BehaviorSim Model Builder: Lessons Learned Designing for JiTAI Developers

Tylar Murray
University of South Florida
Tampa, USA
tylarmurray@mail.usf.edu

Eric Hekler
Arizona State University
Phoenix, USA

Donna Spruijt-Metz
University of Southern California
Los Angeles, USA

Daniel Rivera
Arizona State University
Phoenix, USA

Andrew Raij
University of Central Florida
Orlando, USA

ABSTRACT
In this paper we present design considerations relevant to the development of software for behavioral scientists developing Just-in-Time-Adaptive-Interventions (JiTAIs). Our findings are based off of a user-survey, expert-panel reviews, and UX interviews performed in development of the BehaviorSim model-building tool. Lessons learned through iterations of this tool, a JiTAI developer user persona, and guidelines for those targeting behavioral scientists are highlighted.

Author Keywords
Authors' choice; of terms; separated; by semicolons; commas, within terms only; this section is required.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous; See http://acm.org/about/class/1998/ for the full list of ACM classifiers. This section is required.

INTRODUCTION
With the increasing prevalence of wearable technologies and user-behavior collection, behavioral researchers are becoming increasingly overwhelmed with data. Researchers have seen the potential for optimizing a behavioral intervention to suit not only the user [?], but also the context [?, ?]. Just-in-Time Adaptive Interventions (JiTAIs) promise to provide the optimal nudge at the optimal time to aid users looking to change their behavior [?]. Imagine, for example, an anti-stress app which knows not to interrupt your work meetings, but also knows when to play your favorite song to help you relax on the drive home. Or consider a smoking cessation app that knows precisely when and where you are most likely to crave, and distracts you with a game before you realize you were about to want a cigarette.

The impending wave of context-aware[?], affective computing[?] applications seems to be the holy grail of behavior change, but researchers are finding that our models of human behavior are not capable of supporting this level of interaction. Traditional models of behavior do not offer the granularity and specificity needed for these applications[?]. Existing psychological models of human behavior act as a guide for behavioral scientists looking to predict behavior, but are rarely computational in nature - making their application to adaptive systems impossible. However, methods for creating computational models of human behavior are not well defined, but some early concepts have been published.

USER SURVEYS
A preliminary survey was given to a group of behavioral scientists in order to gauge the general perceptions and opinions on the development of behavioral models to support JiTAIs. In this survey we focused on a few key elements of the model building process to greatly simplify and shorten the modeling exercise. A context and behavioral outcome based on physical activity were given, and users efforts were focused on defining the inner workings of the human system within these constraints. We also asked to complete several survey items about the barriers keeping modeling and simulation out of behavioral science. Approximately 50 surveys were distributed following presentations on behavioral modeling and simulation at the 35th Annual Conference of the Society of Behavioral Medicine. Out of these 50, only 12 surveys were returned. In general, users still had trouble with modeling exercise. Most did not stray far from the given example, and others provided very different solutions which do not fit the "engineer's view" of the problem. In the survey questions participants reported that the mathematics and programming concepts required for developing simulatable models were overwhelming, and nearly all participants expressed a desire for increased collaboration between disciplines and a need for software tools to help them apply and validate these methods.

Using findings from the user study we developed a proof-of-concept software to aid behavioral researchers with the task of building a computational behavioral model. The software - called the behaviorSim Model-Builder (see figure 1) - took a step-wise approach towards the model-building process. First

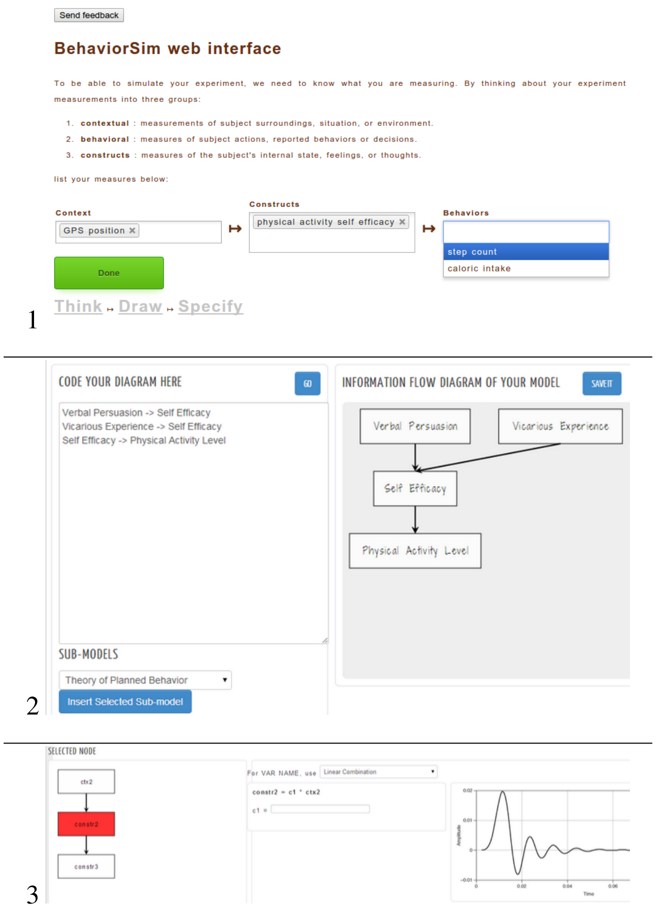

Figure 1. behaviorSim Model Builder v0.1.x implemented three separate stages of model building as separate pages: 1) think, 2) draw, 3) specify.

users are asked to list environmental inflows, internal state variables, and behavioral outflows of the model explicitly during the “think” stage. Next users are prompted to define the connections between nodes, “draw”ing the model’s structure. Lastly, users are required to “specify” the functional relationships at each node’s inflow(s).

The model builder was tested using a think-aloud protocol with 2 behavioral scientists and 1 HCI expert. Though the steps in the outlined model development process seemed appropriate, it quickly became obvious that a step-wise design is not optimal. Users who are forced to explore the process step-by-step have difficulty understanding how earlier choices related to later results, and feel constrained by previous choices rather than backtracking to revise the model. This design does not allow for quick iteration on models, and requires the user to maintain a great deal of planning information internally. Our review concluded that a less constrained approach to the stages of the modeling process was needed, and a greater focus on the graphical model could greatly improve user experience.

After reviewing v1 of the behaviorSim Model Builder tool, a tutorial was designed to help bridge the knowledge gap for new modelers looking to use the tool. In theory, the tutorial would help users see the bigger picture before diving into the

Figure 2. behaviorSim Model Builder tutorial attempted to merge time-series and information-flow graph representations.

stepwise process. The tutorial was implemented as a walk-through of a simple model’s internals. The tutorial also introduced a hybridized information-flow and time-series graph, wherein each node of the graph contains a time-series spanning a common time-frame, and a user interface for adjusting model parameters and updating time-series values instantaneously. This real-time parameter tweaking enables some degree of reconciliation with expectations of the data.

The same think-aloud protocol as used for the evaluation of v1 of the model builder was used to evaluate this tutorial.

Through this evaluation it became clear that even more explicit definition of terms was needed in order to clarify persistent disciplinary differences. Reviewers also wanted better explanation of models’ input parameters and of the functional definition of the system. They were not content with the hard-coded environmental inputs and wanted to be able to define how the contextual inflows changed over time. Though the hybrid graph was found helpful in conveying the connection between path diagram nodes and time series, the shared time-axis was not obvious, and reviewers expressed a need for more explicit x and y axes as well as a better explanation as to what “10 units of self efficacy” actually means. The time-series viewer was found to be both critical for the development of an accurate model, and valuable as pedagogical exercise for users trying to internalize model formulations.

In an attempt to apply what we had learned so far, the behaviorSim Model Building tool was re-designed and re-assessed. In this version all steps of the modeling process (figure **??**) are unified into a single-page application, allowing users to see how choices influence the model in real-time, and thus can iterate on their design more easily. The time-series charts popular in the tutorial were also added as a “mini-simulation”, to help users to visualize how variables change over time according to their model formulation. To address the terminology gap which plagued v1, a set of tool-tips were added which revealed detailed definitions for key terms used in the user interface. In this version of the tool, users declare constructs and define the structure of their model simultaneously by specifying on the connections using a simple Diagram Specification Language (DSL). In contrast to version 1, where the construct type had to be input by the user, the type of each node is inferred from the number of inflows and outflows.

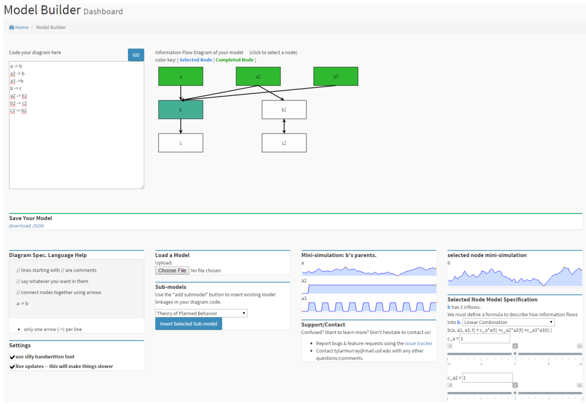

Figure 3. behaviorSim Model Builder v2 combines the think, draw, specify, and play elements into a single view.

The "miniature simulation" concept allows users to specify hypothetical contexts in which to explore the model dynamics, without the need to specify the full model. Users can specify environmental inflows for the simulation, choosing from adjustable presets (square wave, step function, random walk, or constant value). Selecting nodes on the graph by clicking, users can traverse the graph in any order. Time-series plots of inflows as well as the resulting outflow at each node are provided using the miniature simulation model. Internal state and behavioral nodes are specified similarly to environmental inflows through customization of function presets such as "linear combination" and "fluid flow analogy"[**?**].

This version was used as part of a structured exercise and interview outlining the design of a JiTAI to combat obesity. The content of this exercise was based on recent work done to define the JiTAI design use-case [**?**]. Preliminary findings from this exercise completed with 4 behavioral researchers highlight both the strengths and the remaining weaknesses of our design. A think-aloud protocol was used while the software was in use, and the concluding interview gives insight to what users find most valuable, least valuable, and most in need of improvement.

Though the single-page design of version two did seem to allow for increased ability to iteratively explore models, reviewers now found the user interface overwhelming at first. Furthermore the connection between the information flow graphic and the related UI elements below was not always obvious. The unification of construct specification interfaces in this design broke down the distinction between environmental inflows, state variables, and behavioral outputs. This lead to to some confusion when specifying the various types of nodes. Further contributing to this problem, the meaning of the mini-simulation was not always clear to reviewers, though the inclusion of time-series graphs was found helpful for understanding the model functions and their parameters once explained. Nodes on the graph were made to change color when the specification process was complete, but reviews revealed that this was not a significant enough indicator of node "completeness", so that the user is sometimes unsure when they should feel free to move to the next node.

RESULTS

Our findings above reveal specific flaws in our design, and through analysis of these findings we present the following design guidelines for software made to empower JiTAI developers. Firstly we outline a rough JiTAI developer user persona based on our assessment of the general population of researchers in behavioral intervention design. Next, we propose user stories and use-case details for the task of JiTAI design and evaluation. Lastly, we provide some generic design guidelines which we have found to be particularly relevant in this design space.

Who are JiTAI Developers?

In general, JiTAI developers are behavioral researchers who see the powerful potential of ubiquitous computing for high-frequency data collection, automated analysis, intervention deployment, and personalization. It is important to note that the research questions of a JiTAI researcher often differ significantly from the questions a behavioral scientist might typically have. The "traditional" way of modeling for behavior change relies primarily on statistical data analysis techniques to find relationships between variables on large time-scales. In contrast, the JiTAI researcher needs to translate these relationships into a small-time-scale model which provides guidance regarding which interventions are most effective at which specific time(s). Research questions focusing on the detailed dynamics of behavioral change are becoming increasingly prevalent as mobile behavioral health research continues to grow.

The JiTAI developer wants to turn a patient story into a set of equations that can be handled by an automated system. However, JiTAI developers typically do not have the level of familiarity with modeling systems to define abstract psychological models mathematically. Furthermore, the psychological models commonly used are ill-specified at the timescales of greatest interest, and often do not fit commonly used modeling paradigms - making mathematical definition a unique challenge for even a systems engineer.

The JiTAI developer wants to deploy and test a hypothesis by comparing model predictions to experimental data. Statistical analysis techniques typically used to assess control vs experimental group differences are much less applicable to this problem, but the JiTAI developer often has little experience applying goodness-of-fit metrics.

Designing for JiTAI Developers

Promote Expertise Development

The science of JiTAIs is young, and - as our user persona shows - behavioral scientists looking to work with JiTAIs are likely to run into many new concepts. The potential complexity of a JiTAI system, however, may benefit from the use of advanced and specialized graphs, charts, and user interfaces. Thus, the needs of a novice user versus an expert user may be very different. Because of this, software to support JiTAI development needs to promote the development of expertise in both the system and the relevant concepts through steady changes to the user interface [**?**]. In our case, a guided walkthrough of the software interface was sufficient, but we believe that a more graded approach to the complexity would be more effective.

Enable Quick Iterations

The value of iterating on a design spans many domains, and is very applicable to the development of JiTAIs. Through our studies we have found that the development of even a simple JiTAI requires many iterations. Thus, a software to aid in JiTAI development must allow for quick and easy modification, comparison, and reversion. A comparison between the usage of our multi-staged model builder versus the single-page application showed a dramatic increase in the number of model iterations along with reported user comprehension. Iterations on the model tended to follow a moment of realization or the learning of a new concept. Thus, allowing for quick iterations allows for the user to more quickly apply newly gained expertise, yielding a better model and increased understanding.

To encourage iteration in the JiTAI development process, assessment tools available part-way through the process - like our mini-simulation time-series - allow users to test their mental model of the system against its digital representation. Allowing for more assessment points throughout JiTAI development allows users to identify problems early so they may iterate before the error cascades further through the process.

High-Level Visuals to De-internalize Models

Traditionally, psychological models of human behavior are meant to be guidelines for thinking about human behavior. When using these models, the researcher must internalize the model and think through the subject state. With JiTAI models, internalization of the full system becomes impossible due to the rise in specificity and complexity. Thus, JiTAI development software must provide visualizations of the system to ease cognitive load on the user. Focus plus context displays [**?**] can should be used to allow users to delve into the specifics of a portion of the model without losing the larger context. In the behaviorSim Model Builder, we focus on the specification of a single variable at a time, and highlight this variables context in an information-flow path diagram of the model structure. This dual-viewing-area approach works well for comparing variable details, but the use of a zoomable interface such as is employed in some flow-based programming [**?**] tools may be more intuitive.

Adaptive Interface

Though our JiTAI developer persona yields widely applicable general user stories, it is also important to recognize the diversity of the JiTAI developer user group. JiTAIs are applicable to any area of behavior change; just a few popular proposed JiTAI applications are: management of eating behaviors, physical activity, smoking cessation, drug abuse, PTSD, and stress. Within each of these many application domains, are a myriad of behavioral theories - further adding to the diversity of the user group. Each of these sub-user-groups may have slightly different needs as they develop a JiTAI. Furthermore, a JiTAI development software requires a standardized behavioral model or JiTAI format, and with that comes the opportunity to enable easy sharing and searching of JiTAI designs. Thus, personalization of the software interface - to adapt the process or to offer relevant information[**?**] - can greatly improve user experience in this domain.

CONCLUSION

TODO!

ACKNOWLEDGMENTS

TODO!

REFERENCES FORMAT

Your references should be published materials accessible to the public. Internal technical reports may be cited only if they are easily accessible (i.e., you provide the address for obtaining the report within your citation) and may be obtained by any reader for a nominal fee. Proprietary information may not be cited. Private communications should be acknowledged in the main text, not referenced (e.g., “[Golovchinsky, personal communication]”).

Use a numbered list of references at the end of the article, ordered alphabetically by first author, and referenced by numbers in brackets [2, 7]. For papers from conference proceedings, include the title of the paper and an abbreviated name of the conference (e.g., for Interact 2003 proceedings, use Proc. Interact 2003). Do not include the location of the conference or the exact date; do include the page numbers if available. See the examples of citations at the end of this document and in the accompanying BibTeX document.

References *must be the same font size as other body text*. References should be in alphabetical order by last name of first author. Example reference formatting for individual journal articles [2], articles in conference proceedings [7], books [9], theses [10], book chapters [11], a journal issue [6], websites [1, 3], tweets [4], patents [5], and online videos [8] is given here. This formatting is a slightly abbreviated version of the format automatically generated by the ACM Digital Library (**http://dl.acm.org**) as “ACM Ref”. More details of reference formatting are available at: **http://www.acm.org/publications/submissions/latex_style**.

REFERENCES

1. ACM. 1998. How to Classify Works Using ACM’s Computing Classification System. (1998). **http://www.acm.org/class/how_to_use.html**.
2. Ronald E. Anderson. 1992. Social Impacts of Computing: Codes of Professional Ethics. *Social Science Computer Review December* 10, 4 (1992), 453–469. DOI: **http://dx.doi.org/10.1177/089443939201000402**
3. Anna Cavender, Shari Trewin, and Vicki Hanson. 2014. Accessible Writing Guide. (2014). **http://www.sigaccess.org/welcome-to-sigaccess/resources/accessible-writing-guide/**.
4. @_CHINOSAUR. 2015. VENUE IS TOO COLD. #BINGO #CHI2016. Tweet. (1 May 2015). Retrieved Febuary 2, 2014 from **https://twitter.com/_CHINOSAUR/status/461864317415989248**.

5. Morton L. Heilig. 1962. Sensorama Simulator. U.S. Patent 3,050,870. (28 August 1962). Filed Februrary 22, 1962.
6. Jofish Kaye and Paul Dourish. 2014. Special issue on science fiction and ubiquitous computing. *Personal and Ubiquitous Computing* 18, 4 (2014), 765–766. DOI: http://dx.doi.org/10.1007/s00779-014-0773-4
7. Scott R. Klemmer, Michael Thomsen, Ethan Phelps-Goodman, Robert Lee, and James A. Landay. 2002. Where Do Web Sites Come from?: Capturing and Interacting with Design History. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 1–8. DOI: http://dx.doi.org/10.1145/503376.503378
8. Psy. 2012. Gangnam Style. Video. (15 July 2012). Retrieved August 22, 2014 from https://www.youtube.com/watch?v=9bZkp7q19f0.
9. Marilyn Schwartz. 1995. *Guidelines for Bias-Free Writing*. ERIC, Bloomington, IN, USA.
10. Ivan E. Sutherland. 1963. *Sketchpad, a Man-Machine Graphical Communication System*. Ph.D. Dissertation. Massachusetts Institute of Technology, Cambridge, MA.
11. Langdon Winner. 1999. *The Social Shaping of Technology* (2nd ed.). Open University Press, UK, Chapter Do artifacts have politics?, 28–40.