Simulation and Long Term (Military) Manpower Planning:

a Custom Generic Simulation Tool

Johan Van Kerckhoven (corresponding author)
Oussama Mazari-Abdessameud, Filip Van Utterbeeck

Dept. of Mathematics (MWMW)
Koninklijke Militaire School
Brussels, Belgium
Email: `johan.vankerckhoven@mail.be`

***Abstract*—This article presents a discrete event simulation tool/engine, which we are currently developing in Julia – making use of the SimJulia package – to assist with long term manpower management of our organization. Although this tool is inspired by one particular case, our ultimate goal is to make this tool as generic and configurable as possible, while also being easy to use. We go into some detail about the inner workings of the tool, highlight its current capabilities, and elaborate on the technologies used to develop it and on how users, both expert and casual, can interact with the tool. At each point, we highlight the features that are still in development, and the points where the tool requires improvement.**

***Keywords*–Manpower simulation; manpower planning; discrete event simulation; in development.**

I. INTRODUCTION

In this day and age, proper human resource management becomes more and more important for organizations of all kinds. First of all, managers need to balance the requirements of their organization with the needs and wants of their employees. In addition to that, they must be able to make well-informed planning decisions to ensure the smooth (future) operation of the organization.

These planning decisions cover different time horizons. Long term planning, which is the focus of this article, deals with hiring and promotion strategies to meet the organizations goals without alienating the employees, or compensating for and adjusting to changing legal constraints, for example an increase in the minimum retirement age. Short term workforce planning on the other hand covers among others, setting up employee rosters such that the workload is appropriately distributed over the employees, taking into account shift preferences of particular employees, absences due to time off or illness, and legal constraints.

These long term planning decisions are too important to be left solely to the managers gut feeling. Instead, they can choose to employ one or several mathematical models to provide valuable insights towards making appropriate decisions. Managers have a wide range of models available to them for this task, including Markov models (f.ex. [1] and [2]), optimization models using mathematical programming (f.ex. [3]), simulation models (f.ex. [4] and [5]), and system dynamics models (f.ex. [6]). Each of these approaches can be used to model the transient states or the steady states of the system, and every approach has its particular advantages, and drawbacks, for these tasks.

Due to an upcoming reorganization, the Belgian military is currently faced with such a long term planning problem, and the choice was made to use a simulation-driven stochastic optimization routine. This would allow the user to capture as many of the peculiarities of the organization as desired. In addition to the flexibility of the model, several other factors were taken into account to influence the choice of which particular tool to use. These factors included computation speed, financial cost, and user friendliness, by which we mean that it can be used relatively easily by non-experts. Of the investigated solutions that were available at the time, none satisfied these criteria to a sufficient degree. Hence, the decision was made to develop our own solution in the programming Julia [7], based on the SimJulia package [8], with the express goal of making the tool as flexible, generic, and user-friendly as possible.

This article presents the current state of the tool, which is still in very active development. In particular, we are developing a library of functions, which will allow the user to set up an entire workforce organization, or a part thereof, with all its HR policies (recruitment, promotion/transfer, retirement) in place. The user can then perform a simulation run to see how the composition of the workforce will evolve over time, taking the policies into account, and to generate easily digestible reports on the evolution (of any desired part) of the workforce. These simulation results and reports can then be used as a basis for a stochastic optimization routine to find the optimal set of HR policies for a particular organizational goal. However, the optimization part falls outside the scope of this article. Parallel to the simulator engine, we are also developing a user-friendly way of configuring a simulation, to allow non-expert users to set up and run a simulation in a relatively straightforward manner.

The article is structured in the following manner. In Section II, we describe the setup of the simulation model, where we highlight the current, as of writing, state of the model. The next section, Section III, covers the software and technologies used to implement the simulation tool, as well as the reasoning behind it. In Section IV, we explain how the simulation tool can be used, where the focus will be on the “non-expert” user. In each of these sections we also mention what features of the tool are still planned. Section V gives a small example of the present capabilities of the simulation tool. Finally, we briefly summarize in Section VI.

ISBN: 978-1-61208-672-9

II. SETUP OF THE SIMULATION MODEL

We have opted to build the simulation model as a discrete event simulation where we model each individual personnel member with their own attributes in the system. The simulation then allows each person to move through the system and their attributes to change based on the organization's policies, legal requirements included, as provided by the user.

The rest of the section goes into more detail on each of the three flows that a person in the simulation will undergo:

1) Flow in – recruitment
2) Flow through/transitions – transfers, promotions, etc.
3) Flow out – retirement, resignation, other forms of attrition

Please note that, as the tool is still under active development, and in a rather early stage at that, the features described below are not set in stone, and that the actual approach towards implementing these can still change considerably.

A. Entering the system

Persons can enter the organization in two ways. First of all, they can be recruited into the organization throughout the simulation run time; alternatively, the simulation can start with an existing population (start from a snapshot).

1) Recruitment: Recruitment happens in periodic recruitment cycles throughout the simulation: every few time steps a number of people will be generated and entered into the organization. Hence, instant recruitment the moment a person is needed is not allowed. The main reason for this choice is that in real life applications, it is impossible to recruit a person at any given moment. This is in part due to the organization's policies, and in part due to practical considerations: if recruitment at any moment were permitted, it'd mean that every recruit would (potentially) need a personal training schedule and trainer.

At the start of each cycle, the number of people to recruit is determined. This can either be a fixed number, a number drawn from a user provided distribution, or a number determined by the needs of the system, where this number can vary between certain bounds.

When a person is generated in the system, all their attributes are initialized to appropriate values, where those attributes and the initial distribution of those values are provided by the user. This includes the person's age upon recruitment.

Currently, implementation of the recruitment process has been mostly completed. That is to say, it is possible to set up multiple recruitment schemes as described above. However, all those recruitment schemes will use the same distributions for the initial values of the generated persons: differentiation between recruitment schemes is not possible yet.

2) Snapshot: The snapshot, i.e. starting from an inital population, has not been implemented yet.

B. Transitions

Transitions are nothing more than a change of a state that a person in the simulation can have. We have defined a state as a collection of attributes that must have specific values. Note that these states need not be mutually exclusive. For example: a person in a military organization can be an officer, be part of the ground forces, etc. We also aim at allowing states to have their own specific attributes, such as name (already implemented), and the number of people preferred to be in this state (to be implemented).

Just as with recruitment, transitions are assumed to happen periodically, where each transition has its own cycle. Currently, we only allow that these transitions happen automatically after a set period of time. In the near future, we wish to implement that these transitions have a certain probability of occurring, and that each person has a set number of chances of undergoing the transition. In addition, we will also implement that each transition has a set of extra conditions, not expressed in the initial state, that need to be satisfied before the transition can take place. For example, to become a helicopter pilot, a person must be in the air force, and have a certain rank (state), but they must also have a certificate that they can actually operate a helicopter and not be older than a certain age (extra conditions).

C. Leaving the system

Similar to recruitment, there are two ways to leave the system: retirement and attrition, where attrition covers everything from voluntary resignation to forced resignation due to medical reasons.

1) Retirement: At present, retirement occurs on a periodic schedule, similar to recruitment, and is defined by age and career length. If a person reaches a certain age, they have to retire, and similar if their career reaches a certain length. Currently, there is one retirement schedule, which is valid for all people in the simulation. In a later stage, we will allow for different retirement policies for different groups of people.

2) Attrition: Attrition is defined as a probability, per period, that a person will leave the organization for any reason except mandatory retirement. If a person will leave the organization during the next attrition period, an exponential distribution is used to determine the exact time they leave. This attrition probability can change during the lifetime of a person, for example, people will leave the organization much more frequently in the early stages of their career. As with retirement, it is not yet possible to configure different attrition schemes for different groups of people, but this is, again, a planned feature.

III. TECHNICAL CHOICES AND MOTIVATION

We have chosen to develop the software in the programming language Julia [7], a high-level, high-performance, open source programming language for numerical computing which is developed at MIT. In particular, we rely on its SimJulia package [8] for setting up discrete event simulations. This package manages simulated time, the event queue etc. while our own package builds on top of that.

We chose to use the Julia language due to the in house experience available, and due to its ease of use and computational speed, which in many cases rivals that of C [9]. In addition to this, benchmark tests by the developer of SimJulia have consistently shown that this package is computationally the most performant tool to drive a discrete event simulation, outpacing other implementations sometimes by an order of magnitude. Considering the intended application of the manpower simulation tool and the size of the organization for which it is intended, computational performance is essential. The final reason for choosing Julia is its open source character, which allows us to determine what's going on behind the scenes, and tailor the optimization to it.

To store the data generated throughout the simulation, we rely on an SQLite [10] database, which we access through the SQLite package [11] of Julia. This is an implementation of SQL which does not rely on a database server and client-server interactions. Instead, it uses a file saved locally, which means we do not have to rely on the availability of the connection between the machine which runs the Julia code, and the server housing the database of simulation results. An additional benefit is of course speed (again), as local files are generally available nearly instantaneously, especially when using a solid state drive. A roadblock here was the large number of SQLite calls, resulting in a large number of time-intensive file operations. However, we have circumvented this issue by keeping the database in memory and only saving it to file at certain points throughout the simulation.

IV. USING THE SIMULATION TOOL

This section gives a brief overview of the ways the user can configure a manpower simulation in this tool, and how they can extract information once a simulation is completed.

A. *Configuration*

The tool is being developed in such a way that every aspect of a manpower simulation can either be configured from a more-or-less user friendly Excel file, or from Julia itself through the use of various configuration functions. The first configuration method is intended for the non-expert user who wishes to evaluate a certain scenario (set of manpower management policies), whereas the second method, configuring a simulation straight from Julia, is more intended for developers who wish to use this simulation tool as a driver for their own manpower planning tools, for example an optimization tool.

B. *Output*

As indicated above, the results of the simulation are stored in an SQLite database. This database holds a record of all the entities in the simulation (past and present), as well as all the state changes and changes to entity attribute that have occurred during this simulation.

To extract intelligible information from this database, we provide several functions which generate simple reports on the following:

- Size of the population, or a specified subpopulation, at a regular time grid. These counts can be broken down by the values of a specified attribute, for example a breakdown of all air force personnel by rank category (enlisted, non com, officer).
- Flux in/out of the population or a specified subpopulation, on regular time intervals. By flux in we mean that a person is not part of the (sub)population at the start of an interval, and they are part of it at the end of that interval, similarly for flux out. These fluxes can either be broken down by source/target state, or by transition.

These reports are produced in an easily accessible Julia data structure, and can be used as they are by developers for further processing. Alternatively, the user can request a variety of plots to be compiled based on these reports, where the request is made through the Excel file:

- Line plot of the totals;
- Line plot of the breakdowns of the population and/or the in/out fluxes;
- Stacked area plots of the breakdowns;
- Percentage area plots of the breakdowns.

These plots allow the user to see, at a glance, the workforce's composition, and to gauge whether a particular set of HR policies leads to a stable organization or a constantly varying situation.

TABLE I. ATTRIBUTES AND INITIAL DISTRIBUTION.

Grade cat	prob	BDL	prob
Vol	60%	Limited	100%
NonCom	30%	Appointed	0%
Off	10%		

TABLE II. TRANSITIONS.

Start – Target	Time (y)
Vol + Limited – Vol + Appointed	6y
NonCom + Limited – NonCom + Appointed	8y
Off + Limited – Off + Appointed	10y
Vol + Appointed – NonCom + Appointed	12y
NonCom + Appointed – Off + Appointed	12y

Finally, it is also possible to request a graphical representation of the system or part thereof, such that the user can get an idea of how people can move through this system. With this representation it is of course important to keep in mind that entities in the simulation can be in several states at the same time.

Towards the future, we will also allow the user to export this graphical representation to graphML such that it can be consulted in greater detail through the use of software, such as yEd [12], or to other common graph representation formats.

V. EXAMPLE

In this section, we provide a small, highly unrealistic(!), example of the current capabilities of the tool. The example itself is described in Section V-A, and the results are summarized in Section V-B.

A. *Description*

The simulation settings that we use in the example assume an organization with a target size of 25,000 people, simulated over 50 years, and two recruitment schemes. The first scheme recruits a random number of people, between 400 and 500, every 12 months from the start of the simulation, and the second scheme recruits anywhere between 300 and 500 people depending on how many are needed still, every 12 months starting 6 months into the simulation. Ages are chosen randomly between 18 and 25 years.

We consider two attributes to be of importance, and these are listed in Table I along with the probability of each value on person recruitment. 'Grade cat' is the grade category, and BDL means limited duration, implying a person is not guaranteed to be allowed to stay in the organization after completing a sort of "trial" period. The possible states are all the possible combinations of these two attributes, and the only transitions allowed are the ones listed in Table II.

Persons in the simulation retire at 67 years of age or after a 45 year long career, and they have a 4% chance per year to resign during the first 4 years of their career, 2% until 20 years of tenure, and 1% thereafter.

B. *Results*

The runtime of the simulation is around 65 seconds with observed variation of ± 3 seconds between simulation runs on a single core of a 2.70 GHz (max. turbo 3.70 GHz) processor. A more detailed breakdown of the computation time, listing what percentage of the time is used for each task, is not available at present.

The breakdown of the population (stacked + percentage) is shown in Figure 1. From the top plot, which shows the

ISBN: 978-1-61208-672-9

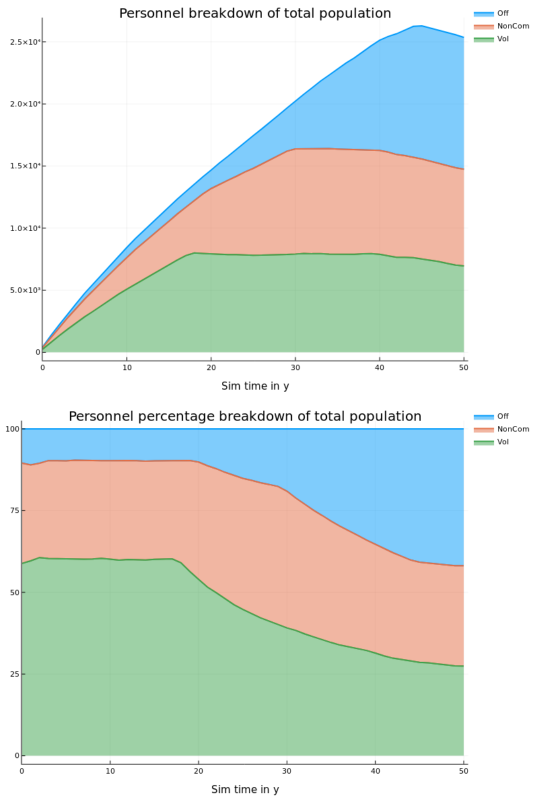

Figure 1. Breakdown of population by rank category.

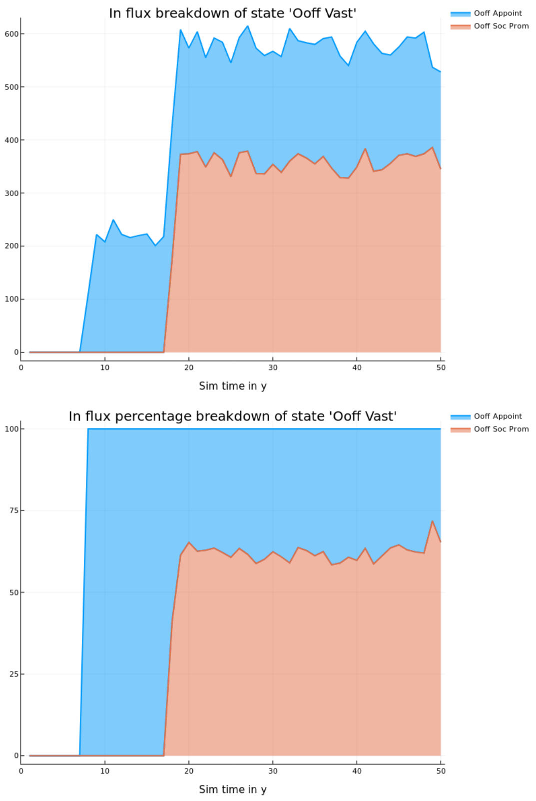

Figure 2. Breakdown of influx into the state NonCom + Appointed.

breakdown per grade category, we observe that the total population steadily rises to a level somewhat above 25,000 before slowly decreasing towards it. It we would have run the simulation for longer, we would see that the population will stabilize around this number with some fluctuations. The percentage breakdown plot (bottom) shows that the ratios of the different grade categories are as expected at the start, until the promotions start to kick in, leading to a vastly skewed population composition after 50 years, which we would expect from the way this simulation was set up.

Figure 2 shows information of the influx into the state of 'Ooff Vast' (NonCom + appointed). The breakdown of this influx shows that the first influx into this category comes from the people who are NonCom and get their appointment (Ooff Appoint), but some time later a second influx is added to this, from the people who are Vol + Appointed (Ooff Soc Prom). The percentage breakdown plot shows that the latter type of influx becomes the dominant one. This is but a small sample of all the information that can be extracted from even this simple example, and serves to give some idea of the potential uses for this simulation tool.

VI. CONCLUSION

We are currently developing a generic simulation tool to assist with making manpower planning decisions, which can be used either for evaluating specific scenarios as well as serve as a driver for optimization routines to determine an optimal policy. In its present state, the tool is nearly functionally complete, where most features have been implemented albeit in a restricted form. As such, we wish and are encouraged to develop this tool further such that the additional extensions, mostly related to the flixibility and genericity of the tool, mentioned in the article are present in the final product, without excessively sacrificing the performance of the tool.

ACKNOWLEDGMENT

The authors would like to thank Guy Van Acker for his valuable insights into the practical side of personnel planning, and his help in steering the project in the right direction. We also would like to thank the reviewers for their interest in the paper and their insightful comments.

REFERENCES

[1] T. De Feyter and M.-A. Guerry, "Markov models in manpower planning: a review," 01 2011, pp. 67–88.

[2] V. O. Ezugwu and S. Ologun, "Markov Chain: A Predictive Model for Manpower Planning," Journal of Applied Sciences and Environmental Management, vol. 21 (3), 2017, pp. 557–565.

[3] B. A. Kassa and A. E. Tizazu, "Personnel scheduling using an integer programming model – an application at Avanti Blue-Nile Hotels," Springer Plus, 07 2013, online publication; doi: 10.1186/2193-1801-2-333; PMCID: PMC3923920, PMID: 24555163.

[4] Q. Chang, J. Ni, P. Bandyopadhyay, S. Biller, and G. Xiao, "Maintenance staffing management," vol. 18, 06 2007, pp. 351–360.

[5] B. Denkena, M. A. Dittrich, F. Winter, and C. Wagener, "Simulation-based planning and evaluation of personnel scheduling in knowledge-intensive production systems," Production Engineering: Research and Development, 10 2016, pp. 489–496.

[6] N. S. Narahari and K. N. Subramanya, "System dynamics modeling for human resource planning – a case study," URL: https://www.academia.edu/18935980/System_Dynamics_Modeling_For_Human_Resource_Planning_-_A_Case_Study [retrieved: August, 2018].

[7] "The Julia language," URL: https://julialang.org/ [retrieved: August, 2018].

ISBN: 978-1-61208-672-9

[8] “SimJulia,” URL: https://github.com/BenLauwens/SimJulia.jl [retrieved: August, 2018].

[9] “Julia Micro-Benchmarks,” URL: https://julialang.org/benchmarks/ [retrieved: August, 2018].

[10] “SQLite,” URL: https://www.sqlite.org/index.html [retrieved: August, 2018].

[11] “SQLite in Julia,” URL: https://github.com/JuliaDatabases/SQLite.jl [retrieved: August, 2018].

[12] “yEd,” URL: https://www.yworks.com/products/yed? [retrieved: August, 2018].

ISBN: 978-1-61208-672-9