Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

INCREASING MODEL TRANSPARENCY IN SYSTEM DYNAMICS MODELS

Ignacio J. Martinez-Moyano[1]

[1]Decision and Infrastructure Sciences, Argonne National Laboratory, Lemont, IL, USA

ABSTRACT

Models are simplified descriptions of real systems. To increase their usefulness, models need to be as transparent as possible so that the formulation, the data, and all the assumptions about the real system implemented in the model, are salient, evident, documented, and readily available for inspection. In this paper, transparency in models—particularly models formulated using the System Dynamics approach—is discussed and the use of an automated tool—the System Dynamics Modeling Documentation and Assessment tool—for model documentation and assessment is advanced. The proposed tool is showcased using a simple susceptible-infectious epidemics model and the Urban1 model—a simplified version of the full Urban model documented in Forrester (1969). The use of the tool and the benefits derived from using it during the model building process are presented and explained.

1 INTRODUCTION

Transparency in models is critical to gain confidence in model results, for reproducibility of results, and to enhance understanding of the model, the problem under study, and the system in which the problem exists. Transparency refers "to the extent to which interested parties can review a model's structure, equations, parameter values, and assumptions" in an easy and straightforward way (Eddy, Hollingworth et al. 2012, p. 844). Particularly, system dynamics models tend to be more transparent than other types of models due to its (relatively) simple mathematical demands, multistage iterative process, ability to represent model structure using a clear and graphical format, and ability to inspect their formulation in a straightforward manner (see Figure 1, from Martinez-Moyano and Richardson (2013) for a graphical representation of the process).

System dynamics models tend to favor transparency and openness by staying away from black-box implementations. In general, system dynamics models can be shared using easy-to-understand equations written intuitively in plain text (and, on many occasions, in plain, almost natural-language style), intuitive and easy to understand graphics of its structural components, and simple behavior-over-time output graphs and tables. Typically, in system dynamics studies, "the model is a means to an end, and that end is understanding" (Richardson and Pugh 1981, p. 16)—understanding of the model, and understanding of the problem and the system. For an in-depth explanation of the system dynamics approach, several classic publications are very useful including work by Forrester (1958, 1961, 1968), Richardson and Pugh (1981), Sterman (2000), Martinez-Moyano and Richardson (2013), and Martinez-Moyano (2018), among others.

In order to increase model transparency in system dynamics models, researchers at Argonne National Laboratory developed the System Dynamics Modeling Documentation and Assessment (SDM-Doc) tool that inspects models and provides automated documentation and assessment leading to enhanced transparency (for more information, see Martinez-Moyano 2012). The SDM-Doc tool provides an automated way to increase transparency in System Dynamics models and associated dynamic behavior. In the development of the SDM-Doc tool, findings from the best practices in system dynamics study conducted by Martinez-Moyano and Richardson (2013) were used as a guiding set to implement in an automated way. Additional input was received by past presidents of the System Dynamics Society, scholars in System

Dynamics, workshop participants over many years (more than a decade), and by hundreds of power users of the tool all over the world.

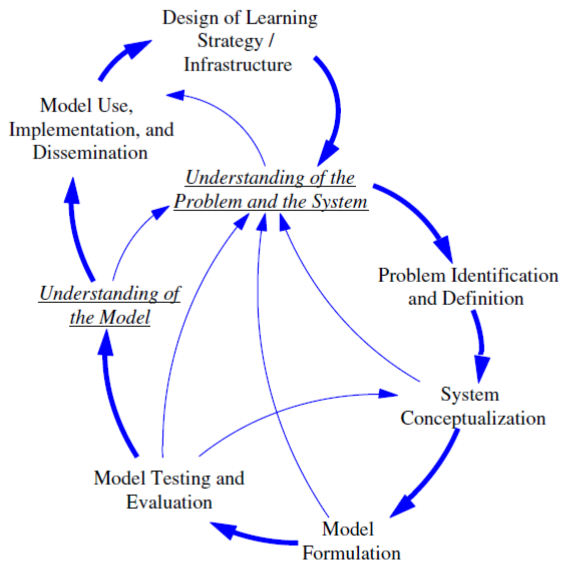


Figure 1: Overview of the system dynamics modeling approach.

2 SYSTEM DYNAMICS MODELING

2.1 The Modeling Process

System Dynamics (SD) focuses on the analysis of feedback structures to learn about the drivers of problems in complex systems. Different SD modeling approaches exist in the literature (for a synthesis of four different approaches see Table 3 in Martinez-Moyano and Richardson 2013, p. 108). Although these approaches differ in some ways, the focus of the modeling effort is always the same: to increase the level of understanding of the problem being studied and of the system in which the problem exists. In SD, the existence of a problem drives the modeling effort, and the model-building process is iterative by design to allow for a new understanding about the problem to generate refinements of the problem definition and of the model. The modeling process has several stages that are followed to enhance our understanding of the problem and of the system (see Figure 1).

The process starts by identifying and defining the problem of concern in the system ("*problem identification and definition*"). In this step, the modeler clarifies the purpose of the modeling effort and makes sure that all the stakeholders understand both the purpose of the model and their role in the modeling effort. A clear statement of the purpose of the model provides "the imaginary line separating what is considered (for modeling purposes) to be inside the system and what is considered to be outside" (Richardson and Pugh 1981, p. 48) of it (i.e., the system boundary). To be useful, the boundary chosen should enclose the system of interest. The boundary should include the minimum number of components whose interactions generate the problematic behavior of interest, and the problem of interest needs to be identified in dynamic terms—in behavior-over-time terms, or time series—before the boundary can be identified. Reference modes of behavior, and an appropriate time horizon are identified as part of this step.

From *problem identification and definition*, the process moves to *system conceptualization*. In this step of the process, a dynamic hypothesis—"an initial concise overview of a feedback structures believed to be responsible for the problem behavior" (Martinez-Moyano and Richardson 2013, p. 29)—is formulated. The dynamic hypothesis also includes the dynamic behavior—time series—of the main variable(s) of interest in the feedback structure proposed. This pair of elements, the systemic structure (set of interconnected variables of interest), and the behavior of such variables (behavior over time, or time series, of the variables of interest) is a key aspect of the dynamic hypothesis that should capture the main dynamic drivers of the problematic behavior as seen from the different points of view of the stakeholders. Part of the purpose of the creation of the dynamic hypothesis, and of the dynamic model afterward, is to produce an endogenous explanation of the problematic behavior of interest.

The next step in the process is *model formulation*. The formal model should always be based on the dynamic hypothesis obtained during the *system conceptualization* stage. The model should start small and simple, as close as possible to the original dynamic hypothesis, and build toward complexity and completeness by adding variables and loops only as evidence is found and understanding developed. All equations written during the formulation step should be logical, make sense, and follow real-world processes. Additionally, only information that is available to decision makers in the real world should be made available to decision makers in the model. SD models are operational models that represent both the mechanics (physics) and the decision-making processes in the real system under study.

Once the model is formulated, *model testing and evaluation* follows. Although "validation and verification of models is impossible" (Sterman 2000, p. 846), model testing is paramount and should start with the first equation written. In SD, it is never too early to conduct model testing, and it is never too late to abandon a model if it does not help in understanding the problem of interest. Over many decades, modelers interested in SD have developed specific tests that include boundary adequacy, structure assessment, dimensional consistency, parameter assessment, extreme conditions, integration error, behavior reproduction, behavior anomaly, family member (class of system), surprise behavior, sensitivity analysis, and system improvement, among others (Forrester and Senge 1980, Barlas 1996, Sterman 2000).

As shown in Figure 1, the process is iterative and can go back to previous stages as needed. The idea is to refine previous iterations of problem identification, improve formulation, continue developing trust in model results, and continue enhancing understanding of the model and of the problem and the system as the modeling process advances. After testing, the step of *model use, implementation, and dissemination* is next in the process. The whole modeling process should revolve around the stakeholders' problems of concern, as well as increasing the understanding of what to do about these problems. A logically coherent model that is mathematically characterized and that can be simulate and that has been thoroughly tested should be used to explore the problem of interest with the stakeholders and other interested parties, keeping in mind implementation and system change. In this phase, policy options are identified and designed for implementation. Modelers and stakeholders collaborate in the phase to identify, and thoroughly understand, how to change the system under study and alleviate the identified problem (or, potentially, fix it).

The final step in the process is the *design of learning strategy and infrastructure*. After thoughtfully constructing the model and thoroughly considering how to implement lessons derived from it, descriptions of model-based insights expressed as (relatively) simple stories – illustrated using as-simple-as-possible diagrams – can be used with the stakeholders as a powerful mechanism to create a long-term learning process for change. As stated before, the modeling process is iterative; it builds on knowledge and insights developed in previous iterations, which allows for enhanced understanding, additional knowledge, and more powerful insights to be developed every time one cycle of the whole process is completed (including intermediate iterations as needed).

2.2 System Dynamics Tools

Causal-loop diagrams represent closed causal connections, forming a circular pattern. The shortest possible loop consists of two variables (called vertices or nodes in network and graph theory) connected with two directional causal links (called edges) represented by directional arrows. Each causal link is represented with a polarity sign to characterize the type of influence the variable at the beginning of the arrow (cause) has on the variable at the end of the arrow (effect). There are two possible link polarities: positive and negative. A positive causal polarity ("+" sign) means that, all else equal, the two variables will follow the same trajectory over time; increases (or decreases) in the variable at the beginning of the arrow will result in increases (or decreases) in the variable at the end of the arrow. Similarly, a negative causal polarity ("–"

sign) means that, all else equal, the two variables will follow opposite trajectories over time; increases (or decreases) in the variable at the beginning of the arrow will result in decreases (or increases) in the variable at the end of the arrow. For example, as shown in the left and central panels of figure 2, the relationship between *birth rate* and *population* is positive because as *birth rate* increases, *population* increases (more people are born) while the relationship between *death rate* and *population* is negative because as the *death rate* increases, *population* decreases (because more people die). In this case, both rates influence *population*, and *population* influence both rates (see right panel of Figure 2). These three variables are linked in such a way that two feedback loops are created, one reinforcing (identified with a letter "R" inside a curved arrow) and one balancing (identified with a letter "B" inside a curved arrow). Both loops are active at the same time and have an influence on the behavior of the variable *population*. *Population*, in this case, can increase or decrease over time depending on the strength (gain) of the two different loops that influence it. When both loops have the same gain, *population* remains constant over time (the people that die are replaced with newborns) generating a dynamic equilibrium.

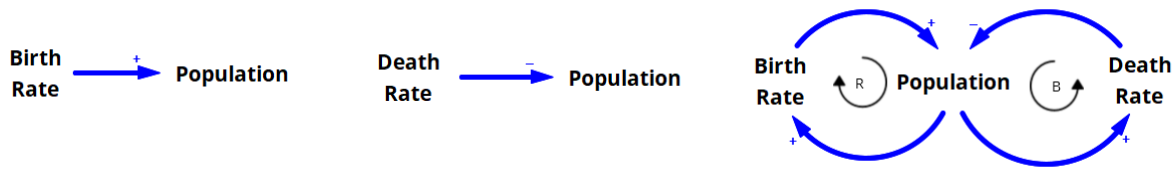


Figure 2: Causal links and causal loops.

In SD, stocks, also called "levels," are accumulations in the system that "characterize the state of the system and generate the information upon which decisions and actions are based" (Sterman 2000, p. 192). When thinking about stocks (accumulations) in a system, the metaphor of a bathtub is useful. A bathtub can be seen as a stock because it accumulates water when the inflow of water from the faucet exceeds the outflow of water through the drain. In the same way, the *birth rate* accumulates people into the *population* stock while the *death rate* depletes the *population* stock. In SD, stocks are represented with rectangles and the flows that change them are represented with double-lined arrows (representing pipes) and faucet-like icons (representing valves that control the flows) going into and out of the stocks (see Figure 3). It is important to note that stocks can only be modified by flows. The cloud-like icons at the beginning/end of the double-lined arrows represent both sources and sinks for the flows. These sources or sinks represent infinite stocks which feed the flows and are outside the boundaries of the model (their associated causal mechanisms are not of interest to the modeler and can be ignored or collapsed into the source or sink).

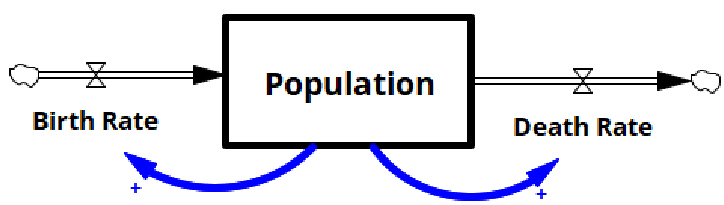


Figure 3: Stock-and-flow representation.

3 SDM-DOC: A TOOL FOR ENHANCED MODEL TRANSPARENCY

3.1 SDM-Doc

The SDM-Doc tool is an automated documentation and assessment tool that uses system dynamics models as input to produce detailed HTML-based documentation and assessment reports. The SDM-Doc tool is free and open source (under a BSD open-source license) and is available at the System Dynamics Society website (https://systemdynamics.org/sdm-doc). The tool is available for Windows and for Unix operating systems. The source code is also available on the same webpage. The tool was originally developed for models developed in Vensim (see https://vensim.com/), and effort is on the way to be able to process models developed in Stella (https://www.iseesystems.com/), Powersim (https://powersim.com/), and XMILE (https://docs.oasis-open.org/xmile/xmile/v1.0/xmile-v1.0.html). The XMILE (XML Modeling Interchange LanguagE) standard describes a standard format for sharing and distributing system dynamics models (see https://systemdynamics.org/resources-old/xmile/).

3.2 SDM-Doc GUI

One can access the SDM-Doc tool through a simple graphical user interface (GUI) as depicted in Figure 4. The GUI provides access to different configuration options through four basic sets of choices: "File", "Language", "Tools", and "Help". The "File" option includes a detailed set of output options and will be described last.

"Language": In the "Language" option, the user can access language administration menus that allows for different languages to be used (the current implementation includes English, Japanese, and Spanish). Here, new languages can be registered and can be updated as needed. As the tool evolves over time, the need for additional labels to be translated emerges.

"Tools": In the "Tools" option, the user can select among eight processes related to the model. The options available include: to extract comments from models, merge new comments in models, extract the names of all the variables in the model, replace variable names as needed, run a process to obfuscate models, access a tool to automatically generate maps based on adjacency matrices in Microsoft Excel, generate model copies, and run an independent loop-length analysis module. The comments and names functionality are implemented using Microsoft Word documents to interact with the tool and model being used.

"Help": In the "Help" option, the user can access the open-source license of the tool, a point of contact for feedback, and a list of the collaborators that have made the tool a reality (including development, design, and internationalization).

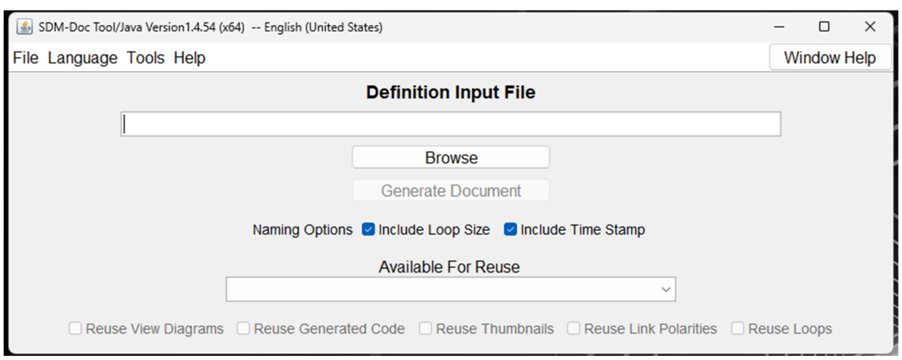


Figure 4: SDM-Doc user interface.

"File": The "File" option provides access to the model assessment results options menus which include "model information," "potential omissions, arrow information, and analyze loops," "variable information warnings," and "graph and output options" as depicted in Figure 5.

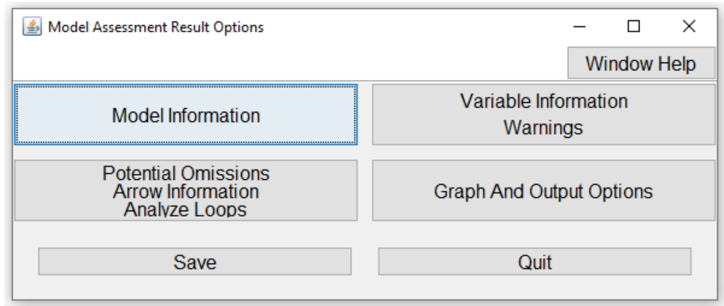


Figure 5: SDM-Doc model assessment result options.

The "model information" section allows for the customization of the results related to this area of the model. The options can be selected by checking the boxes next to each option. The model information section has an option, using a text box, to select the maximum length of the endogenous path to be explored when the information related to endogenous variables is requested. The length of the path will determine the maximum number of variables to be included in each path of the search conducted by the tool. This parameter has an important impact on the results generated by the tool and on the time that is required for the tool to identify information related to endogenous variables in the model. The maximum theoretical length needed to include all possible paths of endogeneity is the total number of variables in the model (i.e., there cannot be a path with more variables than the total number of variables in the model). In practice, the longest path is never as long as the total number of variables in the model. In small models, the choice of maximum length of the endogenous path has little impact on processing time. However, in large models, the impact can be quite large, potentially leading to extremely long processing times making the tool not useful for quick analysis of models as these are developed.

The "potential omissions, arrow information, and analyze loops" section includes options related to elements that describe parts of the structure of the model which deal both with the formulation and the identification of causal mechanisms in the model including causal links and feedback loops.

As in the case of the analysis of endogenous paths just described, loop analysis is based on a parameter that controls the maximum number of variables in each loop that the tool will search for ("loop size maximum"). For example, if the loop size maximum is set to 20, the tool will explore the model and search for feedback loops (cycles, in graph theoretical terms) of lengths up to 20 variables. Therefore, if the model has feedback mechanisms which include more than 20 variables, these loops will not be identified as the tool will stop searching once it identifies the 20th variable in the loop. Also, given that the loop length can have an important impact on the time it takes the tool to go over all the loops in large models, an additional parameter can be set to limit the search to a maximum number of minutes ("max search time in minutes") that prevents the tool from continuing the search for loops after a specified amount of time. When the search is constrained by the time limit, the results are displayed; however, these might not be accurate or stable because the search becomes truncated and there is no mechanism to know how many more loops would have been found had the time been set to longer than it was set to. In addition to the complexity of the model (number of variables and causal links among them), the time it takes the tool to conduct the search is also a function of the processing speed of the computer and the amount of random-access memory available to the tool. All these considerations are not problematic in the case of relatively small models. However, (relatively) large models with hundreds or thousands of variables can potentially yield millions or billions of feedback loops. Currently, the tool identifies unique loops, and work is underway to identify sets of independent loops that can provide additional insight into model structure beyond raw counts of feedback loops. Additional ways of understanding the importance of feedback loops in models include work related to structural dominance analysis (Oliva 2016, Oliva 2020), loop eigenvalue elasticity analysis (Kampmann and Oliva 2006), partition heuristics and feedback structure decomposition (Oliva 2004), and the loops-that-matter method (Eberlein and Schoenberg 2020, Schoenberg, Davidsen et al. 2020, Schoenberg, Hayward et al. 2023), among others. There is effort underway to implement some of these approaches to loop analysis into the SDM-Doc tool soon.

The loop analysis section also allows for the customization of the reports that the tool generates including the possibility of writing all loops to disk to be used for additional analytical purposes. To make the loops search quicker and the report more rapidly accessible, the user can define the number of loops to be included in separate files.

In the "variable information and warnings" section, in addition to being able to customize the reporting of aspects related to variables and warnings, the user can enable an option to ignore embedded constants (zeros and ones) in model equations. The tool defaults to the identification, and reporting, of zeros and ones as embedded data in the model. In some cases, some modelers prefer to ignore embedded zeros and ones because of the extensive use in certain formulations in their models. The use of any data directly in an equation can become a hidden problem that should be addressed.

The tool, as part of the documentation process and to increase transparency, creates graphical representations of all table functions in the model so that the function form of the nonlinear relationships used in the model is clear and transparent. In the "graph and output options" section (see Figure 9), users can customize graph-related aspects of the reports produced by the tool including the width of the graphs and where to center a highlight point in the graphs. Following best practice (Sterman 2000, Martinez-Moyano and Richardson 2013), the tool defaults to the "1,1" coordinate. In this section, the user can also limit the number of thumbnail graphs produced (defaults to all graphs produced) and define the maximum number allowed (the tool defaults to 1,000). The thumbnail graphs are produced via an internal simulation engine that creates baseline results of the model. In this section, additional output is controlled. The tool can also generate cross-impact matrix multiplication applied to classification (MICMAC) analysis of the model (including all intermediate matrices) and can create a ZIP file with all the output and the original model included. Lastly, the user can also control the creation of flat files with all model-related information.

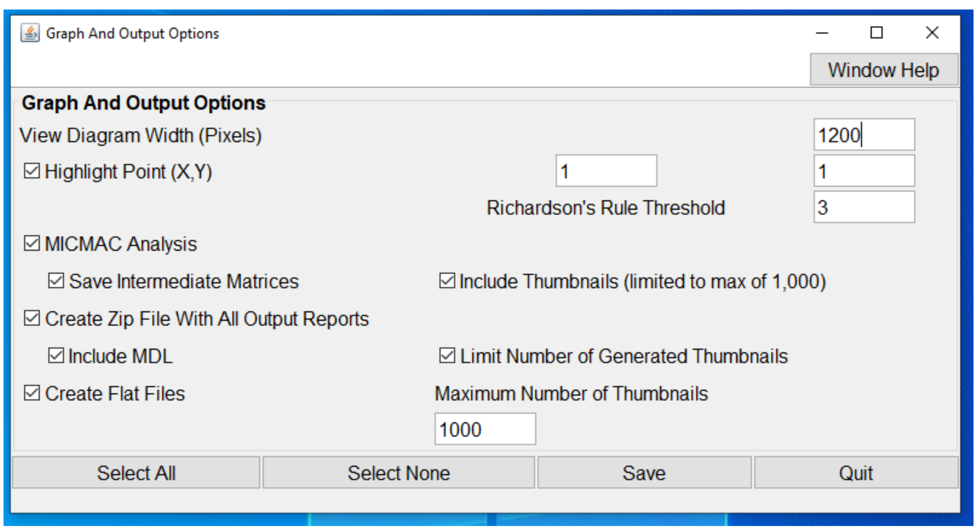


Figure 6: SDM-Doc graph and output options.

The main output file of the tool is an HTML file that includes most of the results and is linked to other files as needed (see Figure 7).

The HTML documentation of the model includes results of the four sections mentioned earlier and hyperlinks to other elements of the model structure. At the top of the file, one can find quick links to a list of all variables in the model ("all variables" link), a variable link detail list including information about polarity and about in-links to variables and out-links from variables ("variable link detail" link), a list of variable types in which the variables are grouped by types of different sorts ("variables types" link), a list of views in the model (“views” link), a list of user-defined groups of variables (“groups” link), a list of the units used in the model (“units” link), a list of the macros used in the model (“macros” link), a list of all the variables in the model and their feedback loops sorted by number of loops (feedback loops” link), a list of all the feedback loops (“loop list No IVV” link), a list of the loops including analytical initialization loops when requested (“loop list with IVV” link), a list of the exogenous variables (“exogenous variables analysis” link), a list of the endogenous variables with information related to the endogenous path to exogenous variables (“endogenous variables analysis” link), a list of the computed polarities of all causal links identified (“link polarity” link), a view summary with information related to variable density per view (“view summary” link), a profile with detailed information of where variables exist in each view (“view-variable profile” link), a list of all pairs of variables present in feedback loops including detailed information about the length of the loops and the specific loops to which the pair belongs (“variable relationships” link), and a list of all variables and their feedback loops including loop length and specific loop identifiers (“variable loops” link).

Model Assessment Results

Model Information	Result
Total Number Of Variables	11
Total Number Of State Variables	2 (18.2%)
Total Number Of Stocks	2 (18.2%)
Total Number Of Exogenous Variables	8 (72.7%)
Total Number Of Endogenous Variables	4 (36.4%)
Total Number Of Feedback Loops No IVV (Maximum Loop Length: 42) [2, 2]	2 (1|1|0)
Total Number Of Feedback Loops With IVV (Maximum Loop Length: 42) [2, 3]	3 (1|2|0)
Total Number Of Causal Links	11 (8|3|0)
Total Number of Rate-to-rate Links	0
Number Of Units Used In The Model (Basic/Combined)	0/0
Total Number Of Equations Using Macros	0 (0.0%)
Variables With Source Information	0 (0.0%)
Dimensionless Unit Variables	1 (9.1%)
Variables without Predefined Min or Max Values	11 (100.0%)
Function Sensitivity Parameters	0 (0.0%)
Data Lookup Tables	0 (0.0%)
Time Unit	Day
Initial Time	0
Final Time	30
Reported Time Interval	TIME STEP
Time Step	0.03125
Model Is Fully Formulated	Yes
Model Defined Groups	Yes

Figure 7: SDM-Doc partial example results (susceptible-infectious, or “SI” model).

4 SDM-DOC USE CASES

In the tutorial, two models will be documented and tested using the SDM-Doc tool. The first model is the simplest epidemics model that exists, the SI (susceptible-infectious) model (for an explanation of this model, and additional relevant variations, see chapter 9 in Sterman 2000). The use of this simple use case will help attendees understand the usefulness of the tool as part of the model development process. The second model used is a simplified version of Forrester’s Urban Dynamics (1969) model (for a policy-oriented description of the model, see Ghaffarzadegan, Lyneis et al. 2011).

4.1 SI Model

The simplest epidemics model possible is a susceptible-infectious (SI) model (see Figure 8). The model shows the relationship between two populations stocks ("susceptible" and "infectious") and an infection rate.

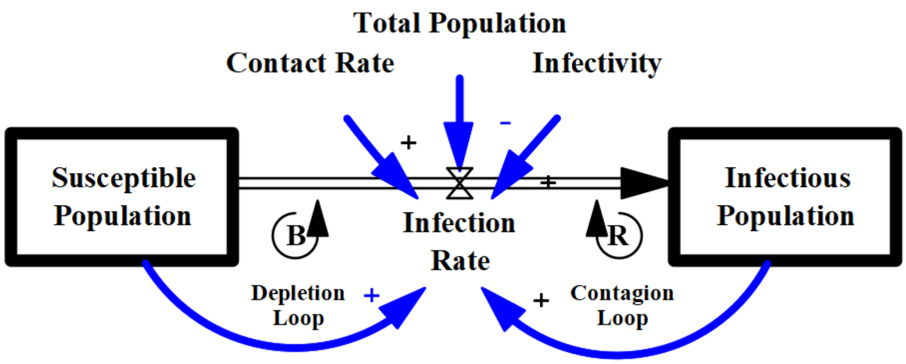


Figure 8: SI model structure.

In the model, individuals in the susceptible population stock, by becoming infected, flow to the infectious population stock. All other variables are linked to the infection rate with arrows representing causal connections. Arrows indicate the direction of causality between variables. Each causal link is represented with a polarity sign to characterize the type of influence the variable at the beginning of the arrow (cause) has on the variable at the end of the arrow (effect). Additionally, Figure 8 identifies two feedback loops: a depletion loop and a contagion loop. The contagion loop (on the right) is a reinforcing loop (also called "positive" loop) that links the infectious population with the infection rate. Reinforcing cycles are engines of growth (or demise) that, when left unchecked, can generate explosive dynamics. In this case, as the infection rate increases, the infectious population increases and, at the same time, as the infected population increases, the infection rate increases reinforcing each other. The depletion loop (on the left) is a balancing loop (also called "negative" loop). In balancing processes, the trajectory of the behavioral path of the variable of focus reverses with each completed cycle. In this model, as the infection rate increases, the susceptible population decreases; at the same time, as the susceptible population decreases, the infection rate decreases as well. The "raw" material for the infection process declines over time slowing down the infection process of the epidemic. In this type of model, the susceptible population becomes infected via the infection rate which is a function of the contacts that people have with infectious individuals and of the infectivity of the specific disease being simulated.

The contact rate is measured in people contacted per person per time period of interest. For some types of diseases, the time period used for a typical simulation of disease progression is days making the units of the contact rate 1/days. Infectivity represents the probability that a person will become infected after exposure to someone with the disease. Infectivity is measured in dimensionless units. The infection rate is the total number of encounters of the susceptible population (susceptible population*contact rate) multiplied by the probability that any of such encounters is with an infectious person (infectious population/total population) and multiplied by the probability that an encounter with an infected person results in infection (infectivity). The infection rate is measured in people/day units. To start a contagion process, the initial infectious population must be at least one individual. If more than one infectious individual is introduced to the simulated system, the dispersion process will be faster due to the increased probability that susceptible individuals will be in contact with an infectious individual. Also, increasing the number of contacts, or increasing the infectivity of the disease, will have a multiplying effect on the process. Figure 10 shows the typical dynamics that this type of model generates. In this example, the contact rate used is 1.75 contacts per person per day, and the infectivity is 0.3 (dimensionless). The initial number of infectious individuals is 1 person, and the total number of people in the systems is 100 (including the infectious person). The simulation runs for 30 days and is named “Base.”

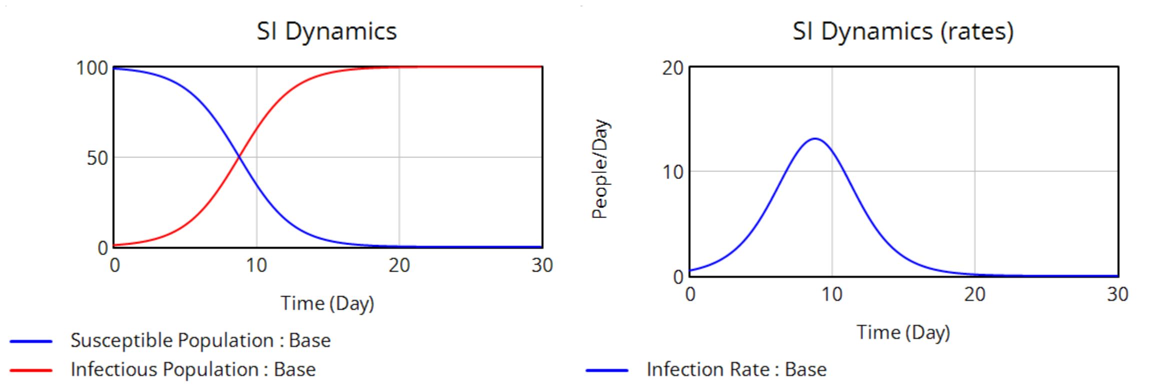


Figure 9: SI model dynamics.

On the left side of Figure 9, the dynamics of the stocks (accumulations) show s-shaped growth (and decline) patterns. On the right side of the figure, the rates of change are shown (in this case just one rate, the “infection” rate) depicting an initial increase that peaks at about 8.5 days to then decline and eventually go to zero (when the epidemic is no longer active). One should note that the point in time where the infection rate peaks (~8.5 days) is the point in time where the susceptible population and the infectious population lines meet. In fact, that is an inflection point for both variables and corresponds to the point in which the depletion feedback loop on the left of Figure 8 overpowers the contagion loop on the right of the same figure. From that point forward, the fact that the susceptible population is being depleted is what matters most in the dispersion process, forcing the infection rate to decrease over time.

Using the SDM-Doc tool to create documentation and assessment of the SI model one can see many aspects of the model that otherwise would have stayed obscured and would require additional effort and interest to uncover and analyze. Being able to connect model structure (Figure 8) with model behavior (Figure 9) is crucial to enhance understating of the model and to increase confidence in its results and usability. SDM-Doc documentation of each equation includes many relevant aspects of how the structure and the behavior of the model are connected. On Figure 10, a partial list of variables of the SI model are shown (stocks). As shown in the figure, the equation is shown with hyperlinks to the variables in the equation so that navigating across equations is quick and easy. Also, the documentation includes a description, the views in which the variables exist, what other equations use the variable of focus, the number of feedback loops that the variable is a part of (also disaggregated into positive and negative), and a thumbnail representation of the behavior using the base parameters of the model.

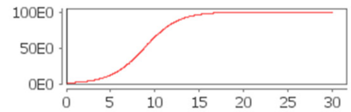
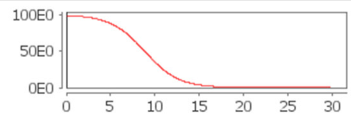
Top	(Type) Level (2 Variables)		
Group	Type	*Variable Name And Description*	Thumbnail
.SI-Model	#2 L,LI	**Infectious Population** (People) = ∫ Infection Rate *dt* + Initial Infectious Population **Description:** The infectious population accumulates the infection rate less the recovery rate. **Present In 1 View:** • View 1 **Used By** • Infection Rate The infection rate is the total number of encounters Sc multiplied by the probability that any of those encounters is with an infectious individual I/N, and finally multiplied by the probability that an encounter with an infectious person results in infection i. • Susceptible Population The susceptible population, as in the simple logistic epidemic model, is reduced by the infection rate. The initial susceptible population is the total population less the initial number of infectives and any initially recovered individuals. **Feedback Loops:** 1 (50.0%) (+) 1 [2,2] (-) 0 [0,0]	
.SI-Model	#5 L	**Susceptible Population** (People) = ∫ - Infection Rate *dt* + Total Population- Infectious Population **Description:** The susceptible population, as in the simple logistic epidemic model, is reduced by the infection rate. The initial susceptible population is the total population less the initial number of infectives and any initially recovered individuals. **Present In 1 View:** • View 1 **Used By** • Infection Rate The infection rate is the total number of encounters Sc multiplied by the probability that any of those encounters is with an infectious individual I/N, and finally multiplied by the probability that an encounter with an infectious person results in infection i. **Feedback Loops:** 1 (50.0%) (+) 0 [0,0] (-) 1 [2,2]	

Figure 10: SI model list of documented variables (stocks).

Additional useful information available includes a list of all pairs of variables connected with causal links (the "variable relationships" report) with information about how distant are from one another and what feedback loops have in common. In the SI model, the variables "infection rate" and "infectious population" are linked in only one feedback loop ("Loop 1"), and they are one step away from one another. Because system dynamics models are directed graphs, the distance for A to B and the distance from B to A are reported separately. Also, the minimum and maximum distances are reported (in the case of several feedback loops).

The system dynamics approach favors transparency and easiness of communication of models and, consequently, system dynamics models are usually built in different "views" or relatively small pieces that are interconnected. In some cases, keeping track of all views and what variables exist in each can become problematic, especially in the case of relatively big models. The SDM-Doc tool creates a report with all the variables and views clarifying what variables exist in what view.

The views-and-variables report is very useful in the development process and when a review of a model developed by someone else is needed. When some variables are not present in any view, this is also reported in the "not in view" column. When some variables are not present in any view, this can be a deliberate choice by the modeler, but one that increases model opaqueness. Alternatively, this can be an omission that needs addressing. The most common variables to find in this category are the control variables of the model ("final time," initial time," "saveper," and "time step").

Another aspect of model complexity that is reported by the SDM-Doc tool is the MICMAC analysis (Arcade, et. al, 1994) of the network of dependencies in the model. Using the MICMAC process, a direct influence and exposure (dependency) analysis is performed automatically (see Figure 11 for results). Four quadrants are identified in the figure. Variables in the upper right quadrant are variables with high levels of influence that also have high levels of exposure (dependency) on other variables. Variables in each quadrant represent mechanisms for change in the model based on different characteristics and network properties.

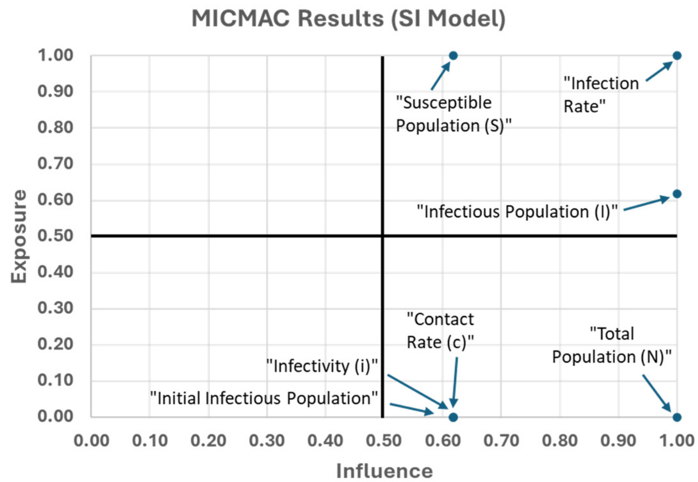


Figure 11: MICMAC results.

4.2 Urban1 Model

The Urban1 model is a simplified version the full Urban model documented in Forrester (1969). This model depicts the dynamics of a prototypical urban area using only three state variables: “business structures,” “population,” and “housing” (see Figure 12). The model describes the interplay of business development, housing development, and population flows as drivers of urban growth, stagnation, and decay. In the model, the land area that the urban area occupies is fixed and, consequently, business structures and houses compete for land in the city. This specific assumption has been thoroughly explored and tested in subsequent work; findings suggest that the fixed area assumption does not change the main dynamics depicted. At the core of the model is the attractiveness of the area for people and businesses both in terms of the labor market and of the housing market. Business developers increase activity in a period of growth when land is available, and a relative surplus of labor exists. As more businesses are developed, people move into the area to take advantage of the labor opportunities. As more people move into the areas, incentives for housing development rise, and more houses are built. Additional housing attracts more people into the area as growth continues. Businesses and houses compete for land throughout the growth process of the city; as the land becomes less and less available, the incentives to build more businesses decline negatively impacting the labor market.

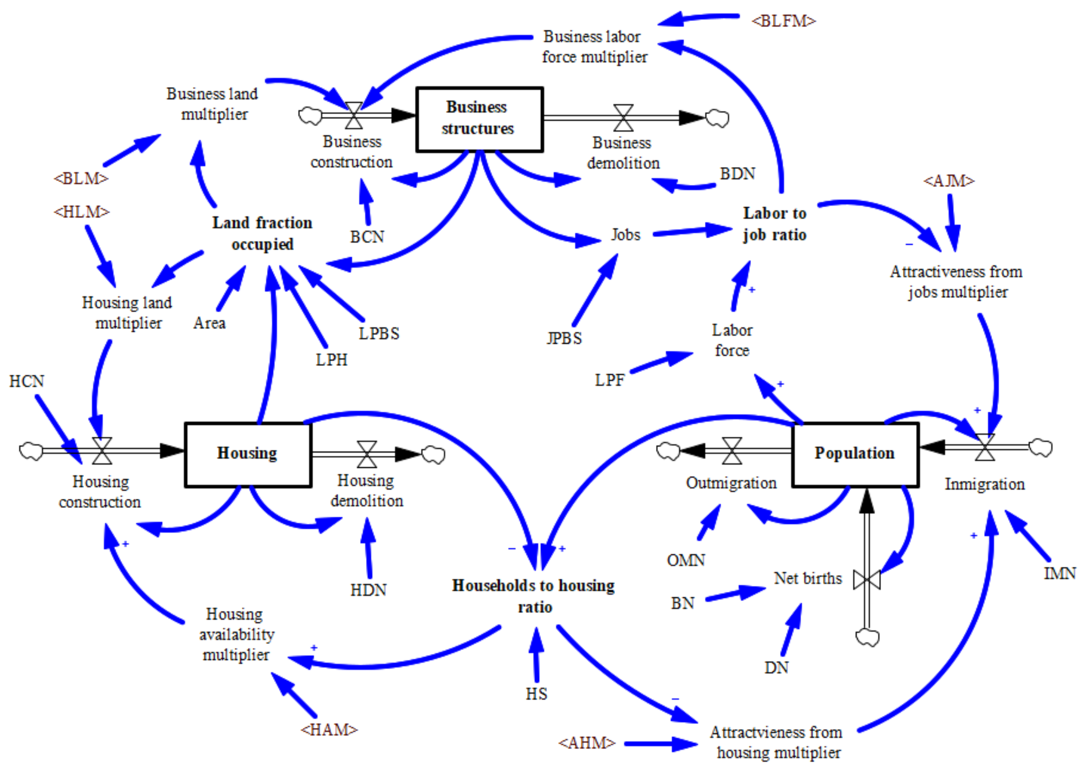


Figure 12: Urban1 model structure.

As business growth stops and businesses find incentives to move to other areas, the labor market suffers and people eventually leave the area creating stagnation first, and then collapse of the urban area (several cities in the United States and the world have gone through this process). As stated previously, because business structures and housing compete for land, relative longevity of businesses vs housing becomes critical. Businesses tend to have a much quicker turnaround time than houses adding to the results shown in Figure 13 (business structures and population decline on the left side of the figure and constant housing after the growth period ends on the right side of the figure). This results in a city with too many houses (many of them vacant) and too few businesses and people.

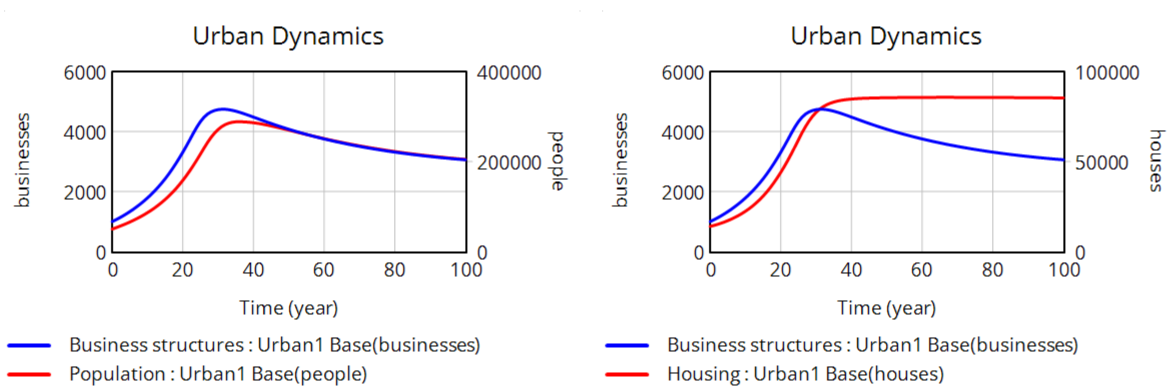


Figure 13: Urban1 model dynamics.

One important model building practice in system dynamics is to always use monotonic functions when formulating a model (for an application of this concept see Repenning 2003). If a non-linear function used in the model is non-monotonic, the polarity of the causal paths can change (from positive to negative and vice versa) during the simulation making the structure of the model and the results produced less transparent and potentially problematic to understand. No mathematical problem is associated with using non-monotonic functions; however, it can be an indication of a lack of understanding of the distinct feedback processes at play in the model or an indication of a lack of clarity in the modeling process. In Urban1, two functions are non-monotonic (see Figure 14). The use of the SDM-Doc tool helps to identify non-monotonic functions in an automated way. In the case of Urban1, the reasons for using this type of function are related to the computing cost at the time (i.e., 1969). A more compact formulation was paramount at the time to get results in an efficient way. Today, computing cost is almost negligible, and it is preferable to gain clarity in model formulation and in causal paths polarity and trajectory by avoiding the use of non-monotonic functions in the model.

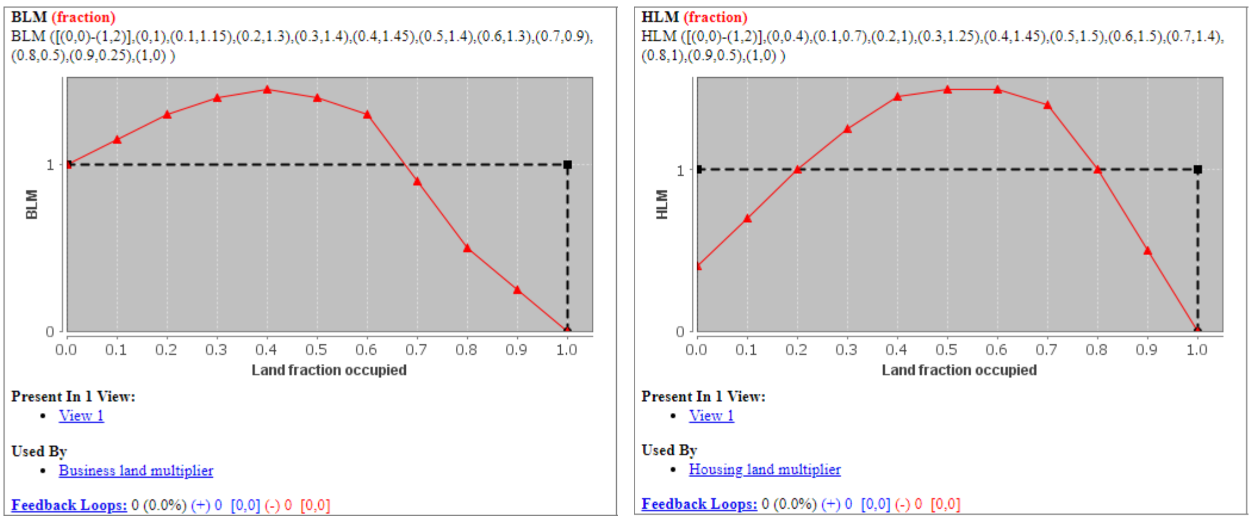


Figure 14: Urban1 non-monotonic functions report.

5 SUMMARY AND CONCLUSIONS

The system dynamics approach uses a closed-loop approach to go beyond open-loop thinking and reveal in a holistic way ***why*** things change over time. System dynamics modelers develop empirically based, broad-boundary models that provide explanations, based on their feedback structures, of the behavioral evolution of the problems of interest. Using this approach, "models go through constant iteration, continual questioning, testing and refinement" (Sterman 2000, p. 87), creating the possibility of rapid deduction-induction cycles in the process. Additionally, the system dynamics approach favors transparency and reproducibility to create useful models. Transparency in modeling moves modeling closer to enhanced understanding, confidence building, and scientific reproducibility of results. Models need to be as transparent as possible so that the formulation, the data, and all the assumptions about the real system implemented in the model, are salient, evident, documented, and readily available for inspection. The creation and use of analysis tools such as the SDM-Doc allows for enhanced transparency in models and for additional automated scrutiny in models.

ACKNOWLEDGMENTS

This manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under

Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. This work was supported in part by the City of Chicago through the Chicago Department of Public Health (CDPH).

REFERENCES

Arcade, J., M. Godet, F. Meunier, F., and F. Roubelat, "Structural analysis with the MICMAC method & Actors' strategy with Mactor method." Futures Research Methodology: American Council for the United Nations University: The Millennium Project (1994).

Eberlein, R. and W. Schoenberg (2020). "Finding the Loops that Matter." arXiv.

Forrester, J. W. (1958). "Industrial Dynamics: A Major Breakthrough for Decision Makers." Harvard Business Review 26(4): 37-66.

Forrester, J. W. (1961). Industrial Dynamics. Cambridge, MA, Productivity Press.

Forrester, J. W. (1968). Principles of Systems. Cambridge MA, Productivity Press.

Forrester, J. W. (1969). Urban Dynamics. Cambridge MA, Productivity Press.

Ghaffarzadegan, N., J. Lyneis and G. P. Richardson (2011). "How small system dynamics models can help the public policy process." System Dynamics Review 27(1): 22-44.

Kampmann, C. E. and R. Oliva (2006). "Loop eigenvalue elasticity analysis: three case studies." System Dynamics Review 22(2): 141-162.

Martinez-Moyano, I. J. (2012). "Documentation for model transparency." System Dynamics Review 28(2): 199–208.

Martinez-Moyano, I. J. (2018). A primer for system dynamics modeling and simulation. Proceedings of the 2018 Winter Simulation Conference. M. Rabe, A. A. Juan, N. Mustafee et al. Gothenburg, Sweden, IEEE Press: 261-275.

Martinez-Moyano, I. J. and G. P. Richardson (2013). "Best practices in system dynamics modeling." System Dynamics Review 29(2): 102–123.

Oliva, R. (2004). "Model structure analysis through graph theory: partition heuristics and feedback structure decomposition." System Dynamics Review 20(4): 313-336.

Oliva, R. (2016). "Structural dominance analysis of large and stochastic models." System Dynamics Review 32(1): 26–51.

Oliva, R. (2020). "On structural dominance analysis." System Dynamics Review 36(1): 8-28.

Repenning, N. P. (2003). "Selling system dynamics to (other) social scientists." System Dynamics Review 19(4): 303-327.

Richardson, G. P. and A. L. Pugh, III (1981). Introduction to System Dynamics Modeling with DYNAMO. Cambridge MA, Productivity Press.

Schoenberg, W., P. Davidsen and R. Eberlein (2020). "Understanding model behavior using the Loops that Matter method." System Dynamics Review 36(2): 158-190.

Schoenberg, W., J. Hayward and R. Eberlein (2023). "Improving Loops that Matter." System Dynamics Review.

Sterman, J. D. (2000). Business Dynamics: Systems Thinking and Modeling for a Complex World. Boston MA, Irwin McGraw-Hill.

AUTHOR BIOGRAPHIES

IGNACIO MARTINEZ-MOYANO is Computational Social Scientist in the Decision and Infrastructure Sciences Division at Argonne National Laboratory. Dr. Martinez-Moyano is also a Senior Scientist at Large at the Consortium for Advanced Science and Engineering of The University of Chicago and a Senior Fellow at the Northwestern-Argonne Institute for Science and Engineering of Northwestern University. Ignacio is Managing Editor of the System Dynamics Review and in 2018 was President of the System Dynamics Society. Ignacio's research work includes approaches to enhanced understanding at the intersection between data and models, dynamic threat analysis, health and disease modeling, security dynamics and operations improvement, judgment and decision analysis, and the effects of policy implementation. Ignacio's research work has been published in academic journals such as *Organization Science*, *Journal of Public Administration Research and Theory*, *ACM Transactions on Modeling and Computer Simulation (TOMACS),* and the *System Dynamics Review*. Dr. Martinez-Moyano holds a Ph.D. in Decision and Policy Sciences from the University at Albany, State University of New York, and an MBA, a M.Sc., and a B.Eng. from the Monterrey Institute of Technology. His email address is martinez-moyano@anl.gov.