O.D.D. description of the COMOKIT Meso model

Projet: COMOKIT - http://comokit.org/
Version: 2.0
Date: February, 14nd 2023

Authors: *Arthur Brugière[1] (0000-0001-7220-449X), Kevin Chapuis[1,5] (0000-0003-1743-2224), Marc Choisy[4,8] (0000-0002-5187-6390), Alexis Drogoul[1] (0000-0002-9685-9199), Benoit Gaudou[1,3] (0000-0002-9005-3004), Huynh Quang Nghi[1,6] (0000-0003-3608-9580), Nguyen Ngoc Doanh[1,7], Pierre Larmande[8] (0000-0002-2923-9790), Damien Philippon[10] (0000-0002-6641-8157), Patrick Taillandier[2] (0000-0003-2939-4827)*

Affiliations:

1. UMI 209, UMMISCO, IRD, Sorbonne Université, Bondy, France
2. UR 875, MIAT, INRAE, Toulouse University, Castanet Tolosan, France
3. UMR 5505, IRIT, Université Toulouse 1 Capitole, Toulouse, France
4. UMR 5290, MIVEGEC, IRD/CNRS/Univ. Montpellier, Montpellier, France
5. UMR 228, ESPACE-DEV, IRD, Montpellier, France
6. CICT, Can Tho University, Can Tho, Vietnam
7. MSLab / WARM, Thuyloi University, Hanoi, Vietnam
8. UMR 232, DIADE, IRD, Univ. Montpellier, Montpellier, France
9. OUCRU, Centre for Tropical Medicine, Ho Chi Minh City, Viet Nam
10. WHO Collaborating Centre for Infectious Disease Epidemiology and Control, School of Public Health, Li Ka Shing Faculty of Medicine, The University of Hong Kong, Hong Kong Special Administrative Region, China.

We describe in this document the COMOKIT model using the standard O.D.D. protocol[1] in its first review version.

Overview

Purpose

This model aims at simulating and comparing the application of COVID-19 spread mitigation policies at the scale of a closed commune, the transmission of the disease being modeled at the individual scale. Its purpose is to support deciders and researchers in answering questions such

[1] Grimm V, Berger U, DeAngelis DL, Polhill JG, Giske J, Railsback SF. The ODD protocol: A review and first update. Ecological Modelling. 2010 Nov 24;221(23):2760–8.

as: Is the containment of a neighborhood more effective than that of an entire village? Does closing schools decrease the transmission peaks ? How does wearing masks impact the dynamics of the epidemy ? How long should a lockdown ideally last ? What proportion of the population should be allowed to undertake activities during a lockdown ?

Several case studies are provided with the model: two Vietnamese communes of Son Loi (Vinh Phuc, Vietnam) and Thua Duc (Ben Tre, Vietnam); the small town of Castanet Tolosan (near Toulouse, France); and two smaller test datasets.

Beyond these case studies, the model has been designed as a framework generic enough to be applied to any case study as long as the correct input data is provided. A great deal of the design effort has been put on this aspect.

Entities, state variables, and scales

Scales

The simulations are executed at the scale of a commune in Vietnam (the Son Loi commune covers a surface of 9,5 km²). The smallest considered spatial units are individual buildings.

The simulations are not launched from a specific starting date, but rather from the introduction of the first infected cases in the population and will run until the end of the epidemic. The simulation step is set to 1 hour. As a consequence, given the simulated area, movements from one activity place to another one are not simulated: individuals are always located in an activity place (that can be a close building or even an outdoor park). The underlying assumption is that no infection cannot occur during the move time.

Entities

The model is designed to simulate the COVID-19 spread at the individual scale. As a consequence, the core entity of the model is the `Individual` kind of agents (or species): it represents individual inhabitants of the commune with their individual characteristics (age, sex, employment status) and their epidemiological status, whether they have been tested, and other epidemiological individual-dependent values (e.g. latent_time, infectious_time … c.f. the epidemiological submodel description for more details). They perform their daily activities (including going to work, school, shopping, eating outside…) depending on their personal agenda. This agenda is a generated set of `Activity` that can be shared by several individuals (e.g. going to a restaurant with some friends), depending on the age and family status of the `Individual` agent. `Individual` agents' attributes include their relatives (their family which corresponds in our model to the other `Individual` living in the same flat in a `Building`), their friends (with whom they can share activities), their colleagues (work colleagues or classmates) and their home, working place and school `Building`.

Building agents are spatial entities where the Individual agents can perform an Activity, this Activity depending on the Building type. Two special Building kinds have been defined as they have an important role in the simulation: outside and Hospital. The Outside agent represents all the buildings outside of the simulated area: it is used to represent the fact that people can be working or doing activities outside the area under consideration. For this zone, a particular contamination dynamic is applied. Hospital agents will be the place where, in some situations, sick Individual agents with critical symptoms can be contained and healed.

The Individual hourly behavior is driven by their agenda attribute that associates to some hours of the day an Activity. An Activity is mainly a way to choose the spatial unit(s) in which the Individual agents have to be located at each simulation step. The choice of the spatial unit to carry out an Activity depends on several factors: the first one is the preferences of the Individual that is defined according to his/her age an sex (through a csv parameter file): for a leisure activity, a child may prefer to go to a game center while an older person may prefer to go to a movie theater. Once the type of spatial units defined, the choice of the place among those of the chosen type depends on the model chosen. COMOKIT offers 3 basic models: choice of a random place, choice of the place closest to the individual or gravity model. In the case of the gravity model, the probability of choosing a place will depend on the area of the place and on its distance to the Individual (the larger the place and the smaller the distance to it, the greater the chances of choosing this place).

We have also defined additional specific Activity species to represent the main classical kinds of Activity: visiting_neighbor, working, staying_home, studying, visiting_friend. Of course, customs activities can also be created from the generic Activity species.

As our main goal is to simulate and compare the application of various mitigation and control policies, a specific focus is made on policies that modify the population behavior to reduce contacts and thus infection between people. As a consequence, the performance of an Individual agent's Activity is constrained by the allowance of the Authority agent. This agent manages the various Policy that are adopted. When an agent asks it for authorization to perform a given Activity, the Authority asks all the policies it has adopted whether any of them denies the given Individual to do the given Activity. Examples of Policy include total containment, close the schools, close the work spaces… These policies can be limited to a given area (using SpatialPolicy) or be more or less tolerant (e.g. containment can be for every body or for every body but some people, or some rate of the population, using PartialPolicy).

All the agent species are summarized and organized on the UML diagram presented in **Figure 1**. A specific diagram describing exhaustively all the policies is presented in the Sub-model section.

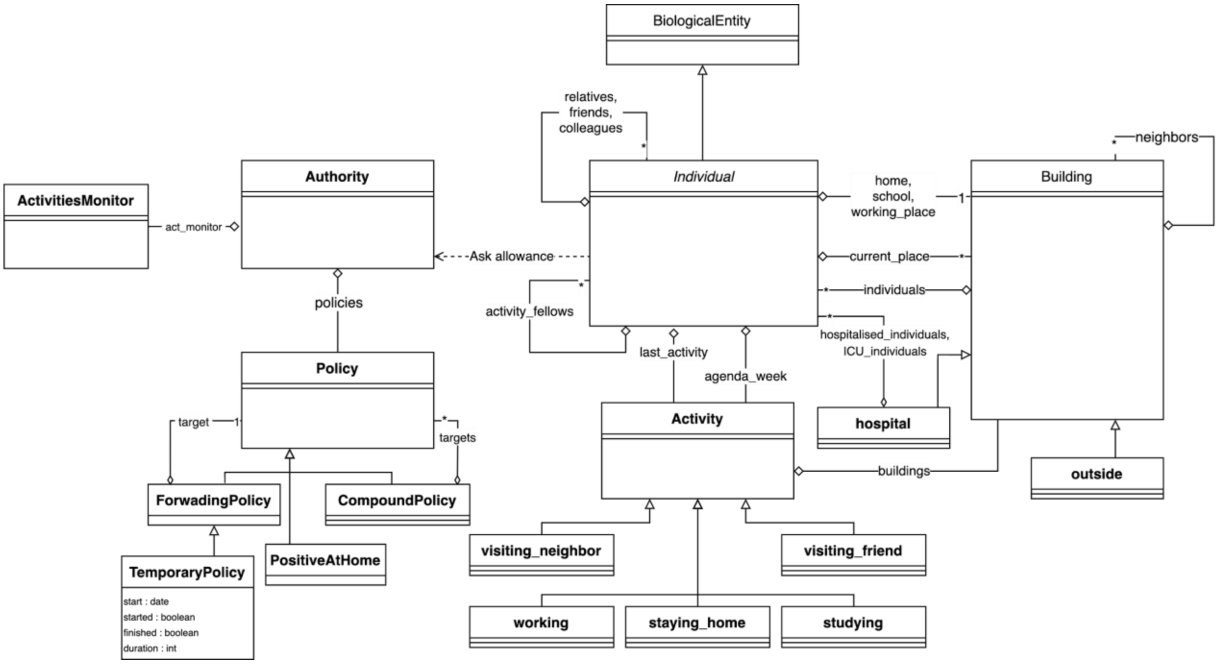

Figure 1. UML class of an overview of the COMOKIT model

Process overview and scheduling

The model simulates the spread of the COVID-19 in a population at the individual level under the control of mitigation policies or individual behaviors (such as wearing masks). The dynamics of the model can thus be summarized by three main dynamics: the epidemic dynamics, the hourly activities of the Individual agents, following their agenda to go from Building to Building and the dynamics of policy adoption and application.

There are two different pathways of infection for Individual agents: either through Individual-to-Individual transmission, or through persistence of the virus in the environment. When an infectious Individual is located in a building, it can release a virus load inside the building, which can survive several hours. Individuals who will come to this building can thus become infected by the viral load present in the building itself. As soon as an Individual is infected, its epidemic status will be described by a set of states and transitions (given probabilities taken from the up-to-date COVID-19 literature).

A simulation step starts by the evolution of the viral load in a building (it decreases over time, before disappearing). Then the Individual agents behave. They first evaluate whether they are infected or infect other Individuals or the current building in which they are located. They then update their epidemic status (given the model detailed in the Sub-model Section) and their individual behavior related to mask wearing. Finally they execute their daily activities: they find the activity corresponding to the current hour, ask the Authority whether they are allowed to execute it and act in accordance.

Finally, the Authority agent checks its current Policy and tries to apply it (executing a test campaign for example).

Design Concepts

Basic principles

As far as the epidemiological dynamics is concerned, we rely on much scientific evidence that the disease could be represented by a SEIR model (44) with an infectious state that can be presymptomatic, symptomatic or asymptomatic, with a certain degree of survivability of the virus in the environment and the possibility of people being infected by it.

The individual agents' behavior is described using an activity-based approach[2]: people have a set of activities associated with some day hours. This agenda makes the agents jump from buildings to buildings (due to the 1-hour step of the simulation).

Emergence

The main emergent (or at least complex to predict) results are the evolution of the number of infected cases given the different parameters that intervene in the definition of a policy, notably its geographical, social and temporal extents. As an example, in Figure 2, a global containment policy with an unconfined rate of 5% (i.e. 5% of the inhabitants are behaving normally) starts by correctly "flattening the curve" but, maybe surprisingly, results in a second wave that appears to be more severe than the first one if it is not applied long enough.

Adaptation

Up to now, only the Authority agent can be considered as having an adaptive behavior when it uses some specific policies. As an example, some policy can be applied when a condition is fulfilled. This condition can be described in terms of a number of infected people, a time period, or a spatial area to quarantine.

Conversely (and this is a limitation that could be overcome in a future version of the model) Individual agents do not have any adaptation behavior: as an example, when children are forbidden to go to school, they only stay home, whereas in reality they would go the park, play with other children in the same street ...

[2] Chapin FS. Human activity patterns in the city: Things people do in time and in space. Wiley-Interscience; 1974. Ilägcrstrand T. What about people in regional science?. InPapers of the Regional Science Association 1970 (Vol. 24).

Sensing

Mainly due to the huge number of asymptomatic infectious people, we consider that the Authority agent cannot have a knowledge of the individuals' epidemiologic state. To access this information (that is mandatory to apply or release a given policy taken into account the number of infectious cases), the Authority can only test people. These tests are not perfect: they have probabilities to return false negative and false positive results. As a consequence, the perception of the Authority agent is neither perfect, nor complete on the epidemiological state of its population: it cannot test everybody at each simulation step (for resource limitations).

Interaction

Individual agents can infect other Individual agents directly through contact or indirectly through building contamination: Infectious Individual agents can release a viral load in a Building agent and, as the virus can survive in the Building for a period of time, a Building agent with a viral load will possibly infect the Individual agents located in it, following the assumption that contaminated surfaces such as doorknobs, tables, on which the virus can survive, are possible transmission pathways. Interactions between Individual agents and the Authority agents occur in both directions: Individual agents ask the Authority's agent authorization to execute a given Activity and reversely the Authority agents can apply a Policy to check the epidemic state of the Individual agents.

Stochasticity

As we integrate in our model an heterogeneous population, and that a great deal of data is impossible to know with certainty (e.g., when infected, what is the time spent by an Individual time in each epidemiological state?), many values are randomly generated in a range of values coming from the literature.

1. At the initialization, here are the Individuals' attributes randomly generated:
    a. the distribution of individual attributes, including household members, employment status, age and sex (when built-in synthetic population generation procedure is used),
    b. the localization of household and bounded buildings for agents' activities such as school and working_place,
    c. the network of friends and colleagues (classmates)
    d. the series of activities for a week, i.e. agendas including time for lunch, time to go to school, evening activity and so on.
2. As far as the Individual agents' epidemiological dynamics:
    a. the test of its epidemic state follows probabilities of false positives and negatives,
    b. the fact of wearing a mask is random given a model parameter,
    c. when an Individual agent becomes infected, some epidemiological attributes are randomly computed following different distributions: its incubation period, serial interval (for presymptomatic period) and infectious period,

   d. the fact of becoming infected outside, of infecting another Individual agent, of being infected in an infected building are random given a model parameter,
   e. the facts of being Symptomatic or Asymptomatic, and of dying from the disease follow some age-related epidemiological parameters,
   f. Individual factors regarding infectivity, increasing or decreasing the transmission risk of that Individual are following a random distribution

3. Concerning Individual agents' activities:
   a. when finding the target of an Activity, there is a probability to do the activity outside.
   b. when visiting friends, there is a random choice of the friends to visit.
4. The application and the allowance choice of some Policy agents can be stochastic:
   a. LockdownPolicy: the choice of the Individual agents that will be allowed to move is random,
   b. PartialPolicy: each Individual agent has a tolerance probability to be allowed to move,
   c. DetectionPolicy: Individual agents that will be tested are chosen randomly in the population.

Collectives

Individual agents are part of several agents collectives: relatives, friends, colleagues, activity fellows. Nevertheless these collectives are not agentified with a specific behavior. The only impact is on the transmission: in a same building, individuals belonging to the same household will have a higher transmission rate with their relatives than with the other individuals in the same building. Similarly, the transmission rate will be higher between colleagues than with other individuals working in the same building. The main idea is that individuals in the same building will not have the same number of contacts with all the individuals in a (working or home) building. They thus have more chances to infect close colleagues or relatives.

Observation

The observation interface of the COMOKIT simulations can be defined depending on the experiments' needs. As the model has been mainly designed to assess and compare policies, most of the experiments run several simulations in parallel with different parameter values. The observation interface will thus contain a display of the spatial evolution of the disease for each parameter value and a chart plotting the evolution of the number of infected individuals over time. **Figure 2** shows an example of such an interface considering five different proportions of unconfined people. It is also possible to display the activity decreases for different activity types (compared to a baseline where no policy would be applied).

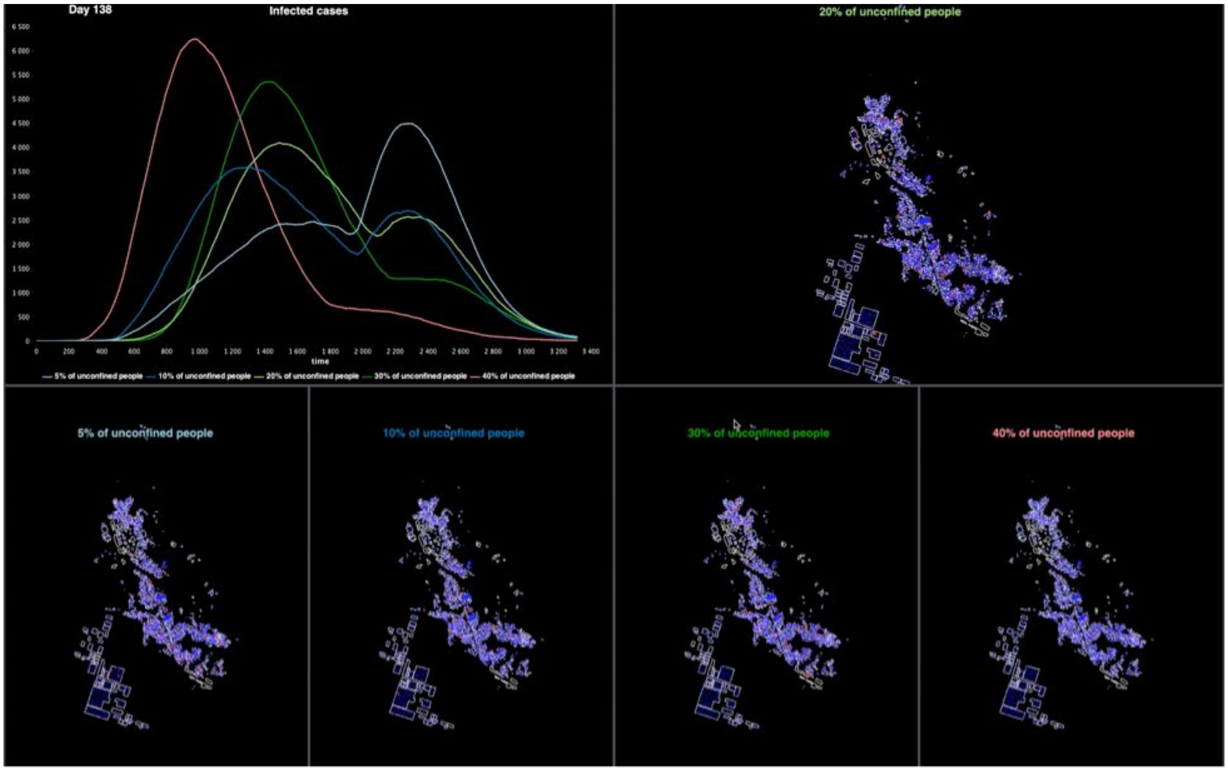

Figure 2. Example of Graphical User Interface of a COMOKIT experiment on the Son Loi commune with 5 simulations with various numbers of unconfined people.

Details

Initialization

In order to keep the model as generic as possible, many parameters and initial values are stored in case-dependent external files. Two parameter files for epidemiological model and activity types links to building types are stored in a general purpose parameter folder. In order to use them in a custom version of COMOKIT, users should either redefine them or give the path to this folder relative to the new project (for example, see Template Projects in COMOKIT). The initialization of a simulation combines reading files and assigning values to the main parameters, with synthetic population characteristics (location, social network, agenda) generators as follows:

1. Initialization of the epidemiological parameters from the associated csv file.
2. Creation of the Boundary and Building agents from corresponding shapefiles, the buildings initialize their type attribute from the type attributes of the shapefile attribute table. Finally the outside building is created.
3. Creation of 1 Activity agent for each built-in activity (visiting_neighbour, visiting_friend ….) and one for each of the meta-type of buildings (a meta-type gathers a set of types with a

close purpose, e.g. shopping can be done in shops, markets, supermarkets and store Building).

4. Creation,initialization and localisation of the Individuals synthetic population. Each house contains one or several household of individuals characterized by their age, sex, employment status (unemployed or not), social networks and home building. Two ways of initializing the population are provided: (i) import of a population csv file containing the individuals' description (age, sex, employement status and household id), that should be located in the houses (according to the number of flats in the buildings); (ii) generation of a standard population diretly from the GAML code: each flat is populated with a household (defined through the population csv file - see input data). Each individual gets a home and a school or a working place building (depending on its age and employment status). Lastly, each individual agent is bound to a set of friends and colleagues (or classmates).
5. Generation of the agenda. Each Individual agent has an agenda at the scale of the week composed by 7 daily agendas (that associate to some hours an activity). The agenda depends on the age: students and workers have an agenda with several working days and leisure days (i.e. a day with only activities that are not working, learning or staying home); retired individuals have an agenda full of leisure days. Individuals that are too young have an empty agenda.
6. Initialize epidemic: a given number of Individual agents become infected: they start by being in the exposed state and their disease-related attributes are initialized (incubation and infectious times and its serial interval).
7. Creation of the Authority agent with its Policy.

Built-in synthetic population generator

The synthetic population of agents comprises three dimensions of agent attributes to be generated: *demographic variables*, including `age`, `sex`, employment status (`is_unemployed`) and households (`household_id`), *social network variables*, including the lists of `friends`, `colleagues` or classmates, and lastly *mobility variables* including activity timeline and corresponding activity locations (`agenda_week`). In this version of the model, only demographic variables can be initialized from an external csv file generated by an external synthetic population generator[3].

Synthetic individual and household

The first step toward initializing the population of agents for the COMOKIT model can be done either using a generic micro-data-like tabular file of the entire population or using the built-in algorithm to generate them. For the former, individual records (a line in the file population.csv) must have attributes for age, sex, employement status and an household identifier as mentioned

[3] We choose to use the Gen* generator (that can also be coupled with the GAMA platform https://github.com/ANRGenstar/genstar.gamaplugin)
Chapuis, K., Taillandier, P., Renaud, M., & Drogoul, A. (2018). Gen*: a generic toolkit to generate spatially explicit synthetic populations. International Journal of Geographical Information Science, 32(6), 1194-1210.

above. Modelers should also provide the model with correct encoded translation from record values to COMOKIT variables, e.g. if the age attribute of records are encoded using range value (for example from 0 to 5, 5 to 16, etc.) modelers should provide the correct mapping to translate them into an integers modifying the corresponding map. This can be done using a csv file named Population Records.csv, specified and included in the case study repository, and that makes it possible to bind variables from the model to variables name and encoded values in the synthetic population. For the latter case, the built-in generation follows a sequential probabilistic procedure: for every homeplace, the household living in is chosen to be either a lone individual or mother/father couple with `proba_active_family` probability. In the second case, a number of children (truncated gaussian) and potential grandparents living in the household are determined following given probabilities (`proba_grandfather` and `proba_grandmother`). Lastly, the algorithm assigns an employment status (`proba_unemployed_M` and `proba_unemployed_F`) to every active agent aged between `max_student_age` and `retirement_age`.

Synthetic locations

When the population of agents grouped in households is initialized, the model will determine the homeplace (`home`) for the household and frequented locations for every agent. This twofold process is made as follows: each household is assigned a flat in a building drawn uniformly, where the feature 'flat' is taken from buildings.shp and used to identify the number of homeplaces per building (1 if there is no feature or no value associated). The second step consists in assigning locations to agents according to age and employment status: a `school` for schoolgirls and boys, a workplace (`working_place`) for employed active adults. In the latter case, the workplace can be a building within the studied area or the outside proxy agent. Each of these locations are chosen according to one of three distinct procedures among that makes it possible to pick one building in the corresponding set (e.g. building with type school for schoolgirls and boys): (i) uniformly among all relevant buildings (ii) the closest one with the proper type or (iii) following a custom gravity functions, with target density being a positive function of the area of the building. For any leisure activities that agents can engage in, the target building is chosen according to one of these procedures.

Synthetic social network

The social networks of agents are built based on agent's demographic attributes as well as activities: the first layer of relationship comprises people living in the same house, which is directly provided by the synthetic household. They are all linked to each other and considered as `relatives`. The second layer of connection among the synthetic population is friendship: for any agent *i,* the list of *n* `friends` is uniformly drawn within agent *i* age category, with *n* being drawn from a parametrized truncated gaussian distribution. The third layer of the synthetic social network comprises `colleagues` or classmates, respectively for active employed adults and school girl/boy children. Colleagues are uniformly drawn from the set of agents working at the same place, while classmates are uniformly drawn from the school with a limitation to agents with age difference less or equal to one. As for friendship, the number of colleagues and classmates are drawn from a gaussian distribution. Age category, gaussian mean and standard deviation for the two last layers can be adjusted using the dedicated parameter file (Population parameters.csv).

Synthetic agenda

Each agent of the model has its own set of timely organized activities, i.e. a synthetic agenda. The generation of agendas is based on demographic attributes of the agent, such as sex, age and employment status, and the specification of workdays and days off during the week. For employed agents and students, typical working days will be made of work and school activity, with starting, lunching and ending hours being determined uniformly within parametric ranges. Working agents can lunch inside or outside the workplace according to the given probability parameter. Retired individuals have an agenda full of leisure days. Individuals that are too young have an empty agenda.

The choice of the activities outside of work and study will depend on the age and gender of the individual. It is indeed possible to parametrize (through a CSV file) the fact that young people will, for example, favour leisure activities or elders will favour shopping, home and visit to friends activities. After regular working hours, agents with allowed age to engage in evening activities can enrol friends to do collective activities. During days off all the agents will be assigned a parametric and probabilistic number of activities (decreasing with age) with relatives and friends. All the leisure activities are determined using a weighted activity list according to the age group given in the file parameters, Activity type weights.csv. For each activity, a list of fellow Individual agents sharing the same activity can be defined to represent for example a group of friends and or colleagues eating at the same table in a restaurant.

Input Data

All the input data are summarized in **Table 1** and **Table2**.

Data file	Data Type	Description	Source
buildings.shp	GIS shapefile	All the building geometries, with type and flats attributes.	OpenStreetMap, Google Maps, and hand digitalization from Google satellite image. For Ben Tre, the initial data come from the Land Use map (DONRE, 2010).
boundary.shp	GIS shapefile	The considered area (e.g. commune) boundary.	GADM website: https://gadm.org/download_country_v3.html
satellite.png	GIS raster file	The background satellite image. (for visualization only)	https://www.bing.com/maps (or other satellite image provider)
population.csv	csv tabular file	The synthetic population generated from a sample using the Gen* library. Each line corresponds to a single individual with age, sex, and household id.	https://international.ipums.org/international/ https://www.gso.gov.vn/default_en.aspx?tabid=774
Epidemiological Parameters.csv	Csv tabular file	The set of epidemic parameters for the COVID-19.	Various sources from the literature (c.f. **Annex 1** for more details)

Table 1. Overview of the dataset (only the bold file names are mandatory)

Only three input data files are mandatory: buildings.shp, boundary.shp and Epidemiological Parameters.csv. The other ones can be omitted: as an example, when the synthetic population file is missing, COMOKIT will generate a population with an ad hoc generator. This generator (and this is the same for the activities and agendas generators) can be parametrized through additional files (c.f. **Table 2).**

Data file	Data Type	Description	Source
Population Records.csv	Csv tabular file	The mapping between variable in COMOKIT and variables values and names in population.csv	**Annex 2**
Population parameter.csv	Csv tabular file	The set of parameters to define the population of Individuals (used when no population.csv file is provided)	**Annex 2**
Activity parameter.csv	Csv tabular file	The set of parameters to define the activities of Individuals	**Annex 3**
Activity type weights.csv	Csv tabular file	According to the age (interval) and sex, the weight of the different activities	**Annex 3**
Building type weights.csv	Csv tabular file	According to the age (interval) and sex, the weight of the building type	**Annex 3**

Table 2. Overview of the files for additional parameters.

Synthetic population

The initialization of the simulation can take a csv file describing the synthetic population generated by the Gen* library (the population can have been generated by any other synthetic population generator, as soon it respects the organisation of the csv file). Each line corresponds to a single individual with age, sex, household id and employement status. It can be generated from IPUMS open access population sample file and any available marginals about modelers' case study. The underlying algorithm is based on Combinatorial Optimization to fit the sample of the real population on known local aggregated distribution of attributes.

Epidemiological parameters

The epidemiological parameter file is a table of parameters. For each of them, the following values are provided: (i) the name of the parameter, (ii) the age category lower bound (the upper bound will be defined according to the lower bound of the next age category of the parameter), (iii) whether the parameter value is given or if it has to be picked in a given probability distribution, (iv) its value (if of type given value) or the first parameter for the distribution, and (v) the second parameter (of the distribution).

Submodels

The epidemiological submodel

In our model, the disease-related state of the Individual agents follows a slightly modified SEIR model (44) using a final state machine structure (c.f. **Figure 3**). We considered the traditional Exposed compartment as a Latent compartment, where individuals are not yet infectious. We assumed the whole population starts the simulation in a **Susceptible** state (**S**) (as this is an emergent disease, nobody is immunised). When an Individual is in contact with an Infectious agent or located in an infected building, it can become infected and move to the **Latent** state (**L**), depending on the successful transmission rate. The successful transmission rate here is defined as the probability for one contact at a given step to be infected by an Individual. The *latent period* is defined as the time period between the exposure to the virus and the time the Individual becomes infectious, whereas the *incubation period* is the period between exposition and symptom onset. The Individual agent will stay in the Latent state for a given latent period computed according to the incubation period, and the *serial interval*, which is the period between the symptom onset of the newly infected individual and a future possible infection that it will cause. Due to presymptomatic infections, this serial interval can have a negative value, which is subtracted from the incubation period to obtain the possible latent period. In the case of having a positive value, we consider the latent period as equals to the incubation period (therefore, not using the serial interval), indicating infectivity of the Individual only once symptomatic (see **Figure 4**).

Once the latent period is expired, the Individual agent will move to one of the three infectious states (whereas the traditional SEIR model contains a single one): it can be **asymptomatic** (**Ia**), **presymptomatic** (**Ip**) or **symptomatic** (**Is**). If the serial interval value was negative, the Individual agent will be presymptomatic for a short period equals to the absolute value of the serial interval before moving to the symptomatic state, therefore having an incubation period equals to the latent period plus the presymptomatic period. The agent will stay in these states during the serial interval (for presymptomatic) and the infectious period for symptomatic and asymptomatic. The transmission risk of a susceptible individual will depend on being symptomatic or asymptomatic, and will be multiplied by a viral factor which is different for each Individual. We consider that asymptomatic and presymptomatic Individuals share the same transmission rate.

Once the infectious period is over, Individual agents reach the **Removed** (**R**) state, representing all individuals that have been infected, but are not infectious anymore. To represent deaths and recoveries, we decided to consider another variable, *clinical_status*, which represents the current clinical status of the Individual agent.

Individual agents begin with a clinical status set to not needing hospitalisation ($\mathbf{N_H}$) (see **Figure 5**). Individuals in the asymptomatic, latent or presymptomatic states will still not need hospitalisation. However, symptomatic Individuals have a probability of needing to be hospitalised (and thus move to the clinical status need_hospitalisation $\mathbf{H_N}$). Symptomatic individuals needing to be hospitalised have a probability of needing to be admitted in an Intensive Care Unit (ICU) (and thus to be in the clinical status need_ICU **H**$_I$), and therefore, die from the disease (corresponding to the clinical status dead **R**$_D$). If the Individual agent is not being taken to a hospital before the end of its expected period needing ICU, it will be considered as dead due to lack of treatment. In the case the agent will never need ICU and has not been taken to a hospital, it will be considered as recovered (**R**$_R$). For Individuals taken by hospitals, the hospital will decide on the clinical status according to the probability of dying for ICU cases. For hospitalised cases, they are considered recovered once they do not show any symptoms (i.e. not in the state symptomatic) and are tested negative for x consecutive days.

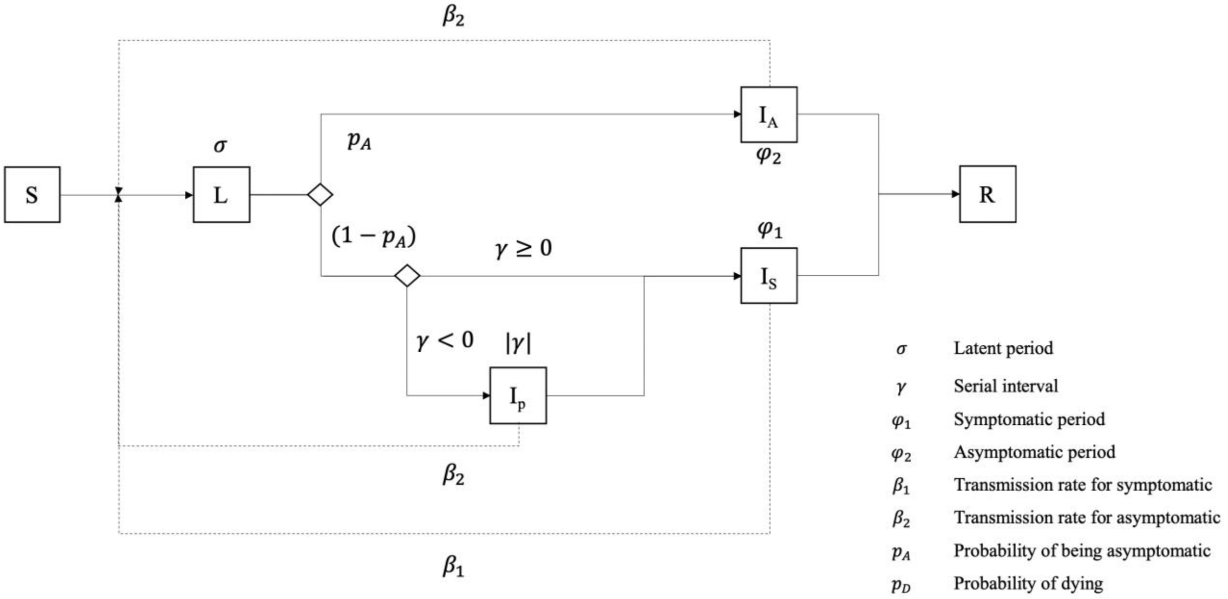

Figure 3. Epidemiological model of the Individual agent.

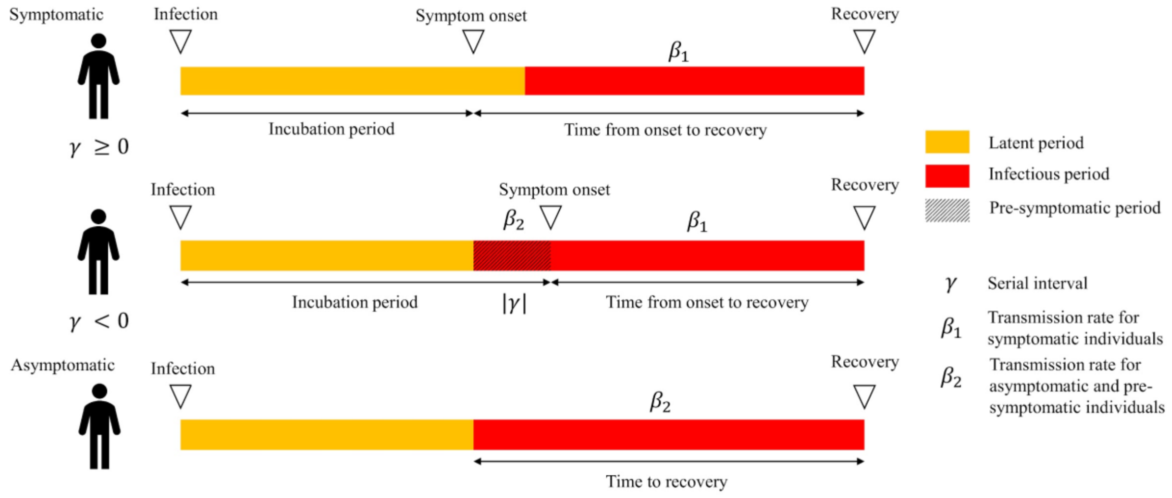

Figure 4. Illustration of the durations in different states (and the associated transmission rates) for both Symptomatic and Asymptomatic Individuals.

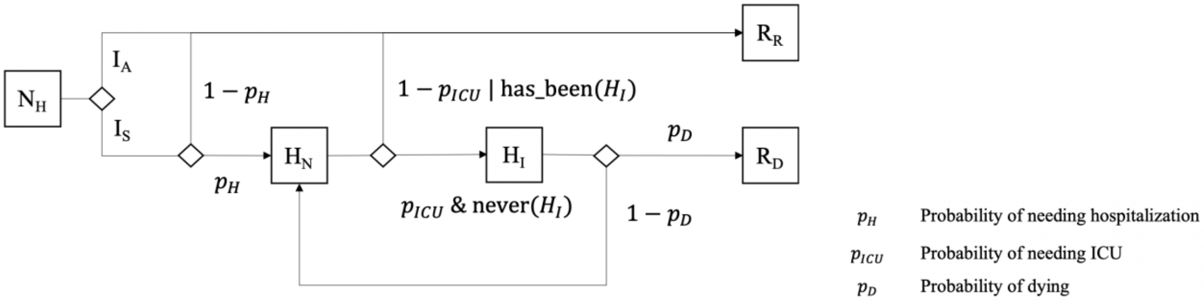
N_H
I_A
I_S
R_R
$1 - p_H$
p_H
H_N
$1 - p_{ICU}$ | has_been(H_I)
p_{ICU} & never(H_I)
H_I
p_D
$1 - p_D$
R_D
p_H Probability of needing hospitalization
p_{ICU} Probability of needing ICU
p_D Probability of dying

Figure 5. Hospitalisation model of the Individual agent.

The various incubation, serial and infectious periods are Individual dependent and are randomly picked following various distributions.

Given a lack of data, we made several assumptions in the model related to parameters (and in particular to their heterogeneity in the population):

- The incubation period does not differ for different age categories.
- The successful contact rate for human to human and environment transmission (i.e. the rate of Individuals becoming Latent after being in contact with an infectious one) does not differ for different age categories.

- The proportion of asymptomatic does not differ for different age categories.
- The successful contact rate does not differ during the infectious period.
- Asymptomatic and symptomatic individuals share the same infectious period distribution.
- The individual factor of the transmission risk does not differ for different age categories

Finally this model of the epidemic states evolution makes also several assumptions:

- The serial interval can be used as a proxy to get the presymptomatic part of the infectious period.
- Individuals could be infected by a contaminated environment, and for a maximal viral contamination in one building, the successful transmission rate is the same as one infectious Individual.
- Sex does not have an impact on the epidemiological model.
- The viral release of an individual in the environment (in our model, in Buildings) is the same for all infectious individuals.
- Presymptomatic and asymptomatic individuals share the same transmission successful contact rate.
- Masks do not deliver any protection, but rather reduce the successful contact rate of an infectious individual and its viral release in the environment.
- Recovered Individuals are totally immunised against the infection.
- Infection can lead to death only for Individuals expressing a need for intensive care.
- Testing is performed only for virus isolation, not antibodies, therefore, recovered people are not considered positive.

Daily Activities

Once weekly and daily agendas have been created at initialization, Individual agents have only to get, at each simulation step, the Activity corresponding to the current day and current hour, asks the Authority agent's authorization to perform it, finds a building associated with the Activity, and moves in it. In addition, the agent will ask the Authority for the number of individuals it can perform the Activity with. It will then pick randomly this number among its possible Activity fellowers, and store them during the Activity period. This set of agents will be used in the disease spread as possible infections with a higher probability.

The time is managed internally by the GAMA platform: from the simulation step duration set to 1 hour and a simulation starting date, the simulation will automatically compute the current day and hour.

Institutions

The Authority agent is in charge of applying one or several mitigation policies on the whole case study or on some local spaces. The policies can impact the simulation in two ways. Every step, the Authority can proactively perform some actions encoded in the policy, e.g. conduct a given number of tests on the population. On the other hand, each Individual agent asks the Authority whether it is allowed to execute a given Activity. In this case, the Authority will make its choice based on what is allowed by its policies that are currently applied.

Technically, any Policy agent has two main actions: apply() and is_allowed(Individual, Activity). *apply()* is called repetitively every time step by the Authority and can consist in anything relevant to this policy, for instance conducting tests. *is_allowed(Individual, Activity)* is called every time an Individual wants to undertake an Activity and should return whether this policy allows it or not. In addition, *is_active()* tells whether or not the policy is currently active (i.e. its conditions are fulfilled, for instance). It is used by some restricting policies.

We chose a modular approach to define policies: a policy is based on a small set of simple concrete policies (e.g. *DetectionPolicy* that allows all the activities, but conducts tests at each step or *ActivitiesListingPolicy* that restricts activities in a given set of allowed ones) that are composed using the composite (implemented through the *CompoundPolicy*) and nesting (through the *ForwardingPolicy*) design patterns:

- *CompoundPolicy* is a policy composed of a list of other policies. When applied, it applies the listed policies in sequence, and allows an activity for a given individual if and only if it is allowed by all the policies.
- *ForwardingPolicy* is a policy that embeds another policy. It can modify the nested policy allowance (for example SpatialPolicy restricts the application of its target policy in a given geographical space, and TemporaryPolicy in a limited time period).

The full class diagram of all the defined Policy species is shown in Figure 6.

Several Forwarding policies have been defined to to restrict or expand other ones:

- *PartialPolicy*: a policy that accepts, given a probability, a proportion of couples of individuals - activity normally forbidden by its target policy.
- *SpatialPolicy*: a policy that restricts the application of its target policy to a given geographical area. If outside, it allows everything.
- *TemporaryPolicy*: a policy that restricts the duration of its target policy. If before, or after, everything is allowed. This policy becomes active (i.e. delegates to its target and runs the countdown) only when its target has become active.
- *AllowedIndividualsPolicy*: a policy that allows certain individuals to undertake activities no matter what. They can be defined by a percentage or by an explicit map. Outside of the list, it asks its target policy for allowance.
- *CaseRangePolicy*: a forwarding policy that activates its target policy only after a number of reported cases and stops after another one. Outside of this range, everything is allowed.

Finally several concrete policies are available:

- *ActivitiesListingPolicy*: a policy described by a map of activities names and boolean values representing, for each, whether they are globally allowed or disallowed. Not explicitly disallowed activities are considered as allowed.
- *PositiveAtHome*: the policy that forbids any activity to any Individual agent who has been tested positive.

- *FamilyOfPositiveAtHome*: the policy that forbids any activity to any Individual agent who has been tested positive and whose any relative has been tested positive.
- *DetectionPolicy*: the policy used to conduct a given number of tests at every step. It allows everything.
- *HospitalisationPolicy*: The policy used to use hospitals for cure.
- *noPolicy*: an empty policy that allows any activity to everyone.

As an example, a “realistic lockdown” experiment has been created to apply a 60 days lockdown policy, where positive people are not allowed to move, others are only allowed to stay home or go shopping, and only 10% of the total population is allowed to go working. This policy is thus built as a *TemporalPolicy*, limiting the application of a *CompoundPolicy* in a period of 60 days. The nested policy is composed of:

- A policy applying a given number of tests at each simulation step (this policy allows any Activity).
- A policy forbidding any activity but shopping and going back home. This policy is nested in another policy that limits its application to 90 of the population (the remaining 10% are allowed to do any activities).
- A policy forbidding any activity to Individuals who have been tested positive.

The associated GAML code is the following one:

```
ask Authority {
	AbstractPolicy d <- create_detection_policy(number_of_tests_, false, true);
	AbstractPolicy l <- create_lockdown_policy_except([act_home, act_shopping]);
	AbstractPolicy p <- create_positive_at_home_policy();
	l <- with_percentage_of_allowed_individual(l, percentage_);
	l <- during(l, 60);
	policy <- combination([d, p, l]);
}
```

Note that the policy used by the authority can be dynamically adapted or changed anytime during the simulation.

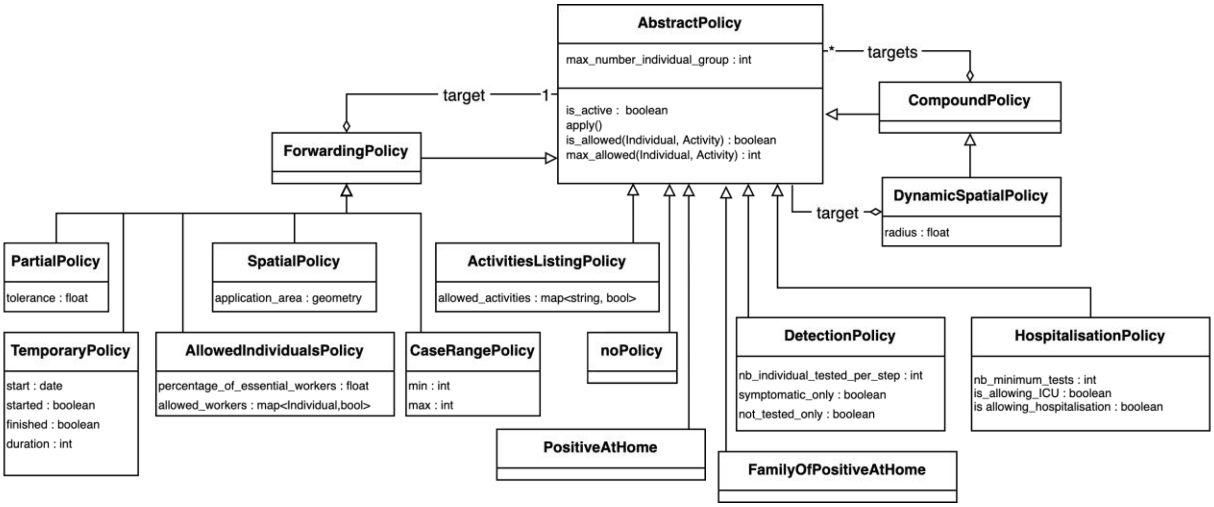

Figure 6. Full class diagram of the built-in policies.

Annex 1: Description of epidemiological parameters and input data related to COVID-19 used in the COMOKIT model

We make here a distinction between **parameters** (whose values are modified to define an experiment scenario) and **input data** (whose values are given from the literature).

Parameters

The parameters can either be used to define an exploration (e.g. to compare epidemics with and without transmission in buildings), or are only assumed and need to be investigated through a sensitivity analysis.

Parameter	Detail	Parameter_1	Parameter_2	Used for
Transmission_human	Fixed	true		Exploration
Transmission_building	Fixed	true		Exploration
Successful_contact_rate_building	Fixed	0.170071		Sensitivity
Basic_viral_release	Fixed	3		Sensitivity
Basic_viral_decrease	Fixed	0.01375		Sensitivity
Proportion_wearing_mask	Fixed	0		Exploration
Reduction_wearing_mask	Fixed	0.5		Exploration
proba_outside_contamination_per_hour	Fixed	0.0		Sensitivity

Input data

Parameter	Age	Detail	Parameter_1	Parameter_2	Region	Source
Successful_contact_rate_human	0	Fixed	0.034014			Derived from R0
Reduction_asymptomatic	0	Fixed	0.45		China	(1)
Proportion_asymptomatic	0	Fixed	0.28		Japan cruise	(2)
Probability_true_positive	0	Fixed	1.0			(3)
Probability_true_negative	0	Fixed	0.91			(3)
Incubation_period_symptomatic	0	Lognormal	1.57	0.65	China, Shenzhen	(4)
Incubation_period_asymptomatic	0	Lognormal	1.57	0.65		Assumed
Serial_interval	0	Normal	3.96	4.75	China not Hubei	(5)
Proportion_hospitalization	0	Fixed	0.025		USA	(6)
Proportion_hospitalization	20	Fixed	0.208		USA	(6)
Proportion_hospitalization	45	Fixed	0.283		USA	(6)
Proportion_hospitalization	55	Fixed	0.301		USA	(6)
Proportion_hospitalization	65	Fixed	0.435		USA	(6)

Proportion_hospitalization	75	Fixed	0.587		USA	(6)
Proportion_hospitalization	85	Fixed	0.703		USA	(6)
Proportion_icu	0	Fixed	0		USA	(6)
Proportion_icu	20	Fixed	0.2019		USA	(6)
Proportion_icu	45	Fixed	0.3675		USA	(6)
Proportion_icu	55	Fixed	0.3721		USA	(6)
Proportion_icu	65	Fixed	0.4322		USA	(6)
Proportion_icu	75	Fixed	0.5281		USA	(6)
Proportion_icu	85	Fixed	0.4125		USA	(6)
Proportion_death_symptomatic	0	Fixed	0		USA	(6)
Proportion_death_symptomatic	20	Fixed	0.0476		USA	(6)
Proportion_death_symptomatic	45	Fixed	0.0769		USA	(6)
Proportion_death_symptomatic	55	Fixed	0.2321		USA	(6)
Proportion_death_symptomatic	65	Fixed	0.2606		USA	(6)
Proportion_death_symptomatic	75	Fixed	0.3387		USA	(6)
Proportion_death_symptomatic	85	Fixed	0.9414		USA	(6)
Infectious_period_symptomatic	0	Lognormal	2.8622	0.0685	China, Shenzhen	(4)
Infectious_period_symptomatic	10	Lognormal	2.949688	0.094	China, Shenzhen	(4)
Infectious_period_symptomatic	20	Lognormal	2.95491	0.047	China, Shenzhen	(4)
Infectious_period_symptomatic	30	Lognormal	2.95491	0.033	China, Shenzhen	(4)
Infectious_period_symptomatic	40	Lognormal	3.072693	0.04	China, Shenzhen	(4)
Infectious_period_symptomatic	50	Lognormal	3.109061	0.037	China, Shenzhen	(4)
Infectious_period_symptomatic	60	Lognormal	3.131137	0.039	China, Shenzhen	(4)
Infectious_period_symptomatic	70	Lognormal	3.113515	0.08	China, Shenzhen	(4)
Infectious_period_asymptomatic	0	Lognormal	2.8622	0.0685		Assumed
Infectious_period_asymptomatic	10	Lognormal	2.949688	0.094		Assumed
Infectious_period_asymptomatic	20	Lognormal	2.95491	0.047		Assumed
Infectious_period_asymptomatic	30	Lognormal	2.95491	0.033		Assumed
Infectious_period_asymptomatic	40	Lognormal	3.072693	0.04		Assumed
Infectious_period_asymptomatic	50	Lognormal	3.109061	0.037		Assumed
Infectious_period_asymptomatic	60	Lognormal	3.131137	0.039		Assumed
Infectious_period_asymptomatic	70	Lognormal	3.113515	0.08		Assumed
Onset_to_hospitalization	0	Lognormal	1.23	0.79		
Hospitalization_to_ICU	0	Normal	7	5.9	China, Wuhan	(7)
Stay_ICU	0	Normal	8	5.9	China, Wuhan	(8)

References

1. Li R, Pei S, Chen B, Song Y, Zhang T, Yang W, et al. Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV2). Science [Internet]. 2020 Mar 16 [cited 2020 Apr 4]; Available from: https://science.sciencemag.org/content/early/2020/03/24/science.abb3221

2. Mizumoto K, Kagaya K, Zarebski A, Chowell G. Estimating the asymptomatic proportion of coronavirus disease 2019 (COVID-19) cases on board the Diamond Princess cruise ship, Yokohama, Japan, 2020. Euro Surveill. 2020 Mar;25(10).

3. Lassaunière R, Frische A, Harboe ZB, Nielsen AC, Fomsgaard A, Krogfelt KA, et al. Evaluation of nine commercial SARS-CoV-2 immunoassays. medRxiv. 2020 Apr 10;2020.04.09.20056325.

4. Bi Q, Wu Y, Mei S, Ye C, Zou X, Zhang Z, et al. Epidemiology and Transmission of COVID-19 in Shenzhen China: Analysis of 391 cases and 1,286 of their close contacts. medRxiv. 2020 Mar 27;2020.03.03.20028423.

5. Nishiura H, Linton NM, Akhmetzhanov AR. Serial interval of novel coronavirus (COVID-19) infections. Int J Infect Dis. 2020 Mar 4;

6. CDCMMWR. Severe Outcomes Among Patients with Coronavirus Disease 2019 (COVID-19) — United States, February 12–March 16, 2020. MMWR Morb Mortal Wkly Rep [Internet]. 2020 [cited 2020 Apr 21];69. Available from: https://www.cdc.gov/mmwr/volumes/69/wr/mm6912e2.htm

7. Yang X, Yu Y, Xu J, Shu H, Xia J, Liu H, et al. Clinical course and outcomes of critically ill patients with SARS-CoV-2 pneumonia in Wuhan, China: a single-centered, retrospective, observational study. The Lancet Respiratory Medicine [Internet]. 2020 Feb 24 [cited 2020 Apr 21];0(0). Available from: https://www.thelancet.com/journals/lanres/article/PIIS2213-2600(20)30079-5/abstract

8. Zhou F, Yu T, Du R, Fan G, Liu Y, Liu Z, et al. Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China: a retrospective cohort study. The Lancet. 2020 Mar 28;395(10229):1054–62.

Annex 2: Description of parameters related to the synthetic population generator in the COMOKIT model

The Individual synthetic population can be created through the Gen* library and read by the COMOKIT model from the file ***population.csv***. When using an external synthetic population, modelers must specify in ***Population Records.csv*** how to translate variables into COMOKIT Individual attributes. If this file is missing, the COMOKIT model will use its ad hoc synthetic population generator to generate the population. The document describes the parameters that can be modified to influence the generated population.

Population records file

The following file snippet illustrates the way to customize the link between external synthetic population and COMOKIT.

var	var_name	var_val	var_map	var_val	var_map	var_val	var_map
age	AGE						
sex	SEX	0	1	1	2		
is_unemployed	EMPSTAT	0	true	1	false	2	true
household_id	parentId						

Population generation parameter file

The following parameters are stored in the optional file ***Population parameter.csv***.

Parameter name	Definition	Type [Range]	Default value
proba_active_family	Probability for a household to be a family (with children)	Real [0,1.0]	0.95

proba_work_at_home	Probability for an individual to work at home	Real [≥0]	0.1
proba_unemployed_M	Probability for an M individual to be unemployed	Real [≥0]	0.03
proba_unemployed_F	Probability for a F individual to be unemployed	Real [≥0]	0.03
number_children_mean	Mean number of children per family	Real [≥0]	2.0
number_children_std	Standard deviation of the number of children per family	Real [≥0]	0.5
number_children_max	Max number of children per family	Integer [≥0]	3
proba_grandfather	Probability to have a grandfather in the family	Real [0,1.0]	0.2
proba_grandmother	Probability to have a grandmother in with the family	Real [0,1.0]	0.3
max_age	Maximal age considered	Integer [≥0]	100
nb_friends_mean	Mean number of friends living in the considered area per Individual	Real [≥0]	5.0
nb_friends_std	Standard deviation of the number of friends living in the considered area per Individual	Real [≥0	3.0
nb_classmates_mean	Mean number of classmates with which an Individual will have close contact	Real [≥0]	10.0
nb_classmates_std	Standard deviation of the number of classmates with which an Individual will have close contact	Real [≥0	5.0
nb_work_colleagues_mean	Mean number of work colleagues with which an Individual will have close contact	Real [≥0]	5.0
nb_work_colleagues_std	Standard deviation of the number of work colleagues with which an Individual will have close contact	Real [≥0]	3.0

Annex 3: Description of parameters related to synthetic agenda generator in the COMOKIT model

This document describes all the parameter files that can be modified to influence the way agendas are generated by the COMOKIT built-in generator.

Agenda generation parameter file

The following parameters are stored in the optional file ***Activity parameter.csv***.

Parameter name	Definition	Type [Range]	Default value
non_working_days	List of non working days (1: Monday, 7: Sunday)	List of integers [1,7]	[7]
work_hours_begin_min	beginning working hour: min value	Integer [0,23]	6
work_hours_begin_max	beginning working hour: max value	Integer [0,23]	10
work_hours_end_min	Ending working hour: min value	Integer [0,23]	15
work_hours_end_max	Ending working hour: max value	Integer [0,23]	18
school_hours_begin_min	beginning school hour: min value	Integer [0,23]	7
school_hours_begin_max	beginning school hour: max value	Integer [0,23]	9
school_hours_end_min	Ending school hour: min value	Integer [0,23]	15
school_hours_end_max	Ending school hour: max value	Integer [0,23]	18
first_act_hour_non_working_min	For non working day (or people who are not working), min hour for the beginning of the first activity	Integer [0,23]	7
first_act_hour_non_working_max	For non working day (or people who are not working), max hour for the beginning of the first activity	Integer [0,23]	10

lunch_hours_min	min hour for the beginning of the lunch time	Integer [0,23]	11
lunch_hours_max	max hour for the beginning of the lunch time	Integer [0,23]	13
max_duration_lunch	max duration (in hours) of the lunch time	Integer [1,23]	2
max_duration_default	max duration (in hours) of other activities	Integer [1,23]	3
max_duration_lunch	max duration (in hours) of the lunch time	Integer [1,23]	2
min_age_for_evening_act	min age of individual to have an activity after school	Integer [≥0]	13
nb_activity_fellows_mean	Mean number of fellows per activity	Real [≥0]	3.0
nb_activity_fellows_std	Standard deviation of the number of fellows per activity	Real [≥0]	2.0
max_num_activity_for_non_working_day	max number of activities for non working day	Integer [≥0]	4
max_num_activity_for_unemployed	max number of activities per day for unemployed individual	Integer [≥0]	3
max_num_activity_for_old_people	max number of activities per day for old people	Integer [≥0]	3
proba_activity_evening	probability for people to have an activity after work	Real [0,1.0]	0.7
proba_lunch_outside_workplace	probability to have lunch outside the working place (home or restaurant)	Real [0, 1.0]	0.5
proba_lunch_at_home	if lunch taken outside the working place, probability of having lunch at home	Real [0, 1.0]	0.5
proba_work_outside	proba for an individual to work outside the considered area	Real [0, 1.0]	0.0
proba_go_outside	proba for an individual to do an activity outside the considered area	Real [0, 1.0]	0.0
building_neighbors_dist	used by "visit to neighbors" activity (max distance in meters of neighborhood).	Real [≥0]	500

Activity type weight parameter file

These parameters provide, according to the Individual's age and the sex category, the weights of the different types of activity (the higher the weight, the higher the chance to carried out this activity). These parameters are stored in the file ***Activity type weights.csv***.

Example of file:

Age min	**Age max**	**Sex**	**Visiting neighbor**	**Visiting friend**	**eating**	**shopping**	**leisure**	**sport**
0	10	0	1.0	1.0	0.5	0.5	2.0	1.0
0	10	1	1.0	1.0	0.5	0.5	2.0	0.5
11	18	0	0.1	0.2	1.0	1.0	3.0	2.0

Building type weight parameter file

These parameters provide, according to the Individual's age and the sex category, the weights of the different types of building to carry out an activity (the higher the weight, the higher the chance to carried out an activity in this type of buildings). These parameters are stored in the file ***Building type weights.csv***.

Example of file:

Age min	**Age max**	**Sex**	**Playground**	**Parc**	**Cinema**	**place_of_worship**
0	10	0	3.0	3.0	0.5	0.1
0	10	1	3.0	3.0	0.5	0.1
11	18	0	2.0	2.0	2.0	0.2