A publication of ISCA*:
International Society for Computers
and Their Applications

INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS

TABLE OF CONTENTS

*** "International Journal of Computers and Their Applications is abstracted and indexed in INSPEC and Scopus."**

Volume 26, No. 1, March 2019 **ISSN 1076-5204**

International Journal of Computers and Their Applications

A publication of the International Society for Computers and Their Applications

EDITOR-IN-CHIEF

Dr. Frederick C. Harris, Jr., Professor
Department of Computer Science and Engineering
University of Nevada, Reno, NV 89557, USA
Phone: 775-784-6571, Fax: 775-784-1877
Email: Fred.Harris@cse.unr.edu, Web: http://www.cse.unr.edu/~fredh

ASSOCIATE EDITORS

Dr. Hisham Al-Mubaid
University of Houston-Clear Lake,
Houston, TX, USA
hisham@uhcl.edu

Dr. Ajay Bandi
Northwest Missouri State University,
Maryville, MO, USA
ajay@nwmissouri.edu

Dr. Antoine Bossard
Kanagawa University,
Graduate School of Science
Kanagawa, JAPAN
abossard@kanagawa-u.ac.jp

Dr. Sergiu Dascalu
University of Nevada,
Reno, NV, USA
dascalus@cse.unr.edu

Dr. Sami Fadali
University of Nevada,
Reno, NV, USA
fadali@ieee.org

Dr. Takaaki Goto
Ryutsu Keizai University,
Ibaraki, JAJPAN
tg@gotolab.net

Dr. Vic Grout
Glyndŵr University,
Wrexham, UK
v.grout@glyndwr.ac.uk

Dr. Wen-Chi Hou
Southern Illinois University,
Carbondale, IL, USA
hou@cs.siu.edu

Dr. Ramesh K. Karne
Towson University,
Townson, MD, USA
rkarne@towson.edu

Dr. Sharad Sharma
Bowie State University,
Bowie, MD, USA
ssharma@bowiestate.edu

Dr. Yan Shi
University of Wisconsin-Platteville,
Platteville, WI, USA
shiy@uwplatt.edu

Dr. Junping Sun
Nova Southeastern University,
Fort Lauderdale, FL, USA
jps@nsu.nova.edu

Dr. Rui Wu
East Carolina University,
Greenville, NC, USA
WUR18@ecu.edu

Dr. Beifang Yi
Salem State University,
Salem, MA, USA
byi@salemstate.edu

ISCA Headquarters···•···P. O. Box 1124, Winona, MN 55987 USA···•···Phone: (507) 458-4517
E-mail: isca@ipass.net • URL: http://www.isca@isca-hq.org.

Copyright © 2018 by the International Society for Computers and Their Applications (ISCA)
All rights reserved. Reproduction in any form without the written consent of ISCA is prohibited.

Editor's Note: March 2019

This is an important note since it serves as the transition of Editorship of the IJCA as I (Fred Harris) hand off the journal to Dr. Gordon Lee. I have served as editor for the past 5 years and Dr. Lee will be serving as the editor for the next year as the journal transitions to another long-term editor in the near future.

It has been my distinct honor, pleasure and privilege to serve as the Editor-in-Chief of the International Journal of Computers and Their Applications (IJCA). I have a special passion for the International Society for Computers and their Applications and I look forward to continuing to assist with ISCA Conferences.

We would like to begin this volume by giving a review of this past year. In 2018 we had 28 articles submitted to the International Journal of Computers and Their Applications. We currently have 5 that are still under review. As a reminder, the journal will not be accepting articles that are less than 10 pages. The authors of papers with less than this number of pages will be encouraged to submit their papers to ISCA conferences.

We are still working towards getting IJCA online. Hopefully we can end up with a nice repository soon.

I (Dr. Gordon Lee) look forward to working with everyone in the coming year to maintain and further improve the quality of the journal. I would like to invite you to submit your quality work to the journal for consideration of publication. I also welcome proposals for special issues of the journal. If you have any suggestions to improve the journal, please feel free to contact me.

Gordon Lee
Professor Emeritus
San Diego State University
Email: glee@sdsu.edu

This year we have 4 issues planned (March, June, September, and December). We begin with a special issue from the best papers at the ISCA Fall 2018 SEDE Conference, followed by a special issue from CAINE 2018. We have a proposal for the best papers from the ISCA Spring Conference cluster (CATA/BICOB) which will appear in the September issue. The December issue is being filled with unsolicited accepted papers.

I would also like to announce that we have begun a search for a few Associate Editors to add to our team. There are a few areas that we would like to strengthen our board with, such as Image Processing and Cyber Intelligence. If you would like to be considered, please contact me via email with a cover letter and a copy of your CV. We look forward to continue with our high-quality journal, sponsored by the International Society for Computers and Their Applications.

Frederick C Harris, Jr.
Editor-in-Chief 2013-2018

Gordon Lee
Editor-in-Chief 2019-2020
glee@sdsu.edu

Guest Editorial:
Special Issue from ISCA Fall--2018 SEDE Conference

This Special Issue of IJCA is a collection of four refereed papers selected from the SEDE 2018: 28th International Conference on Software Engineering on Data Engineering, held during October 8-10, 2018, in New Orleans, LA, USA,

Each paper submitted to the conference was reviewed by at least two members of the International Program Committee, as well as by additional reviewers, judging the originality, technical contribution, significance and quality of presentation. After the conferences, five best papers were recommended by the Program Committee members to be considered for publication in this Special Issue of IJCA. The authors were invited to submit a revised version of their papers. After extensive revisions and a second round of review, these four papers were accepted for publication in this issue of the journal.

The papers in this special issue cover a broad range of research interests in the community of computers and their applications. The topics and main contributions of the papers are briefly summarized below.

JAMES STIGALL, SRI TEJA BODEMPUDI, and SHARAD SHARMA of BOWIE State University, Bowie, MD USA and DAVID SCRIBNER and JOCK GRYNOVICKI of the Army Research Laboratory, Aberdeen Proving Ground, MD USA presented their paper “Use of Microsoft HoloLens in Indoor Evacuation,” where they demonstrate how augmented reality (AR) technology can provide spatial context 3D visualization that promotes spatial knowledge acquisition and support cognitive mapping. This paper presents the research and development of prototype AR application for the communication of indoor evacuation information in real-world spaces by the use of evacuation maps.

PATTAPHOL JIRASSESSAKUL, ZACHARY WALLER, PAUL MARQUIS, CONNOR SCULLY-ALLISON, VINH LE, SCOTTY STRACHAN, FREDERICK C. HARRIS, JR., and SERGIU M DASCALU of the University of Nevada, Reno, USA presented their paper “Simplifying Data Visualization Pipelines with the NRDC-CHORDS Interface,” which proposes an alternative approach for the Nevada Research Data Center (NRDC) to visualize environmental data in near-real time and confirm its viability for usage with other research projects of similar size. They evaluated the success of their implementation by comparing metrics of use, determining that both iterations of our software were much faster and easier to use than CHORDS built-in configuration interfaces.

AAKANKSHA RASTOGI and KENDALL E. NYGARD of the Department of Computer Science, North Dakota State University, Fargo, ND, USA presented their paper “Trust and Security in Intelligent Autonomous Systems,” which surveys and explores concepts of trust in terms of relationships between humans and systems. An ontology that characterizes this relationship is also provided. Trust issues as they pertain to the areas of cybersecurity and autonomy are characterized and the concept of anti-autonomy and counter measures that apply to autonomous weapon systems is also included.

ALEX REDEI of Central Michigan University, Mt. Pleasant, MI, USA and SERGIU DASCALU and FREDERICK C HARRIS, JR of the University of Nevada, Reno USA presented their paper “A Framework for Virtualizing Joystick Controls in a Flight Simulator Training Environment,” in which they explored software arbitration of two joysticks controlled by two pilots, where each joystick is independent of the other and each pilot is potentially equally valid. In this paper, they detail their hardware and software framework for arbitrating conflicts in a multi-axis joystick system, thereby increasing the responsiveness of control input in a potentially conflicted state.

As guest editors we would like to express our deepest appreciation to the authors and the program committee members of the conference these papers were selected from.

We hope you will enjoy this special issue of the IJCA and we look forward to seeing you at a future ISCA conference. More information about ISCA society can be found at http://www.isca-hq.org.

Guest Editors:

Frederick C. Harris, Jr, University of Nevada, Reno, USA, SEDE 2018 Conference Chair
Sergiu M. Dascalu, University of Nevada, Reno, USA, SEDE 2018 Program Chair
Sharad Sharma, Bowie State University, USA, SEDE 2018 Program Chair

March 2019

Use of Microsoft HoloLens in Indoor Evacuation*

James Stigall†, Sri Teja Bodempudi†, Sharad Sharma†,
Bowie State University, Bowie, MD, 20715, USA

David Scribner, Jock Grynovicki, Peter Grazaitis
Army Research Laboratory, Aberdeen Proving Ground, MD, 21005, USA

Abstract

It is critical that building patrons know how to evacuate a building in the event of an emergency. Evacuation training eliminates (or reduces) the likelihood of injury and fatalities suffered by patrons during indoor crises such as a fire, an earthquake, or an active shooter event. Architectural complexity of the built environment can make it difficult for users to visualize and create a mental representation of the 3D space. We demonstrate how augmented reality (AR) technology can provide spatial context 3D visualization that promotes spatial knowledge acquisition and support cognitive mapping. This paper presents the research and development of prototype AR application for the communication of indoor evacuation information in real-world spaces by the use of evacuation maps. We show how AR can be used as an emerging technology for communicating building information to the patrons. The application offers the users a more realistic perspective of the building, compared to 2D floor plans. This paper describes the application's design and implementation and reports the results of a pilot study conducted to evaluate it. The results demonstrate the application's effectiveness in assisting users during an emergency evacuation.

Keywords: Augmented reality, immersive AR, Microsoft HoloLens, building evacuation.

1 Introduction

Death and injuries occur because proper evacuation procedures are either not given or followed. During the 2011 Fukushima accident, 800 hospital patients were evacuated using buses and emergency vehicles. In spite of good intentions, the evacuation process lasted for several hours. In fact, some evacuations lasted for 48 hours. At least 50 elderly patients died during this evacuation [13]. In 2005, Hurricane Rita made landfall near the Louisiana-Texas border. Even though, 110,000 residents fled to safety. However, 107 people died during the evacuation, which saw many issues: one, traffic jams occurred because there were more people evacuating via road than predicted; two, evacuees did not have the appropriate information(or resources) to locate available shelter; and three, cell phone networks were jammed meaning people could not effectively communicate risk by phone [6]. In Santa Maria, Brazil, 230 people died trying to evacuate the Kiss nightclub because they trampled on top of each other while evacuating and the main exits were blocked by security guards [24]. If proper evacuation guidance had been given before the Fukushima accident, Hurricane Rita, and the Kiss nightclub incident, the fatalities seen in those events would not have occurred.

Motivated by the learning potential that AR provides and the critical need for evacuation training, an AR application was built for both HoloLens and mobile devices to help users evacuate a building. Emerging research is proving AR to be an exceptional teaching and training tool because it provides a mechanism for users to retain what they have learned. Additionally, users do not have to be physically present in the simulated environment. So far, two proven methods have been used to aid building patrons of proper evacuation procedures: live evacuation drills and two-dimensional (2D) floor plans. Those two methods, however, can sometimes be ineffective for two reasons – one, people may not have the time to participate in live evacuation drills, two, 2D floor plans do not provide a detailed-enough viewpoint that would allow patrons to know the building's layout. This paper presents an augmented reality (AR) application to enhance the evacuation process by giving all building patrons the opportunity to familiarize themselves with all of the building exits and the path(s) to take to get to each of them. This should improve evacuation time and eliminate the injuries and fatalities occurring during emergencies such as building fires and active shooters. The application was built using Unity 3D and Vuforia AR Toolkit for Microsoft HoloLens. It incorporates existing 2D floor plans in the building which are used as markers so that when the HoloLens detects those floorplans, 3D versions of those floorplans are generated. Figure 1 shows a 3D floor plan of the building as seen through the HoloLens. It also shows the user his present location within the floor being shown. This study offers the following contributions:

- A HoloLens application developed to help users visualize a building, their current locations within the building, and how to exit the building all in a 3D space;
- Voice command operations and calculation of the best path to navigate the user to the nearest exit; and

---

* Extended paper from Proceedings of 27th International Conference on Software Engineering and Data Engineering (SEDE 2018), New Orleans, Louisiana, pp. 20-25, USA, October 8-10, 2018.

† Department of Computer Science.

- A usability evaluation proving the effectiveness of the proposed HoloLens application during the evacuation.

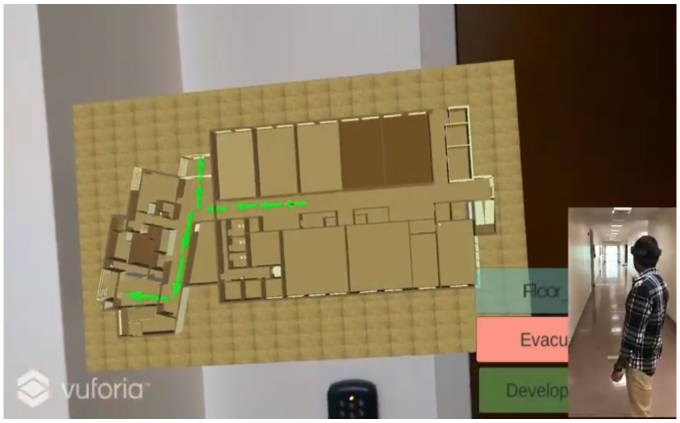


Figure 1: View of the floor plan from HoloLens using existing 2D plan as a marker

The HoloLens application enables people to remain connected to the real world while enriching the environment with more 3D data by enhancing the evacuation process. The HoloLens application gives a visual representation of a building floor plan in three-dimension, allowing people to see where exits are in the building as well as their location. AR comprises the user viewing the real world along with computer-generated graphics juxtaposed upon the real world. Whereas virtual reality *completely* replaces the user's physical environment, AR *supplements* it. Unlike virtual reality, AR has many uses beyond gaming or entertainment. Okuno et al [22] proposed an AR tool to improve the balance of elderly people to prevent their falling. Liu et al [21] developed an AR system for training users on how to perform minimally invasive spine surgery. Fruend et al [6] built an AR tool to train assembly line workers about new and complex assembly processes. AR has been proven to be particularly useful in the education and training domain. AR has been shown to enhance concentration and motivation amongst students using it to learn a specific concept and to help students learn by taking what would be a complex subject and make that subject easier to learn to allow students to visualize it in a 3D space and imagining what that subject entails [2]. Popular opinion suggests that using AR in education would be a more effective teaching strategy than traditional teaching methods [33].

Motivated by the visual appeal of AR and the dire need for evacuation training, an AR application was built to train people on the evacuation of the Computer Science Building at Bowie State University. The application was built for Microsoft HoloLens, a pair of mixed reality smart glasses running the Windows 10 operating system that allows users to interface with applications using hand gestures. The proposed application was built using Unity3D, the Vuforia AR Toolkit, and the Universal Windows Platform (UWP). This paper also shows the evaluation of the AR application for its effectiveness in evacuation training. Section 2 briefly describes the work done previously. The proposed AR application and system architecture is described in more detail in Section 3. Section 4 illustrates the proposed implementation and deployment of the AR application for HoloLens and tablet/phone. The simulation and its results are given in Section 5. Section 6 lists conclusion and future work.

2 Related Work

2.1 AR Applications in Education and Training

The Augmented Reality Sandbox, presented in [19], simulates the changes in the Earth's topography using a sandbox with a Kinect 3D camera mounted above it. When users make changes to the topography found in the sandbox using a rake, the camera senses the changes in the distance to the sand and the contour lines and projects those changes to a computer with the AR Sandbox software installed on it. The computer takes the data from the camera, simulates the changes using warm colors for peaks and cool colors for depressions. The computer beams the simulation onto the sandbox using a projector.

A system proposed and evaluated in [12] displays debuging information atop an embedded board, the image of which is captured by a camera connected to a laptop. With this system, students are able to see the connection status between different components on the board, information about where certain components are, the status of programs deployed and executed on the board, and the like. An AR system teaching nursing and medical students how to read a pulse was discussed in [16]. The system is comprised of a mannequin's hand containing an AR marker located at the pulse point. The hand is connected to a haptic device that simulates the pulse rate that the user is reading. It is also comprised of a webcam which continuously captures an image of the hand and sends that image to the computer. If the user touches the correct point on the hand, the computer tabulates the pulse rate and sends that information to the haptic device. A user study consisting of 26 participants found that the system provides a realistic experience while being easy to use and effective at teaching pulse measuring.

Besbes et al [3] built a system that trains users on industrial maintenance tasks. It is comprised of an optical see-through HMD (OST HMD) and a laser pointer. The system uses two localization techniques to accurately position and orient the user's viewpoint. The first technique involves detecting sharp edges sampled from a 3D CAD model. The second involves detecting the laser dot's location relative to a selected object so that instructions regarding the object can be overlaid on top of it. An evaluation of the system determined that the system is easy to use and that it is very useful for selecting potentially dangerous objects.

Chen et al [7] developed the AR-based Engineering Graphics System where pages of a book act as markers. When users (assumingly, engineering graphics students) point their devices' camera to a page containing a marker, 3D virtual models are generated and overlaid on top of the marker. These models, developed using 3ds MAX, emulate objects related to engineering such as worming wheels casing. The system was developed to give Engineering Graphics students a better visual perspective when viewing illustrations 3D objects as typical illustrations (such as the ones seen in textbooks) hides certain sides of the objects, preventing students from fully studying them. Presenting these objects in AR, as Chen et al argues, allows students to see all sides of these objects since students can directly interact with them in this system by rotating them.

Kim et al [17] built an AR system teaching users about PLC wiring. The construction of the system discussed in [17] was motivated by the onset of more complex wiring mechanisms and the need to teach trainees about these mechanisms. It was also developed to eliminate the distractions encountered by the trainee when he or she has to look back and forth between the textbook (or manual) and the training board, preventing the trainee to fully master PLC wiring. When the trainee attempts to do the wiring, a virtual wiring guide is augmented on the training board. Once the trainee has done the wiring, the system takes an image from the camera of the board, converts it into binary code, and stores the coordinates of the ports. Next, the system determines which ports are connected. Lastly, the system checks to see if the wiring is correct. If it is correct, a message is displayed to the user.

Birt et al [4] implemented a system giving soon-to-be paramedics airways management skills. Specifically, the system teaches them about how to conduct direct laryngoscopy with foreign body removal. Using the system, students use 3D printed laryngoscope, Macintosh blade, and Magill forceps – tools needed to conduct the laryngoscopy. To interface with the system, users wear a hat with a mobile phone mounted on its bill. The system's display consists of a virtual patient lying in front of the user with its mouth open, allowing the user to conduct the laryngoscopy so that a foreign body can be removed from it. The display also consists of the steps the user needs to take to do the procedure. Srivastava [26] proposed an AR application teaching students electronics engineering. When the user holds the device towards a specific image in the student's lab manual, a corresponding video is displayed providing instructions for the experiment. The device also features an Intelligent Breadboard which helps students debug circuits by providing voice-based assistance indicating specifics about the circuit bugs. A user study concluded that students and instructors testing out the application found it to be useful.

Blum et al [5] proposed an AR "magic mirror" that gives the user the illusion that he or she is looking at his or her anatomy. The system projects a CT scan onto the user. The CT scan the user is viewing, though, is not a real-time one of him- or herself, but is a simulated one derived from the Visible Korean Human dataset which scaled to the user's body dimensions and superimposed onto the user. The system in [5] was built for Microsoft Kinect which is used to track the user's body movements and gestures.

An Android-based AR system training neurosurgical students was developed in [23]. Specifically, the system trains users on performing an external ventricular drain (EVD), a common procedure where the cerebrospinal fluid is drained so that intracranial pressure is relieved. The system is composed of a NeuroTouch simulator which contains a mannequin head and a pointing tool so that the user can practice EVD by navigating the mannequin head. The system was developed for Android tablets. When the user points a tablet's camera towards the head, a simulated volumetric scan is juxtaposed atop the head. A user study evaluating the system determined that the participants' accuracy and speed in terms of performing the EVD were enhanced when using the system.

Yaman and Karakose [31] presented a system that uses image processing techniques to clarify images for use in AR. The goal of the system is to enhance images so that the users who are using them in an educational context can better retain the knowledge gained from those images. The system, notably, uses edge extraction which identifies the main lines seen in images. Yaman and Karakose used two sample images to examine the system. The first image was a satellite picture taken of Istanbul using Google Maps. The edge extraction technique distinguished Istanbul's geographical features such as seas, mountains, and residential areas. The second image was a cell, the features of which were also distinguished using edge extraction such as its edges and composition.

Vidal et al [28] have proposed Intramuros – an AR, narrative-based, adventure game where users navigate through Intramuros, Manila, the Philippines using historical markers located at historical sites throughout the city. Along the way, users can "help" troubled historical figures suchas Jose Rizal and Pope John Paul II. The game features an in-game 3D map that assists users in navigating towards the markers. Wichrowski [30] have also reported on two AR projects targeting art students.

Huang et al [15] developed AR-View, a fixed-position telescope that allows users to view a digitally created image of the Dashuifa, a group of fountains encompassed within the ruins of Yuanmmingyan, a royal garden built for the Qing Dynasty that was burnt down in 1860 by Anglo-French soldiers. Users insert a coin into the device before viewing a virtual image of the Dashuifa juxtaposed within the real-life view of Yuanmmingyan. A user study evaluating AR-View determined that the users found that the device was interesting and that it was useful for learning about Yuanmmingyan.

Yangguang et al [32] discussed the Multiplayer Collaborative Training System (MCTS), an AR system where users collaborate with other users while working in a closed space using a mobile AR application. As part of their user study, Yangguang et al utilized an instructional AR game known as Time Walker where players work in groups to solve puzzles pertaining to famous historical figures such as Aristotle or Dante. In the user study, participants were split amongst two environments – one developed using MCTS and the other being an environment without the AR and virtual elements. Those solving puzzles in the MCTS environment did so in less time (on average) than those who did so in the non-AR environment.

2.2 AR Applications for Evacuation Training

An AR application in [14] was built to simulate torrential rain and to provoke a significant enough feeling of risk that would lead them to evacuate the premises. Hirokane et al performed an analysis based on the Human Cognitive Reliability (HCR) which categorizes human responses into three different types: skill base, rule base, and knowledge base. Skill base involves seamlessly responding to an event without much thinking; rule base involves responding to an event based on a rule, and knowledge base involves responding to an event by conceptualizing it when it occurs and reacting to it based on the conceptualization. A user study with the system built in [14] found that women respond to disaster using the skill base or rule base response while men respond to disaster using the knowledge base response. This finding confirmed that women had a higher evacuation rate than men when a disaster occurs. Hirokane et al concluded that implementing training that favors the skill base response could enhance the evacuation rate.

Stigall and Sharma [27] built a mobile augmented reality application (MARA) that helps users evacuate a building. When users hold their mobile devices up to a marker, a 3D floorplan corresponding to that maker is generated. The key feature of the Android-based system is the inclusion of “intelligent signs” – visual cues that help users locate each exit in the building and pinpoints a path to those exits. The application also features avatars to indicate the direction to take when evacuating and virtual fire and smoke to invoke the feeling of urgency in the user. Another MARA was built for building evacuation in [25]. Similar to the application developed in [27], a 3D floorplan is generated when the user holds up the mobile device towards the appropriate marker. A user study evaluating the application found that a majority of the participants thought that the application would help them evacuate during a real emergency and that the application is a suitable substitute for a 2D floorplan.

RescueMe was implemented and evaluated in [1]. The AR application helps users evacuate a building by prompting him or her to take a photo of the closest room number. That photo is sent to a server which calculates the exit times of all possible exit paths based on the user’s walking speed. The server sends back the path with the lowest exit time. A simulation evaluating RescueMe found that, compared to not using an algorithm for evacuation and using the shortest path algorithm for evacuation, found that the RescueMe algorithm either yielded the lowest evacuation time or the same evacuation time as the shortest path algorithm.

Lin et al [20] proposed the PEAR (Personal Evacuation and Rescue) system which provides up-to-date, localized evacuation routes to the user. The system is comprised of four components: PEAR ERCs (emergency response centers), PEAR Local ERCs, PEAR appliances, and PEAR end-user mobile devices. PEAR ERCs combines all emergency information from one area to PEAR Local ERCs and devices. PEAR Local ERC collects data from users and oversees all activities within one building. PEAR appliances are electronic information boards broadcasting evacuation plans to anybody reading them. Lastly, PEAR end-user mobile devices provide users with real-time emergency information and evacuation routes. In the PEAR system, AR is utilized to navigate users along evacuation paths. The end user devices in this system use their cameras to provide users the present view of their surroundings with guidance information augmented to it.

2.3 Microsoft HoloLens

Kučera et al [18] developed an application for the HoloLens with the goal of improving the education of students learning Applied Mechatronics and Automotive Mechatronics. The application features an electronic cart model that can be seen via the HoloLens. Along with the cart, users can see a description of its parts, information about similar vehicles, and a video on e-mobility. Hanna et al [11] discussed pathology residents using the HoloLens application to perform autopsies while drawing within their surrounding environment and communicating with other users in real time. The system discussed in [11] was also used to juxtapose radiographs atop of their respective gross specimens. Those using the system found the HoloLens to be easy to wear and use and thought that it provided adequate processing power for their purposes.

Additional studies involving the HoloLens in education and training include a course in Crisis Management and Communication, reported by Vold et al [29], where students were tasked with designing and implementing emergency scenarios using the HoloLens along with VBS3 (Virtual Battle Space 3 from Bohemia Simulations), a VR simulation tool within which the disaster environment was simulated, and Rayvn, a text-based communication tool. Lastly, Handosa et al [10] utilized an approach for teaching nursing skills that combines the HoloLens with Microsoft Kinect so that users can interface with a mixed reality application using whole body interaction.

3 System Architecture

HoloLens is an augmented reality device which has a capacity to superimpose computer world data onto real-world data. HoloLens contains one main camera and two sensors on each side of the glasses. These two sensors scan the user’s surroundings and detect the user’s gestures. Voice commands and gestures act as the two main inputs for the device. The main disadvantage in HoloLens is that it does not contain a GPS (or any other location-finding capabilities). This means that the user’s present location cannot be accessed using the HoloLens. This project requires location-finding capabilities to find the user’s current location to guide him or her towards the safest and shortest exit. Image targets can help to identify the current user location by scanning and capturing surrounding images. The device can be seen in Figure 2 [8].

The HoloLens runs the Windows 10 operating system. The

Figure 2: Microsoft HoloLens [8]

device can be connected to the Internet and other devices wirelessly via WiFi (IEEE 802.11ac standard) and Bluetooth 4.1. For position and orientation, it contains an accelerometer, a gyroscope, and a magnetometer. As far as audio is concerned, speakers and microphones are built into the HoloLens. It is comprised of four cameras to detect the user's surroundings and one depth camera with a 120°x120° angle of view. It features a Holographic Processing Unit capable of performing 1 trillion computations per second. It also features 2 GB of RAM, 64 GB of flash storage, and four Intel Atom x5-Z8100 Intel Airmount Logical Processors. Lastly, the HoloLens's battery allows for 2-3 hours of active use or two weeks of standby time [30], [33].

The aim of this project was to develop a tool that will help users evacuate a building safely. As a prototypical example, it features the Computer Science Building located on the campus of Bowie State University. This tool detects the current location of the user, detects the location of the fire, and calculates the safest and the nearest path to the exit. Additionally, HoloLens implements voice command operations and draws a safe path to navigate the user to the safest exit. The main challenge of this project is to identify the user's location. This challenge was overcome by using existing objects (images) in the building. For example, existing 2D floor plans in the building were used for location detection as well as for superimposing 3D floor plans on top of them. Figure 3 shows the existing signboards of the floor plans in the building.

Figure 3: Existing signboards of different floor plans in the building a) first floor plan, b) second floor plan, c) third floor plan

Drawing floor plans in Unity 3D can be challenging, but is much easier in SketchUp; thus, the latter was used to create the needed floor plans in 3D. After creating all three floor plans in SketchUp, those floor plans were converted into a Unity-compatible format. Vuforia was also incorporated into our project. Vuforia helped enable image detection services in Unity so that when the HoloLens detects these images, the appropriate floor plan can be generated and juxtaposed on top it.

Users wear the HoloLens with the application installed on it. Using the bloom hand gesture in front of the HoloLens camera, the user can select the developed AR application from the list of installed applications. After the application launches, the HoloLens searches for image targets (markers) that have already been loaded into the application via Vuforia. Some image targets from inside the building are shown in Figure 2. Whenever the user faces a target, HoloLens detects it and generates the appropriate 3D floor plan with the user's current location shown within it. Afterward, the application calculates the distance between the user and the exit and determines the safest and shortest path. Lastly, HoloLens directs the user to the exit by displaying green arrows in the floorplan. The user can then follow those green arrows to the exit.

Figure 4 shows the system architecture diagram of the built application. The system architecture was designed in three stages. The three stages involved were the following: first, the creation of 3D assets, second, the development of the HoloLens application, and third, the deployment of the HoloLens application. The initial step for this process was to create a 3D model of each floor plan for the building. This was done by using SketchUp. After completing the 3D floor plan in SketchUp, they were converted into the fbx format so that Unity can use it. Unity cannot handle 3D models developed in another format so it was necessary to convert each floor plan into the fbx format. In the second stage, app development took place. For this stage, different capabilities available in Unity were exploited such as C# scripting, animation, 3D assets, images, and textures. In the third stage, the application was deployed onto HoloLens. Deploying from Unity to HoloLens led to many challenges like connection issues, packets missing etc. To overcome this problem, an apk file was generated. Next, the apk file was opened in Visual Studio. The HoloLens was connected to the computer through a USB cable. By choosing HoloLens as the targeted device in Visual Studio, the application was onto HoloLens. Once the deployment completed successfully, application appeared in HoloLens's main menu.

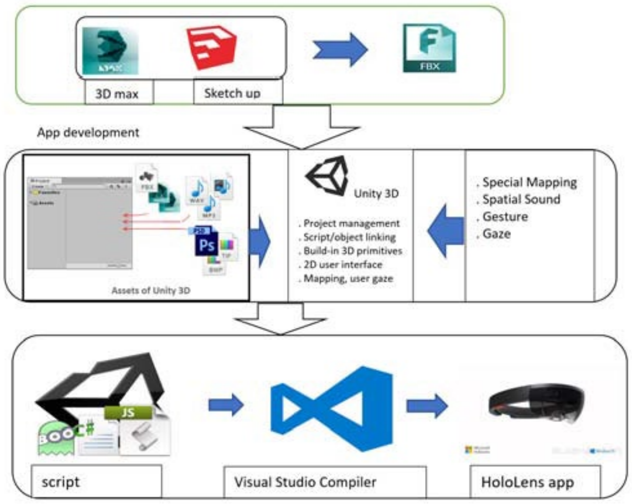


Figure 4: System Architecture diagram

4 Implementation and Deployment

This paper proposes an architecture and a method to leverage the Microsoft HoloLens for building evacuation purposes. Figure 4 shows the full life cycle of this project, with different stages involved in developing the HoloLens application. The creation of the application as a Unity project is shown as the first stage. Transferring the built Unity project to Visual Studio is shown as the second stage. Opening the project up in Visual

Studio and letting the HoloLens pair to Visual Studio forms the next stage. After successfully pairing to HoloLens, the last stage is deploying the application to the HoloLens where the user can open and use it through the use of gestures. The implementation and deployment of the application were completed in four phases:

Phase 1: 3D floor plans for each of three floors in the Computer Science Building were created using SketchUp. Other objects such as desks, chairs, tables, and computers were drawn and added onto the floors in SketchUp. Each floor was saved as a SketchUp file and converted to the fbx format so that it can be opened up in Unity 3D. There, smoke, fire, and other animations were added.

Phase 2: A project was created in Unity 3D within which the proposed application was built. Vuforia was used to support the application's image detection capabilities. Using Vuforia, all of the image targets were uploaded as the markers. Also, each of the floors drawn in SketchUp was integrated into Unity 3D, where arrows and other shapes were added onto each floor to help the user determine where he or she is on the floor and how to get to the exit. The data file containing the uploaded image targets was downloaded. The floor plans were placed atop of their corresponding markers so that when the HoloLens targeted one of the markers, the appropriate floor plan can be generated.

Phase 3: Once the floor plans have been properly built in Unity 3D, the project was opened in Visual Studio. There, additional functionalities were added into the application such as path calculation to enable the application to determine the best path that the user should take to get to the exit and button actions to allow the user to toggle features (i.e. developer information) on and off. After that was complete, the application was compiled in Visual Studio, built as a mobile application (meaning that it would be compatible with other mobile devices, such as phones and tablets) and deployed to HoloLens. The HoloLens was connected to the computer through a USB cable so that after the application was built, it would be transferred to the HoloLens to be installed there.

Phase 4: Upon being deployed to and installed on the HoloLens, the user can open the application using HoloLens's Start menu. When the user opens the application, the camera on the HoloLens scans the environment for the existing signboards throughout the building. Pictures of these signboards were taken and uploaded to Vuforia as the markers in Phase 2 so that the application knows what to detect. When the HoloLens detects one of these images, a floor plan is generated as shown in Figures. 5-7.

5 Simulation and Results

A limited user study was conducted to evaluate the effectiveness of the proposed HoloLens AR application. The study illustrated its success and demonstrated the effectiveness of the application in an emergency evacuation. Responses were collected from 10 participants, 80% of whom were male and 20% were female. The post-test part of the questionnaire measured participant's perceptions of motivation, usability, educational and training effectiveness, and AR applications (HoloLens, Mobile phone, and Tablet) appropriateness. Figure 8 shows that the majority of the users (60%) felt that HoloLens was more suitable for evacuation than with a tablet. The following questions were asked in the user study:

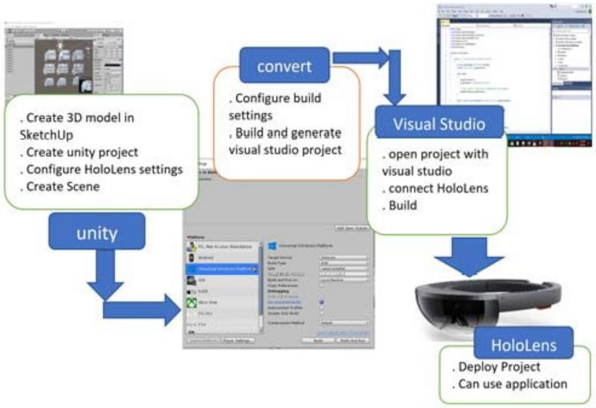


Figure 5: Build life cycle of the project

Figure 6: Projecting floor plan on targeted image for the first floor

Figure 7: Projecting floor plan on targeted image for the second floor

- Do you consider this system useful in unknown buildings with a complex structure?
- Will viewing this HoloLens App help during the real-time evacuation
- Substitute for evacuation plans (2D plan) in a building
- Used for educational or training purposes in evacuation

Figure 8: Projecting floor plan on targeted image for the third floor

The HoloLens application received more positive answers regarding usability as shown in Figure 9. All of the participants (100%) felt that this system will be useful in unknown buildings with a complex structure while 90% of the participants felt that viewing this HoloLens app will help during a real-time evacuation.

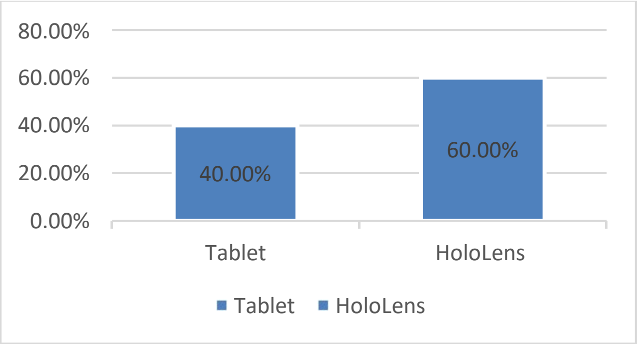


Figure 9: Device suitability for the study

Additionally, 90% of the participants felt that the HoloLens application can be used as a substitute for evacuation plans (i.e., 2D plans) seen throughout the building. Sometimes, it becomes difficult for users to visualize a building through 2D plans. The use of an AR application, such as the one described in this paper, equips the user to visualize the building and its exits in a 3D space.

6 Conclusion

An AR application was developed for Microsoft HoloLens offering users an enhanced evacuation experience by featuring enthralling visuals. This will help them learn the best evacuation path to use during a situation where evacuation is necessary. The application was implemented, specifically, to help people evacuate the Computer Science Building at Bowie State University. However, future work could include extending the application so that users can evacuate other buildings. Each floor in the Computer Science Building was drawn in SketchUp as 3D models. Each of the models were imported into Unity so that they could be integrated with their respective markers. The Vuforia library was used so that the models can be overlaid upon the markers when the markers are detected. Using Visual Studio, the application was packaged and deployed onto HoloLens.

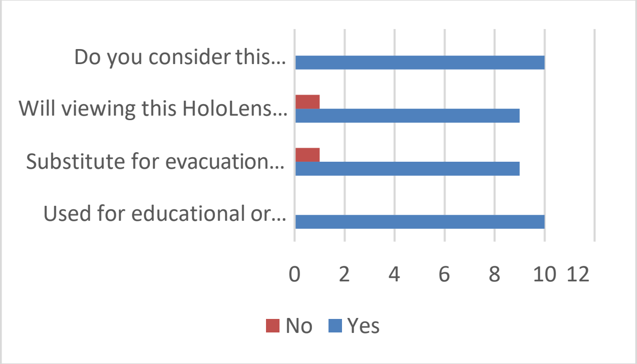


Figure 10: Evaluation of usability of HoloLens application

The application was evaluated through a limited user study involving 10 participants. The study revealed that the participants preferred to use the HoloLens to evacuate over the tablet (see Figure 8). This was likely because in order to use the application on the tablet, one would have to hold the tablet and point the camera towards the markers which could strain the user's arm. On the other hand, the HoloLens requires that the user look at the markers to see the 3D floorplans. Also, the user study revealed that the application could ideally be used to evacuate an unfamiliar building and that it could be a perfect substitute for 2D evacuation plans (see Figure 9).

Conclusively, it can be recommended that AR technologies such as HoloLens should be adopted by people for evacuating others from buildings during emergencies. They should be adopted because they are affordable and offer lifelike experiences in navigating large-scale environments.

Acknowledgments

The authors would like to acknowledge the support of the VR Laboratory at Bowie State University. This work is funded in part by the ARL Award: W911NF-17-2-0133 and Award: W911NF1820224.

References

[1] J. Ahn and R. Han, "ResueMe: An Indoor Mobile Augmented-Reality Evacuation System by Personalized Pedometry," *2011 IEEE Asia- Pacific Services Computing Conference*, Jeju Island, South Korea, pages 70-77, December 12-15, 2011.

[2] M. Alptekin and K. Temmen, "Design Concept and Prototype for an Augmented Reality based Virtual Preparation Laboratory Training in Electrical Engineering," *2018 IEEE Global Engineering Education Conference (EDUCON)*, Santa Cruz de Tenerife, Spain, pp. 963-968, April 17-20, 2018.

[3] B. Besbes, S. N. Collette, M. Tamaazousti, S. Bourgeois, and V. Gay-Bellile, "An Interactive Augmented Reality System: A Prototype for Industrial Maintenance Training Applications," *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Atlanta, GA, pp. 269-270, November 5-8, 2012.

[4] J. Birt, E. Moore, and M.A. Cowling, "Piloting Mobile Mixed Reality Simulation in Paramedic Distance Education," *2017 IEEE 5th International Conference on Serious Game and Applications for Health (SeGAH)*, Perth, Australia, April 2-4, 2017.

[5] T. Blum, V. Kleeberger, C. Bichlmeier, and N. Navab, "Miracle: An Augmented Reality Magic Mirror System for Anatomy Education," *IEEE Virtual Reality Conference 2012*, Costa Mesa, CA, pp. 115-116, March 4-8, 2012.

[6] S. K. Carpender, P. H. Campbell, B. J. Quiram, J. Frances, and J. J. Artzberger, "Urban Evacuations and Rural America: Lessons Learned from Hurricane Rita," *Public Health Reports*, 121(6):775-779, 2006.

[7] H. Chen, K. Feng, C. Mo, S. Cheng, Z. Guo, and Y. Huang, "Application of Augmented Reality in Engineering Graphics Education," *2011 IEEE International Symposium on IT in Medicine and Education*, Cuangzhou, China, pp. 362-365, December 9-11, 2011.

[8] G. Evans, J. Miller, M. I. Pena, A. MacAllister, and E. H. Winer, "Evaluating the Microsoft HoloLens through an Augmented Reality Assembly Application," *Proceedings of SPIE,* Anaheim, CA, 10197:101970V-1-101970V-16, April 11-12, 2017.

[9] J. Fruend, M. Grafe, C. Matysczok, and A. Vienenkoetter, "AR-based Training and Support of Assembly Workers in Automobile Industry," *The First IEEE International Augmented Reality Toolkit Workshop*, Darmstadt, Germany, September 19, 2002.

[10] M. Handosa, H. Schulze, D. Gračanin, M. Tucker, and M. Manue, "An Approach to Embodiment and Interactions with Digital Entities in Mixed-Reality Environment," *2018 IEEE Conference on Virtual Reality and 3D User Interfaces*, Reutlingen, Germany, pp. 569-570, March 18-22, 2018.

[11] M. G. Hanna, I. Ahmed, J. Nine, S. Prajapati, and L. Pantanowitz, "Augmented Reality Technology Using Microsoft HoloLens in Anatomic Pathology," *Archives of Pathology & Laboratory Medicine*, 142(5): 638-644, 2018.

[12] T. Haramaki and H. Nishino, "An Engineering Education Support System Using Projection-Based AR," *2016 19th International Conference on Network-Based Information Systems*, Ostrava, Czech Republic, pp. 267-272, September 7-9, 2016.

[13] A. Hasegawa, T. Ohira, M. Maeda, S. Yasumura, and K. Tanigawa, "Emergency Responses and Health Consequences after the Fukushima Accident: Evacuation and Relocation," *Clinical Oncology*, 28(4):237-244, 2016.

[14] M. Hirokane, K. Nishiwaki, and T. Endo, "Experiment of Disaster Prevention Education and Evaluation Using a Torrential Rain Simulated Experience Application," *2016 Join 8tht International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS)*, Sapporo, Japan, pages 433-437, August 25-28, 2016.

[15] Y. Huang, Y. Liu, and Y. Wang, "AR-View: An Augmented Reality Device for Digital Reconstruction of Yuangmingyuan," *2009 IEEE International Symposium on Mixed and Augmented Reality – Arts, Media and Humanities*, Orlando, FL, pp. 3-7, October 19-22, 2009.

[16] M. Kandee, P. Boonbrahm, and S. Boonbrahm, "Realistic Pulse Simulation Measurement using Haptic Device with Augmented Reality," *IECON 2016 – 42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, Italy, pp. 773-778, October 23-26, 2016.

[17] Y. S. Kim, J. O. Lee, and C. W. Park, "A Hybrid Learning System Proposal for PLC Wiring Training using AR," *2011 5th IEEE International Conference on E-Learning Industrial Electronics (ICELIE)*, Melbourne, Australia, pp. 90-94, November 7-10, 2011.

[18] E. Kučera, O. Haffner, and R. Leskovský, "Interactive and Virtual/Mixed Reality Applications for Mechatronics Education Developed in Unity Engine," *2018 Cybernetics & Informatics*, Lazy Pod Makytou, Slovakia, pp 25-30, January 31-February 3, 2018.

[19] S. N. Kundu, N. Muhammad, and F. Satter, "Using the Augmented Reality Sandbox for Advanced Learning in Geoscience Education," *2017 IEEE International Conference on Teaching, Assessment, and Learning for Education (TALE)*, Hong Kong, China, pp. 13-17, December 12-14, 2017.

[20] C. Lin, Y. Tseng, and C. Yi, "PEAR: Personal Evacuation and Rescue System," *WMuNeP '11: Proceedings of the 6th ACM Workshop on Wireless Multimedia Networking and Computing*, Miami, FL, pp. 25-30, October 31, 2011.

[21] X. Liu, H. Bai, G. Song, Y. Zhao, and J. Han, "Augmented Reality System Training for Minimally Invasive Spine Surgery," *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Macau, China, pp. 1200-1205, December 5-8, 2017.

[22] S. Okuno, T. Ito, M. Suzuki, and S. Tami, "Gait Diagnosis for Elderly People to Prevent Fall and Development of Training Assist Tool using Augmented Reality," *2017 International Symposium on Micro-NanoMechatronics and Human Science (MHS)*, Nagoya, Japan, December 3-6, 2017.

[23] S. de Ribaupierre, R. Armstrong, D. Noltie, M. Kramers, and R. Eagleson, "VR and AR Simulator for Neurosurgical Training," *2015 IEEE Virtual Reality Conference*, Arles, France, pp. 147-148, March 23-27, 2015.

[24] S. Romero, "A Brief but Frantic Struggle for Victims of Fire in Brazil," *The New York Times: Americas*, p. A4, January 29, 2013.

[25] S. Sharma and S. Jerripothula, "An Indoor Augmented Reality Mobile Application for Simulation of Building Evacuation," *Proceedings of the SPIE Conference on the Engineering Reality of Virtual Reality*, San Francisco, CA, February 9-10, 2015.

[26] A. Srivastava, "Enriching Student Learning Experience using Augmented Reality and Smart Learning Objects," *ICMI 2016: Proceedings of the 18th ACM International Conference on Multimodal Interaction*, Tokyo, Japan, pp. 572-576, November 12- 16, 2016.

[27] J. Stigall and S. Sharma, "Mobile Augmented Reality Application for Building Evacuation Using Intelligent Signs," *Proceedings of the ISCA 26th International Conference on Software Engineering and Data Engineering (SEDE 2017)*, San Diego, CA, pages 19-24, October 2-4, 2017.

[28] E. C. E. Vidal, J. F. Ty, N. R. Caluya, an M. M. T. Rodrigo, "MAGIS: Mobile Augmented-Reality Games for Instructional Support," VRCAI '16: Proceedings of the 3rd Asia-Europe Symposium on Simulation and Serious Gaming, Zhuhai, China, pp. 191-194, December 3-4, 2016.

[29] T. Vold, H. Haave, O. J. S. Ranglund, G. O. Venemyr, B. T. Bakken, L. Kiønig, and R. Braun, Flipped Gaming – Testing Three Simulation Game," *2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET)*, April 26-28, 2018.

[30] M. Wichrowski, "Teaching Augmented Reality in Practice: Tools, Workshops and Students' Projects," *MIDI '13: Proceedings of the International Conference on Multimedia, Interaction, Design, and Innovation*, Warsaw, Poland, June 24-25, 2013.

[31] O. Yaman and M. Karakose, "Development of Image Processing Based Methods Using Augmented Reality in Higher Education," *2016 15th International Conference on Information Technology Based Higher Education and Training (ITHET)*, Istanbul, Turkey, September 8-10, 2016.

[32] L. Yangguang, L. Yue, and W. Xiadong, Multiplayer Collaborative Training System based on Mobile AR Innovative Interaction Technology," *2014 International Conference on Virtual Reality and Visualization*, Shenyang, China, pp. 81-85, October 1, 2015.

[33] S. Zareen and S. Ghulamani, "Educating Students in Remote Areas Using Augmented Reality," *2018 International Conference on Computing, Mathematics and Engineering Technologies – iCoMET 2018*, Sukkur, Pakistan, March 3-4, 2018.

James Stigall is currently a doctoral candidate majoring in Computer Science at Bowie State University. Previously, he earned a Master of Science degree in computer science from Virginia State University; before that, he graduated Magna Cum Laude from Saint Paul's College with a Bachelor of Science degree in computer science with a minor in mathematics. His work has been presented in research conferences such as the 2015 International Conference on Computational Science and Computational Intelligence (CSCI '15) and the ISCA's 26th and 27th International Conferences on Software Engineering and Data Engineering (SEDE '17 and SEDE '18). He is currently working under the direction of Dr. Sharad Sharma researching the uses of virtual reality (VR) and augmented reality (AR) in instruction and training.

Sharad Sharma is an Associate Professor in the Department of Computer Science, Bowie State University, Bowie, MD 20715 USA. He has received a Ph.D. in Computer Engineering from Wayne State University, Detroit, MI, USA in 2006 and M.S. from University of Michigan, Ann Arbor, MI, USA in 2003. Dr. Sharma is the Director of the Virtual Reality Laboratory at the Bowie State University. The VR laboratory applies virtual reality and augmented reality as a tool for learning, training, and education. His research focus is on modeling and simulation of multi-agent systems (MAS) and multi-user virtual reality (MUVR) environments for emergency response and decision-making strategies. His work is motivated by the need of research in real-time agent navigation for reaching a goal in emergency situations like evacuation. His proposed human behavior system integrates both artificial intelligence and fuzzy logic parameters. He is also exploring socio-cultural issues in Collaborative Virtual Environments (CVE) for emergency response and decision making in dense urban environments.

Sri Teja Bodempudi is a doctoral student in the Department of Computer Science at Bowie State University. He is working as a graduate research assistant in the Virtual Reality Laboratory. His research focus is on the use of AR and VR for training and education. His work is motivated by the need of research in emergency evacuation for public safety. His research interests include software engineering, data mining, data visualization, VR and AR.

David R. Scribner (Photo not available) has worked at the U.S. Army Research Laboratory's Human Research and Engineering Directorate (HRED) for over 24 years and recently earned his Ph.D. in 2013. He has worked in the Weapons Branch at Redstone Arsenal, Alabama and the Dismounted Warrior Branch, both within the Soldier Performance Division. He is currently working within the Complex Ground Systems and Operations Branch of the Soldier Battlefield Integration Division. His research was primarily focused on cognitive workload and multi-tasking in several domains including teleoperated and robotic systems, and marksmanship performance.

Jock O. Grynovicki (photo not available) is the chief of the Complex Ground Systems & Operations Branch of the Human Research Engineering Directorate of the Army Research Laboratory (ARL). His background includes numerous field and laboratory studies examining battle command performance and metrics, metrics for command and control systems, virtual systems, and managing Human System Integration programs for the U.S. Army. He has a master's degree in public health policy and doctorate degree in statistics. He is currently the ARL lead research and application coordinator for Soldier Modeling and Simulation Tools Development, Soldier Centered Analysis for the Future, and Social Cultural Analysis at ARL HRED.

Peter J. Grazaitis has worked at the U.S. Army Research Laboratory's Human Research and Engineering Directorate (HRED) for over 30 years and has worked to advance Army logistics and field logistics planning systems for much of his career. He has assisted with fielding various logistics programs for ammunition, petroleum, water, and other logistical commodities for the Combined Arms Support Command. Mr. Grazaitis has also recently received the prestigious Tibbets Award for his numerous Small Business Innovative Research (SBIR) work in promoting innovative technologies. He is now the manager of the human dimension area of the Dense Urban Environment program at HRED

.

Simplifying Data Visualization Pipelines with the NRDC-CHORDS Interface

Pattaphol Jirasessakul*, Zachary Waller*, Paul Marquis*, Connor Scully-Allison*,
Vinh Le*, Scotty Strachan*, Frederick C. Harris, Jr.*, and Sergiu M. Dascalu*
University of Nevada, Reno, Nevada, 89557, USA

Abstract

In the physical sciences, the observation and analysis of environmental readings, such as wind speed, sap flow, atmospheric pressure, temperature, and precipitation, benefit greatly from real-time visualization as they allow environmental scientists to create faster actionable intelligence. However, the scarcity of easily accessible and customizable real-time visualization software often creates logistical problems for researchers focused in environmental sciences. The goal of this paper is to present an alternative approach for the Nevada Research Data Center (NRDC) to visualize environmental data in near-real time and confirm its viability for usage with other research projects of similar size. This approach involves creating multiple iterations of open-source near-real time interface to act as middle-ware between the NRDC's data repository and CHORDS, a cloud-hosted data visualization package. We evaluate the success of our implementation by comparing metrics of use, determining that both iterations of our software were much faster and easier to use than CHORDS built-in configuration interfaces.

Key Words: Data visualization, environmental science, middleware, web scraping, web service.

1 Introduction

Data visualization is a critical tool for scientists working with large and constantly updating data streams. However, these scientists are often presented with two options for robust visualization: expensive proprietary solutions, or programming language libraries that require developer-level knowledge to use.

This is where Cloud Hosted Real-Time Data Services for the geosciences, or CHORDS, comes in. CHORDS is a project that was developed by EarthCube, an NSF-funded project that supports the development of cyberinfrastructure for the geosciences. CHORDS provide a visualization platform tailored for environmental research to make real-time data available to the research community in standard formats [1].

However, CHORDS by itself comes with some limitations that makes the use of a middleware vital for operation in larger projects, such as the ones hosted at the NRDC. CHORDS provide an HTTP API that allows for the population of real-time data entry, but the API is only partially exposed and often forces configuration and setup onto the user. This is especially tedious for larger environmental research groups because the addition of new sensor equipment is not an uncommon occurrence during a multi-year operating period.

In this paper we describe the iterative software engineering process we undertook to develop and refine this middleware between the NRDC and CHORDS. Our first iteration attempted to approach the problem of missing APIs by using web scraping and automation tools to emulate the manual setup of new CHORDS instances and visualizations. This approach allowed us to automate tedious metadata inputs by drawing necessary metadata data from NRDC databases and inserting it into form fields. Out second iteration improved on this by further mitigating the need for users to select desired data streams from the NRDC, and instead provides for an *en-masse* download and configuration. Preliminary results of this software indicate that this middleware improves on speed and usability compared to the CHORDS built-in configuration UI.

The remainder of this paper is structured as follows: Section 2 introduces a basic background of the project and some related works, Section 3 goes into the specifications of the software, Section 4 discusses the overall design of the software, Section 5 contains details on the UI design of the web client, Section 6 and 7 includes details of the prototype development, Section 8 details validation metrics and, finally Section 9 shows conclusions and future work.

2 Background & Related Work

This project was made in coordination with the NRDC and EarthCube, both of which fall under the umbrella of the Cyberinfrastructure research as defined by the National Science Foundation (NSF). The following section will go over the goals of both organizations as well as a more detailed explanation on other visualization options inside and outside of the Nexus Project.

---

* Department of Computer Science and Engineering. E-mails: (pjirasessakul, cscully-allison)@nevada.unr.edu, (zacharydwaller, paul.marquis1, vdacle)@gmail.com, scotty@dayhike.net, (fred.harris, dascalus)@cse.unr.edu.

2.1 NRDC

The NRDC was born out of a data portal that was developed during a previous Track 1 NSF EPSCoR project on climate change called the Nevada Climate Change Portal. The current project that the NRDC is affiliated with, the Solar Energy-Water-Environment Nexus, was created in order to increase research awareness, and productivity of alternative energy sources, and the conservation of natural resources in the state of Nevada. The NRDC serves in a critical role of cyberinfrastructure within the Nexus Project, which includes the provision of technical skills and resources to members of the research project. The tasks that the NRDC covers include the acquisition, transport, storage, querying, and dissemination of observational data gathered by automated digital sensor systems. The NRDC participates in cutting-edge software and systems development to enhance next-generation science that leverages the Internet of Things (IoT). Their goal is to transform the scale, quality, impact, and bottom-line cost of research projects in Nevada that seek to deploy automated sensor systems as part of their scientific workflow [5].

2.2 EarthCube

EarthCube is a quickly growing community of scientists across all geoscience domains, including geoinformatics researchers and data scientists. They are a joint effort between the NSF Directorate for Geosciences and the Division of Advanced Cyberinfrastructure. EarthCube was initiated by the NSF in 2011 to transform geoscience research by developing cyberinfrastructure to improve access, sharing, visualization, and analysis of all forms of geoscience data and related resources. As a community-governed effort, EarthCube's goal is to enable geoscientists to tackle the challenges of understanding and predicting a complex and evolving solid earth, hydrosphere, atmosphere, and space environment systems. The NSF's Directorate for Geosciences (GEO) and the Division of Advanced Cyberinfrastructure (ACI) partnered to sponsor EarthCube, which NSF anticipates supporting through 2022 [3].

2.3 CHORDS & Other Visualization Options

Currently, there are a handful of projects within the NRDC that utilizes data visualization: VISTED and VFire. VISTED (Visualization Tool of Environmental Data) is a web application, which enables data selection, extraction, download, conversion, and visualization of environmental data sets that extends for over 30 years (1980 - 2009) [8]. VFire is an immersive visualization application that uses remote sensing data in conjunction with a simulation model to predict the behavior of wildfires [4]. RWWSS (Real-time Web-based Wildfire Simulation System) is a web application that provides users with wildfire simulations using data from the Lehman Creek Watershed in Great Basin National Park [13]. A workflow dedicated to visualizing big data on web applications was created as an alternative to expensive third-party software [14]. Finally, a system revolving around MongoDB and some accompanying tools were developed to visualize big data as a way to address the mass influx of data in the recent years [12].

Aside from CHORDS, there are also alternatives out for near real-time data visualization. There are multiple programming languages out there with data visualization library along with existing proprietary data visualization software. Libraries such as D3.js and Plotly.js, are well known libraries within the field of data visualization. D3.js is a library made for visualizing data using web standards. It combines powerful visualization and interaction techniques with data driven approach to give users the freedom to design the visual interface anyway they want. Plotly.js is an open-source library that supports many chart types including scientific ones such as heatmaps and contour plots to use for plotting sensor data in real-time [2, 7]. Unfortunately, both of these and other libraries suffer from the same problem as they require some sort of programming knowledge on their respective programming languages. This results in lower accessibility of these libraries for smaller research teams as they may not have someone with the programming knowledge in the team or have the time and patience to learn the language and library by themselves.

Alternatives to using programming libraries would be to use real-time data visualization software. Examples of these software are Tableau and Visualr, both offer many features such as the ability to connect with multiple data sources such as MySQL, Oracle and MS Excel, being able to fetch data from API Data Providers and a plug and play feature where all the users have to do get the software running is to install it [10, 11]. However, as compensation for having many features, they often come with a hefty price tag along with some sort of training session in order to use them effectively.

3 Software Specification

The main requirements of this project were elicited from multiple stakeholders using a formal interview process. The answers we received from these interviews went on to inform the project requirements. These requirements were split between functional requirements, which describe the overall technical functionality of the system, and nonfunctional requirements, which outline constraints on the system.

3.1 Elicitation Interviews

In order to best ensure that the functional requirements composed by developers met the expectations of project stakeholders, several interviews were undertaken to understand their needs and desires *vis a vis* this software. The interviewees were selected for their knowledge of the CHORDS software platform and for their technical understanding of the NRDC. Specifically, we interviewed Scotty Strachan (Environmental Scientist/ UNR Director of Cyber-infrastructure), Vinh Le (Software Developer for the

NRDC), and Zachary Waller (Developer for this Project).

From questions asked of Zachary Waller, the general theme of his needs and requests were oriented around technological specifications and limitations on what software should or should not be used. By contrast, Vinh Le's answers to questions were focused much more strongly about architectural suggestions. His input was crucial in understanding how data should be retrieved from the NRDC and manipulated by the program. Finally, Scotty Strachan provided the perspective of a highly technologically literate user. Additionally, he was the only stakeholder already familiar with the CHORDS platform. From him, we gleaned significant detail about the use cases of the software we were building in addition to understandings of how it should interact with CHORDS.

3.2 Functional Requirements

From those interviews, detailed in Section 3.1, we elicited seven base level functional requirements that define the operations of our solution. The first and second requirements are to create an interface that will not only be able to successfully communicate with a running instance of CHORDS and manage the data inside, but also, to talk to the research team's data source in the NRDC in order to query data. The third requirement involves implementing the functionality to visualize data displayed on CHORDS in near-real time. The fourth requirement is streamlining the user experience by creating a web client to simplify the originally tedious visualization process for users. The NRDC sensor networks currently exist in a structured hierarchy, so the fifth requirement is to fetch that hierarchy and format it into integrates intuitively with the user interface. Finally, the sixth and seventh requirements are the functionality that allows users to specify whether they wish to stream data in a near-real time mode, or stream from a specific date range.

There are four higher level functionalities that this solution provides outside of the scope of the main functionalities. The first requirement let users compute and display summary statistics, such as the minimum, maximum, mean and standard deviation, of the data streams that they chose for their visualized session. The second requirement enables users to share their visualized session by adding in the functionality to export a snapshot, which is an interactable instance of the user's visualization at the time it was created. The third requirement enables users the option of having the visualized instance alert them through email when the data leaves an expected range. The fourth requirement builds upon the web client by embedding a customized Google Maps API onto it. The map will list all available sites in NRDCs hierarchy network that represented a marker on the map. Additionally, when the user clicks on a marker, the latest photo streamed from that site will be displayed along with specific information about that site, such as name, latitude, longitude, and current measurements.

3.3 Non-Functional Requirements

This project operates under four non-functional requirements acting as constraints on the design and development of this system. The first requires that the interface portion of the software be written in C# with the .NET Web API library as its framework. The second requirement is that the development team uses a modified instance of the CHORDS visualization package for the visualization aspect. The third requirement is that the software should be compatible with all major web browsers. The fourth requirement is that the software should be able to consistently maintain near-real time execution when streaming data from sensor instruments.

4 Software Design

This project consists of three major parts: NC-Client, NC-Interface and the Chord's visualization. The goal of this section is to show, in detail, the design of this software and everything that is required to produce a similar design. Beginning with a high-level explanation of each component and delving into how each of these components interact with each other. The architecture of this project is component based to ensure fast and robust development as well as strong interoperability as each component is loosely coupled. A high-level design of the project can be seen below in Figure 1.

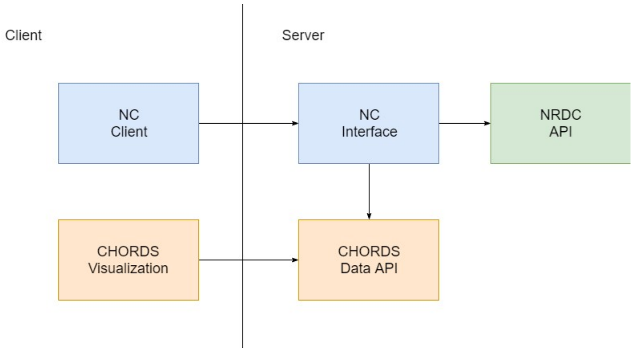


Figure 1: A diagram showing the high-level design of the project

4.1 Components

NC-Client - The NC-Client consists of two web pages and a scripts file. The script contains code to drive the spawning of views and navigation logic for each web page on the web page. It also implements an auto-refreshing function call to allow for near-real time streaming.

NC-Interface - The NC Interface is the main component of our project. It contains four modules: ChordsBot, DataCenter, GrafanaManager, and SessionManager. Three of these four modules also have a Web API controller associated with them. Each of these modules are further explained in Section 6.

Chords Visualization - The Chords Visualization is the main component of visually and actively interfacing with the data from the NRDC repositories. After a user has made a selection of a desired datastream using the NC-Client, the NC-Interface will fetch that data, reformat it, create the users CHORDS session (via SessionManager) and then push the data to the newly created session with ChordsBot. Figure 2 shows an example of a visualized CHORDS session.

5 UI Design

There are two primary interface users that interact with when using the *Generalized Software Interface for CHORDS*: The interface web client and the CHORDSs interface. Although functionality was implemented to integrate with Graphana, we cannot include it here due to space limitations.

5.1 Interface Web Client

Our custom-built web client, visible in Figure 3, is a single-page-web-application allowing users to view the sensor network hierarchy, select a deployment, and begin streaming data. This feature enables users to specify the type of data they want to stream.

The NRDC sensor networks exist in a hierarchy. Each sensor network (NevCAN, Solar Nexus, Walker Basin Hydro) contains a list of sites, which refer to geographic locations. Each site contains a list of systems, which are logical groupings of deployments or sensors. In order to make it easy for a user to access specific deployments to view their data on the user interface, we implemented a way to retrieve and display this entire hierarchy on said interface. This requires our web interface to make calls to the NRDCs. Infrastructure API and format the data returned in a user-readable format. By implementing this feature, it makes accessing specific data streams much easier for the user.

Upon visiting the client page, the user can select between the three sensor networks. Then, the user can select which site they want to see data for from a list of all sites in that sensor network. Next, the users select which data streams they would like visualized. The user can select one stream or multiple. Finally, the user can save their session with a name and specify the time period for which they would like the data streamed. Leaving the end date of the stream blank will result in a continuous live data stream.

5.2 CHORDS Interface

CHORDSs UI primarily functions on the back end of our software by creating new CHORDS instruments for user created data streams. At the top of the CHORDS page, the name the user chose for the session in our web client is displayed as the name of the CHORDS instrument, along with the total number of measurements reported and include above Figure 3 the dates of those measurements. Additionally, a list of all visualized sessions created by a research team is available to them as well as seen in Figure 4.

The main section of the visualization page displays the actual graphed data from the data stream displayed alongside of the names of the data streams and above the times that the data was received by CHORDS like the one seen in Figure 2. Below, each variable corresponding to a selected data stream is displayed. For each variable, the user is shown the units of the variable, the property measured, and the name of the variable, which is a combination of data about the stream including the location of the sensors, what the sensors are measuring, and other information.

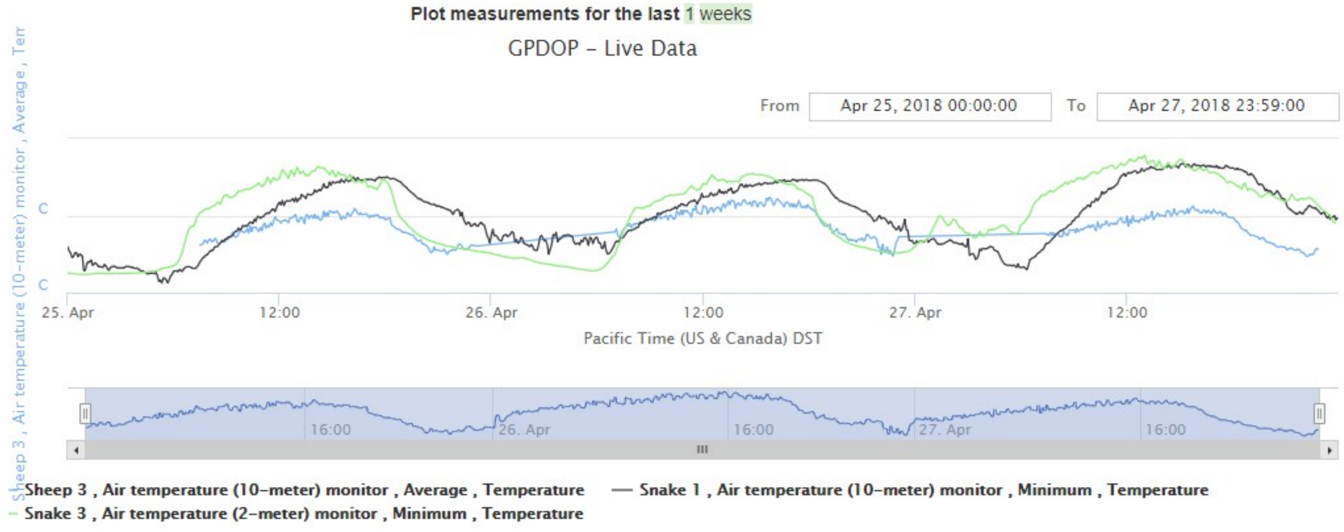


Figure 2: An example of a visualized session of 3 different data streams on a CHORDS instance

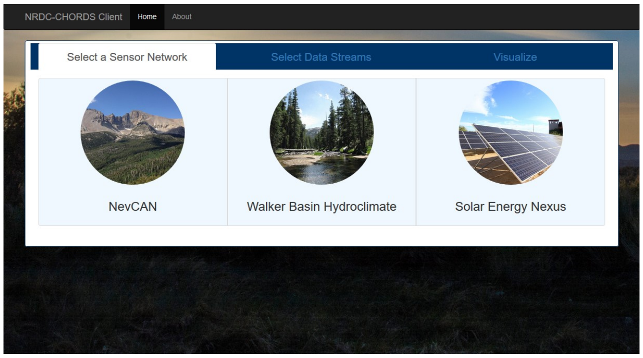


Figure 3: The main page for the NRDC-CHORDS interface web client. A user can begin finding a datastream to visualize by clicking on one of the three available site networks associated with the NRDC.

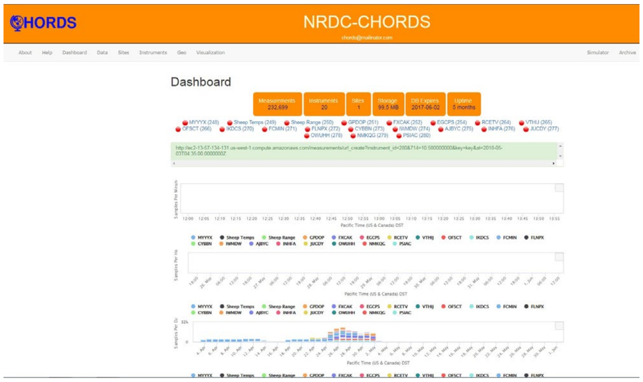


Figure 4: A picture of a research team's CHORDS's portal. Currently displayed is a list of the team's currently visualized session

6 Prototype Development

A prototype of this project was developed as a proof of concept for the NRDC. It implemented the majority of the functionality detailed in Section 3. On the server side, we created a service called NC Interface. It acts as an interface between the NRDC's data center and CHORD's Data API so that data can be gathered from the former, formatted and sent to the latter. On the client side, we created a single page web application called NC Client that allow users to select the specific data stream they want to visualize out of the NRDC's sensor network hierarchy.

The NC Interface was developed using the .NET Web API framework. It is composed of four different modules: ChordsBot, DataCenter, GrafanaManager and SessionManager. When the user first opened up the web client, the DataCenter module is called to fetch the NRDC's sensor hierarchy network for the user to select their data streams. Once the user selects their stream(s), that information is sent to "Session Manager" that will create the user's session on the research team's CHORDS Portal. Once the session is created on CHORDS, ChordsBot is called to automate the data streaming to the session along with performing other functionalities like filling forms in the session with information regarding the selected data streams (their name, location, units of measurements, etc.) and generating the session's Grafana dashboard. Automated tests are performed in order to confirm that DataCenter was getting the correct data and that ChordsBot was performing the functionalities that it was automating for the user on CHORDS properly.

CHORDS's Data API was very inflexible in terms of what our development team wanted from it. While certain things like data put and fetch activities are well documented on the API, functionalities like the automation of sites and instrument creation on a particular CHORDS instance are not. From our communication with the API's developer, we have learned that since the API was developed using Ruby on Rails, a lot of its functionality are written for them which doesn't give their development team a lot of room to formalize the API.

To explain our workaround in detail, we utilized the web automation software packages of Selenium and PhantomJS to simulate user interaction with the CHORDS UI [6, 9]. More specifically, after a user successfully selected a data stream, they wanted to visualize from the NRDC-CHORDS interface, our ChordsBot would parse and interact with the various CHORDS web pages required to set up a data visualization on the fly with all necessary metadata about the location, datatypes, variables, etc. required.

The use of this automation software helped us surpass what seemed like an insurmountable requirement given the lack of API support on the CHORDS side. However, it also changed our initial designs somewhat substantially by turning a generic interface into a UI driven web crawler. Additionally, this approach introduced some unique problems which limited the casual use of this software. Most notably, when a data stream was selected from our interface it spawned an entirely new CHORDS session each time. This, in itself, was not an insurmountable problem as we added a module for saving and returning to old sessions, however it seemed to go further against the original design intention of this interface as our own UI was becoming increasingly complicated to act merely as a wrapper for the CHORDS UI. Accordingly, we decided to reexamine our design and make a substantial iteration on our existing work.

7 Prototype Iteration

In the spirit of upholding agile design principles and practices, the initial prototype, detailed in Section 6 was thoroughly examined to evaluate how well the software met the needs and goals of project stakeholders. Accordingly, as part of our evaluation we consulted with multiple project stakeholders and asked them how well it fit their vision. This evaluation produced significant meaningful feedback which provided sufficient cause for us to reconfigure the NRDC-

CHORDS interface in many ways to better fit the express needs and desires of stakeholders.

7.1 Stakeholder Feedback

In order to best determine what adjustments should be made for an iteration on the NRDC-CHORDS interface software, we informally interviewed stakeholders and asked them to play with software. As they interacted with the software package we asked them to speak their minds and express what they like or did not like about the first iteration. We performed this exercise with three individuals who are key to NRDC and the development of this project: Connor Scully-Allison (NRDC System Administrator), Vinh Le and Scotty Strachan.

In these informal sessions we noted several similar points of feedback from the above stakeholders. First, though all stakeholders acknowledged the API limitations we were working with, they unanimously indicated that our UI/web crawling-based solution did not fully meet their earlier expectations of what the software would be. More ideally, they wanted a continuously running CHORDS instance, preconfigured with NRDC data streams that passively accepted new data as it was dropped into the NRDC database.

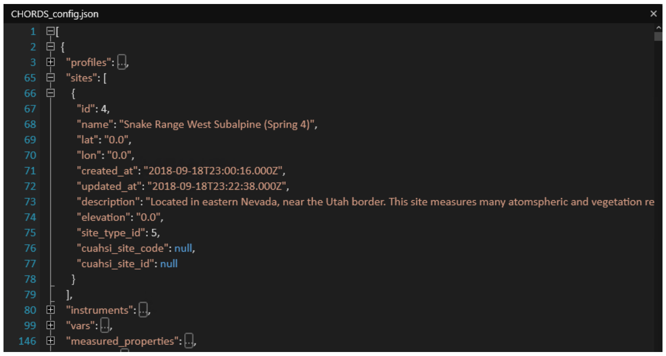
From there, these stakeholders generally indicated that they did not see much of a need for the UI we have built if the CHORDS service could just be preconfigured. Especially since the UI seemed to complicate the adding of many data streams, on different sensors, and at different sites. Although it reduced the tedious typing required of the CHORDS UI, it still seemed to them that there should be a better workaround.

7.2 NRDC-CHORDS Redesign

In exploring options to better meet the expectations of our potential users and stakeholders we redoubled our efforts to find a means to better automate CHORDS configuration. After some searching, we found a previously overlooked function on the CHORDS primary configuration page: a "download configuration" link and an "upload configuration" link. These links enabled us to download a standard-format JSON configuration file to our local machine, modify it and send it back up to the CHORDS instance, whereupon the instance would reload and reflect all the configurations put into the configuration file. To illustrate, an excerpt of this file can be seen in Figure 5.

From this revelation we were able to alter the current architecture and design of the NRDC-CHORDS interface towards a more streamlined and automated package with minimal human interaction required. However, to implement this redesign we needed to first break down the existing code from the prior prototype and evaluate what we could reconfigure for use in this iteration.

We ended up with a short list of re-usable modules after examining our code. We found that the two "managers" were not especially important if we could maintain a static and constant pre-configured instance of instance of CHORDS running, so we scrapped those. From there we determined that we no longer required the web crawling component of ChordsBot, so we excluded that codebase as well. In the end we ended up re-using large parts of ChordsBot's backend functionality. Specifically, we adapted those parts that managed the near-real time data upload to CHORDS instance in addition to those that managed the mapping between NRDC's schema and CHORDS. We were also able to reconfigure part of the DataCenter module which handled communication with the NRDC's data and metadata APIs.



```
[
{
 "profiles": ...,
 "sites": [
  {
   "id": 4,
   "name": "Snake Range West Subalpine (Spring 4)",
   "lat": "0.0",
   "lon": "0.0",
   "created_at": "2018-09-18T23:00:16.000Z",
   "updated_at": "2018-09-18T23:22:38.000Z",
   "description": "Located in eastern Nevada, near the Utah border. This site measures many atomspheric and vegetation re
   "elevation": "0.0",
   "site_type_id": 5,
   "cuahsi_site_code": null,
   "cuahsi_site_id": null
  }
 ],
 "instruments": ...,
 "vars": ...,
 "measured_properties": ...,
```

Figure 5: An example of the CHORDS configuration file. The sites, instruments, vars and other fields in this JSON file can be manually edited (or edited by a program) and re-uploaded to CHORDS, just so long as they contain the same data fields which CHORDS expects

From this redesign the new high-level architecture shown in Figure 6 was produced. In this figure you will note that the high-level architecture has undergone significant change from that detailed in Figure 1. Now it is no longer broken up as a basic client server pattern but corresponds more roughly to a pipeline that runs on a server chosen to host the "Automated CHORDS Client". Now this new architecture is not without its own need for manual interaction from users, however in choosing to utilize this configuration file as the main means of configuration, the need for direct user interaction is significantly lessened.

7.3 New NRDC-CHORDS Workflow

The new workflow of this application begins with the user manually downloading the default configuration provided with a new CHORDS instance. This configuration file provides a template informing our configuration service how certain metadata items should be formatted. For example, as seen in Figure 5, a "site" will always have a *name* field, an *id*, a *lat*, a *lon*, etc. The configuration document output by our configuration program has to conform to these fields to be properly accepted by CHORDS.

After our "template" configuration file has been successfully downloaded, and placed in a specified folder, the user can run the configuration program, signified by the three green-colored submodules. The configuration program first

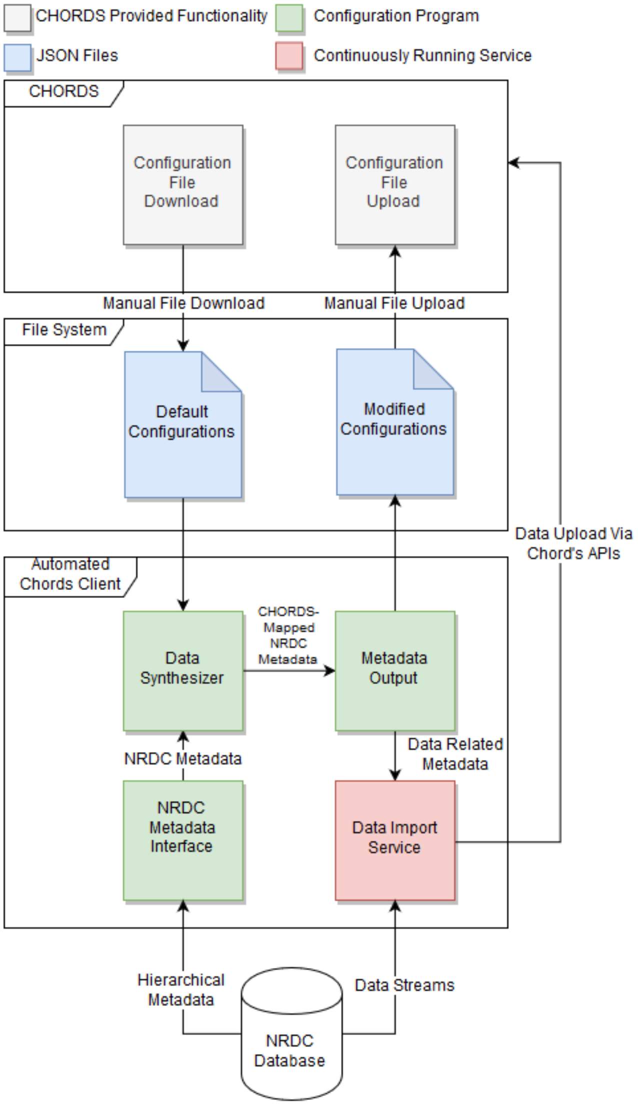


Figure 6: The refined and re-configured architecture of the NRDC-CHORDS interface application. This architecture improves on prior designs by eliminating configuration UIs and utilizing existing configuration files to lessen user interaction in setting up visualization streams.

takes in the JSON configuration file and holds it in memory as a dictionary. From there it calls the NRDC Metadata Interface submodule and retrieves relevant metadata from a particular database in the NRDC. For the purposes of this prototype, the specified database was a database for one project of limited size: NevCAN.

From the database it recovers the full sensor network hierarchy which maps data streams to specific sites and sensors, explained in Section 5.1 of this paper. After recovering the full sensor network hierarchy we use ChordsBot's reconfigured schema mapping functionality to map this hierarchy to the CHORDs expected schema structure. In our case, the NRDC "sites" maps well to CHORD's "sites", however our "deployments" are combined with our "components" to map to CHORDS "instruments". Essentially the NRDC has most of the metadata required to set up these sites and instruments, however the data must be extracted from various different tables in the NRDC database and meaningfully mapped to specific fields in the CHORDS configuration file. This mapping is the primary responsibility of the Data Synthesizer module.

For each "site" and "deployment" retrieved from the NRDC database, a new, synthesized "site" and "instrument" object is pushed into their respective arrays. These arrays are then loaded into the CHORDS configuration object we read earlier. This object is then passed along to the metadata output service which is in charge of writing out a properly formatted output file containing the modified configuration information. This file must be manually uploaded back to the CHORDS instance and will cause a full reset of all exiting configuration and data contained in the instance. Accordingly, this configuration and setup should only be done when first starting a new CHORDS instance or when major changes were made to the number of sensors or sites have occurred since the last configuration.

The final module in this new architecture can be seen in the lower left-hand corner colored red. This module is an active service which queries a specific set of data streams, on a timed loop, from the configured data base and pushes them up to the pre-configured CHORDS visualizations as they come in from various sites around Nevada. This service works in exactly the same way as in the prior iteration of our software, using the CHORDS-provided GET calls to pass up data into the CHRODS instance.

8 Validation

The initial iteration of our software was successful in significantly reducing the time and keystrokes required to set up a short-term CHORDS instance. Although not formally evaluated with a user study, we made rough estimations from our own experiences with this software on the time saved to set up a visualization with our UI based interface.

To set up a single data stream, instrument and site from scratch takes on the order of minutes (approx. 2-5 depending on one's familiarity with the necessary metadata fields). With our interface we significantly reduce that setup time by allowing the user to just click through the NRDC hierarchy and select their desired data streams. The same process of setting up a single data stream, instrument and site with the UI could be easily done in under 30 seconds if one is acquainted with the interface. Additionally, no manual typing is required when using this method.

With the second iteration of this software package, we improve on these metrics of usability and speed even further by reducing user interaction to the mere download of a file, the running of a command line program and the upload of an output file. In addition to this simplified workflow from a user's perspective this new architecture allows for establishing a long running, readily accessible CHORDS instance which has data continuously streaming to it over a long period of time. The prototype developed for the second iteration of the NRDC-CHORDS interface was successfully running for the month of October 2018, ingested over 100,000+ data points, and readily visualized all uploaded data in real time when the site hosting the instance was visited.

While many data visualization solutions on the market support live streaming of data, most come at a high price that limits their availability to those outside of industrial applications. Our service is open source and could be adapted to work with systems other than the NRDC database, which could allow for greater availability of live-streaming data visualization.

9 Conclusion and Future Work

In this paper we detailed the iterative software process undertaken to build and refine a generalizable software interface connecting a data repository like the NRDC to the CHORDS visualization service. We additionally validated our software interface by comparing it against the usability and speed of CHORDS built-in configuration interface.

Although significant future work is planned for this software solution, the most obvious addresses the need for enhanced configurability with the data and metadata sources. For the purposes of this prototype much of the NRDC schema and connection information was hard-coded into our prototype. In order to enhance the generalizability of this software package, a clearly defined configuration file or module should be developed to enable other users to define their own schemas and data sources.

With some modification this software could be used as an open-source data visualization solution for labs that cannot afford more expensive software. This software can also help those who are not knowledgeable enough at programming to interface their database to CHORDS. Additionally, there are plans for the software to be used by the Desert Research Institute to monitor data incoming from lysimeters. This software has great potential to help many people visualize their data.

Acknowledgements

This material is based upon work supported by the National Science Foundation under grant number IIA1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. D. Daniels, S. J. Graves, B. Kerkez, V. Chandrasekar, F. Vernon, C. L. Martin, M. Maskey, K. Keiser, and M. J. Dye, "Connecting Real-Time Data to Algorithms and Databases: Earthcube's Cloudhosted Real-Time Data Services for the Geosciences (Chords)," *AGU Fall Meeting Abstracts*, 2015.

[2] D3.js, https://d3js.org/\#introduction, [Online; accessed February 22, 2019].

[3] Earthcube, https://earthcube.org/info/ about, [Online; accessed February 22, 2019].

[4] Roger V. Hoang, Matthew R. Sgambati, Timothy J. Brown, Daniel S. Coming, and Frederick C. Harris Jr., "Vfire: Immersive Wildfire Simulation and Visualization," *Computers & Graphics*, 34(6):655–664, 2010.

[5] Nevada Research Data Center, www.sensor, nevada.edu, [Online; accessed February 22, 2019].

[6] PhantomJS, "PhantomJS - Scriptable Headless Browser," *PhantomJS - Scriptable Headless Browser*, [Online], Available: http://phantomjs.org/, [Online; accessed February 26, 2019].

[7] Plot.ly, https://plot.ly/javascript/, [Online; accessed February 22, 2019].

[8] Likhitha Ravi, S. Dascalu, Frederick C. Harris, John Mejia, and Noureddine Belkhatir, "Visted: A Visualization Toolset for Environmental Data," *Proceedings of the 2015 International Conference on Computers and their Application (CATA2015)*, pp. 335–342, 2015.

[9] Selenium, https://www.seleniumhq.org/ about/, [Online; accessed February 26, 2019].

[10] Tableau, https://www.tableau.com/, 2018-1-features, [Online; accessed February 22, 2019].

[11] Visualr, https://visualrsoftware.com/ features.html, [Online; accessed February 22, 2019].

[12] Rui Wu, *Environment for Large Data Processing and Visualization Using MongoDB*, University of Nevada, Reno, 2015.

[13] Rui Wu, Chao Chen, Sajjad Ahmad, John M Volk, Cristina Luca, Frederick C. Harris, and Sergiu M. Dascalu, "A Real-Time Web-Based Wildfire Simulation System," *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society,* IEEE, pp. 4964–4969, 2016.

[14] Rui Wu, Jose T. Painumkal, Nimrat Randhawa, Lisa Palathingal, Sage R. Hiibel, Sergiu M. Dascalu, and Frederick C. Harris, "A New Workflow to Interact with and Visualize Big Data for Web Applications," *2016 LInternational Conference on Collaboration Technologies and Systems (CTS)*, IEEE*,* pp. 302-309, 2016.

Pattaphol Jirasessakul graduated from the University of Nevada Reno in 2018. He is experienced with Python, C++, and R with interests in big data, statistics, data mining, and web design. He has one conference publication and is currently working as a Freelance Software Engineer.

Zachary Waller, born and raised in Reno, is experienced with C++ as well as C# and the .NET framework. He graduated from the University of Nevada, Reno in 2018. He has a strong interest in web services and Human-Computer Interaction. He has one conference publication and is employed as a software developer at Conformance Technologies in Reno, Nevada.

Paul Marquis is a Las Vegas native with experience in C++, python and an interest in big data and data streaming technologies. He graduated the University of Nevada Reno in December of 2018 and currently has one conference publication. He is currently employed at Hamilton Robotics as a software engineer.

Connor Scully-Allison received his B.A. in Philosophy in 2012 from the University of Nevada, Reno (UNR). Accepted into the master's program at UNR for Computer Science and Engineering in 2015, he is currently working as a research assistant on the Track 1 Nexus Project for the Cyber-Infrastructure lab located in the College of Engineering. His research interests include Human Computer Interaction, High Performance Computing, and Software Engineering. He has published 7 Conference papers and 3 Journal Papers. As of February 2019, Connor holds a position as a student fellow for the Earth Science Information Partners (ESIP) Organization.

Vinh Le graduated from the University, Reno with a B.S in Computer Science and Engineering in 2015 and a MS in August of 2018. Vinh is a Graduate Research Assistant affiliated with the Cyber Infrastructure Lab at the University of Nevada, Reno. His research interests consist primarily of Software Engineering, Internet Architecture, and Human-Computer Interaction. He is currently employed with the Reno-based software startup Inlumon.

Scotty Strachan is the Director of Cyberinfrastructure in the Office of Information Technology at the University of Nevada, Reno. Strachan graduated from the University of Nevada, Reno in 2001 with a bachelor's degree in geography and minor in economics. After spending some additional years as a geotechnical consultant and project manager, he returned to the University and completed a M.S. and Ph.D., both in geography, along with a graduate minor in business administration. Strachan's primary research interests lie in mountain ecosystems and observational networks, and he relies heavily on the integration of information technologies with research to accomplish his goals of producing useful, long-term science.

Frederick C. Harris Jr. received his BS and MS degrees in Mathematics and Educational Administration from Bob Jones University, Greenville, SC, USA in 1986 and 1988 respectively. He then went on and received his MS and Ph.D. degrees in Computer Science from Clemson University, Clemson, SC, USA in 1991 and 1994 respectively.

He is currently a Professor in the Department of Computer Science and Engineering and the Director of the High-Performance Computation and Visualization Lab at the University of Nevada, Reno, USA. He has published more than 200 peer-reviewed journal and conference papers along with several book chapters. His research interests are in parallel computation, computational neuroscience, computer graphics, and virtual reality.

He is also a Senior Member of the ACM, and a Senior Member of the International Society for Computers and their Applications (ISCA).

Sergiu Dascalu is a Professor in the Department of Computer Science and Engineering at the University of Nevada, Reno, USA, which he joined in 2002. In 1982 he received a Master's degree in Automatic Control and Computers from the Polytechnic University of Bucharest, Romania and in 2001 a Ph.D. in Computer Science from Dalhousie University, Halifax, NS, Canada. His main research interests are in the areas of software engineering and human–computer interaction. He has published over 180 peer-reviewed papers and has been involved in numerous projects funded by industrial companies as well as federal agencies such as NSF, NASA, and ONR.

Trust and Security in Intelligent Autonomous Systems

Aakanksha Rastogi* and Kendall E. Nygard*
North Dakota State University, Fargo, ND, USA

Abstract

Autonomy is defined in terms of the degree of capability of a system or machine to function without human intervention. Degrees of autonomy can vary from requiring fully engaged human involvement at one extreme to having none at the other extreme. Levels of trust on the part of humans concerns the extent of belief or confidence in the system. When a system with some degree of autonomy makes decisions and carries out its functions, trust in the system may rise or fall in accordance with perceptions or measurements of the system performance. Measurement of trust is typically related to ethical, moral, social and legal norms of society, along with metrics related to taking responsibility. Trust is related to cybersecurity in that insecure systems inherently have low trust. The work of this paper surveys and explores concepts of trust in terms of relationships between humans and systems. An ontology that characterizes this relationship is provided. Trust issues as they pertain to the areas of cybersecurity and autonomy are characterized. The concept of anti-autonomy and counter measures that apply to autonomous weapon systems is also included.

Key Words: Autonomy, security, trust, intentionality, semi-autonomy, anti-autonomy, vulnerabilities, human-in-the-loop, and human-on-the-loop.

1 Introduction

Advances in the use of Artificial Intelligence (AI) in autonomous systems is revolutionizing decision making in society. Human-centric decisions have long been the norm in many application domains such as medical diagnosis, financial operations, driving cars, flying airplanes, and legal case research. However, in many domains, at a fast rate of change, humans are relinquishing decision making to autonomous systems that have intelligent capabilities. The quest for fully autonomous self-driving cars is a good example of machines undergoing a steady march toward increasing levels of autonomy and intelligence over time. Anti-lock brakes are now an old technology, but for many years some drivers viewed them with fear and regarded them as an inappropriate encroachment on driver control. More recently, some new cars are equipped with lane following assist technology, which automatically does things like keeping the vehicle in a lane, maintaining offset distances, accelerating, braking, etc. At some point self-driving cars may be fully autonomous. Many cyber-physical systems are heavily equipped with sensors, actuators, and controllers, but unlike earlier generation machines, in the march toward intelligent autonomy, also involve integrated symbolic or sub-symbolic AI to do their work.

When systems with some level of autonomous operation deviate from their expected behavior in negative ways, humans tend to decrease their level of trust in intelligent machine performance. In the other direction, it can also be true that repeated positive performance can incrementally increase a trust level. In some cases, autonomy is adjustable, and the machine itself may call for human intervention. For example, some robots are programmed to request human intervention under certain circumstances.

By definition, autonomous systems are capable of changing their behavior in response to unanticipated events during operations [41]. According to Hancock, “autonomous systems are generative and learn, evolve and permanently change their functional capacities as a result of the input of operational and contextual information. Their actions necessarily become more indeterminate across time” [16]. They can decide for themselves what to do and when to do it [13]. These systems can achieve their assigned goals by constructing and executing a plan without requiring any human intervention even in the face of unexpected events [45]. They can be deployed in remote environments where direct human control is not feasible or in environments that are hostile and dangerous to humans [13]. When compared to human-centric systems, some autonomous systems can have advantages of providing faster response times at lower cost. In addition, autonomous systems may not require as much training and monitoring as people often do. People also need things like medical support, guaranteed safe environments, and legal oversights. Most autonomous systems have a decision-making agent that is responsible for making decisions that simulate the human mind. Machine learning has opened new possibilities for systems to become intelligent enough to autonomously operate under widely varied circumstances with minimal or no human intervention. In a broad sense, such capabilities explains much of the rise of artificial intelligence (AI) in our society.

---

*Department of Computer Science. Email: {aakanksha.rastogi, kendall.nygard}@ndsu.edu.

In some cases, systems may be semi-autonomous systems inherently requiring human intervention to be successful in completing certain tasks. Even though inter-device communication may play a role, these systems require human operators to control higher levels of decision making [45]. Zilberstein refers to semi-autonomous systems as "systems that can operate autonomously under some conditions but cannot always complete an entire task as their own" [45]. Semi-autonomous systems can be classified as of SAS-I type if their planning process does not take human intervention into consideration. SAS-II types have planning processes that include knowledge of human intervention into consideration [45]. To coordinate with semi-autonomous systems, it is often important that they are aware of human interventions and can recognize the conditions under which autonomous actions cannot solely perform the operation to complete the tasks without human input. Moreover, humans must be thorough and precise with the decision-making processes and AI protocols with which these systems are equipped. This paper addresses key questions, understanding, and clarification of human-machine relationships.

The paper is organized as follows. Section 2 describes the concepts of trust in autonomous systems. Section 3 describes human interfaces with autonomous systems. Section 4 describes the measurement of trust in autonomous systems related to cybersecurity. Section 5 describes the understanding of the concept of anti-autonomy. Within Section 5 concerning drones, a description of counter measures or anti-drone technologies that help avert potential collateral damage and casualties caused by autonomous weapon systems is given. Section 6 describes the relation of security with autonomy. Section 7 provides conclusions.

2 Trusting Autonomous Systems

Humans have created intelligent machines and systems using AI protocols and advanced machine learning concepts and techniques. When the singularity occurs, intelligent machines are expected to create ever more intelligent machines, triggering an unrelenting escalation. Autonomous systems that use advanced AI techniques have shown improvements in deception and the use of experience, interventions, and control. With these changes in the nature of autonomous functionalities, trusting these systems has become more complex and challenging. In defining trust, there are often references to attitudes, beliefs, intentions, and behaviors. In the many definitions of trust offered, there is typically reference to expectations regarding outcomes or behaviors [28]. Specific to automation, trust can be described as "the attitude that an agent will help achieve an individual's goal in a situation characterized by uncertainty and vulnerability" [28]. Dutta et. al. defined trust in an autonomous system as "the ability of the system to successfully carry out a task, at a particular time, and in a situation characterized by vulnerability and uncertainty" [11].

Trust among humans and between human and automation are fundamentally different due to a lack of intentionality in autonomous systems [28]. Since human-centered trust is related to concepts of benevolence, value congruence and loyalty, human-automation trust presents difficulties concerning purpose [28]. Lee and See pointed out that trust between people is a part of a social exchange relationship which makes trust between humans different from trust in automation [28]. There is also a lack of symmetry (where trustor and trustee are aware of behavior, intents, and trust of others) between humans and machines [28]. Trust in automation is an attribution process whereby trust can be derived from direct observation of behaviors (performance), an understanding of the underlying mechanisms (process), or from the intended use of a system. Trust is an important mediator of relationship between humans and automation. Choi and Ji described trust to be a major determinant of reliance on and acceptance of automation, standing between people's beliefs toward automation and their intention to use it [4]. For instance, while investigating the importance of trust while adopting autonomous vehicles, the following dimensions are prominent: 1) transparency, for understanding and predicting vehicle operations, 2) technical competence, for perceiving vehicle performance, and 3) situation management, for confidence in the vehicle adaptively maintaining control in an unanticipated situation [4]. Anthropomorphism, which is attributing human characteristics to a nonhuman entity, can be an important determinant of trust [42]. Anthropomorphism can be viewed as a process of inductive inference, particularly concerning capacity for rational thought or agency and conscious feeling [42]. Trust is a multifaceted concept that refers to a belief in another behaving with integrity, benevolence, predictability or competence, a prediction of anthropomorphism increases confidence and trust [42]. To establish an assertion that anthropomorphism affects user trust, an experiment was conducted in which participants using a National Advanced Driving Simulator were instrumented for psychological assessment and randomly assigned to degrees of autonomy corresponding to normal, agentic, and anthropomorphic driving conditions. Human voices and gender information were a part of the anthropomorphic features [17, 42]. In terms of overall trust, participants who drove their vehicle in anthropomorphic condition had the highest trust, followed by Agentic, then by Normal [42]. This established statistically that the degree of anthropomorphism mediated the relationship between the vehicle condition and overall trust in the vehicle. These findings strengthen the concept that trust is related to perceptions of human mental capacities. Similar results on the effectiveness in the elicitation of positive perceptions of the agent upon introducing humanlike appearance and high autonomy in self-driving cars have also been reported. Additionally, mediation analyses revealed that the introduction of humanlike appearance and high autonomy induced by greater levels of anthropomorphism introduced feelings of social presence which imposed a positive impact on perceived intelligence, safety, and trust in the agent. This suggests that feelings of social presence during interaction with an agent is a determinant of the extent to which users perceive a driving agent as safe, trustworthy, and intelligent [27]. Even when automation is limited but anthropomorphism is high, an elevated feeling of trust and safety followed and resulted in positive perceptions of the system [26].

The concept of trust is somewhat elusive in the sense that intentionality, purpose, belief and reputation play a role, but so do credibility, consciousness, empathy, sympathy and responsibility. Correlations and relationships of these concepts with trust takes on importance when we analyze human-machine relationships. Again, in the self-driving car example, all of these concepts are in the minds of people when they consider peace of mind when riding in such a car and influence the level of trust a person has that the vehicle will safely and efficiently get them to their destination. One factor that influences the level of trust that a person has in any vehicle even if a driver is in control concerns the possibility of encountering unforeseen situations and the need to react appropriately and safely. Examples include encountering drunken drivers, construction zones, or a vehicle failure such as a tire spontaneously going flat. If software is handling such situations, there are serious questions about updates, such as how effective and how frequently they are done to account for new information. Apart from self-driving cars, robots and robotic humanoid assistants must deal with similar problems.

If data-centric machine learning methodologies are used for intelligent training, the ever-present issues surrounding the appropriateness and completeness of the data that is employed are encountered. In addition, there is a significant challenge that lies in supporting consciousness in a machine, and the attendant need for the machine to be fully aware and have an understanding of the human ways of doing things and reacting to their own stimuli as they conduct their activities. Seamlessly merging such machines into our daily lives as people will require these kinds of capabilities. It is also the case that humans are far from infallible. Every person has made decisions for which they thought twice or multiple times, agonized over whether they made the right decision, or wished that they had been able to predict unforeseen consequences. Advanced sensors, massive data sets, and rapid communication capabilities have resulted in great strides in machine awareness. Given that awareness is only one necessary ingredient of trust, and that trust is a key component of responsibility, it is clear that developing truly responsible machines is still in the future. Research concerning interactions among humans has established that high levels of trust drive responsible behaviors. This suggests that there is potential to build responsible systems whose behavior sympathetically and empathetically complement our expectations.

It is widely held that autonomous systems lack human emotions such as happiness, sadness, fear, anger, surprise and disgust. The concept of empathy concerns awareness and sensitivity to the feelings and thoughts of another, even if not fully communicated in an objectively explicit manner [8]. Empathy is often thought of as a vicarious experiencing of those feelings and thoughts of the other person. The related concept of sympathy is an affinity, association, or relationship between persons or things wherein whatever affects one similarly affects the other [7]. Robotic humanoid assistants can use speech recognition software to mimic human emotions. Software uses digitally converted speech waves as parameters, turns them into words and then uses a semantic decoder to convert words into meaning. Though there has been significant research into developing socially and emotionally adept robots with the help of speech recognition software, the software still makes mistakes, which can leave these robots with failures to understand human intent and emotions [14].

Trusting autonomous systems and delegating tasks to them to gain realization of their full values also requires a belief that these systems are going to be truly effective [33]. Once these systems are trusted, delegated tasks that are then fulfilled can help reduce associated personnel costs and improve safety. Developers who program the AI engines of autonomous systems can focus on providing the ability to carry out complex tasks, which can help to minimize the number of people tied to operations, resulting in money saved [33]. When the level of autonomy in weapon systems is increased there are improvements in war fighting capabilities while reducing the need for human operators [24]. However, the high level of autonomy in such systems in using advanced algorithms to detect targets and deploy attacks comes with major trust issues. Groups such as Human Rights Watch are constantly attentive to the issue and promote negotiations to impose preemptive bans on the development, production, and usage of fully autonomous weapons on the battlefields. Once deployed, autonomous weapons can be difficult to recall if a scenario changes, new information is obtained, or there is misidentification of a target [24]. The infamous 1991 failure of a Patriot missile system to track and intercept an incoming Scud missile resulted in the deaths of 28 soldiers and was caused by a subtle programming error.

3 Humans Interfacing with Autonomous Systems

Partial, sliding and semi–autonomous systems often require humans to interface with them during their operations. This introduces the concepts of human-in-the-loop and human-on-the-loop [31]. Human-in-the-loop (HITL) or semiautonomous robotic systems (RAS) autonomously perform a task for a certain time, then pause and wait for commands from a human operator before continuing. For instance, in autonomy used by HITL autonomous systems for searching, detecting, and evaluating threats; selection and engaging of targets are controlled and decided by humans. Human-on-the-loop (HOTL) systems can execute a task fully and independently but have a human in a monitoring or supervisory role, with an ability to intervene if the system fails or if an error condition arises. They are capable of being fully autonomous in performing an entire function on their own if allowed by their human supervisors. By keeping a human-on-the-loop, a need for interactive human and system interfaces is eliminated. However, deadly outcomes can occur. An example is the recent similar crashes of Boeing 737 Max aircraft in Indonesia and Ethiopia that killed 346 people. The trigger of the crashes was the failure of a sensor intended to accurately report the attack angle of the aircraft. Each aircraft was equipped with an autonomous system called the Maneuvering Characteristics Augmentation System, or MCAS, which is intended to use sensor input and autonomously take corrective action that down points the nose of the aircraft if it is about to stall. In each case, when the human pilot intervened, the MCAS system reacted by again initiating the down pointing action. Back and forth interactions of the automated system with pilot corrective action meant that eventually the multiple down pointing actions resulted in a crash known in the industry as uncontrolled flight into terrain. Many pilots have expressed frustration at being caught off guard by automated sudden descents of the aircraft. In the autonomy employed by HOTL weapons systems, RAS systems select and engage targets that were not decided upon by human supervisors. Humans can monitor the intention and performance of the weapon system and can cancel, interfere, or stop its operations if necessary. Applications where humans use supervisory control to directly control the system either involves an autonomously running process where human intervention includes a control algorithm which adjusts set points whenever necessary; or a process accepting a command, carrying it out autonomously, reporting results and waiting to receive further commands from the human [30].

A human who has an in-depth knowledge and understanding of technical details of code, algorithms, functionality, and behavior of a machine he/she is operating can better handle critical situations while averting or overriding machine decisions and taking control. This raises a concern as to how a situation should be dealt with when, for example, an autonomous vehicle is compromised by a malicious user while a non-technical human was interfacing with the vehicle. Little research has addressed this kind of question, which results in attack vulnerabilities. Predicting human behaviors in unavoidable situations is difficult given that autonomous systems are typically preprogrammed to not make choices that can be construed as dangerous [9]. Extending system modelling techniques to capture human behavior is extremely difficult due to complex psychological, physiological, and behavioral aspects of human beings.

4 Measuring Trust in Autonomous Systems

Trust assessment, measurement, and management require a thorough understanding of the concept of trust, often uncovering degrees of, and multi-dimensional nature of trust. Trust management concerns collecting, analyzing and presenting trust related evidence and making assessments and decisions regarding trust relationships between entities in a network [22]. Measuring trust relies upon quantitative values for traits such as reliability, competence, security vulnerabilities and robustness as well as transparency of control. Trust assessment is furthered when these factors are measured in uncertain and certain environments [11].

As autonomous systems become more complex, instability and uncertainty in workplace situations can increase due to increased cognitive complexity [32]. Accompanying feelings can be unsettling to people. When comprehension of an intricate automation system becomes difficult or impossible, high levels of trust are helpful in coping with uncertainty, particularly when situations are dynamically changing and there is little basis for decision making or means of exercising control by humans who are in or on the loop. The degree of trust influences the performance of the systems and also affects acceptance and reliance on automation, along with the strategies that operators use during automation. Hence, measurement of situations of trust and mistrust are necessary in predicting system performance [32]. One example of the cognitive process is illustrated by work of Oh et al. [32] analyses in which electroencephalograms (EEGs) were used to measure brainwaves in situations that involved trust and mistrust. Trust levels were found to be associated with effective decision-making and performance elevation through measurable increases in concentration. When mistrust was evident, stress and anxiety interrupted and was inimical to effective decision-making. A study by Wang et al. [40] used EEG signals and facial images to establish that human identity is important is assessing trust and assurance and drives effective human-machine interaction. In a tie-in with the Software Development Life Cycle, the study in [18] asserts that human trust in autonomy can be achieved by applying formal methods. Basically, a formal model of the autonomy software can help to verify that critical properties such as safety and service are in place, providing assurance that an autonomous system will satisfy requirements. As trust applies to autonomous vehicles, formal methods in conjunction with simulations and employing the Software Cost Reduction (SCR) toolset have been used to establish elevated levels of trust [18].

Another method of measuring trust in self-driving cars was described in [19]. The approach uses gaze behavior and eye-tracking in a visually demanding nondriving-related task during highly automated driving. Situational, dispositional, and learned automation settings were configured and trust levels were self-reported. Associations between gaze behaviors and the level of trust in the automation were established. The work sets the stage for further studies in which trust is evaluated by quantitative methods apart from self-reporting. There is an advantage in being non-invasive. In a further study a real-time sensing of trust levels was based upon an innovative model that maps psychophysiological measurements of human trust levels into human-machine interactions [20]. Finally, in the domain of self-driving cars, aircraft, and pharmaceuticals, trust levels can be derived from standards utilized by regulatory bodies that provide certifications for safety and reliability of safety-critical technologies that are employed [9].

5 Anti-Autonomy

When the integrity, behaviors and functionalities of an autonomous system is compromised in an attack, not only does it become vulnerable to future attacks, but also becomes a potential source of danger or a threat to other autonomous systems, agents and humans. Lack of intentionality makes it more difficult for humans to establish trust. A constant concern is that a wrong piece of code can make a system potentially dangerous and capable of wreaking havoc in their surroundings. One such example is the use of autonomous robots on battlefields. Potentially blamed for noncombatant casualties and collateral damages [2, 12], the usage of fully autonomous robotic weapons systems is banned by military operations in the United States. Autonomous weapon systems are accused of violating fundamental human values from an ethical point of view, including the ethical standards established by International Law of War [36]. Since battlefield robots and weapon systems do not experience anger, fear, or frustration in ways that humans do, they potentially pose greater risks towards noncombatants [2]. Similarly, in self-driving cars in situations where decisions must be made as to who is saved and who is killed it poses ethical dilemmas. The fundamental question of how to impart ethical and moral values into such systems arises. This gives rise to the concept of anti-autonomy, which leads to the perplexities of decision making in the event of the behavior of an autonomous system going awry.

Anti-autonomy basically considers how to counter attacks by autonomous systems. The work of Huang and Wicks [21] begins with analysis of attack strategies, then applied to a large-scale distributed intrusion detection framework to address issues of work division, information exchange and coordination among available Intrusion Detection Systems (IDS). One approach employs autonomous local IDS agents to perform event processing coupled with cooperative global problem resolution. The idea is to discern enemy intent as pioneered by Howard and use the framework to drive a basis for how different IDS components work together [21]. It has also been established that cooperation of a single detection system with remote detection systems located in other parts of the network can improve detection performance. The Cooperating Autonomous Detection Systems (CATS) is an instantiation of the approach [10]. A distributed monitoring environment can improve detection results. To increase the availability of the overall systems and to avoid instances of a detection system falling victim to itself, it is important that each subsystem performs attack detection autonomously. Although, detection accuracy may be increased by the intercommunication between subsystems, this is not a pre-requisite for global detection functionality and hence, autonomous behavior of systems should possess self-configuration, self-maintenance, self-healing, and self-optimization capabilities. Intrusion detection approaches using inclusion and distribution of agents on a network-wide basis to monitor the system effectively and efficiently and improve detection has also been reported by Crosbie and Spafford [5, 6].

5.1 Automation of Anti-Autonomy

Robots on the battlefield traditionally carry out tasks like detecting and neutralizing improvised explosive devices (IEDs); using sensors to detect hazards like radiation, biological agents, or chemicals; or conducting surveillance. The complexity of what robots are capable of is rapidly increasing. Unmanned Aerial Vehicles (UAVs) or drones are showing the way in terms of technological advancements, including the decision making and deploying of weapons. However, autonomous weapons systems are still largely designed for human-in-the-loop decision making. It is now common for a UAV to support military operations with both onboard weapons and surveillance capacities. Such systems can identify, locate and eliminate targets in combat zones. Equipped with radar antennas, navigation systems and satellite communication, they can identify, lock study, and attack targets, with weapons dispatched under human control. In some cases, remote operators are working from very distant locations. Newer generations of battlefield weaponry can autonomously assess a battle zone much faster and more thoroughly than a human can, and react very quickly if authorized to do so. UAVs are often carrying out missions that include coverage of non-combat zones, increasing the risk of collateral damage to civilian populations, which often raise special social and ethical concerns. Unlike replacing human forces in a battlefield, UAVs that are lost in battle can often easily be replaced with spares. Forecasts indicate that as many as 7 million drones will take flight by the year 2020. This is in contrast to a figure of under 40,000 piloted aircraft in operation today.

Given the enormous investment in technologically advanced UAVs and automated robotic systems, it is natural to consider ways to develop counter measures and protect against automated attacks. Terminology like anti-UAV and anti-drone technologies have entered the military vernacular. Basic approaches focus on detecting and intercepting unmanned aircrafts, with similarities to how Patriot missile systems have functioned for many years. Approximately 235 counter-drone products have been developed by 155 manufactures in 33 countries [29]. Products can be ground-based, hand-held or airborne. Detection strategies include radar, Radio Frequency, Electro-Optical, and Infrared. Interdiction methodologies include GPS spoofing and jamming of tracking systems and communication devices on enemy systems. For anti-drone systems specifically, trade and technology names include DroneGun, Advanced Test High Energy Asset, Laser Weapon Systems, Radar-guided missiles, DroneCatcher, SkyWall, SkyFence, Eagle Power and Drone Malware [35]. Challenges in the design and development of such systems include issues of precision, performance, practicality, detection, tracking and interdiction effectiveness. An example of a complication is that a C-UAS jamming system designed to stop UAV communication can also jam networks in small or commercial airplanes in the vicinity. Electro-optical systems and acoustic sensors have been known to confuse drones with birds or other airplanes. When used near airports, electromagnetic and radio frequency interference can cause air traffic control issues. In addition, many counter-measures are illegal or restricted in certain countries [37].

6 Security in Autonomy

Since autonomous systems are potentially threatening to humans, security and privacy issues are of importance, and lapses often require immediate attention and rectification. Hacking of these systems can also cause casualties and collateral damage. When autonomous weapon systems accept and process commands on the battlefield, although intended to respond and act within laws of war and rules of engagement, significant damage can be caused if victimized by hacking attacks [25]. When the software involves complex algorithms and control systems, software testing is often incomplete and vulnerabilities are present, offering invitations to hacking and hijacking. Internet connectivity opens other avenues for hacking. Networks of autonomous systems are supported by the interdomain routing protocol called the Border Gateway Protocol (BGP). Because the dynamic nature of the routing infrastructure that includes competitive, self-interested autonomous nodes, the BGP network is prone to vulnerabilities, failures, communication interruptions and malicious attacks [3, 23]. Multiple approaches have been developed to enhance BGP network security. Pretty good BGP (PGBGP) can detect anomalies and respond, including assessing the minimum number of autonomous systems that are required to adopt a distributed security solution that would provide protection against known exploits [23]. Other secured versions of BGP include secure-origin BGP, secure-BGP and pretty-secure BGP [3]. IP routing infrastructure is susceptible to critical security vulnerabilities and malicious attacks. Shue et. al. [38] found multiple autonomous systems with high concentrations of malicious IP addresses, and others that were disproportionately experienced high malicious activities in comparison with their equivalently sized peers. To determine which internet service providers and autonomous systems reveal high malicious behavior they used 10 blacklists, extensive DNS solutions, and local spam data. The blacklists, exploited hosts, phishing sites, bot command and control and malware downloads were used as inputs. Malicious activities were found among autonomous systems that were peering with other systems on a regular basis [38]. While malicious attacks are frequently launched by botnets, the originating autonomous systems and the systems with higher degrees of maliciousness resulted in penalizing legitimate traffic on the internet and causing extensive collateral damage.

Vulnerabilities and malicious attacks have reached into the world of autonomous and unmanned vehicles. Most vehicles are equipped with automation features such as satellite navigation, anti-lock braking systems (ABS), cruise control, lane departure assist, moving object detection (MOD) and parking assist, which provide at least semi-autonomous functionalities to these vehicles. The software behind these features are vulnerable to failures caused by cyberattacks, software and hardware anomalies, and defects that have been accidently introduced by the developers [44]. Yağdereli et al. [43, 44] provided a classification of threats and attacks and proposed development guidelines and mitigation strategies to use in the development of autonomous and unmanned vehicle systems [44]. Security of autonomous vehicles has also been discussed by Thing and Wu where they presented a comprehensive taxonomy to categorize security vulnerabilities, threats, attacks, and potential defenses in an autonomous vehicle in order for its infrastructure to be more secure and dependable [39]. To implement the quality, security, and safety during the development of autonomous embedded electronic systems inside autonomous vehicles, Gifei and Salceanu [15] proposed a Quality Safety Cyber Security Integrated Management System (QSCSIMS) which enhances security and safety aspects and also decreases time spent on following standards and associated costs. Autonomy systems are surrounded by security issues, vulnerabilities, and attacks induced by human elements who introduced bugs while programming; by hackers invading systems, or implementing complex algorithms that produces self-manifesting bugs.

7 Conclusion

This work provides a characterization of trust issues as they pertain to the areas of cybersecurity and intelligent autonomous systems. Self-driving cars and drones are prototypical examples of intelligent autonomous systems that are widely viewed as having positive impacts on human lives. However, such systems have not been completely successful in establishing that humans will fully trust that their behaviors always follow ethical, social, and legal norms of society. For self-driving cars in particular, people express concerns underlying safety, security, and decision-making, especially during critical situations and circumstances. Anthropomorphic characteristics are helpful in elevating trust, especially in robotic humanoid assistants. There is substantial technical progress in making these types of systems more reliable and in modeling and training them to be compliant with human values. Consciousness, empathy, and sympathy are characteristics that are difficult to support in intelligent autonomous systems, yet humans want these characteristics. Battlefield robots and weapon systems are advancing rapidly, yet are still largely operated with human-in-the-loop designs. Fully autonomous weapons systems that make strike decisions have not achieved the high level of trust needed for deployment, even in settings where they can be shown to exceed human decision-making performance. As weapons systems exhibit more autonomy, systems to counter them with anti-autonomy designs are becoming more prevalent. It is widely held that one small mishap can translate into disasters with collateral damage and loss of life, including non-combat civilian casualties. Autonomous systems are not limited to physical machines and devices, but also exist throughout the internet in the form of smart software agents and botnets accepting and executing commands. While some of these systems have gained some measure of trust by humans, measurement of trust is hampered by a lack of established standard measurement procedures. In addition, since such systems are developed and programmed by humans, they are prone to security issues, attacks, vulnerabilities, and threats, for which researchers have been exploring intrusion detection techniques, approaches, and methods to avert failures, outsmart hacking attacks, and prevent disasters.

References

[1] Fahad Alaieri and André Vellino, "Ethical Decision Making in Robots: Autonomy, Trust and Responsibility," 8th International Conference on Social Robotics, Kansas City, MO, USA, pp. 159-168, November 1-3, 2016.

[2] Ronald C. Arkin, "Ethics and Autonomous Systems: Perils and Promises [Point of View]," *Proceedings of the IEEE*, 104(10):1779-1781, 2016.

[3] Sara Bakkali, Hafssa Ben Aboud, and Mouad Ben Mamoun, "Security Problems in BGP: An Overview," *2013 National Security Days (JNS3), IEEE*, Rabat, Morocco, pp. 1-5, April 26-27, 2013.

[4] Jong Kyu Choi and Yong Gu Ji, "Investigating the Importance of Trust on Adopting an Autonomous Vehicle," *International Journal of Human-Computer Interaction*," 31(10):692-702, 2015.

[5] Mark Crosbie and Eugene H. Spafford, Active Defense of a Computer System using Autonomous Agents," *Computer Science Technical Reports. Paper 1187,* 1995.

[6] Mark Crosbie and Eugene H. Spafford, "Defending a Computer System using Autonomous Agents," *Department of Computer Science Technical Reports. Paper 1200*, 1995.

[7] Definition of Empathy, "Merriam-Webster.com," https://www.merriam-webster.com/dic tionary/empathy. December 2018.

[8] Definition of Sympathy, "Merriam-Webster.com," https://www.merriam-webster.com/dic tionary/sympathy, December 2018.

[9] Louise Dennis, Michael Fisher, Marija Slavkovik, and Matt Webster, "Formal Verification of Ethical Choices in Autonomous Systems," *Robotics and Autonomous Systems,* 77:1-14, 2016.

[10] Falko Dressler, Gerhard Münz, and Georg Carle, "Attack Detection using Cooperating Autonomous Detection Systems (CATS)," *Wilhelm-Schickard Institute of Computer Science, Computer Networks and Internet*, 2004.

[11] Raj Gautam Dutta, Xiaolong Guo, and Yier Jin, "Quantifying Trust in Autonomous System under Uncertainties," *29th IEEE International System-on-Chip Conference*, Seattle, Washington, USA, pp. 362-367, September 6 -9, 2016.

[12] Amitai Etzioni and Oren Etzioni, "Pros and Cons of Autonomous Weapons Systems," *Military Review*, 97(3):7281, 2017.

[13] Michael Fisher, Louise Dennis, and Matt Webster, "Verifying Autonomous Systems," *Communications of the ACM*, 56(9):84-93, 2013.

[14] Pascale Fung, "Robots with Heart," *Scientific American*, 313(5):60-63, 2015.

[15] Simona Gifei, Alexandru Salceanu, and Gheorghe Asachi, "Integrated Management System for Quality, Safety and Security in Developing Autonomous Vehicles," *2017 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, Bucharest, Romania, pp. 673-676, March 23-25, 2017.

[16] P. Hancock, "Imposing Limits on Autonomous Systems," *Ergonomics*, 60(2):284-291, 2016.

[17] Renate Häuslschmid, Max von Bülow, Bastian Pfleging, and Andreas Butz, "Supporting Trust in Autonomous Driving," *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, Limassol, Cyprus, pp. 319-329, March 13-16, 2017.

[18] Constance L. Heitmeyer and Elizabeth I. Leonard, "Obtaining Trust in Autonomous Systems: Tools for Formal Model Synthesis and Validation," *Proceedings of the Third FME Workshop on Formal Methods in Software Engineering*, Florence, Italy, pp. 54-60, May 16-24, 2015.

[19] Sebastian Hergeth, Lutz Lorenz Roman Vilimek, and Josef F. Krems, "Keep Your Scanners Peeled: Gaze Behavior as a Measure of Automation Trust during Highly Automated Driving," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 58(3):509-519, 2016.

[20] Wan-Lin Hu, Kumar Akash, Neera Jain, and Tahira Reid, "Real-Time Sensing of Trust in Human-Machine Interactions," *IFAC-PapersOnLine*, 49(32):48-53, 2016.

[21] Ming-Yuh Huang, Robert J. Jasper, and Thomas M. Wicks, "A Large Scale Distributed Intrusion Detection Framework based on Attack Strategy Analysis," *Computer Networks*, 31(23-24):2465-2475, 1999.

[22] Tao Jiang and John S. Baras, "Trust Evaluation in Anarchy: A Case Study on Autonomous Networks," *Proceedings IEEE INFOCOM 2006, 25TH IEEE International Conference on Computer Communications,* Barcelona, Catalunya, Spain, pp.1-12, April 23-29, 2006.

[23] Josh Karlin, Stephanie Forrest, and Jennifer Rexford, "Autonomous Security for Autonomous Systems," *Computer Networks*, 52(15):2908-2923, 2008.

[24] Keith Kirkpatrick, "Can We Trust Autonomous Weapons?" *Communications of the ACM*, 59(12):27-29, 2016.

[25] Michal Klincewicz, "Autonomous Weapons Systems, the Frame Problem and Computer Security," *Journal of Military Ethics*, 14(2):162-176, 2015.

[26] Jae-Gil Lee, Jihyang Gu, and Dong-hee Shin, "Trust in Unmanned Driving System," *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human Robot Interaction Extended Abstracts,* Portland, Oregon, USA, pp. 7-8, March 2-5, 2015.

[27] Jae-Gil Lee, Ki Joon Kim, Sangwon Lee, and Dong-Hee Shin, "Can Autonomous Vehicles be Safe and Trustworthy? Effects of Appearance and Autonomy of Unmanned Driving Systems," *International Journal of Human-Computer Interaction*, 31(10):682-691, 2015.

[28] John D. Lee and Katrina A. See, "Trust in Automation: Designing for Appropriate Reliance," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 46(1):50-80, 2004.

[29] Arthur Holland Michel, "Counter-Drone Systems," Center for the Study of the Drone at Bard College, 2018.

[30] Sirajum Munir, John A. Stankovic, Chieh-Jan Mike Liang, and Shan Lin, 'Cyber physical System Challenges for Human-in-the-Loop Control," *8th International Workshop on Feedback Computing*, San Jose, California, USA, June 25, 2013.

[31] Saeid Nahavandi, "Trusted Autonomy between Humans and Robots: Toward Human-on-the-Loop in Robotics and Autonomous Systems," *IEEE Systems, Man, and Cybernetics Magazine*, 3(1):10-17, 2017.

[32] Seeung Oh, Younho Seong, and Sun Yi, "Preliminary Study on Neurological Measure of Human Trust in Autonomous Systems," *IIE Annual Conference. Proceedings*, Pittsburgh, Pennsylvania, USA, pp. 1066-1072, May 20–23,2017.

[33] Phillip E. Pournelle, "Trusts Autonomous Machines," *U.S. Naval Institute Proceedings*, 143(6):38-41, 2017.

[34] Sabrina Deutsch Salamon and Sandra L. Robinson, "Trust that Binds: The Impact of Collective Felt Trust on Organizational Performance," *Journal of Applied Psychology*, 93(3):593-601, 2008.

[35] Rahul Sasi, "Maldrone the First Backdoor for Drones," Garage4hackers Forum. http://garage4hackers.com/entry.php?b=3105, January 2015.

[36] Frank Sauer, "Stopping 'Killer Robots': Why Now is the Time to Ban Autonomous Weapon Systems," *Arms Control Today*, 46(8):8-13, 2016.

[37] "7 Big Problems with Counter Drone Technology (Drone Jammers, Anti Drone Guns, etc.)," Rupprecht Law P.A. https://jrupprechtlaw.com/drone-jammer-gun-defender-legal-problems. December 2018.

[38] Craig A. Shue, Andrew J. Kalafut, and Minaxi Gupta, "Abnormally Malicious Autonomous Systems and Their Internet Connectivity," *IEEE/ACM Transactions on Networking*, 20(1):220-230, 2012.

[39] Vrizlynn L. L. Thing and Jiaxi Wu, "Autonomous Vehicle Security: A Taxonomy of Attacks and Defenses," *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Chengdu, China, pp. 164-170, December 15-18, 2016.

[40] Min Wang, Hussein A. Abbass, and Jiankun Hu, "Continuous Authentication using EEG and Face Images for Trusted Autonomous Systems," *14th Annual Conference on Privacy, Security and Trust*, Auckland, New Zealand, pp. 368-375, December 12-14, 2016.

[41] David P. Watson and David H. Scheidt, "Autonomous Systems,' *Johns Hopkins APL Technical Digest*, 26(4):368376, 2005.

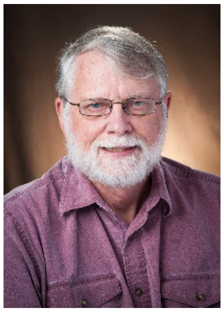
[42] Adam Waytz, Joy Heafner, and Nicholas Epley, "The Mind in the Machine: Anthropomorphism Increases Trust in an Autonomous Vehicle," *Journal of Experimental Social Psychology*, 52:113-117, 2014.

[43] Alexander M. Wyglinski, Xinming Huang, Taskin Padir, Lifeng Lai, Thomas R. Eisenbarth and Krishna Venkatasubramanian. Security of autonomous systems employing embedded computing and sensors. *IEEE Micro*, 33(1):80-86, 2013.

[44] Eray Yağdereli, Cemal Gemci, and A. Ziya Aktaş, "A Study on Cyber-Security of Autonomous and Unmanned Vehicles," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 12(4):369-381, 2015.

[45] Shlomo Zilberstein, "Building Strong Semi-Autonomous Systems," *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, Texas, USA, pp. 4088-4092, January 25-30, 2015.

Aakanksha Rastogi is a Ph.D. student in Software Engineering at North Dakota State University (NDSU). She also earned her MS in Software Engineering from NDSU and has four years of industry experience as a Software Developer and a Quality Assurance Analyst. She has instructed the UCodeGirl summer camp program in coding and cybersecurity for middle and high school girls. Her research interests include cybersecurity, software engineering, secure software development practices, autonomous and semi-autonomous systems.



Kendall E. Nygard is a full professor and serves as Department Chair of Computer Science at North Dakota State University (NDSU). He is also the founder and Director of the NDSU Institute for Cybersecurity Education and Research. He has served in Washington D. C. as a Jefferson Science Fellow at the U. S. Department of State and Senior Science Advisor at USAID. He is currently a U. S. Department of State Virtual Fellow. Dr. Nygard is also a fellow of the International Academy, Research, and Industry Association. He is a recipient of the Chamber of Commerce NDSU Distinguished Faculty Service Award. Dr. Nygard earned his PhD degree at Virginia Polytechnic Institute. He has advised 25 PhD students and more than 150 Master of Science students. Application areas in which he has conducted research include cybersecurity, smart grid, sensor networks, unmanned air systems routing and scheduling, wireless ad hoc networks, and social media. His primary methodologies are big data analytics, optimization models, artificial intelligence, and simulation.

A Framework for Virtualizing Joystick Controls in a Flight Simulator Training Environment

Alex Redei*
Central Michigan University, Mt. Pleasant, MI 48859, USA

Sergiu Dascalu† and Frederick C. Harris, Jr.†
University of Nevada, Reno, NV 89557, USA

Abstract

In aviation there can be little room for error. This paper explores software arbitration of two joysticks controlled by two pilots, where each joystick is independent of the other and each pilot's actions are potentially equally valid. In such scenarios, it can be difficult to know which commands are valid, and which commands should be ignored. Inspired by historic conflict resolution scenarios in commercial aviation history, we developed a framework for virtualizing joystick commands from two multi-axis joysticks. The framework has been utilized in a two-person flight simulator, where different joystick conflict resolution techniques were modeled and evaluated. In this paper, we detail our framework for arbitrating conflicts in a multi-axis joystick system, thereby increasing the responsiveness of control input in a potentially conflicted state. Both the framework's hardware prototype and software system are described and the results of implementing and evaluating three joystick conflict resolution techniques are presented and discussed.

Key Words: Human-computer interaction, joystick, input mapping, flight simulator, pilot training.

1 Introduction

In a training environment, the learning curve can be greatly reduced by offering immediate and focused feedback to the person being trained. In the realm of pilot training, this feedback is especially crucial, as good training is of the utmost importance. The correct maneuvering and operation of an aircraft can be a life or death situation.

Motion flight simulators are an innovative training tool that can help with this problem. They help pilots learn the ropes of flying an aircraft, and train for component failures, without the risk of death.

The closer the simulation is to real life, the more meaningful the experience will become. This means that any simulation system must provide a constant feedback loop to its pilots. Such feedback can come in many forms, from audible alerts, to visual feedback, to the tactile responses of a control mechanism.

In the real world, planes are not just piloted by one person – there is always at least two pilots on any commercial airliner in the United States (per FAA's two-person cockpit rule [6]). European airlines, while not all strictly two-pilot planes, seem to be following this trend [13]. In order to keep the training consistent with the real world, we have to support two people in the cockpit as well: a pilot and a co-pilot.

But what happens when both are at the controls? Who should the simulator respond to? Who should get the feedback? Moreover, if everyone has a duplicate set of the same controls, whose commands should be listened to?

Clearly, a system is needed to consolidate and prioritize the various inputs being provided to the simulation software.

In this article, we discuss a methodology for handling the intricacies of a multi-person, multi-axis motion simulator using an approach developed in our lab. The rest of this paper is organized as follows: Section 2 is devoted to the aviation history of Air France Flight 447, from which much of this framework's inspiration has been drawn. Related works are presented in Section 3, hardware is described in Section 4, and software architecture and design presented in Section 5. The methodology of our conflict resolution software is detailed in Section 6. Section 7 addresses the experimental results we acquired from our work with the newly developed framework, and Section 8 wraps up our findings with several concluding remarks and directions of future work.

2 Air France Flight 447

Air France flight 447 is the tragic story of how poor software design and a miscommunication between the pilots can lead to catastrophe [3, 19]. Air France flight 447 departed Rio de Janeiro's Galeão airport on May 31st 2009, and was expected to arrive in Paris' Charles De Gaulle airport the following day. Flight 447 is interesting because geographically it crosses both the Atlantic Ocean and the equator, as depicted in the route map shown in Figure 1.

En-route to Paris flight 447 crossed the intertropical convergence zone, also known as the ITCZ, a band of powerful storms situated around the equator [14]. Storms in this region can reach 50,000 ft in altitude, well above the altitude commercial airplanes can fly at [10]. Violent thunderstorms form as air masses from the two hemispheres interact in this region. On the particular day of Air France flight 447's tragedy,

---

*Dept. of Computer Science. E-mail: redei1a@cmich.edu.
†Dept. of Computer Science and Engineering. E-mail: {dascalus@cse.unr.edu, fred.harris@cse.unr.edu}.

Figure 1: Flight plan of Air France Flight 447 [4]

Figure 2: The Intertropical convergence zone [4, 8]

they had no choice but to fly through the storm. A picture of the ITCZ from July 12th, 2000 can be seen in Figure 2.

The decision to fly through the storm led to tragedy. Upon flying through the storm, devices called pitot tubes located at the nose of the aircraft iced over. A pitot tube is a critical device in aviation because it measures the airspeed as air moves over the body of the aircraft. For a period of 47 seconds, two of the three pitot tubes froze over and malfunctioned, reporting invalid airspeed information [3]. The onboard flight computer, being unable to reconcile the differing and invalid airspeed measurements, errored out, turning off the autopilot and returning control to the pilots.

Co-pilot Pierre Bonin took control of the aircraft upon discovering that the autopilot had turned off. He was unfamiliar with flying through the ITCZ, and his anxiety in this situation was apparent in the voice recorder. Co-pilot Bonin, feeling that the aircraft was losing altitude, made the fatal decision that day to command the plane to climb under manual control using his joystick, without communicating to his co-pilot David Robert. On an Airbus A330, which is the aircraft involved, there are two joysticks that control the pitch and roll of the aircraft, symmetrically located on the right and left side of the cockpit, that is called a "side-stick." An image of the Airbus A330 side stick is shown in Figure 3.

Figure 3: The Airbus A330 side stick [3]

David Robert, suspecting that the aircraft may have been in a stall condition, attempted to correct the situation by commanding the aircraft to pitch down. However, unbeknownst to co-pilot Robert, co-pilot Bonin was still commanding the aircraft to climb using his joystick, so the on-board computer was receiving conflicting information from the two side-sticks. A diagram showing how the side-stick is used to manipulate the control surfaces on the aircraft is shown in Figure 4. The software was programed to override the joystick inputs in such scomes, and in this case that meant the pilot's conflicting commands were cancelling each other out.

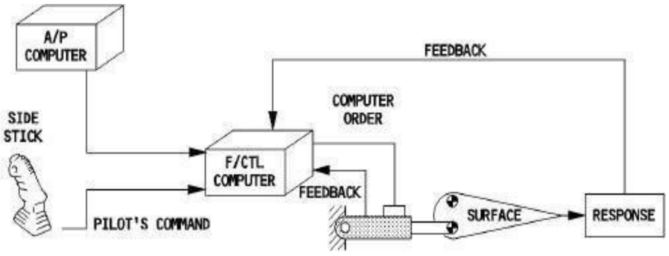


Figure 4: Details of the Airbus A330 side stick mechanism [9]

In the end, the tragedy of Air France flight 447 was completely preventable. Poor software design and a miscommunication between the co-pilots doomed the flight. The decision to command the aircraft to climb at such a high altitude, and the confusion over who was in control of the aircraft, led to a stall from which Air France flight 447 never recovered. Co-pilot David Robert's final words were recorded at 2:14 am where he exclaimed “Dang it, we’re going to crash. This can’t be happening!” [19]. The impact into the Atlantic Ocean killed all 228 onboard [3].

In this paper, we explore different software techniques for joystick conflict resolution through virtualization, a technique we hope will reduce the likelihood of another situation similar to Air France flight 447 repeating itself.

3 Related Work

To control an aircraft in flight, a pilot has 6 main controls: engine speed, ailerons, elevators, the rudder, flaps, and spoilers [18]. A NASA-made diagram with the location of each of these controls is shown in Figure 5.

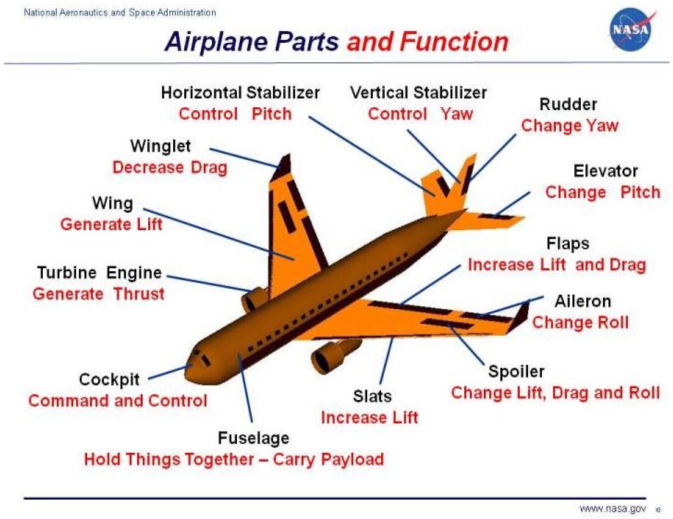


Figure 5: Critical parts of airplane flight [4]

These 6 controls allow the pilot to control the pitch, roll, and yaw of the aircraft (as well as its speed, and other aerodynamic properties). The pilot needs to keep constant mind of these controls, so that he or she can keep the plane flying and headed in the right direction.

As technology has improved, computers have been integrated into almost everything [12]. This is no different in the aviation industry. Nowadays, computer software is an essential part of keeping traffic organized in the air traffic controller (also sometimes abbreviated as ATC) [5, 17]. An example of this software’s interface is shown in Figure 6.

Technology has changed not just the ATC, but also how the pilots fly planes. An auto-pilot can fly the plane for hundreds of miles without needing human input. In addition, technology has also changed how pilots are trained.

Airline pilots have used flight simulators as an essential training and educational device for decades [1]. Flight simulators have been in consistent use in commercial aviation since the 1960s, and are a safer alternative to training their pilots [2, 11].

Ironically, sometimes the simulator used for training can cost more than the actual airplane.

This paper explores different software mechanisms for addressing the problem of joystick conflict resolution. In terms of physical equipment, the problem of two-person pilot cockpits is replicated in a hardware prototype.

Figure 6: An example of the software used in the air traffic controller [5]

4 Hardware Prototype

In order to model a software framework for the joystick conflict resolution, first a hardware prototype was needed. We used a flight simulator for this. There are two input devices in our simulator, and each operator (pilot and co-pilot) has a duplicate set of controls. The existing simulator does not handle multi-axis joystick input well. For example, if there are two pilots (a pilot and a co-pilot) and both press the deploy landing gear button, then the simulator will be unable to distinguish who pressed the button (and hence to whose responsibility it was to be assigned to). This is similar to the problems encountered by Air France flight 447, as outlined earlier in Section 2.

The first control is a joystick. The joystick in the simulator has two axes (x, y) that allow the pilot to control the elevators and ailerons. The ailerons control the roll, and the elevators control the pitch of the aircraft. In the joystick, this input is obtained via two analog potentiometers. The two potentiometers capture the commands on the x and y axes. When the joystick is swung to the right or to the left, the resistivity of the potentiometer changes, and thus a different voltage is read from the joystick in that axis. This style of input conversion is called an analog joystick, and according to an ACM SIGCHI bulletin, this joystick would be classified as a multi-axis joystick [7].

In addition, there are 4 buttons that are mappable to various controls, and a 4-way hat switch which allows the pilot to look around the cockpit.

An image of the first joystick is shown in Figure 7 and a potentiometer is shown in Figure 8.

The next control is the throttle body. The throttle controls the thrust of the engines, by way of controlling the speed of the engines.

The throttle is a single-axis joystick that uses an analog potentiometer to measure the “throw” of the throttle. In

Figure 7: The 2-axis, analog, 4-button joystick [15]

Figure 8: The potentiometer inside the joystick x-axis. The potentiometer senses the axis position and conveys the information over 3 voltage regulated wires

addition, the joystick has two buttons that can be mapped to various functions (such as deploying/retracting the landing gear). The second joystick is shown in Figure 9.

There are exact duplicates of the controls described above inside the cockpit: one for the pilot, one for the copilot. Thus, we have a challenge, which set of controls should we use?

In a real airplane, the controls would typically be tied together via a mechanical linkage. However, in the simulator we have no such linkage. Thus, we need to create a system which can respond to inputs from both pilots with some sort of definable priority.

Figure 9: An example single-axis analog joystick similar to the one we use, but not exactly the same model [16]

For the hardware portion of our solution, we used an Arduino Leonardo. We chose this board because unlike the very popular Arduino Uno, the Leonardo has an ATMega 32u4 chipset that can emulate HCI devices, and operate in USB-slave mode.

We found that we wouldn't be able to get an Arduino Uno recognized by the computer as a joystick without having to write our own Windows drivers. Instead, we found that using the ATMega 32u4's built in HCI device emulation allowed us to get around this issue.

To connect everything together, a wiring harness was used to convert the Molex connectors supplied by the manufacturer on the ends of the joystick, into jumper cables that can be plugged into the Arduino.

The six analog joystick inputs wire into six analog inputs, and the 20 buttons wire into digital inputs (via a shift register due to there being only 14 inputs available). A circuit diagram showing how all these ties together is shown in Figure 10. A picture of the hardware breadboard prototype is presented in Figure 11, and the wirings to the joysticks are shown in Figure 12.

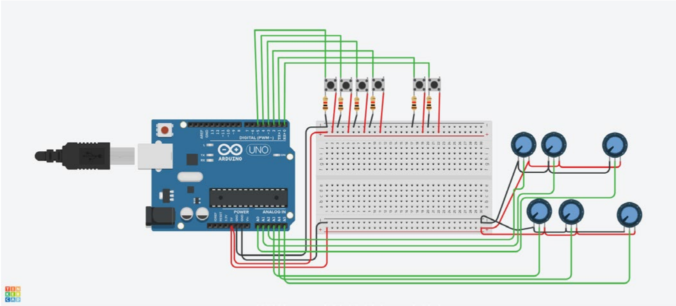


Figure 10: A depiction of the circuit diagram. Note that the breadboard is not used, except to consolidate the 5V power and ground lines

5 Software Design

First, the requirements of a software system that would process the multi-axis input from multiple joystick controllers are outlined in Table 1.

To better understand the interaction of the system, use case diagrams were created, as shown in Figure 13.

For example, in one of the use cases, only one operator is

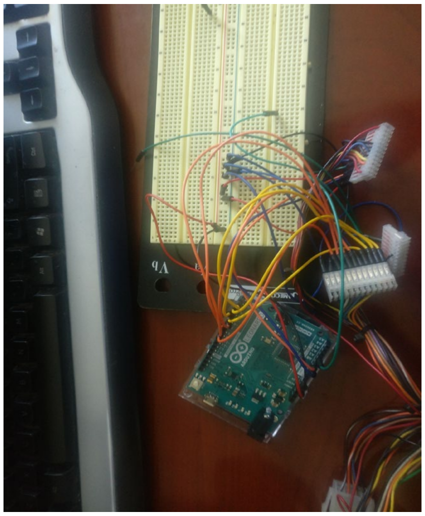


Figure 11: A photo of the actual hardware

Figure 12: Additional details of the actual hardware

pushing a control, say the deploy landing gear button. In this simple case, the controller maps the control command to a keyboard command and passes it along to the flight simulation software.

A more complicated use case scenario might involve both pilots providing input to the controller simultaneously, as shown in the MultiInput use case. In this case, the input of each pilot's control is tested against a weight, and combined together to form a "score" for that axis. Afterward, that axis is passed along to the simulation software via the USB joystick interface.

The code itself was written as a C++ 'sketch' in the Arduino IDE (v 1.6.5).

The analog inputs are read (those would be the axes: rudder, ailerons, elevators, etc.) and converted to a 10-bit digital value (in the range 0-1023).

Table 1: Software requirements

Requirement	Priority	Description
1	High	The software must be able to convert an analog axis input into a digital 0-1023 value.
2	High	The software must be able to convert button presses into a digital 0/1 value.
3	High	The software must be able to handle at least 4 analog axes of input.
4	High	The software must be able to handle at least 6 different types of button presses.
5	Medium	The software should be able to handle simultaneous inputs from multiple sources (analog axes, buttons, etc.).
6	Medium	The software should be able to map the analog axes to various software controls (such as rudder, throttle, elevators, or ailerons).
7	Medium	The software should be able to prioritize multiple inputs from the pilot and co-pilot.
8	Low	The software should be able to emulate keyboard presses.

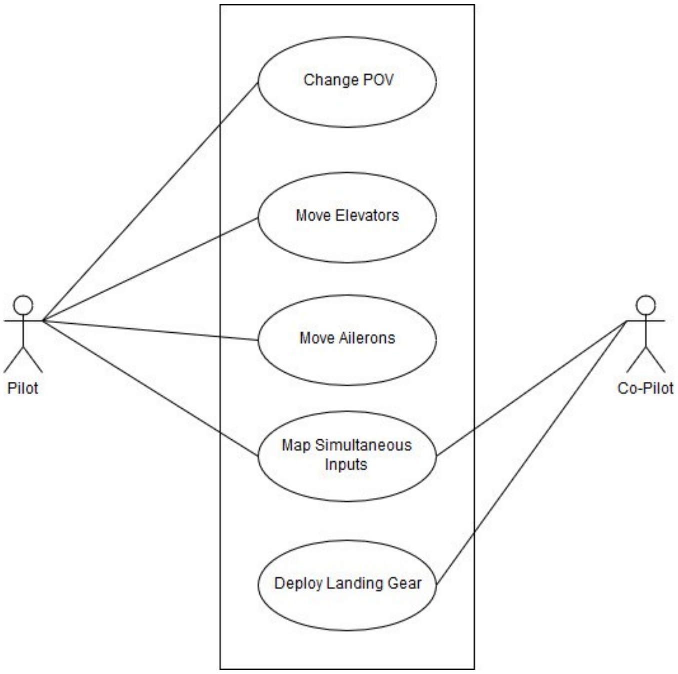


Figure 13: The use case diagram of the developed software

The button presses are read via digital inputs hooked into a pull-down resistor. The software keeps track of the prior known state of the button. When the circuit senses a change in the voltage, the current state is compared against the prior known state. If the state has changed, we report that change to the OS, and override the prior known state with the current state. At the moment, we can detect changes as fast as 1/20th of a second. It may be possible to detect changes faster, but for our purposes (civil aviation simulation) this seems sufficient.

6 Methods Evaluated

When co-pilots David Robert and Pierre Bonin provided conflicting input through the Airbus's side-sticks during the Air France flight 447 flight, the computer handled the conflict by averaging the inputs from the two joysticks. This led to disaster, as discrepancies within the joystick system, known as dual-inputs, was resolved in the Airbus software by averaging the joystick axis values of the two joysticks. Since the joystick inputs were the exact opposite, the flight computer cancelled out the pilots' commands and did nothing as the aircraft plummeted into the ocean. The co-pilots struggled to understand why they were not in control of the aircraft. Each had assumed their joystick was in control.

There are several methods that could be employed in such scenarios. Using the virtual joystick controller described above, we replicated three different scenarios. The first is rather simple: it involves replicating the value-averaging technique of the Airbus A330. Simply put, the firmware takes a voltage reading on each potentiometer, and averages the two values across the joystick axes. This technique has the advantage that agreeable dual-input can multiply the speed of a turn, but it does not handle miscommunication well between the pilots.

The second technique implemented and evaluated was to simply make the left-seat joystick the master joystick. Whenever the left-joystick was active, commands from the right-seat joystick would be ignored. While this technique would have prevented the Air France flight 447 tragedy, what if the roles were reversed and the pilot in the left seat was in a confused state? Or what if the joystick in the left seat was damaged, providing incorrect input and overriding the commands of the valid right-seat pilot? One of the advantages to having two pilots with two independent joysticks in the cockpit is the ability to transfer control in the event some part of the cockpit was damaged or sabotaged. Unfortunately, no matter which joystick was set as the master, experimentation with this technique left much to be desired in the way of redundancy.

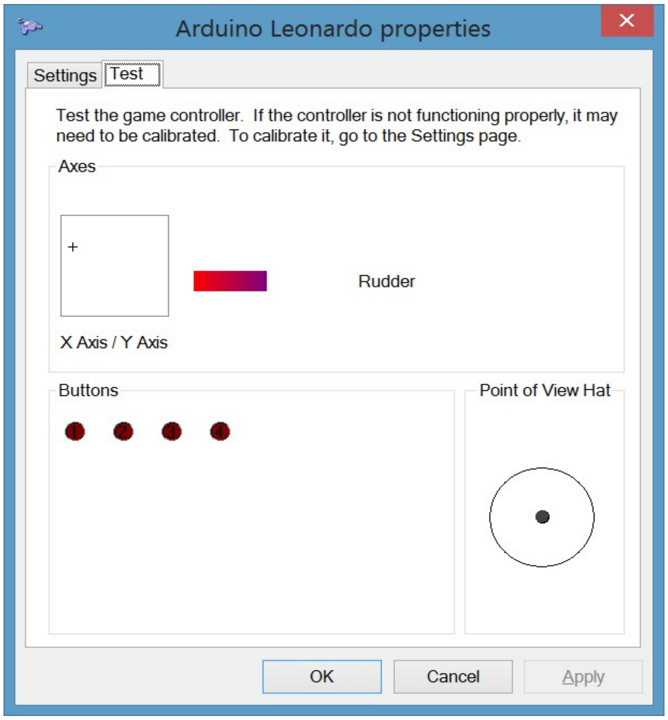
Finally, the third technique evaluated was the use of a joystick toggle button. Upon pressing the toggle button, the virtualization firmware would transfer control to the right-side or left-side joystick. Commands from the other joystick would be ignored when the active joystick was in use. Commands from the non-active joystick may have been accepted when the active joystick was not in use. To handle such scenarios, the software uses a weight distribution table. Each joystick's input is converted into an integer value (such as 0-1023 for an analog joystick axis). That value is then multiplied against a weight in the table (based on the role of the person in control of that joystick), and then assigned an action in software (such as moving the elevators to achieve a change in pitch). In the end, we believe that this was the right conflict resolution technique for the joystick virtualization. It allowed for a redundant cockpit, by utilizing the joystick toggle to switch joystick control, while not permitting there to be confusion about who is in control, as it is the case with the other two techniques.

7 Experimental Results

Utilizing the ATMega 32u4 micro controller, we emulated a native USB joystick to the flight computer. Putting the device into slave USB mode, it is recognized as a peripheral. This peripheral is both powered by the USB cable and can send/receive data through it. The firmware on the ATMega then converts the input signals from the joystick axes, buttons, and controls into a virtual game controller device in Windows.

This interface does not require a driver. One simply connects it to the computer and it emulates a game controller, thus using the built-in game controller of Microsoft Window®.

A virtual joystick with 4 buttons, two axes (x, y), a rudder control, and a POV (point of view) hat switch is depicted in Figure 14. The joystick axes are mapped to pitch and roll control of the aircraft.



Test the game controller. If the controller is not functioning properly, it may need to be calibrated. To calibrate it, go to the Settings page.

Figure 14: A joystick being recognized in windows by the system and the 3rd analog axis being mapped to the rudder

The initial experience was seamless. For those that are familiar with flight simulators, all the controls were mapped in the typical way.

The pilot controls the ailerons and elevator (pitch and roll) via the first joystick, and the throttle (engine speed) via the second joystick. The co-pilot controls the landing gear via a button on their second joystick (and has other possibilities with his or her controls that are not yet defined).

Things get interesting when multiple people, such as the pilot and the co-pilot, start to provide input at the same time. Based on each operator's role, and a weight tied to their particular joysticks, the software computes a resulting feedback value that then gets passed along to the simulation. For example, the system allows for a co-pilot to take over a flight if the pilot is incapacitated (of course, all these happen in simulation).

Better yet, this solution allows us to try new things, such as mapping the controls in a unique way. For example, we could map the co-pilot's joystick to control the rudders (pitch), while the pilot's joystick could control the ailerons and elevators (pitch and roll).

Our flight simulator was initially lacking any sort of pitch movement. This made the plane harder to control. Others have got around this problem with hardware solution: a joystick that can "twist." The twisting motion is measured and transformed into rudder movement that controls the pitch of the aircraft. However, in our flight simulator this would not be easily done because replacing the joysticks would require more space, and may not be compatible with other components. Hence, this solution allows us to achieve novel things such as repurposing a co-pilot control to maneuver the rudders, something that was not possible with our existing setup.

8 Conclusion

In this paper, we presented a method for handling input from multiple axes in a flight simulator application. As our related research showed, while joysticks and peripherals are commonplace, the systems to handle multi-joystick input, especially with different people at the helm of the controls, are not as mature.

The initial experimental results showed that our method could create a system capable of reading and responding multi-axis controls, without the need to write a driver in Windows. In the future we hope to extend this method to include the ability to emulate key presses on a keyboard.

References

[1] BAA Training Inc., "Full Flight Simulators – In 20 Years' the Number Will Double," [Online], Available: https://www.baatraining.com/full-flight-simulators-in-20-years-the-number-will-double/, [Accessed January 15, 2019].

[2] R. Bradley and D. Abelson, "Desktop Flight Simulators: Simulation Fidelity and Pilot Performance," *Behavior Research Methods, Instruments, and Computers,* 27(2):152-159, 1995.

[3] B. Gilissen, "The Last Four Minutes of Air France Flight 447," [Online], Available: http://www.spiegel.de/international/world/death-in-the-atlantic-the-last-four-minutes-of-air-france-flight-447-a-679980.html, [Accessed April 1, 2019].

[4] N. Hall, "Airplane Parts and Functions," [Online], Available: https://www.grc.nasa.gov/www/k-12/airplane/airplane.html, [Accessed August 17, 2018].

[5] C. Howard, "Airservices Australia Selects Saab Integrated Tower Automation Technology for Four Airports," [Online], Available: https://www.intelligent-aerospace.com/articles/2015/02/airservices-australia-selects-saab-integrated-tower-automation-technology-for-four-airports.html, [Accessed August 14, 2018].

[6] T. Inefuku, "FAA Requires Two People in Cockpit on U.S. Flights at all Times," [Online], Available: https://www.khon2.com/news/local-news/faa-requires-two-people-in-cockpit-on-u-s-flights-at-all-times_20180309115902834/1025802042, [Accessed July 3, 2018].

[7] J. S. Lipscomb and M. E. Pique, "Analog Input Device Physical Characteristics," *ACM SIGCHI,* 25(3):40-45, 1993.

[8] National Aeronautical and Space Administration, "The Intertropical Convergence Zone," [Online], Available: https://earthobservatory.nasa.gov/images/703/the-intertropical-convergence-zone, [Accessed April 3, 2019].

[9] F. Nobre, "A330 Flight Deck Systems and Briefing for Pilots," [Online], Available: https://www.slideshare.net/FernandoNobre1/a330-flight-deck-and-systems-briefing-for-pilots, [Accessed April 8, 2019].

[10] N. North and F. Zhang, *Encyclopedia of Atmospheric Sciences*., Academic Press, 2015.

[11] ProFlight Inc., "Pilot at Zero Altitude: A Brief History of Flight Simulators," [Online], Available: https://www.proflight.com/en/full-flight-simulatoren/historie.php, [Accessed April 17, 2018].

[12] A. Redei, E. Tumbusch, and J. Koberstein, "AVRATAR: A Virtual Environment for Puppet Animation," *Proceedings of the International Conference on Software Engineering and Data Engineering*, Las Vegas, NV, pp. 14-19, 2007.

[13] A. Sims, "Germanwings Crash," [Online], Available: https://www.independent.co.uk/news/world/europe/germanwings-crash-co-pilot-andreas-lubitzs-final-email-reveals-depression-and-fear-of-going-blind-a6915736.html, [Accessed July 7, 2018].

[14] B. Skoloff, "Equatorial Region Known for Massive Storms," [Online], Available: https://phys.org/news/2009-06-equatorial-region-massive-storms.html, [Accessed April 11, 2019].

[15] Suzo-Happ Inc., "Analog Flight Joystick with 5k Potentiometers & B5 Grip," [Online], Available: https://na.suzohapp.com/products/joysticks/95-0251-00. [Accessed June 10, 2018].

[16] Suzo-Happ Inc., "Speed Shifter," [Online], Available: https://na.suzohapp.com/products/driving_controls/50-8018-00, [Accessed June 10, 2018].

[17] Wikipedia, "Air Traffic Controller," [Online], Available: https://en.wikipedia.org/wiki/Air_traffic_controller, Accessed [July 15, 2018].

18] Wikipedia, "Flight Simulator," [Online]. Available: https://en.wikipedia.org/wiki/Flight_simulator, [Accessed April 8, 2019].

[19] J. Wise, "What Really Happened Aboard Air France 447," [Online] Available: https://www.popularmechanics.

com/flight/a3115/what-really-happened-aboard-air-france-447-6611877/, [Accessed April 11, 2019].

Alex Redei is an Assistant Professor at Central Michigan University working in the Department of Computer Science. He received his MS and PhD degrees in Computer Science and Engineering from the University of Nevada, Reno in 2013 and 2019, respectively. His interests include flight simulation, software engineering, and human-centered design. His research focuses on using flight simulators to experiment with new techniques for improving pilot training.

Sergiu Dascalu is a Professor in the Department of Computer Science and Engineering at the University of Nevada, Reno, USA, which he joined in 2002. He received in 1982 a Master's degree in Automatic Control and Computers from the Polytechnic Institute of Bucharest, Romania and, in 2001, a PhD in Computer Science from Dalhousie University, Halifax, NS, Canada. His main research interests are in software engineering, human computer interaction, and data science. He has published over 180 peer reviewed papers and has been involved in numerous projects funded by industrial companies as well as federal agencies such as NSF, NASA, and ONR. He has advised 10 PhD and over 40 Master students who graduated so far.

Frederick C. Harris, Jr. received his BS and MS degrees in Mathematics and Educational Administration from Bob Jones University, Greenville, SC, USA in 1986 and 1988, respectively. He then went on and received his MS and Ph.D. degrees in Computer Science from Clemson University, Clemson, SC, USA in 1991 and 1994 respectively. He is currently a Professor in the Department of Computer Science and Engineering and the Director of the High-Performance Computation and Visualization Lab at the University of Nevada, Reno, USA. He has published more than 200 peer-reviewed journal and conference papers along with several book chapters. His research interests are in parallel computation, computational neuroscience, computer graphics, and virtual reality. He is a Senior Member of the ACM, and a Senior Member of the International Society for Computers and their Applications (ISCA).

Instructions for Authors

The International Journal of Computers and Their Applications is published multiple times a year with the purpose of providing a forum for state-of-the-art developments and research in the theory and design of computers, as well as current innovative activities in the applications of computers. In contrast to other journals, this journal focuses on emerging computer technologies with emphasis on the applicability to real world problems. Current areas of particular interest include, but are not limited to: architecture, networks, intelligent systems, parallel and distributed computing, software and information engineering, and computer applications (e.g., engineering, medicine, business, education, etc.). All papers are subject to peer review before selection.

A. Procedure for Submission of a Technical Paper for Consideration:

1. Email your manuscript to the Editor-in-Chief, Dr. Fred Harris, Jr. Fred.Harris@sce.unr.edu.
2. Illustrations should be high quality (originals unnecessary).
3. Enclose a separate page for (or include in the email message) the preferred author and address for correspondence. Also, please include email, telephone, and fax information should further contact be needed.

B. Manuscript Style:

1. The text should be, **double-spaced** (12 point or larger), **single column** and **single-sided** on 8.5 X 11 inch pages.
2. An informative abstract of 100-250 words should be provided.
3. At least 5 keywords following the abstract describing the paper topics.
4. References (alphabetized by first author) should appear at the end of the paper, as follows: author(s), first initials followed by last name, title in quotation marks, periodical, volume, inclusive page numbers, month and year.
5. Figures should be captioned and referenced.

C. Submission of Accepted Manuscripts:

1. The final complete paper (with abstract, figures, tables, and keywords) satisfying Section B above in **MS Word format** should be submitted to the Editor-in-chief.
2. The submission may be on a CD/DVD, or as an email attachment(s). **The following electronic files should be included:**
    - Paper text (required)
    - Bios (required for each author). Integrate at the end of the paper.
    - Author Photos (jpeg files are required by the printer)
    - Figures, Tables, Illustrations. These may be integrated into the paper text file or provided separately (jpeg, MS Word, PowerPoint, eps). title of the paper.
3. Specify on the CD/DVD label or in the email the word processor and version used, along with the title of the paper.
4. Authors are asked to sign an ISCA copyright form (http://www.isca-hq.org/j-copyright.htm), indicating that they are transferring the copyright to ISCA or declaring the work to be government-sponsored work in the public domain. Also, letters of permission for inclusion of non-original materials are required.

Publication Charges:

After a manuscript has been accepted for publication, the author will be invoiced for publication charges of $50 **USD** per page (in the final IJCA two-column format) to cover part of the cost of publication. For ISCA members, $100 of publication charges will be waived if requested.

January 2014