Drunk User Interfaces: Using Smartphone-Based Human Performance Tests to Detect Inebriation
Sayna Parsi, Senior, Linguistics, Informatics
Mary Gates Scholar
Mentor: Jacob Wobbrock, The Information School

The standard quantitative method for assessing inebriation is to use a breathalyzer. However, breathalyzers are primarily owned by law enforcement and used only after a drunk individual is caught behind the wheel. Unlike breathalyzers, smartphones are one of the most ubiquitous technologies in today’s society. If smartphones could be used to reliably detect inebriation, they could be employed in ways to prevent drunk driving (e.g., by linking the smartphone to the car) or to incent good behavior (e.g., by lowering insurance for those who pass an app-based test before driving after 10 pm). We are running an ongoing study to examine whether or not challenging human performance tasks (e.g., typing, swiping, reacting) can be implemented on the smartphone to detect inebriation and prevent drunk individuals from getting behind the wheel. These tasks not only assess drunkenness from a performance perspective, but also from the perspective of the smartphone sensors.

Verifying the Chord Distributed Lookup Protocol
Ryan Hart (Ryan) Doenges, Senior, Mathematics (Comprehensive)
Mary Gates Scholar, UW Honors Program
Mentor: Zachary Tatlock, Computer Science and Engineering

In order to provide high availability and operate in unreliable environments, many critical applications are implemented as distributed systems. Unfortunately, the need to handle unreliable networks, machine crashes, and churn—the spontaneous arrival and departure of nodes to and from a network—has made these systems difficult to implement correctly. Recent work has proved the correctness of such implementations using proof assistant software. However, these frameworks do not support reasoning about churn. To address this limitation, we introduced support for implementing and verifying systems under churn to the Coq-based Verdi framework. With a model for the network in hand we could implement and begin to verify systems. However, for most systems under churn we need to prove punctuated safety properties. These properties allow a system to break some promises during and after churn, but stipulate that safe behavior be restored sometime after churn stops. Such a property is best proved by reasoning about infinitely long system executions, but this was impossible in Verdi and always awkward in Coq. We built tools for high-level reasoning about infinite sequences of events in Verdi using existing ideas from linear temporal logic in order to prove punctuated safety. To establish the efficacy of all these new tools, we have built an implementation of the Chord distributed lookup protocol and are currently proving it correct. Chord was designed to evenly distribute resources or tasks across a set of nodes even as some enter and leave the network. Recently, a study using a formal model of Chord’s protocol showed that certain kinds of node failures or joins could cause system failure. While a new specification has been developed which avoids these problems, it has not been implemented or fully verified. We are developing the first verified implementation of Chord.

High-Assurance Synthesis of Efficient Algorithms for Hierarchical Computation
Nathaniel Arfaa (Nate) Yazdani, Senior, Computer Science
UW Honors Program, Washington Research Foundation Fellow
Mentor: Rastislav Bodik, CSE

Today’s problems in fields like biomedical research require complex analysis of vast amounts of data in a short window of time, and the state of the art in computer science is not up to the challenge. The status-quo approach to such com-
Computationally demanding problems has been to design a custom algorithm, but this process is difficult, time-consuming, error-prone, expensive, and rarely yields the best solution. To overcome these challenges, computer science is undergoing a paradigm shift, from solving problems with individual algorithms to solving entire domains of problems with automatic generation, or synthesis, of algorithms. In particular, many problems boil down to scheduling or, in other words, planning when to compute what. For instance, a data center must plan how to divide a given complicated task across multiple machines. When this plan or schedule is created ahead of time, the process is known as static scheduling, and this schedule constitutes an algorithm specialized to the problem. Until now, static scheduling for every new problem domain has required either a complex, custom algorithm or a complex, custom model for a logic solver. Even in simple cases, these existing approaches are highly susceptible to human misinterpretation of requirements and necessitate complex, error-prone engineering to scale to real-world usage. The situation is worse still if the schedule needs to be optimized for efficiency. My research with Professor Bodik automates the creation of scalable tools for static scheduling from simple code to interpret the meaning of a candidate schedule. This work enables application of static scheduling to sophisticated problem domains while maintaining high confidence in schedule correctness. I applied this framework to create a static scheduling tool for hierarchical computation, supporting advanced and notoriously difficult techniques like parallel and incremental computation. With this tool, I created high-performance algorithms for data visualization and web browsers.

Incarnate: Using Programming Language Tools and Techniques to Improve the Efficiency, Accuracy, and Reliability of Commercial 3D Printing
Melissa Hovik, Senior, Human Ctr Des & Engr: Human-Computer Int, Computer Science
Washington Research Foundation Fellow
Mentor: Zachary Tatlock, Computer Science and Engineering

Additive manufacturing - the process behind 3D printing - has been growing in widespread use since the introduction of the first commercially-available desktop printer and later improvements in cost, quality, and accessibility. Expanding horizontally and vertically in the global market, the 3D printing industry has been one that not only appeals to hobbyists, but has demonstrated potential as a solution to a variety of challenges faced around the world today, including those in biotechnology, aerospace, and industrial engineering. Seeking to take advantage of the increasing demands in 3D printing technologies, a number of companies and open-source organizations are offering new types of 3D printers and associated software. However, the rapid growth and diversity of the products being released have resulted in a loss of universal standards. Often intended to be accessible and reliable regardless of a user’s technical background, 3D printing technologies have been known to require consumers to learn the more technical aspects of their printer, G-Code, slicing software, materials, etc. in order to configure their overall setup and reduce the prevalence of errors. Our project aims at using advanced parsing strategies, program optimization, and statistical analysis to provide a universal, easy-to-use tool that both analyzes and optimizes G-Code files prior to printing. By automating the common fixes users are required to implement in their G-Code files and identifying common factors in 3D prints (material warping, inadequate temperatures, speed of extruder heads, etc.), this tool is intended to be usable for any consumer, regardless of engineering or design experience. Ultimately, our goal is to help provide a universal solution to the lack of consistency between products, and help leverage the promising field of 3D printing to reach its potential as both a home essential and reliable solution in fields such as biotechnology and industrial engineering.

DDoS Attacks through the Internet of Things
Emilio Mena, Senior, Information Technology (Tacoma)
Mentor: Yan Bai, School of Engineering and Technology

The “Internet of Things”, or IoT, is a term that is used to describe a series of devices that connect to each other via a network. These devices can range from security cameras to robots and medical devices. They communicate in the same way that people use the Internet to communicate. Recently, due to the large explosion of information and devices out there, security is becoming a bigger issue. One of these issues is the presence of DDoS attacks. This research study analyzes the different types of DDoS attacks that affect networks and services. The research has been done through literature survey (peer-reviewed articles). This research also investigates the techniques used against various DDoS attacks and their limitations. The research results bring up a foundation for ongoing preventative techniques to tackle with DDoS attacks. We are currently developing and conducting an experiment study to improve the effectiveness of techniques against DDoS attacks in the IoT environment.

Block-Busters: Finding Features that Shine Light on the Darkweb
Christopher Zachary (Chris) Ross, Senior, Informatics: Data Science
Trevor Daniel (Trevor) Allen, Senior, Informatics: Data Science
Corey Leonard (Corey) Cole, Senior, Informatics
Andrew Louis (Drew) Before, Senior, Informatics
Mentor: Babak Salimi, Computer Science

The Darkweb is a section of the internet only accessible by special TOR browsers that anonymize users’ IP addresses. There are markets on the Darkweb where people buy and
sell illicit goods and services using Bitcoin as their currency. Both the Darkweb and the Bitcoin blockchain are considered anonymous and independent of each other. There are patterns that we believe present themselves similarly in both networks through the growth in popularity and activity in nodes of the networks. The first step necessary is to create “ personas” that represent the actual people behind the different users on all the Darkweb forums in order to create a network graph. We are then able to create the network graphs that we will be able to use the Page-Rank algorithm and other similar network centrality analysis to determine the growth of the two networks over time. All of this will allow us to determine which subsets of Bitcoin transactions are related to the Darkweb. After an initial thematic analysis our team was able to determine different interactions that would lead up to a sale of a product and thus a Bitcoin transaction. All of this work allows our team to start and determine exactly how well the security of the crypto currency and crypto markets work in conjunction with each other. By determining how easily we can connect these two groups of users we either show that the security measures used within the systems and communities are as secure as believed or that they can be defeated. This leaves an opening for future work into similar crypto markets and currencies that usual similar systems to that of Bitcoin.

**Securing Application Output of Augmented Reality Systems**

*Kimberly Christine Ruth, Sophomore, Computer Engineering*

*Mary Gates Scholar, UW Honors Program*

*Mentor: Franziska Roesner, Computer Science and Engineering*

*Mentor: Tadayoshi Kohno, Computer Science and Engineering*

*Mentor: Kiron Lebeck*

Augmented reality (AR) technologies are being actively developed, from head-mounted displays such as Microsoft’s HoloLens to AR-enabled car windshields by companies such as Hyundai. These emerging technologies, which integrate virtual content with the user’s perception of the real world, have the potential to greatly benefit their users. Applications running on an AR platform, however, may be malicious or malfunctioning, and so could harm to the user: for example, oculding safety-critical real world information, startling the user, or collecting sensitive data about the user’s surroundings. These risks may be further compounded by multiple applications running concurrently. Though previous work has examined the risks of applications receiving unfiltered sensor input, here we examine the risks of unregulated application output. We propose enabling the operating system of an AR platform to enforce safety and security properties of applications in dynamic, multi-app contexts. In this work, we prototype an *output policy module* in a simulated AR operating system. The module mediates applications’ visual output, filtering based on *policies* specified by the system, user, and applications: for instance, blocking rapid resizing of objects or occlusion of pedestrians crossing the road. We deploy the simulated operating system in several mock scenarios that reflect proposed everyday uses of AR, including home, office, and automotive settings. Our results demonstrate that the output policy module successfully blocks negative application behaviors while permitting desirable functionality. We show that our solution is both effective and practical: our output policy framework is expressive but does not impose an undue burden on developers, and the computational overhead for application performance is modest. Our solution thus presents a promising step toward designing more secure AR systems.

**Relating Dice to Voting Systems**

*Arthur Vartanyan, Senior, Mathematics*

*Savitik Agarwal, Sophomore, Computer Science (Data Science)*

*Jueyi Liu, Senior, Applied & Computational Mathematical Sciences (Scientific Computing & Numerical Algorithms)*

*Dorothy Truong, Junior, Applied & Computational Mathematical Sciences (Discrete Mathematics & Algorithms)*

*Mentor: Jonah Ostroff, Mathematics*

*Mentor: Lucas Van Meter, Mathematics*

For our research, we are focusing on dice and how we can relate them to voting systems. We want to see if we can create dice systems that mimic voting systems (with >2 candidates) and the traits that the systems possess. Two ideal traits of these voting systems are Pareto, which means that if everyone likes candidate A more than B, candidate B should lose; and IIA, which is if candidate A wins, then everyone mixes their ballot but keeps A’s rank relative to B the same, then A should still win. However there is Arrow’s Theorem, a central focus of our research, which states that if your system is Pareto and IIA, you must live in a dictatorship. One major topic of our research is trying to show whether or not this holds for our die systems. In order to accomplish this, we set out to define what these traits would be in terms of dice and how to translate dice outcomes to voter results, which we have successfully done for Pareto and IIA thus far. Our methods for this included testing certain systems that would reflect moving candidates/voters around, and testing these systems using some code we wrote for this purpose. Some other topics for our research are translating specific voting systems, such as popular vote, pairwise competitions, and point systems to dice, and showing whether or not these ideal traits hold for these translations, which we have successfully done for some. With this, we hope to gain a better understanding of these systems and how using something truly random, such as dice, has any reflection on the systems we have in place.