Research Plan Synthetic Data Generation for Virtual Society

Dr. Raymond G. Hoogendoorn, Sjef van Leeuwen & Prof. Dr. Lampros Stergioulas

August 20, 2020

1 Introduction

Virtual Society is an aid in order to test software using life-like simulations. In comparison with other industries, digitalization has started much slower within our Dutch government. Citizens and companies expect that information from the government is instantaneously available and easily accessible (Van Leeuwen, 2020).

Governments have many reasons to meet these expectations through investing, time, effort and money in digital transformation for the public sector. Principles of automation and especially algorithms among which Artificial Intelligence and simulations are indispensable in order to achieve these digital transformations.

Primarily Virtual Society is being developed for designers within software engineering and data science. Virtual Society offers simulations within a chain of automated systems, through providing these systems with life events (Van Leeuwen, 2020). These events are modelled within the simulation and executes these. Examples of these life events are births and deaths, leaving the parental home, starting a family, etc.

It is crucial to note that this information is completely fabricated: it is synthetic data generated by the simulation. Several elements within the simulation are based on statistics. For example, the CBS (see also www.cbs.nl) is an important resource for Virtual Society. Other elements which provide the foundation for the simulation are based on complex models, such as machine learning models, which can approximate the behavior of automated systems within the actual system being a part of the simulation.

Since Virtual Society generates data through its simulations, data cannot be traced back to real personal information. The upside of using synthetic data is that the privacy of citizens is not affected. On top of this, synthetic data is much more accessible for a broader audience.

Simulations are a replication (digital twin) of reality. Simulation is a dynamic and complex process. Through Virtual Society we achieve the aforementioned based on life events. This simulation model determines through the use of rules derived from algorithms how the simulation progresses.

During the simulation runs we obtain insights into how the situation changes during the simulation time. Many modelling approaches towards life events can be distinguished in literature. For example, the probability of births and deaths can be modelled. During the life cycle of a human being several life events can take place, among which the birth of children, following which specific situations arise within a family. When these are modelled using real statistics, life events arise which replicate reality. The consecutive occurrence of life events create a complex model with many interactions, without actually modelling this complexity. In this sense, Virtual Society aims at designing simulations able to replicate the development of populations.

In Baines et al. (2004) it was already discovered that the results of simulation studies, when human factors are accounted for, show a difference of 35% compared to more traditional research when human factors are not accounted for. Within Virtual Society simulations are set up using Discrete Event Simulation (DES).DES models the system as a discrete sequence of events in time.

Each event takes place at a certain moment in time and marks a change in the status of the system. The subset of DES for simulations within Virtual Society is based on the modelling of processes. Many automated systems within the government are based on process automation. Between consecutive

events it is assumed that no changes occur in the system (next-event-time progression). This leads to a substantial reduction in computation time compared to for example agent-based modelling.

2 Research objectives and questions

The research objective of the current project is to create a Discrete Event Simulation tool able to accurately replicate reality in terms of births and deaths of the Dutch population using external data for instance obtained from the CBS.

In order to achieve this, the following research questions need be answered:

1. Which factors can be distinguished from the (scientific) literature as possible determinants influencing the life-events births and deaths?
2. What are the goals and objectives that can be formulated as simple Boolean or numeric decisions?
3. What conceptual model best represents the Discrete Event Simulation model to be constructed?
4. Which input models using external data can best be used in order to model births and deaths using Discrete Event Simulation?
5. How can the aforementioned conceptual model and input models best be converted into a computational model?
6. Which methods can be used in order to validate the Discrete Event Simulation model? To what extend is the simulation model valid?

3 Research methods

Discrete event simulation modeling is the process of codifying behaviour of a complex system as an ordered sequence of well-defined events. In this context an event (or in this case life event) comprises a specific change in the system's state at th e specific moment in time

Discrete-event simulation models are both stochastic and dynamic with the discrete event property that the state of the system changes value at discrete times only (Hill, 2007). Discrete-event simulations are therefore characterized by three different properties:

- stochastic: at least some of the system variables are random
- dynamic: the evolution of the system state variables is important
- discrete event: significant changes in the system state variables are associated with events that occur at discrete time instances only

From the aforementioned it can be summarized that a good Discrete Event Simulation model includes the following characteristics:

- predetermined starting and ending points which can be discrete events or instants in time
- a method of keeping track of the time that has elapsed since the process began
- a list of discrete events that have occurred since the process began
- a graphical, statistical or tabular record of the function for which DES is currently engaged.

It is clear that the process of developing a discrete-event simulation model cannot be reduced to a simple sequential algorithm. In general however, the following steps need to be taken in order to develop such a simulation model (see also Hill (2007)).

1. **Goals and objectives.** We need to start with outlining the goals and objectives of analysis once the system of interest has been identified. These goals and objectives are best formulated as simple Boolean decisions or numeric decisions. An example of the former is the question whether an additional queuing network service node should be added, while an example of the latter is how man y parallel servers are necessary in order to provide satisfactory performance in a multi-server queuing system. Without the formulation of the goals and objectives, the next steps are useless.

2. **Conceptual model.** Based on the goals and objectives a conceptual model needs to be build. What are the state variables, how are they interrelated and to what extend are these variables dynamic? How comprehensive should the model be? What is the relative importance of the state identified state variables? Which variables can be ignored?

3. **Specification model.** The formulated conceptual model should then be converted into a specification model. This step involves the collection and analysis of data in order to provide the input models that drive the simulation. If these data are not present, the input models must be constructed in an ad hoc manner using stochastic models believed to be representative.

4. **Computational model.** The next step is the conversion of the specification model into a computational model. At this point, a fundamental choice should be made. Are we going to use a general-purpose programming language (e.g., Python, Java, etc.) or a special-purpose simulation language (e.g., SimuLink)?

5. **Verification.** The computational model should be consistent with the specification model. Was the model implemented correctly?

6. **Validation.** In this phase the question needs be answered whether the computational model is consistent with the actual system. Was the right model built? A popular (non-statistical) Turing-like method for model validation is to place actual system output alongside similarly formatted output from the computational model. This output is then examined by an expert familiar with the system. Model validation is attained when the expert is not able to determine which is the model output and which is the real thing.

4 Planning

In general the project is divided into three different phases:

1. **Exploration**: in this phase a literature review needs to be performed. Furthermore in this phase possible data sources are identified. Based on the results of this work, a conceptual model needs to be constructed which can be used for the transformation towards a computational model. In sum, the following elements need to be completed:

    - problem description (definition of the context and a broad description of the problem)
    - (re)formulation of the research objectives and relevance of the study
    - (re)formulation of the research questions and formulation (if desired and possible) of hypotheses
    - more detailed account of the chosen research approach (step-by-step description of the process, description of the chosen techniques, description of the evaluation methods)
    - extensive literature review; this needs to consist of a report on what is happening. Furthermore, a brief overview of the history of the topic needs be be included and existing approaches need to be discussed.

2. **Development**: in this phase the actual computational model is developed (including the input models using external data). Also a random number generation method is chosen (e.g., Monte Carlo).

3. **Validation**: int this phase the model needs to be validated. Using the chosen validation methods an account of the for instance internal and external validity is provided.

5 Supervision

The students are supervised by prof. dr. Lampros Stergioulas, dr. Sjef van Leeuwen and dr. Raymond Hoogendoorn (daily supervisor). Before the actual start of the project, the following aspects need to be resolved:

- clear and detailed tasks need to be described
- dependencies between the different tasks need to be identified
- timelines, milestones and deadlines need to be agreed upon

The separate tasks to be executed will be monitored closely by the supervisors. The progress needs to be made explicit for every separate task. Divergence from the initial objective or approach needs to be identified as quickly as possible.

Issues and problems that arise need to be addressed as quickly as possible and, together with the supervisors be re-mediated. The students are then responsible for the execution of the chosen remedial action. In this sense, it is needed to plan for potential failure.

References

Baines, T., Mason, S., Siebers, P. O., & Ladbrook, J. (2004). Humans: The missing link in manufacturing simulation? In *Simulation modelling practice and theory.* doi: 10.1016/S1569-190X(03)00094-7

Hill, R. (2007). Discrete-Event Simulation: A First Course. *Journal of Simulation*. doi: 10.1057/palgrave.jos.4250012

Van Leeuwen, S. (2020). *Virtual Society: Open Social Innovation* (Tech. Rep.).