

# Stride project

## User Manual

Version 1.0 (March 20, 2020) .

Centre for Health Economics Research & Modeling of  
Infectious Diseases, Vaccine and Infectious Disease  
Institute, University of Antwerp.

Modeling of Systems and Internet Communication,  
Department of Mathematics and Computer Science,  
University of Antwerp.

Interuniversity Institute for Biostatistics and statistical  
Bioinformatics, Hasselt University.

**Willem L, Kuylen E & Broeckhove J**

---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Software</b>	<b>3</b>
2.1	Source code . . . . .	3
2.2	Installation . . . . .	4
2.3	Documentation . . . . .	4
2.4	Directory layout . . . . .	4
2.5	File formats . . . . .	5
2.6	Testing . . . . .	5
2.7	Results . . . . .	5
<b>3</b>	<b>Simulator</b>	<b>6</b>
3.1	Workspace . . . . .	6
3.2	Running the simulator . . . . .	8
3.3	rStride Wrapper . . . . .	9
<b>4</b>	<b>Concepts and Algorithms</b>	<b>10</b>
4.1	Introduction . . . . .	10
4.2	Population . . . . .	11

# CHAPTER 1

---

## Introduction

---

This manual provides a brief description of the Stride software and its features. Stride stands for **S**imulation of **t**ransmission of **i**nfectious **d**iseases. It is an agent-based modeling system for close-contact infectious disease outbreaks developed by researchers at the University of Antwerp and Hasselt University, Belgium. The simulator uses census-based synthetic populations that capture the demographic and geographic distributions, as well as detailed social networks.

Stride is an open source software. License information can be found in the project root directory in the file `LICENSE.txt`. The authors hope to make large-scale agent-based epidemic models more useful to the community.

More info on the project and results obtained with the software can be found in

- Willem L, Stijven S, Tijskens E, Beutels P, Hens N & Broeckhove J. (2015) Optimizing agent-based transmission models for infectious diseases, BMC Bioinformatics, 16:183 [1]  
Kuylen E, Stijven, S, Broeckhove, J, & Willem, L (2017) Social Contact Patterns in an Individual-based Simulator for the Transmission of Infectious Diseases, Procedia Computer Science, 108C:2438 [2]  
Kuylen E, Willem L, Broeckhove J, Beutels P, Hens H (2019) Clustering of susceptible individuals within households can drive measles outbreaks: an individual-based model exploration, Medrxiv, [3]

Please note that a lot of functionality has been improved, extended and added since those publications.

---

### 2.1 Source code

---

The source code is maintained in a GitHub repository <https://github.com/lwillem/stride>. We use continuous integration via the TravisCI service. Every new revision is built and tested automatically at commit. Results of this proces can be viewed at <https://travis-ci.com/github/lwillem/stride>, where you should look for the master branch. The integration status (of the master branch) is flagged in the GitHub repository front page for the project.

Stride is written in C++ and is portable over Linux and Mac OSX platforms that have a sufficiently recent version of a C++ compiler. To build and install **Stride**, the following tools need to be available on the system:

- A fairly recent GNU g++ or LLVM clang++
- make
- A fairly recent CMake
- The Boost library
- Doxygen and LaTeX (optional, for documentation only)

A detailed list of current versions of operating system, compiler, build and run tools can be found in the project root directory in the file `PLATFORMS.txt`.

---

## 2.2 Installation

---

To install the project, first obtain the source code by cloning the code repository to a directory . The build system for **Stride** uses the **CMake** tool. This is used to build and install the software at a high level of abstraction and almost platform independent (see <http://www.cmake.org/>). The project also include a traditional **make** front to **CMake** with For those users that do not have a working knowledge of **CMake**, a front end **Makefile** has been provided that invokes the appropriate **CMake** commands. It provide the conventional targets to “build”, “install”, “test” and “clean” the project trough an invocation of **make**. There is one additional target “configure” to set up the **CMake/make** structure that will actually do all the work.

More details on building the software can be found in the file `INSTALL.txt` in the project root directory.

---

## 2.3 Documentation

---

The Application Programmer Interface (API) documentation is generated automatically using the Doxygen tool (see [www.doxygen.org](http://www.doxygen.org)) from documentation instructions embedded in the code .

The user manual distributed with the source code has been written in **L<sup>A</sup>T<sub>E</sub>X**(see [www.latex-project.org](http://www.latex-project.org)).

---

## 2.4 Directory layout

---

The project directory structure is very systematic. Everything used to build the software is stored in the root directory:

- **main**: Code related files (sources, third party libraries and headers, ...)
  - **main/<language>**: source code, per language: cpp, python, R
  - **main/resources**: third party resources included in the project:
- **doc**: documentation files (API, manual, ...)
  - **doc/doxygen**: files to generate reference documentation with Doxygen
  - **doc/latex**: files needed to generate the user manual with Latex
- **test**: test related files (scripts, regression files, ...)

---

## 2.5 File formats

---

The Stride software supports different file formats:

### **CSV**

Comma separated values, used for population input data and simulator output.

### **JSON**

JavaScript Object Notation, an open standard format that uses human-readable text to transmit objects consisting of attribute-value pairs. (see [www.json.org](http://www.json.org))

### **TXT**

Text files, for the logger.

### **XML**

Extensible Markup Language, a markup language (both human-readable and machine-readable) that defines a set of rules for encoding documents.

---

## 2.6 Testing

---

Unit tests and install checks are added to Stride based on Google’s “gtest” framework and CMake’s “ctest” tool. In addition, the code base contains assertions to verify the simulator logic. They are activated when the application is built in debug mode and can be used to catch errors at run time.

---

## 2.7 Results

---

The software can generate different output files:

### **cases.csv**

Cumulative number of cases per day.

### **summary.csv**

Aggregated results on the number of cases, configuration details and timings.

### **person.csv**

Individual details on infection characteristics.

### **logfile.txt**

Details on transmission and/or social contacts events.

### 3.1 Workspace

By default, **Stride** is installed in `./target/installed/` inside the project directory. This can be modified by setting the `CMAKE_INSTALL_PREFIX` on the CMake command line (see the `INSTALL.txt` file in the project root directory) or by using the `CMake-LocalConfig.txt` file (example file can be found in `./src/main/resources/make`).

Compilation and installation of the software creates the following files and directories:

- Binaries in directory `<install_dir>/bin`
  - *stride*: executable.
  - *gtester*: regression tests for the sequential code.
  - *rStride*: R-based simulation wrapper (under construction)
- Configuration files (xml and json) in directory `<install_dir>/config`
  - *run\_default.xml*: default configuration file for Stride.
  - ...
- Data files (csv) in directory `<project_dir>/data`
  - *contact\_matrix\_flanders\_conditional\_teachers.xml*: Social contact rates for flanders for different locations, conditional upon presence, including teaching activities.
  - *contact\_matrix\_flanders\_conditional\_teachers\_15min.xml*: Idem as above... but selecting only contact of at least 15 minutes.
  - *disease\_xxx*: Disease characteristics (incubation and infectious period) for xxx.
  - *holidays\_xxx*: Holiday characteristics for xxx.

- *pop\_belgium600k\_teachter\_censushh.zip*: compressed folder with population data of 600k people for Belgium, including teaching activities.
- *immunity\_xxx*: Age specific immunity profiles.
- Documentation files in directory `./target/installed/doc`
  - Reference manual
  - User manual

The install directory is also the workspace for **Stride**. The **Stride** executable allows you to use a different output directory for each new calculation (see the next section).



---

## 3.2 Running the simulator

---

From the workspace directory, the simulator can be started using the command “./bin/stride”. Arguments can be passed to the simulator on the command line:

USAGE:

```
bin/stride [-c <CONFIGURATION>] [-o <<NAME>=<VALUE>>] ...  
[-i] [--] [--version] [-h]
```

Where:

```
-c <CONFIGURATION>, --config <CONFIGURATION>  
    Specifies the run configuration parameters. The format may be either  
    -c file=<file> or -c name=<name>. The first is mostly used and may be  
    shortened to -c <file>. The second refers to built-in configurations  
    specified by their name.  
  
    Defaults to -c file=run_default.xml  
  
-o <<NAME>=<VALUE>>, --override <<NAME>=<VALUE>> (accepted multiple  
    times)  
    Override configuration file parameters with values provided here.  
  
-i, --installed  
    File are read from the appropriate (config, data) directories of the  
    stride install directory. If false, files are read and written to the  
    local directory.  
  
    Defaults to true.  
  
--, --ignore_rest  
    Ignores the rest of the labeled arguments following this flag.  
  
--version  
    Displays version information and exits.  
  
-h, --help  
    Displays usage information and exits.
```

---

### 3.3 rStride Wrapper

---

An R wrapper is under development to perform multiple runs with the C++ executable.

---

#### 4.1 Introduction

---

The model population consists of households, schools, workplaces and communities, which represent a group of people we define as a “ContactPool”. Social contacts can only happen within a ContactPool. When school or work is off, people stay at home and in their primary community and can have social contacts with the other members. During other days, people are present in their household, secondary community and a possible workplace or school.

We use a *Simulator* class to organize the activities from the people in the population. The ContactPools in a population are grouped into ContactCenters (e.g. the different classes of a school are grouped into one K12School ContactCenter). These ContactCenters are geographically grouped into a geographical grid (sometimes called GeoGrid)

The *ContactHandler* performs Bernoulli trials to decide whether a contact between an infectious and susceptible person leads to disease transmission. People transit through Susceptible-Exposed-Infected-Recovered states, similar to an influenza-like disease. Each *ContactPool* contains a link to its members and the *Population* stores all personal data, with *Person* objects. The implementation is based on the open source model from Grefenstette et al. [4]. The household, workplace and school clusters are handled separately from the community clusters, which are used to model general community contacts. The *Population* is a collection of *Person* objects.

---

## 4.2 Population

---

### 4.2.1 Background

To explain the algorithms used for generating the geography of the countries and their respective population, we have to introduce some background concepts.

**ContactPool** : A pool of persons that may contact with each other which in turn may lead to disease transmission. We distinguish different a number of types of ContactPools associated with the household, the workplace, the school, .... The Household is a key type because it fixed the home address of a person.

**K-12 student** : Persons from 3 until 18 years of age that are required (at least in Belgium) to attend school. Students that skip or repeat years are not accounted for.

**College student** : Persons older than 18 and younger than 23 years of age that attend an institution of higher education. For simplicity we group all forms of higher education into the same type of ContactCenter, a College. A fraction of college students will attend a college “close to home” and the others will attend a college “far from home”. Most higher educations don’t last 6 years, but this way we compensate for changes in the field of study, doctoral studies, advanced masters and repeating a failed year of study.

**Household profile** : The composition of households in terms of the number of members and their age is an important factor in the simulation. In this case the profile is not defined by the age of its members or fractions, but through a set of reference households. This set contains a sample of households which is representative of the whole population in their composition.

The population files are created in a separate project and consist of:

**Households**

**K-12 Schools**

**Colleges**

**Workplaces**

**Communities**

---

## Bibliography

---

- [1] L. Willem, S. Stijven, E. Tijskens, P. Beutels, N. Hens, and J. Broeckhove, “Optimizing agent-based transmission models for infectious diseases.,” *BMC Bioinformatics*, vol. 16, p. 183, 2015.
- [2] E. Kuylen, S. Stijven, J. Broeckhove, and L. Willem, “Social Contact Patterns in an Individual-based Simulator for the Transmission of Infectious Diseases.,” *Procedia Computer Science*, vol. 108C, pp. 2438–2442, 2017.
- [3] E. Kuylen, L. Willem, J. Broeckhove, P. Beutels, and N. Hens, “Clustering of susceptible individuals within households can drive measles outbreaks: an individual-based model exploration,” *medRxiv*, 2019.
- [4] J. J. Grefenstette, S. T. Brown, R. Rosenfeld, J. DePasse, N. T. Stone, P. C. Cooley, W. D. Wheaton, A. Fyshe, D. D. Galloway, A. Sriram, H. Guclu, T. Abraham, and D. S. Burke, “FRED (A Framework for Reconstructing Epidemic Dynamics): an open-source software system for modeling infectious diseases and control strategies using census-based populations,” *BMC public health*, vol. 13, no. 1, p. 940, 2013.