10-16: To aid making decisions, the information relevant to that decision should be grouped together.

Guideline 10-17: The interface should clearly communicate how to perform a task.

Guideline 10-18: Minimize the number of steps and the time required to complete a task.

Guideline 10-19: Design controls so that their design suggests how they should be operated.

Guideline 10-20: Controls should be large enough so they can be easily grasped.

Guideline 10-21: Controls should operate in a manner consistent with their appearance.

Guideline 10-22: The operation of controls and movement of displays should conform to population stereotypes.

Guideline 10-23: The operation of a control should not have unintended consequences.

Guideline 10-24: Forces to operate controls should be not so excessive as to interfere with intended operation but not so low that inadvertent operation can occur.

Guideline 10-25: Provide feedback that a control has been operated.

Guideline 11-1: The information in a display shall be in the format the driver needs to make a decision that is part of a task.

Guideline 11-2: As a starting point, text height should be at least 0.007 times the viewing distance. Important displays should be double to quadruple that size.

Guideline 11-3: Use the same combination of truncation and vowel deletion rules when creating all abbreviations.

Guideline 11-4: Use the symbols in ISO 2575 supplemented by text.

Guideline 11-5: Allow for destination entry in many methods and in any order.

Guideline 11-6: For planning, show the map north up. For guidance, show the map heading up.

Guideline 11-7: For route following, provide turn guidance directly in front of the driver, for example on a HUD or augmented reality display.

Guideline 11-8: For navigation decisions, providing good voice guidance is a priority and is even more important than the visual display.

Guideline 11-9: Provide a “say again” button for voice guidance that allows the driver to have the last spoken instruction repeated without their hands leaving the steering wheel.

Guideline 11-10: Include landmarks in turn instructions.

Guideline 11-11: Provide countdowns to indicate the distance to turns.

Guideline 11-12: For closely spaced turns, present them together.

Guideline 11-13: Developing solutions to the “last block” problem will lead to much more useful navigation systems.

Guideline 11-14: For 2 state selections, use rocker switches, with pushbuttons as alternatives. For 3 or more states, use slide switches or knobs with discrete positions.

Guideline 11-15: Stereotype 1: Numbers should increase from left to right (except in Arabic and Hebrew).

Guideline 11-16: Stereotype 2: Dimensional Expectancies - Clockwise, up, to the right, or forward for increase or on.

Guideline 11-17: Stereotype 3 - Clockwise-Clockwise Hypothesis -Clockwise control movements lead to CW display movements.

Guideline 11-18: Stereotype 4 - Parallel Movement - Linear controls and linear scales move together.

Guideline 11-19: Stereotype 5 - Warrick's Principle – An indicator moves in the same direction as the side of the control adjacent to it (scale side effect).

Guideline 11-20: Compatibility Principle 6 - The initial position of the indicator affects the control action.

Guideline 11-21: Compatibility Principle 7 - Available Limb Motion - Clockwise with right hand, counterclockwise with left.

Guideline 11-22: For touch screen widgets, the following are recommended.

Guideline 11-23: In general, for typing in English, a Qwerty keyboard arrangement is recommended.

Guideline 11-24: Use computational models to select menu architectures.

Guideline 11-25: The display needs to be large enough to show all of the menu options for that choice simultaneously (and sometimes more).

Guideline 11-26: There is a tradeoff between menu breadth (how many items are shown at the same time) and depth (how many menu levels are provided), but there is not universal optimum.

Guideline 11-27: Use card sorting and naming methods to develop menu grouping and category names.

Guideline 11-28: Menu category names should be distinct and indicate to what they refer.

Guideline 11-29: Consider using examples of category content to help with category naming.

Guideline 11-30: The need to have a home page varies with the system.

Guideline 11-31: For now, use 11 s as a reasonable worst case for takeover (AH transfer) reaction time and 8 s as a reasonable worst case for handback reaction time.

APPENDIX B – LEVEL OF AUTOMATION TECHNOLOGY

Source: Save, Feuerberg, and Avia (2012) pages 48-50.

A INFORMATION ACQUISITION	B INFORMATION ANALYSIS	C DECISION AND ACTION SELECTION	D ACTION IMPLEMENTATION
A0 **Manual Info Acquisition**	**B0** **Working Memory Based Info Analysis**	**C0** **Human Decision Making**	**D0** **Manual Action and Control**
The human acquires relevant information on the process s/he is following without using any tool.	The human compares, combines and analyses different information items regarding the status of the process s/he is following by way of mental elaborations. S/he does not use any tool or support external to her/his working memory.	The human generates decision options, selects the appropriate ones and decides all actions to be performed.	The human executes and controls all actions manually.
A1 **Artefact-Supported Info Acquisition**	**B1** **Artefact-Supported Info Analysis**	**C1** **Artefact-Supported Decision Making**	**D1** **Artefact-Supported Action Implementation**
The human acquires relevant information on the process s/he is following with the support of low-tech non-digital artefacts. *Ex. 1) Identification of aircraft positions on an aerodrome/airport according to Procedural Air Traffic Control rules and without use of radar support.*	The human compares, combines, and analyses different information items regarding the status of the process s/he is following utilising paper or other non-digital artefacts. *Ex. 1) Use of flight strips to compare altitudes/levels/pl. times of different aircraft and to pre-plan future traffic.*	The human generates decision options, selects the appropriate ones and decides all actions to be performed utilising paper or other non-digital artefacts.	The human executes and controls actions with the help of mechanical non-software based tools. *Ex. 1) Use of a hammer or leverage to increase the kinetic energy of human gesture.* *Ex. 2) Use of a mechanical or hydraulic rudder to achieve a change in direction.*
A2 **Low-Level Automation Support of Info Acquisition**	**B2** **Low-Level Automation Support of Info Analysis**	**C2** **Automated Decision Support**	**D2** **Step-by-step Action Support:**
The system supports the human in acquiring information on the process s/he is following. Filtering and/or highlighting of the most relevant information are up to the human. *Ex. 1) Identification of aircraft positions in the airspace by way of Primary Radar working positions.*	Based on user's request, the system helps the human in comparing, combining and analysing different information items regarding the status of the process being followed. *Ex. 1) Activation by ATCOs of Speed Vectors for specific tracks on the CWP, in order to anticipate potential conflicts in a defined time frame.*	The system proposes one or more decision alternatives to the human, leaving freedom to the human to generate alternative options. The human can select one of the alternatives proposed by the system or her/his own one. *Ex.1) AMAN visualization of the proposed sequence of aircraft.*	The system assists the operator in performing actions by executing part of the action and/or by providing guidance for its execution. However, each action is executed based on human initiative and the human keeps full control of its execution. *Ex. 1) The aural and visual component of TCAS RA in current TCAS II version 7.0 (also LOA C5)*

A3 **Medium-Level Automation Support of Info Acquisition**	B3 **Medium-Level Automation Support of Info Analysis**	C3 **Rigid Automated Decision Support**	D3 **Low-Level Support of Action Sequence Execution**
The system supports the human in acquiring information on the process s/he is following. It helps the human in integrating data coming from different sources and in filtering and/or highlighting the most relevant information items, based on user's settings. *Ex. 1) CWP allowing ATCOs to set flight level filters to display only certain traffic on the screen.*	Based on user's request, the system helps the human in comparing, combining and analysing different information items regarding the status of the process being followed. The system triggers visual and/or aural alerts if the analysis produces results requiring attention by the user. *Ex. 1) ERATO Filtering and What-if function.* *Ex 2). VERA Tool to display the closest point of approach between two aircrafts.*	The system proposes one or more decision alternatives to the human. The human can only select one of the alternatives or ask the system to generate new options.	The system performs automatically a sequence of actions after activation by the human. The human maintains full control of the sequence and can modify or interrupt the sequence during its execution. *Ex. 1) Explicit initiation of an electronic coordination with adjacent sector via digital input (replacing use of telephone).*
A4 **High-Level Automation Support of Info Acquisition**	**B4** **High-Level Automation Support of Info Analysis**	**C4** **Low-Level Automatic Decision Making**	**D4** **High-Level Support of Action Sequence Execution**
The system supports the human in acquiring information on the process s/he is following. The system integrates data coming from different sources and filters and/or highlights the information items which are considered relevant for the user. The criteria for integrating, filtering and highlighting the relevant information are predefined at design level but visible to the user. *Ex.1) D-TAXI tool (including graphical route information)*	The system helps the human in comparing, combining and analysing different information items regarding the status of the process being followed, based on parameters pre-defined by the user. The system triggers visual and/or aural alerts if the analysis produces results requiring attention by the user. *Ex. 1) MTCD visual alerts (allowing some tuning of parameters by the user)*	The system generates options and decides autonomously on the actions to be performed. The human is informed of its decision. *Ex.1) Aural and visual component of TCAS RA in current TCAS II version 7.0 (also LOA D2)*	The system performs automatically a sequence of actions after activation by the human. The human can monitor all the sequence and can interrupt it during its execution. *Ex.1) Acknowledgment by pilot of a clearance received trough CPDLC (data-link) and automatically sent to FMS and autopilot.* *Ex. 2) Autopilot following the FMS trajectory.*

A5 **Full Automation Support of Info Acquisition**	B5 **Full Automation Support of Info Analysis**	C5 **High-Level Automatic Decision Making**	D5 **Low-Level Automation of Action Sequence Execution**
The system supports the human in acquiring info on the process s/he is following. The system integrates data coming from different sources and filters and/or highlights the information items considered relevant for the user. The criteria for integrating, filtering and highlighting are predefined at design level and not visible to the user	The system performs comparisons and analyses of data available on the status of the process being followed based on parameters defined at design level. The system triggers visual and/or aural alerts if the analysis produces results requiring attention by the user. *Ex. 1) STCA visual and aural alerts.*	The system generates options and decides autonomously on the action to be performed. The human is informed of its decision only on request. (Always connected to to an *Action Implementation* level not lower than D5.)	The system initiates and executes automatically a sequence of actions. The human can monitor all the sequence and can modify or interrupt it during its execution. *Ex. 1) Implicit initiation of an electronic co-ordination with adjacent sector as agreed exit conditions (according to Letter of Agreement) cannot be met anymore after changes to the a/c trajectory has been made.*
		C6 **Full Automatic Decision Making**	D6 **Medium-Level Automation of Action Sequence Execution**
		The system generates options and decides autonomously on the action to be performed without informing the human. (Always connected to an *Action Implementation* level not lower than D5.)	The system initiates and executes automatically a sequence of actions. The human can monitor all the sequence and can interrupt it during its execution. *Ex.1) TCAS AP/FD during a corrective TCAS RA.*
			D7 **High-Level Automation of Action Sequence Execution**
			The system initiates and executes a sequence of actions. The human can only monitor part of it and has limited opportunities to interrupt it.
			D8 **Full Automation of Action Sequence Execution**
			The system initiates and executes a sequence of actions. The human cannot monitor nor interrupt it until the sequence is not terminated.

APPENDIX C – BILLINGS GUIDELINES FOR HUMAN-CENTERED AVIATON AUTOMATION

Category	Guideline
General guidelines	1. The human operator must be in command.
	2. To command effectively, the human operator must be involved.
	3. To remain involved, the human operator must be appropriately informed.
	4. The human operator must be informed about automated systems behaviour.
	5. Automated systems must be predictable.
	6. Automated systems must also monitor the human operators.
	7. Each agent in an intelligent human-machine system must have knowledge of the intent of the other agents.
	8. Functions should be automated only if there is a good reason for doing so.
	9. Automation should be designed to be simple to train, to learn, and to operate.
Specific requirements and guidelines	10. Automated systems must be comprehensible to pilots.
	11. Automation must insure that the pilot is not removed from the command role.
	12. The primary objective of automation is to maintain and enhance situation awareness. All automation elements and displays must contribute to this objective.
	13. Management automation should make airplanes easier to manage.
	14. Designers must assume that human operators will rely on reliable automation, because they will.
Guidelines for aircraft control automation	15. Control automation should be limited in its authority. It must not be permitted to become "insubordinate".
	16. Control automation must never be permitted to perform, or fail, silently.
	17. Designers should not foreclose pilot authority to override normal aircraft operating limits when required for safe mission completion without compelling reasons for doing so.
	18. Control automation should provide the human operator with an appropriate range of control and management options.
	19. Designers should keep the flight crew involved in the operation by requiring of them meaningful and relevant tasks, regardless of the level of management being utilized by them.
	20. Control automation should be designed to be of most help during times of highest workload, and somewhat less help during times of lowest workload.
	21. Control automation should be designed both for maximum error resistance and maximum error tolerance.

Guidelines for aircraft information automation	22. Emphasize information in accordance with its importance.
	23. Alerting and warning systems should be as simple and foolproof as possible.
	24. Integration of information does not mean simply adding more elements to a single display.
	25. Automation poses additional monitoring requirements; pilots must be able to monitor both the status of the automation and the status of the functions controlled by that automation.
	26. Design system automation to ensure that critical functions are monitored as well as executed.

Guidelines for aircraft management automation	27. Flight management system interfaces must be as error tolerant as possible.
	28. Insure that flight operations remain within the capacities of the human operator.

Guidelines for error management	29. Standardize critical interfaces across fleets and manufacturers wherever possible to prevent flight crew errors in operation.

APPENDIX D – ENDSLEY SITUATION AWARENESS DESIGN PRINCIPLES

Source: Endsley, Bolte, and Jones (2003), pages 251-252

Category	Guideline
General	1. Organize information around goals 2. Present Level 2 information directly—support comprehension 3. Provide assistance for Level 3 SA projections 4. Support global SA 5. Support trade-offs between goal-driven and data-driven processing 6. Make critical cues for schema activation salient 7. Take advantage of parallel processing capabilities 8. Use information filtering carefully
Certainty Design Principles	9. Explicitly identify missing information 10. Support sensor reliability assessment 11. Use data salience in support of certainty 12. Represent information timeliness 13. Support assessment of confidence in composite data 14. Support uncertainty management activities
Complexity Design Principles	15. Just say no to feature creep—buck the trend 16. Manage rampant featurism through prioritization and flexibility 17. Insure logical consistency across modes and features 18. Minimize logic branches 19. Map system functions to the goals and mental models of users 20. Provide system transparency and observability 21. Group information based on Levels 2/3 SA requirements and goals 22. Reduce display density, but don't sacrifice coherence 23. Provide consistency and standardization on controls across different displays and systems 24. Minimize task complexity
Alarm Design Principles	25. Don't make people rely on alarms—provide projection support 26. Support alarm confirmation activities 27. Make alarms unambiguous 28. Reduce false alarms, reduce false alarms, reduce false alarms 29. Set missed alarm and false alarm trade-offs appropriately 30. Use multiple modalities to alarm, but insure they are consistent 31. Minimize alarm disruptions to ongoing activities 32. Support the assessment and diagnosis of multiple alarms 33. Support the rapid development of global SA of systems in an alarm state

Automation Design Principles	34. Automate only if necessary 35. Use automation for assistance in carrying out routine actions rather than higher level cognitive tasks

SA Design Principles	36. Provide SA support rather than decisions 37. Keep the operator in control and in the loop 38. Avoid the proliferation of automation modes 39. Make modes and system states salient 40. Enforce automation consistency 41. Avoid advanced queuing of tasks 42. Avoid the use of information cueing 43. Use methods of decision support that create human/system symbiosis 44. Provide automation transparency

Multioperator Design Principles	45. Build a common picture to support team operations 46. Avoid display overload in shared displays 47. Provide flexibility to support shared SA across functions 48. Support transmission of different comprehensions and projections across teams 49. Limit nonstandardization of display coding techniques 50. Support transmission of SA within positions by making status of elements and states overt

APPENDIX E – CAMP GUIDELINES

LEVEL 2 Safety Principle Criteria
Safety Principle 2.1 For Level 2 driving automation, avoidance of hazards depends on the vehicle driver properly using the vehicle and automation, which includes: i. Driver ensuring the vehicle operational readiness before engaging driving automation features ii. Driver completing the Object and Event Detection Response (OEDR) subtask in order to complete the DDT, by providing the appropriate responses to all relevant objects and events, in cases when the driving automation does not provide the appropriate response to avoid hazards
Safety Principle 2.2 The driving automation system shall be integrated with the vehicle in a manner that does not inhibit the driver's ability to perform the DDT, which includes: i. Control of the vehicle, lateral and longitudinal ii. OEDR
Safety Principle 2.3 The driving automation system shall prioritize predefined driver inputs for full control over driving automation commands.

LEVEL 3 Safety Principle Criteria
Safety Principle 3.1 The driver shall ensure operational readiness before engaging driving automation feature.
Safety Principle 3.2 The driving automation system shall be integrated with the vehicle in a manner that does not inhibit the driver's ability to perform the DDT, which includes: i. Control of the vehicle, lateral and longitudinal ii. OEDR
Safety Principle 3.3 The driving automation system shall prioritize predefined driver inputs for full control over driving automation commands
Safety Principle 3.4 The driving automation system must not engage unless activated by the driver. After initially being activated, the automation can automatically resume if appropriate and within the same drive cycle. As a default, automation is not activated at the beginning of each drive cycle.
Safety Principle 3.5 The driving automation system shall provide persistent indication to the driver that signifies when a vehicle is operating in high automation state, i.e., at Level 3 or higher.

Safety Principle 3.6

The driving automation system shall provide indication to the driver upon any request to transition from Level 3 to a lower level (2, 1, or 0).

Safety Principle 3.7

When activated, the driving automation system shall perform the DDT within its application-specific operational design domain, including providing the appropriate responses to relevant objects and events. This includes but is not limited to:

i. Continuous assessment of operation within actual vs. operational design domain

ii. Inhibit operation when operational design domain is not achieved

Safety Principle 3.8

i. The driving automation system shall be designed in such a way that a failure in the driving automation system does not lead to an immediate loss of the longitudinal and/or lateral control in order to allow the driver to respond as prescribed by SP 3.10 (iii)

ii. When the driving automation system is engaged it shall operate in such a way that, if a vehicle failure occurs that impacts longitudinal and/or lateral vehicle dynamics, systems shall continue to stabilize the vehicle's path within the given physical and technical limits in order to allow the driver to react as prescribed by SP 3.10 (i)

Safety Principle 3.9

Before exiting the operational design domain, upon occurrence of a driving automation system failure that prevents performance of the DDT, the driving automation system shall request the driver to take control.

i. Verified driver control inputs shall cause transition from Level 3 into a lower level of automation

ii. The driving automation system shall maintain an operating condition that affords a controlled transition to driver control, regardless of whether the transition is prompted by fault within the driving automation system, or prompted by violation of the intended operational design domain

Safety Principle 3.10

The driver must understand the following:

i. The driver's role is to determine if there has been a vehicle failure that may impact the safe operation of the vehicle, and to take over control of the vehicle when such a failure occurs

ii. In response to a driver request to take over performance of the DDT, the primary response from the driving automation system is to transition out of Level 3 automation and into a lower driving automation level

iii. When the driving automation system is requesting the driver to take control of the vehicle, the driver's role is to respond by taking over control

iv. After requesting the driver to take control, the driving automation system will remain in control for a limited time period

LEVEL 4 Safety Principle Criteria
Safety Principle 4.1 The vehicle operator shall ensure vehicle operational readiness before engaging the driving automation system.
Safety Principle 4.2 When activated, the driving automation system shall perform the DDT and fallback as needed within its application-specific operational design domain, providing the appropriate responses to relevant objects and events. This includes but is not limited to: i. Continuous assessment of operation within actual vs. operational design domain ii. Prohibiting entry into automated driving when the operational domain is not achieved iii. Ability to achieve minimal risk condition if necessary due to any 1 of the following: a. Operator failure to respond appropriately to pending exit of operational design domain b. A failure that prevents performance of the complete DDT
Safety Principle 4.3 The driving automation system may delay its response to operator take-over requests, and/or operator requests to stop driving automation when necessary to avoid causing a hazard.
Safety Principle 4.4 The driving automation system must not engage unless activated by the operator .

LEVEL 5 Safety Principle Criteria
Safety Principle 5.1 The vehicle operator shall ensure vehicle operational readiness before engaging driving automation.
Safety Principle 5.2 The vehicle system overall (including the driving automation system and its integration) shall be designed such that the driving automation system is capable of performing the DDT and fallback as needed, including: i. Providing appropriate responses to relevant objects and events ii. Ability to achieve a minimal risk condition if necessary due to a failure that prevents performance of the complete DDT
Safety Principle 5.3 The driving automation system may delay response to operator requests to take over, and/or operator requests to stop driving automation when necessary to avoid causing a hazard.
Safety Principle 5.4 The driving automation system must not engage unless activated by the operator.

APPENDIX F – EUROPEAN STANDARD OF PRACTICE FOR HMI – ABRIDGED

Note: The - digit sequence before each principle refers to the section number in the guidelines.

Design Goals

4.3.1.1. Design goal I
The system supports the driver and does not give rise to potentially hazardous behaviour by the driver or other road users.

4.3.1.2. Design goal II
The allocation of driver attention while interacting with system displays and controls remains compatible with the attentional demand of the driving situation.

4.3.1.3. Design goal III
The system does not distract or visually entertain the driver.

4.3.1.4. Design goal IV
The system does not present information to the driver which results in potentially hazardous behaviour by the driver or other road users.

4.3.1.5. Design goal V
Interfaces and interface with systems intended to be used in combination by the driver while the vehicle is in motion are consistent and compatible.

Installation Principles

4.3.2.1. Installation principle I
The system should be located and securely fitted in accordance with relevant regulations, standards and manufacturers' instructions for installing the system in vehicles.

4.3.2.2. Installation principle II
No part of the system should obstruct the driver's view of the road scene.

4.3.2.3. Installation principle III
The system should not obstruct vehicle controls and displays required for the primary driving task.

4.3.2.4. Installation principle IV
Visual displays should be positioned as close as practicable to the driver's normal line of sight.

4.3.2.5. Installation principle V
Visual displays should be designed and installed to avoid glare and reflections.

Information Presentation Principles

4.3.3.1. Information presentation principle I
Visually displayed information presented at any one time by the system should be designed in such a way that the driver is able to assimilate the relevant information with a few glances which are brief enough not to adversely affect driving.

4.3.3.2. Information presentation principle II
Internationally and/or nationally agreed standards relating to legibility, audibility, icons, symbols, words, acronyms and/or abbreviations should be used.

4.3.3.3. Information presentation principle III
Information relevant to the driving task should be accurate and provided in a timely manner.

4.3.3.4. Information presentation principle IV
Information with higher safety relevance should be given higher priority.

4.3.3.5. Information presentation principle V
System generated sounds, with sound levels that cannot be controlled by the driver, should not mask audible warnings from within the vehicle or the outside.

Interaction with Displays and Controls Principles

4.3.4.1. Interaction with displays and controls principle I
The driver should always be able to keep at least one hand on the steering wheel while interacting with the system.
4.3.4.2. Interaction with displays and controls principle II
The system should not require long and uninterruptible sequences of manual visual interface. If the sequence is short, it may be uninterruptible.

4.3.4.3. Interaction with displays and controls principle III
The driver should be able to resume an interrupted sequence of interface with the system at the point of interruption or at another logical point.

4.3.4.4. Interaction with displays and controls principle IV
The driver should be able to control the pace of interface with the system. In particular, the system should not require the driver to make time critical responses when providing inputs to the system.

4.3.4.5. Interaction with displays and controls principle V
System controls should be designed in such a way that they can be operated without adverse impact on the primary driving controls.

4.3.4.6. Interaction with displays and controls principle VI

4.3.4.7. The driver should have control of the loudness of auditory information where there is likelihood of distraction.

4.3.4.8. Interface with displays and controls principle VIII
Systems providing nonsafety related dynamic visual information should be capable of being switched to a mode where that information is not provided to the driver.

System Behaviour Principles

4.3.5.1. System behaviour principle I
While the vehicle is in motion, visual information not related to driving that is likely to distract the driver significantly should be automatically disabled, or presented in such a way that the driver cannot see it.

4.3.5.2. System behaviour principle II
The behaviour of the system should not adversely interfere with displays or controls required for the primary driving task and for road safety.

4.3.5.3. System behaviour principle III
System functions not intended to be used by the driver while driving should be made impossible to interact with while the vehicle is in motion, or, as a less preferred option, clear warnings should be provided against the unintended use.

4.3.5.4. System behaviour principle IV
Information should be presented to the driver about current status and any malfunction within the system that is likely to have an impact on safety.

Information About the System Principles

4.3.6.1. Information about the system principle I
The system should have adequate instructions for the driver covering use and relevant aspects of installation and maintenance.

4.3.6.2. Information about the system principle II
System instructions should be correct and simple.

4.3.6.3. Information about the system principle III
System instructions should be in languages or forms designed to be understood by the intended group of drivers.
4.3.6.4. Information about the system principle IV
The instructions should clearly state which functions of the system are intended to be used by the driver while driving and those which are not.

4.3.6.5. Information about the system principle V
Product information should be designed to accurately convey the system functionality.

4.3.6.6. Information about the system principle VI
Product information should make it clear if special skills are required to use the system as intended by the manufacturer or if the product is unsuitable for particular users.

4.3.6.7. Information about the system principle VII
Representations of system use (e.g. descriptions, photographs and sketches) should neither create unrealistic expectations on the part of potential users nor encourage unsafe use.

Recommendations on Influencing Use Principles

5.2.1.1. Recommendation on influencing use I
Employers should ensure that all in-vehicle information systems are maintained in accordance with the manufacturer's instructions.

5.2.1.2. Recommendation on influencing use II
Employers' procedures and incentive schemes should not cause or encourage system misuse. There should be a clear distinction between systems or functions that are intended (by the employer) to be used while driving and those that are not.

5.2.1.3. Recommendation on influencing use III
Adequate training should be given on all in-vehicle systems that employers require drivers to use while driving. Employers should ensure that employees can use the systems without endangering themselves or other road users.

5.2.1.4. Recommendation on influencing use IV
Employers should ensure that a copy of the manufacturer's instructions for use is available in every equipped vehicle.

5.2.1.5. Recommendation on influencing use V
Point-of-sale promotion (e.g. advertising) should not encourage unsafe use.

5.2.1.6. Recommendation on influencing use VI
Point-of-sale information should inform vehicle purchasers of the safety issues associated with in-vehicle information systems.

5.2.1.7. Recommendation on influencing use VII
Vehicle hire companies should ensure that all information and communication systems are maintained in accordance with the manufacturer's instructions.

5.2.1.8. Recommendation on influencing use VIII
Vehicle hire companies should ensure that a copy of the manufacturer's instructions for use is available in every equipped vehicle.

5.2.1.9. Recommendation on influencing use IX

Vehicle hire personnel should have adequate knowledge of in-vehicle information systems within the vehicles they make available and should offer instruction in their safe use.

APPENDIX G – NAUJOKS (BMW) EXTENDED CHECKLIST

#	Items
1	Unintentional activation and deactivation should be prevented Assessment + System design ensures driver readiness before transfer of control (e.g., pushing of two buttons simultaneously, need to have both hands on the steering wheel, need to have eyes on the road, etc.) Surprising or inexplicable driver-initiated activation/deactivation during regular usewww
2	The system mode should be displayed continuously + Minimum set of mode indicators present (1) functioning properly (2) currently engaged in an automated driving mode (3) currently unavailable for automated driving (4) experiencing a malfunction (5) requesting a control transition from the automated driving system to the operator Indicators missing Indicators not distinguishable from each other Indicators only displayed for short periods of time Mode indication discontinued (e.g., through pop-ups)
3	System state changes should be effectively communicated + Recognizable change of pictorial indicator + Auditory/haptic feedback + Communication of responsibility (e.g., by disclaimer) + Pop-up messages + Error messages are provided in case (e.g., failed activation) + Delayed reaction to control input displayed in HMI
4	Visual interfaces used to communicate system states should be mounted to a suitable position and distance. High-priority information should be presented close to the driver's expected line of sight + Important information displayed in 30 cone about normal line of sight + Safety-critical information displayed in 20 cone about normal line of sight + Peripheral displays support noticing of mode changes (e.g., movement or size of displays) + Status information mirrored on NDRT device
5	HMI elements should be grouped together according to their function to support the perception of mode indicators Unnecessary glances to retrieve information from display (e.g., to interpret a symbol and perceive accompanying text) + Indicators pertaining to the automation are grouped together + High priority messages are easily distinguished from low-priority messages
6	Time-critical interactions with the system should not afford continuous attention Important information is displayed too shortly (e.g., only for a few seconds)

	While the driver is responsible for the DDT, sustained attention (longer than 1.5 s) is needed to accomplish an interaction
7	The visual interface should have a sufficient contrast in luminance and/or colour between foreground and background + Sufficient colour and/or luminance contrast to identify different automation modes
8	Texts (e.g., font types and size of characters) and symbols should be easily readable from the permitted seating position + Displayed text and symbols are big enough to be easily readable + Display resolution is good enough to be easily readable + Character width and stroke width appear to be appropriate + Text-fonts are easily readable
9	Commonly accepted or standardized symbols should be used to communicate the automation mode. Use of non-standard symbols should be supplemented by additional text explanations + Commonly accepted or standardized symbols are used + Non-standard symbols are supplemented with a text label + The symbols are representative for the responsibility of the driver (e.g., displaying hands on a steering wheel in case of a hands-on request)
10	The semantic of a message should be in accordance with its urgency + Use of notification-style to present non-critical information + Use of command-style to present critical information + Wording in accordance with criticality of the situation (e.g., ‘‘caution”, ‘‘danger”, ‘‘warning”)
11	Messages should be conveyed using the language of the users (e.g., national language, avoidance of technical language, use of common syntax) + Use of national language + Use of simple language + Avoidance of abbreviations + Displaying functionality rather than SAE/NHTSA/BASt-level
12	Text messages should be as short as possible + Messages are as short as possible + Not more than four chunks of information are displayed
13	Not more than five colours should be consistently used to code system states (excluding white and black) + Colours are used consistently throughout an automated driving mode + Note more than five colours are used
14	The colours used to communicate system states should be in accordance with common conventions and stereotypes + Colours are in accordance with common stereotypes of the user population + Red = imminent danger, yellow/amber = caution, green = hazard-free operating state
15	Design for colour-blindness by redundant coding and avoidance of red/green and blue/yellow combinations + green/red and yellow/blue combinations are avoided + system states are redundantly coded in a suitable way

16	Auditory output should raise the attention of the driver without startling her/him or causing pain Generic auditory output + suitable length (100–500 ms) + suitable loudness (50–90 dB, should be 15 dB above background noise) + frequencies between 500 and 4000 Hz Vibrotactile output + suitable length (50–200 ms) + comfortable stimuli 15–20 dB above threshold + frequencies between 150 and 300 Hz
17	Auditory and vibrotactile output should be adapted to the urgency of the message Generic auditory output + Auditory output of varying urgency is distinguishably different by pulse rate, frequency or loudness + Low-priority information is either unobtrusive or without auditory output Vibrotactile output + Vibrotactile output of varying urgency is distinguishably different by pulse rates, intensities, vibrating area, etc. + Different information is coded through a variation of location and timing, not frequency and amplitude
18	High-priority messages should be multimodal + high-priority information is presented in more than one modality + auditory or vibrotactile stimuli are also visually presented
19	Warning messages should orient the user towards the source of danger + warning messages lead to an orienting response to the source of danger, causing the driver to look in the direction of the hazard + warning messages to not focus the driver's attention to a display
20	In case of sensor failures, their consequences and required operator steps should be displayed + unavailability of sub-systems because of sensor degradation is displayed + consequences of sensor degradation are displayed + required operator behaviors is displayed

Note:
Each item in the checklist can be scored as:

- "major concerns": definite non-compliance with guideline
- "minor concerns": partial fulfillment of guideline, but some aspects of the HMI are non-compliant
- "no concerns": compliance of all HMI aspects with guideline

In every case, measurements are necessary.