Designing a Simulation Model Viewing Tool for Behavioral Scientists

Tylar Murray
University of South Florida
Department of Electrical Engineering
tylarmurray@mail.usf.edu

Daniel Rivera
Arizona State University
Control Systems Engineering Laboratory

Eric Hekler
Arizona State University
School of Nutrition and Health Promotion

D. Spruijt-Metz
University of Southern California
Center for Economic and Social Research
Los Angeles, California

Andrew Raij
University of Central Florida
Institute for Simulation and Training

Abstract—**In this paper we present relevant definitions and design considerations of special interest to those interested in human-behavior modeling software. The guidelines presented here are based off of a user-survey and expert-panel review performed in development of the BehaviorSim model-building tool designed especially for use in behavioral science. Lessons learned through iterations of this tool and unique considerations for those targeting behavioral scientists are highlighted. Our initial survey of 12 behavioral scientists reveals the diversity of opinions on and approaches to behavior modeling within the community. In order to address this, a theory-agnostic method for defining Human Behavior Models (HBMs) is proposed and techniques for supporting different modeling paradigms within a single user interface are discussed.**

I. INTRODUCTION

* what is an HBM? (in short)

* why are they important for the future behavioral research? [**?**]

* why do researchers need a tool for HBM development?

II. RELEVANT LIT

* other modeling softwares (and why it doesnt fit this use case)

* HBM examples /citeRivera

A. What is a Human Behavior Model?

In contrast to a cognitive model, a human behavior model attempts to model the human systems conversion of measurable contextual information into measurable behavior outcome. The variety of applications including something which may be considered a human behavior model is vast, but some of these applications better fit with the goal of increasing understanding of human behavior. For our purposes, we describe three classes of human behavior models based on the expected application domain 1) models developed to act as automated decision agents, 2) models developed to identify emergent behavior of a population, and 3) models developed as predictors of behavior. Using this classification scheme, it becomes clear that the methods underlying automated agents and emergent behavior simulations are less in line with the goal of behavioral theory development, and thus a focus on behavioral forecasting models is taken.

1) Automation Agents: Anyone familiar with agent-based modeling is likely familiar with the behavior-desire-intent (BDI) agent architecture which has found countless applications in the automation of industry tasks such as (EXAMPLES). Agents of this type attempt to reproduce specific decision-making tasks reliably and predictably so that the task may be automated. Though extremely useful in robotics and artificial intelligence systems, these models often do not seek to represent true human-like behavior, particularly the seemingly unpredictable and irrational components. The goal of automation agent research is to develop an agent which can be easily told what is to be done, can express an interpretation of the given task, and can automate the task at hand reliably and accurately.

TODO FIGURE: Human-Behavior Models designed to automate tasks such as controlling the flow of fluid between two tanks of liquid.

Agents designed to replicate decision-making tasks have also found use in military strategy planning and video game AI, and through these applications have begun to incorporate variation, randomness, and personality into the agents (http://www.agent-software.com.au/applications/realistic_virtual_actors_2/, cojack,). According the terminology as defined here, this type of agent is no longer classified as an automation agent – though they may be based on the BDI architecture, because the goal is no longer to simply automate a task.

2) Emergent Behavior Studies: Though the term agent-based modeling is often used to refer to models developed for the automation of a task or decision process (such as agents based on the BDI architecture), agent-based modeling techniques are also used to study the emergent behavior of a many-agent system. These emergent-behavior systems typically use a relatively simplistic agent definition and an environment with many agents to observe the behavior of the system as a whole. These agent models have found use in traffic simulation (Zhao

2013), MORE, and even molecular modeling (GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit, Identifiability and observability analysis for experimental design in nonlinear dynamical models, bionetgen, stochsim).

TODO: Figure: Emergent Behavior studies focus on interactions of many behavioral agents sharing one environment.

Human behavior models developed to explore emergent behavior often neglect the intricacies of each agents inner workings because they are designed for efficient computation and accuracy on the level of the system rather than the individual.

3) Behavioral Forecasting: Lastly, models of human behavior are created to allow for predictive modeling of a behavioral phenomenon. Models in this category fit the definition given by Glanz and Rimer (2005 p4): "a set of interrelated concepts, definitions and propositions that present a systematic view of events or situations by specifying relations among variables, in order to explain or predict the events or situations". Since this type of model is useful for predicting and understanding the state of a user, formulations of theory may help overcome one of the most critical hurdles of affective computing and allow for automated intervention personalization and contextualization. A forecasting model can allow a human-computer interface to alter its behavior to help the user achieve a desired state. These models are also useful for those designing a human-computer interface or a cyber-physical system in that they can be used to estimate the ways a human may act.

TODO: Figure: Human behavior models which forecast human behavior focus on the translation of environmental context and internal state to future actions.

These models can be personalized to a subset of the population, to an individual, or may be generalizable to any healthy individual. Many machine learning models of human behavior and cognitive theories both fall into this category, and this type of model is of special importance to the future of automated health management systems. Tailored behavioral interventions, wearable sensing devices, and mHealth applications all operate based on an underlying model of human behavior which has (thus far) remained implicit in nature. Though all three presented classes of human behavior model may contribute to the development of the next generation of models and theories, the primary focus of this paper henceforth is on terminologies as applied to this type of human behavior model (type 3: personal forecasting).

It is important to note that behavioral forecasting models have applications outside of personalized predictions. Behavior forecasting models can be applied in a simulation of a population of virtual humans and compared against real data to examine the fidelity of the model or to analyze the possible effects of certain contextual stimuli.

* UI for specialist systems

III.   USER STORIES

how does HBM model building fit into research process?

IV.   MODEL BUILDING

User Interface Design for HBM-builder The following flow chart represents the movement of a user through the process of building an HBM, or any graphical model for that matter.

[TODO: insert graphic from http://7ych.blogspot.com/2015/01/user-interface-design-for-hbm-builder.html]

It is important to note that the process of building is pseudo-linear; users will likely want to jump back to an arbitrary point in the process and change their mind. This is because the model is often not clearly defined before beginning. As the user moves through the process, they will re-assess the meaning of variables and connections and their model will solidify. UI for an HBM-building tool should enable easy backtracking and should work to provide feedback on the characteristics of the model early. The first iteration of the behaviorSim Model-Builder took a step-wise approach to this task, which made backtracking difficult and evaluators reported difficulty understanding how earlier choices related to later results.

[TODO picture of Model-Builder v0.1]

Successive generations of the Model Builder present a guided process which takes place within a single-page application. In this way, users can see how choices influence the model in real-time, and thus can iterate on their design much sooner.

While keeping the non-linearity of this process in mind, we will now walk through the diagram and address design considerations for each part of the process. Draw the Diagram

This task involves the creation and connecting of all nodes in the diagram. Nodes and their connections should be primarily based in theory, so a user should be able to build a first draft of their model with little feedback. Once the formulas for each node have been specified, however, it is likely that the user will want to return to this step and reassess the network. Several paradigms diagram-drawing have been well tested, but there are basically two options here: drag-and-drop node creation and drag-to-connect pros: easy and intuitive to use cons: can be difficult to position nodes to reduce edge overlap users often distracted trying to make graph "pretty" requires constant switching between mouse and keyboard2 Textual notation "language" for connecting nodes pros: can be much faster than mouse-based cons: parsing errors can be a source of confusion user must learn the markup "language" Use Graph-Wide Rule?

This choice can come before or after the initial drawing of the graph, since it is likely the user has a graph-wide assumption in mind when starting the process. This choice determines whether or not the interface for specifying nodes individually is shown, so it is logical to think of the "none", "node-specific", or "node-unique" option as one of the choices available here. Graph-wide rules which may implemented include: Probabilistic Graphical Network Linear Sum Fluid-Flow Analogy / Differential Equation Formulation 1st order 2nd order If a graph-wide rule is set, the formulation section for each node should show as locked to the rule's respective formulation. Editing an individual node's formulation will

disable the graph-wide rule, and should require confirmation from the user.

The interface for choosing one of these rules may be as simple as a select-option box or may include a detailed rule description and example.

An additional feature which may be useful is to allow for user-defined graph-wide rules. In this case, users might choose "create new rule" and an interface to allow users to edit and save graph-wide rules is needed.

For Each Node

Items encapsulated by the "for each node" swim-lanes must be executed for each node in the graph. The graph-wide-rule-yes case differs from the graph-wide-rule-no case only in that the graph-wide-rule-no case requires the additional step of setting the formulation at each node.

Choose a Formulation

This item is identical to the graph-wide rule choice, except that it applies only to a single node. Similar design considerations apply, and user-defined choice is even more important at this scale.

Define Model Constants

The general solution of an HBM does not require definition of the constants, but a simulation cannot be run until some numerical value is assumed. These constants often have theoretical significance in that they often have meaningful influence upon system behavior. Scaling-coefficients, for instance allow for relative weighting of each inflow. Similarly, the coefficients of a dynamical equation define how quickly variables react to a change "upstream".

UI for setting these constants could be as simple as a numerical input for each coefficient's name, but additional model details provided here may be of great benefit to the model designer.

Firstly, UI should include equation-specific descriptions of the constants whenever possible. This means that a separate UI for each formulation type is necessary.

Secondly, adjustment of coefficients would ideally show changes in the dynamics of a node's value in real-time. This can take the form of a time-series chart of the selected node's numerical value over time along with (editable) time-series for the inflows. Adjustment of these inflow series may also take a number of different forms.

Thirdly, model creators may want to set these coefficients in three different ways: Set the coefficient value to a constant. All simulations derived from this general solution will have this exact value. Set a probability distribution for the coefficient. Simulations derived from this general solution will select a numerical value based on the given distribution. This distribution may be learned from a training dataset, or bounds may be set for later training. Leave the variable unbounded and assume a flat probability distribution (this is essentially no2 with a flat distribution from -inf to +inf These different ways of defining coefficients in a HBM become very important when it is time to run simulations and compare model results to real data.

Lastly, model creators may want to use a measured value to set the value of these coefficients. In this way, one measured input (in real life), such as big-5 personality type, can be used to set the value of multiple coefficients across different variables in the model.

Designing UI for the model-constant-defining process is complex enough to warrant it's own article, which I will write later and link to here(TODO).

Nodes Left?

This is the most simple element of the flowchart. No if there are un-specified nodes remaining, yes if all are complete. It should be left to the user to confirm model-completeness, however, since a great deal of model-tweaking is likely now that simulation results can be displayed.

Next Node

Progression downstream through the model should be relatively straightforward in the case of tree-like models, but feedback loops can cause issues here. Perhaps the best paradigm here is to progress as naturally as possible, but allow the user to over-ride and select any node. This interaction mode is needed to allow for later model-tweaking anyway.

The behaviorSim Model Builder v0.2 currently does not include significant enough indicator of node "completeness", so that the user is sometimes unsure when they should feel free to move to the next node.

V. DESIGN

(overview and definitions)

* single-page design

* information-searching / focus and context design

* walk-through first use instead of tutorial (see, copy, do) /cite?

* use existing terminology (but also define it)

In the ideal modeling toolkit, trained machine learning models can be tested alongside dynamical systems formulations of behavioral theory. Though greater accuracy may often be achieved using a data-driven model, the theoretical formulation may provide additional insight into the system behavior, and connections to conceptual models can be drawn. Enabling easy system configuration allows researchers to leverage the best of all fields, creating a balance between predictive accuracy and model clarity depending on the users needs. The system configuration must have an intuitive high-level representation, and a highly-detailed interface for defining specifics. The user should be given the option to make many modeling assumptions to maximize ease of use, or to make few assumptions to ensure modeling specificity. Model re-use and extension must be supported by allowing concepts to operate in interchangeable modules, requiring carefully-defined standards for interface between modules. The ideal toolkit also supports model evaluation, comparison, optimization, and exploration to aid the user.

A "standard" approach to describing, designing, and visualizing human behavior models for personal forecasting is needed. There are countless approaches to systems modeling, but concepts and methods ought to be incorporated and fused when possible in their application to behavioral science. The resulting specification will allow for the formal description of a Human Behavior Model (HBM).

We will begin with the most basic definitions and then progress into the intricacies and usage examples of the proposed model-modeling paradigm.

A model is comprised of a network of variables and their interrelations. In order to visually represent the connectivity of the model a network graph can be used. Network graphs have many applications across multiple disciplines, and though the different implementations can be visually similar they can differ significantly in their meaning. In general, however, a graph is comprised of nodes and edges which represent the variables and their connections, respectfully.

We argue that the most intuitive representation is one which uses a directed graph wherein edge arrows to represent the flow of information between nodes. Thus, a HBM directed graph edge from node A to node B indicates that information flows from node A into node B.

$A =>$ B Graph 1: Can be read as "A influences B", "A informs B", or similar

This choice of notation is in agreement with graphs used in information theory, communications models, and behavioral science. In contrast, some graphing paradigms (such as probabilistic graphical models) prefer to use notation wherein an edge is used to represent dependency. In these paradigms edges may be read from tail to head as "depends on" or similar.

While the network graph does an excellent job showing the connectivity of a model, it fails to indicate the meaning of each connection. In the majority of existing applications, the mathematical form of the relationship is implied or else it is neglected completely. For instance, path diagrams from the behavioral sciences frequently denote dependence and do not specify functional form. Adding even further to the confusion is the notion that these graphs are often developed using different statistical analyses which may make different assumptions about the functional definition of inter-variate dependency. The most common analyses assess linear relationships between variables, and thus it is perhaps reasonable to assume that this is the intention of most authors. Assuming this is the case we can return to our simplistic example in Graph 1 and interpret the implied relationship as:

$B = coeff_a b * A + const$ Equation 1: coeff represents the correlation coefficient which relates A to B, and const represents a scalar constant

For nodes with multiple inflow edges, such as node B in the following graph:

$A => B <= C => D$ Graph 2: Example; now with more nodes!

The resulting formulation is simply a sum of the inflows:

$B = coeff_a b * A + coeff_c b * C + const$ Equation 2: Combining multiple inflows via sum is referred to as the superposition principle and is the defining characteristic of linear systems (not to be confused with this linear formulation)

Using this formulation, the general form of our HBM is expressed via the network graph alone (perhaps along with a statement about what edges mean). To express an ideographic implementation of this general model, we must also include a table of coefficient values. This representation is useful, but also very limited. One important feature which this formulation does not take into account is the dynamics of the relationship. This is very important for human behavior modeling because the variables in a HBM will often change their value over time. Taking time into account our formulation for Graph 1 becomes:

$B(t) = coeff_a b * A(t) + const$

Looking at this closely we can note that at each point in time the value of A depends only on the value of B at that same instant in time. This assumption is fine for many applications, but we argue that this is a very poor assumption for human behavior models. What if A is influenced by B but there is some lag before the effect manifests? What if A is influenced by B, but only if B is above some threshold value? What if A is influenced by the rate of change in B rather than the value of B itself? All of these scenarios cannot be expressed using this simple linear relationship.

To resolve some of these issues, a differential equations can be used to describe the relationship between variables as described by (CSEL paper which connects path diagrams and fluid-flow). Using the differential formulation our equation for Graph 1 becomes:

$B(t) = TODO..$

Just as before, our general model is not expressed entirely through the graph, and an ideographic example is specified by providing table of coefficient values. Our table is now quite a bit larger, but these coefficients have meaningful definitions which relate to our theory. While this formulation offers a huge improvement over the linear formulation, we can still imagine relationships which this formulation cannot express. Thus, a graph-wide assumption that each edge represents a differential equation may not be general enough for our HBM specification.

It should be noted at this point that although the linear formulation is too simple to express the dynamics of the differential formulation, the differential formulation is capable of expressing linear relationships. This is accomplished by setting coefficients of dynamical components to 0. One might think then that there is some general formula which could express any functional form and that this form should be used to express the relationships between variables in our HBM graphs. While such formulations do exist (such as Taylor or Fourier series approximations or even ANN-based relations), this usage tends to make the model difficult to understand and to simulate with. Indeed, linear and differential formulations are in such widespread use because of the relative ease with which we can understand and solve them. An additional problem raised by this approach is that of redundant formulations. Indeed formulations could even subvert the "arrow direction"

by drawing information contrary to our original choice of notation. The graph can then no longer be considered a directed graph, and the meaning of an edge becomes less clear. Additionally, the table of coefficients needed to express an ideographic case of the model quickly becomes prohibitively large, and the effect of each coefficient on the outcome is not intuitively meaningful.

Let us now consider the case where a graph-wide assumption is NOT made. That is, we will specify the functional form of each node individually so that each edge on the graph may be linear in form while another may be differential. This has the benefit of allowing for both complex relationships between variables as well as simplistic ones. In this way one could craft a model in which two variables are linearly related and a third is dependent on the variance of another variable (a particularly odd formulation, but one which is relevant to behavioral theory). Unfortunately, this approach also means that a table of formulations must now be included with our graph to show the meaning of each edge in the graph. Consider for example the table below for Graph 2:

node— formulation — type
==
B — $coeff_ab * A(t) + coeff_cb * C(t) + const$ — linear eq. D — TODO.... — differential eq. Table 1: an example formulation table for Graph 2

If a fixed number of functional forms is adhered to, the graph can be made to visually represent these functional forms through the use of different node icon shapes. This approach quickly begins to resemble applications which use flow-based programming. Indeed, they are quite similar in their approach, and the specification of a HBM is quite similar to the writing of a program.

Note also that a node can set any arbitrary formulation if needed. This is the least desirable situation, since the meaning of an inflow to each node even more convoluted. Though this usage reduces the ability of the graph to reduce processing load on the user through abstraction, there may be some cases where this type of formulation is desirable. Consider the following graph and formulation table:

A =¿ B ¡= C =¿ D =¿ E Graph 3: The final example graph.

node— formulation — type
==
B — $coeff_ab*A(t)+coeff_cb*sqrt(C(t))+const_b$ — custom D — TODO.... — diff. eq. E — $coeff_de * D(t) + const_de$ — linear eq. Table 1: an example formulation table for Graph 2

In conclusion, we propose that an HBM should be specified using the following rules:

1 use a graph-wide formula assumption if possible

2 when choosing a formulation, consistency between nodes is most important

3 when choosing a formulation, simplicity and clarity is second only to consistency

4 Thus a HBM specification must include:

An information flow graph

one of:

* a graph-wide formulaic assumption

* a formulation table

In order to express a particular solution of a general HBM, an ideographic HBM must also include a table of coefficient values.

B. UI considerations

Ease of use and intuitive user interface is a primary design consideration for existing softwares for modeling and simulation in systems theory. (reference and cite vensim, etc, from here) The ease of use of a system is one of the most important factors in the adoption of the system. (cite technology acceptance model?) The trans-disciplinary nature of a human-behavior modeling toolkit may require special consideration in order to be usable by those who may be unfamiliar with modeling and simulation concepts.

TODO: Figure: path2flow.png example of the rise in model complexity for the theory of planned behavior when mathematical assumptions are made explicit. (from: Ajzen 1985,1991,2002 (left), Nandola et al. 2013 (right))

VI. METHODOLOGY

* user survey

* expert panel review

VII. INFORMING USER STORY W/ SURVEY DATA

what do the users want? (ref user survey)

VIII. DESIGN GUIDELINE SUPPORT AND EXAMPLES FROM PAST ITERATIONS OF MODEL BUILDER

* multipage vs single-page (v0.1-¿v0.2)

* inquiries in panel-reviews

* single-page is overwhelming

* multiple terminologies from user surveys /citeontology customizable?

IX. CONCLUSION

The conclusion goes here.

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.