Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

PROTOTYPING AN ANALOG COMPUTING REPRESENTATION OF PREDATOR PREY DYNAMICS

Karen Doore
Paul Fishwick

Creative Automata Laboratory
Arts and Technology
University of Texas at Dallas
800 West Campbell Road, AT10, USA

ABSTRACT

Analyzing systems can be a complex task especially when there is feedback across several variables in the model. Formal mathematical notation makes it difficult to understand the influences of feedback and cause/effect. Forrester created the System Dynamics methodology as a means to assist in this understanding by employing a hydraulic analogy. In this methodology, variables become simulated objects such as water valves or tanks. A variety of implementations allow users to construct and simulate these models. The problem is that for many implementations, the intuitive nature of water flow, intended by the methodology, is not as clear as it could be. For instance, the rate of flow or level in a tank may not be visualized. For novices, we suggest that this issue, as well the ability to understand relationships and linking across multiple representations can be problematic. We designed and describe a web-based interface that solves these problems.

1 INTRODUCTION

System Dynamics is a modeling framework developed by Jay Forrester in the mid 1950's to provide enhanced understanding for decision-makers of complex dynamic systems across a wide variety of domains. Forrester strongly advocated teaching System Dynamics (SD) as part of the K-12 education. He suggests that the process of learning to develop qualitative models of complex systems, coupled with using computer simulation software to explore the quantitative behavior of the system over time would equip students to better understand the nature of many real world systems. Forrester was a pioneer in promoting the use of computer simulation for teaching interdisciplinary 'system thinking' in K-12 curriculum. His recommended approach for introducing system dynamics within the K-12 curriculum relied on having students deconstruct problems based on case studies or narrative descriptions to identify problem features that could be mapped to either causal loop diagrams or stock-flow diagrams. Then students would use computer software to design models and simulations to generate 'behavior over time' (BoT) graphs to verify the structure of their designed models. However, extensive research has shown that even highly educated novices have difficulties understanding some fundamental concepts of system dynamics modeling (Sterman 1994; Sterman 2000; Sweeny et al. 2000). Whereas most SD learning environments provide extensive support for students to design simulations using visual language components, our application is designed to give students a deeper understanding of the processes represented by these icon based visual languages and their relation to traditional mathematical notations and behavior graphs. Figure 1 is a screenshot of our interactive learning environment. It consists of four main panels that provide different representations of the Lotka-Volterra (LV) system.

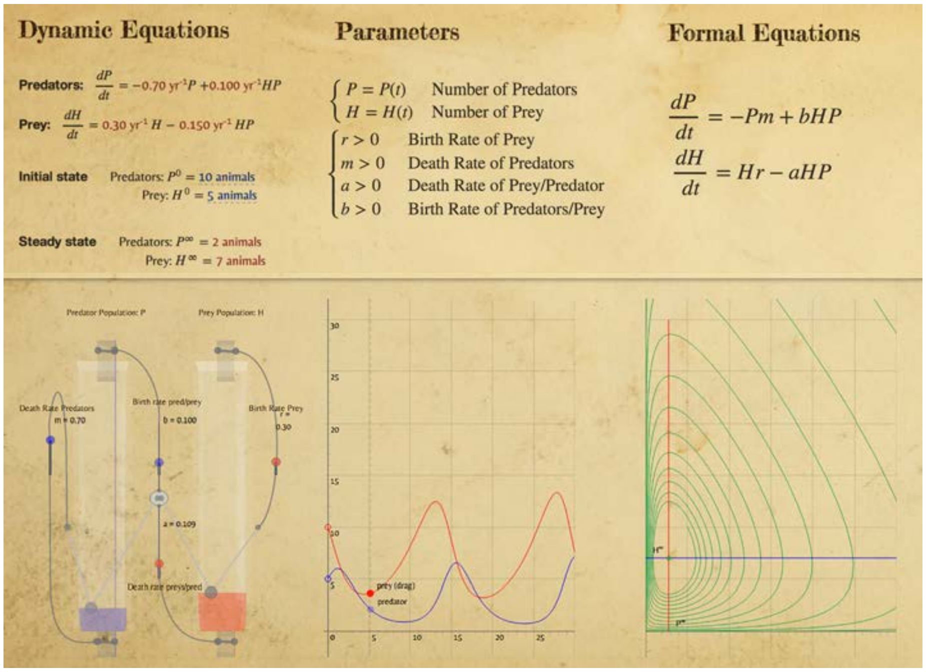


Figure 1: Interactive learning environment with dynamically linked representations. The upper section features dynamic mathematical equations, while the lower modules feature an analog water machine on the left, a behavior over time graph in the center, and a phase diagram on the right.

This JavaScript-based web application features dynamic linked interactivity between all of the different modular representations. We posit that in order for novices to understand the relationship between system dynamics behavior and system structure, *they must first develop a deep understanding of the concepts behind the water analogy, which inspires the stock and flow diagram notation.* Stock and flow diagram notation provides a powerful grammar for representing the structure of dynamic systems, however the sparse notation also corresponds to information dense icons, which can be problematic for novices learning non-intuitive systems. Therefore, we have designed our analog water machine so that novices can develop a more concrete understanding of levels, flows, valves, feedback loops, decision points, and the dynamic nature of their relationship with the system behavior graphs.

2 BACKGROUND

2.1 System Dynamics in Education

Forrester advocated that a System Dynamics Education would provide students with a foundation for developing critical thinking skills for the 21st century (Forrester 1994). Forrester proposed that system dynamics modeling should be an integral component of a K-12 curriculum because it can provide students a ‘more effective way of interpreting the world around them’ (Forrester 1993). Forrester emphasized that system dynamics modeling should be ‘learning by doing’; he suggests that ‘immersion in such active learning can change mental models’ (Forrester 2009). He conjectured that if students could learn to model the structure of complex dynamic systems, and then study the behavior of their model using computer simulations, then those students would be able to develop mental models, which reflected this enhanced understanding of complex system behavior. Forrester recommended that students develop experience in modeling a diverse range of systems, in order that students would recognize the universal nature of complexity and of the principles of dynamic systems (Forrester 2009). He suggests that the process of creating simulation models requires students to develop precision thinking skills, which are necessary in order to translate from a descriptive expression into an explicit system modeling language.

2.2 System Dynamics Modeling

In his initial book on system dynamics, *Industrial Dynamics*, Forrester (1961) identified several principles for 'effective modeling of complex systems: counter-intuitive system behavior is driven by system structure, structure involves non-linear relationships, computer simulation is necessary to explore behavior' (Lane and Sterman 2011). The basic concepts of system dynamics modeling are: stocks, flows, feedback, time delays, and nonlinearities (Fishwick 1995). Researchers in the field of system dynamics have proposed that learning system dynamics concepts corresponds to developing an internal 'mental model of a dynamic system' (MMDS). Doyle and Ford (1999) define MMDS as "a relatively enduring and accessible, but limited, internal conceptual representation of an external dynamic system (historical, existing, or projected). The internal representation is analogous to the external system and contains, on a conceptual level, reinforcing and balancing feedback loops that consist of causally linked stocks, flows, and intermediary variables. The causal links are either positive or negative, are either linear or non-linear, and can be delayed" (Groesser and Schaffernicht 2012). This type of mental model is quite different from how novices would instinctively interpret complex system behavior. However, before a student can develop an internal mental model of system dynamics, they must be able to understand the meaning of representations of external models of this system.

2.3 Lotka-Volterra Dynamic System

The Lotka-Volterra (LV) equations describe one of the most basic complex dynamic systems that students can relate to, particularly when used to model a simple predator-prey system. LV equations represent an archetypal system, which have been used to model a diverse range of systems from ecological systems to economic theories. From a formal mathematics perspective, the Lotka-Volterra equations are a nonlinear-coupled system of ordinary differential equations. Typical of most SD systems, there is no closed form solution, so numerical methods are used to analyze the behavior, which oscillates over time. Figure 2 shows the formal mathematical notation for the predator prey system, which our system represents.

For a typical inquiry-based project, students are given some narrative description of a real-world or hypothetical situation, which provides the details of a predator and prey system. Then, students are tasked with identifying the relevant features of the system in order to develop an understanding of the system behavior and structure. Figure 2 shows some of the features of the LV system, which include the populations for predator and prey, births that increase populations, and deaths that decrease populations. In addition, problem descriptions also include some form of data that shows how the population of both species have varied over a given time period. Students are then guided to develop a causal diagram, such as in Figure 2, which uses arrow arcs to indicate cause-effect relationship between the identified system features. Used in this manner, causal loop diagrams (CLD) provide a tool that can support group discussion about the nature of the predator–prey system.

2.4 Causal Loop Diagrams

Causal loop diagrams use '+' and '-' notations to indicate the directionality that exists in causal relationships. In Figure 2, a feedback loop is denoted between prey and prey births, where the arrow polarity shows that as the population of prey increases, it has a positive influence on the number of births of new prey, which in turn increases the number of prey. This circular relationship is indicated as 'positive' or 'reinforcing' by the small arrowed-arc in the center of the prey population and prey births cycle. In contrast, a 'balancing' or 'negative' feedback loop occurs between predator population and predator deaths. It can be seen as a balancing loop because as the number of predators decreases, the number of deaths also decreases; therefore, the balancing loops indicate a form of stabilization within a system. The dynamic interplay between positive and negative feedback loops in a system cause oscillation behavior (Fishwick 1995).

The LV system is a 'coupled' system, meaning that there's interdependency between the species. In Figure 2 we have introduced a predation density variable to indicate that predators eat prey and we have drawn causal arcs that show how this interdependency impacts both populations. The arcs indicate that an increase in prey population provides more food for the predator population; this results in predators eating more prey, which results in more prey deaths but also more predator births. It is this inter-species coupled interaction that creates a complex system with non-intuitive, oscillating behavior. The system exhibits emergent behavior because analysis of either predator or prey populations independently would not provide insight into the dynamic behavior of the species populations. The CLD model design process is a critical phase in system dynamics curriculum, as it provides a notation that supports qualitative mapping of causal relationships, and denoting the identified relevant features of the closed system.

Lotka-Volterra Equations

$$\frac{dP}{dt} = -Pm + bHP$$

$$\frac{dH}{dt} = Hr - aHP$$

$$\begin{cases} P = P(t) & \text{Number of Predators} \\ H = H(t) & \text{Number of Prey} \end{cases}$$

$$\begin{cases} r > 0 & \text{Birth Rate of Prey} \\ m > 0 & \text{Death Rate of Predators} \\ a > 0 & \text{Death Rate of Prey/Predator} \\ b > 0 & \text{Birth Rate of Predators/Prey} \end{cases}$$

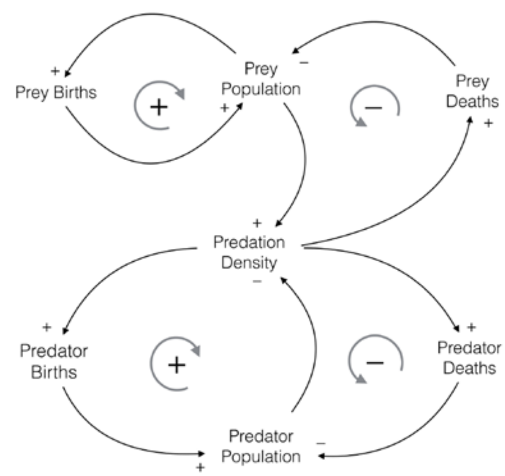


Figure 2: Lotka-Volterra Formal Equations and Corresponding Causal Loop Diagram of Predator Prey

2.5 Stock and Flow Diagrams

The traditional notation for system dynamics modeling is stock and flow (SF) diagrams which were originally called system flow structure diagrams and were designed to support decision making in business management contexts. This diagram notation was designed to provide a common language for interdisciplinary group discussion between business managers and scientists who were working together to develop models to guide decision making in business systems (Lukaszewicz 1976). The notation used for SF diagrams has continued to evolve and varies between authors and between different software systems. Lukaszewicz's notation allowed for a direct 1-1 mapping between mathematical functions and diagram notations to insure all relevant features of a system were completely specified between the diagram and related mathematical formulas, however this resulted in a highly complex visual diagram.

2.6 Simplified Diagram Notation With an Emphasis on Decision Point Structure

Morecroft (1982) discussed the evolution of SD modeling notational use, and noted that the increased use of causal loop diagrams as a conceptual tool could be problematic for several reasons, but he conjectured that CLD diagrams were popular due to their simplicity. Morecroft proposed a new 'policy structure' format which provides a higher level of abstraction than system flow structure diagrams but also provides notations to capture essential stock, flow, and feedback loop structures which can not be properly encoded in a CLD diagram. Morecroft's diagram notation is similar to what is currently used for SF diagrams.

Morecroft's diagram notation was designed to support conceptualization and understanding of the decision-point structure in dynamic systems with a focus on determining the decision-making information which are referred to as policy rules in business domains. Our application has been designed to highlight Morecroft's policy decision-point structure, *by explicitly representing the adjustable variables* in the system and providing users with dynamic interaction to modify these values and to observe the dynamic behavior of the system in response to manipulation of these parameters.

2.7 Interactive Stock and Flow for the Lotka-Volterra System

Our application, includes an interactive module that is a visual analog of a stock and flow model. Figure 3a shows a stock and flow diagram of the LV system and Figure 3b shows a schematic diagram of our interactive water machine. Although these two models are isomorphic, our model has been designed to provide an amplified perspective of the water flow analogy so that novices can build on their intuitive understanding of water flow to extend their existing mental model of water flow to include these new system dynamic concepts.

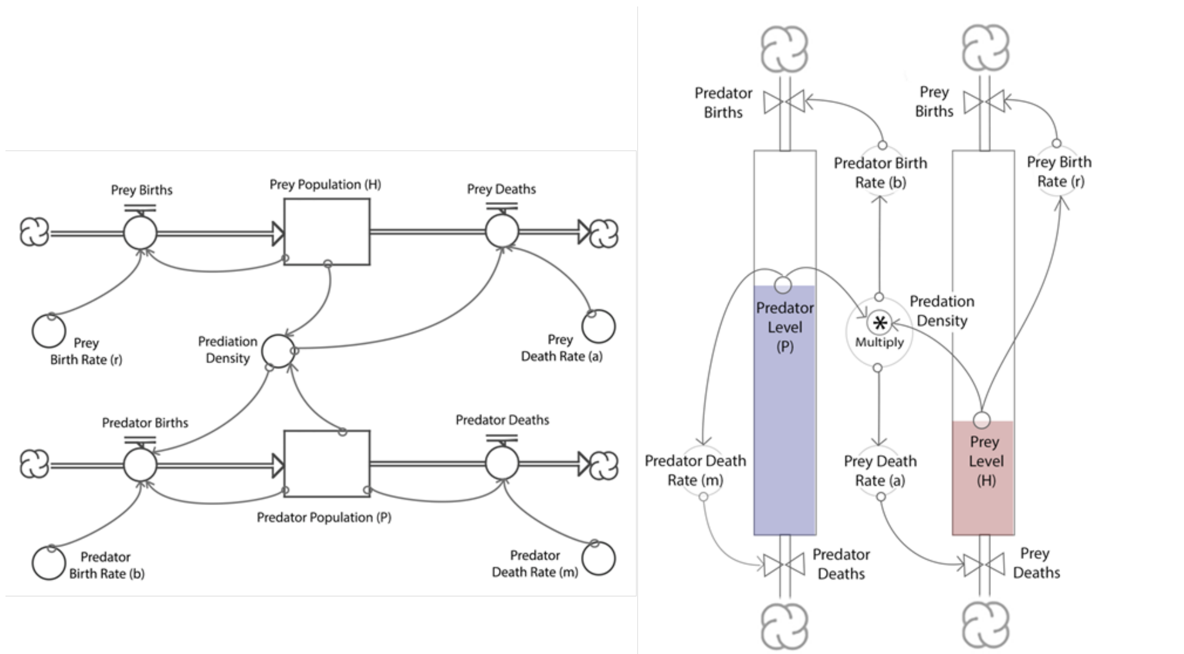


Figure 3a: (left) shows a typical stock and flow diagram of a predator-prey system. Figure 3b: (right) is a schematic design of our interactive virtual water machine. The diagrams are isomorphic, however Figure 3b provides an intuitive orientation when using a water flow analogy.

In order to create a SF diagram based on the predator-prey CLD, the first step is to identify features of the system that represent stocks. Meadows (2008) notes that stocks are elements of a system 'that you can see, feel, count and measure at any given time' and that stocks encode the 'memory of changing flows within a system.' The predator and prey populations represent stock values in these SF models. These SF diagrams use circle notation to indicate auxiliary variables like birth rate; these features of a system can be adjusted to manipulate the flows in a system. In contrast, stock variables like population levels cannot be directly manipulated in a closed system, these accumulation variables only change as a result of changes in connected flows. An important feature of LV systems are the feedback loops, which can be seen as arcs in the SF diagram that stretch from the level of the stock and connect back to a valve that controls flow into or out of that same stock. We have designed our system to scaffold users understanding of how modification of auxiliary variables combined with the feedback loop structure dynamically impact the system. Essentially we are opening the black box that is represented by valves in SF diagrams, and providing users dynamic interaction with auxiliary variables to provide insight into the relationship between these adjustable parameters and the behavior of the system. We are expanding on the water metaphor with a focus on dynamic behavior in the valves, because this is a critical for novices when learning system dynamics, and other systems do not provide novices an interactive window into these dynamics. In section 4 we elaborate on these design features.

2.8 Difficulties in Understanding Accumulation and Feedback Thinking

Extensive research has shown that novices have difficulty understanding fundamental concepts of complex dynamic systems. One important area of difficulty is referred to as 'understanding of accumulation' (UoA). Research has demonstrated that even highly educated individuals have difficulty in understanding how changes in flow rate of a material impact the accumulation of that material in a container over time (Sterman 1994; Sterman 2000; Sweeny et al. 2000). For novices, intuition about stocks and flows often follows a correlation heuristic, which implies that they assume the level of the stock container should 'look like' the input or flow into the container (Cronin 2009). Additionally, another area of difficulty for individuals trying to develop an understanding of dynamic systems is a concept referred to as 'feedback-thinking' (Schaffernicht et al. 2012). Incorrect intuitions about feedback-loops in a system correspond to the failure of a person to recognize circular cause-and-effect relations and time delays (Schaffernicht and Groesser 2012). In essence, the intuitive way of perceiving situations is to assume that a linear cause and effect relationship drives the behavior of a system.

2.9 Interventions and Intuitions

Some studies have analyzed the impact of educational interventions on students' understanding of difficult SD concepts, such as students completing a university level course on system dynamics. University students in these studies improved their performance and the researchers suggest that learning these concepts essentially requires a conceptual change in the student's mental model of a dynamic system (Sterman 2009; Schaffernicht et al. 2012; Kopainsky and Saldarriaga 2012). A number of researchers have investigated the nature of children's understanding of accumulation using a wide variety of assessment formats and these researchers argue that the task format of previous studies may be responsible for the results which indicated difficulty in understanding accumulation. These researchers suggest that there is benefit in building on users intuitive understanding of accumulation based on DiSessa's 'knowledge in pieces' conceptual framework, which suggests building on users existing intuition can provide enhanced understanding (Saldarriaga 2011, DiSessa 2001). As Sterman (2002) notes, 'it is clear that people have poor understanding of these concepts, we know that people can learn to think in feedback terms, to recognize and understand stocks and flows, time delays, and nonlinearities.'

3 INTERACTIVE SYSTEM DYNAMICS REPRESENTATIONS

3.1 Interactive Representations to Enhance Learning

While research has shown that even well educated novices have difficulty in understanding fundamental concepts in system dynamics modeling, some research has also shown that learners can improve their mental model of dynamic systems thought targeted interventions. Overall, there has been limited research on the development of interactive learning environments to help novices develop improved intuitions about the fundamental features of complex dynamic systems. DiSessa's (2001) research suggests that the use of computational media can have a dramatic impact on learning methodologies. Golman and colleagues (2014) conducted research to compare several commonly used modeling approaches for conceptualizing system dynamics models, they suggest that the SD community 'should begin to take more seriously and address more actively the strengths and weaknesses of alternative approaches to helping participants surface and articulate their understanding relevant to a problem situation'. They explored the representational ease and expressiveness of traditional SD modeling approaches based on the notion suggested by Larkin and Simon that 'the ease, and accuracy with which a representation can be processed by the human mind determines the cognitive effectiveness of the representation.' However, their research was primarily focused on group problem solving where users designed different types of causal models, and they did not research use of stock and flow diagrams (Golman et al. 2014).

3.2 Cognitive Theories of Multimedia Learning

Mayer (2005) developed the Cognitive Theory of Multimedia Learning (CTML) in order to provide guidelines for designers of multimedia learning environments, with a focus on increasing learning transfer, which he defines as the ability to use learned information to solve new problems. Mayer's theory is based on three cognitive science principles of learning, which imply that 'that the human mind is a dual-channel, limited-capacity active-processing system' (Mayer 2005). CTML suggests that excess visual content can increase diagrammatic complexity, which can make it difficult for novices to determine the most important features to actively focus on. One function of a well-designed interface for multiple representations is that it can allow learners to construct a deeper understanding of abstract concepts if learners can identify relationships between different representations. In order to support learners' discovery of relationships between representations, we have provided dynamic linking of features across representational modules.

3.3 The Educational Value of Multiple External Representations

Ainsworth (2008) developed a framework to describe the function of multiple external representations (MERs) when used in educational contexts to teach complex scientific concepts. This framework 'integrates research on learning, the cognitive science of representation and constructivist theories of education'. Ainsworth proposes that the effectiveness of MERs can 'best be understood by considering three fundamental aspects of learning: the design parameters that are unique to learning with multiple representations; the functions that multiple representations serve in supporting learning; and the cognitive tasks that must be undertaken by a learner interacting with multiple representations'. Ainsworth notes that 'there is abundant evidence showing the advantages that [MERs] play in supporting learning' (Ainsworth 2006). Ainsworth's framework provides a guideline for discussion of the design decisions that we made in designing our interactive model; in particular, our design decisions support learners by presenting complimentary views, demonstrating constraints, and supporting deeper understanding. In this paper, 'deeper understanding' will be considered in terms of using MERs to promote abstraction, to encourage generalization and to teach the relation between representations (Ainsworth 1999). Fishwick (2004) defines a 'multimodel' as a model that contains multiple coupled, hybrid, heterogeneous models, which are developed to stress the human interface to models. Specifically, our research focus is on the design MERs which can be considered multimodels for learning contexts. Ainsworth indicates that MERs can function to constrain a user's interpretation, and this can provide enhanced understanding of relationships in a system. Ainsworth suggests 'the primary purpose of the constraining representation is not to provide new information but to support a learner's reasoning about the less familiar one' (Ainsworth 1999). Kaput (1989) proposes that 'the cognitive linking of representations creates a whole that is more than the sum of its parts. It enables us to 'see' complex ideas in a new way and apply them more effectively'.

4 PROTOTYPE DESIGN AND DEVELOPMENT

We constructed our Lotka-Volterra application using web-based technologies in order to provide high accessibility to our application. As suggested by Wagner (2012), one important approach to increase the use of simulation in education, is through the creation of freely available HTML5 based simulation resources. Wagner suggests that the only types of simulation that are 'currently acceptable for use in education' are HTML5/JavaScript based, and he further states that simulations based on java or flash are becoming 'obsolete in the future' (Wagner 2012). We used several open-source client-side JavaScript libraries for the framework of our interface as this allowed for the rapid development of this prototype version. We used JSXGraph (2014), which is a library for interactive geometry and graphs, for the behavior over time graph, the phase plot, and the analog water machine. We used an example JSXGraph of Lotka-Volterra equations (2014) as the starting point for this prototype. We used MathJax.js (2014) and Tangle.js (2014) libraries to enable dynamic linking between the LV formal equations and the behavior in the other modules. We used HTML5 and the Bootstrap (2014) library to create a responsive layout to insure our application works on mobile devices and tablets. The vast number of open-source JavaScript libraries creates an attractive option for rapid development of integrative multimodeling simulation prototypes as envisioned by Fishwick (2004), who suggested 'that future models will be less expensive to reproduce, be more engaging, and be customized for a specific task or person.'

4.1 Interface Design Decisions to Enhance Understanding

Moody (2009) provides extensive guidelines for designing interface notations to improve user understanding. We have utilized these recommended practices in designing our application interface. Our interface design decisions support novices in learning system dynamics concepts by building on their intuitive understanding of water flow and accumulation. Whereas many popular SD simulation packages provide support for users to design simulations using visual icons, our environment is designed to give students a deeper understanding of *the processes represented by these visual icons.*

4.2 Dynamic Levels to Enhance Understanding of Accumulation

The interaction design features of our analog water machine will allow novices to further develop their intuitive understanding that accumulation in a container is the result of net flows into and out of that container over a period of time, and that this understanding of accumulation is analogous to the LV system population levels and flow rates. In Figures 3b and Figure 4, a buoyant float indicates the current level of a fluid in the containers, and the flow into and out of the containers is shown with varying width of the flow stream. When a user slides the current prey indicator node, shown in Figure 5, along the behavior over time plot, the levels and flows in the containers adjust in real time to reflect the changing levels in the BoT plot. In addition to providing dynamic linking in order to support novices in understanding that the stock-level in the water representation is directly analogous to the behavior over time plot, the following design features further strengthen the encoding of this relationship :

- Both modules are identically scaled along the vertical axis
- The vertical aspect ratio of cylinders indicates height is the dimension of interest.
- Cylinders are aligned along the BoT and Phase Plot graph baselines.
- The modules are adjacent in layout (gestalt proximity).
- The color of the level's liquid matches the color of the corresponding BoT graph.
- The phase plot is also dynamically linked to both the water machine and the BoT graph.

Users can manipulate the initial values for the population levels using either the sliders on the BoT graph or they can drag numerical text values in the formal math notation in the top module of the application shown in Figure 1. Manipulation of the initial population values can provide insight into the concept that levels encode a memory state for the system. The use of gestalt principles, such as proximity and alignment to enhance perceptual understanding of relationship between MER modules, has been a longstanding practice in graphic design and visualization communities.

4.3 Dynamic Valve Control to Enhance Understanding of Feedback

Our virtual water machine was also designed to highlight the dynamic feedback nature of SD systems. The system has been designed to amplify the notion that the population level acts in conjunction with auxiliary parameter sliders to determine the current flows through visible dynamic control flow valves.

Figure 5a shows a zoomed view of these parameter sliders and auxiliary variable nodes. Figure 5b shows a comparison of our control valve dynamic features as compared with several other SD notation valves, including Morecroft's simplified SF notation. As mentioned in section 2.6, we have designed our system to reflect Morecroft's emphasis on understanding of how policy or rule changes dynamically impact the system behavior, we believe this is critical to a user's understanding of how feedback-loop structure impacts the behavior of these systems. In the Lotka-Volterra system, these policies or rules are the predator and prey birth and death rates. The TRUE (2014) software system is open-source application that can be used to create complex dynamic stock and flow diagrams using notation similar to that in the lower right of Figure 5b. Although the TRUE system models do show dynamically adjusting levels and flows rates, the models are somewhat complex, and are not designed specifically for novice users.

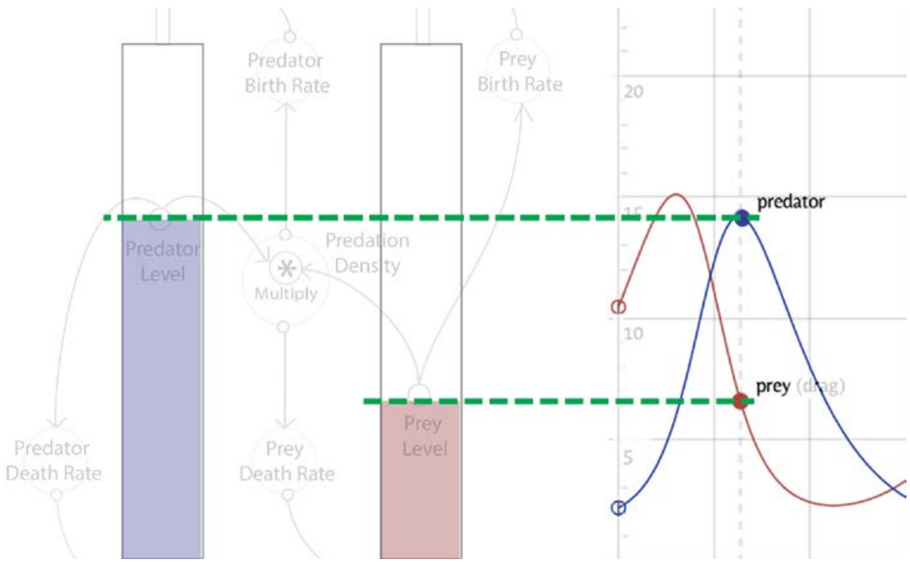


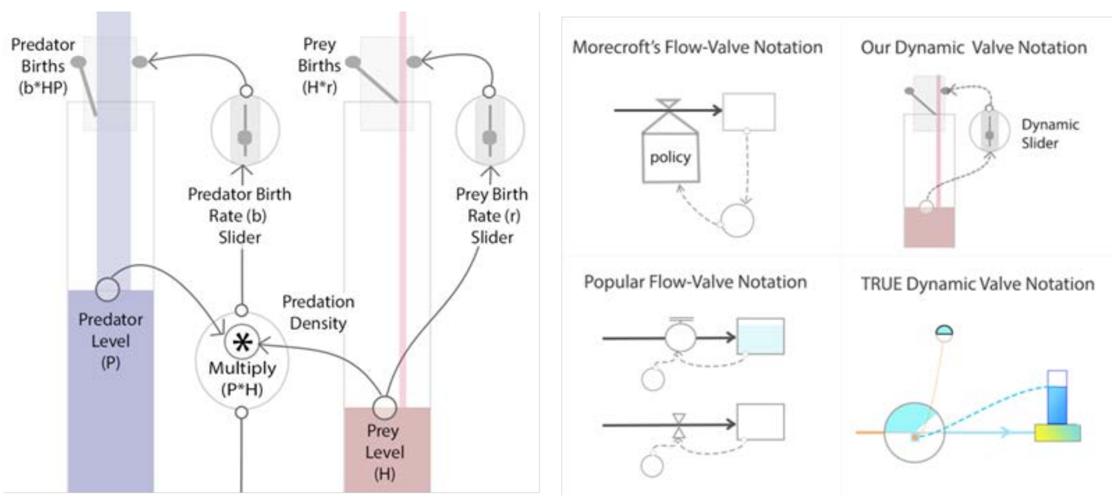
Figure 4: A schematic showing design features to enhance user understanding of relationship between dynamic population levels and the plot of behavior over time.

4.4 Flow Valve Design Details

The dynamic valves in our application have several design features to allow users compare flows between containers and over time. We have adjusted and limited the range of the parameter sliders, and normalized the flow in the valves so that the maximum flow corresponds to a fully open valve. Therefore, users can directly compare the current flow against the maximum and minimum possible flows. This type of comparison cannot be done in the TRUE system because that system uses circle radius to encode relative magnitudes, so it is not possible for users to know the relative maximum or minimum size of a circle. Because we are modeling a fixed system we can tune these design details to support novice users, however, in the TRUE system, their interface design must provide flexibility for all possible system configurations and this may impede their ability implement similar design features. In addition, since it would be difficult for users to visually distinguish between slight variations in the width of a flow, we have also encoded the flow volume using an angular hinge. This hinge angle provides a magnified view when comparing flows. In the TRUE system, it's not obvious what feature is encoded by circle size or what feature is encoded by the blue pie-chart angle. We have designed the visual details and interactive behavior of our application's interface to highlight salient features of the dynamic relationship between stocks and flows in order to scaffold novices understanding.

The ability of users to see dynamic changes in the population levels, flows and BoT plot in response to manipulation of the auxiliary parameter sliders will provide novices an understanding of the dynamic nature of feedback-loops, these trends would be difficult if not impossible to understand if users had no control over parameter values, or if they could only adjust the values using a text box as opposed to our dynamic slider. We conjecture that the following design features can scaffold users to develop enhanced understanding of feedback-loop structure through guided inquiry of dynamic interaction phenomena.

- Parameters ranges were capped, and flows were normalized so that relative flows can be observed and compared.
- The hinge valve angle provides a second method to visually encode flow magnitude
- Sliders were designed so that a vertical increase in the slider value corresponds an increase in flow and increase in valve opening size.
- Due to the multiplicative factor for the predation density, the sliders which control the predator births and prey deaths have the highest leverage in the system.
- Users can observe that small changes to sliders can result in large changes to the BoT graph if the population levels are currently high. This supports understanding of the non-linear behavior due to feedback-loop structure.
- Users can observe that large chances to the sliders do not instantaneously result in large changes to the system if the current population level is low.
- When users modify a slider, they will observe that the entire BoT graph shifts, this supports understanding that stocks / levels act as delays or buffer in the system.



a) b)

Figure 5: Figure 5a (left) shows interactive sliders which allow dynamic adjustment of system parameters. Notation also provides indication of auxiliary functions and relative flow values. Figure 5b: (right) shows a comparison of our notation with several other types of SD notation, our dynamic valve notation provides insight into feedback-loop dynamics.

5 CONCLUSIONS AND FUTURE WORK

We have designed an integrative multimodeling web-based application to scaffold users in learning System Dynamics modeling using the Lotka-Volterra equations which represent a wide range of interesting real world complex systems. Our application contains multiple dynamically-linked representations which have been designed to build on users intuitive understanding of water flow to scaffold their understanding of the system dynamic concepts of accumulation and feedback loops. Our application is not intended to be a full-featured modeling environment; therefore we were able to fine-tune the design of our user-interface to support users in extending their existing mental models of water flow to include these advanced water flow concepts. We have used visual, multimedia, and interaction design methodologies to inform our interface design decisions to minimize complexity, increase coherence across representations, and to provide redundant encoding of important features. Initial informal feedback from several STEM education curriculum experts suggests that our application may be a good fit for use in inquiry-based curriculum modules.

In the future, we intend to conduct user studies to build on existing research which focused on learners developing an understanding of accumulation and feedback-thinking. We are also planning to expand our research focus to explore how users construct understanding when using multiple interactive representations of abstract models. While our current web application prototype covers several different representational forms for the LV system, we plan to integrate our application with other LV prototype representations developed in our research lab which include a virtual environment model, and a tangible, micro-controller driven water machine. In addition, we are also interested in exploring representational forms of models that span Forrester's SD learning progression, this would include enhanced models of causal diagrams, narrative forms, and game based learning environments.

ACKNOWLEDGMENTS

We thank the Arts and Technology Department at the University of Texas at Dallas and the Computer Science Department at the University of Texas at Dallas. We also thank the office of the Dean of Engineering and Computer Science for the EEF Fellowship sponsorship of the first author. We also would like to thank Michael Howell, David Vega, and Rebecca Tull for discussions about virtual, tangible, and other representational models of the Lotka-Volterra and System Dynamics models.

REFERENCES

Ainsworth, S. 2006. "DeFT: A Conceptual Framework for Considering Learning with Multiple Representations." *Learning and Instruction* 16 (3): 183–98. doi:10.1016.

Ainsworth, S. 2008. "The Educational Value of Multiple-Representations When Learning Complex Scientific Concepts." In *Visualization: Theory and Practice in Science Education*, 191–208. Springer.

Bootstrap 2014. "Bootstrap." http://www.getbootstrap.com.

Cronin, M. A., Gonzalez, C., and Sterman, J. D. 2009. "Why Don't Well-Educated Adults Understand Accumulation? A Challenge to Researchers, Educators, and Citizens." *Organizational Behavior and Human Decision Processes* 108.

DiSessa, A. 2001. *Changing Minds: Computers, Learning, and Literacy*. MIT Press.

Doyle, J. K., and Ford, D. N. 1999. "Mental models concepts revisited: Some clarifications and a reply to Lane." *System Dynamics Review*, 15, 411–415.

Fishwick, P.A. 1995. *Simulation Model Design and Execution: Building Digital Worlds*, Prentice Hall, Englewood Cliffs, NJ.

Fishwick, P. A. 2004. "Toward an Integrative Multimodeling Interface: A Human-Computer Interface Approach to Interrelating Model Structures." *Simulation* 80 (9): 421–32.

Forrester, J. W. 1992. "System dynamics and learner-centered-learning in kindergarten through 12th grade education." D-4337. Text of remarks delivered December 1.

Forrester, J. W. 1994.." Learning through system dynamics as preparation for the 21st century. D-4434-1. MIT." Presented at the Systems Thinking and Dynamic Modeling Conference for K-12 Education, June 27-29, Concord, MA.

Forrester, J. W. 1993. "System Dynamics as an Organizing Framework for Pre-College Education." *System Dynamics Review* 9 (2): 183–94.

Forrester, J. W., 1961. *Industrial Dynamics*, Pegasus Communications, Waltham MA.

Forrester, J. W. 2009. "Learning through System Dynamics as Preparation for the 21st Century." In *Keynote Address for Systems Thinking and Dynamic Modeling Conference for K-12 Education*.

Golnam, A., Sanchez, R., Tapandjieva, G. and Wegmann, A. 2014. “Choosing the Right Tool for the Job: A Framework to Compare the Effectiveness of Problem Structuring Methods in System Dynamics Modeling.”

Groesser, S. N., & Schaffernicht, M. 2012. Mental Models of Dynamic Systems: Taking Stock and Looking Ahead. *System Dynamics Review*.

JSXGraph 2014. “JSXGraph.” http://jsxgraph.uni-bayreuth.de/wp/.

JSXGraph Lotka-Volterra Equations 2014. “Lotka-Volterra Equations” http://jsxgraph.unibayreuth.de/wiki/index.php/Lotka-Volterra_equations.

Kopainsky, B., and Saldarriaga. M. 2012. “Assessing Understanding and Learning about Dynamic Systems.” In *30th International Conference of the System Dynamics Society, St.Gallen, Switzerland*.

Lane, D. C., and Sterman, J. D. 2011. *Profiles in Operations Research : Jay Wright Forrester*. Springer.

Larkin, J. H., and Simon, H. A. 1987. "Why a Diagram is (Sometimes) Worth Ten Thousand Words**." *Cognitive science* 11(1): 65.

Lukaszewicz, R. 1976. “The Direct Form of Structural Models within System Dynamics”. Dynamica.

MathJax 2014 “MathJax.” http://www.mathjax.org.

Meadows, D. 2008. *Thinking in Systems: A Primer* . Chelsea Green Publishing.

Moody, D. 2009. “The Physics of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering.” *IEEE Transactions on Software Engineering* 35 (6).

Morecroft, J. D. 1982. “A critical review of diagramming tools for conceptualizing feedback system models.” *Dynamica*, *8*(1), 20-29.

Saldarriaga, M. 2011. “Using a water tank analogy to transform students´ intuitive knowledge of dynamic systems. A qualitative study of the case of motion.” Unpublished PhD dissertation, University of Bergen, Norway.

Schaffernicht, M. and Groesser, S. N., 2012. “Learning to Think in Circles: Improving Mental Models of a Dynamic System.” In *30th International System Dynamics Conference, St. Gallen, Switzerland*.

Sterman, J. D. 1994. “Learning in and about complex systems.” *System Dynamics Review*, 10, 291–330.

Sterman, J. D. 2002. “All Models Are Wrong: Reflections on Becoming a Systems Scientist.” *System Dynamics Review* 18 (4): 501–31.

Sterman, J. D. 2009. “Does Formal System Dynamics Training Improve People’s Understanding of Accumulation?” *Organizational Behavior and Human Decision Processes* 108 (1): 116–30.

Sweeney, Linda Booth, and John D. Sterman. 2000. “Bathtub Dynamics: Initial Results of a Systems Thinking Inventory.” *System Dynamics Review* 16 (4): 249–86.

Tangle.js 2014. “Tangle.js” http://worrydream.com/tangle.

TRUE Software 2014. “TRUE Software” http://true-world.com/htm/en/index.html.

Wagner, G. 2012. “Simurena: A Web Portal for Open Educational Simulation.” In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, 1588–1599. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

KAREN DOORE is a PhD candidate in the Department of Computer Science at the University of Texas at Dallas. Her research interests include computer science education, visualization to enhance modeling and simulation education, and human computer interaction. Her email address is Kdoore@gmail.com.

PAUL FISHWICK (Ph.D., University of Pennsylvania) He is Distinguished Endowed Chair of Arts and Technology (ATEC) and Professor of Computer Science at the University of Texas at Dallas. His email address is Paul.Fishwick@utdallas.edu