Stride extended*

Andrei Bondarenko, Mathias Ooms, Stan Schepers, and Laurens Van Damme

University of Antwerp, Prinsstraat 13, 2000 Antwerpen, Belgium

Abstract. This paper is about the Stride simulation software which is a system used to do research about epidemiology. Our modifications to Stride make it possible to see a simulation visually, choose sizes for workplaces, read and write files more efficiently, etc.
Two researches were done with Stride. The first comprises of simulations in Belgium with multiple scenarios and different demographics. The second one is about how our changes to the Stride software changed the obtained results and/or performance.

Keywords: Stride · Epidemiology · Simulations · Belgium · Work size distribution · Demographic profiles · Daycare & Preschool · GUI.

1 Introduction

Stride (**S**imulate **tr**ansmission of **i**nfectious **d**is**e**ases) is epidemiological simulation software that can be used to examine how epidemiological diseases spread over a population in a given time.
The purpose of this paper is to give perspective in the understanding and the effect of the newly implemented features of the simulation software, so the user can perceive the implications of each of the new features. The new features help the simulator simulate more accurately by adding more realism. There are features which improve how the workplaces are generated, visualizing tools to interact easier with the results gained by the simulator, adding more places like day cares where people can get in contact with each other and so on. With everything combined a simulation for Belgium is done. More specific a comparison of how a disease behaves on a population in Flanders if Flanders is isolated and if Flanders is a part of Belgium. It seems that isolating Flanders has effect in the spread of a disease.

2 Daycare & PreSchool

This feauture introduces new contact-pool types: daycare's and preschools. Previously baby's and toddlers only belonged to household pools. The algorithms used to generate and populate these new contact are quite analogue to those of K12-schools, this choice is made because in our contours it's common that preschools are fused with K12-schools. This is not the case with daycare's, but

* Supported by organization COMP.

adjusting the parameters gives a good approximation of a real life situation. These parameters are the number of pools per daycare/preschool and the size of those pools. By researching different statistics (cfr. [5] and [6]), the following values are used:

Contact-pool type	**Number of pools per unit**	**Number of people per unit**
Daycare	1	9
Preschool	10	200

Table 1: Parameters for generation and population of daycare's/preschools. Ref: [5] [6]

By creating these extra pool types, there will be an increase in contact. So it's expected that when a disease is spreading, the total number of infected is increased in comparison to am absence of daycare's and preschools.
Taking an average of 100 simulations with a cumulative count of infected people (figure 1, this is clearly noticeable. Observing the different outbreak rates (figure 2), clearly shows an increase when this feature is applied.

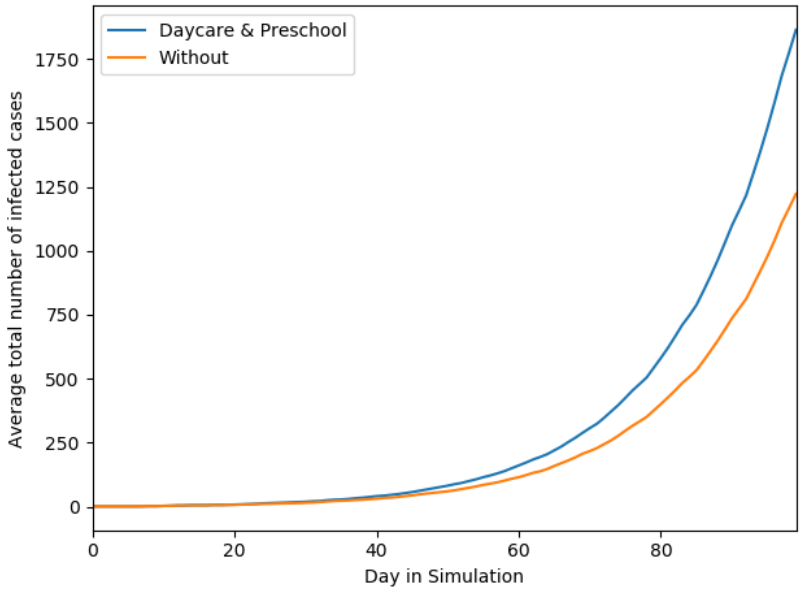

Fig. 1: Average cumulative cases per day of a 100 simulations (per category) using a 100 random seeds

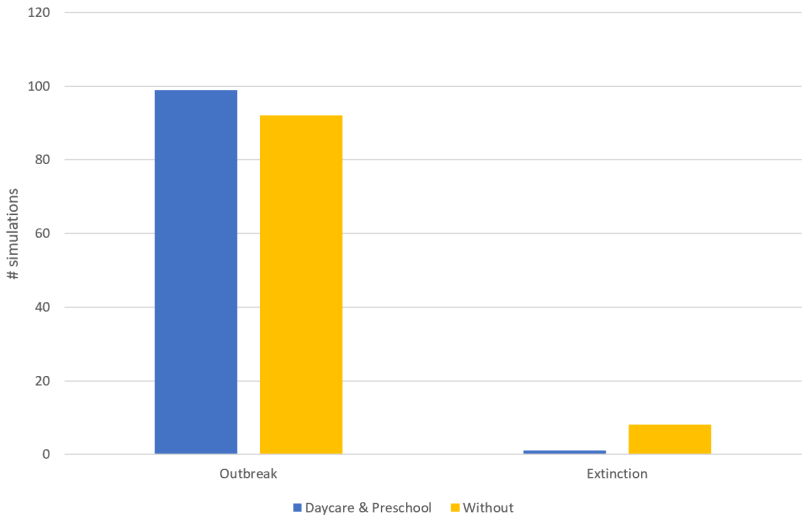

Fig. 2: Outbreaks of a 100 simulations (per category) using a 100 random seeds

3 Data formats

Stride required improvements in two areas that have to do with data input/output; Household files and GeoGrid files.

3.1 Household files

Household files were written and processed using a comma-separated values (CSV) file format. The first limitation of this format is that to support bigger households than those already present in the file, all existing households had to be extended with an additional column, introducing a lot of "junk" data. The JSON format solves this by by keeping each household in a separate array, allowing them to adapt any size they want independent of the size of other households. This "junk" data is also (one of) the cause(s) of the second limitation, namely the speed and efficiëncy of processing CSV versus JSON formatted files. A HouseholdJSONReader was implemented to improve upon these limitations. To benchmark the performance of the JSON format we performed two tests. First we converted the households_flanders.csv to a prettified JSON (with one space indentation) and a minified JSON to compare file sizes. Next we timed 10 runs of the SetReferenceHouseholds methods for each of the files, and calculated an average.

The results can be seen in table 2, where it's clear that a prettified JSON represenation decreases the file size by approximately 20%, while a minified JSON

represenation bumps the file size down by a staggering 62%. The use of a JSON representation also cuts the average runtime by 74% and 76%, for prettified and minified representations respectively, so the processing efficiëncy of JSON is unaffected by whether or not it is minified.

Format	**File Size (KB)**	**Average Runtime (μs)**
CSV	312	340846,6
JSON (prettified)	249	88484,9
JSON (minified)	118	81017,1

Table 2: Household format benchmarks

3.2 GeoGrid files

When choosing a format for GeoGrid files while using Stride three factors have to be considered; file size, I/O speed and human readability. The existing GeoGrid readers and writers were extended to support HDF5 and JSON format representations on top of the already supported Protobuf format. To compare the performance and space consumption of each format three tests were performed. First a GeoGrid file in each format was generated and their size compared. Next we wrote a GeoGrid to each format ten times and and calculated the average time needed to do so. Finally we did the same for reading the GeoGrid from a file. All tests were run using the provided *run_generate_default* and *run_import_default* XML-files, with the only change being the file formats used for I/O.

The results can be seen in table 3, where we can clearly see that Protobuf has a big performance advantage at the cost of its readability, because of its binary format. JSON's file size and read time on the other hand suffer from its excellent readability, the write time however is still acceptable when comparing to HDF5. HDF5 doesn't have excellent readability because we need special software to view/edit the files, but it's still possible. Its writing speed is very slow compared to JSON and Protobuf, however once it's written away, it takes up less space than JSON and its reading speed is also approximately twice as fast.

Format	**File Size (MB)**	**Avg. Write (ms)**	**Avg. Read (ms)**	**Readability**
Protobuf	14,5	2120,8	3681,6	None
JSON	297,2	17536,3	32204,1	Excellent
HDF5	153,3	21085,7	14398,7	Good

Table 3: GeoGrid format benchmarks

4 Data visualization

This section will discuss how the visualizer was built, what changes were needed in the Stride software to get it working and why it could be a help in epidemiological research.

The visualizer is built with the Qt5 framework using its markup language QML and C++ libraries. The MapController, which makes sure the visualization gets the right data, makes use of the GeoGrid data-structure from the Stride library to keep the data loaded. Some changes were needed in the Stride software to get this working. The Location data-structure of the Stride library didn't have the right information for the visualizer. It held the information to create epi-output but didn't have the epi-output itself. So the Location data-structure had the wrong content for the visualizer, but the right content for the simulations. The solution for this problem was to make templatize the Location class. The template is the type of content that will be stored in a location. For the simulations we need the Epidemiologic data-structure as the content of a location, which holds all the necessary information for the simulations and creation of the epi-output, and in the visualizer the EpiOutput data-structure is needed, which only holds the epi-output. It also has a get<0/1>function which returns the latitude/longitude of the location.
Since the GeoGrid class is also used by both the simulator and the visualizer, it was templatized as well to indicate what template argument is used by the locations in the grid.
A simulation run of a 100 days with the run_generate_default.xml configuration file takes around 55 seconds before the changes and around 57 seconds after. So it's doesn't have a significant impact on the run time of a simulation. (measured in the same circumstances)

A second change is the addition of a "generate epi-output function" for the EpiDemiologic and ContactPool data-structures. The function is called by the epi-output writer which supports the json, protobuf and hdf5 file formats. The writer is controlled by the epi-output viewer. The viewer can be used by the standard sim controller by adding the epi-output variable to the configuration file. (shown in the user manual)
The visualization needs that epi-output so a reader that supports the same file formats has also been provided.

At the end the visualizer provides an easy way to see how certain health categories evolve during the simulation and how certain diseases spread. It's also handy to get the information about a certain area in stead of only one location. For example Brussels consists out of multiple small regions but together they are one big city. Now it is easy to see how the epi-output changes in that whole area if we select it with a circle on the map as shown in figure 4.

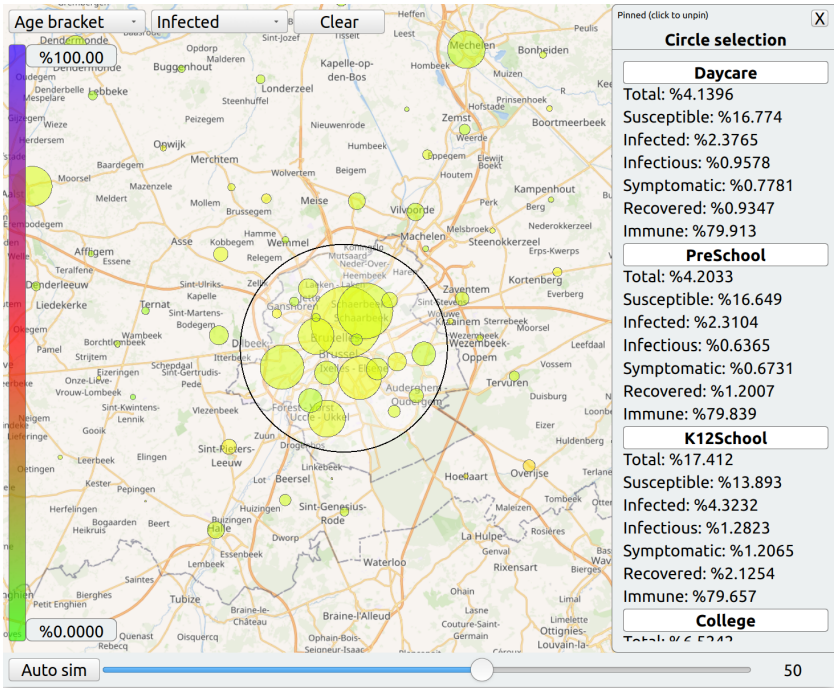

Fig. 3: Selection of the Brussels area in the data visualization tool for Stride

5 Demographic profile

This feature consists out of 2 components: using multiple different reference households in the simulation and reflecting these demographic differences in the rest of the simulation.

5.1 Using multiple reference households

This change is obtained by assigning every location a reference households that reflects its demography. The possibility to assigning each location its own reference households the most complex demographic differences can be reflected in the simulation. For example locations situated in a different region can have different reference households or big cities can have different reference households then more rural towns.

When populating households different households are randomly drawn from the reference households assigned to its location.

5.2 Young/Old fraction

The young/old fraction is defined by the fraction between persons of age 15 until 25 and persons of age 55 until 65. This fraction is calculated for every different household reference used. Every location has its individual fraction. If a locations uses the same household reference its only calculated once. A global average reference household is also created from the different given reference households.

This fraction is used when generating Daycares, Preschools and K12-Schools. The total amount of these types in the simulation are still calculated with the global average reference households. However the distribution of its locations will change. The location of this types is randomly assigned to locations with different weights. The weight is calculated from the fraction of people living in the location and its young/old fraction. The likelihood that a daycare, preschool or K12-school is placed increases when its population is larger and when the young/old fraction is larger.

This reflects demographic differences found in real life where there are more accommodations for children in a location when the population is on average younger. Colleges are deliberately not taken in account because the location of a college or university are more based on history then current population.

5.3 Results

These changes will not influence a simulation with for each location the same reference households. When populating the households the populators will draw from the same reference households. The young/old fraction will be the same

for all the locations so this will have no effect on the location of the daycares, preschools and K12-schools.

The global results of simulations with different reference households and the global results of simulations with one kind of reference households, the average of the different reference households, will differ little or not. The chances of picking a certain household in different reference households is equal then the chances of picking the same household from reference households created by appending all the different reference households. With the use of the global average reference households the differences will not affected the amount of daycares, preschools and K12-schools. Only the location will be different. However on average this will result in the same results.

For testing the statements above 3 different configurations are created from the default *run_flanders.xml* configuration. The first one is the default Flanders configuration using an average household file based (1 refHH). The second one uses different household files for the different provinces without the feature of increasing the likelihood that a younger location gets more schools assigned (no feature). The third one is the same as the second but with the extra feature (feature). We run the three configuration for 100 days and each with 100 different seeds. Out of this test configuration came these boxplots which describe the amount of cumulative infected cases.

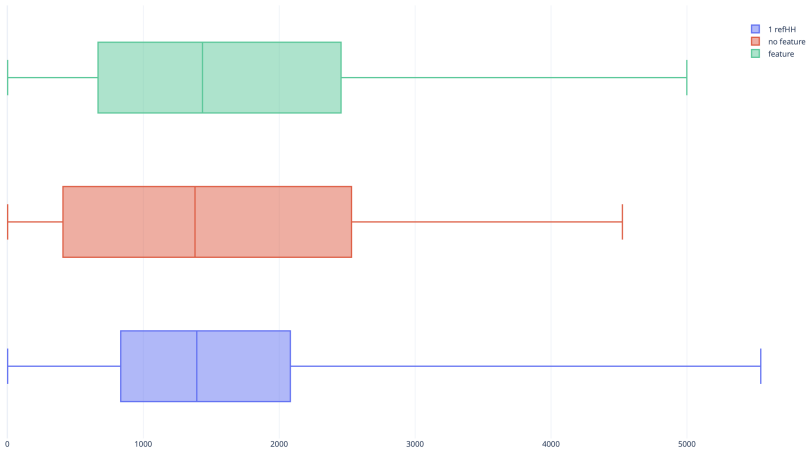

Fig. 4: Boxplots of the tests

The median is for every boxplot the same and the quantiles are close to each other. The added features for Demographic Profile don't influence the simulation when compared to global average of the demograhy.

5.4 Reference households in the Flemish provinces

The Stride simulation software contains reference households for the 5 Flemish provinces. They are created by taking households from the Flemish reference households that have "old" people and people who neither "old" or "young". We added households that have "young" people until the young/old fraction is matched with the young/old fraction of the provinces found in [1]. the different reference households are generated with representing Flemish households and have the right demographic profile.

6 Size class for company staff

6.1 Distribution format

This feature makes it possible the generate and populate workplaces more precisely and is controlled by a distribution file. This distribution file has to be a CSV-format file, this because it's easy to use for the user and most common for this kind of information. The distribution describes the ratio of workplaces that contain a size, set between a given minimum and maximum. It makes it possible to increase/decrease the number of small/big workplaces throughout the simulation, by adjusting the ratio, minimum size and maximum size (cfr. User manual).

6.2 Workplace generation

The generation of workplaces makes use of this data as follows. First it makes a more precise estimation of the total number of workplaces. When no distribution is given, the number of workplaces is calculated by:

$$\#workplaces = \frac{\#employees}{poolsize} \tag{1}$$

Where the number of *employees* is still determined by a household file, participation factor,... and *poolsize* is one of the predefined constants (for workplaces in this case). However making use of the distribution will calculate a average poolsize:

$$average\ poolsize = \sum ratio * (\frac{minimum\ size + maximum\ size}{2}) \tag{2}$$

For each entry/class in the distribution, the average poolsize is multiplied by it's ratio. Adding these values results in the *average poolsize*, to calculate the number of workplaces:

$$\#workplaces = \frac{\#employees}{average\ poolsize} \tag{3}$$

The next step in generation would be distributing the workplaces (pools) over the different locations by the calculated weights of those locations and that would be it. But to make the population of these workplaces (pools) later on easier (and gaining a lot of performance), an extra step is executed.
Each pool gets it's own soft limit, implying that pools are still able to add people when exceeding it's limit, but now it's possible to check if this pool exceeds it's limit. This is a major advantage over other implementations, as there isn't needed any extra storage to save these pools, no mechanism needed to hand over this information to the populator and thus no searching needs to happen when the populator needs to know the size of a certain workplace (pool). It even gives a very good foundation, to implement distributions for other pool-types.

So when distributing the workplaces over the locations, a discrete generator from the random engine (with the ratios as weights) is used to select the correct class. Followed by a uniform integer generator (random engine) to select a size between the minimum and maximum size of the selected class. This is very important to keep respect to the distribution, as otherwise the big locations would always end up getting filled with workplaces of the same class.
This is due to the fact that locations also make use of a weighted discrete generator from the random engine, so big locations will appear more often. By also making use of a weighted discrete generator with the ratios as weights, we select the correct amount of workplaces for each class and solve this problem.

6.3 Workplace population

The principle behind the population is now straightforward. When an employed person is found and determined if he commutes, a correct location (commuting/non-commuting) and associated pool are selected, but the employee is not yet added. First a check occurs, if this selected pool has not exceeded his limit, the employee is added to the pool. If otherwise, another pool from this location that doesn't exceeds it's limit will be selected. If there aren't any available pools left in this location another location is selected. Note that when there aren't any available locations left, the (few) employees that are left are distributed uniformly over the workplaces (pools), hence not ruining the distribution.

6.4 Scenario analysis

To compare the effect of using a workplace distribution, different scenario's were run with and without distribution files. To achieve the same expected values used for non-distributed simulations. This can be done by choosing specific ratios, mininum and maximum sizes, so the average poolsize would equal the constant poolsize (for workplaces this is equal to 20). Now the correct number of workplaces is generated, but to keep the total number of infected people in the same margin, the size needs to be fixed at 20. This way each workplace (pool)

gets the same size when there isn't used any distribution.

Ratio	Min size	Max size
1	20	20

Table 4: The distribution to simulate a non-distribution.

After running the different scenarios with this distribution (tests with influenza, measles,...) the same margins are respected relative to the targets (cfr. Test plan). So it's still possible to achieve the same results over different simulations using a distribution.

The following scenarios contain more accurate workplace size distributions for Flanders and Belgium. Based on research from [3] and [4] respectively. The results of these simulations are then compared to the targets and margins of non-distributed simulations, to perceive the impact of using distributions.

Ratio	Min size	Max size
0.834489433779642	0	9
0.129018004306445	10	49
0.027066076576166	50	199
0.009426485337747	200	1000

Table 5: The used distribution for Flanders. Ref: [3]

Ratio	Min size	Max size
0.191412708972848	0	1
0.200995210694430	2	9
0.069097501857753	10	19
0.077378215079115	20	49
0.131041002836234	50	249
0.330075360559621	250	1000

Table 6: The used distribution for Belgium. Ref: [4]

After these simulations, the same conclusion shows up. The acquired results show that using accurate distributions result in an overall reduction in the total number of infected people. For example (Influenza) a target of 554000 infected people gets a result of 548197, which is a significant difference. In general for all scenarios a margin of 1.1E-02 is needed such that the targets are still reached.

7 Simulation of Belgium

In this section simulations of the measles in Belgium are discussed, making use of the new implemented features. The main aim is to observe the difference between Belgium, Flanders as whole and Flanders as part of Belgium. The used data and the outcomes of these scenario's are explained further in more detail.

7.1 Simulation configuration

When running these simulations, there are a couple of things to keep in mind, concerning the configuration. Dividing these parameters into constants and variables, among the three different scenarios, should create more clarity.

All parameters concerning the conditions of the people are kept constant in each of the scenario's. This is important to only notice the geographic differences. The contact between people of different ages, the immunity of people, vaccination,... are examples of this, so that no simulation can be influenced other then the geographic aspect.
To see the overall differences of a disease spreading, the spreading is observed over a period of 300 days. This way the start-, peak- and end values can be compared. The measles disease is simulated and other technical parameters are kept constant. For generating the population, the participation factors and fractions are also not altered.

Changing the cities, households per province and the way people commuting are of essence, concerning the geographic differences. To make it even more interesting, different workplace size distribution are used (cfr. [3] and [4]). This way the accuracy of the approximation of a real life spreading is maximized for each scenario. For the exact values, see [7] for Belgium and [8] for Flanders.

7.2 Creating reference households for Wallonia Belgium

The reference households were created as described in Section 5.4. The data for calculating the young/old fraction was obtained from [2]. Data for Flanders was obtained from 5.4.

Province	Young/Old Fraction
Walloon-Brabant	0.98
Namur	0.92
Liege	0.91
Hainaut	0.90
Luxembourg	1.0

Table 7: Young/Old fraction of Walloon provinces

7.3 Modifications

The results from Flanders and Belgium were created without any changes to the new software. To extract the information about Flanders out of the simulation of Belgium, the Stan controller had to be modified. Instead of saving the amount of infected people of the population, only the amount of infected people of the locations in a province of Flanders were saved. This way the result only consists of data from Flanders within Belgium.

7.4 Results

Figure 5 shows us the average number of infected cases over a period of 300 days, for Belgium, Flanders and isolated Flanders. The first, quite obvious, observation that can be made is that the average total number of infected cases, as well as its growth ratio is a lot higher for Belgium as it has a bigger population. Next we see that the number of cases is higher for Flanders as a part of Belgium than for isolated Flanders, this is probably due to the easier spread of the illness because of presence of large cities near the Flanders-Wallonia border. The faster growth is a direct consequence from this too. The more people there are, the easier a disease gets spread. So when Flanders is part of Belgium there are overall more people and they can now also get infected by people from Wallonia and not only from the people in Flanders like when it is isolated.
However, figure 6 shows that there are is a smaller number of outbreaks in Flanders when it is part of Belgium than when it's isolated. This can be due to the fact that the fact whether or not a simulation ends in an outbreak is dependant upon the distribution of the first few infected people. As seen on figure 7, which shows how many percentage of the population of each location is infected of certain time stamps, the disease is first noticeable in the area of Ghent. It slowly spreads to Brussels. Brussels is the capital and biggest city in Belgium, so a lot of people will commute there for work for example, and continues to spread in the direction of Walloon. Later the disease is also very present in Antwerp while other cities slowly recover.
Imagine now that the first infected people would live in a small city in the "outskirts" of Wallonia down in the Ardennes. The chance that the disease will

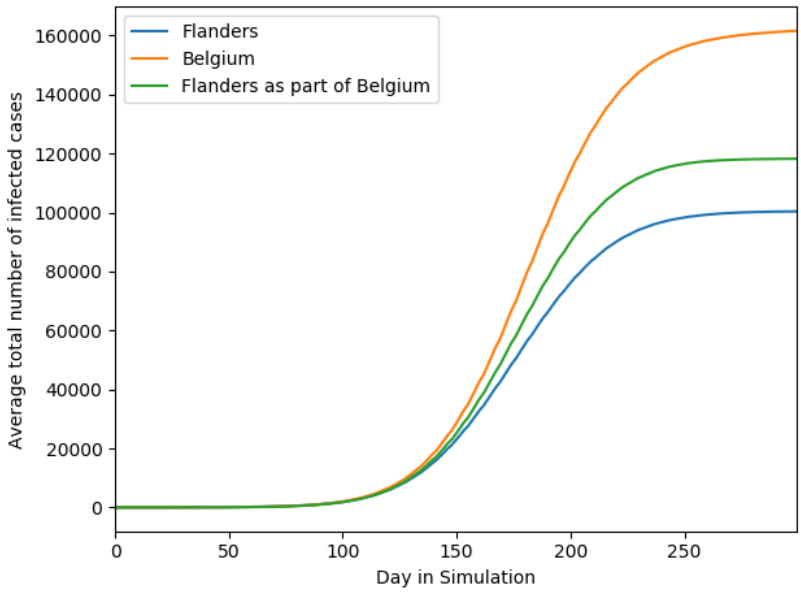

Fig. 5: Average cumulative cases per day of a 100 simulations (per category) using a 100 random seeds

spread to the rest of Belgium in such a scenario is much smaller. When Flanders is isolated, the first infected people will always live in crowded Flanders. So extinction cases are more rare if a simulation is run on isolated Flanders. When it's part of Belgium the probability of an extinction happening is greater because of the greater presence of smaller, more remote cities in Wallonia from where illnesses are less likely to spread. This can be the reason why there are on average less outbreaks when isolated than when not.

To summarize, the number of outbreaks is greater for Flanders when we look at it in isolation from its surroundings, but when these outbreaks do occur, they are less severe than those encountered when Flanders is simulated as a part of Belgium.

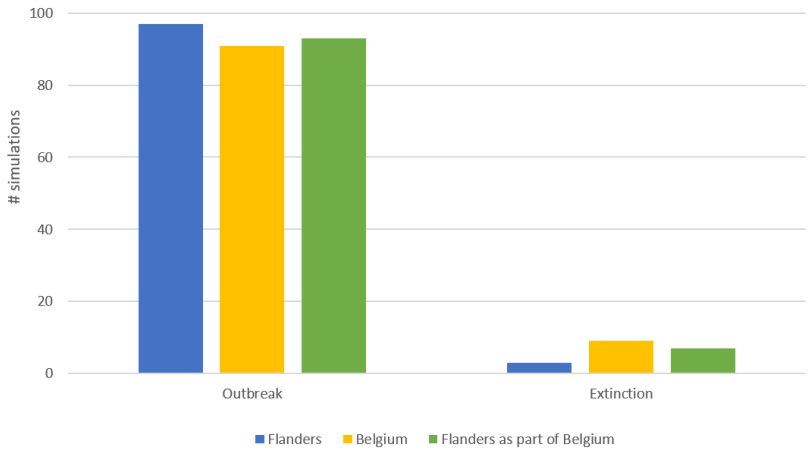

Fig. 6: Outbreaks of a 100 simulations (per category) using a 100 random seeds

8 Conclusion

Adding the new features to the Stride software gave it more options to run more realistic simulations, achieving more accurate results and gives a more user-friendly way to analyze results. Some features weren't that easy to implement and the collecting and editing of the needed data wasn't always straight forward. After all the Stride software has some new useful features as proven above and gives some interesting results.

The visualizer gives the opportunity to make more conclusions out of the resulted data from simulations that weren't visible before. Giving the Stride software the option to include different reference households for each location should reflect more realistic scenario's. The simulation can be more fine-tuned. This without influencing the global simulations. The results of these simulations can also be written and read in/from formats that are more readable to us humans than Protobuf's binary format, while still maintaining respectable performance figures. Contact between young children have a big impact on how fast a infectious disease spreads and sing realistic workplace size distributions will in general decrease the total number of infected people.

Finally the simulation of Belgium gave the result that adding Wallonia to Flanders has an effect on outbreaks in Flanders. Outbreaks will be likely to be larger. However the amount of outbreaks will be smaller.

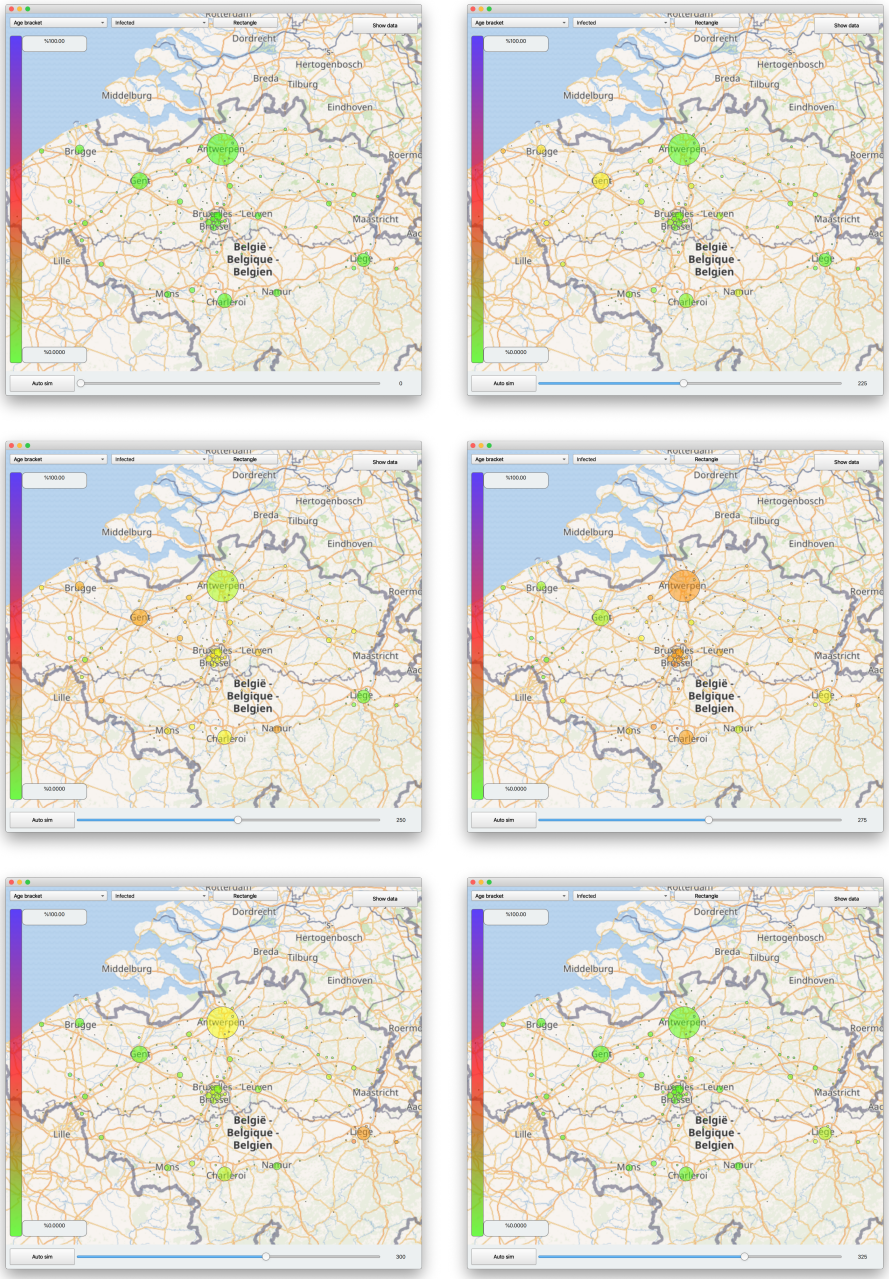

Fig. 7: Simulation of Belgium shown in the visualizer

References

1. Vlaamse Overheid (2012) *BeleidsBarometer 2012 de Vlaamse arbeidsmarkt in kaart*, p12, https://www.vlaanderen.be/publicaties/beleidsbarometer-2012-de-vlaamse-arbeidsmarkt-in-kaart
2. Statbel (2019) *Loop van de bevolking* https://statbel.fgov.be/nl/themas/bevolking/loop-van-de-bevolking
3. KU Leuven (2014) *Arbeidsdynamiek bij KMOs in Vlaanderen: groeitrends en ontwikkeling van een gebalanceerde groei-index* , p8, https://steunpuntore.be/publicaties-1/wp2/STORE-B-13-030-arbeidsdynamiekKMOsVlaanderen-final.pdf
4. Statbel (2016) *Structuur van de ondernemingen volgens grootteklasse van tewerkstelling voor de dienstensector* https://bestat.statbel.fgov.be/bestat/crosstable.xhtml?view=8878e258-858b-4c9b-b89e-2010de7e15b9
5. KindGezin (2019) *Kinderopvang voor baby's en peuters* https://www.kindengezin.be/cijfers-en-rapporten/cijfers/kinderopvang-baby-peuter/
6. Vlaanderen in cijfers (2017-2018) *Vlaams onderwijs in cijfers* https://www.vlaanderen.be/publicaties/vlaams-onderwijs-in-cijfers-2017-2018?section=5
7. Configuration file (2019) *Configuration file used in simulation of Belgium* https://github.com/stanschepers/stride/blob/SimulationBelgium/main/resources/config/run_belgium.xml
8. Configuration file (2019) *Configuration file used in simulation of Flanders* https://github.com/stanschepers/stride/blob/SimulationBelgium/main/resources/config/run_flanders.xml