# Architectures for Modelling Covid-19

Robert Worden

UCL Theoretical Neurobiology Group

rpworden@me.com

May 2020

**Abstract**:

Modelling the spread of the Covid-19 virus, and assessing the social and economic impact of containment measures, are essential tools to recover economies and social activity as fast as possible. There are many efforts worldwide to develop Covid-19 modelling tools. Now is the time to consider the architecture of these tools - and to get the architectures right before they are set in stone.

This paper proposes a set of requirements for Covid-19 modelling architectures, and discusses what modelling tools and frameworks can meet those requirements.

If we can get the architecture decisions right in this period of rapid development, the modelling tools will make a major contribution to society, supporting the fastest possible economic recovery. We should avoid repeating some costly mistakes which have been made in healthcare IT - such as supplier lock-ins which have been detrimental to the provision of healthcare worldwide.

The best possible outcome would be a thriving competitive market in local Covid-9 tracking and modelling frameworks, supporting rapid and precise response to local outbreaks, the maximum of possible economic activity, and personal risk assessment apps to help citizens manage their lives. To reach this outcome, the right architectural choices need to be made now - including open exchange of models and forecasts, to support peer review and competition.

Suggestions are made for required features of modelling frameworks - such as configurability and open-endedness to run a wide range of models, testability and self-test, ability to federate local models, and interoperability. No single modelling framework will meet all requirements, and diverse competing frameworks are needed. To stimulate diversity, there can be Covid-19 modelling competitions (challenges) like those that have worked in speech recognition, robotics and image understanding.

Components of the architecture are illustrated by a modelling framework which is available in Open Source on GitHub.

**Keywords**:

Covid-19 modelling; architecture of models; containment measures; economic and social modelling; tracking disease data; federated local models; diversity of models and modelling frameworks; interoperability; modelling competitions; commercial lock-in

## 1. Introduction

This paper discusses the architecture of programs used to model the spread of the Covid-19 virus, and its social and economic impacts. These models are now a vital tool in recovering from the pandemic and in recovering economic and social activity. There are worldwide efforts to develop and apply these tools [e.g. 1]. Now is the time to ensure that the tools are built with appropriate architectures, to support the fastest possible recovery of economies and societies.

IT architectures should be shaped by requirements. Key requirements for Covid-19 modelling tools are emerging:

- Tools need to model the (a) spread of the virus, (b) the measures used to contain it, and (c) their economic and social impact, in an integrated framework.
- Accuracy and responsiveness of models is paramount.
- The best way to ensure accuracy is continually to calibrate models against current data
- Tracking the progress of the disease, modelling and forecasting should be closely integrated - to keep models close to the data
- Local and regional models are more responsive than national models, and better attuned to local conditions. It is easier to ensure data quality and timeliness at a local level.
- Models and their outputs should be accessible for peer review and challenge.
- There needs to be a vigorous competitive market in modelling tools and frameworks, free of the commercial lock-ins that have been so detrimental in healthcare IT.
- Modelling frameworks should be generic, configurable and interoperable, able to re-run models from other sources.

These requirements have consequences for modelling architectures, which are explored in this paper. If, at this time of rapid ferment and development of modelling tools, we can get the architecture right, there will be enormous benefits for society, in accelerating social and economic recovery. Nations which get it wrong will be trapped in recession and lockdown for longer.

The structure of the paper is a sandwich:

a. The paper discusses architectural requirements and architectures in general terms (sections 2 - 4)
b. The architectural issues are illustrated by reference to a specific modelling framework, which is available for download and experimentation (sections 5 - 12)
c. In the light of this illustration, the architecture issues are revisited (sections 13-17).

The modelling framework used to illustrate the architecture issues is available in runnable or source form at https://github.com/robertworden/Covid-Modelling.

Developers of modelling tools may wish to download the modelling framework , explore it to hone their understanding of the architectural issues, and bear these issues in mind for the tools they are developing.

## 2. Requirements for a Covid Modelling Architecture

This section sets down a list of requirements for a Covid modelling Architecture. By 'Architecture' I mean a large-scale architecture - the set of modelling applications which is used in any nation, or possibly across nations.

The list is intended to start discussion, and you will not agree with all of it. Please improve the list.

1. **Tracking and modelling should be closely integrated:** Economic and social recovery will depend on good data about disease progression; on the quality of models that use those data; and on the reliability of modelling forecasts. The best way to improve models is continually to compare them with data. With widespread testing and contact tracing, there will be plenty of data; rapid response to disease data is needed to control renewed outbreaks. Modelling architectures should track those data in as close to real time as possible, continually adjusting the models to fit recent data, to make them reliable and responsive - to help nip outbreaks in the bud, while allowing the maximum of economic and social activity.
2. **Tracking and modelling tools should address (a) the spread of Covid 19, (b) the effectiveness of containment measures, and (c) their social and economic impact:** The key question now facing every society is: what containment measures are effective in preventing further outbreaks, and what is their cost in terms of economic and social life? These tradeoffs are complex, and cannot be made without good modelling forecasts of impact. Modelling the disease on its own, or even in the presence of containment measures, is not enough. Models need to encompass the economic and social impacts. To assess the tradeoffs, all three sets of variables (disease, containment measures, social/economic impact) need to be integrated in the same models, so they can be varied together in 'what if' scenarios
3. **Federated local models work better than national-level models:** Good models are data-

hungry. If the first requirement (integrated tracking and modelling) was met only at national level, it would require heroic efforts of national data collection and data quality management; a central team would have little feel for local variability; and the feedback loop from events to models would be extended in time - making the system ponderous and unresponsive. The task of gathering near-real-time data about the progression of the disease (and about economic and social impacts) is much more tractable at local level; local people can understand local conditions, in healthcare and in society [2]. It follows that local models will be of better quality, and have quicker response to events, than national models. The best national model will be a federation of local models.

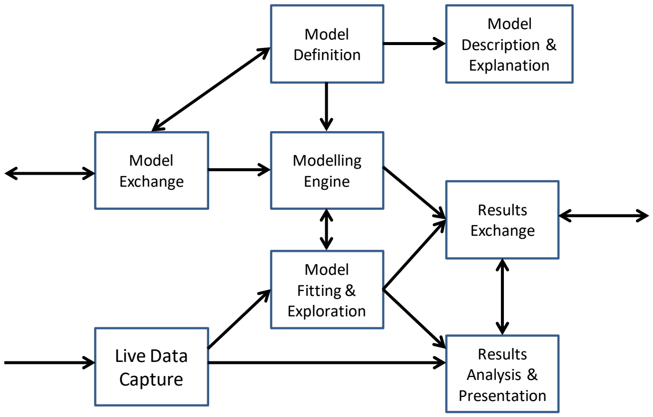
4. **Modellers should be able to choose between models:** There are currently two main flavours of epidemiological model - analytic population-level models, and stochastic agent-based models. There are variants within each flavour. It is not yet clear what works best in different contexts. The important point is that any modeller should have a choice of models, and should be able to choose which model to use, based on the specific context and requirements. This implies that modelling frameworks and tools should be generic, and easily configurable to run different models; so that users can switch to whatever model best fits the data, or is most effective, without switching tools.
5. **Model developers and users should have a choice of modelling frameworks:** Models and their predictions should be accessible not just to professional model developers, but to others who will use the model predictions to make decisions, or to critique and debate the consequences of decisions. Models and their results should be accessible (or at least demonstrable) to lay people. The best route to this is competition. Just like the 80-100 current vaccine development efforts, the current ferment of experiment and development in Covid modelling frameworks will produce a few winners - but preferably, not just one winner; because if it does, competition stops. Healthcare IT is rife with local monopolies and supplier lock-in. We can learn from that example, and avoid the lock-ins, which would only prolong lockdown
6. **Any modelling framework should be able to re-run models from other sources**: With rapidly improving data about the course of Covid-19, we can expect the epidemiological component of models to improve and become more reliable. That requires peer review of models and their projections - which works best if modellers can re-run other peoples' models, to check them and subject them to sensitivity analyses. Equally important for society are the economic and social predictions from models. Here, we should not expect the same level of scientific consensus to emerge, and vigorous debates needs to take place about the relative merits of different containment policies or release strategies. These debates are best conducted if the models which drive them are open, transparent, and re-runnable.

## 3. Architecture to Meet the Requirements

The previous section discussed requirements for Architecture with a capital A - a whole set of applications across a nation which meet its modelling and tracking requirements.

This section discusses architecture with a small 'a' - the architecture which an individual modelling tool or framework needs to have, if it is to fit into the large-scale Architecture. A national Architecture could consist (mainly) of a federated set of local modelling frameworks or tools, each with an architecture as described here.

An architecture for a local modelling framework is proposed below. It contains architectural components (boxes in the diagram) needed to meet the requirements of the previous section.


Model Definition
Model Description & Explanation
Model Exchange
Modelling Engine
Results Exchange
Model Fitting & Exploration
Live Data Capture
Results Analysis & Presentation

There follow descriptions of the components. These will then be amplified by reference to a small open source demonstrator modelling framework[1]:

- **Model Definition:** This component includes the tools needed to define a model. As many epidemiological models are defined by comparatively small numbers of parameters (e.g. a few disease states and the transitions between them), the tools needed to define those models

[1] In some cases, the framework may demonstrate how an architecture component can be implemented. For other components, the implementation is primitive or incomplete, and the gaps serve to illustrate the issues involved.

may be rather simple; but requirements will become more complex as models are refined. For the social and economic facets of models, diverse tool support may be required - for instance, to define measures of types of economic activity, and the dependence of one type of economic activity on another. To forecast the economic impact of virus containment measures, all these components of a model will be required.

- **The Modelling Engine** is a central component of the architecture, and has received the most attention so far. This component needs to be able to run a model - to project it forward in time to forecast outcomes. Crucially, it is not just an epidemiological model. It is required to forecast (and in order to do so, to model) the social and economic impacts of virus containment measures such as lockdowns and contact tracing. These impacts are the urgent questions that societies are rushing to answer. Therefore models need to include models of containment measures, and of social and economic activity. Modelling engines need to be generic and driven by model definitions, in to be able to run a variety of models.
- **Model Exchange:** In order for one modelling framework to run models built in another framework, there needs to be a common model interchange data format, and each modelling framework needs a component to interface between the model exchange format and the internal model definition it uses in its modelling engine. This will be important when a national modelling capability is built with federated regional and local modelling tools; a network of modelling centres will need to share changing models. It is also needed to interface with more detailed models, such as detailed models of air flow and virus diffusion, or models of group behaviour. Open standards for exchange of model definitions will help to avoid one kind of lock-in - to tools using some inaccessible or proprietary model definition.
- **Model Description and Explanation:** Whenever model outputs are to be inspected, reviewed or relied upon by people who did not develop the model - whether they are other modellers, or non-specialists - those people need to understand the basis of the model and its assumptions. There needs to be a part of the user interface which presents and explains the definition of the model in use.
- **Live Data Capture:** It is essential to calibrate models against data on the spread of the disease, and to keep that calibration up to date. For this, an integrated tracking and modelling framework is required, with capture of disease data in as near real time as possible, and tools to measure and manage data quality - before fitting models to the data. The data will include data from randomised testing and from contact tracing, as well as economic and social data. This will require interfaces with a wide variety of data formats - such as HL7 Version 2 for test results, HL7 FHIR for other healthcare data, and diverse data formats for demographic, social and economic data. To avoid GIGO, data quality tools are an essential part of this component.
- **Model Fitting and Exploration:** Exploring model outputs with different input assumptions and parameters is required for many purposes, including: (a) fitting model predictions to recent data about disease progression, (b) sensitivity analyses assessing how strongly model outputs depend on input assumptions, (c) trying out different combinations of containment measures, (d) comparing how different containment measures perform, or (e) understanding the range of uncertainty in forecasts. For these purposes, it is necessary to run the modelling engine repeatedly under the control of another component - here called the model fitting and exploration component - which compares the results of different runs, for instance to find the best fit to recent sample data
- **Results Analysis & Presentation:** All modelling results are ultimately for human consumption, so tools for analysis and presentation of results to many different audiences are essential. Tools and insights from business intelligence and data visualisation have a role to play.
- **Results Exchange:** free exchange of model outputs will play an important role - for instance in peer review and critiquing of models, or in federated networks of local models. Now is the time to define open standards for exchange of modelling outputs, so to avoid another form of lock-in.

Architectures can be described in many ways. There are dimensions of description of this architecture, such as the common multi-layer description of IT architectures (user interface/program logic/data & communication) which have been left out for simplicity. The multi-layer view of architecture will be revisited later in the paper.

## 4. Avoiding Lock-In

Two components of the architecture - model exchange and results exchange - involve the use of open standards to achieve interoperability and avoid software lock-in. It is worth briefly describing the harm that has been done to healthcare provision by proprietary IT lock-in.

Many industries such as finance, travel or e-commerce have converged on worldwide standards for information exchange, whose impact has been hugely beneficial - supporting worldwide banking networks, travel booking networks and so on. Compared with these industries, healthcare has conspicuously failed to achieve free exchange of information between IT systems. It is notoriously difficult for healthcare professionals of one specialty to find out even basic details of how their patients are being treated by other specialists, even in the same district or the same hospital. The UK NHS is still the world's largest user of fax machines.

The lack of connectivity can be partly blamed on the complexity of healthcare information. The human body and its ailments are more complex than finance, travel or retail. However, that is only a part of the problem. For more than forty years, healthcare providers have bought IT systems from suppliers whose commercial interests have been to protect their own markets, and who have had little interest in providing open data interfaces which would allow their customers to migrate to other suppliers. In procuring healthcare IT systems, the free exchange of information with other systems has been a low priority. As a consequence, a typical hospital has hundreds or thousands of IT systems which can hardly exchange information with each other, let alone with other healthcare providers; and it would be prohibitively expensive for it to change its IT systems and suppliers. Healthcare data and organisations are locked in to their IT suppliers [3].

Healthcare IT supply has been a competitive market in theory for over 40 years; but competition has largely failed to deliver results for customers (healthcare providers and patients) because of lock-in.

For more than the past twenty years, the suppliers of 'interoperability' (aka free information exchange) , and organisations such as Health Level 7 [6] and OpenEHR [3] which set standards for healthcare information exchange, have been playing catch-up - trying to achieve retrospectively the levels of free information flow which are needed for properly coordinated patient care.

It is generally agreed that while advocates of healthcare interoperability are trying to catch up with the reality of healthcare provision (and have scored some notable successes) they are not actually catching up. The target is moving away from them faster than they can run to catch it. The target moves because medicine is becoming more complex, personalised, and IT-dependent each year; and in many nations, more people are living longer and are suffering from multiple healthcare conditions. Joined-up healthcare is becoming harder to achieve each year. The cumulative damage to standards of patient care is hard to estimate; but the cost in lives is huge. It probably dwarfs the expected cost in lives of Covid-19..

Covid-19 is simpler than the whole of healthcare; from that point of view, it should be easier to achieve interoperability of Covid-19 modelling and tracking tools. However, these tools also need to model and track the impacts of the virus, and of containment measures on economic and social activity; to do that, a high level of complexity may be re-introduced. This is the time at which it would be easy to go wrong, and unknowingly introduce high levels of lock-in to the tools used for tracking and forecasting. Alternatively, a little forethought now about interoperability issues can have big benefits for society, in allowing different tools to work together effectively, avoiding lock-in and allowing the best tools to thrive and be widely adopted.

## 5. A Modelling Framework to Illustrate the Architecture

The architectural principles of the previous sections have been stated at a general level. They become more tangible if they can be seen in working software - which either illustrates how the principles can be applied and what they mean, or (if the software does not embody the principles) may clarify what needs to be done to apply them.

So the paper moves to the middle layer of the sandwich - a more detailed description of specific software to illustrate the architecture.

I have developed a small framework for modelling the spread of Covid-19, the measures to control it, and their impact on social and economic activity. The framework is available in Open Source on Github at https://github.com/robertworden/Covid-Modelling.. It can be downloaded and run to help understand how the architectural principles are applied (or on some cases, not applied) in running software.

This framework has been developed in a very short timescale, so many features are lacking or basic; for instance, its user interface is distinctly old-fashioned. However, the framework includes a capable agent-based modelling engine. By design it is open-ended and data-driven, having the flexibility to model both disease spread and social and economic factors to increasing levels of detail. Its purpose is to illustrate and clarify the architectural issues of this paper.

The framework allows the variables in all three groups - population, disease progression, and control measures - to be controlled and varied. The components of these groups are described below.

Models are run on the framework in the following stages:

- The program reads in data files to define static details of the population, the parameters of progression of the disease, and the containment policies. This defines the state of the model on day 0 of the simulated timeframe, and defines how containment policies will be applied over simulated time.
- The model is forward, a day at a time. This uses a stochastic agent-based (Monte Carlo) simulation of encounters between people, constrained by the current containment policies. During these encounters the virus may be transmitted, with configurable probabilities. A statistical model simulates the progression of the disease in each person.
- There are repeated runs of the model, for several purposes:
  - varying parameters of a model, to fit results to known historic data about the disease in the modelled context
  - Comparing different scenarios, with different parameters of the disease containment policies, to explore the tradeoffs between costs and effectiveness of the policies
  - To assess a minimal degree of uncertainty in predictions, introduced by random processes of virus propagation in the real population
  - Sensitivity analyses, to see how strongly model outputs depend on their inputs.
- The results of the simulation runs are analysed and presented, to project out summary results and results for selected sub-populations, such as the costs and impact of the containment measures. Impact can be assessed in health, economic or social terms.

Models are simulated to the level of individual simulated people, because this is the level at which demographic data about populations, behaviour and economies intersects with data about the disease and its effects on individuals. At this level many statistical correlations are known, which may need to be represented to make a model realistic. Any higher level of aggregation in models would make it harder to realistically model the population, and to analyse the inter-dependences between factors. The next sections describe the three main components of models in the framework, and how the degree of realism in a model can be increased by providing more fine-grained data (and by using real data from real populations), and by modest extensions to the software.

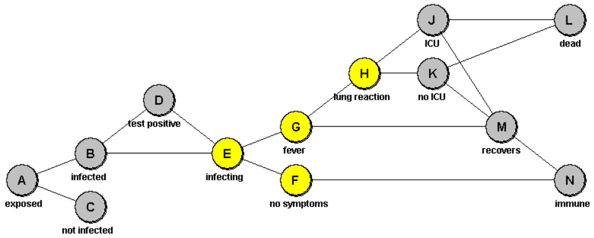
## 6. Disease Models

The **model definition** component of the architecture has three sub-components - defining the effect of Covid-19 on individuals; defining demographics and population behaviour, and defining containment measures. This section describes the first sub-component.

The progression of Covid-19 in each individual of a simulated population is modelled as a Finite State Machine, in which a person suffering from the disease progresses through a small set of states, with transition probabilities and durations of the states which are modelled statistically, to whatever level of detail is necessary.

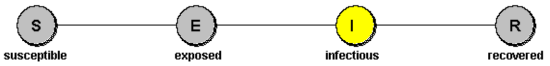
In epidemiological modelling terminology, the states of the finite state machine are compartments; common epidemiological models such as SEIR are denoted by acronyms for their compartments.

An illustrative finite state model of disease progression is shown in the diagram below- taken from the running framework:



The states marked in yellow are the states in which a person may infect others in public places. This finite state model and its parameters are not yet based on specialist medical knowledge, but have been used to test the modelling framework.

The model shown above is more complex than commonly used epidemiological models such as the SEIR model. The framework can be used to run these models. A SEIR model is shown below in the framework user interface:

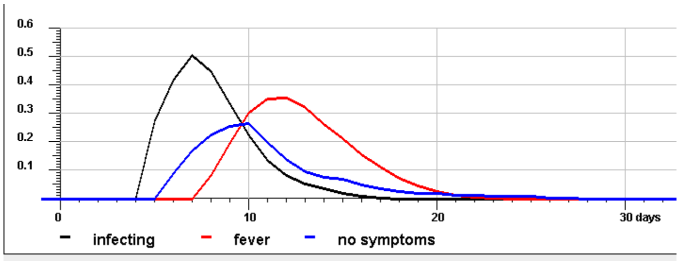


For each state in the model, small data files define:

- The probabilities of transitions to other states (e.g. the probability going from 'fever' to 'recovered')
- The probability distributions of each transition taking different elapsed times, in days.

These probabilities and elapsed times can depend on many characteristics of a person, such as their age, and co-morbidities. In the model used here for illustration, only the dependence on age has been modelled.

Since models define the times needed for transitions between states, the framework can compute the probability of being in each state, as a function of days after exposure to the virus. Some (illustrative) results of this computation for a few states are shown in the diagram below:



Model parameters should be tuned so the curves match actual data about the disease and its spread.

Finite state diagrams like those above, and charts of disease state progression like that above, are part of the **model explanation** component of the architecture.

The modelling framework is agnostic about the states (compartments) of the disease model. As the framework is data-driven, it can be used with a complex finite state model like that in the first figure, or for a simpler model such as SEIR. As models like SEIR are widely used, one may  ask: why use more complex disease models? There are several possible reasons:

- One might want to experiment with variants of the disease model, adding states to see which model gives the best fit to medical data. In a generic modelling framework, these experiments can be made rapidly by changing the input data.
- It is necessary to model measures to contain the disease, and those measures may depend on disease progression in an individual in complex ways, requiring more states.
- It is required to model social and economic impacts of the disease, such as the workload on healthcare providers in primary or secondary care, or workload on contact tracing teams. To model these effects, more disease states may be needed.

Disease models are defined in small data files (comma-separated value files, or csv files) , so that the only tool required to define a disease model is an editor for csv files - such as a spreadsheet tool. This is a very basic approach to model definition, but it has benefits:

- It is highly flexible, allowing anybody to define disease models as simple or as complex as they need.
- It is generic, allowing the framework to capture and re-run models for other sources - without program changes
- csv files are simple to understand, and easy to read and edit, with no learning curve required

Common epidemiological models, such as variants of SEIR, may make assumptions and approximations about probabilities of disease state transitions and of delay times - for instance, assuming exponentially distributed transition times. These approximations may be needed to make the equations of population models soluble. As inputs to the stochastic modelling framework, such assumptions are optional. All input probability distributions are defined in data files (typically small data files), which can either be pre-calculated from simple mathematical forms, or derived from real-world data. This keeps the framework open-ended and generic, so it can be used to run other models, such as the common analytic models, or tuned to fit actual data.

## 7. Demographic Models

Demographic models are the second sub-component of the **model definition** part of the architecture.

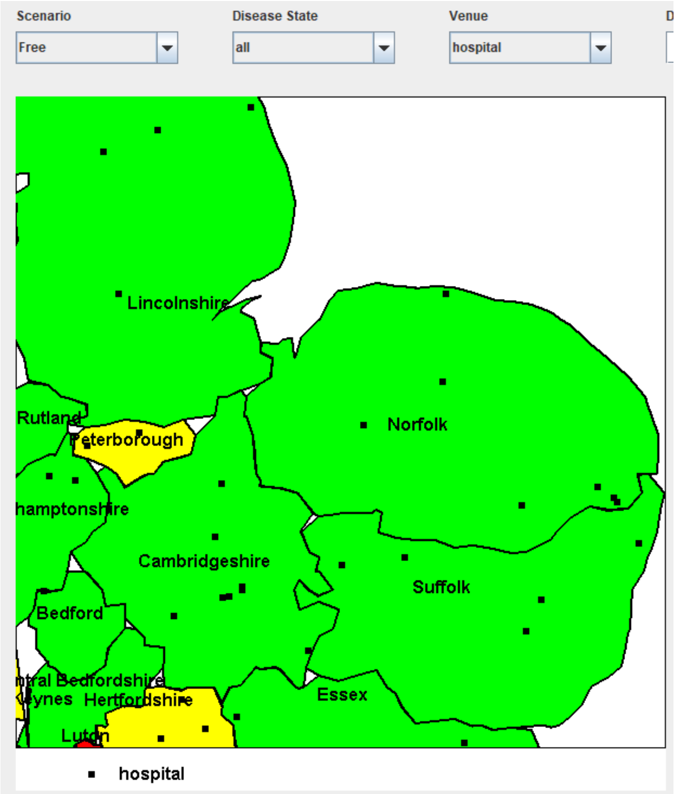
The demographic model currently consists of the following entities.

- A **Territory** - on a rectangular map
- **Regions** within the territory - each of which has a defined population density
- **Households**, which are placed within the regions, according to their population densities. Households have a statistical distribution of occupancies (number of people in each house)
- **People** , who live in the households and modelled with the usual properties - age, name, profession, gender and so on.
- **Meeting Places** (also called Encounter Groups) which have types such as workplace, school, hospital, or transport. Each type of meeting place has a density (number of meeting places per 1000 simulated population), a circular catchment area of the people who meet in it, and properties defining how many people visit it on average each day, and how many encounters they typically have when they visit it.
- **Potential Encounters**: which are the encounters between people which would take place if there were no virus and no containment policies, generated from the data above.

The distributions of these demographic entities are defined either statistically or from real population data. If they are defined statistically, the driving parameters are input in csv files like those used for the disease model.

Different types of meeting groups can be used to define the levels of economic and social activities (such as manufacturing, services, education and healthcare), and the impact of containment measures on them.

A map display of the demographic model which was used when developing the framework is shown below:



Scenario: Free
Disease State: all
Venue: hospital

These map displays are part of the **model explanation** component of the architecture - and are also used as part of the **results presentation** component.

The regions of Eastern England are colour-coded according to their population density, taken from the UK Office of National Statistics. Hospitals have been placed at random within the regions - but they could have been defined by actual hospitals, had the data been available in a convenient form. This illustrates how the framework can be loaded with any mix of real data and randomly generated data.

As above, the program can show the locations of any type of meeting place on a map of the territory.

As for the disease model, the demographic elements of the model are defined in comma-separated value (csv) files. As before, this quite basic approach has benefits of transparency and flexibility - for instance, one can easily switch between randomly-generated pseudo-data (with small csv files defining the generation parameters) and real demographic data (larger csv files).

It is not yet clear how much actual geographic detail (such as precise locations of meeting places) is required to give useful results. Using only simulated meeting places rather than real data, the results of running a model may shed light on important policy issues (which may vary across regions) such as:

- The workload on hospitals and other healthcare providers, caused by people needing Covid-19 treatment, leading to extra mortality from other conditions
- The role of particular types of meeting place (such as schools) in propagating the virus
- Economic and social activity, derived from the numbers of encounters in meeting places of different types (schools, offices, shops, transport, etc.)
- The rate of spread of virus outbreaks from one region to another

The demographic model sets the stage for meetings between the people, through which the Covid-19 virus propagates. This is done by pre-generating a demographic model, including potential meetings in each day, and storing it as data, so that one instance of the demographic model can be reused to compare different policy choices for containment of the virus. Two points about the pre-stored demographic model:

- The initial distribution of the virus on day 0 is modelled by randomly seeding a proportion of the population into infected disease states.
- For every day that is to be modelled in the simulation, the stored population model contains the set of potential meetings between people. In an uncontrolled state, when all these meetings take place, the number of infected people grows exponentially. Containment policies are modelled by preventing certain types of meeting from taking place.

## 8. Models of Virus Containment Policies

This is the third sub-component of the **model definition** component of the architecture.

Models are run in a small number of **scenarios**. Each scenario is divided into a number of non-overlapping **periods**, which together cover a set of days 0..N over which the simulation runs. For each period, there is a small set of **containment policies**, which are applied during that period of that scenario. Then, the outcomes for different scenarios can be compared side by side in a range of graphs - illustrated below.

There are currently two types of containment policy:

- **Meeting policies**: which act to prevent or discourage potential meetings
- **Notify policies**: These are triggered by events (such as fever, or positive test results) happening to people, and cause notifications to be sent to them and to other people (e.g. asking them to be tested, or to self-isolate).

Both types of policy can have a range of parameters, set in csv data.

Meeting policies are used to model selective lockdowns. For instance they can be used to prevent or discourage meetings in certain types of meeting place, or of certain age ranges of people, or in certain regions.

Notify policies can be used to model contact tracing. When a person enters a certain disease state (such as fever), then that person and all the people which that person has met in the last several days can be notified that they need to self-isolate - e.g. until they are tested.

An important variable parameter for most policies, which is easily represented in models, is the level of adherence to the policy - how successful it is in altering people's behaviour.

Any number of policies of either type can be applied in any period of any scenario. This gives a way to compare side-by-side the results of different policy mixes, applied for different time periods, in the same graphical displays. (this involves other components of the architecture, particularly the **model fitting and exploration** component)

## 9. Running Models

This section describes how the **modelling engine** and **model fitting and exploration** components are realised in the framework.

Models predict the impact of containment policies by doing a day-by-day time step. In each day, the potential meetings (involving only the infectious people) are retrieved from the saved demographic model, and the policies in effect for that period of that scenario are applied to determine whether each meeting took place. If it did, with a certain probability the non-infected participant moves to an exposed or infected state. As time progresses, each infected person moves through the states of the disease model, with probabilities and delays as defined in the model.

Epidemiological models are usually run in two main ways - either an analytic, population based approach, or in a stochastic, agent-based approach. It can be shown that the two approaches give identical results in certain limits - either in the limit of infinite population (the thermodynamic limit), or in the limit of very many runs of the stochastic approach, taking the mean of the results. So for a pure epidemiological model, there is a choice. The modelling engine described here uses only a data-driven stochastic approach, for a number of reasons.

1. The stochastic approach is easily re-configured through its input data to run any disease model - whereas for analytic approaches it may be necessary, when changing the disease model, to change some equations and reflect the consequences of the changes in changed code.
2. Analytic approaches may rely on certain assumptions and approximations to make them tractable - whereas in a stochastic approach, any assumption is defined in data, so approximations can always be replaced by more realistic approximations defined in more fine-grained data.
3. The requirement is not just to model the spread of the disease, but also to model containment measures and their social and economic impact. For modelling these impacts, stochastic agent-based modelling is much more tractable and flexible than any analytic approach. Stochastic modelling of populations and their behaviour can be joined seamlessly with stochastic modelling of the disease.

Stochastic agent -based modelling (Monte Carlo) is a highly flexible technique which can be applied in almost any domain, and can be highly data-driven. A data-driven architecture is required to give the maximum flexibility and maximum potential level of realism - by making the data more fine-grained and realistic, with minimal changes to program code.

The modelling engine is not yet as data-driven as it could be. Further work is needed to make it possible to define any multi-variate probability distribution in data; and to introduce new types of modelled entity, and new attributes of modelled entity, entirely in data and without changes in code.

Since the propagation of the virus is a random process with a high degree of amplification, and the framework simulates this random process by Monte Carlo simulation, the simulation outcomes vary between different runs - reflecting the uncertainty of outcomes in the real world. When running any model, it is important to get a feel for this level of uncertainty. This is done by running a model in a set of identical runs (with the same inputs and different random events), and displaying the results by upper and lower quartiles (or percentiles). The spread between quartiles is a measure of the level of uncertainty of outcomes in the real world. This is an aspect of the **model exploration and fitting** component of the architecture.

## 10. Examining Model Results

This section describes how the **Result Analysis** components are realised in the demonstration framework..

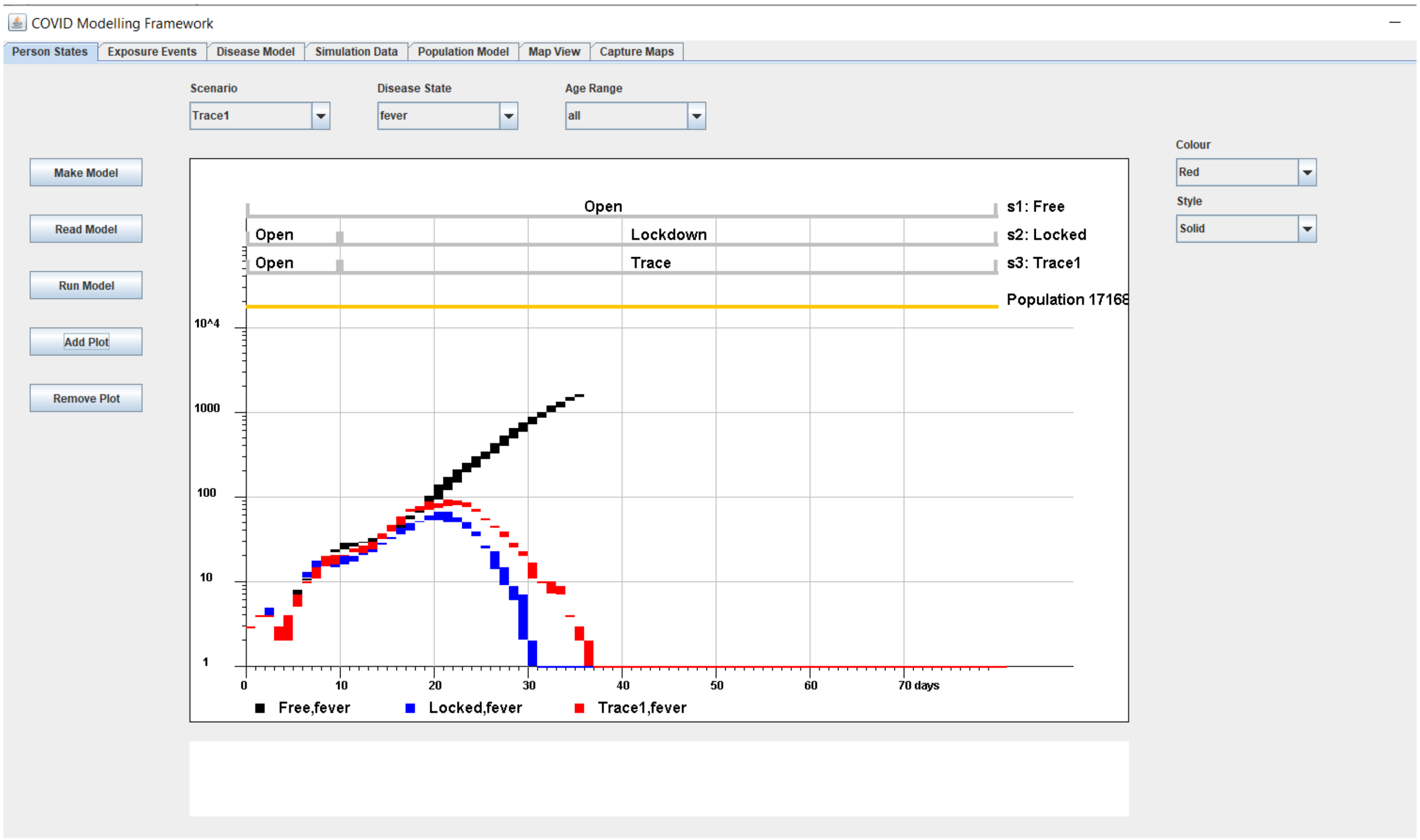
When a model has been run (through several identical runs of a set of scenarios, with periods and containment policies within each scenario) the simulation results can be displayed in many different ways. Some of these are shown below.

The next screen shows the result of running a model for 80 days on a population of 17000 people, in three scenarios - the first scenario with no controls; the second with a complete lockdown (no meetings in any public places) imposed after 10 days; and the third showing contact tracing, also used after 10 days . The graphs show the number of people over time in different disease states in the scenarios. The legend below the graph shows the meanings of the different symbols.

There is a choice of graphs which can be shown (populations in any scenario, in any disease state, in any age range), The graphs below show the numbers in the 'fever' state, as a measure of spread of the virus. All curves are on a logarithmic scale.

The black curve shows the exponential growth in the number of cases, which occurs in the absence of controls, and which depends on the reinfection rate R. The blue points show the impact of a complete lockdown (preventing all meetings in public places) and the red points show the effect of a  contact tracing regime.

These curves show only the impact of idealised containment measures. a key purpose of any modelling framework is to model more fine-grained or  less idealised containment measures. To do this, it is necessary to ensure that the model parameters are realistic, reflecting medical knowledge, realistic demographics, and (if data are available for the context) fitting recent data on disease spread. Only then can a model make predictions with confidence.



Scenario: Trace1

Disease State: fever

Age Range: all

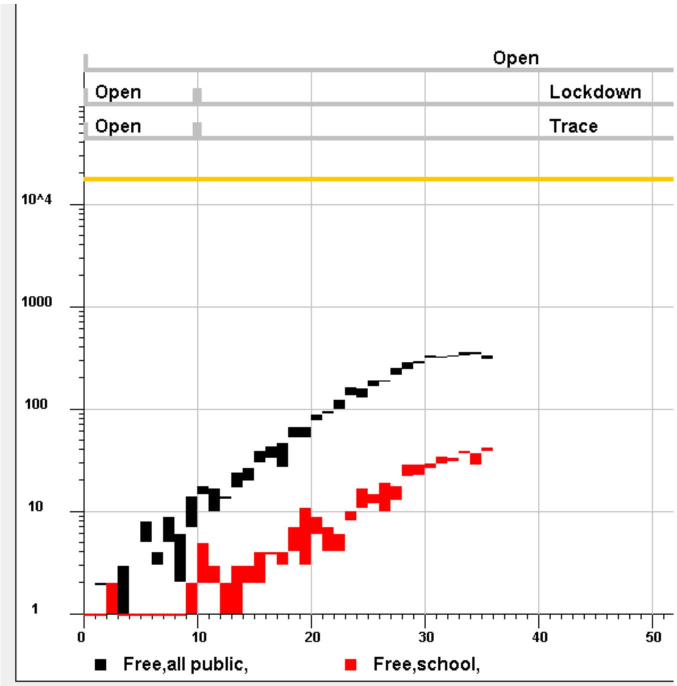
Colour: Red

Style: Solid

As these are logarithmic plots, the ratio of the numbers in any two curves is given by the vertical distance between the curves. The spacing of grey lines represents a ratio of a factor of 10, and the total modelled population is shown by the orange bar. The three modelled scenarios and their division into periods is shown in the three grey bars at the top. Graphs are created or removed by the menus and buttons.

The vertical sizes of the points are quartiles, showing the amounts of variation between different runs of the simulation, arising from random processes of infection. Because infection by the virus is a random process in the real world, these error bars represent a fundamental limit on the precision of the predictions of any model.

Because models track exposures to the virus, it is possible to chart the rates of exposure events, as well as the resulting disease states. This is shown below. As before, the details of the curves are only illustrative, because the parameters of the model which was used have not been tuned to match reality. The point is that models can be used to predict and display these things, when they have been tuned.



This chart shows the number of all infective encounters, and of those which take place in schools, in a scenario with no controls. Values are not intended to be realistic, but could be made so. The program can also chart the total numbers of encounters, without infection, as measures of social and economic activity. It is possible to compare different scenarios, to assess the impact of different containment policies on social activities such as education, or the provision of healthcare (with consequences for non-Covid-19 sickness and death) or on economic activity.

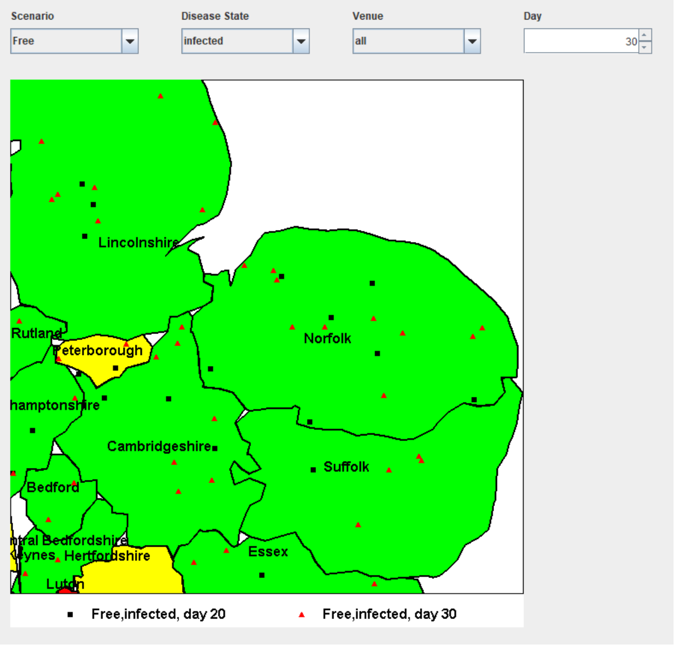
In this way, models can be used to assess not only the spread of Covid-19 in the population, but also the social and economic impacts of containment policies - addressing the questions which governments are currently grappling with.

While this illustration of social impact has been quite coarse-grained, it is possible to define types of meeting place to any level of detail (e.g. to the level of specific types of business), and to model them using real data rather than randomly generated data.

We can expect that models of disease spread in a population will need to import results of more specific behavioral models or other models - such as models of behaviour in certain types of location. This requires architectural components for **model exchange** and **model results exchange**.

So it is possible to build fine-grained and useful models of economic and social activity, and of how they are affected by the virus and by the containment measures. Comparative assessments of the economic impact of different virus containment policies will be essential to restore economic activity, and to restore the normal fabric of society, as rapidly as possible.

It is possible to track the spread of the virus across regions, by looking at the rates of infection at different times on the map view:



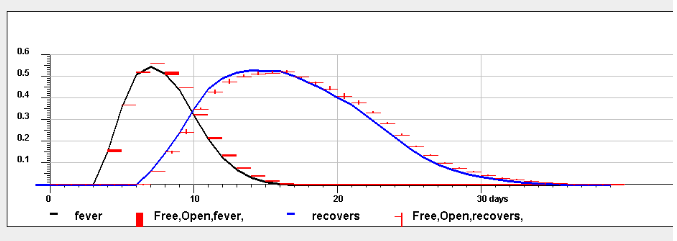
Scenario: Free
Disease State: infected
Venue: all
Day: 30

This view shows the locations of infected people on two days of a simulation run. Regions can be shaded according to the densities of people in defined disease state, or the rates of infection events.

There is an important check of the correctness of the Monte Carlo modelling engine, by looking at the progression of simulated people through disease states. This progression can be computed in two independent ways:

1. Theoretically, by computing the probabilities of disease state transitions for each day after infection (checking that the probabilities of all states add up to exactly 1 on each day)
2. By summing the results of the Monte Carlo simulation for any period of any scenario

These two probability calculations should give identical results, which can be displayed in the framework:



This graph shows two disease states - fever and recovered - each calculated in the two different ways, showing detailed agreement. The comparison can be checked for any period of any scenario.

This is a simple test, but gives confidence in the underlying agent-based simulation. Similar self-checking tests are a useful architectural feature for any modelling engine.

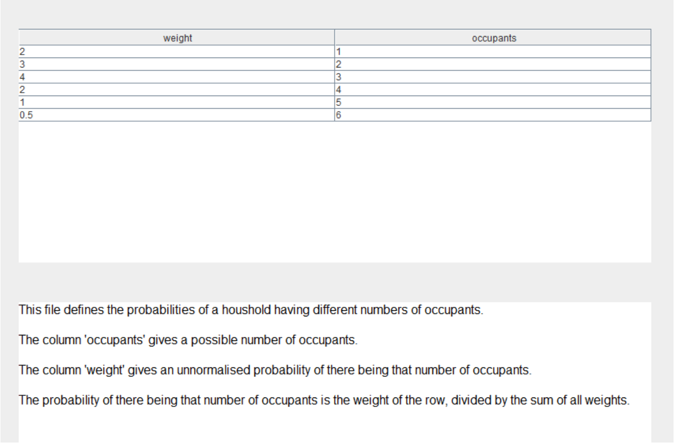
## 11. Understanding Model Inputs

This section illustrates the **Model Description and Explanation** component of the architecture.

When inspecting results of a model, it is important to know whether any feature of the results is a reliable prediction of the model, or is just some artifact of the data that has been put into it. Understanding the inputs of a model is as important as understanding its outputs.

This aspect of the modelling framework is at an early stage of development. There are two tabs allowing users to inspect either the parameters used to drive a model, or to inspect the stored demographic model.

Models are driven by csv files (spreadsheets) of configuration data. You can inspect any of these files in the framework. Two example views of the Simulation Data tab is shown below:



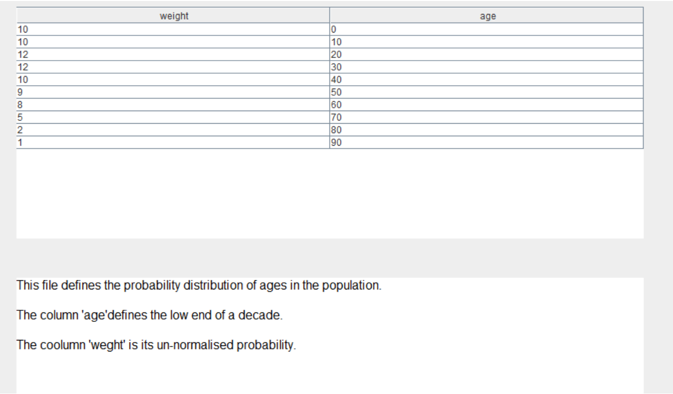
| weight | occupants |
|---|---|
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |
| 2 | 4 |
| 1 | 5 |
| 0.5 | 6 |

This file defines the probabilities of a houshold having different numbers of occupants.

The column 'occupants' gives a possible number of occupants.

The column 'weight' gives an unnormalised probability of there being that number of occupants.

The probability of there being that number of occupants is the weight of the row, divided by the sum of all weights.



| weight | age |
|---|---|
| 10 | 0 |
| 10 | 10 |
| 12 | 20 |
| 12 | 30 |
| 10 | 40 |
| 9 | 50 |
| 8 | 60 |
| 5 | 70 |
| 2 | 80 |
| 1 | 90 |

This file defines the probability distribution of ages in the population.

The column 'age'defines the low end of a decade.

The coolumn 'weght' is its un-normalised probability.

The second example shows the age distribution of the modelled population, defined in terms of 'weights' which are un-normalised probabilities of ages in each decade.

This enables users to inspect any of the data files, and to see help file documentation describing how each type of data is used in the model.

## 12. Performance of the Models

This section discusses performance requirements for the **modelling engine** component.

It might appear computationally expensive to model a large population to the level of individual simulated people, and individual simulated transmission events.

The costs of agent-based simulation are limited, because the only encounters which need to be modeled are encounters involving infected people, who (in recovery situations) are only a small proportion of the population. Therefore it is possible to model populations as large as a million, in several scenarios of many days each, on a modest PC in a few hours.

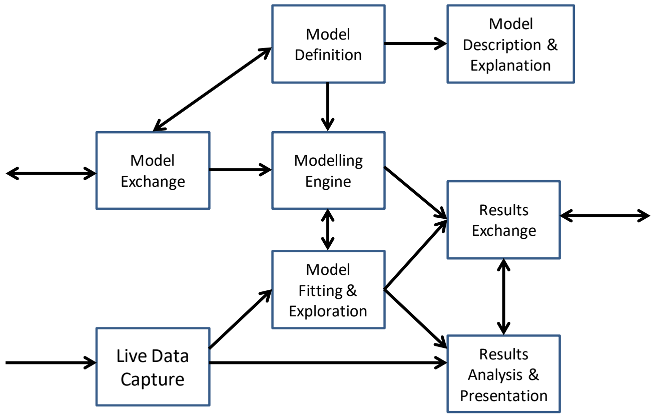
However, it is easier and faster to model a smaller population, yielding insights which can be scaled up to a larger populations (which, after all, are made up of sets of smaller regional sub-populations). It should be possible fairly rapidly to develop insights about how model results scale with the modelled population size. You can run many small-scale models to explore ideas, and a few larger-scale models to validate the results.

It is not prohibitively expensive to model large populations on modern parallel hardware, and the model is amenable to parallel running. The modelling framework could be cloud-hosted for easy deployment and scaling. The costs of one run of the simulation grow linearly with the size of the infected population. There is usually a need to run the model many times, to explore the dependence of results on several variables simultaneously - to fit models to data, or explore options. This replication is suitable for parallel running, or big data techniques.

So while the most promising uses for this modelling framework may be the smaller-scale applications which can be run rapidly, it is possible to scale it up to large populations - especially if infection rates are low.

## 13. Modelling Architecture Revisited

For convenience the architecture diagram is repeate, before summarising on the components:


Model Definition
Model Description & Explanation
Model Exchange
Modelling Engine
Results Exchange
Model Fitting & Exploration
Live Data Capture
Results Analysis & Presentation

**Model Definition**: the tools for model definition in the demonstrated framework are very simple - editing csv files. While one can easily imagine more purpose-built notations and facilities (for instance, to validate features of models as they are defined), csv files have the benefit of simplicity, transparency and of being open-ended. Because models need to incorporate the spread of the disease, the containment measures, and social and economic consequences, and because we do not yet know what level of detail will be necessary in useful models, open-endedness is a necessity.

Therefore, while other model definition notations might be superior to csv files (e.g. using XML, JSON, or domain-specific languages (DSLs)) , we may expect more special purpose languages and facilities to co-exist with general workhorses (such as csv files) for some time.

In the **Model Exchange** component, similar comments apply. The demonstrated framework uses folders of csv files, which can be zipped for for model exchange. This would do the job, but we can expect other model exchange formats to emerge (for instance, based on XML, JSON, HL7 FHIR, or DSLs). We need to avoid a plethora of incompatible formats, while allowing room for experiment in formats, and recognising the need for open-endedness, driven by the wide scope of the modelled domains. This will require a delicate balance between experimentation and the need to converge on standard formats.

**Model Description and Explanation** involves putting an accessible user interface on top of an underlying model definition notation. In the demonstration framework, the model definition notation is csv files, and the user interface consists of (a) tabular display of the csv files, and (b) text descriptions of what they mean. This is about as basic as it gets, but can still be useful.

As model definition notations evolve, so there will be new interfaces to help users understand them. Experience in building user interfaces suggests that designers cannot anticipate what interfaces will work well - they will need to try things out with real users and see what works.

For the **Modelling Engine** component I have described the benefits of using a stochastic agent-based (Monte Carlo) approach, mainly because it is open-ended, data-driven and can give a seamless join between epidemiological modelling and wider modelling of social and economic impacts. Facilities for automated testing and self-test of the modelling engine are important.

Whatever approach is used, some models need to be run many times for a variety of purposes, and the scaling of the model engine performance will matter. Here, there are no big complexity barriers - modelling costs scale linearly with population sizes, and with the number of replicated runs. There will still be a need, in some contexts, for high - performance hardware, or cloud hosting of models for easy deployment and scaling. At the same time, small-scale models can be run on personal hardware.

**Model fitting and exploration** has only been developed in basic form in the demonstration framework. It is possible to run the same model many times, extracting statistical properties of the results such as quartiles or percentiles of any quantity; or to run several scenarios and inspect the results side by side. What the framework does not yet have (but which can easily be added) is the ability to vary parameters to automatically optimise the fit of a model, for instance to fit disease data according to some $\chi 2$ data fitting criterion, or to optimise a measure of cost and performance of a containment measure.

For model fitting and exploration, there are major benefits in adopting a Bayesian modelling approach, such as the Dynamic Causal Model (DCM) of [4]. This is a generative model, which avoids over-fitting data by an over-complex model. Being generative, it infers the underlying causes which generate observed events (for instance, the distribution of infection events which causes the observed states of infection) as well as the confidence levels in the causes. The ability to work back to causes is required for rapid response to events - for instance, it is more effective than using a retrospective measure such as the reinfection rate R.

Similarly, the **Data Capture** component has only been developed in rudimentary form in the demonstrator framework. Currently the only data which it can capture is UK map data and demographic data from the UK Office of National Statistics, in specialised forms such as KML (an XML dialect) or as csv data.

It would be  easy to carry on adding specialised data capture modules - for instance, using HL7 Version 2 [5] ,which is widely adopted worldwide  for laboratory test data, and HL7 FHIR [6] which is widely adopted for  other healthcare data, or other formats for economic and social data - but the cumulative complexity (and maintenance costs) of these interfacing modules should not be under-estimated. Interoperability and integration is a large IT cost driver. Defining and using standards has a crucial role to play, in controlling the costs and avoiding lock-in.

In the data capture component, the demonstrator framework provides no facilities yet for managing data quality. In live modelling and tracking frameworks, data quality management will be of paramount importance. The known tools for data quality management will need to be applied, to avoid GIGO.

For **Results Analysis and Presentation**, the demonstrator framework provides moderately flexible charting and mapping facilities, designed to chart and display the outputs of model runs. Much more can be envisaged. There is a full panoply of tools for data visualisation and business intelligence, available commercially or as Open Source, which we can expect to see applied.

For **Results Exchange** , the  demonstrator framework does not yet have facilities; but the considerations for  this component are the same as those for model exchange. It would be easy to add interfacing modules piecemeal; the challenge is to allow the necessary degree of experimentation, to exchange diverse sets of model results, whilst moving towards open standards for results data which facilitate exchange,  minimise software maintenance costs, and avoid lock-in to proprietary formats.

The modelling architecture has been described as a set of interacting components, without regard to the layering which is frequently used to describe IT architectures. The components have been described in terms of functionality rather than technology. It is useful briefly to consider a layered technology view of the modelling architecture.

In a simple multi-layer description, the layers are:

- User Interface
- Application Logic
- Data Storage
- Data communication

Examples of the **user interface** layer in the demonstrator framework appear through the paper. These were developed on Java AWT and Swing, a capable but somewhat old-fashioned user interface toolset, which was chosen for its familiarity, to build a demonstrator quickly. This technology would almost certainly not be used in a live modelling and tracking framework; one would use web-based tools, for instance around HTML and Javascript, to allow cloud hosting of a modelling framework.

The **application logic** is the logic required to run Monte Carlo simulations in an open-ended domain which concerns virus spread, containment measures, and social and economic behaviour. Uncontroversially, the best way to do this is in an object-oriented language such as Java, in which the different things being simulated are in classes representing each type of thing (person, meeting place, and so on), and the simulation includes multiple instances of each class.

We are still at an early stage of model development. We do not yet know either the scope of what we will need to model (e.g. encompassing social and economic behaviour) or the level of detail to which it will need to be modelled (e.g. modelling local 'bubble' outbreaks to fractal levels of detail) . So flexibility and open-endedness  modelling engines is paramount. For this, a highly data-driven approach, like that used in the demonstrator framework, appears to be necessary.

In the demonstration framework, persistent **data storage** used files such as csv files. This is a tactical choice, and one would expect other persistent storage mechanisms such as database management systems to play a role. When moving to other storage technologies, it is important not to lose the flexibility, transparency and portability of simple readable data files.

D**ata communication** is currently under-developed in the demonstrator framework, but in the model exchange and results exchange components, it is expected to play an important role.

## 14. The Importance of Local Modelling

There are currently diverse national approaches to recoveri economies from lockdowns. However, national-level directives are probably not a good way to recover economies from the pandemic [2]. If there are local outbreaks in London or Manchester, it would be wrong to subject a whole nation to renewed lockdown. Setting policy at national level is a blunt instrument. Local and regional models are required.

Consider the impact of widespread testing for the disease. If, as well as prioritised testing (e.g. of key workers and healthcare professionals), there is frequent testing of a random sample in every locality  (e.g. weekly testing of a random sample of 1,000 people in a population of 100,000; following up all those who test positive; antibody testing), within weeks this would build up a reliable and timely picture of the local progression of the disease - rapidly identifying local outbreaks and hot spots for disease spread.

Local sample data provides valuable input to local policy making - identifying, for instance, the impact of opening a certain kind of workplace or school, and allowing rapid local response to events.

The value of local random sample data is enhanced if it is input to a predictive model of disease progression, such as that described in this paper. The sample data can be used to calibrate the local model (testing that it can reliably 'postdict' the past few weeks' data); and then the model can be run forward in time, under different scenarios, providing a forward view which is more accurate than simply extrapolating curves forward in time; and more reliable than national level predictions scaled down to local level, because it is based on actual local data.

As an example, in the region around Cambridge (UK), Covid-19 deaths are higher in the two North Cambridge wards of King's Hedges and Arbury than in other more prosperous neighborhoods. Possible reasons are not hard to find. For instance, while many Cambridge residents can work from home, the residents of Arbury are more likely to need to travel to work on public transport, and to meet others in their workplaces. With several alternative explanations of the effect, one needs to know which one matters the most. To go beyond general observations, and to forecast the impact of containment policies on these populations, any model of disease spread needs to be fine-grained enough and precise enough, to account for local hot spots. This also illustrates how disease spread may be influenced by social and economic factors, which need to be modelled in the same framework

The spread of Covid-19 is a localised, non-linear, bubble-like phenomenon - much more akin to a boiling pot of water, or to ferro-magnetic domains, than to a uniform gas. So population-based modelling, which treats a population in a 'thermodynamic limit' like a gas of people, is likely to be inaccurate. National level models, with their intrinsic need to aggregate data, are less suited for modelling these effects than fine-grained local models, which are closer to the fractal detail of the spread of the virus.

The value of local data collection and predictive modelling are further enhanced if the results and the models are made publicly available - allowing citizens and groups to assess the impact of polices on themselves (e.g. to see their own personal risk level) and to critique public policy. For instance, a decision to open or close some facility would be accessible to public review and debate, with citizens able to compare models of the two scenarios.

## 15. How to Build a Local Model

This section outlines the steps involved in building a local predictive model of the kind described in the previous section. Not in this order, the steps are:

a. **Calibrate the local disease model**: With local random sample testing and follow-up, the data can be used directly to calibrate a local finite-state model of the disease in individuals, like those in section 4; for instance, to measure the transition times and probabilities from one disease state to another, as a function of age, co-morbidities, race or any other variables. Statistical tests can check whether the data and the analysis have adequate granularity to pick out sub-populations with different disease susceptibilities - for instance, do the numbers of people in particular disease states have a Poisson distribution (indicating a homogeneous random process), or does it have larger variance - for instance, is it a bi-modal distribution, indicating distinct sub-populations?

b. **Build the demographic model**: Capture a map of the region, and identify the important sub-regions and types of meeting place, down to the level of the larger individual workplaces and institutions, and the densities of other workplaces. Set the main properties of the sub-regions and meeting places - population densities and makeup, and catchment ranges of meeting places.

c. **Build the economic and social model**: Perhaps the most important economic output from any model is the number of people in work. To get this as a model output, you need to input the frequency of encounters needed for the operation of specific workplaces or types of workplace. Similarly, to predict social activity, you need to input the frequency and proximity of encounters needed for activities at various kinds of meeting place - schools, parks, and so on.

d. **Build the model of containment measures**: whatever mix of lockdown, contact tracing or other measure is applied locally, these measures need to be modelled, including their limitations - slow response time, low takeup, and so on.

e. **Calibrate the model on recent local data**: Ensure that the parameters of the model are sufficiently realistic that, for instance, feeding it data up to one month ago, and asking it to 'predict' the last month, it gives an acceptable level of agreement. Tune parameters of the model until this is the case. This defines what level of confidence can be placed in the model's future predictions.

f. **Track current data and monitor data quality**: Since a key use of the model will be the interpretation of current disease data and near-casting of the next few weeks, it is essential to keep its data current (e.g. the latest weeks' randomised disease testing, contact tracing and antibody testing data) and to constantly test the quality of those data.

In this way it will be possible, as it is not yet possible, to forecast with confidence - to gauge the quality and sufficiency of incoming disease data and social data; to forecast the near term and to know the reliability of the forecasts; to spot renewed local outbreaks as soon as possible; to know the effectiveness of the containment measures they require; and to forecast the social and economic costs of those measures.

There are now many efforts worldwide to build organisations and systems to track the spread of Covid-19 and to coordinate activities including contact tracing. It will be highly beneficial to combine these new tracking and management applications with modelling of virus spread. A Bayesian approach to local modelling and model fitting [4] provides important benefits, beyond simple tracking of disease data. Bayesian modelling can trace back to infer the pattern of 'hidden' events that caused the data, defining confidence levels for the inferred causes, as a basis for rapid response to the causal factors; it can avoid over-complexity and over-fitting of models; and it can provide valuable quality checks of the source data (e.g. checking that disease models change only slowly with time), and reveal where better data are needed.

Local tracking and modelling applications will not operate in isolation. They will need to interface to neighboring local applications, in a network of federated models to understand the spread of Covid-19 between localities, and feed in to regional hubs to build up a larger picture. These interfaces should use open standards to avoid regional and national lock-in to favoured applications and frameworks, and to allow vigorous competition. HL7 FHIR is a possible basis for these standards.

Local tracking centres should use open standards to capture local disease data, to deter lock-in and allow continued competition. HL7 Version 2 (for test results) and HL7 FHIR (for other healthcare data) are the best basis for these standards. Now is the time to apply open data interchange standards, avoiding lock-in to whatever happens to be built first.

## 16. Competition and Diversity of Models

Frameworks for modelling Covid-19, and for modelling the related social and economic issues, will be needed for many years to come; and they will be needed for a range of different purposes. It is likely that no single modelling framework will emerge as the best, and it is desirable that so single winner should emerge. There should be continued competition of modelling frameworks, both commercial and Open Source, in different product categories.

To ensure that this competition delivers the best results to society - to policy-makers at national and local level, to interest groups, and to the public - two measures are needed:

1. **Deter Lock-in**: the effects of lock-in on the healthcare IT market have been highly detrimental to the quality of healthcare. Every step should be taken to avoid a similar lock-in for Covid-19 tracking and modelling frameworks. This requires defining open interface standards for the exchange of data, models and model outputs; and adhering to those standards in public procurements.
2. **Hold Modelling Challenges**: Periodically (every 6 months), there should be a prominent public competitive challenge in modelling Covid-19, open to anybody worldwide - with evaluation criteria and test data parameters known in advance, but with actual test data released only during a limited competitive timeframe. The winners would get high visibility in worldwide markets, so the best models can achieve rapid uptake.

Challenges like these have worked well in fields such as speech and natural language understanding, robotics (e.g. self-drive vehicle competitions) and image understanding. They have advanced the state of the art in those fields. Similar advances are needed in modelling Covid-19, with huge benefits for society.

Diversity will be reflected in different levels of modelling framework. Software products have often appeared in three levels - traditionally labelled as 'Personal', 'Professional' and 'Enterprise' levels[2] - and the 'personal' level of Covid-19 modelling might take the form of a personal risk assessment app. This app would provide answers that people now need to know, such as:

- What is my personal risk of dying from Covid-19 in the next year:
    - If I carry on living as I have done for the past month?
    - If I start going to work?
    - If I travel daily by public transport?
    - If I go to the pub every week?
- How does that compare to my background risk of dying from other causes?
- What is my risk of passing on the disease to someone more vulnerable, under the same set of 'what if?' scenarios?
- What are the levels of risk for those close to me?

People are hungry for specific guidance, and the app can provide it. Such an app would take information from local, regional or national tracking and modelling centres - much

[2] For Covid tracking and modelling applications, the 'Enterprise' level might be re-named the 'Society' level.

in the manner of a local weather forecast - and would pass nothing back. Small groups of people who meet often (such as families) could agree to link their risk tracking apps to be aware of each other's levels of risk, to make their personal forecasts more reliable.

Recently in the UK, citizens have been asked to 'be alert' and 'use their common sense'. They understandably express frustration at the vagueness of this advice. Through no fault of our own, we have no common sense about Covid-19; the virus is so novel and invisible that we have no natural intuitions about risk levels, or how our behaviour affects them. A personal risk tracking app would help everyone develop those intuitions.

As before, it is important that both the modelling frameworks and the personal risk apps should use open interfaces to exchange data, to deter regional or national monopolies, and to sustain competition.

Managing Covid-19 and its consequences require organisations and systems which are joined up at local level - in a way that most healthcare IT systems have failed to support, because they have not applied open interchange standards. Now is the time to apply open standards to managing Covid-19. If we fail now, we will suffer the harmful effects of fragmented systems and data for years to come - not just in blighted healthcare, but in blighted economies and societies.

## 17. Conclusions

We are at the start of an era in which understanding the spread of the Covid-19 virus, and its impact on societies and economies, will be the central question facing all societies. We are facing the worst economic depression of our lifetimes, and economic depression blights lives. Whether and how and when each society recovers from the depression will depend on how well that society understands and manages the consequences of Covid-19.

Understanding the spread of Covid-19 at any level - local, regional or national - is the combined scientific and social challenge of the era. As with any branch of science, success depends on linking good quality data (e.g. from widespread testing, contact tracing and antibody testing) to accurate models, with which to analyse and forecast. To link the two together in an ecology of fast-response systems, we are dependent on IT, some of it as described in this paper - combining tracking and modelling of the spread of Covid-19, and of its economic and social consequences.

These IT tools are still in their infancy. They will mature and evolve and diversify even in the next few months. We may need to move away from population-level epidemiological models, which treat a population as an ideal uniform gas - to models which reflect the non-linear, localised, fractal-like, ferromagnetic domain-like or boiling water bubble-like nature of reality. Such models are currently unknown territory. We will move from a few centrally maintained models to a huge diversity of models and modelling tools at regional and local level, for use by different groups and individuals, and in different product categories.

The best outcomes - and the societies which will recover fastest - will come from open competition between modelling tools and frameworks; from open peer review of models and their predictions; and from open debate of the consequences of actions.

For these things to happen, we must attend now to the architectures of the tools - ensuring that the tools are open-ended to evolve with increasing knowledge and changing conditions; have open interfaces for free interchange of models and their results; and avoid the lock-ins that have beset healthcare IT.

This paper and its illustrative modelling framework are intended to start the discussion of the IT architectures needed for tracking and modelling Covid-19. I ask others to take up the discussion - borrowing, disputing and improving the ideas in this paper.

## References

[1] See for instance https://www.nature.com/articles/d41586-020-01003-6

[2] The first independent SAGE report: https://drive.google.com/file/d/1MD4-8z-yy-lO5ZsfmXAxTUo79iFk1zfy/view

[3] Healthcare IT and lock-in; see e.g: https://www.openehr.org/about/what_is_openehr

[4] Dynamic Causal Modelling: Friston K, Parr T, Zeidman P, et al. Dynamic causal modelling of COVID-19 [version 1; awaiting peer review]. Wellcome Open Research 2020; 5(89).

[5]HL7 Version 2 for laboratory test results: https://www.hl7.org/implement/standards/product_brief.cfm?product_id=279

[6] HL7 FHIR: https://hl7.org/FHIR/