COMPARE AND CONTRAST SOFTWARE ARCHETECCTURE PATTERNS

HOW MUCH WORK IS REQUIRED TO MAKE SELF-DRIVING CARS ETHICAL

A Thesis Prospectus
In STS 4500
Presented to
The Faculty of the
School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

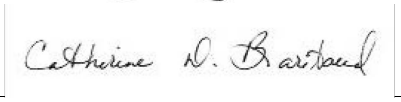
By
Aman Garg

October 31, 2019

Technical Project Team Members
Vineeth Gaddam, Sai Konuri

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Signed:  Date: 11/06/19

Approved:  Date: December 3, 2019
Catherine D. Baritaud, STS Division, Department of Engineering and Society

Approved: Nada Basit Date: 11/06/2019
Nada Basit, Department of Computer Science

While working in the computer science industry, it is imperative that when a software developer writes code, it should be written such that another individual can easily continue the work without wasting time trying to understand the organization of the codebase. Choosing a pattern to follow before development starts can help solve this issue (Richards 2015). These patterns, known as software architecture patterns, can dictate the way the codebase is set up. Software architecture patterns are a fairly new concept, but there are several that companies or developers may follow, and each one has its own benefits and drawbacks (Schmidt et al. 2013).

Companies, however, often do not select the pattern that best fits the type of application they are building. The first reason for this is usually that a company will try to incorporate too many different architectures into one code base and end up with a codebase lacking in coherent organization. Such a codebase such is almost as unorganized as a codebase without a pattern and can even be more confusing to follow because components are not where a newcomer would expect. Second, a company will often hear a term thrown around as a buzzword in the industry and adopt that pattern without truly giving it thought.

To provide an avenue for developers to better decide what architecture pattern meets their performance needs, a team of researchers will evaluate different architectures against different performance metrics. All code that the team develops will be made opensource so that it is available publicly. While the technical paper will focus research he STS paper will focus on the ethics of self-driving cars and how much work is required to make autonomous vehicles ethical. The paper will explore the implications of opensourcing the development of autonomous vehicles. These topics are coupled through the opensource nature of both topics. The technical research will be opensource, and the STS paper will explore some of the ethical implications of making a project opensource.

COMPARE AND CONTRAST SOFTWARE ARCHETECCTURE PATTERNS

Under the guidance of Assistant Professor Nada Basit, a professor of Computer Science at the University of Virginia, Vineeth Gaddam, Sai Konuri and I will be performing research on performance metrics of different software architecture patterns. This research will be conducted over the course of two semesters. As mentioned previously, a software architecture pattern is manner of organizing code to ensure that the codebase is easily understood by any new developer. An example of the Model View Controller (MVC) architecture is provided in Figure 1. The figure shows two examples of a browser's interaction with a webpage. While the diagram on the right requires many more steps for the same interaction, it is easier for a developer to follow the code when it is clearly organized as opposed to when it consists of a single interaction. The desire for cleanliness is the main reason an architecture pattern is necessary in software development. It is common practice to use architecture patterns, but the issue arises when developers are unsure of how to

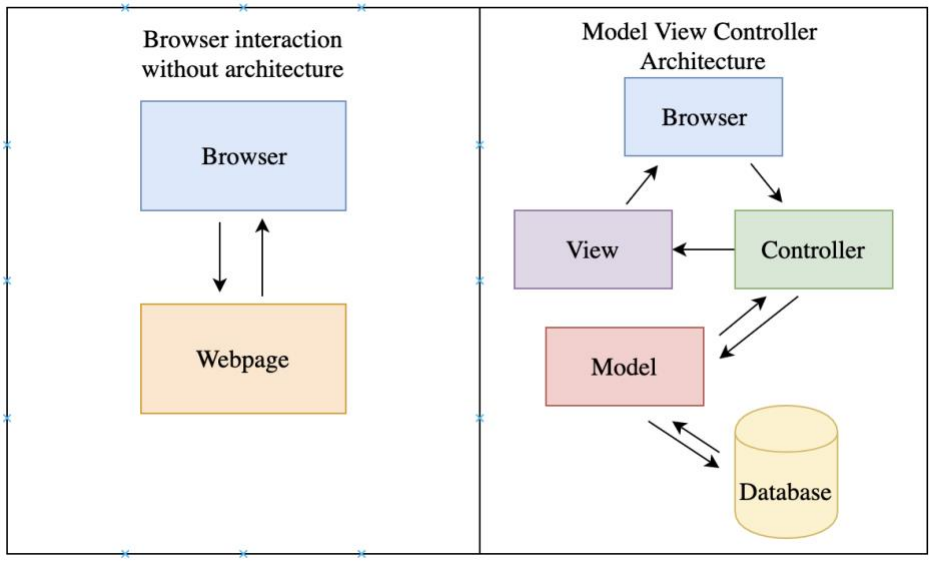


Figure 1: Non-Architecture Setup vs Model View Controller Setup: Left diagram shows the interaction of a browser and a webpage with no architecture pattern while right shows the same interaction split into the components of the MVC architecture (Garg 2019a).

select the proper architecture to choose in order to meet the performance needs required by their application. Performance requirements may not seem important at first glance, but they can negatively impact user experience of a product which, in turn, can hurt customer trust and brand loyalty.

The research team has two goals that it is trying to meet by the end of the research project. The first goal is to deliver a guide that developers can reference when deciding on an architecture to use for their application. The guide should be unbiased and refrain from making absolute assertions about using a specific pattern in certain cases. Instead, it will provide the reasoning and metrics used to make the recommendation. Presently, there are many blogs and articles on the Internet that describe different architecture patterns and different situations in which they should be used, but most fail to provide sufficient reasoning and proof for those recommendations. Additionally, the team will opensource all code used in the research. Opensourcing the code will enable developers to run their own tests on the applications so that they have the opportunity to measure metrics specific to their needs.

There are many software architectures that can be used, the team has selected a handful of architectures that are most commonly used in the software industry. The architectures being researched are MVC, Microservices, Event-Driven and 3-tier. These patterns differ only in how the code is organized. In the MVC architecture, the code is organized into sections containing the business logic, the data retrieval and the presentation logic. Meanwhile, in a microservices architecture each individual component is deployed as its own unit. An event-driven architecture is designed to wait for the state of the application to change, this change is the event, and then it will react to that change. The 3-tier architecture, similar to an MVC pattern, also splits the code into different layers but does not allow for tight coupling within its components. To measure the

performance of these architectures, a sample application will be developed for each one. After each application is built, it will be tested using JMeter, a tool used to aid in conducting load testing by simulating requests from multiple clients (Chopra and Kumar, 2017). After all applications have been tested, the results will be used to determine how the different patterns perform in different scenarios. The results will be compiled into a conference paper. The paper will contain an evaluation of the pros and cons of each architecture along with some high-level background on the design decisions and implementation details used to build the application.

Through this research, the team hopes to learn more about different architecture patterns but also to enable others to learn about those architecture patterns as well. To accomplish this, the team will present the results of the research in a compare-and-contrast conference-style paper. As mentioned previously, this document will entail the team's findings, how those results were found, and some general information about how the application was implemented for that specific architecture. Additionally, all source code for this research will be available publicly on GitHub. This will provide the opensource nature of the project and allow easier and wider access to the project and will enable others to continue on the research and potentially add more information on other architectures that out scope for the duration of the project.

HOW MUCH WORK IS REQUIRED TO MAKE SELF-DRIVING CARS ETHICAL

In the world of emerging technologies, artificial intelligence and autonomous cars are two of the most common buzzwords in topics of conversation and controversy. In the case of autonomous vehicles, people are not able to reach a consensus on the ethicality of such vehicles. There are extreme cases where the car's decisions result in a moral dilemma. One such case is

shown in Figure 2. In this case the car must kill either two humans or two cats, the ethical debate arises in regards to what the car should do in such cases. In the past few decades, cars have seemed to settle on a point of technological homeostasis. While companies work to improve areas like car safety, luxury, and performance, there have not been any major changes to the concept of an automobile. It is still completely under the control of the driver because it must be driven by a human. Self-driving cars would be a change to cars that is significant enough to cause unease due to peoples' fear of the unknown. It is a new technology that people are not accustomed to, and so there will be hesitation resulting from it. In order for such a change to occur and be accepted, people must first agree that autonomous cars have reached a standard which can be considered ethical.

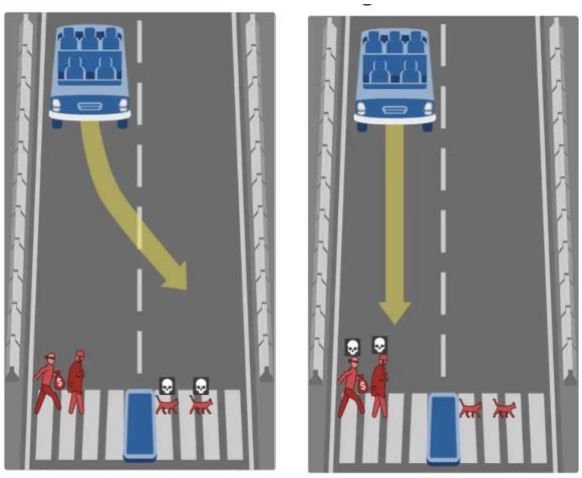


Figure 2: What should the self-driving car do?: this figure shows a scenario where the car must make a decision that will result in either two people dying or two cats dying (Johnson 2018).

This STS paper will attempt to provide an argument for why the development of autonomous vehicles should be opensource. It will set a benchmark that autonomous cars must reach in order to be considered ethical and will analyze how opensource development could help to reach this goal sooner.

To achieve the goal of determining the amount of work necessary in order to make self-driving cars ethical, it is necessary to define the benchmark of ethical. For the purposes of this paper, consider an autonomous vehicle be considered ethical when a majority of people and relevant social groups feel safe on the streets while the vehicle is driving. In research conducted by Bike PHG, it can be seen that about 21% of survey candidates said they felt very safe using Pittsburgh streets while autonomous vehicles were in use (Penmetsa et al., 2019). While this is not fully representative of the world population, or even the United States for that matter, it demonstrates that there is still a significant gap between the ethical self-driving cars and the current state. Penmetsa et al. also state that people seem to approve of self-driving cars more as they have had the chance to experience them (2019). So, after developers have agreed on a model, the general public can be eased into accepting such cars by slowly increasing the experiences that people get with the cars. This period will be gradual and will be handled at a pace dictated by companies producing self-driving cars and not by any debate or development. Thus, the period of introducing people to the cars will not be considered when determining how much work is required to make autonomous vehicles ethical.

SHOULD AUTONOMOUS VEHICLES DEVELOPMENT BE OPENSOURCE?

To bridge the gap between the current state and having ethical self-driving cars, the main necessity is to give people a chance to overcome their ethical reservations. In order to do that, the development of self-driving cars should be opensource. By opening the development to the public, the engineers are giving people the chance not only to become familiar with the work, but also to review it. Any developer who feels self-driving cars are unethical will be able to look at the development and reassess their position based on the code they are presented with. If it is the

case that they are still at unease, they will have the opportunity to either provide feedback to the developer or they could take the code and start working on it to try modify it to meet their ethical standards.

Eventually, it will be the case that any developer who has ethical reservations about autonomous vehicles has provided a stamp of approval or a sign of apathy. At that point, there will likely be many different designs for autonomous cars, but everyone's concerns about the ethics will be placated. From there, as is dictated by the Theory of Social Construction of Technology (SCOT), the design will start moving toward stabilization (Bijker and Pinch, 1987). In Figure 3, a SCOT model for autonomous vehicles shows the relevant social groups that will impact and be impacted by the development of self-driving cars. Each group will have different ethical needs and standards that they will put forth, and each group may end up presenting their own version of how an autonomous car should work. Developers will most likely be focused on the functionality of the cars, while private companies will most likely be focused on the production and release of the cars. Ethics researches will likely focus on the extreme cases such as the dilemma demonstrated in Figure 2 (Etzioni, 2017). Situations like the one presented in Figure 2 are considered extreme because they are outlier cases that are expected to be rare occurrences. Finally, the general public will probably not care about most of those topics as long as they feel safe. As such, that social group will likely care about the number and frequency of incidents that surround autonomous vehicles. Figure 3 on page 8 helps demonstrate the relationships that may exist between within the network of driverless cars. Figure 3 presents a diagram that depicts a network as presented in Deborah Johnson's article, The Social Construction of Technology (2005). The diagram includes autonomous vehicles and relevant social groups. It shows that all relevant groups have an impact on the development of driverless

cars and some groups exert influence over other groups as well. Private companies would be most concerned with getting the cars to a production stage quickly, so they will likely influence developers to work faster and the public to accept the car designs sooner. Ethics researchers, will influence the development of the cars but could potentially also influence the general public's opinion of autonomous vehicles. That connection has been marked with a dotted line because it could exist but does not seem like it would be as direct an influence as the other connections present in the diagram. Drivers and pedestrians would want companies to deliver a safe product so they would most likely cause development to be slower than the companies' desire. All of these different components within the network have influence other others and will impact the development of autonomous vehicles.

Out of the many options that are put forth based on input by different groups, the ones that are seen as the most ethical and functional will start to be selected while others are filtered out. When a design has stabilized it will start to be widely adopted and will start to be put into implementation. As was the case with the bicycle, the design of the self-driving car may not be the best possible design, it is the one that will be generally agreed on by different people (Bijker and Pinch, 1987). Once such a

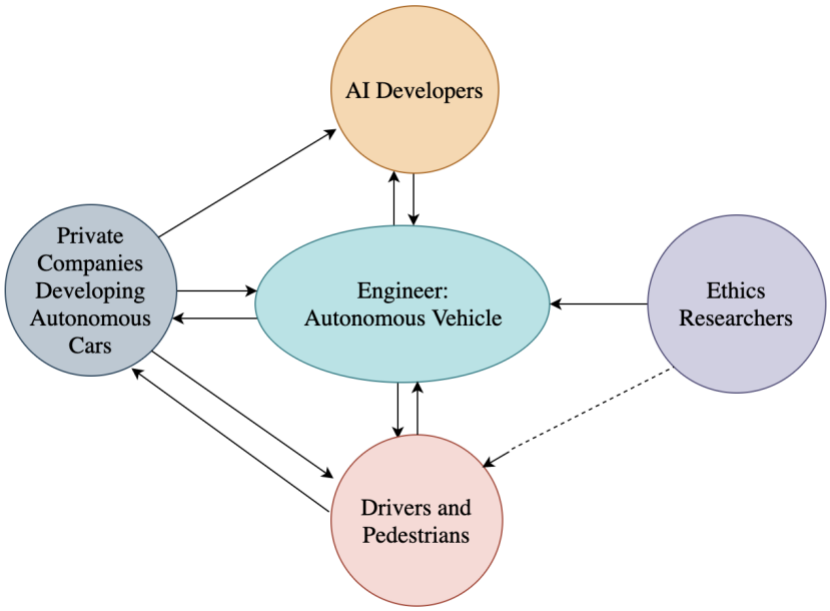


Figure 3: Actor Network model for Autonomous Cars: The different parts of the network with and against each other and exert influence over other groups (Garg, 2019).

point is reached, all relevant social groups with ethical concerns will have had the chance to provide their input on the development of autonomous cars. One of the large remaining social groups would be the common people who are not experts in technology but must share the roads with such cars. To acclimate the drivers and pedestrians with the introduction of autonomous vehicles, companies would have to start exposing them to the cars to increase comfort. As shown in the study conducted by Penmetsa et. al, people become more comfortable around driverless cars as they gain exposure (2019). So, once a design is in production, drivers and pedestrians can be eased into a road system shared between themselves and self-driving cars.

The process of reaching that point of having multiple designs is what must be analyzed in order to determine how much work is required to make self-driving cars ethical. This paper will look at case studies done on opensource software development in order to determine the advantages, disadvantages, and lifecycle of such development. One of the main benefits is that the people who contribute to opensource are invested in their work. Their role in developing the software is beneficial for them because they get to showcase their skills, but they are not being paid for that work (Feller et. al, 2006). Once they have developed the code, they also continue to maintain and support it. According to Fielding, Herbsleb, and Mockus, 90% of pull requests on opensource software are resolved within in 140 days (2000). That is a relatively quick time to resolve issues on code that nobody is being paid for, but it also highlights one of the challenges that opensource development faces. Because there is no deadline for the product to be delivered, the work could be delayed (AlMarzouq et. al, 2005). This paper will look at the benefits of opensource in order to gain an understanding of how opensource development of driverless cars can make them more ethical. After looking at the different case studies, the paper will aim to draw parallels between those studies and potential opensource development of autonomous

vehicles. Those parallels will hopefully provide a better sense of what opensource development of self-driving cars will look like in terms of complexity and time.

This STS paper will explore the implications of how much work is required to make self-driving cars ethical. It will take a deeper look at how much work is required to create a successful opensource product and it will look at what implications opensource development would have for the development of autonomous vehicles. To accomplish this, the paper will look at different case studies done on opensource software development and analyze the benefits and challenges that are discussed. All findings will then be presented them in scholarly article to present a case for why the development of self-driving cars should be opensource.

WORKS CITED

Albrechtslund, A. (2007). Ethics and technology design. *Ethics and Information Technology*, 9(1), 63-72. doi: 10.1007/s10676-006-9129-8

AlMarzouq, M., Grover, V., Rong, G., Zheng, L. (2005). Open source: Concepts, benefits, and challenges. *Communications of the Association for Information Systems*, 16, 10.17705/1CAIS.01637

Bijker, W. E., Pinch, T. J. (1987). The social construction of facts and artifacts: Or how the sociology of science and technology might benefit each other. In Hughes, T. P., Pinch, T. J., Bijker, W. E. (Ed.), *The social construction of technological systems*. Cambridge, MA: The MIT Press

Chopra, V., Kumar, V. (2017) Design & implementation of Jmeter framework for performance comparison in Ruby, PHP, & Python web applications. *International Journal of Innovative Research in Science, Engineering and Technology*, 6, 16447-16460, doi:10.15680/IJIRSET.2017.0608218

Feller, J., Fitzgerald, B., Lakhani, K., Hissam S., Scacchi, W. (2006). Understanding Free/OpenSource SoftwareDevelopment Processes. ITALICS Software process improvement and practice, 11(2), 95-103, doi:10.1002/spip.255

Fielding, R. T., Herbsleb, J., Mockus, A. (2000). *Proceedings - International Conference on Software Engineering*. Limerick, Ireland: IEEE Explore

Garg, A. (2019a). [Figure 1]. Prospectus (Unpublished undergraduate thesis). School of Enginering and Applied Science. University of Virginia. Charlottesville, VA.

Garg, A. (2019b). [Figure 3]. Prospectus (Unpublished undergraduate thesis). School of Enginering and Applied Science. University of Virginia. Charlottesville, VA.

Etzioni, A. (2017). Incorporating Ethics into Artificial Intelligence. *The Journal of Ethics,* 21, 1-3, doi.org/10.1007/s10892-017-9252-2

Goodall, N. (2016, June 7). Can you program ethics into a self-driving car? *IEEE Spectrum, 53,* 25-28. doi:10.1109/MSPEC.2016.7473149

Johnson, C. Y. (2018, October 24) Self-driving cars will have to decide who should live and who should die. Here's who humans would kill. *The Washington Post*. Retrieved from https://www.washingtonpost.com

Johnson, D. G. (2005). The social construction of technology. In Mitcham, C.*, Encyclopedia of science, technology, and ethics*. Detroid, Michigan: Macmillan Reference USA

Praveena, P., Emmanuel, K. A., Dustin, W., Teng W., Jones, S. L. (2019). Perceptions and expectations of autonomous vehicles – a snapshot of vulnerable road user opinion. *Technological Forecasting and Social Change, 143*, 9-13

Schmidt, D. C., Stal, M., Rohnert, H., Buschmann, F. (2013). *Pattern-Oriented Software Architecture, Patterns for Concurrent and Networked Objects*. Hoboken, NJ: Wiley

Richards, M. (2015). *Software Architecture Patterns* (H. Scherer & C. Lobner, Eds.). Sebastopol, California: O'Reilly Media Inc.