Table of Contents

Go Back

# System Blocks: A Physical Interface for System Dynamics Learning

**Oren Zuckerman and Mitchel Resnick**

MIT Media Laboratory

20 Ames Street E15-020, Cambridge MA 02139 USA

{orenz, mres} @ media.mit.edu

**ABSTRACT**

We present System Blocks, a physical interface that makes it easier for children to model and explore dynamic systems. A set of computationally enhanced blocks, made of wood and electronics, System Blocks can assist K-6 educators to teach the complex concepts of system dynamics and causalities. Learning to understand dynamic systems is an essential step in understanding the world around us. However, learning it at university, high school or even middle school level might be too late. By this age children have already developed their own models of how the world works. In this paper we will show how a set of physical objects can be used as a modeling and simulation tool, merging hands-on tinkering with computer simulation. Using blocks that behave as stocks, flows, variables and constants, our hope is that System Blocks will enable children younger than sixth grade to model, simulate and analyze systems that are meaningful to them.

**KEYWORDS**

System Dynamics, System Thinking, Education, Manipulative Materials, Tangible Interfaces

## INTRODUCTION

Walk into any kindergarten, and you are likely to see a diverse collection of "manipulative materials." You might see a set of Cuisenaire Rods: brightly colored wooden rods of varying lengths. The colors and lengths of the rods are carefully chosen to engage children in explorations of arithmetic concepts and relationships. Children discover that each brown rod is the same length as two purples or four reds. On the next table, you might see a set of Pattern Blocks. Children can use these polygon-shaped tiles to create mosaic-like patterns and, in the process, learn important geometric concepts.

As children build and experiment with these manipulative materials, they develop richer ways of thinking about mathematical concepts such as number, size, and shape. But there are many important concepts that are very difficult (if not impossible) to explore with these traditional manipulative materials.

System dynamics (SD) and system thinking (ST) are methods for studying the world around us. They deal with understanding how complex systems change over time, and how structure influences behavior (Forrester 1968; Senge 1990).

Learning to understand dynamic systems is an essential step in understanding the world around us. However, learning it at university, high school or even middle school level might be too late. By this age children already develop their own models of how the world works.

It has been long known that children construct their own meaning of the world based on their interaction with the physical world around them (Piaget 1972). In addition, constructionist research showed that people learn most effectively about the world when they are engaged in projects they care about (Papert 1991; Kafai and Resnick 1996).

Jay Forrester has emphasized that modeling and simulation are critical factors in understanding the "deeper lesson" of SD – for example, in his keynote address for the "Systems Thinking and Dynamic Modeling Conference for K-12 Education" (Forrester 1984).
In this paper we will show how a set of physical objects can be used as a modeling and simulation tool, merging hands-on activity with computer simulation and enabling K-6 children to model, simulate and analyze systems that are meaningful to them.

Our hope is that through constructive processes System Blocks will contribute to a gradual development of a "systems mental model" that will serve children throughout their adult lives, in making better decisions for themselves, their society and their environment.

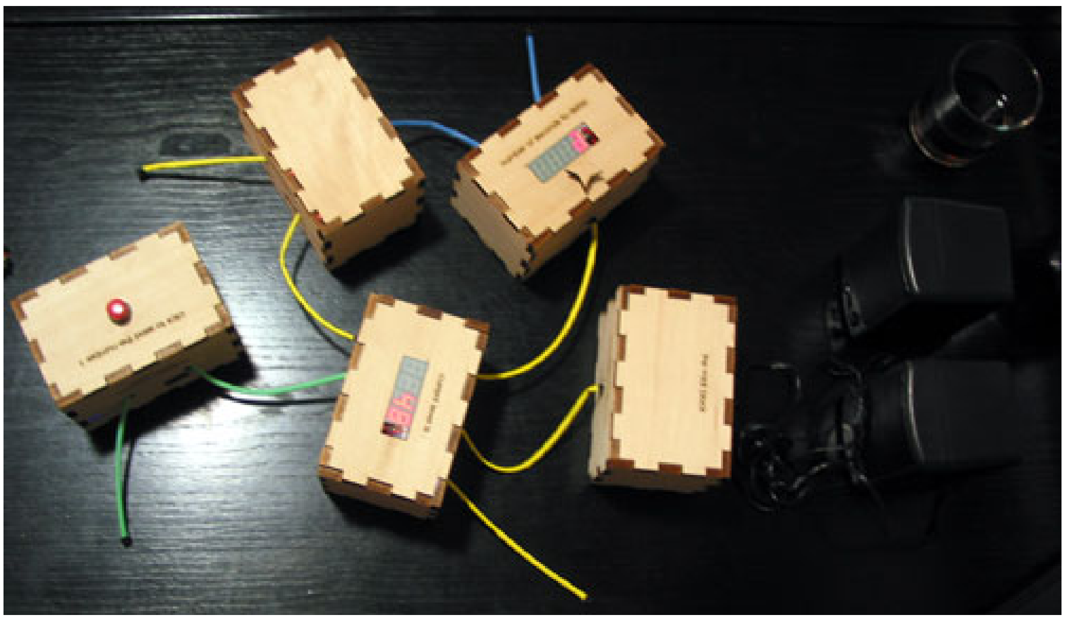

Figure 1 – System Blocks

**THEORETICAL FRAMEWORK**
The System Blocks theoretical framework comes from thinkers in the education/epistemology and SD research areas (Figure 2). On the education/epistemology side, this project shares its theoretical framework with a broader research effort at the MIT Media Lab, called 'Digital Manipulatives' (Resnick et al. 1998).

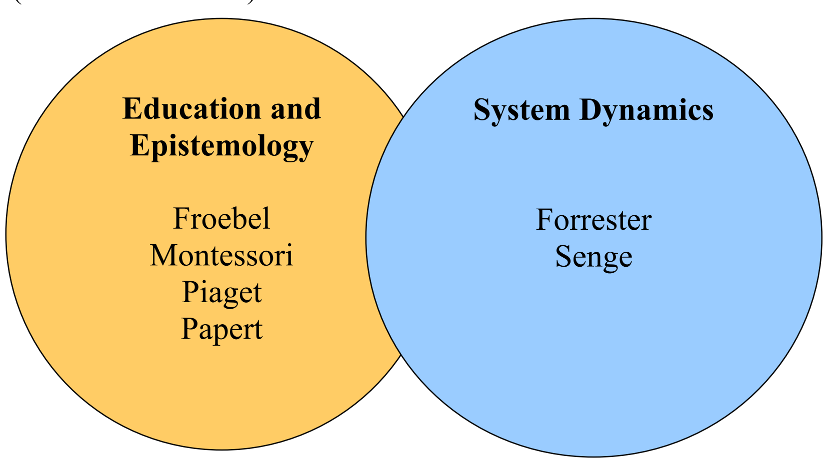


Figure 2 – Theoretical Framework

The idea that physical objects might play an important role in the learning process is a relatively new idea. Until the 19th century, formal education focused almost exclusively on lectures and recitations. One of the first advocates for "hands-on learning" was the Swiss educator Johann Heinrich Pestalozzi (1746-1827). Pestalozzi asserted that students need to learn through their senses and through physical activity, arguing for "things before words, concrete before abstract" (Pestalozzi 1803).

Friedrich Froebel, who created the world's first kindergarten in Germany in 1837, was very influenced by Pestalozzi's ideas. Froebel's kindergarten was filled with objects for children to play with. Froebel developed a specific set of 20 "gifts" -- physical objects such as balls, blocks, and sticks -- for children to use in the kindergarten. Froebel carefully designed these gifts to help children recognize and appreciate the common patterns and forms found in nature. Froebel's gifts were eventually distributed throughout the world, deeply influencing the development of generations of young children. Indeed, Frank Lloyd Wright credited his boyhood experiences with Froebel's gifts as the foundation of his architecture (Brosterman 1997).

Maria Montessori extended Froebel's ideas, developing materials for older children and inspiring a network of schools in which manipulative materials play a central role. In an effort to create an "education of the senses" (Montessori 1912), Montessori developed new materials and activities to help children develop their sensory capabilities. Montessori hoped that her materials would put children in control of the learning process, enabling them to learn through personal investigation and exploration.

Jean Piaget provided an epistemological foundation for these educational ideas. Piaget theorized that children must first construct knowledge through "concrete operations" before moving on to "formal operations" (Piaget 1972). During the past decade, a new wave of research has suggested that Piaget, if anything, understated the importance of concrete operations. Sherry Turkle and Seymour Papert, for example, have argued for a "revaluation of the concrete," suggesting that "abstract reasoning" should not be viewed as more advanced than (or superior to) concrete manipulations (Turkle and Papert 1990).

## RELATED WORK

The System Blocks design was inspired and influenced by research in the following three domains: screen-based simulation environments for systems concepts; tangible interfaces for digital information representation; physical programmable devices for children. Specifically, at the MIT Media Lab, people and projects from the Tangible Media, Lifelong Kindergarten and Grassroots Invention groups were a source of inspiration.

Impressive screen-based simulation environments for complex systems were developed in the last several decades, such as Stella (Roberts 1983), Vensim, Model-It, and StarLogo (Resnick 1996). System Blocks shares many themes with these tools, but adds the physical interface.

The Tangible Media Group at MIT Media Lab has done pioneering work in the field of tangible interfaces (Ishii 1997). They introduced a new approach for interacting with system dynamics models, using the “Sensetable” tabletop display surface (Patten 2001). System Blocks are based on a more constructionist approach enabling creation of new models. Also, System Blocks embed new systemic behavior inside the physical blocks rather than projecting it from standard system dynamics software.

The Lifelong Kindergarten group at MIT Media Lab has previously introduced a collection of digital manipulatives (Resnick 1998) – computationally augmented versions of programmable bricks, beads, balls, and badges. Children program the programmable bricks from a personal computer and use them to create robotic toys and kinetic sculptures. System Blocks follow the tradition of digital manipulatives, but adds a "physical programming" approach. Rather than programming the System Blocks from a personal computer, children change blocks' arrangements to create new models of different systems.

Previous work has been done on "physical programming" interfaces for children, such as the Electronic Duplo Blocks (Wyeth 2002) or the Tangible Programming Bricks (McNerney 2000). These projects focused on general programming concepts, while System Blocks focus on dynamic systems simulation.

Tim Gorton from the Grassroots Inventions group at MIT Media Lab has recently introduced the Tabletop Process Modeling Toolkit (Gorton 2002), a physical network interface that can be programmed to simulate complex systems. System Blocks share many themes with this project, but differ in the age of the target audience, in the SD principles representation and in the ability to model new types of systems without programming but rather by changing blocks' arrangements.

## IMPLEMENTATION

System Blocks are a physical construction kit. Children connect different blocks using cables, and interact with the blocks using dials and buttons mounted on the blocks' cover. When two blocks are connected, digital data is being passed from one block to the other. The blocks follow the same concept implemented in Stella and Vensim, where the direction of a connection between blocks determines the direction of the causality. By connecting the blocks in different ways, children can create different types of systems.

The implementation of System Blocks can be divided into three categories: Infrastructure, Behaviors and Representations. The Infrastructure stands for the hardware and firmware architecture. The Behaviors stands for the SD principles and equations implemented in each block. The Representations stands for the different media types used to present the dynamic behavior to the student.

**Infrastructure** - System Blocks are designed as a decentralized system. Blocks are not aware of their neighbors, and there is no central control. This is a simple yet robust design. It scales without compromising performance and it can be easily extended with new functionality using new blocks' behaviors.

The hardware implementation had two phases. First, the blocks were prototyped using the Tower System, a modular prototyping toolkit developed at the Grassroots Invention group at the MIT Media Lab (Lyon 2003; Gorton 2003). The Tower, and specifically the serial communication capabilities developed by Tim Gorton, enabled quick and effective prototyping of programmable hardware. Second, after few months of experimentation with the first implementation, a dedicated Printed Circuit Board (PCB) was designed and fabricated, using the PIC microprocessor and the Logochip development kit (Mikhak, Silverman and Berg 2002; Lyon 2003). The System Blocks PCB was designed with K-6 teachers in mind, using the following design guidelines: simplicity, cost and programmability. The Logochip provides a powerful programmable development platform at low hardware cost, so it was a natural choice for this project. The main features supported by the System Blocks PCB are: transfer power between blocks, 16-bit number system, software-based serial communication with multiple input ports, analog to digital sensor ports, Cricket bus device port and the Logochip Virtual Machine, enabling programmability using the Logo programming language (Papert 1980).

**Behaviors** – Each block has a predefined behavior that it performs when needed. Behaviors are the equivalent of Stella's and Vensim's equations. Each block has a predefined equation. Unlike Stella's or Vensim's equations, System Blocks equations are predefined and cannot be changed. In order to change a specific equation, one must switch to a different block. For the end user, the block's behavior is the equation. This conscious design decision limits the possible models the System Blocks can represent, but on the other hand significantly simplifies the modeling process. After several iterations of design, implementation and testing, the following behaviors were implemented. Note that the blocks' names are not yet adapted to K-6 audience.

***Discrete Sender Block*** has a large button as its interface. When clicked, the number "one" is sent out through the output cables. Each additional click sends out an additional number "one." There is no internal timing mechanism; rather, the user has complete control over the rate.
***Continuous Sender Block*** has a slider as its interface. Each number on the slider scale represents the value to be sent out. This block has an internal timing mechanism, and values are sent out at a fixed rate.
***Accumulator Block*** has an LED display as its interface. It receives input from multiple ports, either "in" or "out" ports. Each value received is either added to or subtracted from the accumulated level, based on its input port ("in" will be added and "out" will be subtracted). This block has an internal timing mechanism. At each timing event the accumulated level is sent out through the output cables.
***Multiplier Block*** has no interface. It receives input from other blocks and multiplies them by each other. The result is sent out through the output cables at every internal timing event.
***Subtraction Block*** has no interface. It receives input from other blocks and subtracts them from each other. The subtraction order is defined by the order of the input ports. The result is sent out through the output cables at every internal timing event.
***Addition Block*** has no interface. It receives input from other blocks and adds them to each other. The result is sent out through the output cables at every internal timing event.

**Representations** – Multiple representations can enhance the learning experience. Some of the representations used by Stella and Vensim are graph display, "bathtub-like" level or a table. In System Blocks we have implemented several "Representation" blocks. Each Representation block can be connected to any output cable of any other block and perform its representations based on the received values.

***Digit Display*** – an LED-based display that can show up to 4-digit numbers.

***Graph Display*** – an LCD-based display that can show a graph of the behavior over time.

***MIDI Sound*** – a sound block that connects to standard computer speakers. It translates the received values to musical notes using the MIDI format. A piano scale is used to play the notes, where 0 is used as the lowest note on the scale and 127 as the highest.

***Motor Control*** – a physical movement block, with a build-in motor. The motor can operate standard or custom-made devices, such as a propeller or a kinetic sculpture.

**MODELING AND SIMULATION**

The System Blocks can simulate Stella or Vensim models with the following limitations: the variety of equations is limited, the time step is fixed, the number of connections between blocks is limited and the range of numbers used for simulation is limited.

When building a model using the System Blocks, the following translation from Stella/Vensim models can be used:

| **Stella / Vensim** | **System Block equivalent and limitations** |
|---|---|
| Stock | ***Accumulator block.***<br>Equation is fixed as “flow in” – “flow out”.<br>Initial value is set as “0”.<br>“flow in” connections are limited to two.<br>“flow out” connections are limited to two. |
| Flow | ***Multiplier block.***<br>Equation is fixed as “var 1” * “var 2”<br>“Variable” connections are limited to four.<br>Feedback is possible by connecting the Accumulator output back to the Multiplier as a Variable. |
| Auxiliary Variable | ***Multiplier block.***<br>Same as “flow” definition. |
| | ***Subtraction block.***<br>Equation is fixed as “var 1” - “var 2”<br>“Variable” connections are limited to four. |
| | ***Addition block.***<br>Equation is fixed as “var 1” + “var 2”<br>“Variable” connections are limited to four. |
| Constant | ***Discrete sender block.***<br>Equation is fixed as “value”.<br>Value is limited to an integer, 0 – 9. |
| | ***Continuous sender block.***<br>Equation is fixed as “value”.<br>Rate is fixed at 10 values a second.<br>Value is a real number between 0 - 9, limited to two digits after the decimal point. |

**Model 1** – Continuous Accumulation

In model 1 we see the classic stock and flow model: one stock; one flow in; one flow out and two constant variables as input into each flow. The variables "sender 1" and "sender 2" will adjust the rate of flow in and flow out. Different dynamics can be simulated using this model, including linear growth, linear decrease and linear dynamic equilibrium. There are sliders on the "continuous senders" blocks to enable real-time interaction with the system. Figure 3 shows model 1 implemented in Vensim. Figure 4 shows Model 1 implemented in System Blocks.

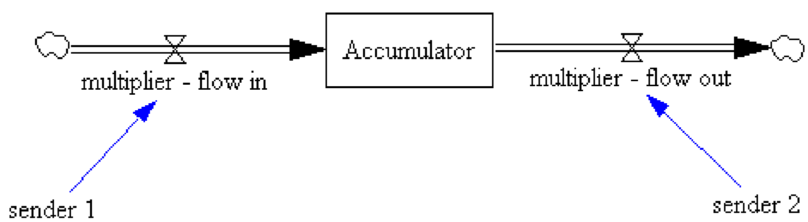


Figure 3. Model 1. Continuous accumulation using Vensim.

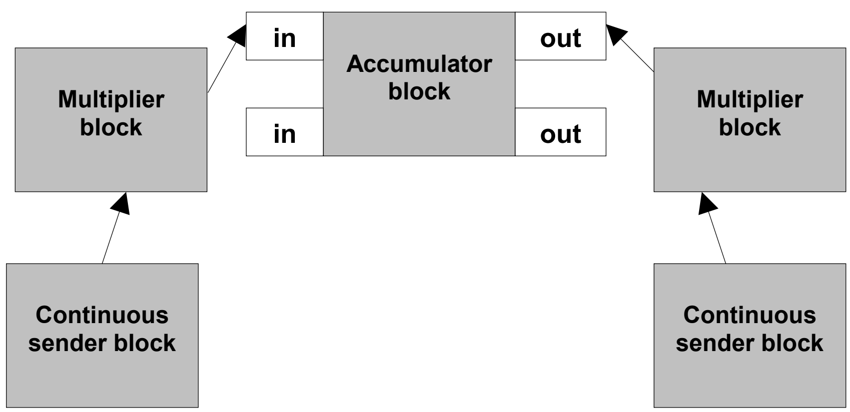


Figure 4. Model 1. Continuous accumulation using System Blocks.

**Model 2** – Exponential Decay

In model 2 we see non-linear behavior. It's the same model as the continuous accumulation one but with feedback. The accumulated level is reported from the stock to the flow out. In this model, the flow out is a multiplication of the stock level and the "sender 2" variable, so the lower the stock level is, the less is the amount flowing out. If the simulation starts with an initial value for the stock, the dynamic behavior is exponential decay. There are sliders on the "continuous senders" blocks to enable real-time interaction with the system. An analogous example can be water flowing into and out from a bathtub. Figure 5 shows Model 2 implemented in Vensim. Figure 6 shows Model 2 implemented in System Blocks.

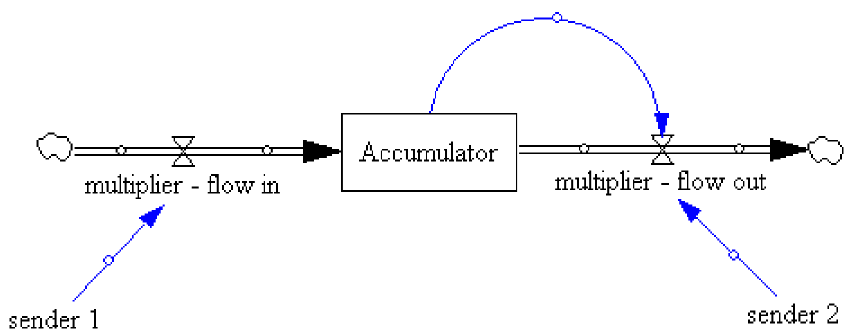


Figure 5. Model 2. Exponential decay using Vensim.

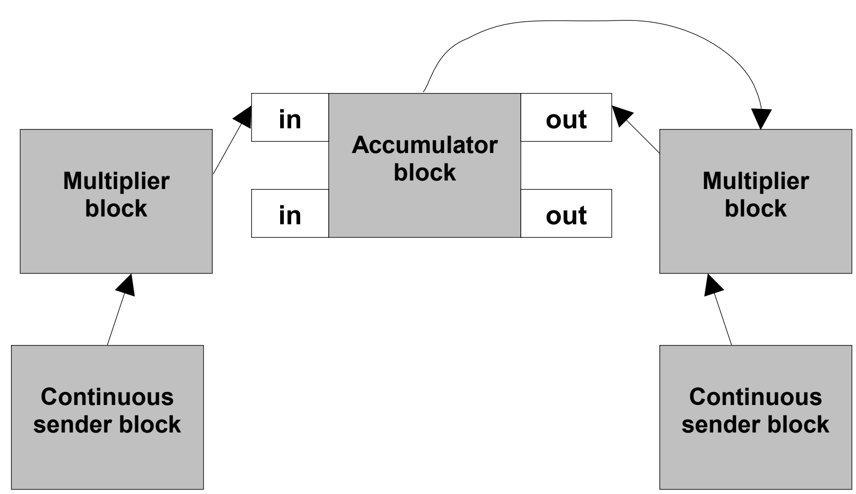


Figure 6. Model 2. Exponential decay using System Blocks

**Model 3** – Discrete Accumulation
In this model we see discrete behavior. Each "discrete sender" block has a button to enable interaction with the system. An analogous example can be a child's stack of toys: she can add one toy at a time into the stack (flow in), and another child can take toys out of her stack, one at a time (flow out). Figure 7 shows Model 3 implemented in System Blocks.

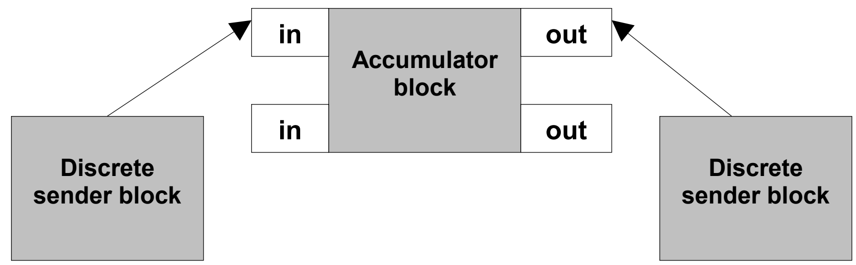


Figure 7. Model 3. Discrete accumulation using System Blocks.

## CONCRETIZING THE ABSTRACT

A SD model is abstract. Learning is most effective when children engage with concrete activities that are meaningful to them (Papert 1991). How do we bridge this gap? How do we "concretize the abstract?" A physical, tangible interface can help. Children can touch it, can tinker with it, but a System Blocks model is still abstract.

Stella and Vensim force you to name your variables. So when you build a rabbit population model, your stock is called "number of rabbits" and the flow-in and flow-out are called "births" and "deaths". This can help, but still does not feel so concrete. Linda Booth Sweeney has done excellent work by connecting SD principles with children stories (Sweeney 2001). Stories might be more concrete, and are used throughout history to convey important concepts. But how does one translate this into modeling?

Uri Wilensky offers his perspective on abstract vs. concrete: 'the more connections we make between an object and other objects, the more concrete it becomes for us. The richer the set of representations of the object, the more ways we have of interacting with it, the more concrete it is for us. Concreteness, then, is that property which measures the degree of our relatedness to the object, (the richness of our representations, interactions, connections with the object), how close we are to it, or, if you will, *the quality of our relationship with the object*.' (Wilensky 1990).

System Blocks have the potential to become concrete, but they are not there yet. In the spirit of the outstanding work done by the Waters Foundation teachers, System Blocks needs class materials and lesson plans to scaffold the learning experience. Our next design steps include making the blocks' abstract interface customizable, so children can insert their own metaphors and make the experience more meaningful to them. Adding card slots on each block for small drawings and pictures, or adding whiteboard material on a block's top face so children can use markers to write and draw on it might be two ways to address this issue. New ways of representing the dynamic behavior is another important area to pursue. The MIDI Sound block seemed to be a good start, the motor control will add a kinetic dimension. Connection back to the computer can be useful to present on-screen animations that can both entertain and concretize a model, especially when displayed through a video projector.

## EVALUATION

An initial evaluation was performed with four children, 4, 6, 10, and 13 years old, mainly to test how children interact with the blocks, not how they build SD models. The children reacted very well to the blocks. They were engaged for more than 45 minutes, tinkering with the interface and connecting different arrangements. The four-year-old wanted to look inside the blocks, and said that the electronics inside were "beautiful." The MIDI sound block was well accepted, and some children said it helped them understand the accumulation direction (based on the notes played – a scale going up means accumulation increasing and scale going down means decreasing.) The initial evaluation showed that the blocks can be treated as play objects, and that a physical interface with multiple representations is engaging.

The next step is to perform a comprehensive evaluation in a classroom environment, using SD curriculum. The goal of the second evaluation will be to test if K-6 children are capable of building, simulating and analyzing SD models using System Blocks. We plan to collaborate with existing SD K-6 teachers and use existing SD curriculum and projects, such as the Waters Foundation lesson plans.

## CONCLUSION AND FUTURE WORK

We have presented System Blocks, a physical interface for SD learning. Our vision is that K-6 children will model using System Blocks and then move on to model using Stella or Vensim where they could benefit from a comprehensive simulation environment.

Our initial evaluation showed that System Blocks are playful and engaging for children as young as 4 years old, and that multiple representations and specifically sound can be an effective way to communicate dynamic behavior.

In the future we plan to continue to develop the blocks to be an effective teaching tool for SD and ST. We believe that learning should be playful, hands-on and meaningful.

Our vision is that through constructive processes, System Blocks will contribute to a gradual development of a "systems mental model" that will serve children throughout their adult lives, in making better decisions for themselves, their society and their environment.

## ACKNOWLEDGMENTS

We would like to thank: John Hernandez, Alda Y Luong, Ji Zhang, Timothy Brantley, Tim Gorton, Bakhtiar Mikhak, Robbie Berg and Brian Silverman for their support with hardware design; Gokhan Dogan and Hazhir Rahmandad for clarifying SD principles; Jay Forrester and Peter Senge for their interest and input; Linda Booth Sweeney for her inspiration and design ideas; Hiroshi Ishii and the TMG group for their input; Tina Grotzer for her kind advice and Michael Smith-Welch for his insights.

This research could not have been done without the generous support of the LEGO Company, the MIT Media Lab's Center for Bits and Atoms (NSF CCR-0122419), Things That Think and Digital Life consortia.

Portions of this paper were taken with permission from the "Digital Manipulatives" paper (Resnick et al. 1998).

**REFERENCES**


Brosterman, N. 1997. Inventing Kindergarten. Harry N. Adams Inc.

Forrester JW. 1990. Principles of Systems. Pegasus Communications: Waltham, MA.

Forrester JW. 1994. Learning through System Dynamics as Preparation for the 21st Century. Keynote Address for Systems Thinking and Dynamic Modeling Conference for K-12 Education. http://sysdyn.mit.edu/sdep/papers/D-4434-1.pdf

Gorton, T., Mikhak, B., Paul, K. 2002. Tabletop Process Modeling Toolkit: A Case Study in Modeling US Postal Service Mailflow. Demonstration at CSCW 2002. http://gig.media.mit.edu/publications.html

Gorton, T. 2003. Tangible Toolkits for Reflective Systems Modeling. Masters Thesis, Massachusetts Institute of Technology. Cambridge, MA.

Ishii, H. and Ullmer, B. 1997. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms, in Proceedings of Conference on Human Factors in Computing Systems (CHI '97), ACM Press, pp. 234-241.

Kafai, Y, Resnick, M., Eds. 1996. Contructionism In Practice: designing, thinking and learning in a digital world. Lawrence Erlbaum Associates, Mahwah, New Jersey.

Lyon, C. 2003. Encouraging Innovation by Engineering the Learning Curve. Master of Engineering Thesis, Massachusetts Institute of Technology. Cambridge, MA.

McNerney, Tim S. 2000. Tangible Programming Bricks: An approach to making programming accessible to everyone. Masters Thesis, MIT Media Lab. Cambridge, MA.

Mikhak, B., Silverman, B., Berg, R. 2002. Logochip: A Playful Introduction to Electronics. Grassroots Invention Group internal memo. MIT Media Lab, Cambridge, MA.

Montessori, M. 1912. The Montessori Method. New York Frederick Stokes Co.

Papert, S. 1980. Mindstorms: Children, computers and powerful ideas. Basic Books, New York,

Papert, S. 1991. Situating constructionism. In Papert & Harel, Eds., Constructionism. MIT Press: Cambridge, MA.

Patten, J., Ishii, H., Hines, J., Pangaro, G. 2001. Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces. Proceedings of Conference on Human Factors in Computing Systems (CHI '01), ACM Press, pp.253-260.

Pestalozzi, H. 1803. ABC der Anschauung, oder Anschauungs-Lehre der Massverhaltnisse. Tubingen, Germany: J.G. Cotta.

Piaget, J. 1972. The Principles of Genetic Epistemology. New York Basic Books.

Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., and Silverman, B. 1998. Digital Manipulatives: New toys to think with. Proceedings of Conference on Human Factors in Computing Systems (CHI '01), ACM Press, pp.281–287.

Resnick, M. 1996. Turtles, Termites, and Traffic Jams. MIT Press.

Roberts, N., Anderson, D., Deal, R., Garet, M., and Shaffer, W. 1983. Introduction to Computer Simulation: A System Dynamics Modeling Approach. Reading, MA: Addison-Wesley.

Senge PM. 1990. The Fifth Discipline: The Art and Practice of the Learning Organization. Doubleday, New York.

Sweeney, LB. 2001. When a Butterfly Sneezes: A Guide for Helping Kids Explore Interconnections in Our World Through Favorite Stories. Pegasus Communications.

Turkle, S., and Papert, S. 1990. Epistemological Pluralism. Signs, vol. 16, no. 1.

Wilensky, U. 1990. Abstract Meditations on the Concrete and Concrete Implications for Mathematics Education. Constructionism. Harel, I. & Papert, S. (Eds.). Norwood, MA: Ablex Publishing.

Wyeth P, Purchase C. H. 2002. Tangible programming elements for young children. Proceedings of Conference on Human Factors in Computing Systems (CHI '02), ACM Press, pp.774 – 775.


Back to the Top