INSIGHTMAKER INTRODUCTION COURSE

MANUAL

VERSION 2

Fons Wijnhoven

University of Twente, Enschede, The Netherlands

a.b.j.m.wijnhoven@utwente.nl

April 15, 2023

Abstract

What if data about the past are lacking for knowing anything about the future? Can we than still make predictive models by reasoning alone and simulate what could happen if the reality would work out differently as in the past. There are three techniques that can be used in this context: system dynamics simulation, agent-based modelling and simulation, and foresighting. InsightMaker gives many tools for working in a user-friendly way with these techniques.

Table of Contents

1. System dynamics

Rahim et al [1] define system dynamics as an approach to understanding the nonlinear behaviour of complex systems over time using stocks, flows, internal feedback loops, table functions and time delays. System dynamic models can be created by constructing mental models of project situations that may be partially informed from data but basically assumes that not all relevant data are available or trustworthy [2, 3]. Consequently, system dynamic models are thinking machines for answering "what if" questions about possible futures that are hard to create and experiment with in a non-virtual way. Well-constructed system dynamic models can be used as thinking engines for digital twin systems [4].

Hristotoski and Metresvky [5] further describe system dynamics modelling as a method for exploring systems on an aggregate level. By 'aggregate', it is meant that system dynamic models look at collections of objects, not the objects themselves. For instance, a system dynamic model of the e-Customers population would look at the population as a whole, not at the individual e-Customers. System dynamic models are visually constructed from a set of basic building blocks, known as 'primitives'. However, behind the scene, these primitives are 'converted' into a system of differential equations that describe the modelled system mathematically [5].

Lyneis and Ford [6] state that there are many tools for system dynamics modelling, but Insight Maker (which we use further throughout this text) is *"....an innovative, free-of-charge, Web 2.0-based, multi-user, general-purpose, online modelling and simulation environment, completely implemented in JavaScript, which promotes online sharing and collaborative working. (...) To the best of our knowledge, it is the first, yet the one and only free-of-charge Web 2.0-based Internet service that can deliver a plethora of advanced features to its online users, including Causal Loop Diagrams, Rich Pictures Diagrams, Dialogue Mapping, Mind Mapping, as well as Stock & Flow simulation. (...) By supporting agent-based scenarios, storytelling, and sensitivity analysis, Insight Maker exhibits a wide gamut of features that not* *only rival but also, in many cases, outperform the traditional, commercially available simulation software packages*" [5].

The key system dynamic primitives are Stocks, Flows, Variables, and Links. These are represented by Insight Maker as follows. Stocks, like materials, customers, or money, are graphically presented by rectangles. Flows are graphically depicted by bolded solid lines with arrows that give the direction of the 'material' flows. Variables are graphically portrayed by ovals; they can be dynamically calculated values that change over time (governed by an equation), or they can be constants (fixed values), e.g., e-Customer arrival rate. Links are graphically shown by dashed lines with arrows that show the transfer of information between the different primitives in the model.

In the system dynamics methodology, a problem, or a system (e.g., ecosystem, political system, or mechanical system) may be represented as a causal diagram. A causal loop diagram is a map of a system with all its constituent components and their interactions. By capturing the structure and interactions of a system, it becomes possible to ascertain a system's behaviour over a certain period. The causal loop diagram of a new product introduction may look like Figure 1.

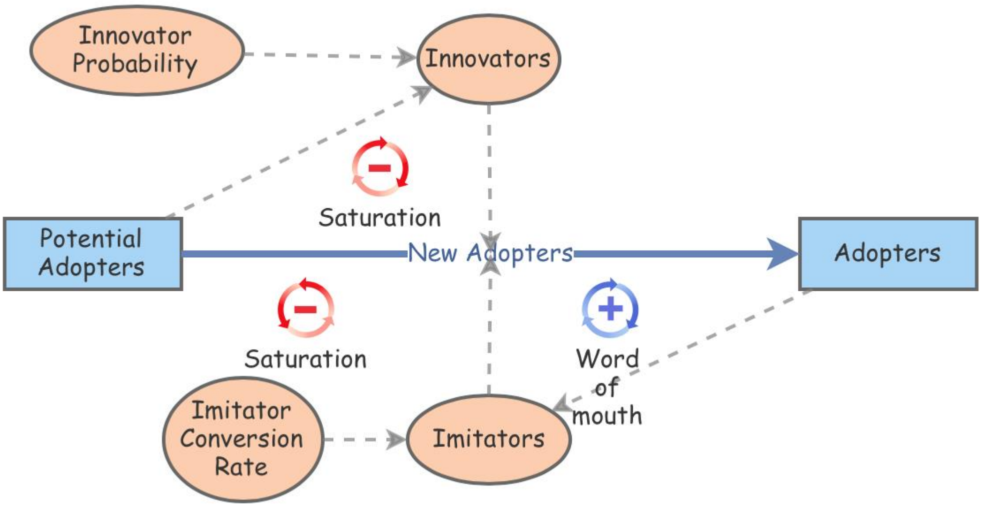


Figure 1: Causal loops of New Product Adoption dynamics (Remake of [7])

There are two feedback loops in this diagram. The positive reinforcement “word of mouth” loop on the right indicates that the more people have already adopted the new product, the stronger the word-of-mouth impact. There will be more references to the product, more demonstrations, and more reviews. This positive feedback should generate sales that continue to grow. The second feedback loop is a negative reinforcement. Clearly, growth cannot continue forever, because as more and more people adopt, there remain fewer potential adopters and thus this stock will be empty after a while. Both feedback loops act simultaneously, but at different times and they may have different strengths. Thus, one might expect growing sales in the initial years, and then declining sales in the later years. A system dynamic causal loop diagram thus has stocks and flows. A stock is the term for any entity that accumulates or depletes over time. A flow is the rate of change in a stock. There are two stocks in product adoption models: Potential adopter stocks and adopter stocks. There is one flow: New adopters. For every new adopter, the stock of potential adopters declines by one, and the stock of adopters increases by one, resulting on a new product adoption model. The actual volume and speed of this market growth and decline (thus also note the non-linearity in this case) may be influenced by variables like the number of imitators and adopters, which are represented by ovals in Figure 1.

We explain here how one can create a system dynamic model by a few steps in Insight Maker.

Step 1: Select stock and flow primitives.

The key primitives for System Dynamics models are Stocks, Flows, Variables and Links. Stocks store a material. For instance, a bank account is a Stock that stores money. A bucket is a Stock that stores water. A population is a Stock that stores people. A Flow moves material or other entities between stocks. For instance, in the case of a bank account you could have an inflow of deposits and an outflow of withdrawals. You can add elements to your model by selecting "Add Primitive" in the toolbar (see Figure 2) then choosing your desired primitive type.

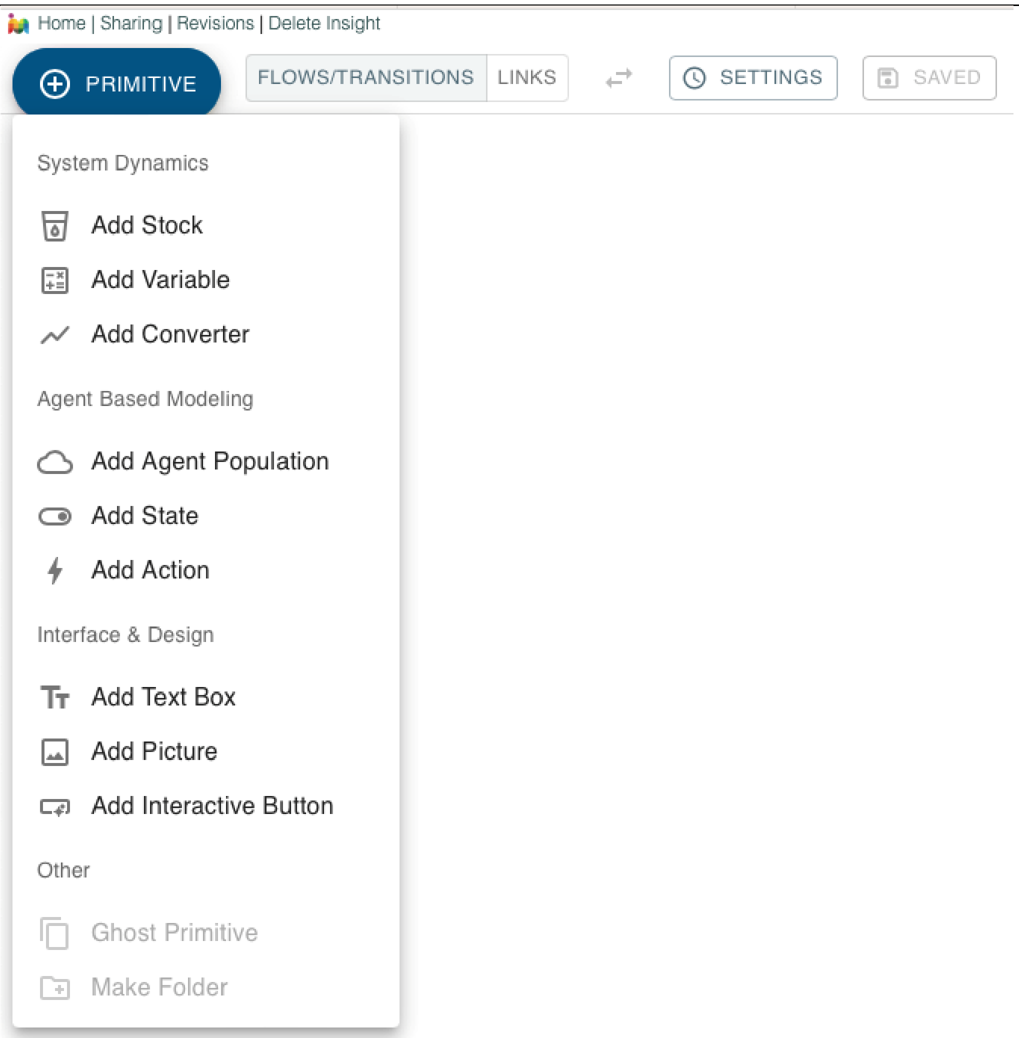


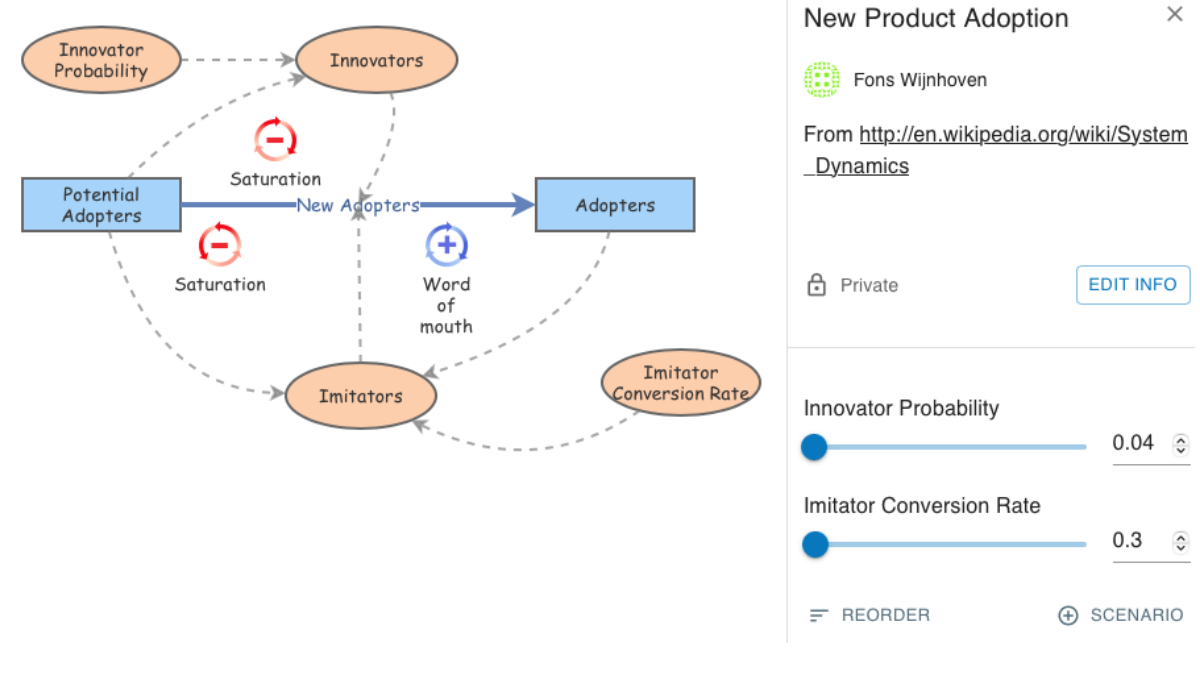
Figure 2: Insight Maker's System Dynamics primitives

For our case, we have two stocks, potential adopters and adopters, which we link by a flow of New Adopters. In Insight Maker, primitives allow you to enter equations that determine their values or behaviour. To access the equation editor for a primitive, hover your mouse over the primitive and click the "=" sign that appears. You can also use the configuration panel on the right side of the main window to edit the equation when the primitive is selected. We can give a number of Potential Adopters in the initial state, for example 1,000 at the = when this stock is opened. (BTW, note that Insight Maker cannot use the comma separator, but the point separator can be used).

Step 2: select variables.

Variables are dynamically calculated values or constants. In the bank account model, you could have a Variable representing the interest rate. It could be a fixed value or the outcome of an equation with different outcomes over time. Links show the transfer of information between the different primitives in the model.

In our case, we choose for two variables that directly influence the size of the flow (i.e., new adopters), the number of innovators and the number of imitators, which both are respectively influenced by an estimated probability of their occurrence in the population of potential adopters and the influence of word of mouth by adopters. For this, we insert in the variable formula for Innovators the product of "[potential adopters]*[Innovator probability]" and for Imitators we insert the formula "[Adopters]*[Imitator conversion rate]. For the probability and the conversion rate, we use more or less arbitrary .4 and .03, but we also give a slider (see Figure 3) so that the user of the model can interactively adapt alternative values. The equations can be arbitrary mathematical or programming expressions. There is a wide range of built-in functions you can use to develop equations and many advanced programming features that can also be utilized.



New Product Adoption

Fons Wijnhoven

From http://en.wikipedia.org/wiki/System_Dynamics

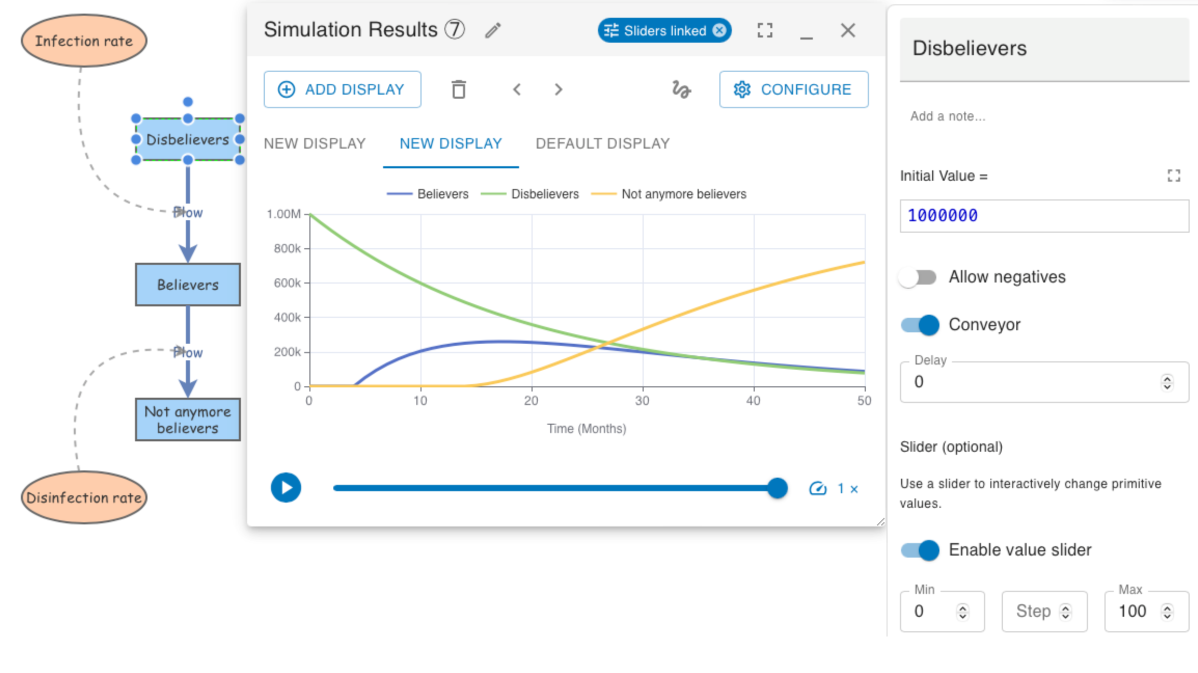
Innovator Probability 0.04

Imitator Conversion Rate 0.3

Figure 3: The system dynamic model with slider variables

Alternatively, the value of a stock – or any other valued primitive for that matter – can be referenced in equations using a pair of square brackets around the primitive's name. For instance, let us presume we are attempting to simulate an aging population. One way to simulate this would be to create a number of different stocks representing different age groups. For instance, we could have a Young stock, a Middle-aged stock, and an Old stock. Each time step we could transfer a fixed fraction of the Young stock to the Middle-aged stock, and a fraction of the Middle-aged stock to the Old stock. We could simulate births by adding new individuals to the Young stock, and deaths by removing people from the Old stock. This would work well if we had a population where the demographics remained roughly constant. Let's assume, however, that we encounter a shock like the baby-boom generation. This shock would add many people to the Young stock instantly. These individuals should age many years before they are passed on to the Middle-aged stock. Unfortunately, because we are moving a fixed fraction of the Young stock to the Middle-aged stock each period, the shock is felt in the

Middle-age population immediately. This will lead to incorrect demographic modelling. We can attempt to remedy this by increasing the number of stocks. Instead of three, we could have six, or 30, or 1000. The more stocks we added, the more accurate our modelling. Creating such a large stock, however, would be infeasible! Fortunately, Insight Maker allows you to convert a stock into a conveyor stock, thereby creating a theoretically unlimited number of sub compartments within one stock.



Disbelievers

Add a note...

Initial Value = 1000000

Allow negatives

Conveyor

Delay 0

Slider (optional)

Use a slider to interactively change primitive values.

Enable value slider

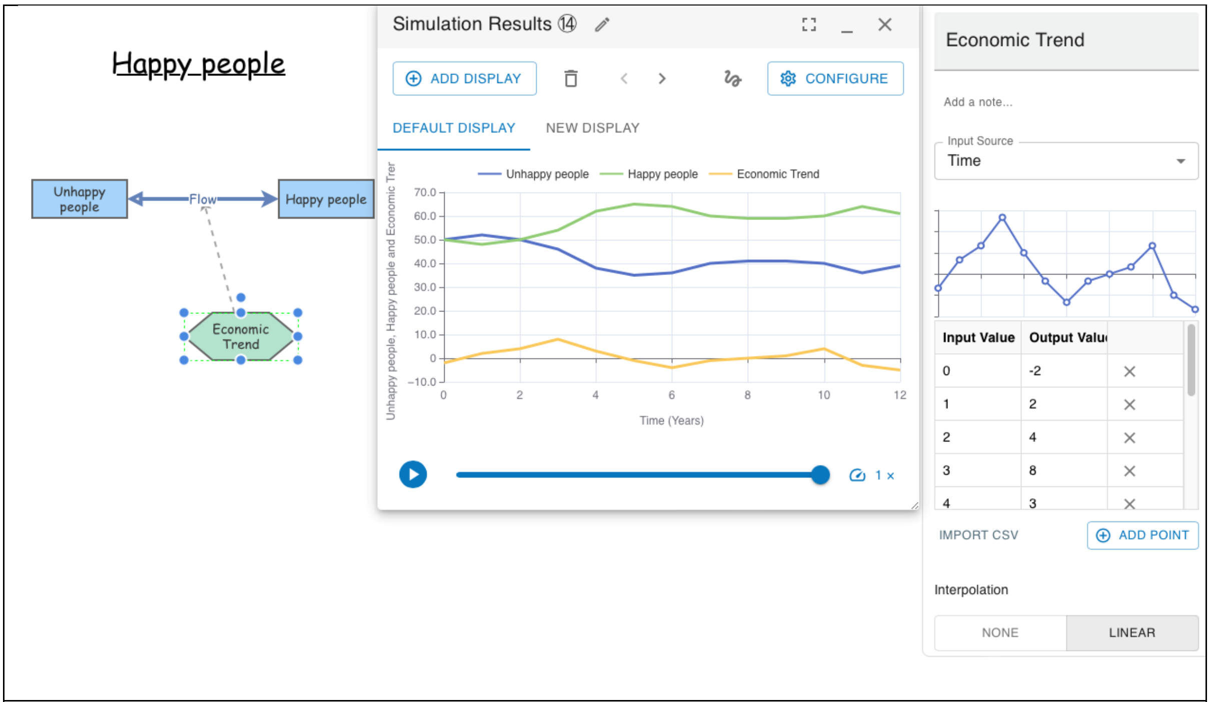
Min 0 Step Max 100

Figure 4: Example of a conveyor stock with stock member

Illustration of 1,000,000 people in a “Disbeliever” state, 5% (Infection rate of 5%) of them exit the group of Disbelievers each time and with a delay of 5 times (months) they enter the “Believers” group. After 10 months a percentage (Disinfection rate of 1%) of the new Believers turn into “Not anymore believers”. The delay for the Disbeliever stock is set 0 so that new people can enter anytime if an inflow is created (not the case here), the Delay for stock Believer is set 5 and for Not anymore believers is the Delay is set 10 months.

Sometimes it is hard to decide which parts of your model should be simulated using stocks and which parts with variables. Generally, stocks should only be used to model items that have inflows or outflows. If you have a stock in your model without either an inflow or an outflow, it would almost certainly be modelled better with a variable.

Often you will want to add structured input or empirical data to your model. For instance, if you were modelling a lake, you might want your model to have access to historical precipitation data that was recorded periodically. Alternatively, you might want to know the actual, surveyed surface area of the lake, given the volume of the lake. Insight Maker makes using this data easy with its converter primitive which allows you to create an input-output relationship or graph. When the input source to a converter primitive takes on an input value, the converter takes on the corresponding output value which may then be used by other primitives that reference the converter. The input can be the current time (in seconds, minutes, hours, etc…) or the value of some other primitive that is linked to the converter. You can enter data into the converter by hand or import a two-columned comma separated dataset (CSV file). For example, see Figure 5.



Input Source: Time

Input Value	Output Valu	
0	-2	×
1	2	×
2	4	×
3	8	×
4	3	×

Interpolation: LINEAR

Figure 5: A converter (right pane) (example data of percentage increase and decline of number of happy people in 12 years of economic trend data)

For converters, if you have not specified a specific output value for a given input, the outputs closest to the input will be combined to develop an appropriate output based on an interpolation method you have specified.

Step 3. Simulation.

The real power of system dynamics is utilised through simulation. Although it is possible to perform the modelling in a spreadsheet, there are a variety of software packages that have been optimised for this. Such simulations may be done for multiple periods like days, weeks, and years using certain adoption percentages and different numbers of adopter stocks. The dynamic simulation results show that the behaviour of the system would be to have growth in *adopters* that follows a classic s-curve shape. The increase in *adopters* is very slow initially, then exponential growth for a period, followed ultimately by saturation. This simulation is realized via the choice of parameters in Insight Maker, resulting in the following outcomes of Figure 6.

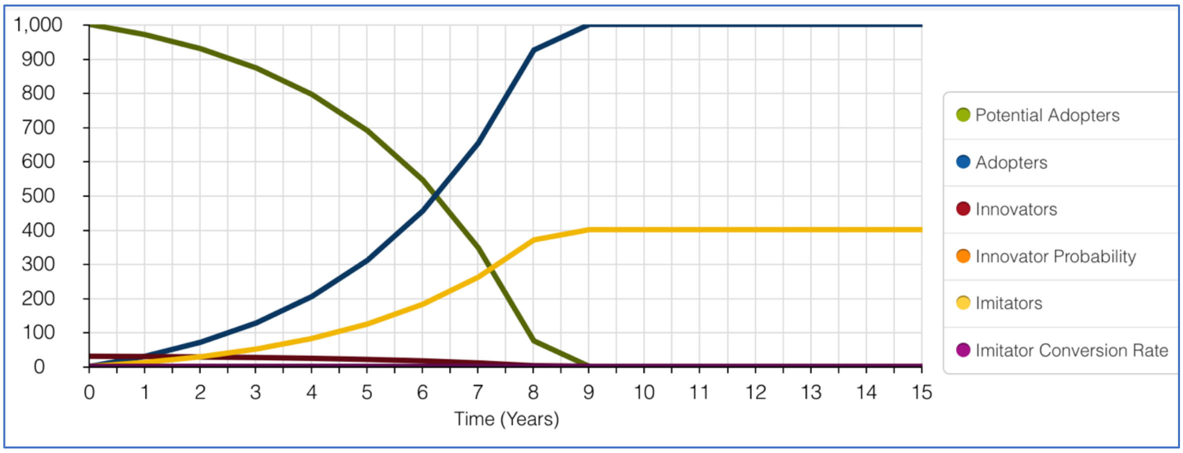


Figure 6: Simulation outcomes of model Figure 1 after 15 years.

System dynamics has found application in a wide range of areas, for example population, agriculture, ecological and economic systems, which usually interact strongly with each other. A system dynamics approach to macroeconomics has been developed by the economist Steve Keen (2013). This has been used to successfully model world economic behaviour from the apparent stability of the Great Moderation to the sudden unexpected Financial Crisis of 2007-2008. System dynamics have various management applications. They are a tool to:

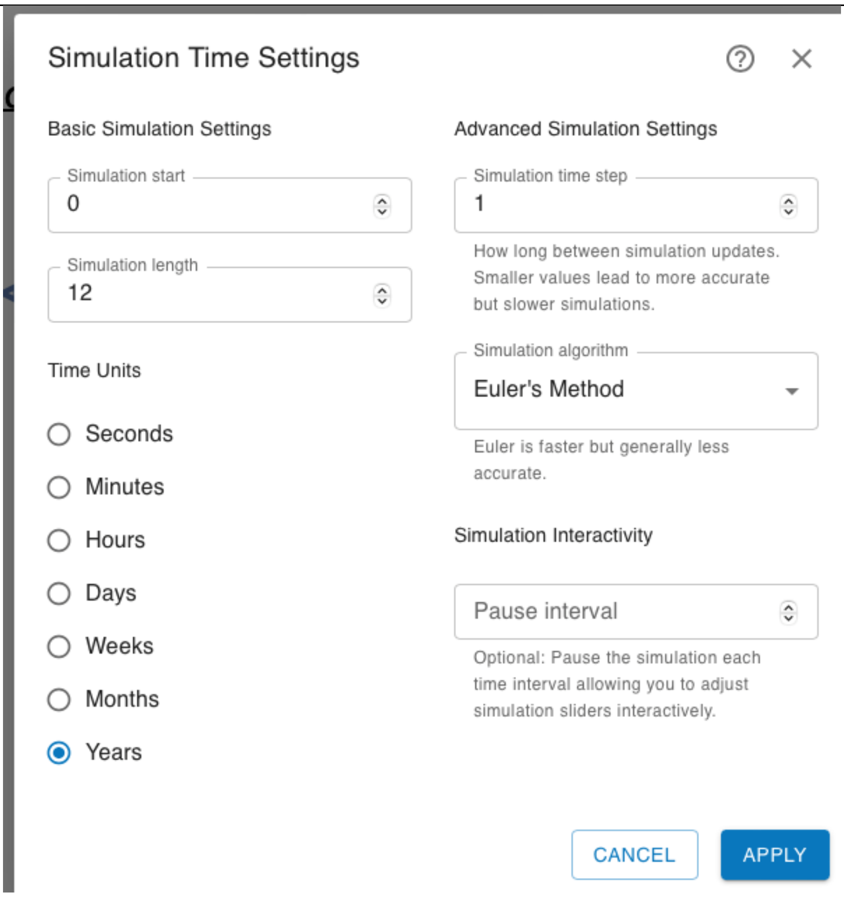
- Teach system thinking reflexes to persons being coached.
- Analyse and compare assumptions and mental models about the way things work.
- Gain qualitative insight into the workings of a system or the consequences of a decision.
- Recognize archetypes of dysfunctional systems in everyday practice.
- Being used as the thinking engine for digital twins. Note that digital twins and simulations differ in the sense that a digital twin always includes a simulation model but additionally a digital twin has continually updated data that will be inserted in the

simulation model to give the decision maker a continues understanding of longer-term impacts of observed changes in reality [4, 9].

Once you have completed diagramming a model and specifying model equations, you are almost ready to run a simulation, but first you need to configure the Time Settings for the model. This can be done by clicking the Time Settings button in the toolbar by which you control the key factors driving the simulation. These factors can be as follows:

- Simulation Start: The start time for the simulation.
- Simulation Length: The length of the simulation.
- Simulation Time Step: The length of time between separate evaluations of the model by the simulation engine. The smaller this number, the more accurate the simulation but the longer the simulation will take.
- Time Units: The time units. The Simulation Start, Simulation Length and Simulation Time Step will all be measured in these units.
- Analysis Algorithm: There are two different algorithms that can be used for simulations. The fourth order Runge-Kutta algorithm is generally more accurate when dealing with System Dynamics models. However, the Euler algorithm is faster and is preferred for Agent Based Models or models containing discontinuities.
- Pause Interval: If set, the simulation will be paused at every interval. You will have the chance to adjust the values of any primitives you have defined sliders for, before resuming the simulation. This allows you to create interactive flight simulators or games.

Figure 7 gives the Insight Maker screen for making your simulation time settings knowns to Insight Maker.



Simulation Time Settings

Basic Simulation Settings

Simulation start: 0

Simulation length: 12

Time Units

○ Seconds
○ Minutes
○ Hours
○ Days
○ Weeks
○ Months
● Years

Advanced Simulation Settings

Simulation time step: 1

How long between simulation updates. Smaller values lead to more accurate but slower simulations.

Simulation algorithm: Euler's Method

Euler is faster but generally less accurate.

Simulation Interactivity

Pause interval

Optional: Pause the simulation each time interval allowing you to adjust simulation sliders interactively.

Figure 7: Simulation settings.

Results are organized as a set of displays each of which contains some data from the model. You can create and configure new displays to examine your model from different perspectives.

Ensuring model accuracy and validity is a critical task and there are several procedures to obtain some level of confidence in your model results. These confidence-building activities include:

- Historical Tests: Comparing the results of the model to known historical values.
- Extreme Value Tests: Testing how the model behaves when extreme values are inputted into it. Does the model behave as one would expect or does it exhibit wild behaviour?

- Submodel Tests: For large models it is sometimes possible to isolate one portion of the model and compare it to some other more detailed model designed for analysing that specific feature.

In addition to these methods, Insight Maker includes a rich suite of features designed for documenting the assumptions that your model is based on and ensuring that these assumptions are not violated during model simulation.

To prevent mistakes, Insight Maker includes a set of tools for documenting and enforcing your model’s assumptions as you develop them. These tools include Unit Checking and Constraint Checking. These tools can be accessed by clicking the model verification button that appears in the configuration pane when you select a stock, flow, converter, or parameter within your model. Once you have defined verification checks for a primitive, a checkmark will appear on the primitive’s verification button letting you know that primitive will be validated during the simulation.

Each time the user clicks the "Step Forward" button, a single step will be executed. If you want to execute multiple distinct steps as a batch, you can add a Group step. Any steps that are within a group will all be executed together in one go. Once you have completed your Story, you can publish it as an "Article". Articles take your story and convert it to a beautiful, static web-page that is easy to share with others.

Often the precise values of model parameters or stocks will not be known. When you build a model, you will use the information available to you to create an estimate for a parameter value, but there will always be some degree of uncertainty in this estimate. Sensitivity analysis allows you to rapidly explore the repercussion of this uncertainty to see how robust your results and findings are to changes in parameter values. Sensitivity testing works by carrying out multiple simulations of your model using different initial values. It then aggregates the results of the simulation and presents a concise report on the potential simulation outcomes.

To use sensitivity testing, replace one or more of the values in your model with random variables. For instance, if you had a stock where your best estimate for the initial value was 100 and you determined you could model an estimate for the true initial value using a random variable with a standard deviation of 7, you would set the initial value of the stock to be the following: “RandNormal(100,7)”. You may then run the sensitivity testing analysis algorithm to repeat the simulations many times, each time the value of the stock will take on a different initial value and you may see how the resulting simulation paths change (this is a classic Monte Carlo algorithm).

Following these insights and techniques, we also could be able to make a system dynamic model of Starlight business or any company for an interesting business model. See let’s see how we can convert a Business Model Canvas into a system dynamic model.

2. Forecasting human behaviour by agent-based modelling

To understand the pros and cons of an Agent Based Model versus a System Dynamics model, we explore how these two techniques might approach the same problem: modelling the spread of news or influencing a population. We can classify people in this model as being in one of three states:

- Nonbelievers, but susceptible of being affected by the messages
- Infected Believers.
- Cured disbelievers.

These three types correspond with the common disease virus spread and recovery models also names susceptible, infected, and recovered (SIR) models. Individuals will move between the three states: moving from susceptible to infected to recovered and back to susceptible. The movement from susceptible to infected will be governed by some infection rate equation that takes into account the status of currently infected individuals. The movement from infected to recovered will be governed by the average duration of the disease. Using the System Dynamics methodology, we would model each of the three states by Stock primitives that stores the number of individuals currently in that state. So, for instance, we have a Nonbelievers Stock storing the portion of the population that is currently in the susceptible state. We then use Flows to move individuals between the Stocks based on different factors. For instance, for the flow moving individuals between the Infected and Recovered Stocks, we would use an equation such as *[Infected]*1/[Average Infection Duration]*. If the average infection duration was ten days, this would move roughly 10% of the infected population every day. In Agent Based Modelling, we first create an agent that defines the behaviour of a single individual. We use three State primitives in this model, one to represent each of the three states a person can be in.

We connect these states with Transition primitives that instruct how a single individual moves between the states. Where in the System Dynamics models, we had flows with rates, in the Agent Based models there are transitions that are given probabilities or state transition conditions. These probabilities determine when the transition will be triggered, and the agent will switch states.

The Agent Based approach allows us to implement features that would be impossible using System Dynamics. For instance, we can look at the geographic or network proximity (for example on basis of a Nearest Neighbour calculation) of agents and use this to affect our transmission probability. Susceptible agents that are closer to infected Believer agents are more likely to become Believers than those that are farther away. Similarly, we could look at social structure: how the connections between agents will influence their probability of encountering Believers. All this would not be possible to look at using System Dynamics.

For Agent Based Modelling, we first create an agent that defines the behaviour of a single individual in our model. We use State primitives to represent each of the states a person can be in. We connect these states with Transition primitives that instruct how a single individual moves between the states.

The states that a person can be in is configured using its initial value equation. This equation may be a simple value such as: 1 or 'true' for Believer, or 0 or 'false' for Disbeliever. The equation can also be a logical equation such as: “not true” or a simple mathematical equation like “5*5>10” or a function that depends on other primitives like “not([Smoker] and [Diabetic]), where the […] represents a population segment.

Agent Populations are a key ingredient in the construction of Agent Based Models. There are two primary factors that control an Agent Population: the size of the population (the number of agents), and the agent base for the population. The population size is the number of agents in the population at the start of the simulation (the size can also be changed dynamically during the simulation by destroying agents or creating new agents). The agent base defines what types of agents will be contained in the population. An Agent-Based model can contain one or more definitions of agents (just like it can contain one or more populations of agents), each definition or Agent Base is specified by creating a Folder primitive around the portion of the model that will define that agent. Then the Extension property of the resulting Folder must be set to "Agent". After this is done, the Agent Base property of the Agent Population can be set to the Folder's name.

We summarize these insights for creating an agent-based model for fake news, where non-believers are infected by fake news messages, after which each agent will decide to believe the fake message (stay infected) or become a disbeliever, by the following steps in Insight Maker.

1. Create a new State named Nonbelief.
2. Create a new State named Infected.
3. Create a new State named Disbelief.
4. Create a new Transition going from the primitive Nonbelief to the primitive Infected. Name that transition Infection.
5. Create a new Transition going from the primitive Infected to the primitive Disbelief. Name that transition Recovery.
6. Change the Initial Active property of the primitive Nonbelief to true.

When a state is active, it means a person is in that state. By setting Nonbelief to start “true”, we have the person start in the Nonbelief state.

7. Change the Trigger Type property of the primitive Infection to Probability.

8. Change the Value/Equation property of the primitive Infection to 0.3 (this in 30% of the cases people will become infected).
9. Change the Trigger Type property of the primitive Recovery to Probability.
10. Change the Value/Equation property of the primitive Recovery to 0.2.
11. Create a new Folder named Person. The folder should surround the primitives Nonbelief, Infected, Disbelief, Infection and Recovery.
12. Change the Behaviour property of the primitive Person to Agent.
13. Create a new Agent Population named Population.
14. Change the Agent Base property of the primitive Population to Person.

The model should now look like Figure 8.

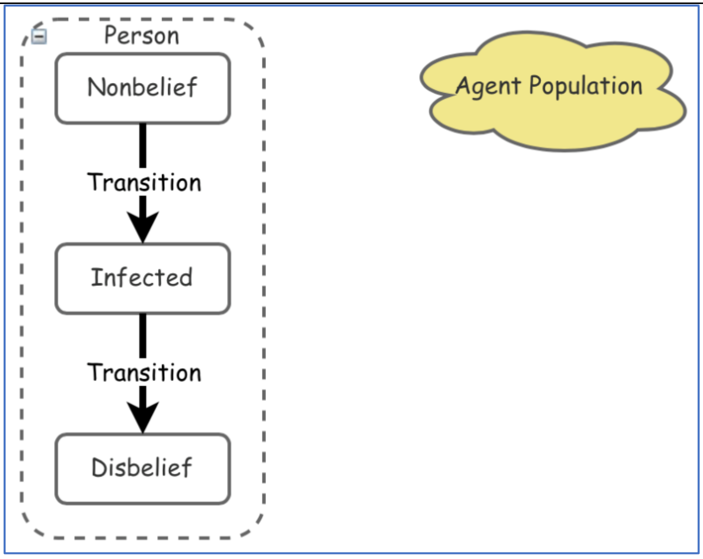


Figure 8: The agent and population model of fake news spread

15. Create a new Variable named Percent Infected.
16. Create a new Link going from the primitive Population to the primitive Percent Infected.
17. Create a new Link going from the primitive Percent Infected to the primitive Infection.

The model now should look like Figure 9.

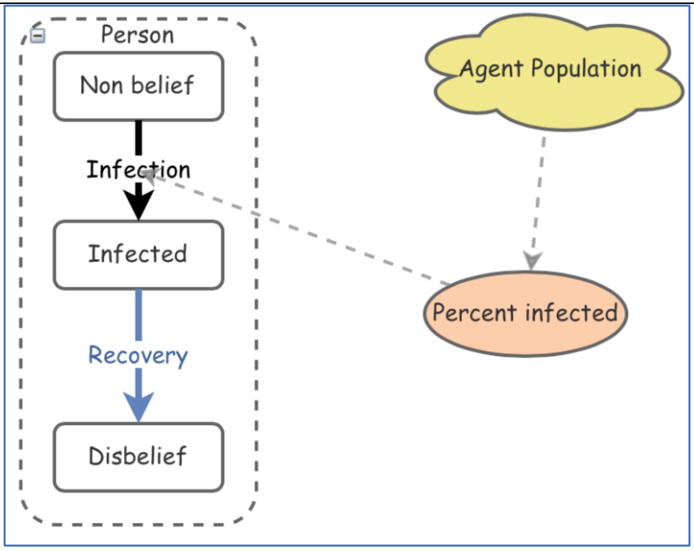


Figure 9: The fake news spread model added with an infection percentage of the population

18. Change the Equation property of the primitive Percent Infected to Count (FindState([Population], [Infected]))/PopulationSize([Population]). This equation uses the FindState function to select all the people in the Population primitive who are in the Infected state. It then divides the count of those people by the total size of the population.
19. Change the Value/Equation property of the primitive Infection to [Percent Infected], and the Recalculate property of the primitive Infection to Yes. The Recalculate property causes the infection rate to be recalculated and updated at every time step of the simulation.
20. Run the model.

    Nothing will happen because we don't have anyone to start the infection. Let's seed the model with a single initial infected person.
21. Change the Initial Active property of the primitive Non-belief to Index(Self) == 1.

22. Change the Initial Active property of the primitive Infected to Index(Self) <> 1.

    By this, each of the people in the population is given a unique Index. The first agent will have an index of 1, the second an index of 2, and so on. "Self" always refers to the current agent. These equations will set the first agent created to start in the Infected state while all the other agents will start in the Non-belief state.

23. Run the model and the outcomes may look like Figure 10.

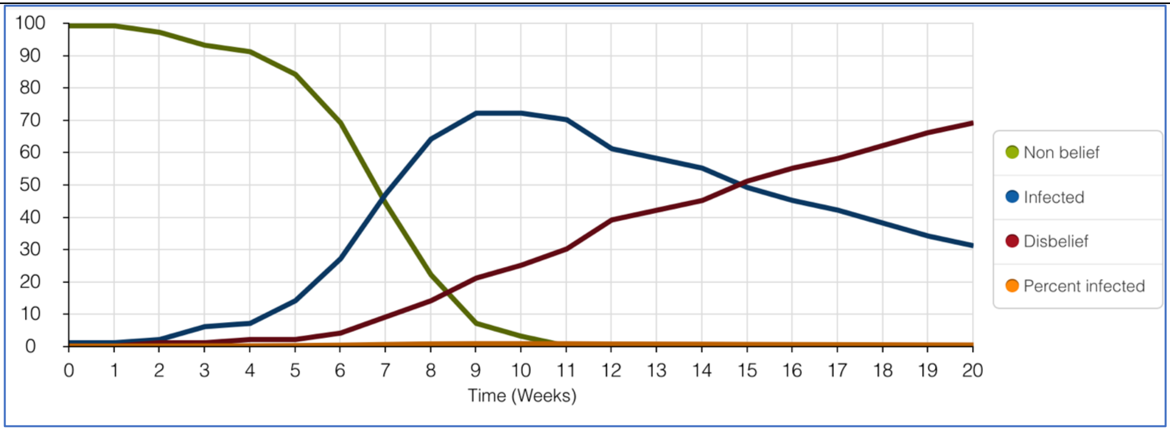


Figure 10: Sample results

The key configuration option for a Transition is what triggers the Transition. There are three different types of triggers:

- Probability: this is a fixed probability per simulation time unit, as we demonstrated above.
- Timeout: A timeout trigger for an infection Transition would switch the Non-belief person to an infected person at a fixed time after the person became Infected (quite likely if people need some reflection time or if it may need several repetitions before a person may switch its state).

- Condition: The condition trigger allows the creation of a Transition that is based on logical relationships to other agents (e.g., a Nearest neighbour) or events (e.g., a moment that money is offered) in the model.

3. Forecasting and foresight

In this manual, we have reviewed methods and techniques for knowing parts of the future with data and models. One may ask the question: How effective are human forecasts? The answer is, depends on how you define effectiveness. If with prediction we can know how the future will be on basis of data of the past or knowledge of the past, the effectiveness will be low. Regarding for example the important subject of macro-economic forecasting, [10] find that macroeconomic forecasts are persistently too optimistic. The common factors related to general uncertainty about U.S. macro financial prospects and global demand drive this overoptimism. The results suggest that an increase in uncertainty-driven overoptimism has dampening effects on next-year real GDP growth rates. Also [11] in their extensive study of forecasting behaviour, find that the best forecasters are not necessarily experts but people with good analytic, critical and collaborative behaviour, but also their predictions are not perfect. So, why bother about the future and knowing anything about it if forecasting is so unreliable? The answer is to not throw the baby with the washing water. Aiming at knowing something about the future is the same as thinking about the future, and that is super valuable because the past is gone but we have to prepare ourselves for the future. Yes, our predictions and preparations may go wrong, but not that wrong as not knowing anything about it. If know reasonable understanding of how the future will behave, any investment in it, not only monetary but also your training and learning efforts or organization, managing and planning will be useful. But we know that it is useful. Banks and companies make profits because they made the wight bat on the future (okay, sometimes wrong, but frequently right). Some people are better prepared for the future than others, not only because of luck.

So, if the social future is not a purely mathematical pattern, how can we talk and analyse it? Schultze and Leidner [12] have identified 4 different ways, named discourses after Deetz [13], of analysing social science knowledge in general, highly applicable to our thinking. These four discourses stem from two dimensions, one dimension reflects on the consensus versus dissensus nature of our thinking about social reality. Clearly much of data science has a highly consensus view stating that data are the objective facts and that analytic algorithms are objectives ways of finding the truth. This is highly different from the scenario and simulation views that propose that we can see the reality from different angles, taking different assumptions, variables, and models, as Insight Maker has shown to be able of. The other dimension is the dimension of a priori or emergent knowledge and insights. The apriori approach starts with some key assumptions about reality that next are rigorously applied and thought through, as we did with our simulations. Some of the popular assumptions here are the assumption of rationalist optimizing behaviour. This approach is also named elitist, because it is used by academics and their superior thinking powers. In contrast knowledge is often highly context sensitive and emerging depending on how the reality changes per day or week or any other period. This emergent approach starts by developing beliefs and cultural norms that change under the influence of social and material conditions in society. For example, 30 years ago most people did not care much about climate change, whereas it has become a key issue in the 2020s.

Following these two dimensions, we can next classify four modes of thinking about the future, and thus add different ways of analysing the future to our analysis. See Table 1.

Table 1: Discourse modes for analysing the future

	Local and emergent modes	Elite and apriori modes
Dissensus modes	Dialogic discourse Goal: Reclaim conflict Method: Deconstruction of reality Hypothesis about the future: Have a more open society with fair participation of all	Critical discourse Goal: Unmask domination Method: Culture and ideology critique Hypothesis about the future: Remove domination and realize a fairer society
Consensus modes	Interpretive discourse Goal: Display the unified culture, beliefs, and norms Method: Hermeneutics (textual) and ethnography Hypothesis: Create a society with more integrated values	Normative discourse Goal: Find lawlike relations Method: Time-series analysis, model development Hypothesis: Know the future and be better in control

In contrast to the data science ideal that data will speak for themselves, the critical, dialogic, and interpretative approaches highly deviate for this simplistic empiricism and propose to go "below" or finding the mechanisms that produce the data. Additionally, they are also critical to the extent into which data can fully represent the events in reality, although this is also a critical perspective of the normative discourse, which aims at reliable and valid representations. So, the actual events are not fully represented by the empirical data. The normative discourse, however, wants to identify patterns and even causations among constructs with data. All the other approaches, however, aim at finding why the data are what they are, to what extent they are incomplete if not even misleading and see finding the mechanisms that create the actual events as their key ambition. This approach to knowing and the future can be summarized as critical realism [14, 15]. We summarize this view as a four layered view of reality and knowing. See Figure 11.

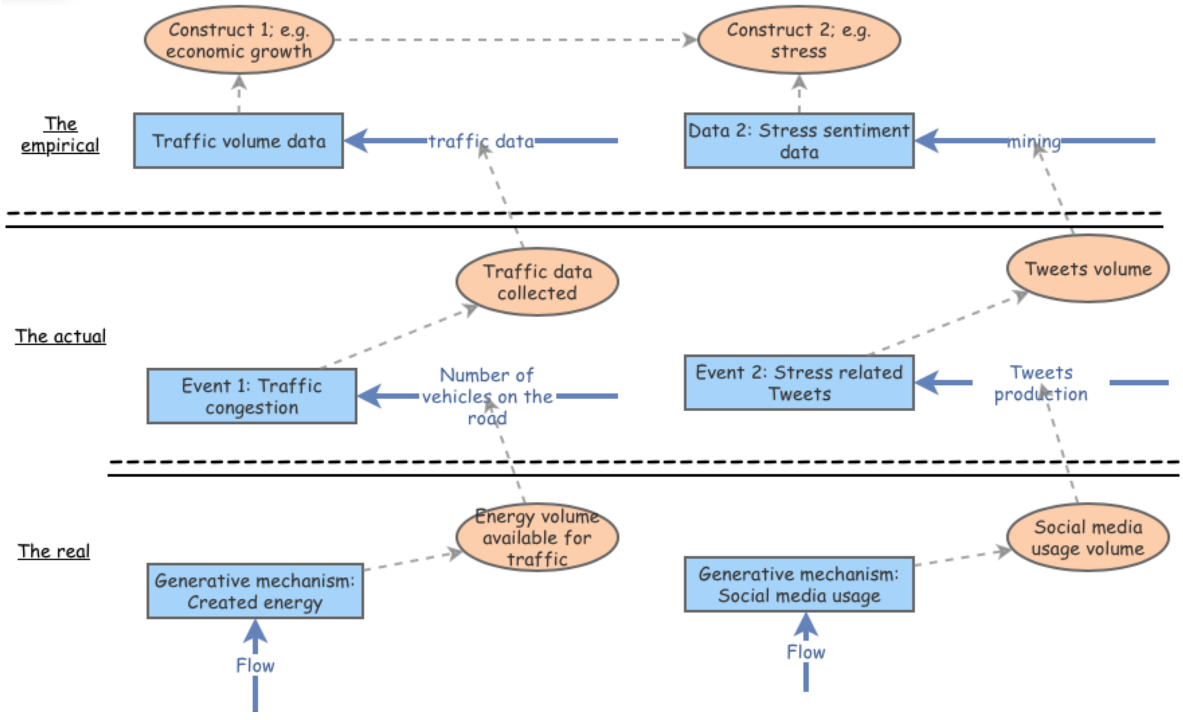

Construct 1; e.g. economic growth
Construct 2; e.g. stress
The empirical
Traffic volume data
traffic data
Data 2: Stress sentiment data
mining
Traffic data collected
Tweets volume
The actual
Event 1: Traffic congestion
Number of vehicles on the road
Event 2: Stress related Tweets
Tweets production
The real
Energy volume available for traffic
Social media usage volume
Generative mechanism: Created energy
Generative mechanism: Social media usage
Flow
Flow

Table 11: Critical realist layers of knowing and reality

This means that finding the generative mechanisms is key to our understanding of the future. These mechanisms can be expressed in system dynamic models, which can be next used for dialog, interpretation, and critique. This is exactly going on in the field of climate change. System dynamic models of the future are used to debate, criticize, and interpret reality for the purpose of decision making for the future.

References

1. Rahim, F.H.A., Hawari, N.N., Abidin, N.Z.: Supply and demand of rice in Malaysia: A system dynamics approach. Int. J. Supply Chain Manag. 6, 234–240 (2017).
2. Isaacs, W., Senge, P.: Overcoming limits to learning in computer-based learning environments. Eur. J. Oper. Res. 59, 183–196 (1992).
3. Sterman, J.D.: All models are wrong: reflections on becoming a systems scientist. Syst. Dyn. Rev. 18, 501–531 (2002).
4. Grieves, M., Vickers, J.: Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In: Transdisciplinary Perspectives on Complex Systems. pp. 85–113. Springer International Publishing, Cham (2017).
5. Hristoski, I., Mitrevski, P.: Evaluation of Business-Oriented Performance Metrics in eCommerce using Web-based Simulation. J. Emerg. Res. Solut. ICT. 1, 1–16 (2016).
6. Lyneis, J.M., Ford, D.N.: System dynamics applied to project management: a survey, assessment, and directions for future research. Syst. Dyn. Rev. 23, 157–189 (2007).
7. Senge, P.M., Sterman, J.D.: Systems thinking and organizational learning: Acting locally and thinking globally in the organization of the future. Eur. J. Oper. Res. 59, 137–150 (1992).
8. Keen, S.: A monetary Minsky model of the Great Moderation and the Great Recession. J. Econ. Behav. Organ. 86, 221–235 (2013).
9. Singh, S., Shehab, E., Higgins, N., Fowler, K., Reynolds, D., Erkoyuncu, J.A., Gadd, P.: Data management for developing digital twin ontology model. Proc. Inst. Mech. Eng. Part B J. Eng. Manuf. 095440542097811 (2020).
10. Chatterjee, P., Nowak, S., Chatterjee, P.: Forecast Errors and Uncertainty Shocks. International Monetary Fund (2016).
11. Tetlock, P., Gardner, D.: Superforecasting: The art and science of prediction. Random House, New York (2016).

12. Schultze, U., Leidner, D.E.: Studying knowledge management in information systems research: discourses and theoretical assumptions. MIS Q. 213–242 (2002).

13. Deetz, S.: Crossroads—Describing differences in approaches to organization science: Rethinking Burrell and Morgan and their legacy. Organ. Sci. 7, 191–207 (1996).

14. Zachariadis, M., Scott, S. V, Barrett, M.I.: Methodological Implications of Critical Realism for Mixed-Methods Research. MIS Q. 37, 855–879 (2013).

15. Mingers, J., Mutch, A., Willcocks, L.: Critical realism in information systems research. MIS Q. 37, 795–802 (2013).