Design and Evaluation of a Virtual Reality Adaptive Driving Intervention Architecture (VADIA): Applications in Autism Spectrum Disorder

By

Joshua William Wade

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Computer Science

December, 2015

Nashville, Tennessee

Approved:

Nilanjan Sarkar, Ph.D.

Xenofon Koutsoukos, Ph.D.

Zachary Warren, Ph.D.

The presented work is dedicated to

my loving wife Caitlin,

my selfless grandparents Deborah and Lyndall,

my tenacious sisters Chelsea and Isabella,

and my dear friend Dalton.

Thank you all.

ACKNOWLEDGEMENTS

I want to first thank Dr. Nilanjan Sarkar for introducing me to the world of research and showing me that it can be both engaging and rewarding. Perhaps more than anyone else, Dr. Sarkar helped set me on a path that I would otherwise probably never have even considered. The journey along this path has proved to be one of the most gratifying experiences of my life. Thank you, Dr. Sarkar, for taking a chance on me.

Next I want to thank Dr. Zachary Warren for allowing me to be involved in numerous projects stemming from the collaboration between RASL and TRIAD. Dr. Warren's guidance has been invaluable during my time at Vanderbilt. I also want to express my sincerest gratitude to TRIAD members Amy Swanson and Amy Weitlauf for all of the hard work that they do.

I extend thanks also to Dr. Xenofon Koutsoukos for graciously serving on my defense committee. The knowledge that I gained during Dr. Koutsoukos's course on Hybrid and Embedded Systems has greatly influenced both the work presented in this thesis and the work that I will continue to do beyond this.

I especially want to acknowledge the wonderful graduate students of the Robotics and Autonomous Systems Lab, who have been excellent colleagues as well as very good friends. Specifically, thanks go to Esube Bekele, Jing Fan, Zhi Zheng, Dayi Bian, and Lian Zhang for frequent discussions, feedback, and support with this work and with others. I additionally want to thank Dr. Medha Sarkar for the great introduction to Computer Science, a field I now love.

Finally, I acknowledge the funding support of the National Science Foundation (grant 0967170), National Institutes of Health (grant 1R01MH091102-01A1) and Vanderbilt Kennedy Center (Hobbs Society Grant).

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

ADHD – Attention Deficit Hyperactivity Disorder

ADOS – Autism Diagnostic Observation Scale, 1st Edition

ADOS-2 – Autism Diagnostic Observation Scale, 2nd Edition

API – Application Programming Interface

ASD – Autism Spectrum Disorder

CDC – Centers for Disease Control and Prevention

EDM – EEG Data Acquisition Module

EEG – Electroencephalogram

FD – Fixation Duration

FSM – Finite State Machine

GC – Gaze-contingent Operating Mode

GDM – Gaze Data Acquisition Module

HA – Hybrid Automaton

HF-ASD – Higher-functioning Autism Spectrum Disorder

HSM – Hierarchical State Machine

JSON – JavaScript Object Notation

LAN – Local Area Network

MoC – Model of Computation

OAM – Observer-based Assessment Module

PB – Performance-based Operating Mode

PDM – Physiological Data Acquisition Module

ROI – Region of Interest

SCQ – Social Communication Questionnaire

SRS-2 – Social Responsiveness Scale, 2nd Edition

TD – Typically Developing

US – United States

VADIA – VR Adaptive Driving Intervention Architecture

VDM – Virtual Reality Driving Module

VR – Virtual Reality

CHAPTER 1

INTRODUCTION

1 Driving in the United States

Injury, loss of life, and tremendous financial tolls have for years plagued drivers in the United States (US) and in nearly every industrialized nation in the world [1, 2]. Although the US has seen a decrease in the number of driving-related deaths in the last 30 years, vehicle crashes unwaveringly remain the most common cause of death for American teenagers [1-4]. The most recent statistics on teenage automobile accidents from the Centers for Disease Control and Prevention (CDC) report 2,163 fatalities and over 240,000 emergency room visits related to vehicle crashes in 2013 [4]. The number of emergency room visits is highest among the age group 16-24 years, suggesting that teens are not the only demographic susceptible to the dangers of driving [5]. This group is also responsible for a disproportionate amount of the monetary cost of vehicle crashes with an annual price tag of $19 billion [4].

2 Driving Intervention

Despite the associated risks, driving remains an essential part of life for over 200 million Americans [6]. Autonomous transportation is one critical component of functional independence; it supports other aspects of quality of life in adulthood, such as vocational and social endeavors. Given the necessity of driving, researchers and policymakers have sought ways to improve the roadway situation. The recent criminalization of texting-and-driving in many states is the latest in a series of juridical attempts to improve driving safety, following mandatory seatbelt wearing and severe punishments for driving under the influence of alcohol. There is also a long tradition of using technology to understand and improve driver behavior [1]. As early as the 1970's, eye trackers were embedded in actual vehicles to monitor drivers' patterns of gaze on the road [1, 7], and driving simulation systems were already being employed to observe and challenge drivers in

safe, controlled environments [1]. Today, heads-up displays and augmented reality promise added safety for the driver by limiting his/her distractions.

Research on highly specialized driving interventions have begun to emerge [8-14]. These systems target specific deficits affecting driving performance and aim to improve performance through various means. Fletcher and Zelinsky [14] developed a gaze-sensitive driver assistance system to augment driver awareness in real world driving environments. They sought to minimize inattention by alerting drivers via audio feedback to objects determined not to have been seen by the drivers. They demonstrated the feasibility of the system to identify road objects (such as signs and pedestrians) using eye-tracking technology coupled with sophisticated image processing software [14]. Rezaei and Klette [13] used two cameras in real world driving scenarios to obtain several features of the environment and the driver including distance from other vehicles, head orientation, yawning, and head-nodding—the latter two being indicators of drowsiness. They showed that this system could reliably detect whether a driver was looking at relevant vehicles in the environment [13]. Ho et al. [15] were also interested in reducing driver inattentiveness, but employed a warning mechanism based on vibrotactile feedback rather than audio or video. Using commercial simulation software, they designed a system that activated a vibrotactile belt worn by drivers when the VR vehicle was either too close to a leading vehicle or too far behind. Vibration was spatially applied to the front (i.e., the stomach) when drivers were too close and in the rear (i.e., the lower back) when drivers were too far behind. They found drivers using this system braked earlier when a leading vehicle decelerated and left more space between themselves and the leading vehicle when stopped [15].

All of the aforementioned intervention systems aim to shift human behavioral deficits that are superficial and alterable. Another type of driving intervention system is one that aims to change innate characteristics of the driver. For example, individuals with Attention Deficit Hyperactivity Disorder (ADHD) have become the focus of a growing body of driver research because they demonstrate behaviors that may be unconducive to safe driving [8-12]. Indeed, individuals with ADHD have been shown to perform more poorly than controls on driving performance tests [12]. While driving performance in individuals with ADHD is still a developing area of research that warrants further investigation, there is another population of drivers researchers know even less about: those with Autism Spectrum Disorder (ASD).

3 Autism Spectrum Disorder

ASD refers to a complex neurodevelopmental disorder characterized by impairments in social interaction and communication as well as restricted, repetitive patterns of behavior and interest [16]. Current prevalence rates estimate that 1 in 68 children in the US have ASD [17]. Although ASD is considered a lifelong diagnosis, much of the current research regarding intervention for ASD focuses on early childhood [18, 19]. However, adults with ASD often have a difficult time meeting educational and vocational goals that driving independence might support. Howlin et al. [20] demonstrated that a minority (23%) of adults with ASD achieve "good" to "very good outcomes," such as paid employment, friendships, and some independence. A majority (58%) achieved "poor" to "very poor" outcomes and remained highly dependent on their families and social services [20]. Similarly, Shattuck et al. [21] indicated that more than 50% of individuals with ASD do not access education or employment in the two years following high school. A targeted intervention that promotes driving independence may provide an avenue for some adolescents and adults with ASD to more easily achieve educational and vocational goals.

A small, but growing body of research has begun to focus on the driving behaviors of individuals diagnosed with ASD [8, 22-30]. This work seems generally to suggest that (1) driving is more challenging for people with ASD, (2) these individuals may experience a greater risk to personal safety while driving than their typically developing (TD) peers, and (3) driving as it relates to people with ASD is highly under-researched. Driving intervention programs often cannot be realistically implemented in real-world driving environments due to the high risk of injury and obvious associated costs. The Virtual Reality (VR) paradigm, however, has the capacity to produce ideal environments that are harmless, controlled, and able to collect a wide variety of data from both drivers and their environment. Such environments are not only capable of operating off of simple performance paradigms, but also hold the potential for integrating other aspects of information processing in the environment in order to optimize learning (e.g., eye gaze, physiology, etc.).

4 The Need for VADIA

A growing body of work suggests that driving is a challenging skill for individuals with ASD and this challenge appears to be a direct result of how individuals with ASD process information in the driving environment. If this is the case, then effective interventions must incorporate methods for altering information processing, not merely focus on repeated exposure without processing support (e.g., performance systems). A primary reason why many current intervention approaches show limited improvements in functional adaptive skills may be that traditional skill-based methodologies often fail to systematically match intervention strategies to specific underlying processing deficits associated with targeted skills. The hypothesis, therefore, is that in order to address the driving skill deficit of individuals with ASD, one would need to design an intelligent driving simulator than can (1) have embedded rules geared specifically towards ASD intervention, (2) provide individualized tasks and feedback to improve driving outcomes, and (3) be integrated with a host of sensors such as an eye tracker and physiological and EEG sensors to measure features of the driver and create dynamic closed-loop interaction. Further, it is hypothesized that realization of an autonomous system capable of providing real-time gaze-contingent feedback would contribute to both enhanced performance within the driving environment (e.g., fewer driving errors) as well as shape how individuals with ASD were scanning the relevant objects in the environment (e.g., alterations in gaze). Although a range of high quality, off-the-shelf driving simulation software exists that can be used to assess driving behaviors, these tools do not provide necessary access to the source code in order to design closed-loop systems that utilize information from both driving performance *and* internal driver state. They thus do not satisfy the above-mentioned requirements for ASD intervention. As a result, commercial driving simulators were not deemed appropriate for this work.

This thesis describes the development and application of a **V**R **A**daptive **D**riving **I**ntervention **A**rchitecture (henceforth referred to as VADIA). VADIA is a VR-based, real-time gaze-contingent driving simulator capable of providing individualized feedback about how drivers scan their visual environment while driving. This system was designed to evaluate participant responses across two feedback modalities: (1) strictly performance-dependent feedback and (2) performance- and gaze-contingent feedback. The intuition behind the second modality is that, although performance is

certainly an important factor in developing driving skills, it is just as important that drivers demonstrate appropriate gaze patterns while behind the wheel.

5 Related Work

Relatively little research specifically investigates driving in the population with ASD [8]. However, since many individuals with ASD display affinity for non-biological motion [31] and may find it difficult to attend to multiple stimuli during driving tasks [28], understanding their driving behaviors and aiding them in their driving skills requires further research. Much of the work that has been done in this area has not utilized technology effectively to obtain objective, quantitative information about these individuals' driving behaviors. Sheppard et al. [30] recruited 23 participants with ASD and 21 TD controls—all non-driving adult males—and showed them videos of hazardous driving scenarios that contained either a social hazard (e.g., a pedestrian) or a non-social hazard (e.g., and automobile). They found that individuals with ASD identified social hazards significantly less often than the controls and were slower than controls overall in identifying hazards [30]. Cox et al. [28] surveyed the parents of adolescents and young adults with ASD about their experiences with their children learning to drive. The results of their survey indicated that a majority of parents were generally very concerned about the safety of their child while driving because of their child's ASD and parents tended to feel that multitasking (e.g., managing speed while entering a highway) was a particularly problematic skill for their child [28]. Another survey-based study conducted by Daly et al. [26] questioned licensed driving adults with and without ASD about their driving histories. They found that individuals with ASD reported being older at the age of licensure, spending less time driving, feeling less confident about their driving abilities, and experiencing a higher number of traffic violations than their TD peers [26].

Very few empirical studies have investigated a driving simulation paradigm to specifically assess the driving performance of individuals with ASD—two of which are the previous work of the present author [22-24, 27]. Classen et al. [22] used proprietary driving simulation software to observe the between group distinctions of seven adolescents with ASD and 22 TD controls. They found that subjects with ASD demonstrated significantly more driving errors than controls. Reimer et al. [27] also used proprietary driving simulation software to analyze the performance of young adult males diagnosed with higher-functioning ASD (HF-ASD) as compared to controls. In

addition, this study observed physiological signals (i.e., heart rate and skin conductance) as well as eye gaze from participants. They found heart rate and skin conductance levels were nominally higher in the HF-ASD group suggesting higher levels of anxiety while driving. Furthermore, compared to controls, the pattern of gaze in the HF-ASD group tended to be higher vertically and more the right on the screen on which the VR driving environment was projected [27]. Similar patterns of gaze and increased anxiety were reproduced in the author's previous work [24], which is described fully in Chapter 4 Section 2. In this pilot study, the driving performance of individuals with ASD was compared to TD controls using a novel driving simulator. The results showed that individuals with ASD displayed significant physiological differences from the TD controls with respect to skin conductance, likely indicating higher levels of anxiety while driving. The same pattern of gaze reported by the Reimer et al. [27] study was present, and additionally, the ASD group demonstrated a significantly higher number of driving errors compared to their TD peers [24].

6 Contributions of Thesis

The contributions of this thesis are:

(1) The design and implementation of a novel adaptive VR driving simulation platform for ASD intervention that combines real-time gaze monitoring and gaze-contingent feedback to the driver in the virtual environment
(2) The presented architecture supports integration of sensor feedback in a closed loop manner, which have been utilized to design an intervention program based on eye gaze information
(3) VADIA has been validated through two distinct pilot studies and findings suggest that the system is functional and robust, and shows promising results as a driving intervention tool

7 Structure of Thesis

This thesis is a comprehensive assemblage of three published conference papers [23-25] and two submitted-for-review journal manuscripts. The content of these documents has been restructured in this thesis in order to provide a clearer, more cogent representation of VADIA and

the experiments conducted. Considering this, the rest of the thesis is structured as follows: In Chapter 2, the modeling and implementation of VADIA is presented in detail. Chapter 3 discusses the two feedback modalities employed within VADIA (i.e., the performance-dependent and gaze-contingent modalities) and includes details on the real-time gaze monitoring system. In Chapter 4, the design and results of our various experiments are presented. These include a range of tests used to validate the robustness of the system, as well as two separate pilot studies involving 34 human subjects with and without ASD. Finally, Chapter 5 concludes with a review of the contributions of this work as well as the limitations of VADIA and planned future work with the architecture.

CHAPTER 2

VADIA DEVELOPMENT

VADIA is comprised of a set of interacting modules in which the central member is the VR driving module (VDM). In addition to the VDM, VADIA contains the following modules: the gaze data acquisition module (GDM), the physiological data acquisition module (PDM), the electroencephalography (EEG) data acquisition module (EDM), and the observer assessment module (OAM). Fig. 1 gives a block diagram of the architecture. VADIA can be configured to operate with various combinations of modules depending on the intervention objectives. In addition, some modules may disabled on-the-fly under special circumstances. For example, the EDM can be disabled without impacting the rest of the system in cases where the participant refuses to wear the EEG device. This chapter discusses in detail VADIA's current modules identifying each module's role within the overall architecture as well as important features of the module.

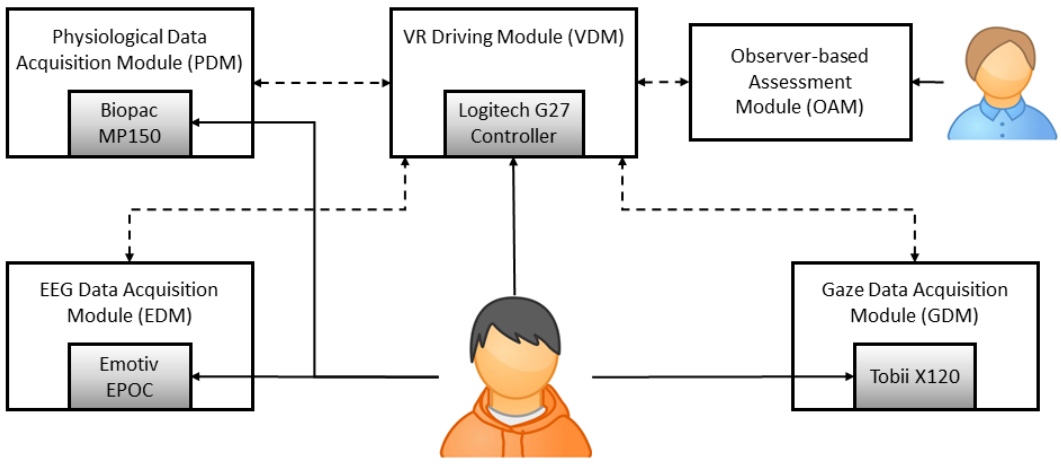

Figure 1. VADIA block diagram. The solid lines indicate human input and the dashed lines represent network communication channels.

1 Models of Computation

Formal models of computation (MoC) are tools used in the development process to design software systems in a way that compartmentalizes the system's behaviors and responsibilities (refer to Appendix A for a thorough review of MoC). There are many reasons to suggest modeling systems using a formal MoC. Because a formal MoC has an established syntax and semantics, it provides a framework for modeling that can be followed by anyone. Software often requires maintenance in order to fix a bug or extension to perform some new task. For very complex systems, these can prove to be major challenges. A model-centric approach to design may reduce the difficulty of these tasks by allowing developers to more easily identify the components of the system that require change as well as to see how the changes will impact the overall system. Additionally, the graphical representation of a MoC may serve as both documentation of a system's behavior and a useful tool for explaining how a system works to others. Finally, a formal MoC is amenable to analysis, which allows developers to make guarantees about how a system will execute in a target environment. Quantitative analyses may be carried out to assess the worst-case execution time of a piece of code or the throughput rate of a network component. Formal verification methods such as model checking have also been heavily researched and can be used to prove that a system will exhibit certain properties under all possible circumstances.

MoCs have been primarily applied in the context of Embedded Systems development, but recently they have begun to gain interest in other domains as well [32, 33]. Many different types of MoCs exist and the choice of which to use depends on the intended behavior of the system. Some examples of MoCs include finite state machines (FSM), statecharts or hierarchical state machines (HSM), discrete events systems, hybrid automata (HA), petri nets, and dataflow networks [34]. Within each of these types of MoCs, there are many recognized variations, each with its own formal syntax and semantics. In this work, a mixture of FSMs, HSMs, and HAs are used to model various components of VADIA's modules. These are discussed in more detail in the next section.

2 Implementation

This section discusses each of VADIA's modules in depth including their role in the overall architecture, associated hardware interface (if applicable), software modeling approach taken, and implementation details of note.

2.1 VR Driving Module (VDM)

The fundamental module of VADIA is the VDM. The VDM consists of both VR driving simulation software and a hardware driving interface that allow users to perform driving tasks in a controlled environment. The simulation system was designed such that users could engage in a variety of meaningful driving tasks and that parameters of the tasks such as complexity, difficulty, and length could be explicitly controlled. Drivers could interact with traffic lights, pedestrians, and other vehicles while completing tasks and a built-in navigation system aided drivers in reaching their respective destinations. Behaviors were defined for autonomous vehicles and pedestrians in order to make the tasks more naturalistic. The system was designed in such a way that tasks could be completed or failed and either outcome would produce some appropriate feedback to the drivers.

The first step towards developing the virtual driving environment was to create a 3D model city. Using CityEngine software (www.esri.com/software/cityengine), a model city was constructed that was sufficiently large to allow the creation of driving tasks lasting up to a few hours. This model consisted of diverse regions including a large downtown area with skyscrapers, a residential community, an industrial park, and a large arboreal region. A wide variety of roads were represented including narrow one-way streets, multi-lane highways, and sharp turns. Next, the city was appropriately populated with typical objects such as vehicles, pedestrians, traffic lights, and signs. Vehicle and pedestrian objects were obtained from free online repositories of 3D models. The rest of the required models were created using Autodesk Maya software (www.autodesk.com/products/maya).

The Unity3D game development platform (or simply "Unity") was selected to implement the VR driving application. Unity was a good option for development for a number of reasons: (1) it supports a variety of high level programming languages, (2) has a robust physics application

programming interface (API), (3) is available in both free and paid versions, and (4) generates executables on all major operating systems. Additionally, Unity natively interfaces with many input devices including the Logitech G27 driving controller, which was used in the present work. The G27 controller includes a steering wheel with customizable buttons, a pedal board with three independent pedals, and a gear shifter which was not used in the studies. Fig. 2 shows both a screenshot of the driving perspective and the G27 controller in use.

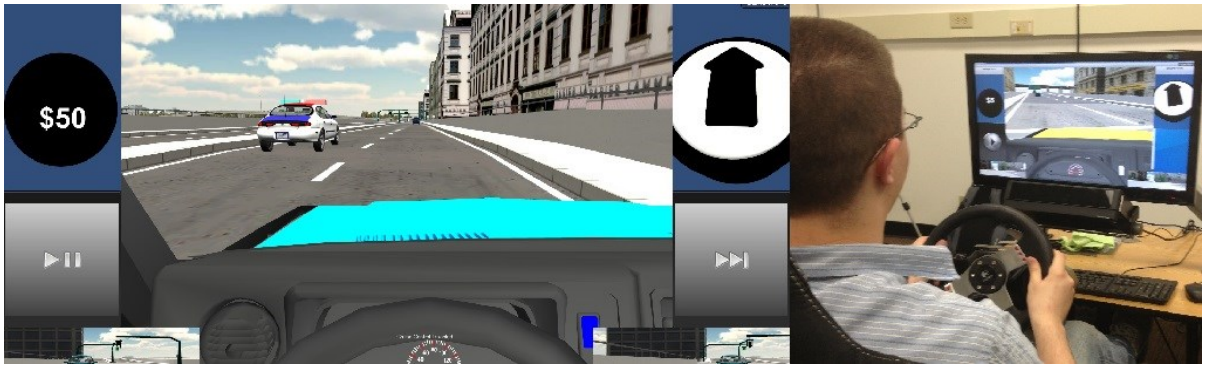

Figure 2. The VDM interface: driver perspective (left) and Logitech G27 controller (right).

Table 1. Major Components of the VDM Software

Name	Purpose
GameMonitor	Supervisory controller manages highest level of application logic such as menu navigation (e.g., vehicle selction), initiating network communication, and beginning driving tasks.
TrialMonitor	Initializes trials and monitors failure and success events.
PathMonitor	Manages placement of driver's vehicle at trial start and trial reset after a failure.
GazeMonitor	Maintains hashtable of fixation duration information.
SoundMonitor	Schedules various audio clips used in the simulator (e.g., congratulatory sounds for successful trial completion).
DataManager	Logs user performance data as well fixation duration data and metadata.
FeedbackModule	Manages the presentation of feedback within the system.
EyeTrackerModule	Handles network communication with the GDM and provides real-time gaze information to the GazeMonitor component.
PhysioModule	Handles network communication with the PDM.
EEGServer	Handles network communication with the EDM.
QServer	Handles network communication with the OAM.
VehicleManager	Updates the driver's vehicle state based on the driver's input via the G27 controller.
AIVehicle	Controls the autonomous vehicles in the driving tasks.
AIPedestrian	Controls the autonomous pedestrians in the driving tasks.
TrafficLightDisplay	Manages the scheduling of traffic light signals.
ExcessiveSpeedFSM	Detects driving failures due to exceeding the speed limit.
OffRoadDetectionFSM	Detects driving failures due to driving off the road (i.e., driving too far onto the shoulder or onto the grass).
IncorrectLaneFSM	Detects driving failures due to driving in the wrong lane.
RidingSidewalkFSM	Detects driving failures due to driving onto the sidewalk.
PassingFSM	Detects driving failures that result from a driver either not passing a vehicle when it is required or passing in an inappropriate way.
RunStopSignFSM	Detects driving failures related to failing to stop appropriately at a stop sign.
RunRedFSM	Detects driving failures related to running a red light at a traffic light intersection.
GpsController	Manages the navigation system used by the driver to complete tasks.

The software of the VDM is made up of several major components (see Table 1 for a description of some of these). At the low level, there are components such as *TurnSignalControl* and *TrafficLightDisplay* that define the behaviors of simple elements of the driving environment. More complex components dictate the behaviors of semi-autonomous objects such as *AIPedestrian* for

pedestrians and *AIVehicle* for vehicles. There are other components like *EEGServer* and *QServer* dedicated to communicating over the network with VADIA's other modules. At the highest level, the supervisory controller component *GameMonitor* manages the lower level components, synchronizes them, broadcasts events, and effectively dictates the flow of the system. Fig. 3 gives a view of the hierarchy among the VDM's major components. Having control over all of these elements of the driving environment was key in developing the embedded rules of the system. For example, traffic light signals and autonomous vehicle behaviors were manipulated to create a range of task complexity that was suitable to drivers across a spectrum. Additionally, feedback presentation mechanisms suitable to individuals with ASD were incorporated, such as object-highlighting and audio/text presentation. A full description of each of the VDM's components is well beyond the scope of this thesis and also not necessary. Instead, the discussion is focused on the supervisory *GameMonitor* component in order to demonstrate the application of model-centric design and to show that it is also appropriate for software design outside the context of Embedded Systems.

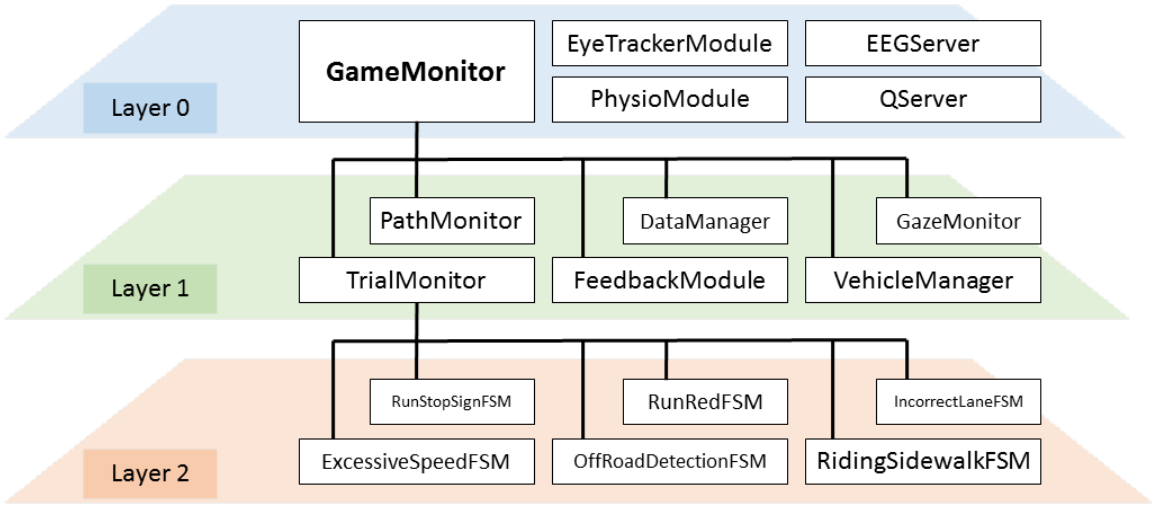

Figure 3. Hierarchical view of some of the major components of the VDM software. Layer 0 is populated by concurrent, independent components such as the supervisory controller and network components while layers 1 and 2 are subordinate to the layers above them.

For the automata presented here, discrete modes or states are represented as rounded rectangles and transitions between states are given as arcs. Transition arcs typically labeled using the format *guard / actions* where *guard* indicates the set of conditions under which the transition is valid and *actions* represents a set of tasks to be performed when the transition occurs. Such actions may include setting the value of a variable or generating an event. States are labeled with a descriptive name and, in the case of HA, include a set of ordinary differential equations that describe the behavior of the continuous variables in that state.

The following desired behavior was specified for the *GameMonitor* component: (1) handle the decision to enter practice mode or assignment mode; (2) manage the selection of vehicle by users; (3) manage the choice of level and assignment by users; (4) initiate the start of an assignment when the user is ready to begin; (5) provide the option to return to the previous state of menu selection in case the user made a mistake in selecting options; (6) keep track of trial success counts and trial failures counts; and (7) schedule the presentation of feedback. Since the behavior described in this specification deals exclusively with discrete events and variables with finite domains, a FSM (shown in Fig. 4) was chosen to model this behavior. Keep in mind that other components such as *EyeTrackerModule* and *EEGServer* were in the same hierarchical layer as *GameMonitor* and so network-related logic is not the responsibility of this particular FSM.

The initial state of this FSM is labeled *Choose Practice or Assignment* and offers the user the choice to run the system in either practice mode or regular assignment mode before allowing the user to select a vehicle to operate. Depending on the user's choice in the initial state, the FSM transition to either the state *Display Practice Mode Instructions* where the user will confirm the start of practice mode, or the state *Select Level* where the user will choose the difficulty level to play. When the user chooses the assignment path, the appropriate assignment is loaded and executed in the state labeled *Assignment In Progress*. It is in this state that the *TrialMonitor* component of the VDM is enabled (see Fig. 3) and begins presenting driving tasks to the user. Because *TrialMonitor* is only active when the *GameMonitor* is in this state, the *TrialMonitor* component is said to be subordinate to the *GameMonitor* component, hence the 'hierarchical' state machine. While the assignment is in progress, the *TrialMonitor* may generate events indicating that a driving task has been either failed or completed successfully. When either of these events occur, the *GameMonitor* transitions to the appropriate next state: to the *Present Failure Feedback*

state when a task is failed, back to the *Assignment In Progress* state if the trial is completed but there are remaining trials to complete, or to the *Present Summary Feedback* state if the trial is completed and there are no remaining trials left to complete. The author has opted not to include some details of the model in Fig. 4 such as self-loops and transition actions, but this model accurately represents the specification outlined in the previous paragraph. Each MoC produced for components in the VDM system was transformed to software code using object-oriented, automata-based programming techniques (see Appendix B for details and examples). In all, over 30 formal models were mapped to source code as part of the VDM's implementation.

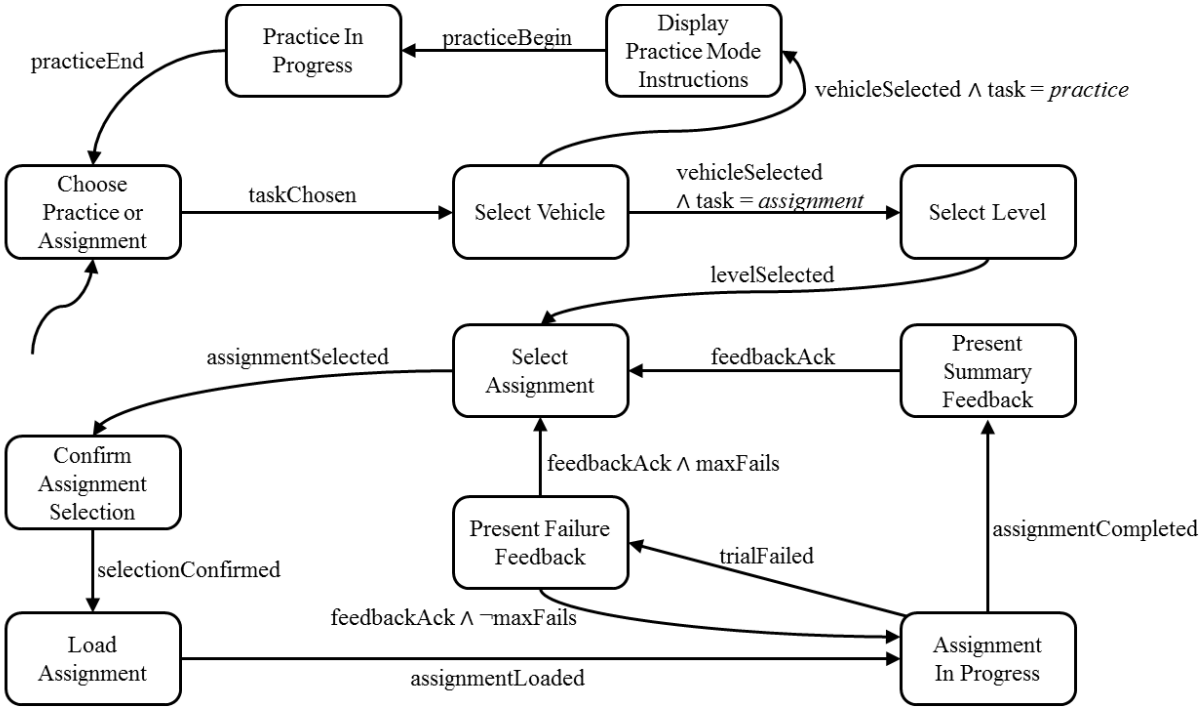

Figure 4. FSM model of the GameMonitor component of the VDM. Note that self-loops, transition actions, and arcs for go-back options are excluded for simplicity.

2.2 Gaze Data Acquisition Module (GDM)

The GDM allows VADIA to obtain eye gaze information from the driver. This information can be logged for offline analysis or utilized in a closed-loop system that uses gaze data to generate individualized driving feedback as described in Chapter 3.

The Tobii X120 remote eye tracking device (www.tobii.com) was used to extract eye gaze data from users at a sampling rate of 120 Hz. The X120 can achieve high accuracy and allows the user the freedom of small amounts of head movement in the range 30 × 22 × 30 cm (w × h × d) [35]. The Tobii software API provides developers access to a variety of information about a subject's gaze including pupil diameter, blink rate, and gaze position on a calibrated surface (e.g., a monitor or project screen). Pupil diameter and blink rate have been shown to be physiological indicators of a person's internal state and can reveal an individual's level of anxiety or engagement while gaze position may yield information about an individual's attention [36, 37].

An existing device interface application was modified to allow communication between the GDM and VDM. A simple FSM was created to model the pairwise network communication logic with the VDM and was implemented within the application. This software performed device calibration onto a 24" monitor (1920 × 1080 resolution), logged the subject's data locally, and transmitted relevant data to the VDM over the network. The subject's gaze position information was used by the VDM for online processing while pupil diameter and blink rate were logged solely for offline analysis.

2.3 Physiological and EEG Data Acquisition Modules (PDM and EDM)

The PDM and EDM acquired physiological and EEG data, respectively, from the driver for the purposes of offline analysis. Their primary roles in the presented work were to develop models of drivers' affect for a series of concurrent studies [38-41]. Both modules were connected to the VDM via a local area network (LAN) and received event messages from the VDM for the purpose of aligning signals with respect to important driving events. These modules are discussed here as part of the presentation of VADIA, but this thesis is not concerned with the development of a driver affect model.

The Biopac MP150 was used for physiological data acquisition (www.biopac.com). This system wirelessly sampled several physiological signals from the driver at 1000 Hz. Signals of interest included electromyogram activities from corrugator supercilii, zygomaticus major, and upper trapezius muscles; electrocardiogram; impedance cardiogram; phonocardiogram/heart

sound; galvanic skin response; photoplethysmogram; skin temperature; and respiration. These signals were selected because of their ability to predict the affective state of the subject [42].

For EEG data acquisition, the Emotiv EPOC headset was used (www.emotiv.com). This device was used to sample from 14 locations on the wearer's head at 128 Hz. The data was streamed wirelessly from the headset to a custom PC-side application via a proprietary dongle. This application logged the EEG readings as well as signal quality and rotational velocity information collected from a gyroscope within the device.

2.4 Observer-based Assessment Module (OAM)

As mentioned in the preceding section, the roles of the PDM and EDM were simply to log data regarding the internal state of subjects in order to derive a model of driver affect. In order to achieve this, the OAM was developed to provide a means for specifying the ground truth of subject affect by allowing an observer to label epochs of data in a meaningful way. As with the PDM and EDM, the OAM is described here for completeness, but the results of the affect model training are not within the scope of this thesis.

The OAM was designed for both online and offline use. In online mode, an observer could attend a driving session and make assessments about a driver's state in real time. These assessments were labeled with timestamps that could be mapped to data collected from the PDM and/or EDM. Video recordings of the subjects' faces and the driving were collected during driving sessions. Using these videos, which included a timestamp overlay, observers were able to use the OAM to code driving sessions offline as well.

The OAM prompted observers for five categories of input. Four measures of subject affect—engagement, enjoyment, boredom, and frustration—and an additional measure of perceived difficulty of the task for the subject were collected from the observer. Each category could be rated on a continuous scale in the range [0, 9], where 0 indicated the lowest intensity, by dragging a slider with a mouse. Observers were permitted to make an assessment at any time during the driving sessions; these were referred to as *periodic* assessments. There were also *summary* assessments that were made by the observer at the end of driving assignments. Assignments are described in detail in Chapter 3.

2.5 Network Communication

VADIA uses a star network topology in which the VDM is the central node. This is not strictly necessary, but was convenient for the conducted studies. All modules required a connection with the VDM, but not to other modules. For example, the PDM had no need to interact directly with the OAM, nor did the GDM have any reason to communicate with the EDM, etc. However, such functionality may be desirable in future studies and therefore is not prohibited within VADIA. All nodes in the network communicated via TCP/IP and used object serialization methods, including JavaScript Object Notation (JSON), to send objects as messages between nodes. The physical nodes consisted of three computers: the first executing the VDM and GDM software and interfacing with the G27 controller and eye tracker, the second running the PDM and EDM software and interfacing with the Biopac system and EPOC headset, and the third running the OAM software. Other assignments of processes to hardware were possible, but this configuration yielded adequate system performance.

CHAPTER 3

INTERVENTION MODALITIES

Two distinct modes of operation were defined for VADIA: a performance-based mode and a gaze-contingent mode, henceforth referred to as PB and GC modes, respectively. A configuration file is used to specify the mode of operation employed by VADIA at runtime. The PB mode of operation presents users with driving tasks in which progression depends entirely on their performance within the system. The GC mode of operation also enforces proper driving performance, but very importantly, it also requires that drivers demonstrate particular patterns of gaze while driving. Each of these modes is described in greater detail in the following sections.

1 Performance-based System

The driving scenarios used to evaluate driver performance were called *trials*. Four classes of trials were defined: turning, merging, speed-maintenance, and laws. Turning trials consisted of all those trials in which the participant made a left or right turn at an intersection. This implies a street change and does not include driving on sharply curved roads. Merging trials were characterized by any scenario in which drivers either passed another vehicle or entered/exited a highway. Speed-maintenance trials were those in which drivers were required to modify their speed to comply with the changing environment. Active school zones and areas of road construction are examples of scenarios in which the speed limit may change and drivers should adjust their speed accordingly. The last category, laws, dealt with an assortment of driving scenarios such as waiting for a school bus to unload and stopping at stop signs, which require drivers to know specific road laws.

A set of eight trials were assembled together, one after the other, into larger tasks referred to as *assignments*. The number eight was chosen because, on average, it produced assignments of desirable duration (roughly five minutes). Three assignments were grouped together into *levels*. In all, six levels were developed (i.e., 18 assignments or 144 trials). The levels increased in difficulty from level one (the easiest) to level six (the hardest). In order to implement six difficulty levels, a

set of difficulty parameters was defined that could be appropriately tuned to fit the desired difficulty settings. Table 2 describes the parameters chosen. The functions of most of the parameters should be evident by their descriptions, but a few are mentioned here in passing. When the parameter H_s is enabled, drivers will be alerted when relevant traffic lights change from red to green. The parameter L controls the intensity of lighting in the environment (i.e., it can be brighter or darker affecting the driver's visibility). The parameter S_d indicates how long drivers can drive on a sidewalk without a penalty; sometimes drivers accidentally drive onto the sidewalk and so a kind of 'forgiveness' factor was quantified.

Table 2. Level Difficulty Parameters

Name	Description	Domain
A_s	Speed of autonomous vehicles, a scalar	$\mathbb{R}$
A_a	Aggressiveness of autonomous vehicles, a scalar	$\mathbb{R}$
H_s	Traffic light alert sound	{enabled, disabled}
R_b	Responsiveness of the brake pedal, a scalar	$\mathbb{R}$
R_a	Responsiveness of the accelerator pedal, a scalar	$\mathbb{R}$
R_s	Responsiveness of the steering wheel, a scalar	$\mathbb{R}$
W	Weather condition state	{sunny, overcase, rainy}
L	Light intensity value, a scalar	$\mathbb{R}$
N_v	Number of vehicles at intersections/hwy entrances	{1, 2, … , 5}
S_d	Duration of time to permit driving on sidewalk, in seconds	$\mathbb{R}$

Trial failures and successes are met with feedback from the system. When drivers successfully completed a trial, a money counter (an arbitrary scoring system) increased by $5 and a congratulatory audio clip played (the obligatory cash register sound). When an entire assignment was completed successfully, drivers were presented with a congratulatory feedback message (e.g., *Great job driving! Get ready for the next assignment!*). On the other hand, when trial failures occurred, a feedback window would appear on the screen with a text message and corresponding

audio explaining what drivers did wrong and how to correct it moving forward (see Fig. 5). Following trial failure events, trials were reinitialized so that drivers could make subsequent attempts at completing the failed trials. The number of reattempts permitted per assignment was three, and a fourth failure would result in the termination of the assignment.

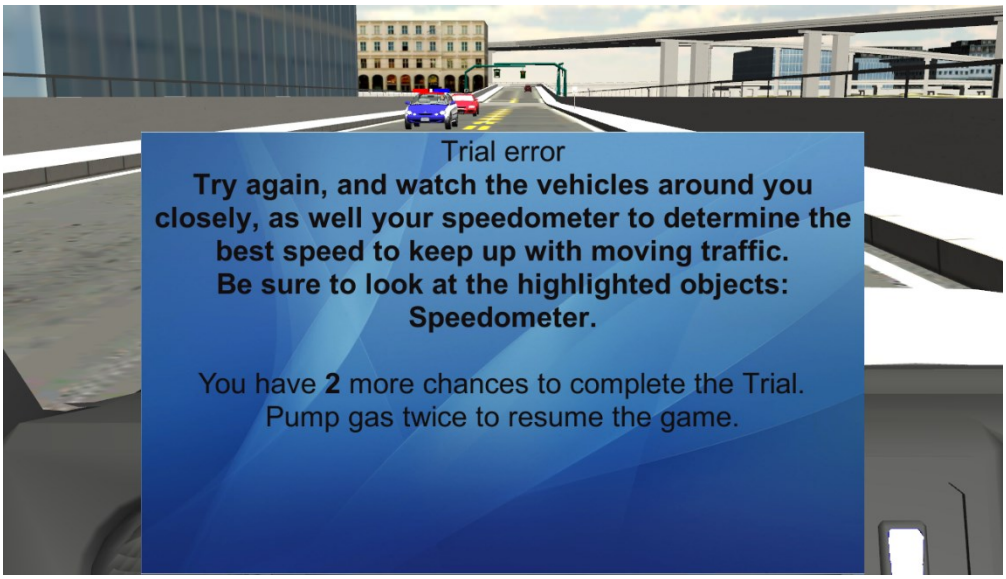

Figure 5. A feedback message appears after a driver fails to look at the vehicle's speedometer during a speed-maintenance trial.

2 Gaze-contingent System

In essence, the GC mode required that drivers look at specified regions of interest (ROI) during trials in order to progress through tasks. Failing to look at these ROIs would result in a *gaze failure* as opposed to a *performance failure*. For trial failures that occurred in the GC mode, the feedback message would include the names of the ROIs that the driver should be looking at (e.g., traffic light, stop sign, left side view mirror, etc.). In PB mode, three trial failures per assignment were allowed and the fourth resulted in the termination of the assignment without a chance to reattempt. In the GC mode, however, three trial failures were granted for both categories (i.e., three performance failures *and* three gaze failures) and a fourth trial failure in *either* category resulted in the termination of the assignment without a chance to reattempt. When a trial was reset after a

gaze failure, all of the ROIs relevant to that trial were highlighted with a green light to draw the driver's attention. Fig. 6 shows an example of a scenario in which the ROIs become highlighted. In this case, the driver did not look at one or both of the male pedestrian and pedestrian crossing sign, and thus are repeating the trial with the ROIs highlighted.

Figure 6. Example of ROI highlighting after a gaze-related trial failure. Both the male pedestrian and the pedestrian crossing sign have become highlighted with a green light to draw the attention of the driver.

2.1 Selection of ROIs

The set of relevant ROIs selected for each trial were purposefully chosen on a trial-by-trial basis. Each of the 144 trials were exhaustively evaluated and a list of only the most crucial ROIs were identified as being key to the successful completion of each particular trial. The basic inclusion criterion for an object to be selected as a ROI was that the act of observing the object must be essential to safe driving. That being said, most trials in the same category (i.e., turning, merging, speed-maintenance, and laws) had very similar sets of relevant ROIs. In every speed-maintenance trial, for example, the relevant set of ROIs included at a minimum the vehicle's speedometer and the nearby speed limit signs. Without observing these two objects, drivers cannot

be demonstrating truly safe driving behavior because they either (1) do not know the speed at which they are travelling or (2) do not know the current speed limit. Similarly, for trials requiring drivers to stop at pedestrian crossings, drivers must always be aware of both the crossing zone (i.e., the pedestrian crossing signs) and the pedestrian. Failing to notice one or both of these objects does not reflect optimally safe driving.

Table 3. Typical Selection of ROIs by Trial Category

Category	**Typical Regions of Interest (ROIs)**
Turning	oncoming traffic, traffic lights, flashing lights, stop signs
Merging	side view mirrors, yield signs
Speed-maintenance	speedometer, speed limit signs, road work signs
Laws	pedestrian crossing signs, pedestrians, school bus stop sign, stop signs

A comprehensive listing of the ROIs selected for each trial is not feasible. Instead, Table 3 identifies the typical choice of ROIs by trial category. In a few cases, multiple virtual objects constituted the same ROI. For example, yield signs were located on both sides of highway entrance ramps. Rather than requiring drivers to observe *both* of these objects, one or the other was deemed sufficient. Similarly, if multiple speed limit signs were present along a stretch of road, then drivers need only look at one of them to satisfy the gaze requirement.

2.2 Online Eye Gaze Monitoring

Tracking drivers' eye gaze in order to know what objects they are looking at while driving is an essential function of the GC operating mode. As such, an algorithm was needed that would accept drivers' gaze as input and in turn report the amount of time drivers had spent looking at various ROIs. The term *fixation duration* (FD) is used to indicate the amount of time that a driver spends looking at a particular ROI during a driving trial.

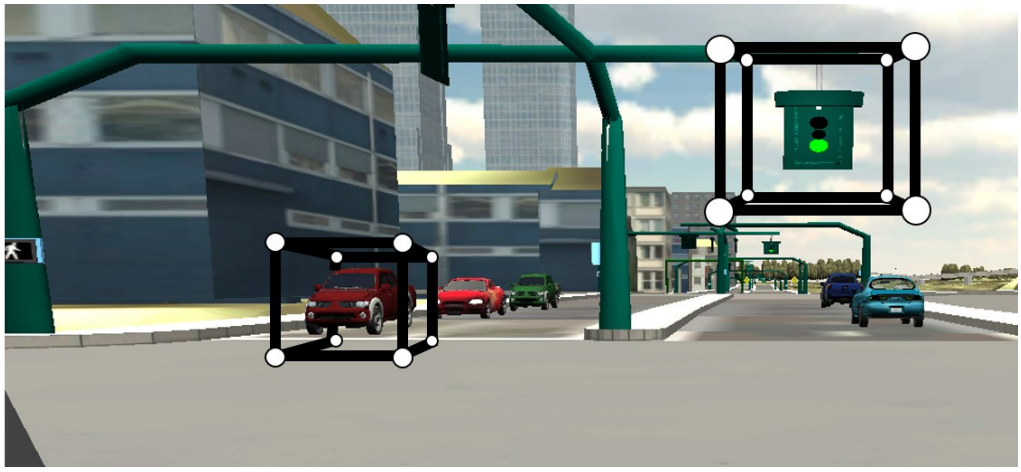

Figure 7. Bounding boxes around two ROIs in a turning-related trial: an oncoming vehicle and a traffic light.

An algorithm was developed to perform the desired FD calculation (Algorithm 1). This algorithm utilized a hashtable data structure to store key-value pairs in which the keys were identifiers for specific ROIs, and the values were the FDs for the associated ROIs in seconds. The algorithm had two parts: an initialization part and an update part. Initialization was performed at the start (or restart) of each trial in order to clear the hashtable of old key-value pairs. The update performed the actual FD calculations. The basic intuition of the algorithm was to check for intersection of the user's gaze $\mathbf{g} \in \mathbb{R}^2$, with a circle of radius $r \in \mathbb{R}$, around each relevant ROI's center of mass $\mathbf{c} \in \mathbb{R}^3$. If the gaze position fell within the circle, then the driver was determined to be looking at the ROI.

Formally, a ROI was defined as a tuple $\rho = (\mathbf{c},\ E)$, where $\mathbf{c} \in \mathbb{R}^3$ was the ROI's center of mass and $E = \{\mathbf{e_1}, \mathbf{e_2}, \dots, \mathbf{e_8}\}, \forall_{\mathrm{i}}\ \mathbf{e_i} \in \mathbb{R}^3$ was the set of extents of a bounding box around the ROI (see Fig. 7). A transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ was required to transform points in the virtual environment to coordinates on the screen; Unity's API provided such a transformation. If the algorithm determined that a gaze intersection for a particular ROI did in fact occur, then the FD time for that ROI was incremented by the elapsed time in seconds between the current and previous video frame. This value was also available via the Unity API and is denoted in Algorithm 1 as Δ_t. In line 5 of Algorithm 1, the furthest extent $\mathbf{e}^*$ from the ROI's center of mass $\mathbf{c_i}$ (in screen space) is obtained. Then in line 6, the radius r of a screen space circle is computed as the largest of either

the screen space distance between $\mathbf{c_i}$ and $\mathbf{e}^*$ multiplied by scalar α, or the constant k, where k is the minimum allowable radius length. The value of k was equal to 1 cm in screen space and was selected based on a validation of the eye tracking system (see Chapter 4 for a justification of the values of α and k). Line 7 of the algorithm determines whether an intersection has occurred and if so, then line 8 increments the FD time for the relevant ROI.

ALGORITHM 1. Fixation Duration Calculation for Regions of Interest

Input: A hashtable $\boldsymbol{H}$ of key-value pairs where a key is a ROI name and the value is the fixation duration in seconds, the driver's gaze position $\mathbf{g}$, the set R or ROIs relevant to the current trial, and the flag $initFlag$ indicating when to (re)initialize $\boldsymbol{H}$.
Output: The updated hashtable $\boldsymbol{H}$.
INITIALIZATION OF HASHTABLE
1: **if** $initFlag$ then
2: **for** i = 1 to n **do**
3: $\boldsymbol{H}[i] \leftarrow 0$
UPDATE HASHTABLE WITH NEW FIXATION DURATIONS
4: **for** i = 1 to n **do** {Iterate over each of the ROIs}
5: $\mathbf{e}^* = \underset{\mathbf{e} \in E_i}{\operatorname{argmax}} \|T[\mathbf{c_i}] - T[\mathbf{e}]\|$ {Find the furthest extent from the center of mass}
6: $r = \max(\alpha * \|T[\mathbf{c_i}] - T[\mathbf{e}^*]\|, k)$ {Compute the screen space ROI radius}
7: **if** $\|\mathbf{g} - T[\mathbf{c_i}]\| \leq r$ **then**
8: $\boldsymbol{H}[i] \leftarrow \boldsymbol{H}[i] + \Delta_t$
9: return $\boldsymbol{H}$

CHAPTER 4

EXPERIMENTAL EVALUATIONS

Results from a series of validation tests as well as two separate user studies are presented. The purpose of the validation tests was to show that both operational modalities were functional and robust. The first presented user study was conducted in order to demonstrate the acceptability and usability of the system. The second user study was conducted to verify the usability and potential of the intelligent gaze-contingent design for future ASD intervention.

1 System Validation Tests

Key elements of the system were quantitatively analyzed to ensure that they were behaving according to specification. Specifically, the network communication performance between the VDM and GDM modules, and the measurement uncertainty of ROI detection using the proposed gaze monitoring method were analyzed. The developed system is far too complex for formal model checking in its entirety; instead, a few important invariant properties of the system are discussed and it is shown that these properties held for every observed execution of the system. These properties include (1) feedback is always presented to the driver after a trial failure has occurred, and (2) after every trial failure, another trial failure will not occur until at least five seconds after the next trial begins.

Network communication throughput between the VDM and GDM is defined as the number of times per second that the VDM received a new driver gaze coordinate from the GDM. The communication speed should be sufficiently fast to reliably represent drivers' gaze within the driving environment. The sample analyzed consisted of 32 assignments (256 trials), and the mean communication throughput was found to be approximately 11.6 Hz, or about every 86 ms. Note that the GDM software logged drivers' gaze data at 120 Hz irrespective of the network throughput. The observed frequency was more than adequate for real-time gaze monitoring because it has been

shown that fixation durations less than about 300 ms are not long enough to indicate actual fixation in a dynamic scene, but rather saccadic movement [43].

The measurement uncertainty of the eye tracker in detecting users' gaze was also validated. Seven volunteers were enlisted to participate in a brief data collection task. The task consisted of a small white circle appearing at nine known locations in a random order for two seconds. The same type of monitor was used for this task as in the driving task (i.e., a 24" monitor with 1920 × 1080 resolution) and subjects were appropriately seated 70 cm from the monitor [35]. A mean error of 0.88° (1.08 cm) was found with the error appearing to be less severe at the bottom of the monitor (0.36°, 0.44 cm) than at the top (1.57°, 1.92 cm). Because this error appeared to be linear, the computed radii for ROIs were scaled based on their vertical position in screen space (α in line 6 of Algorithm 1) so that ROIs near the top of the screen had slightly larger radii than those at the bottom of the screen. Additionally, the mean error provided the basis for selecting the minimum ROI radius k = 1 cm in Algorithm 1.

For the proposed intervention system to be effective, feedback must be presented to the driver after every single driving error. Over the course of the 120 driving sessions, 912 instances of driving errors were registered by the system. The event logs produced by the *DataManager* component of the VDM (see Fig. 3) recorded the time of trial failures, followed by the time of feedback presentation as well as the time that drivers acknowledged the feedback messages. In every instance of a trial failure, the feedback presentation *and* acknowledgement events were present in the log. Descriptive statistics were computed for the duration of time that the feedback was present on screen: the mean duration, in seconds, was 7.44 s, standard deviation 8.25 s, and median 4.55 s. Note that feedback was presented as both text and audio. Therefore, for those subjects who may have quickly acknowledged the feedback text in order to return to gameplay sooner, they were still presented with the audio version of feedback.

Since it is possible that multiple driving errors could occur at the same time (e.g., a driver makes a wrong turn *and* drives onto the sidewalk), it is necessary that multiple failure events are not reported at or near the same time. A timed automaton model was designed specifically to guarantee that no trial failure occurred until five seconds after the next trial started. Of the exactly 600 instances of time intervals between consecutive trial failures measured during the driving sessions,

the shortest time duration was 5.08 seconds (M = 55.39 s, SD = 48.93 s), which satisfies the desired invariant property.

2 Pilot Study 1: Driver Performance Assessment

2.1 Objective and Hypotheses

This study was a preliminary investigation into the development and application of a VR driving simulation system capable of capturing gaze patterns to better understand driving performance. It was hypothesized that group differences would arise with respect to attention (measurable via eye gaze) and ability to complete driving-related tasks, including turns, stops, and obeying speed limits. Differences in how people with ASD attend to information could have important implications for safety and learning within a driving environment.

2.2 Procedures

Participants completed a single session that lasted approximately 90 minutes. At the session start, participants were shown a short video tutorial that explained the game controls and objectives. Following the tutorial, the eye tracking device was calibrated to the participants' eyes using a nine point calibration procedure. After calibration, participants began a three minute, free form practice driving period. During practice, there were no pedestrians or vehicles in the environment apart from the participants' vehicle allowing the participant to simply familiarize themselves with the controls and the virtual environment before starting the game.

When the practice session was completed, participants began the first of six assignments—two assignments from each of the three levels of difficulty. Assignments increased in difficulty along the course of the session (i.e., 2 easy, 2 medium, 2 hard). Participants were free to choose the order of the two assignments choices for each level. Participants could attempt an assignment only once, but progression through the levels did not require successful completion of the previous level's assignments. When a trial was failed, the system generated both text and audio feedback given the context of the failure. For example, if a participant was driving too quickly through an active school

zone, then the system advised, “Did you notice you were in a school zone? Always watch for speed limit signs when entering a school zone.” Participants acknowledged the messages and resumed game play by pressing twice on the accelerator pedal. As trials were successfully completed, a congratulatory audio sounded and $5 were added to the cumulative score visible in the upper left corner of the screen (see Fig. 8). When trials were failed, no points were given.

Figure 8. The driving perspective (Pilot Study 1).

At the end of every assignment (six per driving session), regardless of performance, participants completed a post-assignment self-report that was integrated into the game. The survey prompted participants to rate their affective states (i.e., engagement, enjoyment, frustration, and boredom) on a 5-Likert scale. Additional questions in the survey pertained to perceived quality of the game (e.g., graphics quality, instructional clarity, etc.). Upon completion of the six assignments and corresponding surveys, the session was complete. Participants were compensated for their time and all standards of ethics were followed according to Vanderbilt University’s Institutional Review Board guidelines.

2.3 Participants

Participants included 14 age- and gender-matched adolescents (7 with ASD, and 7 TD) of approximate driving age (16 years) in the state where the research was conducted. Four participants in each group had either a driver's license or a learner's permit.

Participants with ASD were recruited through an existing university clinical research registry. The registry includes individuals who received a clinical diagnosis of ASD from a licensed clinical psychologist and scored at or above the clinical cutoff on the Autism Diagnostic Observation Schedule (ADOS) [44] or Autism Diagnostic Observation Schedule, Second Edition (ADOS-2) [45]. Estimates of cognitive functioning (ASD group IQ $M = 114.3$) for those in the ASD group were available from the registry (tested abilities from either the Differential Ability Scales [46] or the Wechsler Intelligence Scale for Children—Fourth Edition [47]).

Participants in the TD group were recruited through an electronic recruitment registry accessible to community families. The clinical battery for the TD group included the Wechsler Abbreviated Scale of Intelligence, Second Edition [48] to quantify cognitive functioning (TD group IQ $M = 104.9$). To index initial autism symptoms and screen for autism risk in the TD group, parents completed the Social Responsiveness Scale, Second Edition (SRS-2) [49] and the Social Communication Questionnaire (SCQ)—Lifetime Version [50]. No participants in the TD group scored in the at-risk or clinical range on either instrument (see Table 4 for full participant data).

Table 4. Participant Characteristics (Pilot Study 1)

	Group M (SD)	
	ASD (n=7)	TD (n=7)
Gender (% male)	86%	86%
Chronological age	16.3 (0.98)	16.01 (1.14)
IQ	114.3 (10.42)	104.9 (19.02)
SRS-2 total raw score	95.3 (22.22)	9.17 (5.34)
SCQ total score	13.9 (7.86)	0.67 (0.82)
ADOS total raw score	12.1 (1.95)	--
ADOS CSS	7.6 (0.98)	--

Note: SCQ = Social Communication Questionnaire; SRS-2 = Social Responsiveness Scale, Second Edition;
ADOS CSS = Autism Diagnostic Observation Schedule Calibrated Severity Score; IQ = composite score: Differential Ability Scales (General Conceptual Ability) or Wechsler Intelligence Scale for Children (Full Scale IQ).

2.4 Data Analysis

Small sample sizes like those in this study often result in non-normal distributions of variables. In light of this, conservative non-parametric inferential statistics were computed to determine group differences and effect size. Mann-Whitney *U* tests were used to detect group differences and Vargha-Delaney *A* common language effect sizes [51] were computed. Results for Mann-Whitney *U* tests are reported as group medians rather than means and both *p* and *U* statistics are given. Vargha-Delaney *A* effect size is interpreted in the following manner: $A \geq 0.71$ indicates a large effect, $A \geq 0.64$ indicates a medium effect, and $A \geq 0.56$ indicates a small effect. Some correlation measures were also computed using the Pearson correlation coefficient *R*, for which *p* values are also given.

2.5 Results

Prior to analyses regarding performance, the study investigated the relationship between participant age, participant cognitive skills, and performance outcome. No significant relationship between age and number of failures emerged for the ASD group, $R(5) = -0.04$, $p = 0.93$. For the

TD group, younger participants failed significantly more than older participants, $R(5) = -0.79$, $p = 0.03$. Cognitive functioning was not significantly related to number of failures in either group.

Participants in the ASD group failed nominally more trials ($p = 0.06$) than their TD counterparts. Although this result did not reach statistical significance (likely due to low statistical power), there was a medium effect ($A = 0.61$). ASD participants experienced a combined total of 85 total trial failures resulting in 11 failed assignments, whereas the TD group experienced 55 total trial failures with only 1 assignment failure. A closer examination of the types of failures that occurred indicated that ASD participants failed at a significantly higher rate on trials involving turning the vehicle with large effect ($p < 0.01$, $A = 0.97$) than TD participants (see Table 5). Of the trial failures experienced by the ASD group, nearly half (48.2%) were related to turning compared to 34.5% in the TD group. No significant differences emerged in failure rates for the other three trial categories (i.e., merging, laws, and speed).

Table 5. Trial Failures during Sessions, by Trial Type and Group (Pilot Study 1)

	ASD		TD				
Trial Type	Median	Total	Median	Total	*U*	*p*	*A*
Laws	2	17	2	13	52	0.9907	0.5102
Turns	6	41	3	19	29.5	**0.0023**	**0.9694**
Merging	0	7	0	4	50.5	0.8776	0.5408
Speed	3	20	3	19	50.5	0.8228	0.5408
All Trials	11	85	8	55	37.5	0.0583	0.6122

The study also examined total FD for various categories of ROIs. FD in this context was defined as the cumulative amount of time that participants spent looking at particular ROIs during a trial. Note that FD times were corrected for assignment duration as the ratio of time spent looking at ROIs to the total duration of the assignment. This choice was made because failing trials resulted in more time spent attempting a trial. Compared to TD controls ($Mdn = 0.05$), FD rates were nominally higher in the ASD group ($Mdn = 0.09$) for 'dynamic' ROIs in the environment, $p = 0.21$, $A = 0.71$. FD rates for 'social' ROIs (pedestrians and cyclists) were also nominally higher in the ASD group ($Mdn = 0.04$) than in the TD group ($Mdn = 0.02$), $p = 0.21$, $A = 0.71$. It may seem odd

that these statistics are identical for both categories of ROIs, but using non-parametric rank-based tests with such a small sample size tends to result in similar statistics.

Table 6. FD Differences between Regions of Interest (ROIs) in the Environment

	Median FD		Statistics		
ROI Category	ASD	TD	*U*	*p*	*A*
Social	0.0373	0.0181	63	0.2086	**0.7143**
Dynamic	0.0925	0.0516	63	0.2086	**0.7143**
Static	0.1065	0.106	48	0.62	0.5918
All ROIs	0.1997	0.169	56	0.7104	0.5714

Note: ROI = Region of interest

The performance results indicated that a proportionally higher number of trial failures in the ASD group were related to turning; therefore, the study investigated the FD rates for relevant ROIs in turning trials—most of which involved a traffic light. Analysis yielded a strong correlation [$R(40) = 0.439$, $p = 0.004$] between total number of trial failures in an assignment and FD rates for traffic light ROIs during turn-related trials in the ASD group. Conversely, no significant correlation was present [$R(40 = -0.12$, $p = 0.449$] in the TD group. This suggests that longer fixation durations on traffic lights while turning may be correlated with higher rates of driving errors in the ASD group.

Significant between-group differences emerged regarding overall gaze pattern as well. First, the average and median gaze position was higher in the ASD group ($p < 0.001$) by approximately 0.569 cm (see Fig. 9). The average horizontal gaze position in the ASD group tended towards the right portion of the screen ($p < 0.001$) by 0.123 cm. When considering only the gaze from turn-related trials, which were most problematic for the ASD group, the same gaze pattern emerged. Interestingly, during both left *and* right turns, the ASD group gaze tended significantly towards the right side of the screen by 0.417 cm ($p < 0.001$) and 0.315 cm ($p < 0.001$), respectively.

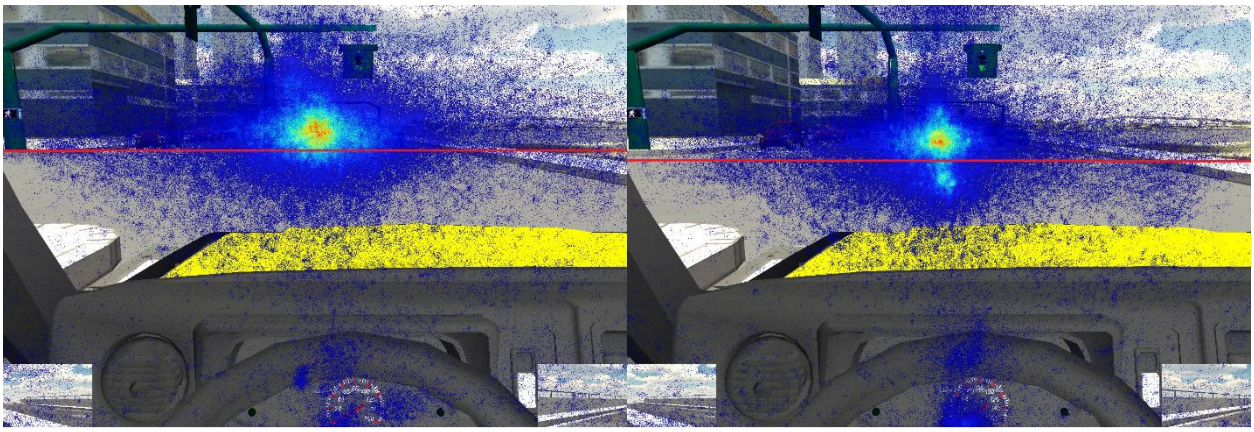

Figure 9. Gaze position heat maps for the ASD group (left) and TD group (right). Horizontal red lines indicate the average vertical gaze position.

Self-report data indicated that ASD participants experienced significantly lower levels of enjoyment ($p < 0.05$), and nominally higher levels of frustration ($A = 0.81$, $p = 0.06$) compared to those in the TD group. Overall, participants in the ASD group reported more negative views of their experience interacting with the game. Specifically, they gave lower ratings pertaining to ease of operating the vehicle ($p = 0.07$, $A = 0.80$), clarity of the instructions ($p = 0.02$, $A = 0.87$), relevance of the objectives ($p = 0.01$, $A = 0.91$), and quality of the virtual environment ($p = 0.10$, $A = 0.76$) and higher ratings pertaining to the difficulty of the assignments ($p = 0.02$, $A = 0.88$). Refer to Table 7 for a full review of the self-report data. The study also looked for correlations between metrics from the self-report and performance outcome, but no statistically significant results were present. Interestingly, there was no correlation in either group between the number of failures and perceived difficulty [ASD group: $R(40) = 0.032$, $p = 0.84$; TD group: $R(40) = -0.27$, $p = 0.084$], nor was there a correlation between the actual difficulty and the reported perceived difficulty [ASD group: $R(40) = 0.187$, $p = 0.235$; TD group: $R(40) = 0.257$, $p = 0.1$].

Table 7. Self-report Results (Pilot Study 1)

	Median Self-score		Statistics		
Survey Item	ASD	TD	*U*	*p*	*A*
1	3.6667	4	67	0.0664	**0.7959**
2	2	4	66	0.0956	**0.7755**
3	2.8333	4	72.5	**0.0099**	**0.9082**
4	2.8333	4.1667	70.5	**0.0192**	**0.8673**
5	3.1667	3.6667	71	**0.0163**	**0.8776**
6	3	4	69.5	**0.0291**	**0.8469**
7	3.5	4.1667	59.5	0.3998	0.6429
8	3.1667	4.1667	67.5	0.0612	**0.8061**
9	4.5	4.1667	53	0.9767	0.5102

Note: See Appendix C for the questions and response options in the self-report.

2.6 Discussion

In this preliminary investigation, a novel VR driving simulation system, including a custom gaze data acquisition module, was used to examine performance and processing differences between adolescents with and without ASD on tasks of driving. Our hypotheses were partially supported, such that group differences emerged when examining some key aspects of performance as well as processing of information within the VR driving environment.

Compared to TD controls, participants with ASD failed more driving trials. Notably, the majority of these failures occurred during driving tasks that involved turning the vehicle, typically with a traffic light ROI present. Given this failure pattern, the study investigated the relationship between failures and eye gaze FD rates during tasks that required turning the vehicle. These analyses supported a relationship between failure and eye gaze fixation duration on the traffic light ROI, which suggests that eye gaze FD, and perhaps an intense focus, on traffic lights may result in driving task failures, and thus, unsafe driving. Moreover, the ASD group demonstrated longer eye gaze fixation duration regarding ‘dynamic’ and ‘social’ ROIs. This further suggests that driving task failures may be associated with eye gaze FD and attentional differences, specifically with moving elements in the environment (e.g., other vehicles, pedestrians).

Differences in gaze patterns also emerged. These findings replicated recent work by Reimer et al. [27] which also found higher rates of driving errors and differences in gaze patterns for adolescents with ASD. Relative to the TD group, the gaze position of the ASD group tended to gravitate towards the top of the screen in the vertical direction and the rights side of the screen in the horizontal direction. This pattern also emerged when analyses isolated gaze during turn-related trials. These results seemed to mirror findings from Reimer et al. [27] in which study participants with ASD tended to look higher on the screen in the vertical dimension and towards the right side of the screen in the horizontal dimension.

Our finding that individuals in the ASD group spent more time than TD counterparts looking at traffic light ROIs during turns fits with our finding that they had a higher vertical gaze position. Since this finding also correlates with a higher number of turn-related trial failures, it may be that ASD participants are distracted by the traffic light itself when performing turns and—to their detriment—are less focused on other key elements of the environment. The current study also revealed that participants with ASD spent more time than TD participants looking at 'social' ROIs with moving features. Again, this may reflect underlying challenges of individuals with ASD regarding their ability to effectively scan the environment, disengage from irrelevant stimuli, and attend to relevant information.

Self-reported affective data was recorded to gauge user engagement, enjoyment, frustration, and boredom regarding our novel VR system. To the best of the author's knowledge, this is the first time that an interactive VR system has been integrated with an affective self-evaluation electronic questionnaire for individuals with ASD. Participants in the ASD group reported generally lower engagement and enjoyment and more frustration than the TD group. Though the perception of the VR system by individuals with ASD did not correlate with performance, elements of the system were less engaging, enjoyable, and more frustrating for the ASD group.

Cumulatively, these findings support the development of a VR driving simulation system not only to evaluate the adaptive skill of driving, but also to potentially train individuals to attend to important stimuli during driving tasks. Real-world driving often requires fast and accurate interpretations of, and response to, others' behavior within environments that allow little room for error. VR driving simulation systems can dynamically display important aspects of functional tasks and evaluate physiological and temporal components relevant to driving demands. Physiological

and temporal data not only provide insight into one's information-processing abilities regarding driving tasks and behavior, but also provide a platform for intervention to shift individuals' attention within the environment. A VR driving simulation system could therefore become a valuable intervention tool be enacting changes based on how participants recognize and process environmental cues in addition to their task performance. Addressing underlying performance vulnerabilities on a processing level may result in changes that more powerfully generalize than current approaches for teaching basic driving skills.

Finally, the study results revealed a correlation between age and trial failures in individuals without a diagnosis of ASD. Consistent with driving literature regarding the general public, this finding suggests that age and experience facilitates safer driving. A VR driving simulation system provides a safe environment in which to practice and obtain experience with the many complex demands of driving. Virtual training environments that respond in real time to shift and train a person's attention may facilitate skill development leading to safe, independent driving in individuals with ASD. This could allow for autonomous personal transportation and enhanced access to a variety of outlets for independent adult living (e.g., vocational settings, continued education/training, social experiences, etc.).

3 Pilot Study 2: Gaze-contingent System Assessment

3.1 Objective and Hypotheses

The major hypothesis of this study was that realization of an autonomous system capable of providing real-time gaze contingent feedback would contribute to both enhanced performance within the driving environment (e.g., fewer driving errors) as well as shape how individuals with ASD were scanning the relevant objects in the environment (e.g., alterations in gaze).

3.2 Procedures

Participants came to our lab facilities to complete six sessions lasting approximately 75 minutes in length. In most cases, sessions were performed on separate days. The first and last of these

sessions consisted of a pre- and post-test, respectively. The pre- and post-tests were identical in terms of the task content and served as the basis of comparison for performance outcomes. The pre- and post-test session tasks consisted of three assignments selected from difficulty levels two and five in order to observe driver performance under a variety of difficulty settings. Sessions 2-5 were each unique in terms of both task content and difficulty. Task difficulty was the easiest during session two, the hardest during session five, and medium during session three and four (see Table 8). Upon arriving for each session, physiological sensors were applied to the participant's body. Next, the EPOC headset was fitted to the participant's head. Following this, the Tobii eye tracking device was calibrated following the calibration procedures recommended by the manufacturer [35]. Fig. 10 gives a schematic of the experimental facilities. Participants were seated in the playseat such that their eyes were approximately 70 cm from the eye tracking device. Each subjects' eyes were individually calibrated to a 24" monitor (1920 × 1080 resolution) on which the VR driving environment was displayed. A researcher administering the evaluation sat to the right of the subject at a PC running the simulation application. An observer sat behind a one-way mirror in a separate room at a PC running the OAM application.

Table 8. Difficulty Level and Assignment Number per Session Given as a Pair (Level, Assignment)

Assignment	**1 (Pre-test)**	**2**	**3**	**4**	**5**	**6 (Post-test)**
1	2, 2	1, 1	3, 1	4, 1	6, 1	2, 2
2	5, 1	1, 2	3, 2	4, 2	6, 2	5, 1
3	5, 2	1, 3	3, 3	4, 3	6, 3	5, 2

At the first session, a brief tutorial was presented to the participants. This tutorial outlined the control of the G27 device, the basics of road safety, meanings of common signs, and how to complete driving tasks. Participants were informed only of the rules of the mode of the group in which they were placed (i.e., PB or GC groups/modes). Following the tutorial, a silent baseline data collection period of three minutes was performed to obtain a reference model of affect for each subject. When the baseline was complete, subjects began a three minute practice driving

session to gain some experience with the controls before beginning the actual driving tasks (frist session only). Finally, subjects attempted the core driving task which consisted of three assignments. As mentioned before, assignments could be attempted only once and when all three assignments had been attempted, the session was complete.

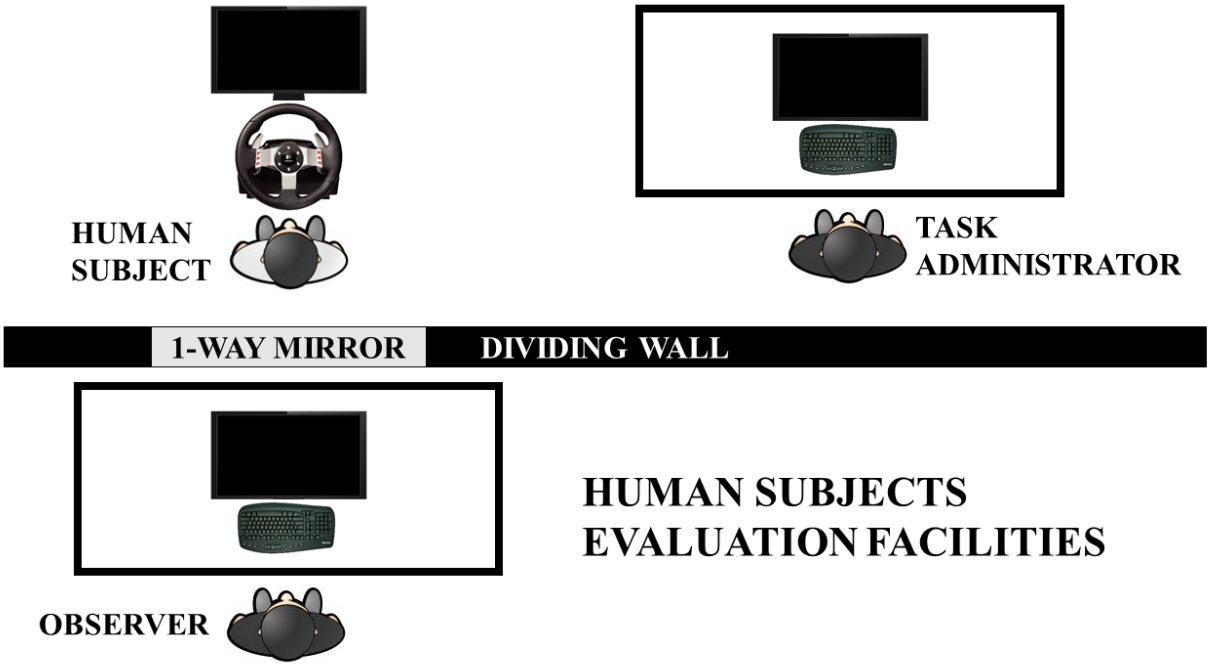

Figure 10. Experimental evaluation room schematic.

3.3 Participants

Twenty adolescents aged 13-18 years (M = 15.29, SD = 1.65) and diagnosed with ASD by our team of collaborating psychologists via administration of gold-standard research diagnostic instruments (e.g., ADOS-2 [45]) were recruited for a small study to evaluate the reliability of VADIA's PB and GC modes. Because ASD is twice as common in males as females, and although recruitment was open to both sexes, the majority of our participants (19 out of 20) were male. Randomized group placement assigned 10 participants to the PB group and the remaining 10 participants to the GC group. Of the 20 participants, three had a learner's permit and only one had

a driver's license. Subject's parents completed the SRS-2 [49] to quantify the severity of their child's ASD symptoms. The randomized group placement resulted in two individuals with learner's permits being assigned to the PB group and one individual with a driver's license and another with a learner's permit being assigned to the GC group. Table 9 gives a detailed comparison of group characteristics. Informed consent and assent was appropriately obtained for each participant and participants were compensated at each visit for their time. This study was approved by the Institutional Review Board at Vanderbilt University.

Table 9. Participant Characteristics (n = 20, Pilot Study 2)

Group M (SD)			
	PB (n = 10)	**GC (n = 8)**	**Failed to complete (n = 2)**
Gender (% male)	100%	87.50%	100%
Chronological age (years)	15.1 (1.58)	15.76 (1.86)	14.37 (1.05)
SRS-2 total raw score	96.1 (31.66)	96.5 (21.05)	112 (35.36)
SRS-2 Tscore	74.3 (10.03)	75.63 (9.53)	80.5 (13.44)
Permit-holders (%)	20%	12.50%	0%
License-holders (%)	0%	12.50%	0%

3.4 Data Analysis

Pre- and post-test outcomes were evaluated using two-sided Wilcoxon signed rank tests so medians are given along with Z and p statistics. This choice was due to the non-normally distributed variables and the nature of the study (i.e., sampling the same population at two different points in time). In addition, Cohen's d is reported to give an estimate of effect size. The significance cutoffs for Cohen's d are as follows: $d \geq 0.8$ represents a large effect, $d \geq 0.5$ a medium effect, and $d \geq 0.2$ a small effect.

3.5 Results

All of the subjects visited the lab facilities for all six sessions and attempted the assignments in each task. An inclusion criterion was formally defined such that subjects must complete at least one assignment over the course of all six visits for consideration in data analysis. This criterion resulted in the elimination of two subjects from the data set. Removing these subjects changed the group balance to 10 participants in the PB group and eight in the GC group.

Two metrics of performance from the pre- and post-tests of both groups were analyzed to determine whether the intervention programs were effective. Median trial duration was computed for each trial completed by each subject and gives an indication of how difficult a task is to complete. Shorter median trial durations in the post-test are preferable because they indicate a higher proficiency in task-completion. The median number of trial failures per assignment were also analyzed; this metric indicates the level of improvement within the task and should be smaller in the post-test than the pre-test. The results of both analyses are presented in Tables 10 and 11.

Table 10. Median Trial Durations during Pre- and Post-tests for PB and GC Groups

	Trial Duration (in seconds)		Statistics		
Group	Pre	Post	Z	p	$\|d\|$
PB	43.8144	23.6224	3.2622	0.0011	0.8218
GC	36.0506	28.8219	2.4213	0.0155	0.4778

A Wilcoxon signed ranks test indicated that, for subjects in the PB group, mean trial duration was significantly longer during the pre-test (Mdn = 43.81 s) than during the post-test (Mdn = 23.62 s) with large effect, $Z = 3.26$, $p < 0.01$, $d = 0.82$. Additionally, the PB group showed a significant decrease in the number of trial failures per assignment from the pre-test (Mdn = 7) to the post-test (Mdn = 3) with very large effect, $Z = 2.37$, $p < 0.05$, $d = 1.43$.

Table 11. Median Trial Failures during Pre- and Post-tests for PB and GC Groups

	Trial Failures		Statistics		
Group	Pre	Post	Z	p	$\|d\|$
PB	7	3	2.3749	0.0176	1.4285
GC	11	6	2.3749	0.0176	1.1227

GC group subjects demonstrated a significant decrease in mean trial duration from pre-test (Mdn = 36.05 s) to post-test (Mdn = 28.82 s), $Z = 2.42$, $p < 0.05$, $d = 0.48$. GC group subjects also showed a significant decrease in total trial failures per assignment from pre-test (Mdn = 11) to post-test (Mdn = 6) also with large effect, $Z = 2.37$, $p < 0.05$, $d = 1.12$.

3.6 Discussion

While the median number of trial failures per assignment in the post-test was higher for individuals in the GC group than the PB group, participants in the GC group were able to experience failures related to both performance and gaze pattern while the subjects in the PB group only experienced performance-related failures. Moreover, the effect size was large for both groups in regards to the decrease in trial failures. This result suggests that both modes of intervention are effective at improving performance of users. However, the fact that the GC group participants performance improved significantly while driving in the more challenging version of the system may suggest that the gaze-contingent intervention system is more suitable as an intervention program than the PB system.

Both the performance- and gaze-contingent groups demonstrated statistically significant improvements in performance over time. While the driving simulation tasks were similar across groups, the metrics for success were different for these groups. Specifically, the gaze-contingent group was held to a higher bar for success in that they had to perform the task correctly and look successfully at salient aspects of the environment. The performance group could achieve success just based on performance. With this caveat in place, the improvement in the GC group may be indicative of both success in scanning the driving environment as well as success in performing the task. These findings are quite promising in supporting the hypothesis that VR systems informed by information processing metrics may be valid tools for alerting fundamental processing

challenges associated with ASD. If such systems are capable of realizing such change, then these systems may be much more powerful in demonstrating change that generalizes to novel environments (e.g., learning and information processing strategies may be successful in other environments).

CHAPTER 5

CONTRIBUTIONS AND CONCLUSIONS

1 Contributions

Driving simulation systems have been in use since as early as the 1970's [1], and a wide range of commercial systems exist that can be used to assess driver performance in controlled environments. However, no system has been made available which provides closed-loop feedback using interchangeable modalities such as physiology, eye gaze, or EEG. Moreover, for populations with cognitive, behavioral, or neurodevelopmental disabilities, off-the-shelf simulators may not provide the challenges necessary to address their specific needs. For example, individuals with ASD demonstrate scanning patterns while driving that may be unsafe [11, 24, 27]. Commercial driving simulators were thus deemed insufficient for driving intervention in these individuals.

This thesis makes three primary contributions while addressing these aforementioned issues with current driving simulation technology. First, a novel, adaptive, VR driving simulation platform (VADIA) was developed and targeted at driving intervention in individuals with ASD. This system combines real-time gaze monitoring and gaze-contingent feedback in order to go beyond merely addressing performance issues, but processing issues as well. Second, the developed system has the capacity for feedback from a variety of modalities, one of which (i.e., gaze) is demonstrated through a pilot study. Third, two separate pilot studies involving a combined 27 individuals with ASD were conducted. The results from both of these studies are quite promising and show that the developed system is robust and may be effective as an intervention tool.

VADIA also makes a few secondary contributions of note. A model-centric approach to design was used to create VADIA following conventions developed in the field of Embedded Systems [34]. This approach has typically been exclusive to Embedded Systems, but has recently begun expanding into other domains because of its many useful properties [32, 33]. Further, a simple, streamlined method of implementing software from the models was developed and used in the

creation of VADIA (see Appendix B). Finally, VADIA has been used effectively in the work of other researchers for a variety of applications [38-41].

2 Limitations and Future Work

Though the results from the intervention pilot studies are promising, these studies were preliminary in nature and several key areas of future work are needed. Both studies' small sample sizes, although characteristic of initial user studies in general, weakens the statistical power of the results to a certain extent. Readers are thus cautioned against making broad inferences at a population level from these data. Larger randomized clinical trials are required to verify the translation of these results to the general ASD population. With regards to the second pilot study, although performance was tested across both feedback modalities (i.e., performance and gaze), a methodologically equivalent metric for calculating true performance and processing differences across the two conditions was not tested. Therefore, comparisons cannot be directly made between the two modalities. Despite these limitations, the results represent meaningful steps toward developing dynamic VR driving environments linked with gaze technology for possible ASD intervention.

One direction for VADIA in the future will be to refine the simulator and further develop the VR program to test the ability of the system to advance learning beyond the intervention environment itself. It is anticipated that providing a safe environment in which to practice driving skills, with ongoing monitoring of performance, engagement, and processing will lead to individuals with ASD having more confidence and successful navigation of driving tasks in the real world.

3 Conclusions

Cumulatively, the presented findings support the potential of developing technological tools such as VR driving simulators with embedded gaze-contingent feedback. Such integrated systems may be able to dynamically display important aspects of functional tasks, potentially guiding and altering gaze processing and attention. Thereby enhancing processing and performance within

these environments over time. Specifically, such a system could become a valuable intervention tool by enacting changes not just based on performance, but also on how participants recognize and process environmental cues. Further, addressing underlying performance vulnerabilities on a processing level may result in changes that generalize more powerfully than current approaches for teaching functional skills, as real-world driving often requires fast and accurate interpretation of, and response to, others' behavior within environments that can be unsafe and allow little room for error. Finally, it is important to note that this difference in processing of dynamic information is likely not circumscribed to driving, but may be related to many other challenges and vulnerabilities associated with ASD. As such, work addressing differences in processing (rather than just performance) may be important to designing intervention paradigms across other areas of skill vulnerability.

APPENDICES

APPENDIX A: Models of Computation

1 Finite State Machines

A finite state machine (FSM) is a mathematical description of the behavior of a system or component of a system. The formal definition of a FSM M can be expressed as a five-tuple where

$$M = (S, I, O, T, s_i).$$

Here, S is a finite set of states, I is a set of input values, O is a set of output values, T is a set of transition functions mapping states and inputs to states and outputs, and s_i is the initial state (i.e., the state at which the system is initialized). A state can be thought of as a particular set of conditions at a point in time within some specific scope. Coupled with the formal definition of a FSM is the notion of updating a FSM. At discrete moments in time, a FSM updates, meaning the state changes (or remains the same) as a function of the current state and inputs.

A FSM with a sufficiently small number of states can be represented graphically. This is useful for purposes of documentation and analysis as well as for explaining a system's behavior to others. The conventional method for drawing a FSM consists of circles (or sometimes rectangles) to indicate states and directed arcs between the states to represent transitions. Often, the *initial transition* is shown as an arc directed towards a state, but emerging from no other state in the FSM. States are typically enumerated or labelled with an appropriate name in order to distinguish between them. Generally, arcs are labelled with two pieces of information: *guard conditions* and *actions*. The guard conditions on an arc leaving a particular state describe the circumstances under which that state transitions via that arc to the next state. The actions associated with an arc are enacted when the guard conditions evaluate to *true* and the transition is taken. Some complexities arise when multiple transitions are possible at the same moment in time. This is referred to as *nondeterminism* and is beyond the scope of this appendix.

A common example used to introduce the concept of a FSM is that of a FSM which describes the behavior of a traffic light. For the purposes of this example, a very simple implementation of

a traffic light is examined that operates on a fixed cycle and does not consider pedestrian crossings or vehicle arrival triggers. Consider a traffic light at a four-way intersection that has three signals: red, yellow, and green. Suppose that the traffic light operates on a regular cycle in which the red light is enabled for 10 s, yellow for 2 s, and green for 8 s. Suppose further that the traffic lights for cross traffic operate on an 'opposite' schedule (i.e., when one traffic light is red, the cross traffic light is either yellow or green). Fig. 11 shows a graphical representation of this traffic light specification with states $S = \{Red, Green, Yellow\}$, inputs $I = \emptyset$, outputs $O = \{red, green, yellow\}$, an appropriate set of transition functions T, and initial state $s_i = Red$.

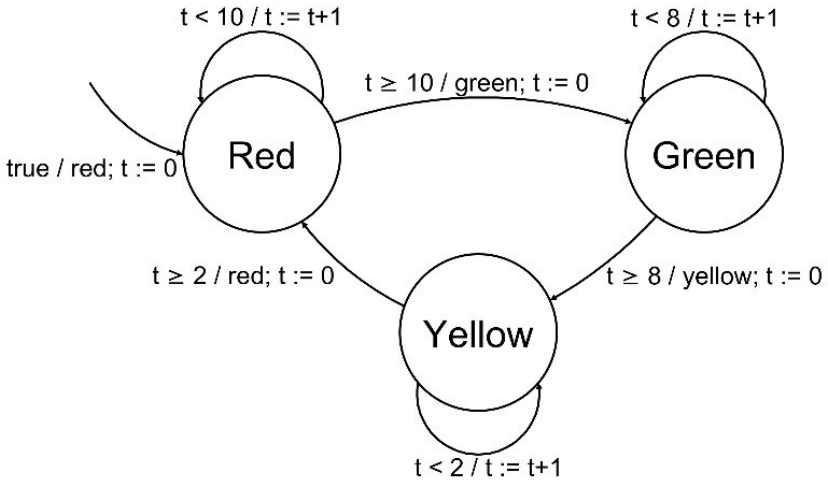

Figure 11. A FSM describing the behavior of a simple traffic light.

From Fig. 11 the reader can see that the initial state of the FSM is *Red*. This choice was made arbitrarily as there is no logically correct choice as to which state should be the initial state. Notice that the guard condition on the initial transition is *true*; this means that the initial transition is enabled regardless of the other conditions in the environment and that the state *Red* will be entered on the very first update of the FSM. A variable $t \in \mathbb{N}$ is used to keep a count of elapsed time in the environment. A value of 0 is assigned to t during the initial transition. Assuming that the FSM updates at a rate of 1 Hz, the state *Red* remains active for 10 updates (i.e., 10 s). For values of t smaller than 10, the machine updates and increments the value of t by 1 each time. Eventually, the value of t will be equal to 10 and the guard condition on the arc from state *Red* to state *Green*

becomes enabled. During this transition, the signal *green* is output and the value of t is reset to 0. The states *Green* and *Yellow* execute similarly to that of the state *Red*, but with different time thresholds for the guard conditions and different output signals on their outgoing arcs.

2 Hierarchical State Machines

It is possible to compose two or more FSMs to form more complex systems. Before describing the details of hierarchical finite state machines (HSM), a motivation is given for why FSM composition might be desirable. Consider the traffic light FSM introduced in the previous section (Fig. 11). Suppose that the behavior demonstrated in this example was desirable during the daytime when traffic is busiest, but at nighttime when there is less traffic, a simple flashing light behavior is preferred. Note that at flashing light intersections, drivers seeing a yellow blink are advised to proceed with caution while drivers seeing a red blink are required to make a full stop before proceeding. One option is to create a 'flat' FSM such as that shown in Fig. 12 which indicates all of the possible states that the traffic light can be in at a given time. This FSM now contains the set of states $S = \{Red, Green, Yellow, Bright, Dark\}$, a variable indicating the current time of day (*ToD*) with respect to a 24 hour clock, and two new discrete output signals *day* and *night*. The state *Bright* indicates that the flashing light (whether yellow or red in color) is lit, while the state *Dark* indicates that the flashing light is not lit. Again, for simplicity assume an update frequency of 1 Hz.

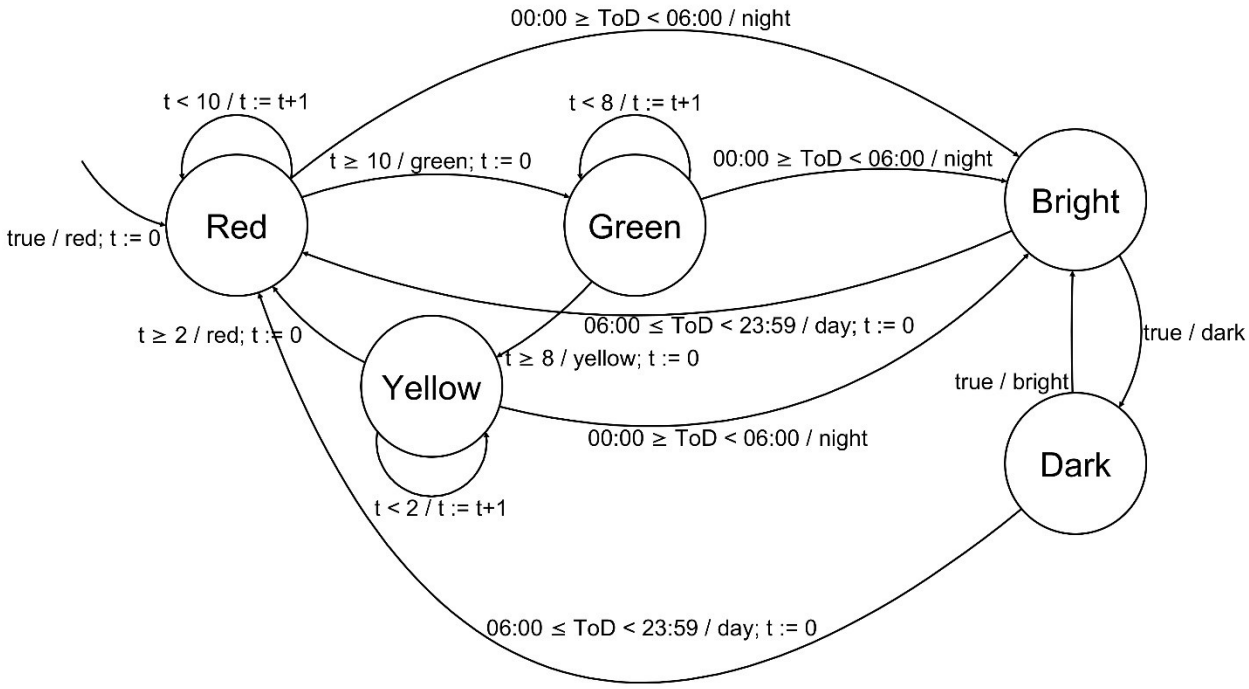

Figure 12. A flat FSM representation of the daytime/nighttime traffic light behavior.

The reader will notice that the FSM in Fig. 12 is somewhat complex and perhaps difficult to read. While this state machine certainly models the specification described, it is messy and would prove difficult to modify later if necessary. An alternative way to model the desired behavior is to model the flashing light behavior separately from the normal daytime behavior, and then compose the two FSMs into a single HSM. Fig. 13 gives a FSM graph for the flashing light behavior by itself and Fig. 14 gives the HSM that composes both the normal daytime and flashing light behaviors.

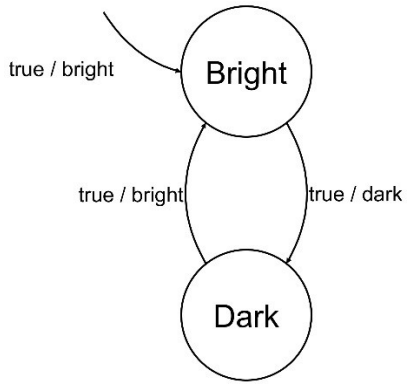

Figure 13. A FSM graph that describes the behavior of a flashing light traffic light.

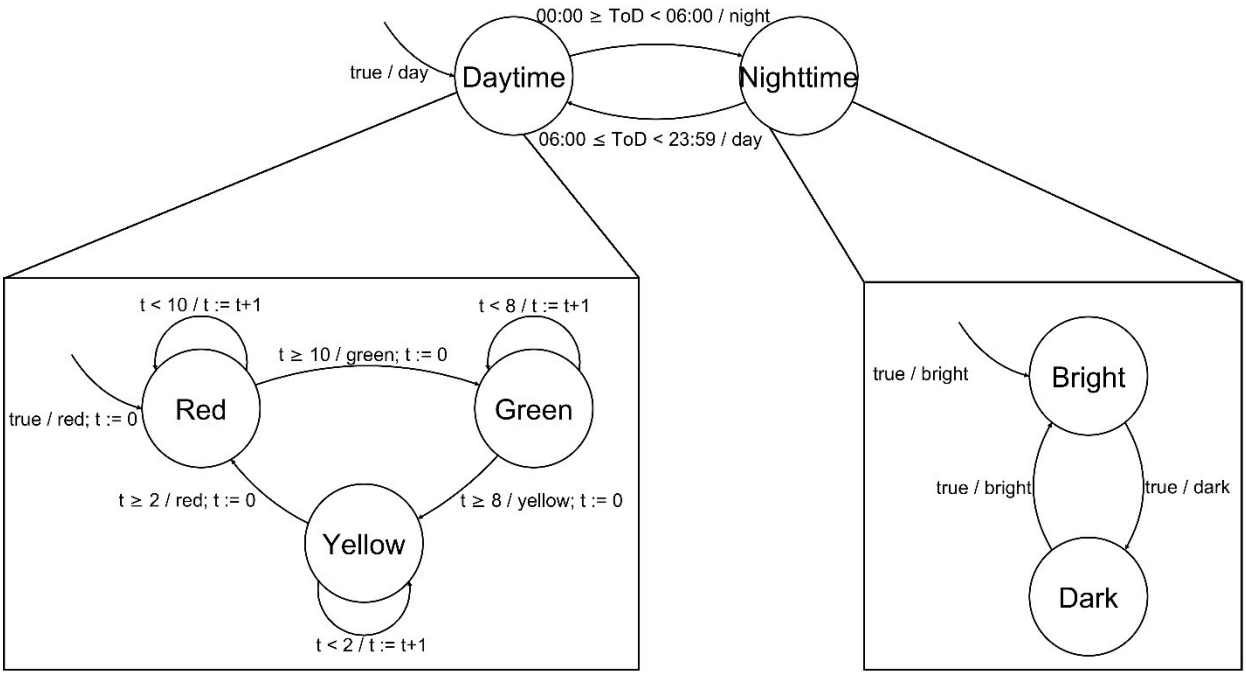

Figure 14. A HSM describing the composed daytime/nighttime traffic light behavior.

The HSM composition in Fig. 14 is an improvement over the composition given in Fig. 12 for at least two reasons: (1) there are fewer arcs between the various states which makes reading the graph easier, and (2) if, in the future, it was desirable to make slight modifications to the behavior of the flashing light aspect of the system, redrawing arcs to the daytime behavior-related states

would be unnecessary, which is a nice feature of compartmentalizing behaviors. However, HSMs also introduce some nuanced complexities which are now discussed.

Consider an example execution trace of the HSM in Fig. 14. On the top-level, the state *Daytime* is entered initially and at the same time, the lower-level FSM refinement of *Daytime* enters the state *Red*. Let us assume that, after some sufficient number of updates, the value of *t* is 10 and the current time *ToD* is exactly 00:00 (or midnight). What happens here? Technically, both the transition from *Red* to *Green* and the transition from *Daytime* to *Nighttime* are enabled, but how can it be decided which to take? It turns out that the outcome depends on the semantics chosen by the modeler at design time. A reasonable semantic approach, and that used by the author when designing the system described in this work, uses preemptive transitions. This means that, when multiple possible transitions are enabled at the same time, only the transition at the highest level of scope will be taken. So, in the scenario described here, the transition from *Daytime* to *Nighttime* will be taken—resulting in a transition to the low-level state *Bright*—and the transition from *Red* to *Green* will not be taken.

Continuing with this example, suppose that after some time has passed, the value of *ToD* is now 06:00 enabling the transition from *Nighttime* back to *Daytime*. The last time this low-level FSM was active, the FSM state was *Red*, but was about to transition to *Green*. The questions is now, when returning to the FSM refinement of *Daytime*, what state should be entered? Again, the answer depends on the semantics selected by the designer. Two reasonable approaches are to use history transitions or reset transitions. With history transitions, the HSM remembers the state that the system was in at the time that the scope changed and returns to that state when the scope is reestablished. Alternatively, a HSM using reset transitions remembers nothing and the initial state is always entered when scope returns.

3 Hybrid Automata

A hybrid automaton (HA) is a type of model used to describe systems that demonstrate both discrete and continuous behaviors. Like a FSM, a HA has a finite set of discrete states (also referred to as modes), but a HA also includes a set of real-valued (i.e., continuous) variables. These real-valued variables may be attributes of a system such as an object's position or orientation and are

governed by ordinary differential equations. The syntax for HA described here follows closely to that of Lee and Seshia [34].

One class of HA known as timed automata is quite popular and simple to explain. A timed automaton is an appropriate model choice for describing systems in which actions should be taken based on elapsed time or at specific time instances. Continuing with the traffic light example presented earlier, let us construct a model for a traffic light that changes color after specific time intervals. Specifically, the red light should be enabled for 10 s, the green light for 8 s, and the yellow light for 2 s. It may be assumed that the update rate for this model is infinitely fast (i.e., continuous). Fig. 15 gives the timed automaton for this specification. The function $s(t)$ indicates the time at instance *t* according to some clock. Upon entering the discrete mode *Red*, the value of the clock is initialized to the value 0 and the discrete signal *red* is output. Within the mode *Red*, the equation $\dot{s}(t) = 1$ tells us that the rate of the clock update is constant (i.e., linear). At the moment when the guard $s(t) \geq 10$ evaluates to true, the discrete signal *green* is output, the clock is reset, and the transition from *Red* to *Green* is taken. Similar behavior to that of the *Red* mode occur at both the *Green* and *Yellow* modes. Although this HA is extremely simple, it is quite useful for defining time-dependent behavior.

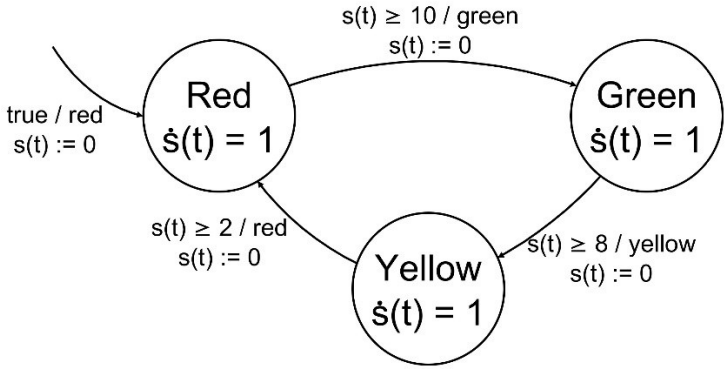

Figure 15. A timed automaton model of the basic traffic light specification.

APPENDIX B: Automata-based Programming

Automata-based programming refers to the conversion of an automata-style model to source code. The approach presented in this appendix is also presented in a paper co-authored by the present author [52]. The code in this appendix is in the C♯ programming language and the examples presented are given *in the context of* the Unity3D API. This appendix provides a step-by-step implementation of the HSM model derived in Appendix A. The described procedure is representative of the implementation of VADIA.

A base class called *Automaton* is used to maintain state variables and provide a state transition function for a particular model:

```
public class Automaton {
     private int m_currState;
     private int m_prevState;

     public void Transition( int NewState ) {
          m_prevState = m_currState;
          m_currState = NewState;
     }

     public int CurrState {
          get{ return m_currState; }
     }

     public int PrevState {
          get{ return m_prevState; }
     }
}
```

Formal MoC may be implemented as either an instance or a derived instance of this *Automaton* object. For example, the FSM in Fig. 11 may be defined thusly:

```
public class TriLightFSM : Automaton {

     /* State Constants */
     public const int RED = 0;
     public const int GREEN = 1;
     public const int YELLOW = 2;

     /**
      * Default constructor.
      */
     public TriLightFSM() : base()
     {
```

```
            //initial transition
            this.Transition( RED );
        }
}
```

Similarly, the FSM in Fig. 13 may be defined as:

```
public class FlashingLightFSM : Automaton {

        /* State Constants */
        public const int BRIGHT = 0;
        public const int DARK = 1;

        /**
         * Default constructor.
         */
        public FlashingLightFSM() : base()
        {
            //initial transition
            this.Transition( BRIGHT );
        }
}
```

Suppose it is desirable now to implement the HSM in Fig. 14. A simple FSM may be defined to represent the two-state behavior:

```
public class TrafficLightHSM : Automaton {

        /* State Constants */
        public const int DAYTIME = 0;
        public const int NIGHTTIME = 1;

        /**
         * Default constructor.
         */
        public TrafficLightHSM() : base()
        {
            //initial transition
            this.Transition( DAYTIME );
        }
}
```

Now, there are three *Automaton*-based object definitions, however, this is obviously not an implementation of the desired behavior. In order implement the intended behavior using these object definitions, automata-based programming is used. The code excerpt below implements the exact specification modeled by the HSM in Fig. 14. Since there are only two layers to this

hierarchy, a branching structure is used to separate the behaviors of the hierarchical components. However, when the number of layers is higher, it may be preferable to implement the *Automaton* objects in separate classes/files.

```
//====================================
// TRAFFIC LIGHT HSM STATE: DAYTIME
//====================================
if( traffLightHSM.CurrState == TrafficLightHSM.DAYTIME ) {

	//if it is now Nighttime
	if( 0 >= DateTime.Now.Hour && DateTime.Now.Hour < 6 )
	{
		//enact transition to the NIGHTTIME state
		traffLightHSM.Transition( TrafficLightHSM.NIGHTTIME );
		//propagate transition through subordinate automata
		triLightFSM.Transition( TriLightFSM.RED );
		//reset output signal
		red = false; green = false; yellow = false;
		bright = true;
		//reset clock variable
		t = DateTime.Now;

		//print event message
		print ( "event[" + (eventNum++).ToString () + ", t=" +
		  TimeStamp () +
			"]--> HSM transitioning to NIGHTTIME." );
	}
	//otherwise
	{
		//-----------------------------------
		// TRI LIGHT FSM STATE: RED
		//-----------------------------------
		if( triLightFSM.CurrState == TriLightFSM.RED ) {

			//if it's time to become green
			if( DateTime.Now.Subtract(t).TotalMilliseconds >= 10000 ) {
				//enact transition to the GREEN state
				triLightFSM.Transition( TriLightFSM.GREEN );
				//set output signal state
				red = false; green = true;
				//reset clock variable
				t = DateTime.Now;

				//print event message
				print ( "event[" + (eventNum++).ToString () + ", t="+
					TimeStamp () +
					  "]--> triLightFSM transitioning to GREEN.");
			}
		}
		//-----------------------------------
		// TRI LIGHT FSM STATE: GREEN
		//-----------------------------------
		else if( triLightFSM.CurrState == TriLightFSM.GREEN ) {
```

```
                //if it's time to become yellow
                if( DateTime.Now.Subtract(t).TotalMilliseconds >= 10000 ) {
                        //enact transition to the YELLOW state
                        triLightFSM.Transition( TriLightFSM.YELLOW );
                        //set output signal state
                        green = false; yellow = true;
                        //reset clock variable
                        t = DateTime.Now;

                        //print event message
                        print ( "event[" + (eventNum++).ToString () + ", t="+
                                TimeStamp () +
                                  "]--> triLightFSM transitioning to YELLOW.");
                }
        }
        //-----------------------------------
        // TRI LIGHT FSM STATE: YELLOW
        //-----------------------------------
        else if( triLightFSM.CurrState == TriLightFSM.YELLOW ) {

                //if it's time to become red
                if( DateTime.Now.Subtract(t).TotalMilliseconds >= 10000 ) {
                        //enact transition to the RED state
                        triLightFSM.Transition( TriLightFSM.RED );
                        //set output signal state
                        yellow = false; red = true;
                        //reset clock variable
                        t = DateTime.Now;

                        //print event message
                        print ( "event[" + (eventNum++).ToString () + ", t="+
                                 TimeStamp () +
                                "]--> triLightFSM transitioning to RED." );
                }
        }
    }
}
//====================================
// TRAFFIC LIGHT HSM STATE: NIGHTTIME
//====================================
else if( traffLightHSM.CurrState == TrafficLightHSM.NIGHTTIME ) {

    //if it is now Daytime
    if( 6 <= DateTime.Now.Hour && DateTime.Now.Hour < 24 )
    {
        //enact transition to the DAYTIME state
        traffLightHSM.Transition( TrafficLightHSM.DAYTIME );
        //propagate transition through subordinate automata
        flashLightFSM.Transition( FlashingLightFSM.BRIGHT );
        //reset output signal
        dark = false; bright = false;
        red = true;
        //reset clock variable
        t = DateTime.Now;
```

```
            //print event message
            print ( "event[" + (eventNum++).ToString () + ", t=" +
              TimeStamp () +
                    "]--> HSM transitioning to DAYTIME." );
      }
      //otherwise
      {
            //-------------------------------------
            // FLASHING LIGHT FSM STATE: BRIGHT
            //-------------------------------------
            if( flashLightFSM.CurrState == FlashingLightFSM.BRIGHT ) {

                  //if it's time to become dark
                  if( DateTime.Now.Subtract(t).TotalMilliseconds >= 1000 ) {
                        //enact transition to the DARK state
                        flashLightFSM.Transition( FlashingLightFSM.DARK );
                        //set output signal state
                        bright = false; dark = true;
                        //reset clock variable
                        t = DateTime.Now;

                        //print event message
                        print ( "event[" + (eventNum++).ToString () + ", t=" +
                      TimeStamp () +
                                  "]--> flashLightFSM transitioning to DARK." );
                  }
            }
            //-------------------------------------
            // FLASHING LIGHT FSM STATE: DARK
            //-------------------------------------
            else if( flashLightFSM.CurrState == FlashingLightFSM.DARK ) {

                  //if it's time to become bright
                  if( DateTime.Now.Subtract(t).TotalMilliseconds >= 1000 ) {
                        //enact transition to the BRIGHT state
                        flashLightFSM.Transition( FlashingLightFSM.BRIGHT );
                        //set output signal state
                        dark = false; bright = true;
                        //reset clock variable
                        t = DateTime.Now;

                        //print event message
                        print ( "event[" + (eventNum++).ToString () + ", t=" +
                                    TimeStamp () +
                                    "]--> flashLightFSM transitioning to BRIGHT.");
                  }
            }
      }
}
```

When the code excerpt above is run in the Unity editor for several seconds, the following output is written to the console:

```
event[0, t=14:46:21:779]--> HSM transitioning to DAYTIME.
event[1, t=14:46:21:785]--> triLightFSM transitioning to RED.
event[2, t=14:46:31:794]--> triLightFSM transitioning to GREEN.
event[3, t=14:46:41:801]--> triLightFSM transitioning to YELLOW.
event[4, t=14:46:51:808]--> triLightFSM transitioning to RED.
event[5, t=14:47:1:815]--> triLightFSM transitioning to GREEN.
event[6, t=14:47:11:822]--> triLightFSM transitioning to YELLOW.
event[7, t=14:47:21:829]--> triLightFSM transitioning to RED.
event[8, t=14:47:31:840]--> triLightFSM transitioning to GREEN.
event[9, t=14:47:41:847]--> triLightFSM transitioning to YELLOW.
event[10, t=14:47:51:854]--> triLightFSM transitioning to RED.
```

The first event indicates that the traffic light HSM is making its initial transition to the *DAYTIME* state and—6 ms later—the tri-light FSM makes its initial transition to the *RED* state. Approximately 10 s later, the tri-light FSM transitions to the *GREEN* state where it remains for 10 s. The process continues until it is stopped, and it is evident that the HSM specification has been successfully implemented using the novel conversion method.

APPENDIX C: Pilot Study 1 Post-task Survey

The following questions comprised the self-report survey given to subjects in Pilot Study 1. There were nine questions in total: five regarding the system itself and four regarding the subject's own affective state. All of the responses were recorded on a 5-Likert scale.

1. Which best describes your experience of operating the vehicle?
    a) It was very difficult
    b) It was difficult
    c) Neutral
    d) It was easy
    e) It was very easy
2. How would you rate the visual quality of the objects?
    a) Very poor quality
    b) Poor quality
    c) Neutral
    d) Good quality
    e) Very good quality
3. Which best describes how you felt about the objectives of the trials?
    a) Very irrelevant
    b) Irrelevant
    c) Neutral
    d) Relevant
    e) Very relevant
4. How would you describe the clarity of the instructions given?
    a) Very confusing
    b) Confusing
    c) Neutral
    d) Clear
    e) Very clear
5. Which describes how you felt about the difficulty?
    a) It was very difficult
    b) It was difficult
    c) Neutral
    d) It was easy
    e) It was very easy

As best you can, rank your level of . . . [BLANK] when completing the assignment.

6. . . . enjoyment
    a) Really did not enjoy completing the assignment
    b) Did not enjoy completing the assignment
    c) Neutral
    d) Enjoyed completing the assignment
    e) Really enjoyed completing the assignment
7. . . . engagement
    a) Totally disengaged
    b) Mostly disengaged
    c) Neutral
    d) Mostly engaged
    e) Totally engaged
8. . . . frustration
    a) Frustrated the entire time
    b) Frustrated most of the time
    c) Neutral
    d) Frustrated once or twice
    e) Never frustrated
9. . . . boredom
    a) Bored the entire time
    b) Bored most of the time
    c) Neutral
    d) Bored once or twice
    e) Never bored

REFERENCES

[1] L. Evans, *Traffic safety and the driver*: Science Serving Society, 1991.

[2] M. Heron, "Deaths: Leading Causes for 2012," *National vital statistics reports: from the Centers for Disease Control and Prevention, National Center for Health Statistics, National Vital Statistics System,* vol. 64, pp. 1-94, 2015.

[3] A. M. Miniño, "Mortality among teenagers aged 12–19 years: United States, 1999–2006," *National Center for Health Statistics, NCHS data brief,* vol. 37, pp. 1-8, 2010.

[4] Centers for Disease Control and Prevention. (2015, 10/20). *Teen Drivers: Get the Facts*. Available: http://www.cdc.gov/nchs/products/citations.htm

[5] M. Albert and L. F. McCaig, "Emergency Department Visits for Motor Vehicle Traffic Injuries: United States, 2010–2011," *National Center for Health Statistics, NCHS data brief,* vol. 185, pp. 1-8, 2015.

[6] Federal Highway Administration, Office of Highway Policy Information, Highway Fincance Data Collection, U.S. Department of Transportation. (2011, 10/20). *Our Nation's Highways: 2011*. Available: https://www.fhwa.dot.gov/policyinformation/pubs/hf/pl11028/chapter4.cfm

[7] R. R. Mourant and T. H. Rockwell, "Strategies of visual search by novice and experienced drivers," *Human Factors: The Journal of the Human Factors and Ergonomics Society,* vol. 14, pp. 325-335, 1972.

[8] S. Classen and M. Monahan, "Evidence-based review on interventions and determinants of driving performance in teens with attention deficit hyperactivity disorder or autism spectrum disorder," *Traffic injury prevention,* vol. 14, pp. 188-193, 2013.

[9] S. Classen, M. Monahan, K. E. Brown, and S. Hernandez, "Driving indicators in teens with attention deficit hyperactivity and/or autism spectrum disorder Indicateurs de la conduite automobile chez les jeunes ayant un déficit de l'attention avec hyperactivité ou un trouble du spectre autistique," *Canadian journal of occupational therapy,* vol. 80, pp. 274-283, 2013.

[10] M. Monahan, S. Classen, and P. V. Helsel, "Pre-driving evaluation of a teen with attention deficit hyperactivity disorder and autism spectrum disorder," *Canadian journal of occupational therapy,* vol. 80, pp. 35-41, 2013.

[11] S. Classen, M. Monahan, and Y. Wang, "Driving characteristics of teens with attention deficit hyperactivity and autism spectrum disorder," *American journal of occupational therapy,* vol. 67, pp. 664-673, 2013.

[12] B. Reimer, B. Mehler, L. A. D'Ambrosio, and R. Fried, "The impact of distractions on young adult drivers with attention deficit hyperactivity disorder (ADHD)," *Accident Analysis & Prevention,* vol. 42, pp. 842-851, 2010.

[13] M. Rezaei and R. Klette, "Look at the driver, look at the road: No distraction! No accident!," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014, pp. 129-136.

[14] L. Fletcher and A. Zelinsky, "Driver Inattention Detection based on Eye Gaze—Road Event Correlation," *The international journal of robotics research,* vol. 28, pp. 774-801, 2009.

[15] C. Ho, N. Reed, and C. Spence, "Assessing the effectiveness of “intuitive” vibrotactile warning signals in preventing front-to-rear-end collisions in a driving simulator," *Accident Analysis & Prevention,* vol. 38, pp. 988-996, 2006.
[16] DSM-5 American Psychiatric Association, “Diagnostic and statistical manual of mental disorders,” *Arlington: American Psychiatric Publishing*, 2013.
[17] M. Wingate, R. S. Kirby, S. Pettygrove, C. Cunniff, E. Schulz, T. Ghosh, C. Robinson, L.-C. Lee, R. Landa, and J. Constantino, "Prevalence of autism spectrum disorder among children aged 8 years-autism and developmental disabilities monitoring network, 11 sites, United States, 2010," *MMWR Surveillance Summaries,* vol. 63, 2014.
[18] A. S. Weitlauf, M. L. McPheeters, B. Peters, N. Sathe, R. Travis, R. Aiello, E. Williamson, J. Veenstra-VanderWeele, S. Krishnaswami, and R. Jerome, "Therapies for children with autism spectrum disorder," 2014.
[19] C. Lord and S. L. Bishop, "Autism Spectrum Disorders: Diagnosis, Prevalence, and Services for Children and Families. Social Policy Report. Volume 24, Number 2," *Society for Research in Child Development,* 2010.
[20] P. Howlin, S. Goode, J. Hutton, and M. Rutter, "Adult outcome for children with autism," *Journal of Child Psychology and Psychiatry,* vol. 45, pp. 212-229, 2004.
[21] P. T. Shattuck, S. C. Narendorf, B. Cooper, P. R. Sterzing, M. Wagner, and J. L. Taylor, "Postsecondary education and employment among youth with an autism spectrum disorder," *Pediatrics,* pp. peds. 2011-2864, 2012.
[22] S. Classen, M. Monahan, and S. Hernandez, "Indicators of Simulated Driving Skills in Adolescents with Autism Spectrum Disorder," *The Open Journal of Occupational Therapy,* vol. 1, p. 2, 2013.
[23] J. Wade, D. Bian, J. Fan, L. Zhang, A. Swanson, M. Sarkar, A. Weitlauf, Z. Warren, and N. Sarkar, "A Virtual Reality Driving Environment for Training Safe Gaze Patterns: Application in Individuals with ASD," in *Universal Access in Human-Computer Interaction. Access to Learning, Health and Well-Being*, ed: Springer, 2015, pp. 689-697.
[24] J. Wade, D. Bian, L. Zhang, A. Swanson, M. Sarkar, Z. Warren, and N. Sarkar, "Design of a Virtual Reality Driving Environment to Assess Performance of Teenagers with ASD," in *Universal Access in Human-Computer Interaction. Universal Access to Information and Knowledge*, ed: Springer, 2014, pp. 466-474.
[25] D. Bian, J. W. Wade, L. Zhang, E. Bekele, A. Swanson, J. A. Crittendon, M. Sarkar, Z. Warren, and N. Sarkar, "A novel virtual reality driving environment for autism intervention," in *Universal Access in Human-Computer Interaction. User and Context Diversity*, ed: Springer, 2013, pp. 474-483.
[26] B. P. Daly, E. G. Nicholls, K. E. Patrick, D. D. Brinckman, and M. T. Schultheis, "Driving behaviors in adults with Autism Spectrum Disorders," *Journal of autism and developmental disorders,* vol. 44, pp. 3119-3128, 2014.
[27] B. Reimer, R. Fried, B. Mehler, G. Joshi, A. Bolfek, K. M. Godfrey, N. Zhao, R. Goldin, and J. Biederman, "Brief report: Examining driving behavior in young adults with high functioning autism spectrum disorders: A pilot study using a driving simulation paradigm," *Journal of autism and developmental disorders,* vol. 43, pp. 2211-2217, 2013.
[28] N. B. Cox, R. E. Reeve, S. M. Cox, and D. J. Cox, "Brief Report: Driving and young adults with ASD: Parents’ experiences," *Journal of autism and developmental disorders,* vol. 42, pp. 2257-2262, 2012.

[29] P. Huang, T. Kao, A. E. Curry, and D. R. Durbin, "Factors associated with driving in teens with autism spectrum disorders," *Journal of Developmental & Behavioral Pediatrics,* vol. 33, pp. 70-74, 2012.
[30] E. Sheppard, D. Ropar, G. Underwood, and E. van Loon, "Brief report: Driving hazard perception in autism," *Journal of autism and developmental disorders,* vol. 40, pp. 504-508, 2010.
[31] A. Klin, D. J. Lin, P. Gorrindo, G. Ramsay, and W. Jones, "Two-year-olds with autism orient to non-social contingencies rather than biological motion," *Nature,* vol. 459, pp. 257-261, 2009.
[32] M. Borgstede, J.-W. Schicke, F. Eggert, and U. Goltz, "Hybrid Automata as a Modelling Approach in the Behavioural Sciences," in *Proc. of the HAS Workshop HAS*, 2011.
[33] A. Schwarze, M. Buntins, J. Schicke-Uffmann, U. Goltz, and F. Eggert, "Modelling driving behaviour using hybrid automata," *IET Intelligent Transport Systems,* vol. 7, pp. 251-256, 2013.
[34] E. A. Lee and S. A. Seshia, *Introduction to embedded systems: A cyber-physical systems approach*: Lee & Seshia, 2011.
[35] Tobii Technology, "Accuracy and precision test method for remote eye trackers," 2011.
[36] B. Jensen, Keehn, B., Brenner, L., Marshall, S.P., Lincoln, A.J., Müller, R.A., "Increased Eye-Blink Rate in Autism Spectrum Disorder May Reflect Dopaminergic Abnormalities," presented at the International Meeting for Autism Research, 2009.
[37] C. J. Anderson, J. Colombo, and D. Jill Shaddy, "Visual scanning and pupillary responses in young children with autism spectrum disorder," *Journal of Clinical and Experimental Neuropsychology,* vol. 28, pp. 1238-1256, 2006.
[38] L. Zhang, J. Wade, D. Bian, J. Fan, A. Swanson, A. Weitlauf, Z. Warren, and N. Sarkar, "Multimodal Fusion for Cognitive Load Measurement in an Adaptive Virtual Reality Driving Task for Autism Intervention," in *Universal Access in Human-Computer Interaction. Access to Learning, Health and Well-Being*, ed: Springer, 2015, pp. 709-720.
[39] L. Zhang, J. W. Wade, D. Bian, A. Swanson, Z. Warren, and N. Sarkar, "Data Fusion for Difficulty Adjustment in an Adaptive Virtual Reality Game System for Autism Intervention," in *HCI International 2014-Posters' Extended Abstracts*, ed: Springer, 2014, pp. 648-652.
[40] D. Bian, J. Wade, A. Swanson, Z. Warren, and N. Sarkar, "Physiology-based affect recognition during driving in virtual environment for autism intervention," 2015.
[41] J. Fan, J. W. Wade, D. Bian, A. P. Key, Z. E. Warren, L. C. Mion, and N. Sarkar, "A Step Towards EEG-based Brain Computer Interface for Autism Intervention," 2015.
[42] J. T. Cacioppo, L. G. Tassinary, and G. Berntson, *Handbook of psychophysiology*: Cambridge University Press, 2007.
[43] K. Rayner, "Eye movements in reading and information processing: 20 years of research," *Psychological bulletin,* vol. 124, p. 372, 1998.
[44] C. Lord, S. Risi, L. Lambrecht, E. H. Cook Jr, B. L. Leventhal, P. C. DiLavore, A. Pickles, and M. Rutter, "The Autism Diagnostic Observation Schedule—Generic: A standard measure of social and communication deficits associated with the spectrum of autism," *Journal of autism and developmental disorders,* vol. 30, pp. 205-223, 2000.
[45] C. Lord, M. Rutter, P. C. DiLavore, S. Risi, K. Gotham, and S. L. Bishop, *Autism diagnostic observation schedule: ADOS-2*: Western Psychological Services Los Angeles, CA, 2012.

[46] C. Elliott, "Differential Ability Scales, Second Edition: Introductory and Technical Handbook," ed. San Antonio, TX: The Psychological Corporation, 2007.
[47] D. Wechsler, "Wechsler intelligence scale for children–Fourth Edition (WISC-IV)," *San Antonio, TX: The Psychological Corporation,* 2003.
[48] D. Wechsler, "Wechsler Abbreviated Scale of Intelligence, Second Edition (WASI-II)," ed. San Antonio, TX: Pearson Education, Inc., 2011.
[49] J. Constantino and C. Gruber, "Social Responsiveness Scale, Second Edition," ed. Los Angeles, CA: Western Pyschological Services, 2012.
[50] M. Rutter, A. Bailey, and C. Lord, *The social communication questionnaire: Manual*: Western Psychological Services, 2003.
[51] A. Vargha and H. D. Delaney, "A critique and improvement of the CL common language effect size statistics of McGraw and Wong," *Journal of Educational and Behavioral Statistics,* vol. 25, pp. 101-132, 2000.
[52] A. Sarkar, J. Wade, and Z. Warren, "Understanding and Improving Collaborative Skills Among Individuals with ASD in a Distributed Virtual Environment," in *Universal Access in Human-Computer Interaction. Access to Learning, Health and Well-Being*, ed: Springer, 2015, pp. 669-680.