

WASP Software Engineering - Assignment 2

Maarten Jensen - Umeå University

September 29, 2022

1 Introduction

The research field I'm doing my PhD in is called social simulation. This field is concerned with understanding and creating models of human behavior to study the behavior of humans. An intuitive example of social simulation is the ASSOCC¹ model which is a coronavirus model made by our research group. This model replicated a city and allowed the user to study the effects of for example closing schools, closing shops or wearing mouth masks on the spread of the corona virus. Interesting insights can be gained by evaluating the model, e.g. What happens when the schools are closed? Do the people just neatly go to their homes and study there (possibly slowing down the spread of the virus) or will they meet their friends at their homes (possibly increasing the spread of the virus)?

One problem with the model however was that it became so big, including so many aspects that the humans needed a lot of time to make a decision which made the model slow. This is where my research topic comes in, which is the study of context in decision making for social simulation. I'm working on understanding and formalizing context to make decision making more efficient.

2 Challenges in behavioral software engineering

Making social simulations requires modeling human behavior in computer code. This creates a big challenge, that is: how do you transform a non-binary thing such as human behavior to a computer program. Much of sociological and psychological research give quantitative results on how humans make decisions. This quantitative research needs to be formalized

¹<https://simassocc.org/>

before it can be implemented. There is also some qualitative work from behavioral studies but the problem is that this is very narrowed down to single scenarios: E.g. questions such as would you rather get A) 50 euros certainly or B) 60 euros with 90% chance or lose 10 euros with 10% chance (for official examples see²). The probabilities of answering either A or B can be directly implemented in a model, however this behavioral model would then only work at representing this specific decision making in the lab session. However this would not translate to other decisions people have to make in daily life. In short abstractions always have to be made when expressing human behavior in software which is one of the main challenges in social simulation.

This complexity in modeling human behavior gives such systems different requirements than traditional IT systems. While it is relatively easy to set requirements for an IT system such as a thermostat: it needs for example 1) a display with the temperature, 2) a temperature control button, and 3) a power button. These traditional requirements are more concrete than the requirements for a model of human behavior. A typical requirement could be: The humans in the simulation should behave human like. Then it becomes the question what is human like? How do people actually make decisions? By analyzing literature and making a conceptual model these requirements can become more specific however this process is never straight forward. And as is the case with traditional IT systems, but even more in social simulation, more requirements popup when extending the model. While in the last decades traditional IT started to move to agile development³ instead of waterfall development methodology⁴. Using agile development method is definitely a must for modeling human behavior as the requirements and the system are hardly able to be made beforehand.

In social simulation many models are build on the fly based on the specific research question. There are many different models that are all specific, but therefore also almost always thrown away after the research project is finished. Hardly ever models (the actual programming code) are reused. A software aspect that could help here is extendability of a model. If a model is extendable may be picked up for a different research question and then be extended. Think of a simple epidemiologist focused coronavirus model, if it is extendible it would be easy to add components e.g. social components to make the behavior of people more realistic. One of the problem with

²Daniel, K. (2017). Thinking, fast and slow.

³<https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development>

⁴https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

extendibility is often that when a model gets expanded deliberation cycle (the decision making part of the humans in the simulation) becomes slow. As more and more aspects get added the deliberation cycle will be filled with a lot of information that is often completely considered. Imagine every time when deciding which food to buy, one also thinks about which house to buy, which car to buy, what food to eat on every day of the year. This problem is where my research project comes in. I'm using context to make an efficient behavioral model that is able to only take into account what it needs to take into account, e.g. making a decision about which food to buy, then only the food that is relevant is taken into account (and nothing of the previously mentioned aspects). This technique could make simulations more extendible as more information then does not directly mean a slow model.

3 Future trends of software engineering

I think we can see an effect in programming/research due to the increased potential of libraries. A couple of decades ago there was Assembly a very primary programming language where you directly specify what registers do. In Assembly it would even be hard and a time consuming task to make a simple calculator that can only perform additions. However nowadays it is possible to do all kinds of things with programming languages with relative ease. For example using python and its libraries, with a single/couple lines of codes one can make a machine learning algorithm or perform video analysis.

The effect on science with this increase of libraries is maybe two fold. On one hand software will be used by more and more researchers also from the non-computational science fields, as more and more advanced libraries become available and coding becomes 'easier'. While other researchers, with the growth of the number of research engineers will actually make more of these libraries. In library creation there is really a merge of software development and science. I therefore think that software development will co-exist along science, at least for the coming decades. However I can also imagine that the use of libraries and software will be larger and larger as they make it possible to do complex things with low effort. Computer science is already being used in biology, medical field, and I think the use will just increase.

When looking at the field of social simulation there are more and more libraries available however there are not so many proper frameworks for human decision making, because of the previously mentioned difficulties. However I hope during my PhD to make a contribution to this.