Sculpting Behavior

A tangible language for hands-on play and learning

Hayes Solos Raffle

B.A. in Fine Arts (Sculpture), Yale University, May 1996
M.S. in Media Arts and Sciences, MIT Media Lab, June 2004

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences
at the Massachusetts Institute of Technology

June 2008

Author ____________________

Hayes Solos Raffle
Program in Media Arts and Sciences
May 2, 2008

Certified by ____________________

Hiroshi Ishii
Muriel R. Cooper Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Accepted by ____________________

Deb Roy
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Sculpting Behavior

A tangible language for hands-on play and learning

Hayes Solos Raffle

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning on May 2, 2008
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences
at the Massachusetts Institute of Technology

Abstract

For over a century, educators and constructivist theorists have argued that children learn by actively forming and testing – constructing – theories about how the world works. Recent efforts in the design of "tangible user interfaces" (TUIs) for learning have sought to bring together interaction models like direct manipulation and pedagogical frameworks like constructivism to make new, often complex, ideas salient for young children. Tangible interfaces attempt to eliminate the distance between the computational and physical world by making behavior directly manipulable with one's hands. In the past, systems for children to model behavior have been either intuitive-but-simple (e.g. curlybot) or complex-but-abstract, (e.g. LEGO Mindstorms). In order to develop a system that supports a user's transition from intuitive-but-simple constructions to constructions that are complex-but-abstract, I draw upon constructivist educational theories, particularly Bruner's theories of how learning progresses through enactive then iconic and then symbolic representations.

This thesis present an example system and set of design guidelines to create a class of tools that helps people transition from simple-but-intuitive exploration to abstract-and-flexible exploration. The Topobo system is designed to facilitate mental transitions between different representations of ideas, and between different tools. A modular approach, with an inherent grammar, helps people make such transitions. With Topobo, children use enactive knowledge, e.g. knowing how to walk, as the intellectual basis to understand a scientific domain, e.g. engineering and robot locomotion. Queens, backpacks, Remix and Robo add various abstractions to the system, and extend the tangible interface. Children use Topobo to transition from hands-on knowledge to theories that can be tested and reformulated, employing a combination of enactive, iconic and symbolic representations of ideas.

Thesis Supervisor:
Professor Hiroshi Ishii
Muriel R. Cooper Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Sculpting Behavior

A tangible language for hands-on play and learning

Hayes Solos Raffle

The following people served as readers for this thesis:

Thesis Reader

Mitchel Resnick
LEGO Papert Professor of Learning Research
Massachusetts Institute of Technology

Thesis Reader

John Maeda
Associate Director of Research
E. Rudge and Nancy Allen Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Contents

This work is dedicated to my family,
Rachel, Paloma and Anika.

Sculpting Behavior

A tangible language for hands-on play and learning

Hayes Solos Raffle

Acknowledgements

The process of designing and producing Topobo has involved a large and fluid team of collaborators, and the end uses to which it has been imagined and used are marked by children's ideas, critique, and work. Children have also been involved at all stages of development and observations of their styles of work as well as their direct feedback has actively influenced the research questions I have pursued. I have many people to acknowledge.

First, I would like to thank Hiroshi Ishii for providing the environment, support and inspiration to pursue this work. He has been an insightful advisor, and this work could not have been completed anywhere besides the Tangible Media Group.

Second, many teachers, researchers, parents, and thousands of children have helped make this research possible. Thank you all.

Third, I would like to thank my major collaborators on the project. I am grateful to Amanda Parkes for her help to get the original system off the ground, and for teaching Topobo how to have a good sense of style. Amanda also provided insightful analysis of many of the user studies. I owe many thanks to Josh Lifton for architecting the Topobo firmware with distributed network protocol and operations. Josh "taught me to fish," showing me how to program and design firmware in this process. I also owe much to the many undergraduates who worked on this project as part of their UROPs. Many of them worked on aspects of Topobo over the course of several years, plugging away at engineering and design details until things worked perfectly. Laura Yip did extensive work on firmware development, Mike Fleder robustifed the network protocol and ported firmware, Andy Lieserson designed many circuit boards, and Elysa Wan kept our fragile prototypes in operation. Thanks to Jonathan Bachrach for the name "Topobo," and thanks to Dave Merrill, for finding the Topobo sound track.

Several chapters of this thesis are based on work that has been coauthored and previously published. Chapters 4 about the original Topobo system was written with Amanda Parkes and includes some sociological analysis by Cristobal Garcia [Raf04]. Chapter 5 Backpacks [Raf06] was written with Parkes and Lifton and Chapter 6 [Raf07] references work developed by Laura Yip. Chapter 8 Topobo in the Wild [Par08] was written with Parkes, based on our joint research.

This project has been funded by the MIT Media Lab and the Things That Think consortium, largely an outcome of Hiroshi's tireless fund raising. The manufacturing and educational outreach was funded by an iCampus (Microsoft) student grant, and many museums have helped support this project through their partnerships.

My greatest supporters through this process – six years of my life – have been my wife Rachel, and our daughters Paloma (3) and Anika (1) who put my individual productivity in perspective as the joy of their lives lights up smaller accomplishments like this thesis along the path of living. Thank you. You are the most important people to me and have kept me balanced in these years at MIT. I love you all without bounds, and this work is dedicated to you. May our lives be filled with play, learning and love.

Foreward

Two of my most memorable childhood toys were my cuisinaire rods, which I still own today, and Sculpey polymer clay.

My cuisinaire rods, with their brightly colored rods were creative objects to make mosaics and discover patterns, and measuring instruments to discover principles of mathematics. I had such a strong relationship to these little blocks that my mother insisted my second grade teacher Mrs. Engh allow me to bring them to the classroom so that I could play with them during free choice periods or use them to help me learn math. They were part of my process of discovery through creative play and research through art and design.

Ever since I can remember, I have loved to sculpt objects, whether with wood, clay, or assemblage of materials. My grandfather, a chemist, was married to an artist and in the course of his life he invented the first polymer clay called Sculpey. Sculpey was ever-present in my childhood, and I would often harden small objects in the oven or play with extruders, knifes, and my hands to manipulate the material. My grandfather taught me to love discovery through invention, to understand that inventing a new medium and putting it in people's hands can open new possibilities to them and allow them to create new and very personal things.

Blocks and clay. Topobo introduces a new set of physical building blocks based not only on number, but also on formal relationships from natural systems like crystals and skeletons. It also introduces a new programming model that is clay-like; *kinetic memory*, amorphous as a gesture — Topobo is programmed without the structure and constraints of blocks (or a block-like language), but instead with the fluidity of the human body. Linguistic concepts lend a grammar to the system, so children can play with gestural programs with flexible and abstract tools. Topobo programming refers back to the first principles of the body, to kinesthesia and children's knowledge about how to initiate action in the world.

Because as we know, the best learning of new language comes from children and play.

Introduction

Learning by doing

For over a century, educators and constructivist theorists have argued that children learn by actively forming and testing – constructing – theories about how the world works. In many cases, this kind of learning has been facilitated by providing specialized toys (manipulatives) for children to build specific kinds of models [Pia52; Bru04]. With the introduction of computers, researchers were inspired to introduce certain complex and dynamic ideas (like feedback and emergence) to children by creating systems for children to author dynamic systems, and systems with behavior by creating computer programs. In combining physical manipulatives with programming languages that enable embedded physical (robotic) and information behavior, "digital manipulatives" enabled children to create physical models with embedded physical (robotic) and information behavior [Res98]. The most popular of such systems rely on a separate screen-based programming interface that use iconic procedural (block based) programming systems that permits abstract and extensible program structures, e.g. LEGO Mindstorms. However, decoupling the tangible manipulative from the programming activity has made many systems concepts inaccessible to younger children.

Tangible Interfaces for learning

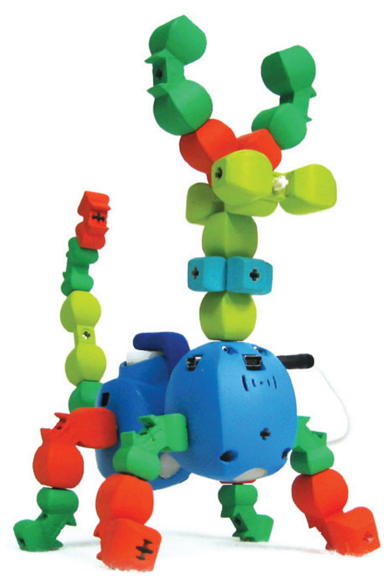
Recent efforts in the design of "tangible user interfaces" (TUIs) for learning have sought to bring together interaction models like direct manipulation and pedagogical frameworks like constructivism to make new, often complex, ideas salient for young children [O'Ma05]. My work extends the breadth of interaction techniques in tangible interfaces so that even young children may more intuitively explore new ideas through the creation of dynamic compositions. Although there is value in a variety of computational media for children (both screen-based and tangible), research has suggested several areas where tangibles may provide advantages over screen-based computational educational media:

- in collocated, collaborative learning exercises
- for tasks emphasizing motor skills and kinesthetic development
- in situations involving spatial problem solving
- in situations where a GUI may be overly complex, distracting or aesthetically inappropriate
- in applications where the user controls many things simultaneously

Tangible interfaces attempt to eliminate the distance between the computational and physical world by making behavior directly manipulable with one's hands. Tangibles have the potential to enable a new class of digital manipulative that is accessible to younger children. Many tangibles have been argued to make computing concepts more intuitive [O'Ma05], allowing people to create simple dynamic constructions. However, the challenge has been to create a tangible system that is both accessible to young children and can remain engaging for children as they develop and grow. Or, in the educator's words, how can a digital manipulative have both a "low floor" (easy to get started) and a "high ceiling" (be flexible and extensible). Such a system must have the benefits of both hands-on design and programming, and be able to support more abstract representation and manipulation of computational behavior.

In the past, systems for children to model behavior have been either intuitive-but-simple, (e.g. curlybot [Fre00]) or complex-but-abstract (e.g. LEGO Mindstorms). In general, Tangibles have been criticized for being intuitive, but too simple, and programming languages have been criticized for being sophisticated, but (especially for younger children) hard to learn. In order to develop a system that supports a user's transition from intuitive-but-simple constructions to constructions that are complex-but-abstract, I look to foundation theory from the learning sciences.

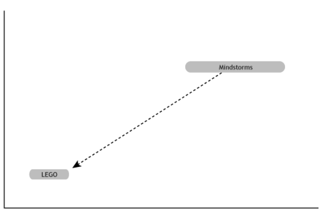
Movement in multiple degrees of freedom: by using multiple Actives to assemble a global motion



Theoretical foundation: an epistemological framework for HCI

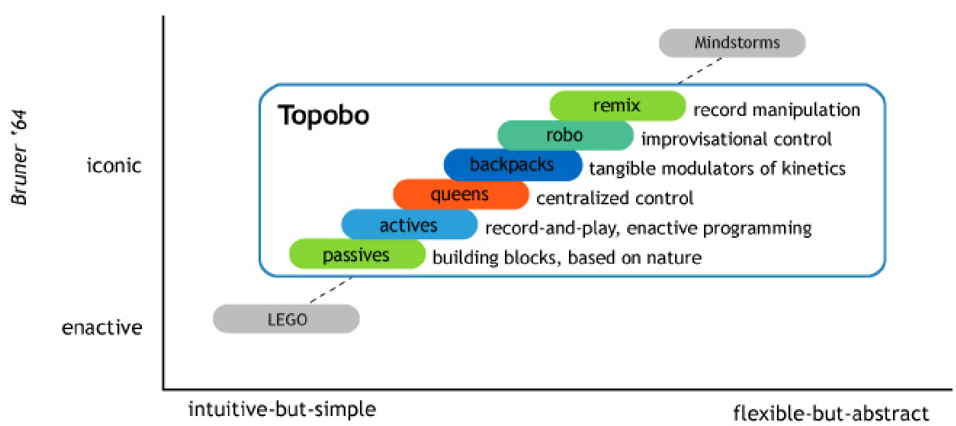
Constructivist educational theorists such as Piaget, Bruner, and Pestalozzi have profoundly influenced HCI researchers such as Seymour Papert [Pap80] and Alan Kay [Kay89]. Both of these visionaries regarded the computer as a "tool to think with" and sought a framework to guide the invention of a computational machine that is a creative medium.

Bruner [Bru04], after Piaget [Pia52, Col01], described a rather linear sequence of stages all people seem to progress through as they represent and acquire knowledge. Knowledge is first represented and acquired through doing things. *Enactive* representations describe how physical things are done. Some knowledge, such as learning to ride a bike, can not be adequately described in any other way, and must be learned through action. Later, most knowledge can be abstracted, represented and manipulated with symbols. *Iconic* representation is characterized by using and manipulating images to identify and represent ideas. Later, language and grammatical rules can be used to play with relationships of ideas, allowing people to fluidly reformulate and test concepts. *Symbolic* representations are characterized by an abstract mapping of a word to an idea, and a grammar with rules that allows flexible manipulations of ideas through symbolic manipulations. It is the hallmark of language, either natural or invented, e.g. the English language, math, music, a computer language. Piaget demonstrated that children seem to progress through these various stages of acquiring and representing knowledge through predictable stages of development, and Bruner argued that much new knowledge must be acquired according to this order.



Previous work was either intuitive-but-simple or flexible-but-abstract. Systems like LEGO Mindstorms (above) have many layers of abstraction: playing with the physical bricks doesn't prepare children to understand the programming language.

Topobo (at right) uses *multi-layered abstraction* so children can progress from concrete to abstract without abandoning the tools and ideas they have already mastered.



When Alan Kay developed the Dynabook (with smalltalk, the GUI icons and iconic programming) in the 1970's, he looked to Bruner's ideas to reformulate how ordinary people could leverage computational power [Kay89]. Kay's great insight to replace a computer's linguistic representations (e.g. command line text) with iconic ones (e.g. icons) led to the development of the modern GUI, which made computers more directly understandable and usable by a larger, non-expert audience.

Papert, who worked with Piaget prior to developing LOGO, also incorporated these theories of epistemology in his activities [Pap80]. Papert anthropomorphized a programming language, by making one that is programmed from a first-person perspective and can be related to a child's personal, concrete experiences. LOGO presented children with linguistic programming tools, and teacher-guided design projects introduced abstract concepts to children. For example, children would "walk the turtle" to understand through bodily movement and thinking how a computer controlled turtle could be programmed to behave. LOGO, like Kay's work, also sought to make symbolic manipulations more intuitive and accessible by representing them iconically, in this case with turtle geometry.

Bruner's framework suggests that certain ideas can be made even more accessible, and at a younger age, if they can be grasped and manipulated physically. In this thesis, I show how tangible programming and interaction can provide an enactive mode of interacting with computers, where tangibles provide a bridge from computers' iconic and symbolic representations to enactive ones, and allow for more intuitive expression and access to certain ideas. My hypothesis is that physical, and especially spatial or 3-D problems are best approached first in the tangible domain, where simple behaviors can be prototyped and manipulated tangibly.

Topobo System

Topobo is a 3-D constructive assembly system embedded with *kinetic memory,* the ability to record and playback physical motion. Unique among modeling systems is Topobo's coincident physical input and output behaviors. By snapping together a combination of Passive (static) and Active (motorized) components, people can quickly assemble dynamic biomorphic forms like animals and skeletons with

A Topobo Moose. To program motions, you just manipulate the toy.

Topobo, animate those forms by pushing, pulling, and twisting them, and observe the system repeatedly play back those motions. For example, a dog can be constructed and then taught to gesture and walk by twisting its body and legs. The dog will then repeat those movements and walk repeatedly.

With Topobo, a domain of knowledge that is *enactive* in nature — knowing how to walk — can be leveraged via tangibles for people to learn about how principles of robotic locomotion. Constructivist theories help guide a UI design that balances ease-of-use (the educator's "low floor) with abstraction and flexibility ("high ceiling"). In trying to reach a broad range of users and complexity, I find that where tangibles are intuitive, abstraction permits a certain flexibility of use.

In this thesis I will present a class of related tools and UI approaches for children to approach ideas first through concrete, enactive representations (e.g. tangibles), later through iconic representations (e.g. tangible+visual representations) and last through manipulations characteristic of language. I will argue that while some ideas may generally only be expressible through one domain or the other, tools can be designed to ease learners' transitions from enactive to iconic to symbolic representation of ideas, helping them climb a mountain of ideas.

Thesis Overview

In the following pages, I will show how children can progress through intuitive-but-simple constructions to more flexible-but-abstract ones, using the Topobo system. I believe that children construct knowledge in spiralling fashion, building new theories by returning to ideas and experiences they have already developed. Similarly, Topobo was developed in a spiralling evolution, based on my personal experiences with the system, and largely in response to observations of children using and learning with the toys.

To a degree, my development of the system may be similar to a child's development learning the ideas it embodies, as the creation of Topobo was part of my five year (constructivist) education at the MIT Media Lab. In documenting the work, I will show how other children have made many of the same discoveries I made while developing the system, and how they far surpassed me in their various approaches and insights into a body of knowledge about robotics and animal locomotion.

Ch. 1 Motivation will begin by explaining how my experiences as a sculptor and toy designer inspired and informed Topobo's conception and design. I will then explain how my experiences with educational tools and tangible interfaces informed the system's design, application and development as an actuated modeling material.

Ch. 2 Background and Related Work draws from various fields of study to argue that there is an educational basis for tangible media. This section follows with a description of inspirational robotics research which will explain how work in distributed actuation informed the engineering and design of the system.

Ch. 3 Early Design Studies will overview the development of the basic system and explain why actuated modeling was conceived as a scalable platform to address various applications beyond toy design.

Ch. 4 Topobo System Design describes the core Topobo system, including the design principles, the development of different versions of Topobo, and an explanation of how children used the many different Topobo components. I overview a series of studies with children age 5-14 that established core pedagogical topics that all children appear to address with the system.

Ch. 5 Beyond Record and Play describes the development of the Backpacks, which are tangible modulators for kinetic behavior. Backpacks were developed in response to children's desire to "raise the ceiling" of the system, and I will show how children used them to explore issues like conditional behavior, sensors, variables, and feedback. Backpacks are argued to help children develop a theoretical foundation to transition from Topobo to formal kinematic modeling languages.

Ch. 6 Remix and Robo considers how controllers can support iteration, reflection, and abstract control of kinetic data. I discuss how children used the controllers to support social interactions, engaging in collaborative performances and competitions.

Ch. 7 Topobo in the wild addresses the longitudinal impact of the system, and analyzes how various educators envision a tangible for learning. Here, the teachers are the "users" and we report how educators incorporated Topobo into their museums, classrooms and robotics clubs as part of their day-to-day work.

Ch. 8 Climbing a Mountain of Ideas presents "multi-layered abstraction," a design strategy that allows children to build on (enactive) knowledge they already have, and develop ideas from simple to complex without abandoning the tools they are already using.

Ch. 9 Beyond tangibles will consider limitations of the system, including transitioning from TUI to GUI-based approaches. The general focus will be a discussion of how to balance the physical and digital elements of a digital manipulative.

Ch. 10 The Future of Play will propose design guidelines for tangibles, based on Bruner's framework. I present Communiclay and Protobo as experiments to consider issues that will arise in pursuing a new class of media, Kinetic Materials.

Conclusion summarizes the thesis and presents Topobo as one example of a toy designed to coevolve with children's interests and abilities.

Appendix A describes how we engineered and built the system. This includes, mechanical, electromechanical and electrical engineering as well as a qualitative description of the firmware. I also briefly considers design approaches and technologies to support future actuated modeling systems.

Appendix B documents the Topobo brochure.

1 Motivation

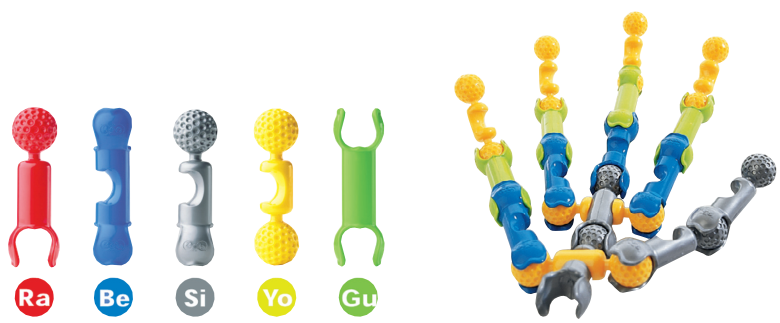
Topobo is a physically manipulable modular robotics system that integrates physically coincident I/O, a constructive assembly system, and distributed computation and control adapted from the modular robotics communities. As such, it has been a technically complex project produced with the invaluable support of a research environment. However, I approached Topobo from an artist's and designer's perspective, with a focus on social interactions and socially constructed meaning rather than as an engineer trying to create an optimized solution for a specific goal. Much of my inspiration comes from experiences and explorations in fine arts, educational toy design, and interactions with tangible interfaces and museum exhibit design. I have maintained an art practice throughout my life, and this section describes the art and design investigations that inspired Topobo and led to its conception. Through my work in museum exhibit design, I will discuss how interactive pedagogical tools have informed the educational approach of the project.

Zoob

The summer between my junior and senior years at Yale, I worked with conceptual artist and sculptor Michael Joaquín Grey to bring dynamic modeling, which was only possible using computers, into physical space with a hands-on tool called ZOOB®. ZOOB is an acronym for Zoology, Ontology, Ontogeny and Botany, and was an idea to create a haptic interface that had the complexity and dynamics of information behavior or living system behavior [Zoo04, Sha02]. Zoob embodies dynamic relationships found in micro and macro systems such as DNA, bones, and the cosmos, and makes their complex interactions accessible and fun.

Zoob was an attempt to create a "spatial language" with a structure (or grammar) that would imply certain types of uses and discoveries. Michael Grey arrived at the project through his work as a sculptor who was trying to develop means to visualize and understand the commonalities between living and information systems. While Zoob was intended to be meaningful to people of all ages, children were an ideal audience because they are curiously developing their own emotional and mental models of the world through working with physical objects [Pia52].

The original Zoob system had 22 primitives, conceptually based on the body's 22 amino acids. I joined Michael and helped develop the conceptual foundation for the system and devise an engineering approach that would allow modern materials and processes to make biological modeling easy for children. By tying the connectivity of the system back to the 5 joints found in the human body, I helped Michael to develop the conceptual foundation for the system and develop its "vocabulary." The resulting "Citroid System" technology introduced 5 Zoob units that can connect to each other in about 20 different ways. We described the Citroid System in the original Zoob Guide:



Zoob has 5 parts that connect in over 20 ways. It is based on protein folding and the joints of the human body.

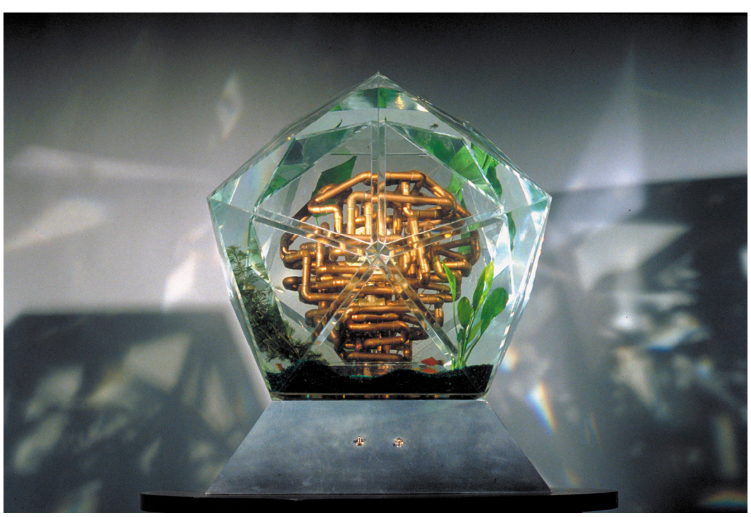
CITROID SYSTEM™ is the organic technology behind ZOOB brand toys. The open-ended, ergonomic design has the potential for a wide array of applications far beyond toys, from complex mathematical modeling to character animation. The CITROID (ball structured with 61-fold symmetry) captures the classic geometries found in nature allowing the articulation of artistic, anatomical and molecular structures. This advanced 3-D operating system, combined with the revolutionary orbit design, connects in over 20 different ways capturing the movement in both Cartesian and polar coordinates. Discover the universal spatial language of the Citroid System!

After graduating from Yale, I worked for several years helping design, produce and market the product, addressing a variety of issues from manufacturing to marketing to visual communication of the system's dynamics via the printed page.

Sculpture and system behavior

I continued with my art practice to explore how interactive systems can use technology to give people insight into the workings of both machines and nature. Some of these art explorations led to Topobo. They include ecological systems, "electronic organisms" and gravity powered walking robots [Raf02].

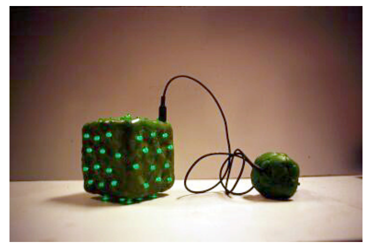
Topobo is intended to be a tool for people to construct and actuate dynamic systems in which many individual elements behave in unison to create a harmonious balance of movement. This idea is thematically similar to my first interactive sculpture, Biosphere (1994), which addressed our culture's intimate relationship to technology. Biosphere is a double walled dodecahedral fish tank with a twisted, heated pipe radiating in its core. The owner of the piece is responsible for maintaining the balance of the ecosystem by regulating the use of the technology that supports it (in this case, the heater). Failure to turn on the heater will cause the fish to die from cold, whereas failure to turn it off will cause the system to self destruct from excessive heat. This living machine is a metaphor for earth with a culture that is precariously reliant on, but not responsible for, technology. By embodying the problem it was about,



Biosphere is an earth metaphor. The owner has to manage the technology that supports the system, or it will die.

Biosphere explored issues of use and social responsibility that stem from the intimate relationships between people and technological systems. It began my investigation into creating a system that could support people's personal explorations of their relationships to animals and machines.

Much as the technology around us is becoming more "intelligent" and autonomous, my art transitioned to reactive and self regulatory, but unbalanced autonomous art works. These projects addressed the role of a person in relation to an autonomous machine. The electronic organisms (2001-2003) were a series of analog electronic



Electronic Organisms like the solar sunflower and Balance Cube investigated ideas of self-sufficience and social context for lifelike machines.

sculptural creations that responded to their environments with both local and global feedback patterns, constantly hovering in the gray areas in between the perfect 1 and 0 of digital electronics. Modeled after single cell aquatic organisms, individual aquatic flora, and floral communities, these creatures respond to a person's presence, touch and ambient interaction through changes in their regulatory mechanisms. For instance, the balance cube will subtly glow in the areas that are most near people or objects, and electronic plants from the Solargarten will avoid people to capture optimum sunlight for their continuing operation. In an attempt to compare synthetic and natural systems, these sculptures explored the interconnected and non-obvious behaviors of analog electronic circuits as dynamic systems capable of mimicking natural systems. They also illuminate the balance between holism (global behaviors) and reductionism (local behaviors) in dynamic systems. Topobo is intended to draw attention to this latter idea through play with coordinated, parallel, kinematic processes.

The Walkers (2002) came out of explorations in passive dynamic robots, a field of robotic research that investigates the implications of geometry on complex motions like bipedal walking. Researchers in passive dynamic walking have shown that gravity-powered walking bipeds, constructed with carefully calculated geometries, can perform natural-looking walking behavior with no sensors or actuators. Passive dynamic walkers are complex inverse pendulums with a minimum of two intersecting oscillations that are stable only when they are walking [Rui04]. From a roboticist's point of view this is interesting because it is an incredibly efficient use of power, and

Walkers inspired Topobo activities in ambulatory locomotion.

uses purely physical "computation" to determine gait and oscillation. I used empirical discovery to understand the workings of these systems and developed novel means for passive dynamic walkers to self-regulate their trajectory on an inclined plane. I learned about the delicate balance between the interrelating oscillations in these mechanically simple, but dynamically complex machines and found it to be an elegant metaphor for living systems. I also remained fascinated with the mechanics of these systems; abstracting bipedal walking is difficult because the inherent dynamics are multidimensional and interconnected. My fascination creating these quirky machines inspired later activities for children to create ambulatory movements with Topobo.

The educational museum experience

My work with the Walkers led me to engineer and design exhibits for the San Francisco Exploratorium. The Exploratorium is a unique educational museum that couples artistic exploration and scientific discovery. The Exploratorium presents hundreds of specially designed exhibits that encourage people to use their own investigations with the exhibits to gain a deeper understanding of the natural world and the scientific method. The Exploratorium was a critical part of my childhood, giving me a love of empirical discovery, an understanding of a machine as a metaphor, and a knowledge of how dynamics could be understood through a person's physical experiences with objects.

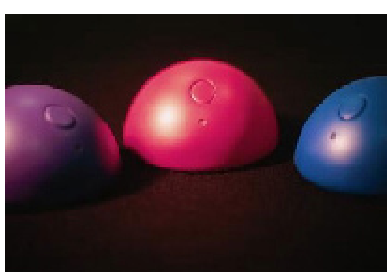
The Exploratorium pioneered what is becoming a popular idea — the interactive science installation — and is a rich source of many exhibits that fill today's children's science museums. Their work informed my motivation to explore concepts like dynamics through physical experimentation with machines and, because the Exploratorium has an *un*impressive history using computers, I was motivated to investigate how computers could be more effectively used in such a capacity.

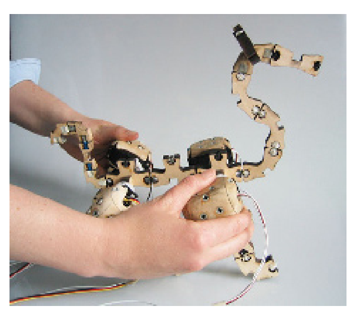
A general deficit in applications of computing technology, evidenced at the Exploratorium and in the limited use of computers in school classrooms everywhere, led me to the Tangible Media Group. Topobo is designed to target both the type of informal learning that takes place in the science museum and to explore alternative modalities for the uses of computers in structured environments like school classrooms.

Tangible media

Phil Frei's *curlybot* embodies playful, empirical discovery. This early TMG project explored how tangible interfaces, with their coincident input and output space, could contribute to the digital manipulative initiative by allowing children as young as four to use physical programming to access computational processes [Fre00]. Curlybot is an autonomous, two-wheeled toy that can record and play back how it has been moved. Every pause, acceleration, and even the shaking in the user's hand, is recorded. Curlybot then repeats that gesture indefinitely, a beautiful and expressive reflection of a person's bodily movements. By seamlessly integrating the physical and digital activities, Curlybot provided an important conceptual foundation for this thesis. Furthermore, my personal experiences playing with Curlybot directly helped inspire the development of Topobo.

Curlybot records and plays back a person's gesture on a flat surface. Topobo makes this idea 3D.





Summary: Actuated Modeling

My experiences with Zoob taught me about building toys and the potential for learning through constructivist play. My childhood explorations at the Exploratorium and my later art practice led me to create tools to understand the behaviors of complex dynamic systems by playing with simplified models of those systems. With the introduction of tangible interfaces, this work motivated me to invent Topobo. Topobo combines physical modeling and computation to create a tool for children and adults to experiment with certain kinds of dynamic systems.

The concept of *actuated modeling* has more generally been intended to help people to physically experiment with, and thus understand, ideas about motion. Physical programming (as with curlybot) presented an opportunity to leverage both the power of programming as a learning activity and motion as a representation of meaning in a computational system.

This thesis looks at one possible application for actuated modeling: to give children a motion modeling toy to learn about dynamic systems like walking robots. As I learned when I was making passive dynamic robots, making robots is fun, but making them walk is a very difficult (and interesting) problem. The evaluation section of the thesis will explore how this activity — creating walking robots — has helped some students begin to understand the roles of balance, leverage and gravity in ambulatory systems. One pedagogical goal of mine has been to support the next generation of thinkers to better appreciate the complexities of animals' movements. Such activities could also help scaffold (support through developmentally appropriate instruction of a parent or teacher) future robotics engineers in developing more beautiful and mobile machines.

2 Background and Related Work
Education, Tangible Media and Robotics

Topobo was designed to help both children and adults learn complex ideas about motion, but this thesis focuses on the child's experience with Topobo for several reasons. Physical manipulatives have an influential role in children's education, and experiences working with physical objects have been shown to be central to a child's emotional and cognitive development [Bro97; Pia76]. Children are already exploring the nature and behavior of the world by interacting with physical tools, and are thus receptive to an open-ended tool like Topobo with which to create metaphors of the natural world.

While tangible interfaces can be successful with people of all ages, an open ended system like Topobo will find a welcome audience in a child's play room or classroom. This section considers background work supporting this idea. I begin by placing Topobo in an educational context, considering the educational implications for physical interactivity and historical trends in educational manipulatives. This educational overview will conclude by looking in more detail at how Topobo contributes to recent work in educational toy design. A review of related robotics research will support the technical conception of the project and the functional aspects of the system design.

An Educational Basis for Tangible Media

Kinesthesia and Learning

Touch is a central aspect of learning, and the study of kinesthesia focuses on the individual's movement and interaction with physical objects as a means of learning. Researchers in education, developmental psychological and cognitive sciences have found that movement occupies a central position in human activity [Lab75] and it is a central feature of early learning [Pia52]. According to Piaget, sensorimotor experience comprises the principal focus of the infant's early knowledge of the world. The advent of symbolic thought occurs when children internalize sensorimotor experience in mental representation. For example, children build speech on prior sensorimotor knowledge [Pia52]. Similarly, scientists who study the brain have shown that physical experience creates especially strong neural pathways in the brain. When people participate in tactile/kinesthetic activity, the two hemispheres of the brain are simultaneously engaged. This type of learning experience helps assure that new information will be retained in long-term memory [Fur75].

Recent evidence supports the further idea of a separate bodily intelligence [Gar83; Joh87]. Children consolidate their development of bodily-gestural skills through play and games [Bru73], and one can think of children's orchestration of a set of motor skills as bodily problem-solving (i.e. skill connotes knowledge). Bodily-kinesthetic intelligence is comprised of two components: masterful coordination of one's body movements and the ability to manipulate objects in a skilled manner [Gar83]. Kinesthetic knowledge provides conscious appreciation of resistance, position and weight of objects. Kinesthetic memory enables a person think about movement by mentally reconstructing muscular effort, movement and position in space. Since the Topobo system — which couples movement, memory and dynamic balance — is a reflection of the child's own kinesthetic knowledge, play with Topobo may support bodily-kinesthetic learning.

Educational Manipulatives

Topobo can be viewed, in part, as a synthesis of the educational toy curlybot, which records and plays back physical motion [Fre00], the biological building toy ZOOB [Zoo04] and the educational software StarLogo that allows children to create software models of distributed systems [Res99]. All of these systems aim to help children learn by building playful models within constraints specific to different processes. They stem from a rich history of educational toys made famous by Frederick Froebel, who invented Kindergarten and a variety of "gifts" (manipulative toys) with which children can learn through play. Although manipulatives are not ubiquitous in formal education, they have a tradition that can be traced back to the 19th century, pioneered by educators such as Pestalozzi, Froebel, Montessori, and Piaget.

Until the 19th century, the core of the educational process was based upon lectures and recitations. At that time, few people believed that young children were capable of being formally educated. One of the first supporters for "hands-on learning" and the education of children was the Swiss educator Johann Heinrich Pestalozzi who claimed that students need to learn through their senses and through physical activity, arguing for "things before words, concrete before abstract" [Pes03].

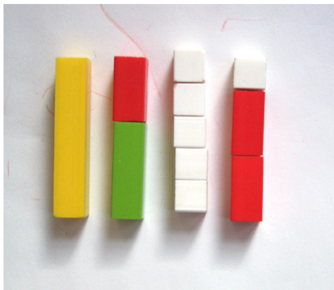
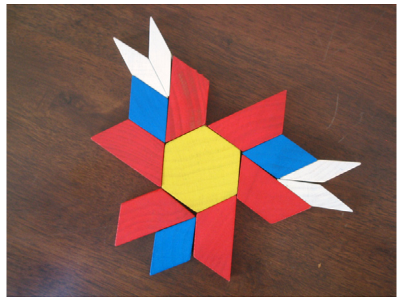
Kindergarten gifts pioneered the use of physical materials to teach children about the common forms and processes in the natural world.

Pestalozzi influenced Friedrich Froebel who created the first kindergarten by the year 1837. Froebel's kindergarten was filled with objects — "the Kindergarten gifts" — for children to use and play. These objects were designed to help children recognize and appreciate the common patterns, shapes and forms found in nature [Bro97].

Maria Montessori received and extended Froebel's practices, and later inspired networks of schools in which manipulative materials play a key role. Montessori tried to develop a framework for an "education of the senses," i.e. materials, objects and learning experiences that help children develop their sensory capabilities, control their own learning process and learn through personal exploration [Mon12].

Epistemology

Piaget developed an epistemological foundation for the educational ideas made successful by practitioners like Froebel and Montessori.

Cuisinaire rods let children experiment with number, equality, and basic algebraic ideas. With pattern blocks, children can explore geometrical principles.

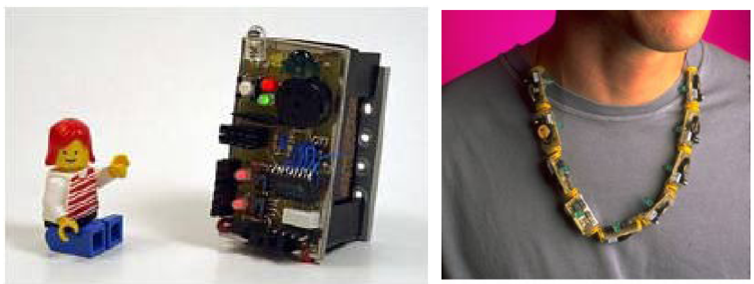
He developed an influential clinical method that focused on close observation of children with the goal of understanding their logic and mental models, and proposed that all children progress through similar "stages of knowledge development." Children construct a particular progression from the concrete to the abstract: they first construct knowledge through "concrete operations" before moving on to "formal operations" [Pia76]. Piaget showed that the physical environment and objects in it have central roles in a child's cognitive development, being a basis for thought and growth.

Seymour Papert, who studied with Piaget before coming to MIT, took Piaget's research into a new direction by using computational tools such as LOGO to reevaluate how concrete operations can open new ways of thinking and learning for children at early stages of development. This perspective gave birth to the constructionist theory of the "child as an epistemologist" who can build his/her own knowledge, and explore the nature of that knowledge, by playing with certain programmable environments [Pap80]. Papert believed that when children engaged in physical construction activities, they were more likely to develop, analyze and understand mental models. Papert's focus on construction systems and the "child as epistemologist" illustrates Papert's particular focus on systems concepts as a basis for both play and learning.

The principles underlying LOGO led to other digital environments and manipulatives designed to engage children in different types of thinking, such as understanding the dynamics of leaderless, rule-based systems. For example, the StarLogo modeling environment was created to give children a tool to model distributed systems like ant colonies that exhibit feedback and emergence, and thus learn about why such systems behave as they do [Res99]. It also encourages an understanding of system dynamics by constructing and observing the behavior of distributed networks. While Topobo does not have the abstraction (and thus conceptual flexibility) of StarLogo, certain types of dynamics and systems concepts are made tangible with Topobo Queens and Backpacks that take advantage of Topobo's physically and digitally embodied parallel processes.

Digital Manipulatives

In an effort to reintroduce tangibility to Papert's vision, Resnick proposed "Digital Manipulatives" that couple digital construction (e.g. programming tools) with physical construction (e.g. blocks). Where wooden blocks allow kids to make towers that fall over, and thus understand static structures and gravity, programmable blocks may



Crickets (programmable brick) and Beads were two early digital manipulatives.

allow kids to understand certain systems concepts. Resnick argues, "children, by playing and building with these new manipulatives, can gain a deeper understanding of how dynamic systems behave…. We expect that digital manipulatives will make [feedback and emergence] accessible to even younger students, enabling students to explore these ideas through direct manipulation of familiar physical objects" [Res99].

Tangible Interfaces

Resnick's original examples of digital manipulatives proposed separate programming and physical activities, where the programming activities were executed via a graphical interface that lacked the kinesthetic affordances of the physical modeling activities. Tangible interfaces' vision of physical computation presented one alternative to this asymmetry, with the goal of making computer programming more intuitive for young children.

Researchers have invented various means for hands-on "programming," and tangible programming models fall into two general categories: those in which the structure of the physical manipulative is a representation of a computational control structure, and those in which it is not. I call the former models "symbolic" after Bruner. The latter "expressive" ones [Mar03] focus more on supporting children's various personal aesthetic explorations.

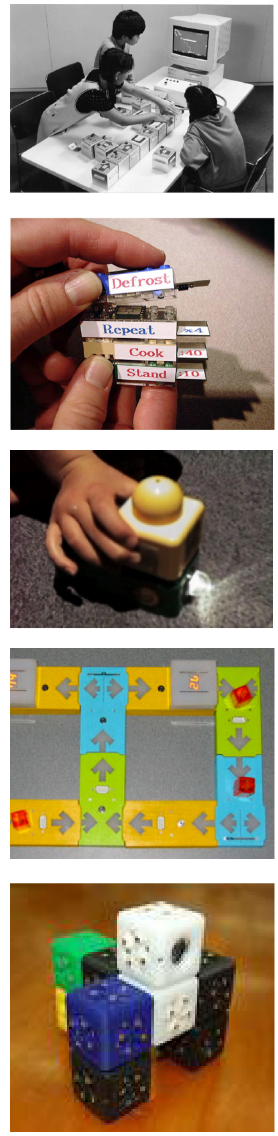
Symbolic tangible programming models

Symbolic tangibles are physical instantiations of mathematical, programming, or dynamic models. For instance, Wyeth's Blocks [Wye02] make simple conditional behaviors tangible through a series of blocks, and Flow-Blocks [Zuc05] make dynamic systems models tangible and manipulable. Such systems make feedback, conditional and other complex system behavior tangible and are developed primarily to help children manipulate abstract ideas. This category also includes projects like AlgoBlocks [Suz93] and many digital construction kits like RoBlocks [Sch06] and Tangible Programming Bricks [Mcn04]. I believe the perspective of these projects is to prioritize children's engagement with a computer or systems concept.

Expressive programming models

Other projects seem to prioritize children's pre-existing action or experience, such as painting, playing with blocks, or dancing, and alter that experience by incorporating a computational element into their existing activity. Many of these systems employ a "record and play" programming model and have been argued to be more experiential in nature and more intuitive for users than other programming paradigms [Ack99, Fre00, Ryo04]. With these systems researchers argues that children can express their own desires, intention and aesthetics in their model, because the structure of the model can reflect a learner's aesthetic desires rather than the symbolic structure of the system. This flexibility has been argued to facilitate learning because people become emotionally engaged with their work and focus on it deeply.

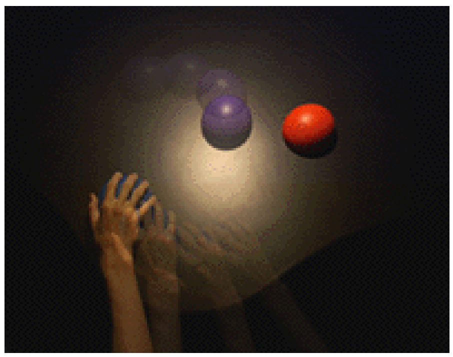
For example, Curlybot coupled input (program) and output (execution) space via programming-by-demonstration [Fre00]. Whereas projects like Logo have successfully allowed children ages 10+ to explore advanced mathematical concepts related to differential geometry, curlybot's physical programming and looping playback were shown to help children as young as four experiment with some of these same ideas through a form of "gestural programming."



From top: AlgoBlcoks, Tangible Programming Bricks, Wyeth's Blocks, FlowBlocks, RoBlocks.

IO Brush is a physical paint brush with embedded video camera that children can use to create static or dynamic paintings. Children "pick up" colors, textures or short video clips of their environment by touching the brush to familiar objects. That "color" is then painted onto a touch screen connected to a computer. Children also used IO brush like a microphone to embed stories in their paintings: touching the mark of the brush would show the origin of the "ink" and reveal the story told while the color was sampled. IO brush was shown to appeal to children with both visual and dramatic [Gar83] learning styles [Ryo04].

Curlybot (left) and IO Brush (right) are both *expressive* interfaces that use a record-and-play programming model.



From Continuous to Discrete

Most expressive tangibles are designed for children to play with continuous ("analog") data sets, and most exploratory tangibles are modelling programming languages that deal with discrete data. The set of operators and operands will be fundamentally different in the two cases, and I am tempted to compare musical performance and composition tools (probably the most expressive and flexible tools to create and manipulate continuous data sets) to symbolic programming languages (the most powerful class of tools to manipulate discrete data). I suspect a child's transition from continuous experience to an ability to describe it with discrete symbols like language or math marks an important developmental transition from enactive to symbolic representations. Our challenge as educational toy designers may be to create computational interfaces that also bridge this spectrum.

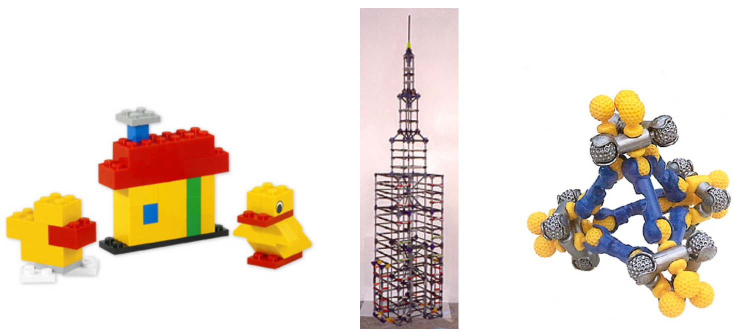
Merging building toys and robotics

This chapter now departs from an investigation of educational theories and tools to describe some of the systems that inspired Topobo.

In spirit and technique, Topobo relates to building toys and certain robotics systems. In fact, my work developing both ZOOB, passive dynamics robots and simple analog control robots directly preceded the conception of Topobo.

Building Toys — Bricks, Sticks and Bones

Manipulatives in general (and construction kits in particular) are often based on building toys. Building toys allow children to explore a certain physical "vocabulary" through physical construction and play and to make certain discoveries through building and experimentation. The popularity of systems like LEGO®, K'Nex®, Lincoln



LEGO bricks stack. K'nex is based on tectonic structures. Zoob is based on biological growth and movement.

Logs® and ZOOB® in toys stores and in classrooms is evidence of our culture's appreciation for educational manipulatives. (Building toys are different from educational manipulatives in only one respect: we expect children to play and learn from them in the absence of an educational structure (classroom, teachers, lesson plans)).

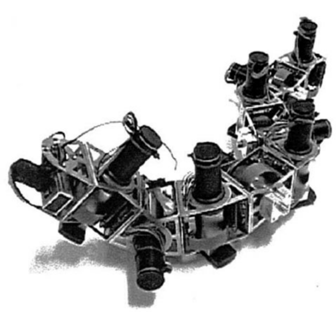
Building toys fall into three general categories: bricks, sticks and bones. Stacking toys (LEGO, unit blocks, Lincoln logs) have two-fold symmetry (up / down). Tectonic toys have 6-fold symmetry (erector, K'Nex). And biological toys (ZOOB, molecular modelling kits) present greater complexity with up to 128-fold symmetry.

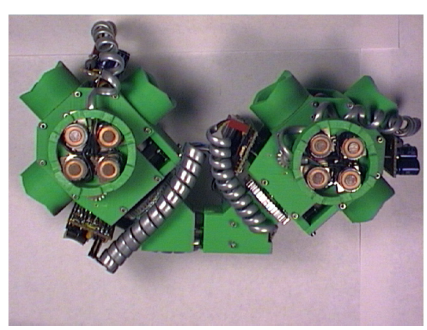
From one perspective, Topobo is a new member of the building toy heritage, and introduces a "biomechanical" paradigm to the class of toys. As discussed in the motivation section, the topology of Topobo's physical modeling system as well as some of its conceptual foundation is inspired by the design and dynamics of the ZOOB building toy, which is based on the movement of skeletons and the folding of proteins [Zoo04]. Zoob addressed how modeling and reflexive investigation with a non-computational toy can help people understand dynamic systems. Zoob is very easy to use, and with only five different shaped parts, the system can scale to represent thousands of different kinds of creations. This dual simplicity and complexity helped inspire the physical and interaction design for Topobo. While Topobo lacks the spatial flexibility of Zoob, the system complements a "biological building" activity by also modeling a structure's dynamic motion.

Topobo also facilitates explorations in topology in a different manner than Zoob. While ZOOB was intended to convey some aspects of the nonlinear nature of information behavior, it does not make information behavior manipulable. Topobo is designed to make certain systems concepts more clear with the Queens, Backpacks, Robo and Remix. These components give children a tool to explore how information can change in a nonlinear system and how simple changes can lead to familiar results (in this case, familiar forms and movements).

Modular, Self-reconfigurable Robots

In order to embed and distribute Topobo's computation and control into the physical building system, we drew from state of the art robotics research and development. Researchers in modular robotics have been working to make a generalized robotic node that can be used to configure robots of varying forms and behaviors. Projects like "Real Molecule" [Kot99] and "PolyBot" [Yim00] draw inspiration from natural systems and provided valuable examples for Topobo's distributed electronics design. While Topobo is not intended to be self-reconfiguring, it is a modular robotic system and thus requires specific design approaches that support modularity such as distributed, scalable sensing and control. However, it is important to note that modular robotic precedents differ markedly from Topobo in intent: reconfigurable robots generally aim to be completely autonomous "smart" machines capable of doing tasks that people can not





Polybot and RealMolecule are biologically-inspired modular robotic systems. They are similar to Topobo.

do, or do not want to do. Topobo is designed to be a medium for thinking that encourages creativity, discovery and learning through active experimentation with the system. This difference is evident in analyzing the design criteria of the systems. For instance, Topobo does not need to have the high degrees of accuracy necessary to create a self reconfiguring robot, nor does the system need to be aware of its own geometry. Conversely, modular robots do not need to be ergonomic nor do they need an intuitive interface for users of the system.

The creators of PolyBot patented several modular toy robot designs that use programming by demonstration for data input [Duf98]. These patents describe several similar systems to Topobo, but the prototypes were never fully designed and implemented as a toy nor were they formally evaluated [Raffle, personal communication]. Furthermore, these systems use centralized control even when they function independently of a PC [Duf98]. Decentralized control — and thus, both physical and computational modularity — was an important design criteria for Topobo and is a unique contribution to a modular robotic toy.

Programming by Demonstration

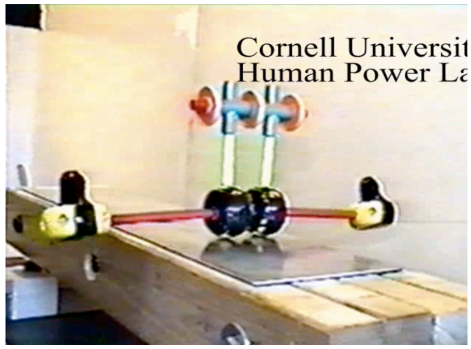
Topobo uses robotic "programming by demonstration" to make the programming activity physical. Other, earlier precedents for robotic programming by demonstration are prevalent in the robotics communities. Researchers in robotic artificial intelligence have for some time used techniques of programming by demonstration to input motions in multiple degrees of freedom. For instance, with the help of a human hand a robot can be taught to pick up a cup [Col98]. Similarly, in manufacturing, an assembly line robot is sometimes physically given endpoints for its trajectory and is then allowed to calculate the optimal path between points. If there are obstacles for the robot to avoid, additional points can be added to obtain the desired trajectory [Tan79]. Like Topobo, these systems use physical input for motion data, sometimes called "physical programming."

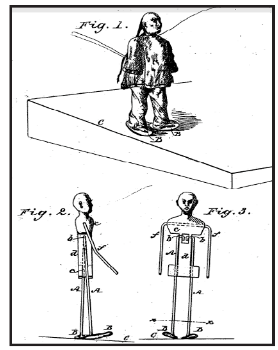
Passive Dynamic Robots

As I mentioned in the Motivation section, walking robots constructed with Topobo share physical simplicity and local-global dynamics that have been explored by researchers in passive dynamic robots.

Researchers in passive dynamic robots aim to deduce the physically elegant designs that can lead to walking robots that require minimal energy input [Col98; Rui04]. Like some Topobo walking creations, these robots combine falling and inverse-pendulum dynamics that are prevalent in ambulatory systems.

This tinkertoy passive dynamic robot is similar to the 19c walking toy. The difference is that psssive dynamic robots are only stable when they are moving.





3 Early Design Studies

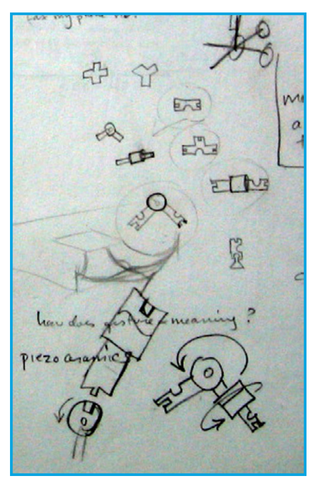
I originally conceived and designed Topobo to apply to several different applications including educational toys and computer-assisted modeling of physical surface meshes. This section will overview the conceptual framework for Topobo and some decisions that led to the current system design.

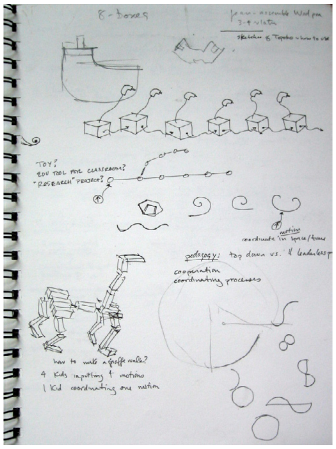
A Tangible Language

My early studies for Topobo sought to sense and actuate a modeling system with the flexibility of the Zoob system. In addressing the fundamental question, "What is the meaning of motion?" I tried to develop a system that could represent dynamic structures from multiple scales. At the body scale, Topobo might give people insight into the dynamics of the body's movement, while at the microscopic scale, representing proteins with Topobo might help people to understand the nature of quaternary protein interactions. Mesh modeling might similarly help people visualize nonlinear surfaces used to visualize environmental or planetary dynamics.

A single system that can model a wide variety of ideas, forms and motions might be described as one quality of a "spatial language" for form and motion, and I will present Topobo as one step in this direction. Topobo is, however, a very limited spatial language due to the absence of suitable 3 degree of freedom (DOF) actuators. This technological deficit led me, at an early stage, to scale back my original domain goals for Topobo and the system does not address applications related to protein modeling. Topobo was thus designed to be a building system to model the shape and movements of things at the body and environmental scales.

In my original designs, all connections were actuated. The parts were shaped like a finger.





Design studies investigated how motors controlled by a peer to peer computer network could allow children to discover natural patterns like waves, spirals, and walking.

Coincident I/O

Coincident I/O was both an end and a means. As an end, it would allow people to directly animate their creations, leading to a sort of magical construction kit where kids could build animals and then physically teach them how to walk (and the animals would then do it by themselves!). For mesh modeling, coincident I/O was a means both to prove tight sensing and control feedback loops and a technique to facilitate tangible interaction designs. Therefore, we developed Topobo as a two stage process, in which a toy would be developed first, and then that toy would be constructed into spatial meshes that could both be physically manipulated and controlled by a computer.

A principal quality of building systems, like natural language, is that they are modular and distributed; each piece is complete and autonomous, but becomes something more interesting and complex as it is combined with other pieces. In like spirit, Topobo was conceived to be a physically and technologically distributed system, a robotic assembly kit that lacked a central "brain," in favor of something more like distributed reflexes and muscle memory.

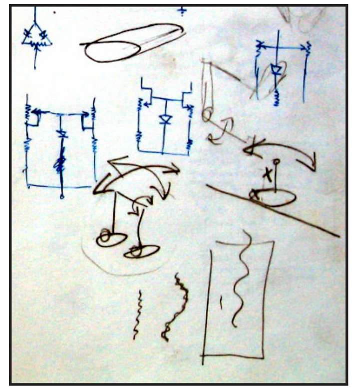
Sensor-Network Architecture

Topobo incorporates a sensor-network architecture in which all Actives in a structure create an ad-hoc network that permits any network topology. This approach is resilient to the myriad configurations children will create: nodes connected to themselves, nodes connected in loops, knots, trees, nodes disconnected and reconnected during communications. While a sensor network architecture was more difficult to implement than a standard network protocol like RS485 or CAN, it had the benefit of robustness and a theoretical connection to Topobo's foundation in decentralized biological systems.

Each Topobo node functions as a network router, and communicate with their nearest neighbors using a custom bit-bang protocol without knowledge of any node IDs. This required that all Topobo algorithms – which come from computer science investigations into modelling physical and biological phenomena – take advantage of decentralized structures.

Rotary Actuators

A variety of modern actuators are being developed to simulate the motions of biological structures. Reports of "polymer muscles" and "muscle wire" fill popular scientific literature. However, many of these actuators are in early stages of development and, because they are inefficient, difficult to obtain, or difficult to use, were not good choices for implementing Topobo. In the end, I found (as many contemporary engineers do) that electric motors are the most efficient, affordable and readily available actuators.

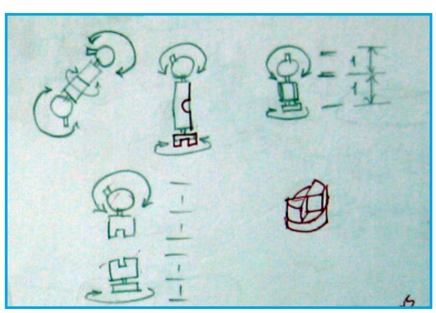


Early Topobo studies focused on similarities between passive dynamic walking and electronic circuit feedback.

I carefully considered the application of rotary motion to a biological modeling system because one finds almost no examples of rotational actuation for locomotion in the natural world (the only exception being a certain type of microscopic flagella). Other possible motions for actuated modeling included linear motions and oscillating rotations. Although linear actuation is beneficial for many types of mesh modeling, we did not use it because linear actuators have a propensity to fail after repeated use. To simplify our mechanical engineering overhead, early designs aimed to describe skeletal and mesh modeling with oscillating, rotary motion as a kinetic constraint. I began my studies with the assumption that I would use a direct-drive mechanism for actuation rather than an arrangement of linkages or tensile and compressive members. While the latter is a popular approach for representing muscles and bones, I chose direct drive in order for users to focus on the complexities of motion rather than on the mechanics of actuation.

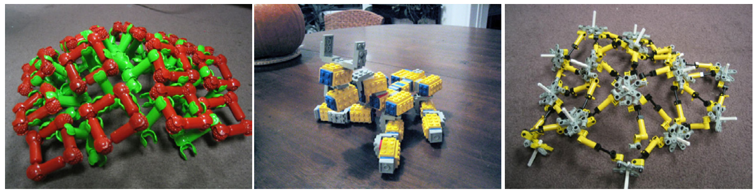
Physical Asymmetry

My early studies included dozens of sketches and models of modular,



We used hobby servos. Because of their shape, I abandoned my original design.

meshing systems. I used various materials including Zoob units, cut and glued LEGO bricks, cardboard and tape. Many of these studies



Mechanical studies revealed that an asymmetrical actuator is versatile.

involved creating chains of actuators between mesh "nodes" that could be grabbed and physically manipulated. Such chains had the problem that joints would not always bend in the right order, so that splines would occasionally get stuck or turn "inside out" when nodes were grabbed and manipulated. This led to my development of an asymmetrical actuator that could be configured to be a lever arm of varying lengths. Using actuators of varying lengths allows the builder to design inter-nodal splines in which a certain actuator, due to its longer length, could be the first in the chain to bend.

Some Limitations of Physical Input

Much of my design studies revolved around the need to accommodate and sense physical input. Input requires sensors, and an actuated assembly system needs to accommodate being mechanically manipulated. Generally speaking, actuators are designed for output only, making coincident i/o difficult.

There are two obvious ways to create coincident i/o with motors. The first is to back-drive a motor, sense the motions of the motor and then recreate those movements during playback. This is how curlybot works [Fre00]. This approach is difficult because motors usually have gearboxes at their output to reduce the motor's speed and increase its torque, and gearboxes are not designed to be driven backwards. Curlybot got around this problem by using very large, strong motors and a very minimal gear reduction. However, Topobo would require much more strength to weight than curlybot, since it needed to compete with gravity. This would require a larger gearbox that provided enough reduction to output decent levels of torque from a small motor, but not so much reduction that the gearbox could not be back-driven.

A second approach is to mechanically decouple the input and output. This can be done by using a slip clutch at the output of the gearbox. During input, the computer senses the movement of the clutch. During output, the motor repeats the movements of the clutch by driving its shaft through a series of gears. Some benefits to this approach are that a stronger gearbox can be used (high reduction gear boxes cannot be mechanically back-driven without destroying them) and that different kinds of gears, such as harmonic or worm drive, can be used. One drawback is that the motor may not be able to reproduce some input motions if it is stronger than the clutch, because the clutch could continuously slip during playback.

In the end I chose the first approach, to back-drive a gearbox, because it required no custom, precision mechanical part design, and was simpler: all input motions are mechanically identical to all output motions.

Developing a System Design

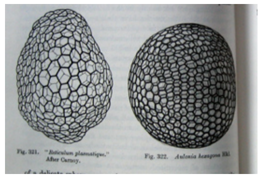
Actives and Passives — An Alphabet

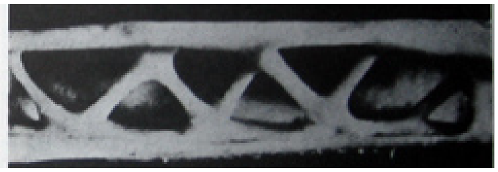
In my early sketches for Topobo, all connectors were actuated and cylindrical, meant to be like one's fingers. However, after researching actuators I had to adopt a different design. The best actuators I could readily find were modern hobby servos. They combine an absolute position sensor, a back drivable gear train, a miniature DC motor and a drive circuit, and from an electromechanical perspective they are functionally complete. However, hobby servos have a limited range of form factors and tend to be square shaped, unlike my sketches of cylindrical parts. Therefore, I developed a system of "Active" and "Passive" pieces that would allow a user to build various branching structures, adding Actives where they needed actuation. As well as being convenient, this approach was also more forgiving than my original sketches because the actuators need not be as small, light, powerful and individually ergonomic as a design in which every element is motorized.

Branching and Spatial Geometry

In order to allow a variety of forms to be built and to provide strength to larger forms, the Topobo Passives embody a branching geometry that is inspired by nature. In nature, one finds a few types of branching structures. One is like a tree, in which branches extend from a common trunk. Another is like a spider's web or the inside of a bird's wing, in which many branches interconnect in spatial loops [Tho42].

Radiolarian skeletons and the inside of a bird’s wing show how Isometric building blocks will grow into spatial meshes and loops to create strong structures.



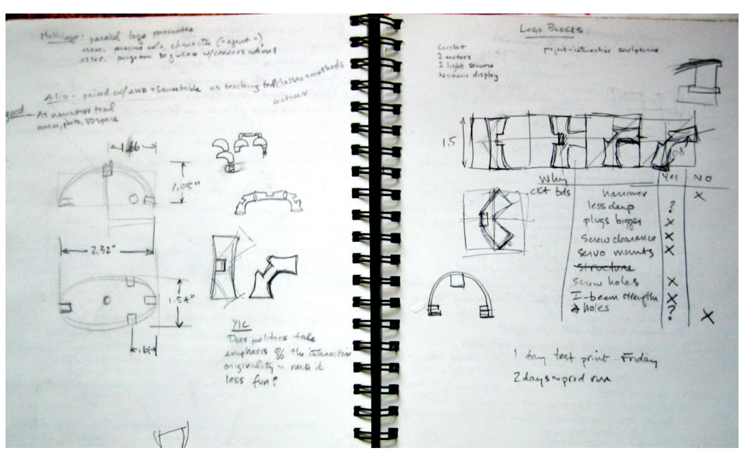


Branching structures like trees are stable because they change scale with growth, growing thickest at their trunks and becoming thinner towards new growth. One approach to an actuated modeling system might be to provide different scales and strengths of actuators, in the manner of a branching tree.

By contrast, smaller scale isomeric structures use spatial looping and weaving of structural members to achieve strength. For instance, the inside of a bird’s hollow wing reveals how spatially distributed structural members can create a strong and flexible bone. This approach was better suited to my designs for Topobo, since I wanted to limit the number of different sizes and shapes of parts in the building system.

Crystals and Loops — Strength and Flexibility

Spatial loops are difficult to create with a system that is limited to rotary actuation. Joints in a ring will tend not to be coplanar, and will therefore bend in unpredictable ways, if they bend at all. A strict geometry that only allows people to build intersecting planes would result in successful actuated loops, but not in "solid" structures. My geometry studies led to investigations of crystalline forms, and especially to crystals that might change shape. Tetrahedral arrangements can be very rigid (e.g. diamonds are hard), but cubic crystals can allow some deformations along different axises. Cubic



crystals are also fairly easy to visualize, compared to other crystal packing geometries. The Topobo geometry is thus primarily based on cubic crystals that allow people to be successful creating flexible, spatial loops, but also includes a tetrahedral element that allows pentagonal and tetrahedral forms to be integrated into more complex models.

Notches: Economy of Form

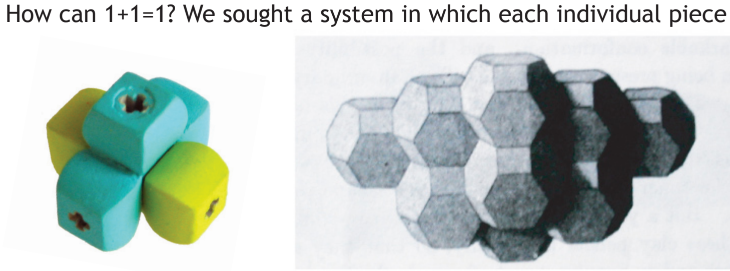
At the metaphoric (and literal) center of my later designs is a universal, hermaphroditic notch that allows different types of structures to be built. The notch allows people to easily move from flat to 3-D structures. The approach is also economical — 8 notched parts can combine in different ways to create 36 different secondary passive pieces. For example, two straight pieces are notched to make a "+" or two T's can be notched to create a Cartesian 3-axis intersection. Assembling notches can also inform more sophisticated use of Actives when playing with Topobo. Assembling flat pieces to create a three dimensional piece is thematically similar to building 3-D movement with 2-D motors. Notches may thus help people learn how to assemble 1-DOF actuators to make a 2-DOF creation. I will describe this process in more detail with my explanation of the interaction design of Topobo in the next section.

Notches allow passives to combine to form secondary parts. Eight notched passives can create 36 different shapes.

An Aesthetics of Unity — 1+1=1

The original proof of concept aestheticized bones to make the crystalline geometry of the Passives feel and look more fluid than its boxy underpinnings. The Actives, conceived as finger-shaped objects, evolved into egg-shaped ones in order to accommodate servo technologies. The resulting proof of concept had contrasting voluminous Actives and planar, flat Passives. In explorations to make Passives and Actives complement each other the Passives were redesigned with a more volumetric quality.

How can 1+1=1? We sought a system in which each individual piece



The system geometry is based on cubic and tetrahedral crystals. Here, Topobo is compared to rhombic dodecahedral crystals.

looked "complete" but parts could assemble into a form that had its own holistic identity. Through dozens of iterations, we arrived at a design based on Brancusi's *Endless Column* which is at once segmented and unified. Brancusi's forms were adapted to the system geometry, and highlighted the Cartesian nature of the LEGO connectors. The Active was then redesigned to incorporate this aesthetic, and look consistent with the Passives.

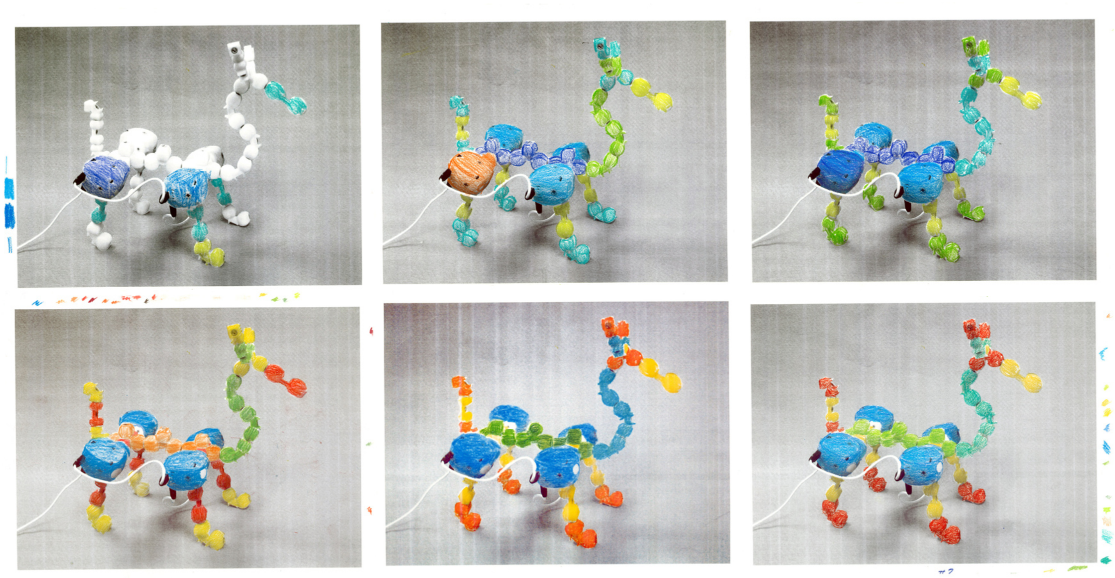
Color — A Visual Language

Color study paintings led to dozens of pencilled-in photos of white topobo creatures. Through this process, we arrived at a color-coded system pallette.

We sought a color palette that was *gender-neutral*, *mulit-age*, and *modular*: We wanted to reach girls as much as boys even though boys are the typical audience of both engineering and building toy domains. As the goal of the system is to evolve with children as they grow, the colors must not look "infant" or even "young" (e.g. no primary colors). A modular palette allows multi-colored parts to come together to a cohesive looking creation.

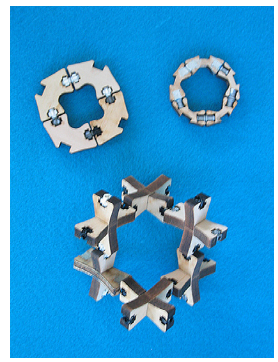
Palettes of monochrome hues or duotone color hues are associated both with adult tastes (e.g. not too young), and with modularity. Cool palettes of blues and greens, painted on paper, provided a foundation because warm palettes (i.e. red and orange) are more common in "boy toys." We added a warm accent color so that children could selectively provide contrast in their models.

Some Topobo passives are similar shapes, so we chose to color-code the system, assigning a single color to each shaped piece.



This would help children locate parts more easily. Achieving color balance required carefully associating specific colors with specific shaped parts, because color density was affected both by the specific color hue, and the size of the part it was applied to. For example, red would look "more red" on a large T shaped piece than on a small tetra.

Color palettes were prototyped by photographing two Topobo creatures children designed in our studies (a Griffon and Moose) that were built entirely of white Topobo prototypes. These photographs of "white" creatures were then printed, photocopied and colored in with pencils, markers and paints. We colored in over fifty different animals, testing color palettes by assigning colors to specific shaped parts in the creation. To fully "test" a palette, both creatures would be colored, and the resulting drawings compared. Through this process, we settled on a final palette for the system.

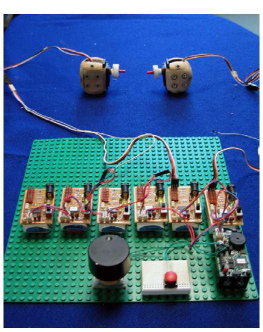


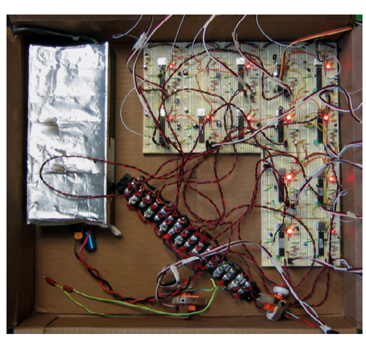
The first generation of parts were laser cut from bass wood and outfitted with LEGO connectors.

Building a Proof of Concept

Early design studies led to a proof of concept using Cricket microcontrollers [Log04] and servo motors. The Cricket prototype was extremely fast to implement and allowed me to experiment with the capabilities of the early system. An inefficiency in the Cricket firmware required that all servos operate synchronously with a single input, and inspired the existing "Queen" functionality.

The first scalable prototype followed, made with wood passives and hand-carved wooden shells encasing hobby servos. Breadboarded electronics tested our peer-to-peer network and parallel processing architecture, and the prototype facilitated early evaluations of Topobo with kindergartners and second graders. These students helped guide the design of the current system.

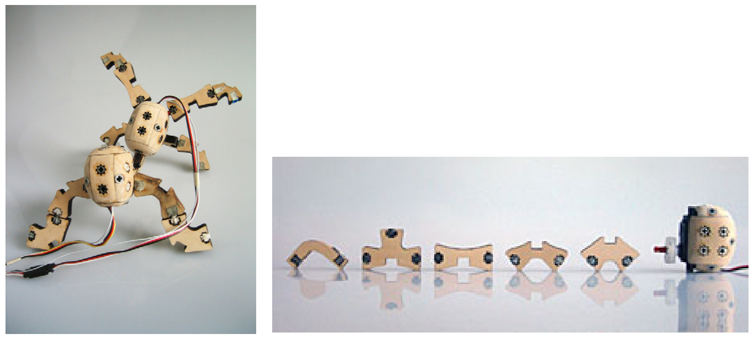




We built two prototypes to test the system design and concept. The cricket prototype tested the Queen and the breadboarded version tested a scalable, distributed electronic architecutre.

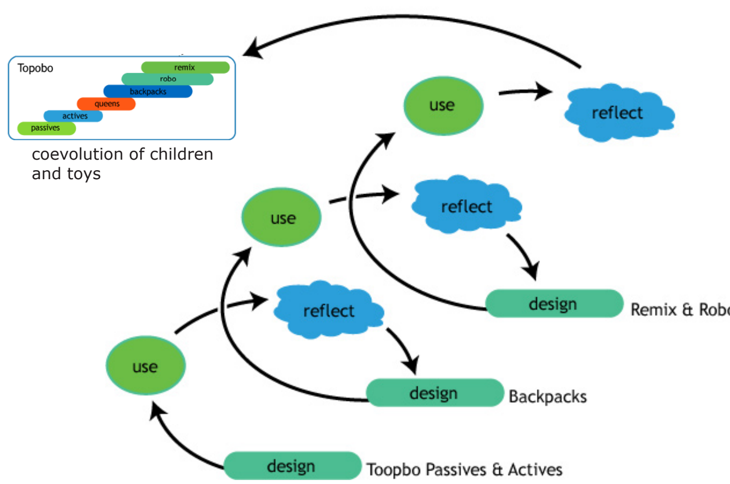
A Spiralling Design Evolution

Following the proof of concept, Topobo developed in roughly three stages over five years of active R&D. Evolution and refinements were all inspired by children's work and critique of the toys. In the next three chapters, I will discuss this evolution. First, I will summarize how the original Passives, Actives and Queens were developed, and how older children's critiques of their shortcomings inspired the de-



The original prototypes were laser cut wooden parts and servos encased in wooden shells. LEGO pins were used to connect the pieces.

velopment of Backpacks. I will then address how the Backpacks were designed and how they were intended to support children's abstraction and more advanced conceptualization of robotics principles. Remix and Robo are then presented as an effort to use children's social structures and play patterns to motivate both personal and collaborative endeavors with Topobo, and how providing tangible controllers can motivate learners to focus more deeply on the core principles that underlie the core system itself.

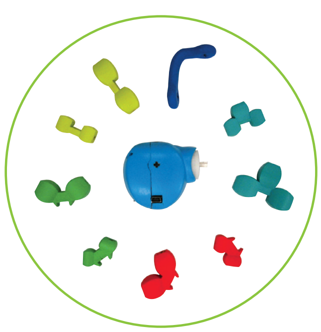


The Topobo system evolved from a spiralling design cycle.

4 Topobo
A Constructive Assembly System with Kinetic Memory

Topobo is a 3-D constructive assembly system embedded with kinetic memory, the ability to record and playback physical motion. Unique among modeling systems is Topobo's coincident physical input and output behaviors. By snapping together a combination of Passive (static) and Active (motorized) components, people can quickly assemble dynamic biomorphic forms like animals and skeletons with Topobo, animate those forms by pushing, pulling, and twisting them, and observe the system repeatedly play back those motions. For example, a dog can be constructed and then taught to gesture and walk by twisting its body and legs. The dog will then repeat those movements and walk repeatedly.

This chapter documents the development of the Topobo system. Our evaluation of Topobo in classrooms with children ages 5-13 suggests that children develop affective relationships with Topobo creations and that their experimentation with Topobo allows them to learn about movement and animal locomotion through comparisons of their creations to their own bodies. Eighth grade science students' abilities to quickly develop various types of walking robots suggests that a tan-



Topobo's 10 primitives combine in many ways to allow people to explore kinetic systems like this one-Active walker.

gible interface can support understanding how balance, leverage and gravity affect moving structures because the interface itself responds to the forces of nature that constrain such systems.

Design Principles

Topobo was designed to retain the best qualities of existing manipulative materials while giving the material a new identity — an identity that can both reveal new patterns and processes to children, and that allows children to creatively express patterns and processes that can not be expressed with existing materials. To achieve this goal, we established 7 design principles:

Be accessible, yet sophisticated — be ergonomic and intuitive for very young children, but support growth across multiple cognitive levels and into adulthood.

Be meaningful even if the power is turned off — technology should add to a toy, without sacrificing the good qualities inherent to its class of toys.

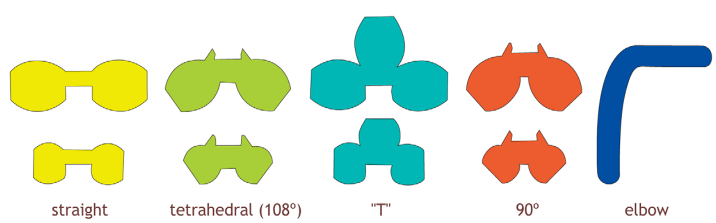
Be expressive — Design multifunction parts that give people latitude for their own personal explorations.

Support exploration of specific ideas — Make certain ideas salient so that people explore them through their activities.

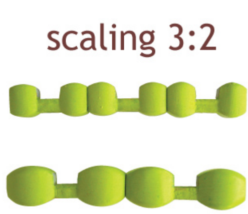
Engage multiple senses — engage sight, sound, and touch to provide rich, memorable interactions.

Be scalable — In the spirit of a modular system, every individual component should be physically and computationally complete and extensible.

Be robust — have a design that would not break or malfunction so that children don't fear making "mistakes."



The passives come in two sizes with a 3:2 scale ratio that is based on the fibonacci ratio found in natural structures like plants and skeletons.



Topobo in Brief

Topobo is comprised of 10 different primitives that are connected with LEGO Technics® connectors. Nine of these primitives are called "Passive" because they form static connections. One "Active" primitive is built with a motor and electronics. The motorized components are the only ones that move, so the system is able to faithfully record and replay every dynamic manipulation to a structure. Queens are Actives that allow for centralized programming. Backpacks, Robo and Remix are computationally "active" but physically passive. They can be used to control, manipulate and play with gestural programs.

Passives

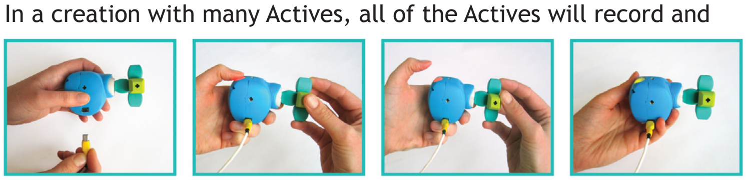
We designed nine different Passives to allow a variety of physical structures to be built. Since Topobo is intended to model various natural forms like skeletons and interlacing meshes, the system allows branching and spatial looping. The Topobo geometry is based on cubic and tetrahedral crystals.

The "elbow" (offset 90°) comes in one size. The "straight," "T," "L" (90°), and "tetra" (108°) shapes come in two sizes with a scale ratio 2:3, based on the Fibonacci ratio that describes scaling in growing systems like mammalian skeletons. These latter 8 pieces are bisected by hermaphroditic notches, allowing any two pieces to connect and branch at a right angle. For example, two straight pieces will form a "+" shape, or two tetras will form a tetrahedron. This arrangement allows the formation of regular meshes like a silicon tetrahedral lattice or simple forms like a pentagon or square. Children notice this regularity quickly because when a child tries to build large, interconnected forms, pieces often fit together.

Actives

The Actives are motorized, networkable, egg-shaped plastic objects with a button and an LED for indicating whether the system is in record (red) or playback (green) mode. To record a movement, the user presses a button on an Active, twists and moves the Active to program a sequence of behaviors, and then presses the button again. The Active immediately goes into playback mode, which repeatedly replays the user's input until the button is pressed a third time, which makes the active stop moving.

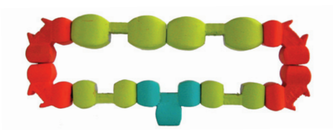
In a creation with many Actives, all of the Actives will record and



plug in the active | press the button to record | turn the axis with a motion | press the button for playback

Programming an Active

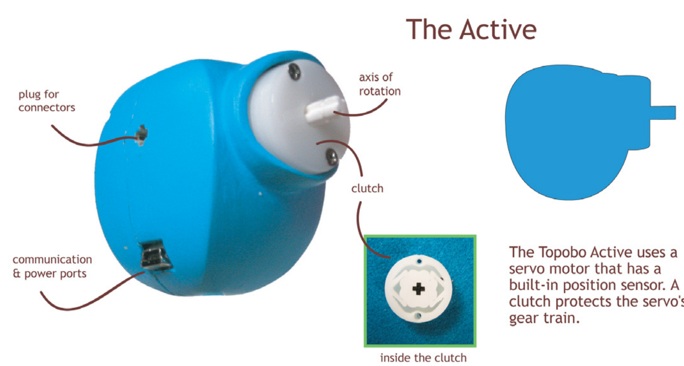
playback at the same time. For example, if a child makes a circular ring of Actives, pressing a button on one of the Actives then sets all of the Actives in the structure to be in recording mode. The child may then move the circular structure of actives in the manner of a tank tread rolling across the floor, and then press any one of the Actives' buttons to set the structure into playback mode. At that moment, the motion that each of the Actives remembers is their local motion, despite the fact that the child has manipulated the global structure. In playback mode, the Actives mimic their local behaviors inspiring the whole system to take on the global motion imparted to it by the child.



Because of the system's geometrical design, when a child builds large interconnected structures, pieces often fit together.

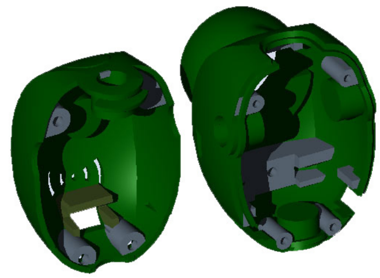
The Active is made of a servo motor and electronics in a plastic housing. The housing has 6 points of mechanical connection, three sockets to connect power/communication cables and a button that is backlit by a red-green LED. One of the mechanical connectors is connected to the output shaft of the servo motor and rotates 170°. On board custom electronics handle power distribution, memory and processing, and peer-to-peer, multichannel serial communications. Each Active is identical and autonomous, and only needs power to function.

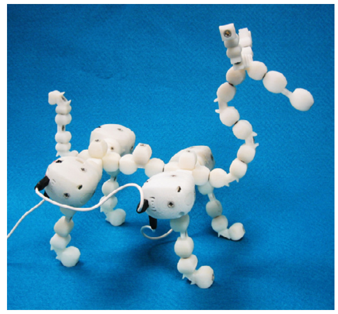
The one-button interface was inspired by Curlybot [Fre00] and



The Active design accomodates multiple electrical and mechanical connections. A clutch protects the motor from excessive torque.

chosen because it is extremely easy to use. While the one-button interface is limited, 3-d motion concepts are complex and the immediacy of the interface design encourages rapid experimentation with motion. Physical programming by example also results in natural looking, emotionally engaging motions because they are the reflection of the user's own body movements [Fre00].





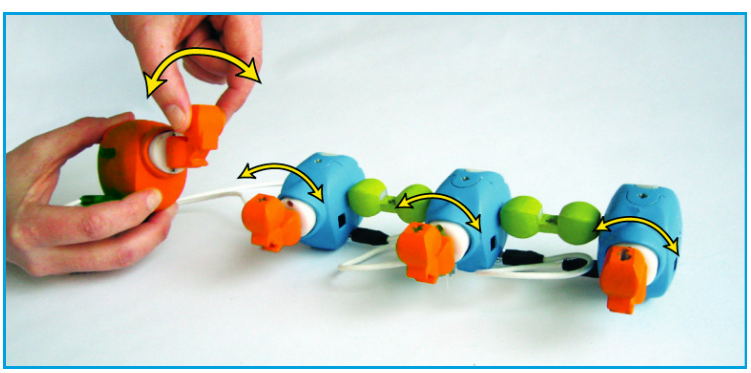
Second generation parts were modeled with 3D CAD software and 3D printed in ABS plastic. This allowed us to house the electronics and motor in a single case. Parts were later mass produced based on this design.

Centralized Control

In recording mode, a user will grasp and wiggle an individual Active component in a creation. In playback mode, that same Active component will mimic the motion that was made to it. The other Actives in the structure have no motion to mimic. In some situations, it may be desirable for all Actives in a structure to mimic the motions made to one individual Active in the structure. To accommodate this complexity, we introduced the Queen. In both recording and playback modes, all motions of the Queen are imparted directly to all Actives connected to the Queen.

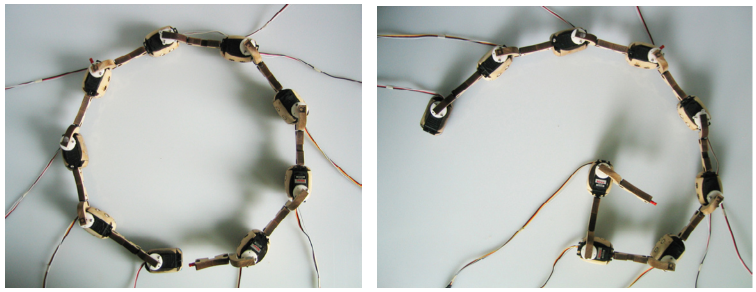
For example, suppose that one constructs a linear structure of actives with a Queen at one end. When the Queen is recording, all of the other Actives will mimic its angular position. Thus, increasing rotations to the Queen cause the entire structure to begin to curl into a circular form. Eventually, the ends will touch.

Topobo Queens can be used to provide tangible examples of spatial translation. For example, two facing Actives that have identical motions will appear to have mirrored motions if their output shafts are facing each other. This can be used to construct scissor-like motions in a walking animal.



Programming with a Queen:
In both record and playback modes, all motions of the Queen are imparted directly to all Actives connected to the Queen.

With a Queen, the Bigger/ Smaller Backpack can lead to familiar forms like the equiangular spiral that is found in snail shells and sunflowers.



A Queen does not need to be mechanically attached to the creation it is programming, so it can also be used as a remote controller. Remote programming with a Queen gives a child synchronous input and output feedback during programming, allowing the child to observe their creation's motion while they are composing it.

In this original system, we imagined simple extensions of the function of the Queen to enable dramatically different behaviors of structures of Actives. To generate these behaviors, we have utilized three different types of "augmented" Queens. The first of these augmented Queens is the Decay Queen. A sequence of Actives connected to the Decay Queen is endowed with a knowledge of how many steps away from the Queen it is. An active will then scale the Queen's motion by a factor which is proportional to this number of steps. Using a Decay Queen, a linear string of Actives can gradually curl into a spiral. Actives connected to the Time Delay Queen mimics the action of the Queen following a temporal delay that is proportional to the number of steps away from Queen that an Active is located. Using a Time Delay Queen, linear strings of Actives can move with wave-like motions. Finally, the Faster/Slower Queen speeds up or slows down Actives as a function of steps away from the Queen. Due to Topobo's looping playback, a linear string of parts can exhibit harmonic resonance patterns. The Augmented Queens were never tested with children, but the ideas them embody inspired the later development of Backpacks.

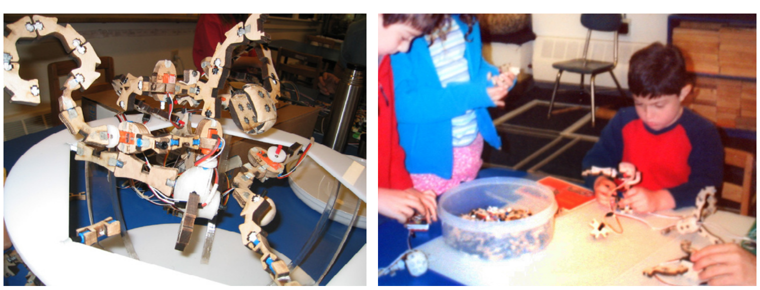
Evaluations with Children

We conducted classroom studies with 25 kindergartners (5-6 years old), 22 second graders, and 32 eighth graders to evaluate Topobo's effectiveness as a educational tool for children at various educational levels.

Kindergarten and Second Grade Studies:
We spent three hours each in a second grade and a kindergarten class playing with an early Topobo prototype, evaluating its technical features, design principles and our educational goals. These classrooms featured many examples of models, toys and manipulative materials. While older school children (who are more adept with abstract manipulation) routinely use a computer lab, these classrooms had only one computer each, and it was strictly for teacher use. Two researchers worked with several groups of approximately 4-5 kids. We started by showing children two possible models and how they could manipulate them. Then we assisted them with assembling and programming their own models.

We introduced Topobo to the second grade group by comparing a walking creation to ourselves walking. When Dave, a normally impatient child, came to one of the tables where we were sitting and manipulating Topobo, he immediately became engaged. First, Dave started to manipulate and rearrange the parts in spontaneous and creative ways but Topobo soon became part of his ongoing activity and experience. Dave was working to create his own walking animal with a Queen. When something stopped functioning as he had expected, Dave drew on the earlier models that we showed him, and tried to emulate some of the configurations, especially the local-global interaction and the feedback between parts. He was trying to run a new creation, but suddenly he realized that the creation didn't work as he has planned. He broke his focus, stopped his ongoing activity and then asked: Why? What happened? Why it is not walking?

This breakdown in the ongoing activity of building a Topobo model may have produced a certain conceptualization in Dave's mind [Bød95; Flo86]: he may have started thinking and manipulating Topobo in new ways in order to produce movement, feedback, global-local interaction and walking. The process of physically de-



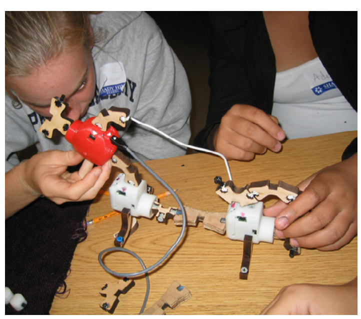
A second grade collaborative creation, and a case study with "Dave" who was trying to create a walking animal.

bugging his creation may have given Dave new insights to kinematic systems.

Dave played with Topobo for over 45 minutes. Our guiding and scaffolding certainly helped him to quickly create and test Topobo models, and it may have helped him to remain engaged for such a long time. In the future, teacher guiding may be very helpful for facilitating in-depth conceptualization and kinematics thinking by comparing Topobo to natural locomotion. For children such as Dave, Topobo may support an "education of the senses" in which materials and objects support learning experiences that help children develop their sensory capabilities, control their own learning process and learn through personal exploration [Pia76].

Studies with Early Adolescents

Later evaluations with two eighth grade "Physics by Design" classes focused on Topobo's role supporting design, experimentation and conceptual abstraction. These students normally engage in group projects using manipulatives like LEGO Robolab, so the evaluation was designed to be like familiar classroom activities. We met with four groups of 8 students twice over two weeks, and students worked in pairs or groups of three. These sessions included three homework worksheets and interviews with students.



All kids related to Topobo models with their familar knowledge about animals and machines.

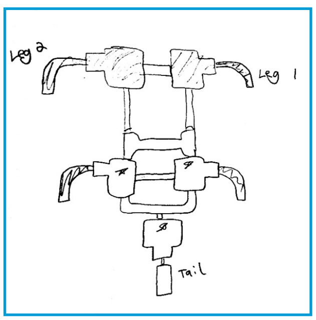
Our first evaluation session introduced the system. Using a preliminary worksheet, students described different types of motion related to their bodies based on both their pre-existing conceptual models of motion and then based on activities we designed. The next day, we explained how to use Topobo with demonstrations and examples.

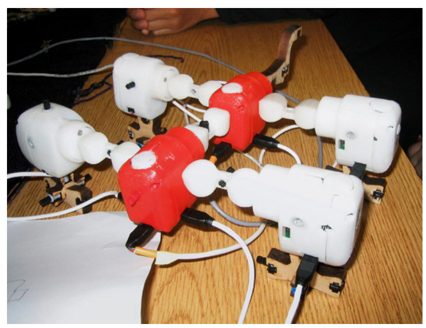
Students began by freely exploring the system. Many students built anthropomorphic creations, programming them to tell stories or wiggle around. Their creations often did not move as they expected.

Falling creations elicited exclamations like "add more legs" and "make it lower, like a baby." For most of these students, Topobo quickly became a tool to experiment with center of gravity and dynamic balance.

Iterative Design

The second evaluation session a week later focused on a task to construct a "walking creature." Students first planned and drew their creature and then tried to build it and make it walk. We observed two different methods of design. The first method involved "active iteration" during the creative process. Students built a small part of a creation, programmed it repeatedly until the desired motion was found and then added components, testing how the new components changed the dynamic balance of the creation. This process





Kids first designed a walking creation on paper and then tried to build it. These students compartmentalized building and programming.

continued until they had their desired creation. The second method involved students who would "compartmentalize" the processes of structural building and programming motion. Students who compartmentalized would build a creation in its entirety and then program its movement only at the end of their process.

Students who employed active iteration were more successful at building creations which walked and balanced. These students' creations tended to be very different from their original designs on paper and the students were generally able to explain how physical constraints had influenced their designs. In comparison, students who compartmentalized building and programming usually ended up deconstructing their creation and trying to rebuild it using a more iterative process.

These findings show that an interface design should support active iteration by allowing users to switch between interdependent processes. Users often need to test many ideas to incrementally develop a successful design. Students who initially compartmentalized the design of form and motion eventually adopted active iteration, suggesting that Topobo supports rapid experimentation with these interdependent processes. However, these findings also suggest that Topobo would benefit from an ability to save and reuse motions, so that forms can be edited and motion can be kept consistent.

This process of designing and testing also shows how building with Topobo leads older students to employ the Scientific Method. Students began by observing the action of their creature, creating a hypothesis on how to improve it, and testing that hypothesis with experimentation. While Topobo can be thought of as a system to specifically teach concepts of kinematics, for children capable of "formal operations," (11+ years) [Pia76] it can also be described as a tool for teaching students to think like scientists.

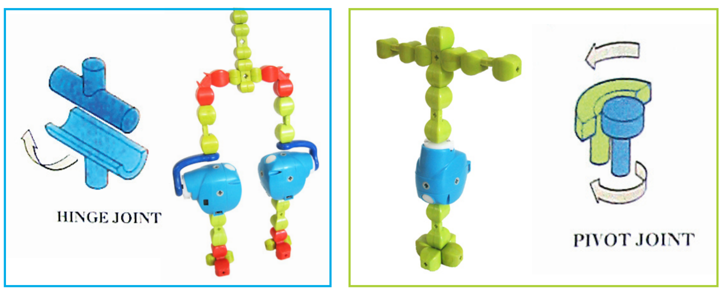
Evaluation of Queen functionality

Our evaluation of the Queen is inconclusive. Some students had success using the Queens, while others experienced a level of frustration with them. We believe some students became frustrated with them because using the Queens requires a different cognitive model than using Topobo with direct manipulation. In direct record mode, children focus on relative movement of the Actives, e.g. "how far did the leg move from its static position." However, this conceptual model does not work well with a Queen. Students would often begin by carefully positioning their creation before programming it. But as soon as the student pressed Record on the Queen, the creation would kick wildly out of position as the Actives mimicked the Queen's absolute angular position. This could be fixed by reorienting the Actives while they are recording, but the kids often thought something had broken and stopped their program before they could analyze and fix it. Their fear of broken parts was exacerbated because a software bug occasionally caused Queens to act erratically. After students were surprised by a Queen a few times, they would often give up and return to direct manipulation.

The Queen needs further engineering and design refinement. This study showed us that a minor bug can be an obstacle to learning if it causes greatly unexpected output. It also showed that in future interactions, Queens may require more scaffolding than direct manipulation with Topobo.

Animals and Machines

Kindergartners, second graders and eighth graders all related to Topobo models with their "familiar knowledge" about animals and machines. Metaphoric allusions to machines (robotics) and especially to animals ("the elephant," "the ant," "the scorpion," "the horse," "the no-walking man") were descriptive and salient. Many 8th grade students changed their creations based on their ideas about how animals and people move. "We tried to make it walk, but it couldn't balance so we made it crawl. You know, like a baby." One group experimented with creating a "frog" with scalloped legs. Another referenced the coordinated motion of a horse's legs, and another the crawling of a six legged insect. One of the groups explained that when their creation did not work as planned, they thought more deeply and specifically about the animal motion they were attempting to imitate than during the initial drawing of their design.



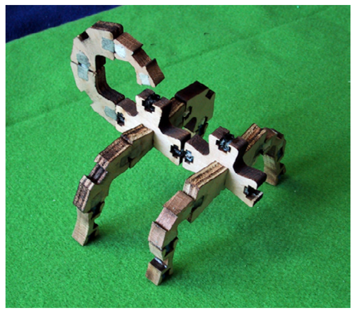
Actives naturally provide a pivot joint. The Elbow connectors allow Actives to behave like a hinge joint. These relationships can help children make connections between mammalian joints and robotics.

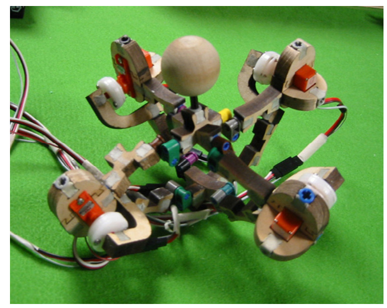
The fact that children can learn about the mechanical world through play with Topobo suggests, to a certain extent, the potential for body and ego syntonic learning as described by Papert [Pap80]. We believe that programming Topobo is a body syntonic activity because Topobo's kinematic motion, feedback, and global-local interactions are firmly related to children's sense and knowledge about their own bodies. Topobo my also be somewhat ego syntonic because it is coherent with children's' sense of themselves a people with intentions, goals, desire, likes and dislikes.

We also found evidence suggesting that for younger children, Topobo's relationship to the body may allow it to function as what Papert considers a transitional object. In Papert's view, a transitional object allows the children to make sense of tasks in terms of everyday familiar experience, but supports them in moving into the world of the abstract [Pap80]. We hope that further research will help us evaluate this hypothesis.

Age Range Findings

It appeared that all groups of kids had similar initial experiences of discovery. The children worked first to understand this unknown toy (or system or machine or thing, depending on the different vocabularies kids used to refer to Topobo). Children then worked to put together and assemble parts in a coherent way, and finally tried to program their constructions and test their movement.

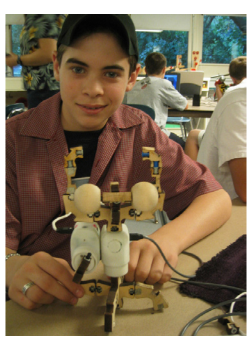




A second grader's static scorpion suggests that we are achieving our goal to make a system that is fun without the technology. A small mesh can change volume.

Kindergartners generally programmed only one Active. Some kindergartners puzzled over cause and effect with the programming and playback, while others understood the interface and playfully experimented with creations and storytelling. The second graders were much more deeply curious about the system, at times spending their entire recess working to refine a creation. This leads us to believe that Topobo may be best suited for children ages 7 and older.

Compared to the second graders, 8th graders were much more adept at programming subtle physical manipulations and were more successful at controlling movement. However, many students did not discover how to use more than one Active to create a single 2 DOF motion, and as a group, 8th graders seemed less comfortable experimenting with irregular arrangements of Actives than the younger children were. This suggests that children ages 8-11 who are in the process of developing abstract mental models, but still experiment very freely, may benefit most from Topobo.



Both second graders and eighth graders thought Topobo was probably designed for their age range.

We tested Topobo with a wide age range to evaluate its capacity to be both accessible and complex to children at widely varying educational levels. Eighth graders compared it to LEGO Mindstorms as a programming tool, and several students suggested that the addition of sensors and environmental feedback would improve the system. Both the second graders and the eighth graders concluded that Topobo was probably designed for their age range. This supports our hypothesis that Topobo can support learners at multiple levels. Vygotsky refers to the "zone of proximal development" [Vyg78] as the optimal learning stage where children are exploring concepts beyond those they would be able to understand independently, but are not dependent on adult support for learning. Our observations that students at multiple developmental levels effectively collaborate with Topobo encourages us that the system may support rich learning experiences during such cognitive transitions.

Domains of Knowledge

We found that Topobo can help students ages 7-13 to learn about several educational concepts:

Balance: When objects move, their center of gravity changes. Topobo draws attention to this fact when children make things that fall over. Learning how to control falling can lead to an understanding of familiar dynamic processes such as walking.

Center of Mass/Center of Gravity: Several groups of students built creations that were initially very tall and tended to fall over when they moved. One student described shortening the creation's legs to keep its weight closer to the ground. He referenced how it is easier for babies to crawl than to walk.

Coordination: When Topobo is directly manipulated, sequential motions are easy to record. A child might shake his Topobo dog's head, and then wag his Topobo dog's tail. However, shaking the dog's head and wagging the dog's tail at the same time is difficult because the child needs both hands to do either one of the activities. In order to coordinate these motions, it is necessary either to cooperate with other children (coordinating people) or to use a Queen (which coordinates movements in time). The Queen encourages developing an understanding of how coordinated movements can change a whole system.

Relative motion: A second grader built a long string of static parts with an Active part at each end. He programmed each end to wiggle back and forth and observed the ends shaking. Upon suggestion from an adult, he tried holding a shaking end, and was amazed to see his entire creation wave wildly back and forth. This drew his attention to the idea that movements in a connected system are relative to one's frame of reference.

Movement with Multiple Degrees of Freedom: A Topobo Active provides motion in one degree of freedom. One pair of eighth grade girls quickly figured out how they could connect two Actives with an elbow piece to create 2 DOF rotational motion. By applying this technique they were able to quickly create a walking moose. They could not explicitly describe how it worked; however they refined the same kind of motion in a different creation a week later.

Eighth graders test their walking creations.

Relationships between Local and Global Interactions: The educational value of understanding relationships between local and global interactions has been investigated at length with object-oriented programming languages such as AgentSheets and StarLogo [Res99]. Topobo makes certain systems concepts tangible with the Topobo Queens. One group of 8th graders discovered that faster legs (local) do not make a faster animal (global). Another group of three boys figured out quickly that they could create two separate networks of legs on either side of an animal, each governed by a Queen. Using this concept, they would be able to program each pair of legs with different motions but the legs in each network would have the same repeated motion.

Summary

Our early research suggests that Topobo can help children to understand certain physical principles affecting kinematic systems, and that Topobo can help children learn about the fields of modular robotics, system coordination, emergent dynamics (local vs. global behavior) and locomotion. Such concepts are not usually taught until high school or college level, and recent research [Ros95] arguing that people learn by building on prior knowledge suggests that qualitative experience with these ideas through playing with Topobo may help scaffold students into these complex fields.

Topobo makes complex ideas accessible by integrating the tangible interface tradition with digital manipulatives, endowing physical immediacy to normally immaterial dynamic computational processes. We believe that this approach can both expand the educational range of manipulative materials and can provide a physical bridge for children to transition from concrete to abstract operations. Physical computation is the basis of both tangible interfaces and traditional educational manipulatives. It can help children to use the next generation of computational educational tools to communicate, cooperate, and more deeply understand the natural world around them.

Looking Ahead

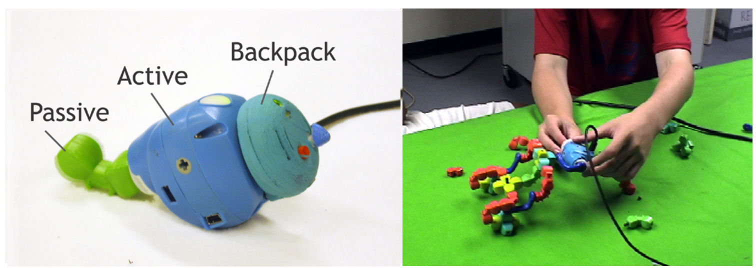
This early work led to user studies which attempt to determine how children are able to transfer knowledge from Topobo activities to other fields of knowledge. This required using Topobo with kids for long periods of time and evaluating a range of activities that target different cognitive levels. The goal of these studies was to encourage the development of different types of tangibles in school classrooms

This work also inspired means to extend the Topobo system to support scalability for expert users and to encourage different types of learners to use the system. For example, eighth graders' specifically requested sensors to control kinetic behavior. In the following chapters I will explain tangible techniques for saving motions, editing playback motions in real-time and making conditional behaviors, functions that are all characteristic of a traditional programming paradigm.

5 Beyond Record and Play

Backpacks: Tangible Modulators for Kinetic Behavior

How can a tangible interface retain the immediacy and emotional engagement of "record and play" and incorporate a mechanism for real time and direct modulation of behavior during program execution?



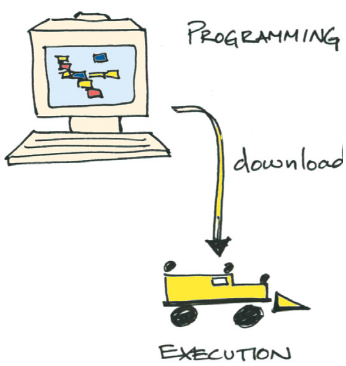
Faster-Slower Backpack attached to Active (left).
A student modulates a creation's playback frequency (right).

Backpacks are modular physical components that children can incorporate into robotic creations to modulate frequency, amplitude, phase and orientation of motion recordings. Using Backpacks, children can investigate basic kinematic principles that underlie why their specific creations exhibit the specific behaviors they observe. We demonstrate that Backpacks make tangible some of the benefits of symbolic abstraction, and introduce sensors, feedback and behavior modulation to the record and play paradigm. Through our review of user studies with children ages 6-15, we argue that Backpacks extend the conceptual limits of record and play with an interface that is consistent with both the physicality of educational manipulatives and the local-global systems dynamics that are characteristic of complex robots.

Playing with Physical Behavior

Digital manipulatives have employed several different styles of interface design that encourage children to create and test their models in different ways. These range from very immediate models like "record and play," a form of programming-by-demonstration, to textual or iconic symbolic programming.

Digital manipulatives that employ a traditional programming paradigm, such as LEGO Mindstorms, are praised for their flexibility and abstraction, but are difficult for novices to learn and use [Res99]. Due to their abstraction, models created with them are easy to fine-tune and edit because behavior is parameterized. Since they are designed after existing engineering tools, these systems can also introduce complex ideas about feedback and emergence in ways that cleanly map to expert design systems. However, these systems present divergent interaction models for physical model making and behavior-creating. Since the GUI and physical modeling paradigms are decoupled and conceptually different, parallel modeling of objects and their associated behaviors can be difficult for some learners.



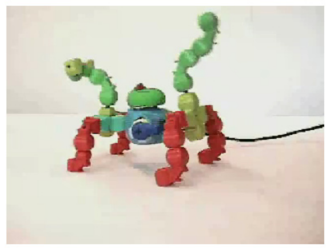
LEGO Mindstorms (left) and TellTale record and play (right).

Systems that employ record and play have been argued to be more experiential in nature and more intuitive for users than other programming paradigms [Ack99; Fre00; Ryo98]. With these systems children can express their own desires, intention and aesthetics in their model, because the structure of the model can reflect a learner's aesthetic desires rather than the symbolic structure of the system. This flexibility has been argued to facilitate learning because people become emotionally engaged with their work and focus on it deeply. However, since decoupling the physical and symbolic models results in systems that have no clear "handles" to edit the programs, in-

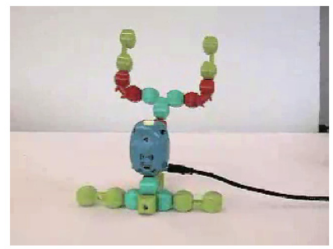
Four Backpacks -
Time Delay, Position Offset,
Faster-Slower and
Bigger-Smaller.

Time Delay (phase shift)

Position Offset (orientation)



Faster-Slower (speed)



Bigger-Smaller (amplitude)

terfaces for manipulating the programs' dynamics are not obvious. This absence of an interface to play with the programs means that children have fewer tools to understand the program's roles in determining the overall system behavior.

In general, systems that employ record and play are not thought to be very extensible. This has implications for digital manipulatives where children are, in part, modeling behavior. Extensibility is critical to make a system remain engaging as learners advance and want to experiment with more abstract concepts. A question then, is how to create digital manipulatives that retain the immediacy and emotional engagement of record and play and incorporate some of the flexibility and sophistication of control structures, feedback and parameterization of data, all concepts that are part of a traditional programming paradigm.

Backpacks

Backpacks introduce parameterized transformations, sensors and feedback to a modular robotic building system. Children use specialized modular components to control the behavior of their Topobo creations.

When using Topobo, a child will make a model, record a motion, and watch it play back. If he would like to change the movement of his creation, he will start over and record a new motion. Although a child can flexibly edit the shape of his physical model, he cannot edit the "shape" of his recording (the program).

Backpacks Design:

Backpacks allow children to modulate recorded Topobo motions. They are physical parts with a button and a knob that can be snapped onto an Active to modulate the phase, amplitude, frequency, or orientation of playback motions. These effects are described using familiar words, where phase is called Time Delay, frequency is called Faster-Slower, amplitude is called Bigger-Smaller, and orientation is called Offset. If we think of Topobo in terms of grammar, a child's physical creation is a "noun," its recorded motion is a "verb," and Backpacks are "adverbs."

Backpacks have three different modes — local, global and distributed — that give children tools to explore their creations' local-global interactions in detail.

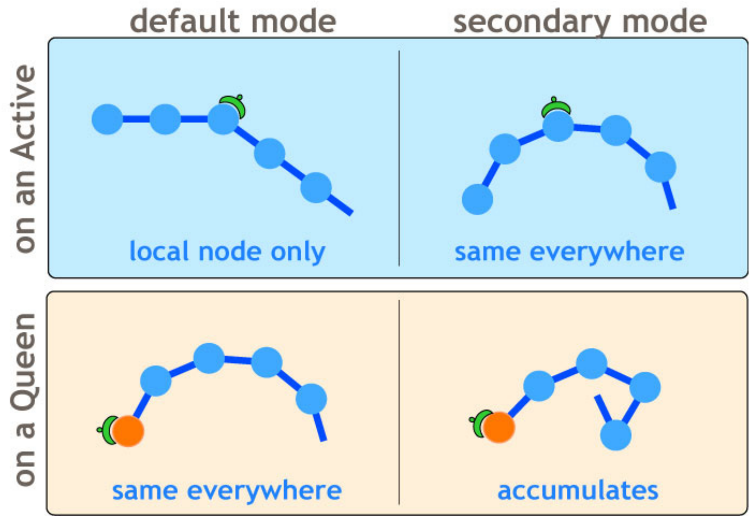
Local: When a Backpack is attached to an Active, it affects only that Active.

Global: A Backpack is attached to an Active, and its button is pushed. Or, the backpack is attached to a Queen. The Backpack identically affects every Active in the structure.

Distributed: A Backpack is attached to a Queen and its button is pushed. The backpack affects all Actives and its modulation is proportional to an Active's number of network hops from the Queen. Here, the rate of change is controlled with the Backpack's knob.

In the spirit of a building toy, Backpacks are modular: many may be used in parallel, in either local or global modes. They are designed to embody the principle of "coincident input/output" that is dominant among tangible interfaces [Ish97]: when a backpack is removed from the system its effect disappears, and the Actives will revert to their original recorded motions. In Topobo terminology, Backpacks are neither "Active" nor "Passive" because they introduce a new paradigm to Topobo that is physically static, but computationally dynamic.

By using Backpacks in different ways, we will explain how they



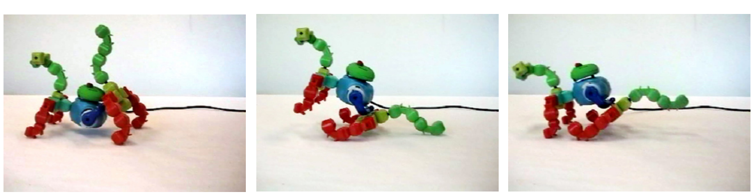
allow children to experiment with sensors, conditional behaviors and feedback in their kinetic creations with a physical model-making paradigm.

Domains of Knowledge

Children can use Backpacks to explore many ideas about local-global interactions that determine the behaviors of their creations. They can also explore ways that motion patterns can generate organized behavior in distributed systems. Although the original Topobo Queens and Augmented Queens illustrated some of these ideas, Backpacks allow children to more specifically test how local motion components like phase can affect a creature's overall movements.

Controlled Asynchrony
A child has made a dog that first turns its body and then shakes its head three times. Faster/Slower Backpack might be used to make a dog's body turn faster. The body is now out of sync with its head's movements.

Frequency is modulated to make this dog gallop.

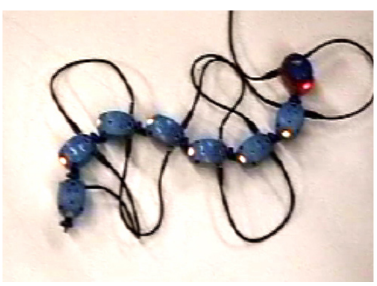


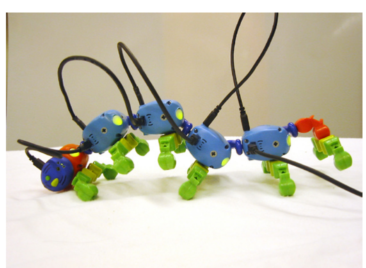
Phase Shift
Time Delay Backpack changes the moment at which an Active will start its loop relative to the other Actives in a creation. For instance, imagine a dog that is initially programmed to wag its tail and then shake its head. A child might attach a Time Delay Backpack to its tail and turn the knob on the Backpack to make the tail wag in sync with the head's shaking. Similarly, a dog that is trained to twist its front and back legs in sync may be adjusted so that it twists its front legs first. In this way, the dog can be made to walk. Conversely, making the rear legs twist first may make the dog walk backwards. This introduces ideas about positive and negative phase shift.

Distortion
Bigger-Smaller Backpack scales the recorded motion of an Active. Motions are scaled relative to the start position of the recording. Children may discover that, since Actives rotate only 170°, amplified motions may get "clipped" during playback.

A seven year old boy used Faster-Slower Backpack to make a walking dog move faster. To his surprise, its oscillating movements got smaller, rather than faster. With an adult's guidance, he understood that the motor could not move fast enough to play his "faster" recording.

Resonance
Faster-Slower Backpack can be used to see if faster motor movements create faster locomotion. Children can explore ideas related to resonance by building creatures that "gallop" and exploring how they may gallop more quickly when the Active itself is moving more slowly. Bigger-Smaller Backpack may also be used to find a structure's resonance, because some creatures walk better by taking larger steps and some walk better taking smaller steps.





Distributed Time Delay leads to waves (left) and a walking caterpillar (right).

Waves

When a child programs a structure with a Queen, all Actives will synchronously mimic the Queen. When a Backpack is attached to a Queen and the user pushes the backpack's button, a Distributed behavior causes the backpack's modulation to increase with distance from the Queen.

For instance, if a Queen is attached to a linear string of Actives, gradual rotations to the Queen will cause the string to curl into a circle. With the Time Delay Backpack, the Queen's movement will be mimicked after a propagation delay that is incremented between each Active in the string. Due to Topobo's looping playback, a wave-like motion results. Turning the knob on the Time Delay Backpack will change the shape of the wave.

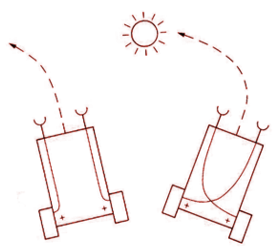
Spirals

If the child replaces Time Delay Backpack from the previous example with the Bigger-Smaller Backpack, he will cause this same string to curl into a flat nautilus spiral.

Harmonics

Faster-slower backpack can cause the same string to exhibit harmonic resonance.

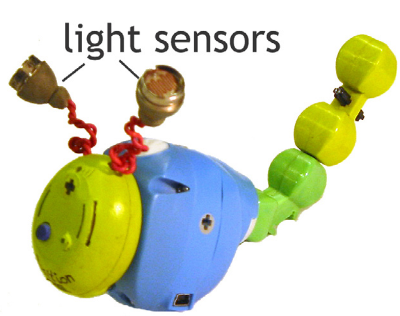
Time Delay Queen also can exhibit harmonic resonance when children turn a Backpack's knob to see if they can make a sinusoidal caterpillar walk both forwards and backwards with minute changes to the amount of time delay.



Following Braitenberg's examples, children can physically cross Backpack antennae to explore how some creatures exhibit opposite behavior, e.g. walking towards light vs. walking away from light.

Sub Networks of Control

Some students have used multiple power cords to create a single creation that has sub-networks that are governed by different

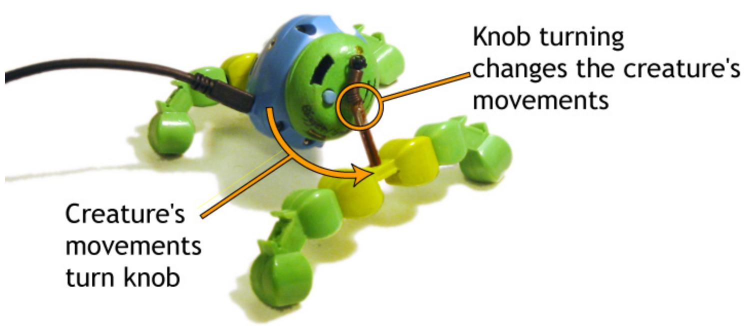


Light sensors replace the knob in this offset backpack, which children use to make their creature walk towards a flashlight.

Queens. With two Time Delay Queens, A centipede might have one network controlling the oscillations of its body and another that controls the wave-like undulations of its many feet. Coordinating the two motions relative to each other could lead to a robust and interesting centipede robot.

Ambient Sensors and Conditional Behaviors

Offset Backpack has two antennae with light sensor "eyes" in place of its knob. It demonstrates conditional behavior and environmental responses when children can use it to build creatures that can change their posture in response to ambient light. For instance, a child can design an ant that walks towards light. By manipulating the orientation of the antennae, children can discover principles about sensors and control; following Braitenberg's examples, [Bra86] a creature that walks towards light can be made to walk towards darkness by crossing the two antenna to opposite sides of the Backpack.



When a backpack becomes part of the structure of a creature, children can explore feedback.

Feedback

Backpacks can also be used to experiment with feedback. The Backpack's knob is fitted with a mechanical connector that allows it to become part of a creature's body. Now, the creature will behave differently when its posture changes. If the backpack is modulating the same motion that is affecting the position of its input knob, it presents a type of physical feedback mechanism.

Evolution of the UI design

We developed the "local," "global," and "distributed" Backpacks over a two-year design cycle. Distributed backpacks came first: we sought to make tangible and manipulable the abstract principles demonstrated with the Augmented Queens that were supposed to show how information behavior can model patterns of growth (like nautilus shells) and morphological change (like waves) over time. Since Augmented Queens were very hard for children to understand, the goal here is to make those principles of information change modular and tangible so that children can fluidly experiment with their effects on system behavior. The Backpacks' knobs allows students to more thoroughly and fluidly investigate the problem space.

The local backpack grew from that effort; it was the most obvious answer to the question "what happens if a (distributed) backpack is attached to a normal Active?" Local modulation suggested rich opportunities for control of creations.

Once we tested the local mode, we realized that creations with only one Active had the advantage that backpack motions could be conceived as global or local modulations. Through informal studies and interview, users told us that global modulations seemed fun and conceptually interesting. This led us to create a global mode for backpacks.

Sensors and feedback techniques also evolved from work with children and professional researchers. Some users have commented that the Offset Backpack is the best design because its "eyes" suggest its function. This has encouraged us to develop more specialized interfaces to physically embody the ideas of time, speed, and scale.

Although we were tempted to create separate backpacks for the three different modes (eliminating "invisible state"), we chose to keep the modes coupled to encourage students to make discoveries about the various effects of modulation to overall behavior. This coupling is also intended to lead students to form and compare both centralized and decentralized conceptual models of dynamic systems.

Evaluations with Children

Our evaluation of the Backpacks took place in a variety of settings with children aged 6-15. Throughout our design process, we frequently showed the system to children to determine its ease of use and affordances for manipulating its controls and combining it fluidly

K-3rd graders suggest new backpacks.

with the Topobo system. These sessions informed the final physical and interface design of the Backpacks.

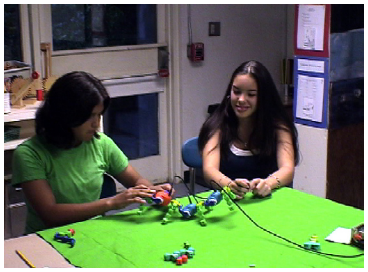
Kindergarten - Third Graders

We evaluated the Backpacks to explore their effectiveness in how tangibly manipulating motion parameters could facilitate the development of abstract ideas about motion. We conducted several informal afternoon sessions in a home environment, with eight children ranging from K-3rd grade, a mixture of boys and girls. The children were first introduced to the Topobo system, demonstrated how to use it and shown several Topobo creations which took advantage of the Backpack capabilities. They then had an afternoon of free play with the Topobo system and Backpacks with help available from researchers accustomed to working with children and Topobo. Most of the children in the session had not played with Topobo before, except for one third grade girl who had experienced early Topobo prototypes in her kindergarten class, and another seven year old boy who had evaluated Topobo informally in approximately six sessions in the previous two years.

Eighth Graders

Our next evaluation took place in the eighth grade classroom, in a physics-by-design class. We conducted two sessions with two separate classes, with a total of 26 students. These students had no previous experience with robotic or programming systems and had not been taught a foundation in dynamics or kinematics. However, the school they attended had a hands-on approach with manipulative materials available as part of the curriculum. In the first session, the students were introduced to the Topobo system and Backpacks and given free play with the system.

In the second session, the children were shown successful walking creations we had built, some of which utilized the backpacks. We demonstrated how the Backpack parameter control could manipulate walking. Following the introduction, half the class was given these built creations to analyze—take apart, change, rebuild—while the other half were instructed to create their own walking creatures. In between the sessions the classes were given homework workshops to test their conceptual understanding of the Backpacks and all the students were interviewed at the end of the last session.



8th graders experiment with models and behavior.

In both of our evaluations, we found that the Backpacks were an accessible interface for children to explore different parameters and introduced a set of concepts that ranged in complexity. All of the children were able to use the Backpacks, although a greater conceptual understanding was articulated by the eighth graders. Showing the children built creations with the Backpacks in use and allowing them to deconstruct their behavior greatly accelerated the children's conceptual understanding. This was a necessary first step with the younger children to engage totally with the Backpacks.

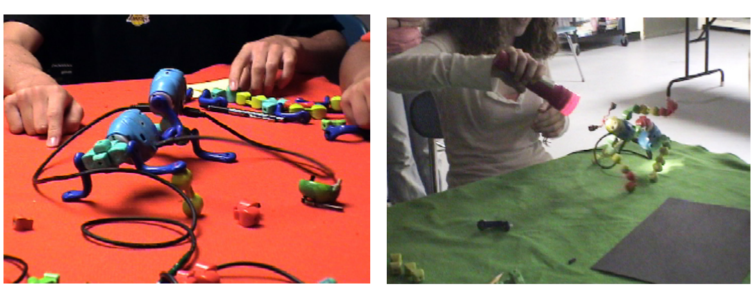
The backpacks that described more concrete physical concepts—moving Faster-Slower or Bigger-Smaller—were easier for all the children to observe, understand, utilize and describe. One eighth grade boy commented on how the Faster-Slower Backpack made getting his creature to walk easier. "You could probably do it without it, but it makes it a lot easier...rather than having to rerecord it every time you want to change the speed...you can also get it a little bit more precise with the Backpack." When employed in a creation, the children were able to understand that the Delay Backpack made the Actives move one after another, thus dissecting a fluid motion into its constituent parts. However, they did not articulate a direct connection to wave-like motion. The Offset Backpack proved to be the most difficult for the children to dissect; children could interpret that the sensor made the creation move toward the light, but only one group of eighth graders was able to articulate an obvious correlation with how the motion of the motor was changing (offsetting to one side) in relationship to the overall walking behavior that the creature demonstrated.

Fluid Integration Into Play

An important attribute of the backpacks was observed in how the backpacks were integrated into the creative process of using Topobo. In past studies with Topobo, researchers found that users who worked iteratively—going back and forth between building the creation and programming motions—had more success in making a creation walk. We found that the Backpacks integrated seamlessly with this iterative process, while adding a new element with which to iterate. In one session, two eighth grade boys were working on a walking creation with the Faster-Slower Backpack. Throughout their process they explored adding and removing passives to change the weight balance of their creature, reprogramming its motion, and changing the speed with the Backpack knob — all in a fluid and experimental manner. They cited the Backpack as being a necessary part of their creature, because it allowed them to control the speed of their creation without having to also reprogram (and thus over-write) the motion pattern.

A Logical Next Step

In one situation, two eighth grade boys had built a creation with a single active that walked forward and then attempted to make their creation turn in one direction. Through experimentation they



Students discover how to "steer" a walking creature with a second Active (left). Offset backpack can also steer a creature, but is harder to conceptualize (right).

found that they could successfully change the form of the structure, adding and subtracting passives to its legs, or could manipulate its motion, adding a new Active to its back which functioned to offset the motion like a steering column. In essence, these boys had struggled to discover the principle embodied in the Offset Backpack, which could have easily facilitated their iterations. This situation supports the idea that the Backpacks are building on motion principles already inherent in the system, but are providing a more abstracted and flexible form to approach and investigate the concepts they demonstrate; the Backpacks' functionality is a logical inclusion in the Topobo system.

Conceptualization

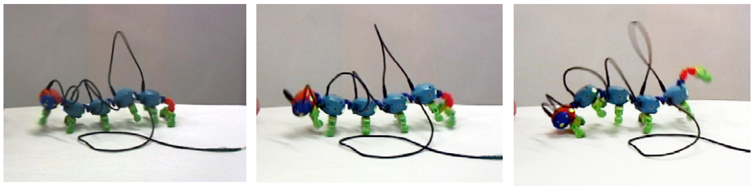
In an interview, Jack, a six year old who had played with Topobo in several sessions over two years, described that he would like to make his own backpack: one that randomized the motion, making Topobo "go crazy." In being able to envision his own backpack, Jack demonstrates that he has conceptually understood the principles behind the Backpacks, as manipulators of parameters of motion. This seems to support our hypothesis that a tangible representation of program manipulation provides a concrete bridge to more sophisticated concepts of control.

Beyond Children

In informal conversations throughout our research, we have encountered enthusiasm for the Backpacks by leading robot designers, especially those who are examining the relationship between geometry and movement. They describe the Backpacks as reflecting the real high level ways of thinking about robotics and motion control, viewing Backpacks as a tool for intuitive manipulation within a control structure.

From Play to Abstraction

A central question to different kinds of design tools concerns ease of entry (the "learning curve") and the potential complexity and sophistication of models created with a tool (the "ceiling"). One of the original pedagogical arguments with Topobo was that children of widely ranging developmental levels became engaged with Topobo



This caterpillar has been built to explore principles of phase shift and wave propagation.

because it was easy to learn and there were many points of entry for different learners; many levels of complexity were embedded in the system. However, children who were adept with manipulating abstract ideas [Col01] wanted to manipulate their recordings in different ways.

Backpacks increase the complexity with which children can design, control and understand their creations. Manipulating properties of the motion can then support students' development and manipulation of abstract ideas [Pia76] about the relationships between local motions and global behaviors. Backpacks provide one level of abstraction for students to analyze their work, and, as Ackermann argues, effective learning often involves temporarily standing back from the learning experience to reflect on it in more objective terms [Ack96; Ack99].

When people build a behavior themselves, they may have a deeper understanding of it. A caterpillar's movement looks very complex, but if a child builds a wave-like caterpillar with a Queen and Time Delay Backpack, the child has learned a simple rule to describe caterpillar-like motion. Since the rules "same motion everywhere" and "time delay" are represented tangibly, they can be seen, touched, edited, and understood in terms of familiar physical activities [Pap80]. Also, the process of building "complex" models from "simple" rules may give children insights into other kinds of complex systems [Res99].

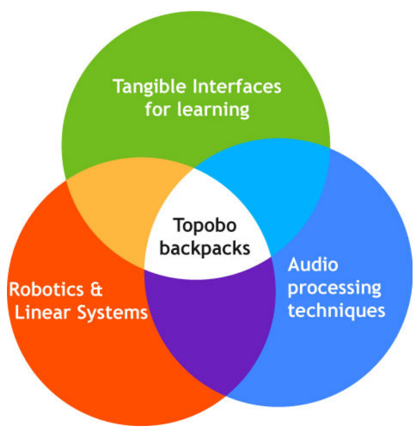
Manipulation and Analysis

Whereas an informal system like Topobo can lead to accidents and discovery, a pedagogical benefit of providing parameterized control via manipulatives is that advanced learners can fluidly transition between building, dissecting, and controlling their model. Control is one level removed from spontaneous creation, and Backpacks may help children to discover what, exactly, makes a behavior successful. When such abstract ideas can be formed and tested through concrete models, they can be used as the basis of new models [Pia76].

If a child working with Time Delay Backpack discovers that some Topobo creations walk almost entirely because of phase relations between parts — and almost any oscillating motion can result in walking — the student may then form a theory about phase and walking. She can later build a walking robot with LEGO Robolab whose movements are based on phase shifts.

Knowledge Transfer

In general, for children to be able to transfer ideas about motion learned via Topobo to other domains like math or programming, they have to develop generalized and abstract ideas about motion that map between the two domains. Topobo and Backpacks do not map onto mathematical kinematic models, but phase shifts, frequency and amplitude shifts are represented and manipulated in both paradigms [Rao04]. Providing immediate and accessible means to modulate parameters that underlie behavior can help people to ask the question, "how does my specific program/recording result in the specific behavior I observe?" For Topobo, Backpacks are convenient interfaces because Distributed and global-local interactions are central to a creation's behavior, and a tangible interface with local-global characteristics is an immediate and accessible means to discover this coupling.

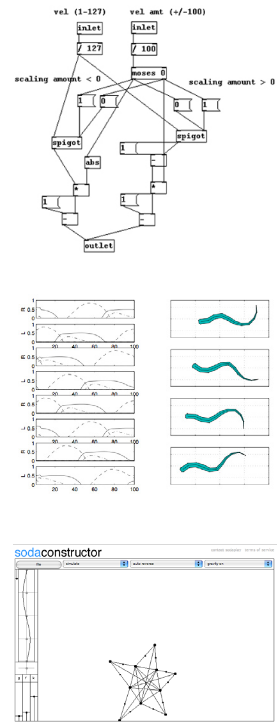


Backpacks integrate ideas from tangible interfaces, audio processing, and robotics.

Related Work

Backpacks incorporate ideas from a number of divergent fields of study, including robotics, educational toy design, audio processing and dynamic modeling. Since much of this work has been reviewed in chapter 2, I will review some highlights here and discuss work which directly inspired the development and use of Backpacks.

Children typically learn about dynamics through physical models, like springs, waves and swings. For instance, a child may be asked to explain why a slinky will "walk" down large stairs but not small ones, and be encouraged to develop a theory about resonance. Some cars develop a "rattle" only at certain speeds but not others; on a swing a child can kick higher and higher, but must kick at the right time. All of these physical examples of dynamics can support learning about more abstract descriptions of waves, resonance and harmonics. However, since underlying parameters like phase cannot be isolated and controlled, principles like phase are hard to understand. Backpacks provide handles to control parameters' effects on physical dynamic systems. When kids can model with these properties, they are encouraged to develop more advanced and abstracted theories about them [Chi97; Mel94].



From top: Puredata (dataflow programming model), central pattern generators, sodaConstructor.

Researchers have conceived of some modular extensions to introduce ideas about conditional behavior to a record and play paradigm. Frei suggested a simple switch for conditional behavior [Fre00] in which a primary motion is recorded, and then a secondary motion is programmed after touching the switch. Subsequent touches to the switch will toggle between primary and secondary motions. This binary state switch is an interesting idea that could be applicable to a system like Topobo, especially because it would result in complex local-global interactions. While this design introduces a hidden state that may be confusing, binary state change may an accessible way to work with multiple recordings.

Other domains have sophisticated tools to manipulate time-sampled data sets. Musicians who sample, mix and modulate recorded sound have employed different paradigms for record modulation. A tangible analog mixer performs transformations on audio (filters, volume) with a centralized interface. For more flexible audio processing, GUI tools are often used. Dataflow models like MAX/msp allow users to design and apply modular filters (small computer programs) to their recordings. Program structures are represented graphically and topologically and the system shares design characteristics with Backpacks, because filters are applied directly to the graphical programs and their effects can be experienced in real time. People have applied MAX/msp to audio, video and robotics, and the "dataflow" programming paradigm suggests interesting GUI extensions to the Topobo system [Dat07; MAX07].

Dataflow models are one approach to linear systems, which are more broadly used by researchers in many domains [Rao04]. Robotics researchers routinely use linear systems to model and understand the dynamics of their creations, and the principles that Backpacks represent tangibly are symbolically manipulated in their mathematical and programmatic models.

Linear systems have also been used in graphical simulation of kinematic structures. Sims' evolved virtual creatures [Sim94] employed directed graphs, a form of dataflow model. Sodaconstructor [Sod07] is a popular online GUI modeler for creating "walking" creatures that respond to a simulated physics environment. Thanks in part to its wide distribution over the internet, a large Sodaconstructor community has explored the roles of frequency, amplitude and phase in simulated locomotion of graphical models.

Other GUI learning tools like Starlogo [Res99] have allowed children to explore the ways local and distributed rules can lead to surprising system behaviors. We have made a few of these principles tangible with the Distributed Backpacks, although this conceptual domain is rich and may suggest future work in tangibles.

Many theories about phase shift and oscillations that come from biological systems, such as central pattern generators (CPGs) [Ijs98], are related to the concepts we present here. Specifically, researchers in modular robotics [Kam04; Yim00; Zha03] have explored the roles of phase, amplitude, frequency and orientation in determining their robots' dynamics. In some cases distributed algorithms similar to the Distributed Backpacks have been employed to create wheels, snakes and walking creatures [Zha03]. Our work intends to make these advanced ideas tangible and manipulable by younger students.

Braitenberg's Vehicles [Bra86] present compelling examples of simple control structures that result in "complex" and "natural" behaviors. These inspired the use of sensors and antennae on Backpacks, and his approach of continuous control of kinetic behavior shares structural characteristics of the Backpacks model.

From a design perspective, our approach is consistent with Full's argument [Ful98] that "preflexes" play a large role in the locomotion of simple animals like crabs or cockroaches. These creatures, and robots like them, exhibit behavior that may come largely from the interrelationships between an animal's morphology and its control system. In his robotics work, Full places great emphasis on developing the physical and control systems in parallel, which our work also emphasizes.

Building on Backpacks

Children have suggested a number of new backpacks to us, including reverse, random, and sound input. Their suggestions make sense and imply that children understand the backpack paradigm. Adults who have used the system have suggested backpacks that allow creatures to exchange motions, and a backpack API with which robotics designers could script behaviors and learning algorithms. For example, Maeda suggested "Scratchpacks" which allow children to script behaviors in the iconic programming language, and download them into backpacks. Full suggested Backpacks with AI algorithms to make robots that can respond more flexibly to their environment. Although extensions like artificial intelligence are inconsistent with our current goal to make the fundamentals of motion tangible, all of these suggestions suggest future extensions to the Backpack paradigm.

In order to more fully understand children's engagement with the ideas presented here, more thorough user studies are required. Although children understand the idea of a Backpack, they are often confused by the resulting behavior. This is understandable, since kinematics are complex and Backpacks draw students' attention to this. Backpacks would be best explored with a longitudinal study to determine how Topobo "experts" use and conceptualize Backpacks. Furthermore, more thorough and formal user studies are needed to identify ways in which different aged children can relate to Backpacks. We believe that the underlying ideas presented here range greatly in complexity, and identifying the developmental levels at which children can understand different ideas will allow us to better target specific ideas (and Backpack activities) to different aged children.

Summary

Backpacks were developed as tangible interfaces to modulate basic parameters of movement in a modular robotic building toy. We have argued that manipulating parameters of motion—with physical knobs, sensors and feedback—enables children to more deeply design and analyze sophisticated robotic behaviors. We have also hypothesized that making fundamental ideas like phase, amplitude and frequency manipulable may help older children transfer their knowledge from physical activities like Topobo to more abstract symbolic representations of movement like linear systems.

Although parameterized control, sensors and feedback are typically part of a traditional programming paradigm, we are not on a path to replace symbolic programming with tangible direct manipulation. There is still a big divide between symbolic descriptions of dynamics and simple record and play systems, and giving people tools to manipulate parameters is not the same as a mathematical approach. Our intention is to maintain the immediacy of record and play, and the analog data sets that result, and introduce some of the manipulation that is traditionally done with programming. We believe the strength of such a system lies not on its high degree of abstraction, but rather in an interaction model that makes certain complex ideas accessible and salient to children. We hope that the ideas presented here will "raise the ceiling" of complexity in record and play paradigms by making fundamental aspects of kinematic systems manipulable, without sacrificing any of the immediacy and playfulness that has been appreciated in record and play interfaces.

Looking Ahead

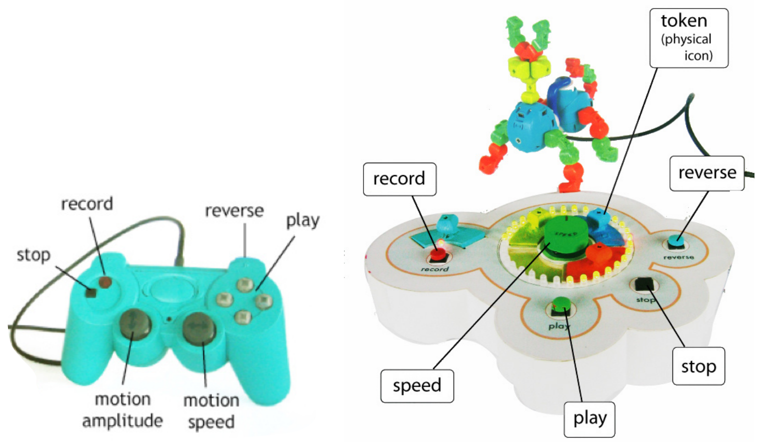
Backpacks help the individual learner play with and test kinematics ideas, but do not reinforce the social play patterns we have observed with children who use Topobo. Backpacks were a response to the question, "how can kids learn more by playing with Topobo behavior?" This question also suggests its counterpart: "how can kids learn more by playing with each other?" This counterpart formed the foundation of the next chapter, in which children's collaborative play patterns provide a foundation to support their inquiry with each other, and the system.

6 Remix and Robo
Sampling, sequencing and real-time control of kinetic behavior

Much of children's play – especially their dramatic play – centers on their desires to perform and act out their ideas through a surrogate object: dolls and puppets stand in for characters and people, remote control vehicles empower children with control of a machine, and video games provide surrogate characters that empower children to navigate through fantasy worlds. These games and toys appeal to children's sense of performance, fantasy and adventure – and represent important educational themes – but they the lack means of design and invention that are known to foster creativity and learning [Pap80].

Educators have found that performative events like robot design competitions motivate children to learn principles of robotic control. But while autonomous control demonstrates a deep understanding of the design of synthetic behavior [Pap80], a building-block approach to control lacks means to reflect children's improvisational performance.

Remix and Robo were designed to provide flexible and accessible tools to control robotic motion created with a tangible interface. We apply an interaction model from the audio domain to the robotic domain: the model *Record, Sample, Sequence, and Perform* is used to compose robotic motion, rather than music. People typically associate sampling and sequencing with rapidly growing music genres like hip-hop, and we explore how this interaction model can make robotics design more intuitive, playful and performative for children.



Robo (left) is a modified video game controller to perform a sequence of kinetic recordings with a Topobo creation. Remix (right) is a tangible sampler / sequencer that can be used to capture and edit Topobo motion recordings.

Robo and Remix leverage the visual language of performative interfaces like video game controllers and deejay turn tables to help children quickly understand the kinds of play the interfaces can support. Through tests with various age users, I will evaluate their usability for applications that are both artistic (e.g. robotic puppet shows) and athletic (e.g. robot competitions).

Remix & Robo: New Tools to Control a Robotic Construction Kit

Hypothesis

I hypothesize that providing means for capturing, organizing and controlling movement in real-time will help children analyze, understand, and refine the design of their robotic creations.

Approach

Remix and Robo are controllers children use to sample and sequence the movements of a Topobo creation. They are designed to support children's narratives and improvisational performances with Topobo.

Remix is a tangible sampler/sequencer to capture, adjust and recompose Topobo motions.

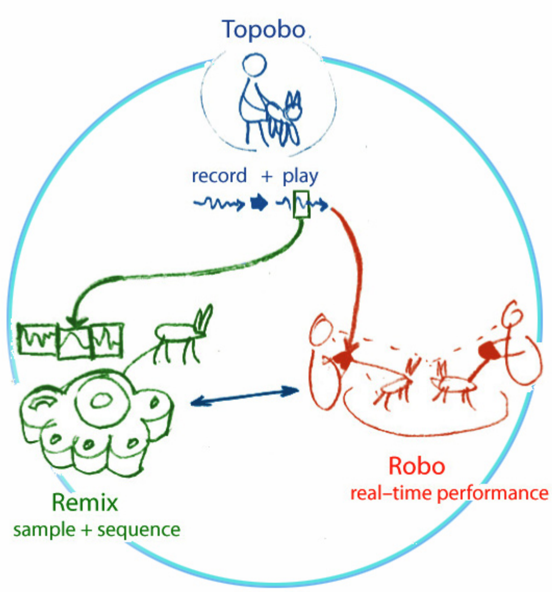
Robo is a modified video game controller that a child will use for real-time performance of his Topobo creation.

Scenario

A child builds a Topobo ant and creates a simple kinetic recording by moving the ant in his hands. The ant replays the child's movements by itself, in this case walking around on a table. The child then uses Remix to capture a favorite segment of this walking motion for later playback. He attaches Robo to the ant and adjusts the walking motion he has just captured with Remix, controlling the motion's speed, scale and direction in real-time.

A parallel to other media composition tools

Topobo, Remix to Robo can be compared to video performance tools: in video performance, a camera will be used for pure data capture (Topobo), an editing suite will be used to sample, sequence and organize a library of video clips (Remix), and video-jockey tools will be used to perform video mixing spontaneously (Robo). Such tools are designed to be used interchangeably, have some functional overlap (e.g. one could conceivably video-jockey with raw unedited video data), and are tailored to support different usage patterns.



Related Work

This design investigation is informed by research in interface design, digital construction kits, and audio/visual sampling and performance equipment. Some of this work has been previously reviewed.

Tangibles and abstraction

In developing tangibles, one challenge has been to create tangible means of assigning and manipulating data. Mediablocks, which inspired the design of the Remix tokens, were wooden blocks that referenced data that resided on a network. Manipulating the blocks could perform various manipulations to the data such as copying and printing a document [Ull99]. Mapping is a general design problem for all of this work: how should an abstract idea be represented and controlled? What is the proper "level of abstraction" to represent?

Performative competitions like FIRST have motivated a generation of young hands-on builders to focus on engineering concepts.

Digital Construction Kits & Performance

Much of the work in digital construction kits [LEG07; O'Ma05; Res98; Suz93; Zuc05] has focused on science and engineering learning. Digital manipulatives like Mindstorms illustrate how toys can stimulate science and engineering activities through application of engineering based tools (e.g. gears, levers, motors, wires, procedural code). Children often choose such tools if they are already motivated by science and engineering activities [O'Ma05]. These tools lean on scientific knowledge and interests that kids already have, before they even use the systems. Children's desires to perform and compete have sparked a number of robotic design competitions, e.g. FIRST robotic competition, although most focus on autonomous control and are based on logical (rather than dramatic) styles of learning [Gar83].

Audio samplers and mixers inspired flexible and performative interfaces.

Audio samplers and mixers

The emergence of sampler/sequencers and performative mixing devices (like deejay turntables) in the audio domain has inspired us to apply similar techniques to robotics. Where audio is concerned with recording and composition of recorded analog sounds, we envision robotics control as a recording and composition of recorded continuous gestural motion.

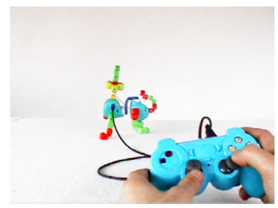
Design Overview

With Remix and Robo Topobo we present a modular system that offers both the benefits of hands-on programming and the flexibility of more abstract controllers. We consider this to be an evolution in tangibles for learning, where a tangible interface's coincident input/output [Ish97] model is extended with the addition of controllers for sampling and sequencing simple programs. This provides a limited form of abstracted control that makes traditional computer interfaces flexible.

Robo Design and Use

A child first builds a creation with Topobo. To record a motion, she presses a button on Topobo and moves Topobo in her hands as desired. She presses the button again to stop recording and start a looping playback mode. She can save the recording with Robo, a customized game controller, by pressing Robo's "record" button and then press one of its four "playback" buttons to assign the entire recording to that button.

Below, a moose is (1) programmed and then its program is (2) saved and controlled with Robo. Left, the moose and flower are performed in a robotic "puppet show."

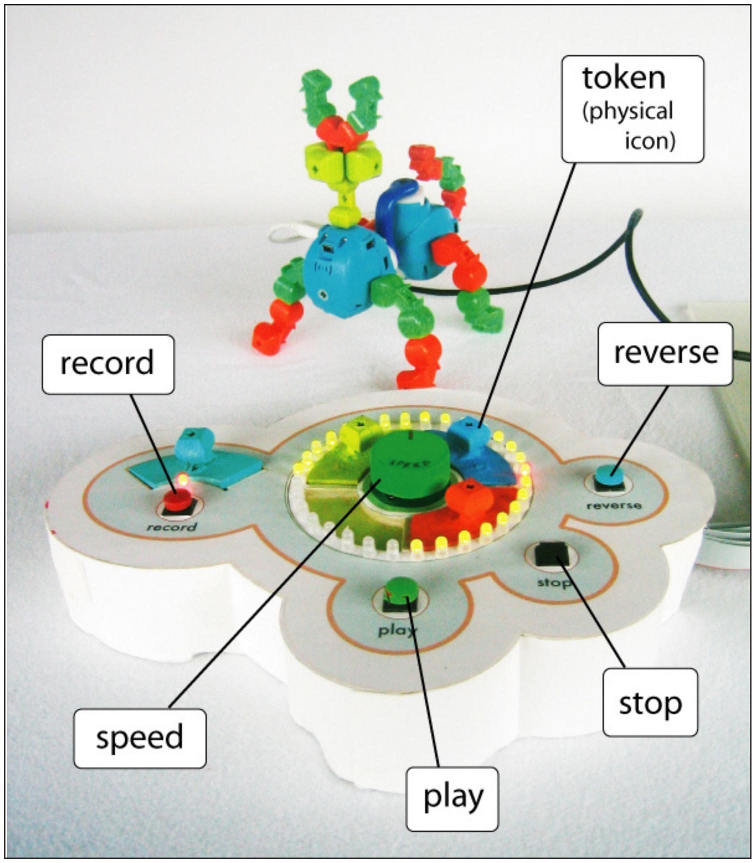


When a creation is in "playback" mode, joysticks adjust speed and amplitude of the motions. Continuously depressing a "reverse" button will cause a recorded motions to play backwards. (Reverse may cause a creature that walks forward to walk backwards.) Users of Robo can spontaneously control Topobo motions in real-time to create original sequences of movements.

Remix Design and Use

While Robo maps an entire gestural recording to a button for later playback, with Remix a user can sample (record) arbitrary amounts of continuous motion with a wooden token. She can sequence up to four tokens (representing different motion records) for looping play-back, while controlling the speed and direction of playback.

A user will first build a creation and set it into looping playback motion. To sample a piece of the motion, she will place a wooden token in Remix's "record" slot and push Remix's "record" button. A red light signifies that Remix is recording. To stop recording, the user will push the record button again or remove the token from the record slot.



To playback this motion, the user will move the wooden token to one of four slots in a donut-shaped "playback arena," and press Remix's "play" button. Green lights beside the token signify that it is mapped to a recording, and a red marquee light advances as the recording plays (fig. 4). The user may turn a green knob on Remix to change the rate of playback, or push a button to change the direction of playback. She can sequence up to four distinct recordings to loop in the playback arena.

Manipulating records

Remix records whatever Topobo is doing when Remix's record light is on. This enables a number of possibilities.

Partial or multiple loop saves:

A user may gesturally create a very long, changing series of footsteps for a walking creature. On playback, she realizes that a very small section of the recording produces satisfactory walking. She uses

Remix to capture only the effective steps. Looping playback of this new recording creates a continuous, repeatable walking movement.

Copying records

A recording is captured with a token and set into playback. A second token is used to record and duplicate the movement.

Saving modulations

Reverses and subtle changes to speed can be saved by recording playback motions that are controlled from Remix. For instance a user first creates a slow gestural recording with Topobo. He will then use Remix to capture the movement, and will use Remix to playback the recording at twice the original speed. This faster playback is again captured with Remix, and then played back at twice its recorded speed (four times the speed of the original gestural record).

Nesting recordings

A user will map a walking motion to the Red token and a dancing motion to the blue token. They are sequenced in the playback arena, and a green token is used to sample (capture) Topobo's performance of both records in series. The green token now references concatenated copies of both the walking and dancing motions.

Improvising

A user may assign several different motions to different tokens. While keeping Remix in "play" mode, he can rapidly place and remove the tokens in the playback arena to force his creation to spontaneously play any single record. The effect is similar to pushing buttons on Robo.

Using Remix and Robo Interchangeably

Remix and Robo reference identical nonvolatile memory banks inside Topobo Actives. This allows users to interchangeably use Remix and Robo to control the same creation. For instance, a child may use Remix to accurately sample specific sections of gestural recordings, and then use Robo to perform those motions more spontaneously. Or, a Robo user may discover that a particular sequence of movements creates a desirable effect, and then use Remix to copy that sequence into a single record.

Design Process

Remix and Robo evolved over two years with graphic, industrial and interaction designs refined in response to user feedback.

Remix

Initial Remix designs were conceived on paper and several months were spent writing firmware. We built a GUI prototype to explore questions of mapping and determine how much abstraction was appropriate for the controller. However, the user experience at a GUI was so different than Topobo play that the simulation did not help us to evaluate our basic questions.

Paper models of tangible controllers allowed us to quickly test for usability, size, and aesthetics for Remix. For our final Remix design, we connected a foam-core and paper prototype with embedded LEDs and switches to breadboarded electronics.

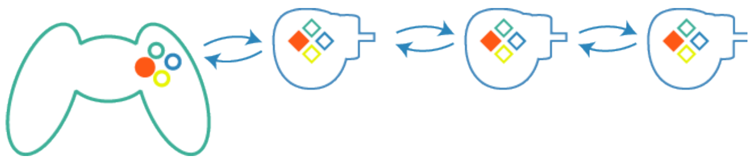
Robo

Robo designs began on paper with storyboard prototyping of interactions and play-acting of its conceived function with novice Topobo users. For our final implementation we modified a standard game controller by removing many functions and creating Topobo-compatible embedded circuitry with backlit buttons.

Robo evolved from experiments with Remix. Some users found that viewing Remix's intricacies and lights distracted them from viewing the movements of their creations, and one objective with Robo was to create a performance–based controller whose operation required only a user's kinesthetic sense. When children can quickly learn to operate the device through touch alone, the child's eyes and ears are free to focus on the Topobo creations themselves, or on other children who are participating in the activity.

Centralized control of a decentralized system

Topobo is a distributed system comprised of individual robotic elements each with their own internal parameters (e.g. speed) that define their behavior. As discussed earlier, Topobo Actives have embedded motors and electronics to manage power distribution, motor control, and a custom distributed peer-to-peer network. Robo and Remix allow for centralized global control of Topobo so that all Topobo Actives share a common set of parameters (fig. 5). All computation is embedded and distributed among the toys, and external power is supplied to a single element for distribution to all others in a creation.



Robo and Remix allow for globalized, central control of a decentralized system.

User evaluations

Our qualitative evaluation is designed to address how controllers can support children to analyze and refine their robotic designs. Developing proficiency with Topobo takes all users a minimum of one or two hours of play. Creating quirky and fun Topobo creatures is easy, but understanding the dynamics of Topobo behavior is extremely difficult.

We assumed that a deep understanding of all tools would require several further hours of practice with them. Therefore, we conducted our study with a wide range of users to capture their usability for people at different levels of development and expertise.

We worked with 16 users from age 4–adult to evaluate the design, usability and function of the interfaces. We sought to understand how users would integrate Remix and Robo into their design and problem solving strategies, and how Remix and Robo might support or interfere with iterative design strategies that successful Topobo users applied in previous studies. The evaluations revealed a variety of styles of performative play.

Methodology
Groups of 2-5 users worked with Topobo simultaneously, in a playful lab environment. A researcher explained the Topobo system, showing how parts could be assembled, gesturally programmed to move, and adjusted to achieve different kinds of behavior. Walking creatures and a video of Topobo locomotion were quickly demonstrated. A researcher then explained how Robo and Remix worked. Users were asked to explore the Topobo system and design a character.

All users elected to work for a minimum of three hours. Typically, a user spent one hour building various creations with Topobo, exploring different kinds of movement and trying to understand how to make a robot walk. Following explorations incorporated the controllers in various ways. If a user seemed to be confused a researcher may have offered suggestions. Some users returned on multiple sessions. One eleven year old reportedly discussed his work with his mother for five continuous days in between play sessions. All sessions were video taped and later analyzed and coded by a researcher for analysis.

Competitive endeavors among users age 7 to Adult

Desires to perform and compete can motivate children to play with our system. Young boys, in particular, love to get together and act out battles with their action figures and other toys. This inspired us to organize "Battle Bots" competitions. The competitions posed a steep challenge: people often discover that their Topobo creations may "walk," but creating Topobo creatures that walk predictably—and can be controlled—is extremely difficult. We hoped that Battle Bots may provide a socially and emotionally motivating reason for boys to develop mastery with Topobo locomotion.

Jonathan programs his robot, and later reaches in to the battle to refine his work.

Jonathan, 7 years old, plays battle bots

"THEY'RE GREAT! GREAT! This is better than action figures... better than video games. Why? It's just funner, I don't know....can we do a little more fighting?"

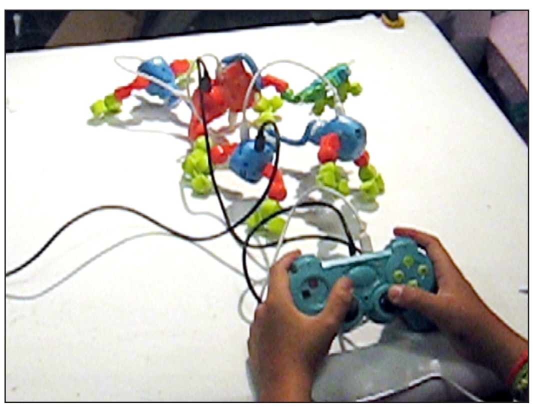
Jonathan and his friend have been playing with Topobo for three hours, spending the first 90 minutes with free play and experimentation. Both boys are paired with an adult (parent or researcher) because they can discover controllable locomotion much more quickly with the support of an older peer or adult [Vyg78].

An adult programs a robot to walk and shows Jonathan how to control the walking with Robo. Jonathan immediately wants to battle his friend, who is not ready. Jonathan then sets himself to learn to use Robo, and gesturally records and captures his own recordings, improving on the adult's design.

Battles ensue. For an hour, the boys compete, redesign, and compete again. A researcher asks: "Was [Robo] confusing at first?" Jonathan: "Yeah, but then it's easy now. You needed to get how to control it. It would have been hard to figure out if no one was teaching me."

Jonathan loves the idea of Battle Bots. "When I want to protect myself I want to do the kicking move [acts out kung-fu moves with his body]." But Robo became motivating for Jonathan only when he could successfully control a creature someone else had designed. For his age and skill level, Jonathan needs more time to develop controllable locomotion himself.

Topobo Battle Bots may be too difficult a task for a young child to engage in alone, unless he is provided with specific examples that allow him to feel successful very quickly. Older children may succeed more easily. This feeling of immediate success seems necessary to motivate a child to develop mastery, but Topobo play typically leads to quirky robots with amusing motions, not vehicles with highly controllable locomotion. However, Jonathan's overwhelming excitement at the idea of battle bots suggests that researchers should establish techniques to support dramatic play with a digital manipulative. The challenge remains to remove the "speed bumps" associated with learning how to transform simple, playful designs into understandable and controllable ones.



Robot wars focus two users on the difficult task of designing controllable ambulatory locomotion.

Character Design for storytelling

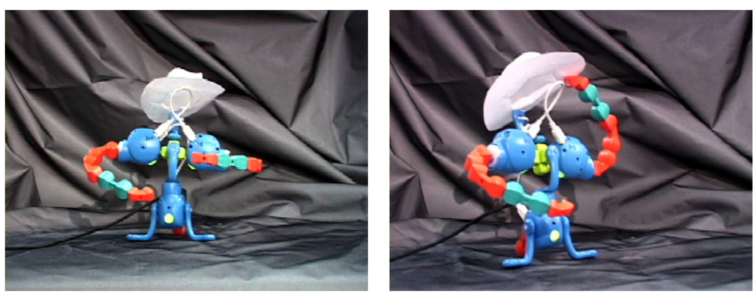
Jasper (age 11) demonstrates a flying "phoenix" that can flap its wings in different ways. He animates the wings and then practices flying it in the air by waving its entire body around. He then hangs it from the ceiling and experiments with recreating his earlier gestural motions with Topobo.

An adult suggests that Jasper picture his Phoenix in a movie, flying over a moving background. Jasper immediately imagines his bird diving for a mouse, and uses Robo to capture a diving posture he invents. Then, Jasper proceeds to create and capture several different recordings. Some are static postures—akin to an animator's keyframes—and others are dynamic recordings. Jasper demonstrates his Phoenix's range of movements in anticipation for a story.

Jasper is most excited to animate his creation when he actively imagines an animated background behind it. This indicates that while Robo provided tools for performance, his activity was lacking a context.

Robotic Puppeteering

This is walking. This is anger. And this is respect... With a few moves, you have enough expression to do a whole movie.



Bob uses Robo to direct his cowboy to show anger (left) and respect (right).

Bob, an experienced adult animator, uses paper to decorate his cowboy creature (fig. 7) and experiments both with continuous animation and with "keyframe" recording using Robo, by recording still gestures. "[Keyframes are] a little more 'real time.' I was constantly pushing buttons to do everything, which was satisfying."

Robo and Remix allow Bob to create characters with a wide range of expressive range. "You couldn't do character animation without these controllers. This is a different problem than getting something to walk.... It would be interesting to work from a script, because I bet we could get something rapidly across." Bob suggests applying the interface to expert puppeteering [Bel00].

From Playful Discovery to the Design of Controllable Behaviors

Our evaluation confirmed our hypothesis that providing tools to control robotic behavior supported children to analyze and refine their designs. All users related to the system first as a building toy and secondly as a robotic vehicle, a character, or a puppet for narrative performance. The introduction of Robo and Remix did not alter the basic character or play pattern with Topobo, evidenced by all users' intense interactions with Topobo prior to employing the controllers. A user explains, "why didn't we use the controllers in the beginning? We needed a creature first!"

Controllers supported users' individual interests
Users who had developed successful characters employed the controllers in various ways—competition, performance, global controls for investigating physics dynamics—depending on users' personal interests. Some people used Remix and Robo to refine their gestural designs, for instance to create more successful locomotion. Others used the controllers to apply their work to a secondary application domain, such as narrative performance. For most users, the controllers played into people's existing hands-on design process, allowing people to adjust and understand abstract variables for motion, and to reflect on their own design and thinking.

Rachel uses both Remix and Robo to design and experiment with her walking creature.

One challenge with Topobo is to predict how a gestural recording will make a creature behave once it is set on a table, reacting to friction and gravity rather than to the movements of one's hands. For several users, controllers were a convenient way to debug motions, since variations to movement could be observed while they were being created.

From direct to remote control

Departure from Topobo's tightly coupled input/output model is a necessary compromise because tight i/o coincidence is very limited. To accommodate increasingly skilled users, an interface must reflect users' thinking at multiple levels of abstraction. Our goal with Robo and Remix is to help children climb a mountain of ideas about dynamic physics, helping them understand how and why moving structures like animals behave the ways they do.

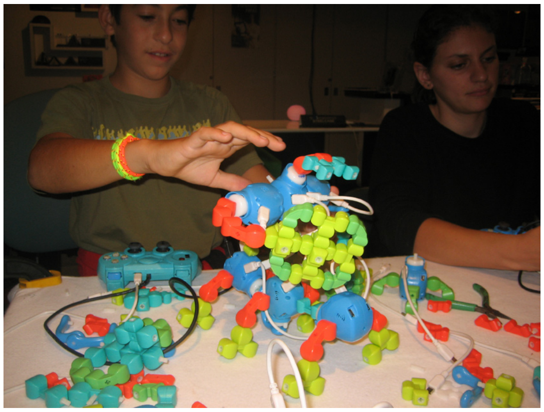
The original Topobo system includes "Queens," special orange Actives that instruct all connected Actives to mimic the motion recorded with the Queen. Some people use the Queens as remote controllers to program the behavior of a creature, observing the movement of the creature as they are programming it. In comparison to Queens, Robo and Remix facilitate less direct interactions with Topobo. Their benefit is greater flexibility and a higher degree of control.

We observed that users ages 7–adult found Remix and Robo to be an important part of their mastery of new ideas. According to one adult who rapidly learned how to achieve his goals with Topobo, "Robo and Remix show that the system does actually develop with you. Even as you get smarter, you can still learn something with Topobo. [Remix and Robo] are something you use in different ways as you get better at it."

Expressive and exploratory learning

Work in developmental psychology suggests that effective learning should involve both expressive activity, where the tangible represents or embodies the learner's behavior (physically or digitally), and exploratory activity, where the learner explores the model embodied in the tangible interface [Ack96; Ack99; Mar03]. The challenge is to engage the learner in an immersive and exploratory activity, and then help him to think about and understand what he has done.

When working with manipulatives, we believe that controllers may facilitate this process: they encourage a physical "stepping-back" and observing of one's work (fig. 8), and by carefully mapping controls to concepts that underlie a system's behavior, they can make important concepts manipulable and salient for users.



Controllers encourage people to step back and reflect on their experiences.

Physical controllers versus GUI controllers:
For hands-on learning the road to abstraction may not lead to the GUI. We asked users over the age of 10 if they would have preferred a graphical interface to Robo and Remix. All of the users said no. While one user suggested that a GUI could enable people to precisely represent and control the motor movements in their Topobo creations, she thought it might be distracting.

People said that specialized controllers "fit the Topobo system" better, that they liked the Remix tokens and enjoyed moving them around, and that the controllers were easy to use. Several users liked that they didn't need to "use a computer" to play with the system. One user commented that the controllers seemed more similar to the basic Topobo system because what the user did with his hands seemed to be more directly related to what Topobo was doing. "They're somewhere in between a tangible interface and a graphical interface."

Summary

Remix and Robo are specially designed controllers that enable sampling, sequencing and real-time modulation of gesturally-recorded robotic motion. The controllers motivate and support users to learn about dynamic physics concepts like center of mass and dynamic balance through focused play with Topobo. Some users employ the controllers as part of an iterative design process, where global control of variables allows users to better understand why their creations behave as they do. Other users focus on learning how to make Topobo perform predictable and controllable behaviors specifically to participate in new applications like competition or storytelling. Remix and Robo appeal to users who are (1) interested in model making with Topobo and (2) have an interest in dramatic social interactions. Remix and Robo support basic playful learning with Topobo and provide valuable tools to both novice and expert users.

Remix and Robo pursue new approaches to constructivist education, or learning by actively experimenting with ideas in the world [Pia76]. In hands-on education, a child may build something, and that thing enters the child's social context [Vyg78]. In some situations, a child may wish to design or control his creation's behavior in that context. Specialized controllers are one tool children may use to design behavior for their creations, in a way that captures the spontaneity and improvisational spirit that radiates from a child's experimentation and play.

7 Topobo in the wild
Longitudinal Evaluations of Educators Appropriating a Tangible Interface

What issues arise when designing and deploying tangibles for learning in long term evaluation? This chapter reports on a series of studies in which the Topobo system was provided to educators and designers to use over extended periods of time in the context of their day-to-day work. Tangibles for learning - like all educational materials - must be evaluated in relation both to the student and the teacher, but most studies of tangibles for learning focus on the student as user. Here, we focus on the conception of the educator, and their use of the tangible interface in the absence of an inventor or HCI researcher. The results of this study identify design and pedagogical issues that arise in response to distribution of a tangible for learning in different educational environments.

Tangible challenges

Because of the physical nature of tangibles, large scale deployment (which could be much more easily accomplished in a software system) is challenging; it is difficult and expensive to produce and maintain the extensive hardware necessary. Research projects are generally evaluated in small scale user studies run by the researchers who created them, and who are looking to qualitatively examine a planned hypothesis and evaluate the children's experiences and/ or use of the interface. Such studies often employ observation and interview with the users, and follow an ethnographic model of qualitative evaluation. However, ethnographic methodology has shown that in real world situations, the issues and results that people confront with products or systems are often divergent of the designer's assumptions, and often arise when the designer is removed from the scenario [Rog12]. Furthermore, in the case of educational tools, the context of use includes both the student and the teacher, but most studies of tangibles for learning focus on the student as user.

Children play with Topobo at a festival in Denmark

Goals

Topobo was provided to educators and designers to use in the context of their day-to-day work, over extended periods of time. Over the past three years, tens of thousands of people in Europe, Asia and North America have experienced Topobo in settings ranging from classrooms, museums, festivals, workshops, community centers, and homes. While numerous, many of the interactions were very short in exposure and confirmed the initial findings of the original Topobo studies. As part of a deeper question concerning the potential educational impact of a tangible interface, we sought to turn sets of Topobo over to educators to address issues related to large-scale use of a tangible for learning:

- In what contexts and environments can Topobo succeed?
- Over what time period will children use Topobo, and how will their use and interpretations of the system evolve?
- What age children will benefit from Topobo, and how will their experiences differ?
- What uses will other educators invent with Topobo?

Methodology

As part of a research initiative pursuing outreach for educational technologies [iCa05], Topobo was reengineered and mass produced with the specific purpose of providing educators with a new means to explore motion construction and kinematics principles. This manufacturing effort was funded by a modest educational outreach grant and required two years of extensive collaboration with an Asian toy manufacturer. Sets of manufactured Topobo were then distributed to educators (teachers, museum developers, educational researchers, graduate students) in the United States and Europe. The sets included Actives, Passives, Queens, power supplies and cables, and simple booklets. The booklets (similar to Appendix B) described the project concept, design and technical details, instructions for programming, and three sample creations with basic assembly instructions. The educators were also directed to the Topobo website which contains additional videos, published papers and visual materials. Educators did not have access to Backpacks, Remix or Robo.

Extensive data has been collected over the past year and a half, mostly in the form of interviews with educators and educational researchers working with Topobo. We are seeking to examine the perspective of the educators, and their reactions and plans when presented with Topobo as a new educational toy or kinetic material. We report how Topobo was used by various educators and what kind of initiatives, programming, or curricula they developed in these different environments when the researcher was removed entirely from designing a study or guiding the technology. In this respect, the teachers (not their pupils) are the "users" we address.

Figure 4. Breakdown of the five selected case studies

Educator	**Context**	**Student Age**	**No. students**	**Time Span**	**Interaction**
teachers	after-school enrichment program	13-15	18	3 mos.	themed sessions, free play
science teacher	4th & 7th grade science classrooms	9-10, 12-13	36	8 mos.	goal-oriented lessons, free play
educational researcher	after-school robotics center	4-6, 8-14	32	5 mos.	guided sessions
exhibit developers & programmers	science museum	4-adult (target 9-15)	200+	4 mos.	on-the-floor museum activities, demos, internal conversation
graduate architecture students	archtecture course/studio	24-29	12 (focus on 1)	8 mos.	self-directed thesis design work

Five case studies

The five following case studies represent a sampling of our research findings in diverse educational contexts with varying aged populations. They represent a cross section of usage environments, target age user and target user scenario. They were chosen because they are representative of common findings while at the same time offer significant depth and layered complexity from which to draw analysis. We aim to highlight the specific issues associated with using a tangible technology in different environments, and to identify the common issues that arise for educators in all environments.

After school enrichment program

Over the summer, sets of Topobo were loaned to an after-school enrichment program for middle and high school students. The director, Lori first saw Topobo in use in a local classroom and inspired by its potential, sought out to procure sets for her summer program. She intended to provide the system as an inspiration material for her program teachers with the hope of incorporating it in a more structured way the following summer. We provided a basic explanation of the system but did not set expectations of what we thought it should be used for or how we saw it fitting into her program. As curriculum director, Lori became the liaison to the teachers, explaining the system. Her enthusiasm for Topobo was shared by Dale, a middle and high school technology teacher in the program who used Topobo in his class.

Putting Topobo to Use
Dale conducted two sessions, two hours long, each of 7-9 students aged 13-15. Students elected to join both sessions and the second session contained many repeat students from the first session, which Dale interpreted as a sign that the students had made progress with Topobo and wanted to learn more. After some quick initial experimentation on his own, Dale began by giving the students a challenge of which he participated, "I'm having trouble getting something to walk [in reality, he was], can you make it walk? " Three boys in the session ended up making a walking robot but did a lot of purely structural experimentation until they began to use the Actives to actually connect, control and locomote the structure.

In the second session, Dale decided to present a series of scientific concepts to enrich the experience of Topobo, but by his own admission, he got carried away with what he wanted to achieve, frustrating himself as well as the students. In the first half hour he used only the passives, looking to explore the systems' geometry and angles, wanting to instill an overall sense of 'engineering platonic solids.' Then he brought in the Actives and shifted to how the system could mimic molecular reactions, like breaking and creating chemical bonds. He described that upon first seeing Topobo, it immediately reminded him of a PBS special he had seen that showed DNA being spliced. In this vein, he wanted to teach chemical bonding with it, explore crystalline structure, and on a different scale, tensegrity. Dale figured out midway though the session that the material was too dense and presented too quickly for the students.

Dale's Conception of Topobo

Dale's sessions ignited both excitement at the possibilities of what Topobo could demonstrate and frustration at his own inability to immediately put them into action. At multiple times during our interview, he suggested the need for a teacher's guide which would provide advice on building creations that walked successfully. He was careful to stipulate that the guide should not didactically provide exact instructions, but rather that it should provide general design guidelines and examples on how to obtain a particular kinetic behavior, combining structure and programming. He described the guide as scaffolding for the teachers to gain a deeper understand of Topobo's possibilities, as opposed to a series of lessons plans to implement in the class. The guide should also feature common mistakes students make when working with Topobo, to keep teachers preemptively informed. Dale and Laurie also suggested running a workshop for teachers, possibly at an education conference, combining teachers of all disciplines.

Pedagogical Ideas

Even after limited initial exposure to Topobo, Dale, Lori and other teachers at the program were overflowing with curriculum ideas of what Topobo could be used for in the classroom. A language arts teacher suggested using Topobo to find the rhythm of poetry, almost like a metronome, programming a creature to move in a particular rhythm and asking the students to write a poem about this creature matching the rhythm of the poem to Topobo's. In addition to his ideas about chemical reactions, Dale mentioned that his 8th grade technology class made Rube Goldberg devices in which Topobo could be easily incorporated. "We could connect it to a ramp or some kind switch then we could set a whole bunch of other events in play." He discussed several scenarios for creating real world models for math and science concepts, such as parabolas, using a Topobo construction to knock a ball into the air, like an automated golf club, observing a parabola created in a real world situation. He also envisioned Topobo to be of use in discussing elementary circuit design: he wanted to figure out a way to create a logic relationship, like an and/or gate, between a Queen and the Actives. He struggled with how he would design it but had a sense that by mimicking a programming structure in a physical behavior, it could become more intuitive and easier to comprehend for the students.

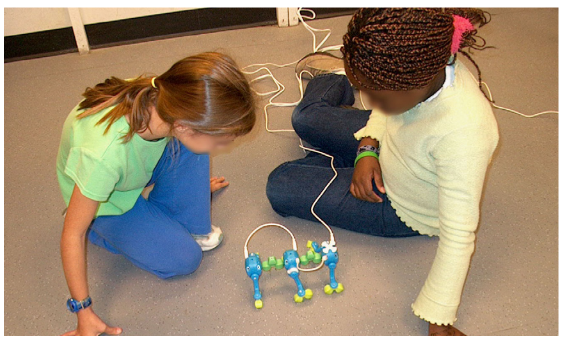
Discussion

Dale begins by using Topobo as a holistic system, creating walking creatures with his students, but soon transitions into a mind set envisioning Topobo as a tool for simulations ranging in scale and time: it becomes an enabling technology for kinetic behavior. This shift shows how Dale has come to recognize Topobo as a flexible and open-ended modeling tool. However, he recognizes the limitations in time and effort of putting those models to work in a classroom, "In general, education is something where you want the fastest and easiest solution, and if it's something you have to stretch your imagination to make something work for a specific situation, that's not something people usually do in a classroom."

Lori offered a more theoretical perspective on Topobo's suitability for a classroom situation, "What Topobo offers is that surprise element...It's intriguing just in its design and its newness, it has that cool factor... maybe I've been taught parabolas before but maybe now that I can make one happen with Topobo, it may sink in. Teachers have to teach and reteach and do it in different modalities and do it in different intelligences in hopes that you hit the one of every kid." She cites its novelty as a factor which can help draw students in, resonating with students of alternative learning styles, and references a multi-modality that is often a specific design principle of tangible technologies.

Elementary / Middle school science classroom

Jane, an elementary and middle school science teacher at a Montessori-inspired school, borrowed sets of Topobo to use in her 4th and 7th grade science classes for 8 months. The school had a hands-on approach to learning and she was accustomed to using manipulative materials in her classes. Our goal with Jane was to learn if Topobo could succeed as a formal educational tool: could it fit within a lesson plan, state educational guidelines and other constraints that teachers juggle daily in designing their class material.



Girls work with Topobo during a lesson on locomotion in a 4th grade science class

Putting Topobo to Use

Jane incorporated Topobo in her classroom in two ways, first as part of a lesson plan with a curricular goal with her 4th grade class, and second, as a free play activity (for recesses on rainy days) for both her 4th and 7th graders. Jane initially experimented with Topobo in her home and watched her own elementary-age children, nieces, and nephews play informally with Topobo. She tested some of her pedagogical ideas on them, and based on these observations Jane designed a formal lesson plan for her 4th graders about locomotion.

Jane's students worked with Topobo as part of a unit on structures. Lessons took place in two sessions. First, Jane isolated the activity of programming, and set up a specific task all the students could accomplish: children were given identical pre-built creatures and challenged to get the creation to walk 30cm, timing for speed. Jane focused on measurement and data collection as part of this exercise, as well as concepts such as friction, gravity and balance. The children expressed desires for free play and experimentation, and it was difficult to keep them focused on a structured task.

In their second session, students were shown video clips of Muybridge's horse [Muyb07] and walking robots as background material on natural and mechanical locomotion. They were asked to build their own four legged creature and make it walk a meter as quickly as possible, and describe the order of the leg movements. In building their own creations, a lot of kids started with a creature very similar to what they had used in the previous session. Jane explained, "its always easier to take a model and tweak it." Overall, she was satisfied with the children's success in the activity and Topobo had engaged the attention of her students the entire time, particularly notable with a student who usually displayed attentional disorder issues in extended exercises.

Jane also provided Topobo as a material for free play, during rainy or bad weather days. Deep engagement characterized students in her 4th and 7th grades. "They really, really, really wanted to play with it. It was unbelievably attractive as a play toy – whoever saw it, whatever the age range, from 19 or 20 to 8, people loved to play with it, but they had a hard time unless they had a model to follow." Topobo was more popular as a play toy than as an educational material for Jane, and this evidence suggests that attractive tools can reach students in school outside the context of formal lessons.

Discussion

Jane represents a teacher who has put in considerable time and effort to understanding Topobo's potential and being able to communicate it to her students successfully in the classroom. She described the time put in as essential for her own understanding. Knowing that she could make basic things gave her the confidence to teach it to the children. However, she still did not feel she had a deep enough understanding of how to start working with Topobo in more complex ways, nor as a teacher did she have time. "It would be really cool if I could make it do that, but I don't have time to figure that out." Jane was enthusiastic about her results using Topobo in her structures lesson, but did not use it for formal teaching again. She felt that one of the most important issues with using Topobo in the classroom was educating the teacher on how to think about Topobo and the opportunities it provides.

When asked if Topobo has a place in the classroom, Jane described her philosophy toward activities. "I go back to simplicity. It's the efficiency question, like the efficiency of straws and paperclips" to explore structures. Simple materials that are easy to work with can get a salient message across in a very direct way. While Topobo provides a certain ease of entry to use, the newness and novelty of the technology is actually a hurdle to identifying and focusing on underlying science concepts.

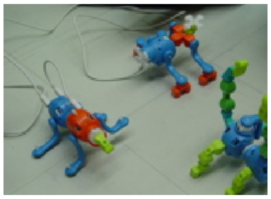
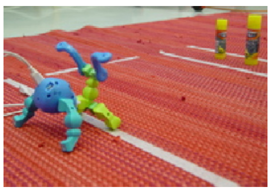
Like her students who found it easier to tweak the Topobo model she had built, Jane would have found it easier to tweak lesson plans we had provided her. Supplementary materials such as a booklet of basic constructions, and principles behind why and how they work (not just examples of full activities), would be very helpful to give teachers confidence to push forward with making their own activities for Topobo. This finding echoes Dale's comments from his experience in the after-school center. One challenge will be to teach sufficiently interesting and new ideas (or old ideas in new ways) so that the cost of learning the technology is outweighed by the benefits of the students using it. From Jane's perspective, it's hard to compete with the simplicity and economy of straws and paperclips.

After school robotics center

Several sets of Topobo were sent to Mary, an educational researcher studying the advantages and disadvantages of educational robotics for learning with normal and special needs children. Mary conducted her research in an after-school robotics center where children could participate in semester-long courses in which they could engage in somewhat unstructured play with technological tools. She requested Topobo as part of a study investigating how a robotics kit - and a tangible interface in particular - could benefit children in special needs education.

Putting Topobo to Use

Mary worked with two groups of children, one group aged 8-14 with mixed attentional disabilities including ADHD and Asperger's syndrome, the second, a group of kindergarten school children (non-special needs) ages 4-6. The study looked at 32 children in 13 sessions over a period of 5 months. Each child participated repeatedly

Creations and play by special needs children at an after-school robotics center

in at least 6 sessions, and Mary focused on collecting longitudinal data of children's uses of Topobo.

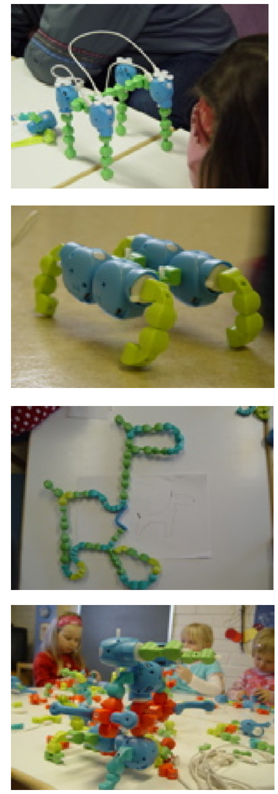
Both groups of children expressed immediate attraction to Topobo and they engaged continuously with it for long time periods (up to an hour), something very unusual for both populations. With special needs children, Mary found that Topobo kept them very focused but that they needed directed and guided tasks, such as small specific problems to solve or very detailed instructions to follow. With kindergarten children, all kids engaged with Topobo over long time periods (typ. 30-60 minutes) but some children needed initial scaffolding to understand the programming model.

For both groups, Topobo had a very easy point of entry, different from other robotic systems, and children could quickly and easily build what they desired because the system did not use a on-screen programming environment. Younger children and children who had difficulty with programming could still easily be successful at programming motions for their creations. Over the course of the study, however, Mary observed that Topobo was more suited to the kindergarten. It kept these younger children continuously engaged throughout the sessions, while the older children began to request added functionality such as sensors to build more difficult or complicated programs and scenarios.

Mary's conception of Topobo

As a classroom tool, Mary believed Topobo touches on a number of pedagogical themes including information and communication technology, mechanics, modeling of environments (interdependencies) and procedural thinking. Mary cited that her country's national curricula states that information and communications technology (ICT) should be integrated into all subject matters, but doesn't specify the tools. In this respect, she saw Topobo as a tool that could be integrated into many subjects with younger children. However, children didn't experience these pedagogical ideas directly from Topobo: core technology concepts would need to be introduced in other ways by teachers first, and Topobo could then becomes a concrete [Pia76] example of the concept.

One area in which Topobo excelled was in promoting collaboration and cooperation between students in both groups. She described that children would first build and program their own creations but then would share and try to program each other's work. They could then use the knowledge gained from each other's experiences to figure out how to make their own creations work better. Why did children collaborate more with Topobo than with other tools? She believed it was because Topobo was easy for everyone to use and understand: not only could a student easily create and program their own model, but they could also easily look around and understand what everyone else is doing. This transparency facilitated group learning and unstructured collaborative design processes.



Creations by kindergartners in an after-school robotics center after many weeks of play

Discussion

Mary had success with much younger children than in previous Topobo studies. Although she didn't believe that Topobo was necessarily more attractive to kindergartners than static manipulatives, all young children in her study engaged deeply with it. Where technology-related concepts are sought as part of a young child's experience, she noted that Topobo, with a tangible programming model, allowed for extended play and engagement with technology at a much younger age than systems which required screen-based (GUI) programming models.

Mary's conception, as well as her specific uses, of Topobo stress the importance of establishing in teachers a deep understanding of the system, in order for teachers to be able to present salient concepts to their students. She conceived of Topobo as a "computer" or "technology" system with which children could play with computer-related concepts. Mary sees Topobo as a technology to play with ideas similar to educational-technology work like Logo [Pap80] or LEGO Mindstorms [Res98].

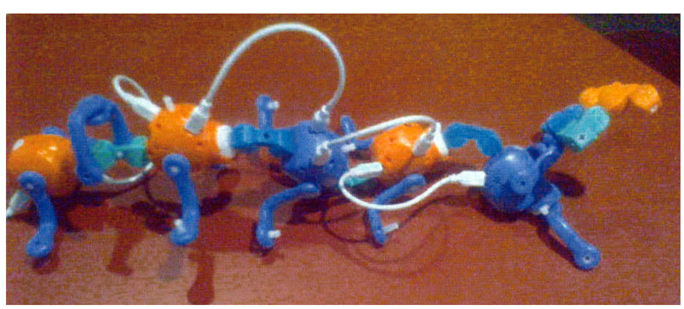
This indicates that tangibles may make certain common technology concepts accessible to children at younger ages than non-tangible technologies, as argued by Frei [Fre00]. However, in failing to identify concepts from biology which her students pursued in building creatures and investigating walking motions, Mary illustrates that preexisting conceptions of technology education can limit an educator's perspective on what technology is actually capable of teaching. If this is true, researchers in educational technology should focus on broadening the scope of themes that technology is "supposed" to teach.

Urban science museum

Sets of Topobo were loaned to a large urban science museum for four months. Topobo had been displayed at many exhibitions in the past but the interactions with visitors were generally very short and while the exhibitions may have been themed in areas such as innovation in play or robotics, no framework had been built around Topobo to guide its pedagogical context. Thus, sets of Topobo were turned over to teams of exhibit developers and programmers to find out how, or if, Topobo could be incorporated into their development process or inspire new experiences in informal education. Use of Topobo would be voluntary, based on interest in the system. Much internal discussion and two different scenarios incorporating Topobo on the museum floor emerged over a period of five months.

Topobo in 'Design Challenges'

The first group to work with Topobo was the development team for 'Design Challenges,' a program which features drop-in activities on the museum floor, staffed for 2 hours everyday, looking to provide "gender neutral non traditional engineering experiences." During the activities, children would build with provided materials to accomplish an engineering goal. The museum staff were present as guides but the focus was on allowing the children to engineers the projects on their own. The activities were planned for children aged 9-15. However, with the varying nature of museum visitors, a much wider range of children and adults participated. The team, led by Leah, took Topobo out on the museum floor for four sessions over a period of 2 months. The activity around Topobo was relatively unstructured but focused on making creatures walk, or if that was too difficult in the time frame, making them wave. She noted that visitors played with it for an average of 20 minutes, considered a very long time for a museum floor experience.



A 'space caterpillar' buit by a visitor and volunteer at the science museum's 'Computer Place' exhibit

Leah's conception of Topobo

When discussing the concept of the Topobo design challenge, Leah described what they had been investigating as biomimicry, attempting to make a connection to how animals walked. But she stated 'I don't think we went into it thinking that there was a science concept that we wanted to get across." She described their initial aim as showing people a new technology that they wouldn't get to experience somewhere else, citing Topobo's novelty as a big draw for museum visitors. The process of designing a 'design challenge' involved brainstorming a concept, prototyping solutions and narrowing the appropriate materials to make available, leaving the experience open enough to make four or five things that are totally different but can still accomplish the same goal.

If she were to design a deeper experience for a Topobo Design Challenge, she found the nature of Topobo as a well designed 'kit' to be a limitation, because the limited range of pieces could make it hard for students to arrive at diverse solutions. It had not occurred to her to mix Topobo with various other materials (cloth, LEGO®, etc.) as it seemed to go against the nature of the how the system 'should' be used. When asked if providing Topobo Actives that had the appearance of a raw motor, she thought 'it would feel like a material, a raw craft experience as opposed to a kit.' While the 'construction kit' might be seen here as a limitation, the attractiveness and completeness of Topobo's design also drew in a wider age group than their usual audience, especially younger children. They were not accustomed to running a design challenge that spanned such a wide age range.

Topobo in 'Computer Place'

Topobo was also incorporated into a staffed exhibit entitled 'Computer Place' whose goal was to introduce visitors to new computer technologies and present emerging computational concepts. Recently they had been moving into demonstrating robotics technologies, as this was seen as an emerging area in computation. Sonia, one of the program coordinators, brought Topobo into Computer Place for a week of continuously use. She and other staff would demonstrate Topobo and then allow visitors to build creations of their own. To visitors, she described the activity with Topobo as biomimicry, with the goal of "making a computer act more like an animal." In referencing Topobo, she also discussed concepts in computing such as programming (Topobo programming occurred with the body instead of code), networking, and swarm behavior, based on visitors' varying interest and engagement.

Sonia's Conception of Topobo

Sonia's relationship with Topobo focused on its identity as an emerging technology. Based on her area within the museum, the concept of teaching people about creating locomotion and biomimicry was an engaging experience which functioned as a stepping stone to draw people into a second and perhaps more fundamental goal of demystifying and teaching people about technology. Sonia thought it would be good to take Topobo apart, to show people what the sensors and motors look like, citing that they had a Robosapien® that was deconstructed and was very popular and engaging for visitors. As others had indirectly done, Sonia was directly tapping into the novelty of the system as one of its educational values. While this was clearly unintended in Topobo's design, it an interesting paradigm for researchers to consider how Topobo's identity will change as it (and perhaps robotics in general) transition into more commonplace technologies.

Discussion

In these two scenarios, and throughout conversations with other developers in the museum, it was evident that Topobo's novelty and 'cool' design was a big attraction in a busy space with many experiences vying for attention. But to make a system like Topobo successful in the context of the museum floor, it becomes necessary to constrain it. For tangibles to contribute to the museum experience, one guideline is to create an experience that is constrained enough so people can absorb an idea in under two minutes, and open-ended enough so that people can make the discovery for themselves. One approach may be to appropriate the Exploratorium [Exp07] model of exhibit design in which an idea is made accessible by providing many different exhibits that all isolate and provide a different way to "discover" an idea.

Graduate architecture school

Topobo was introduced to the teacher of a kinetic architecture course at a leading graduate architecture program. Similarly to the other scenarios, the system was presented as a way that students could explore motion concepts and provided to the professor for a long time period. Unlike the other scenarios, this professor did not try to specifically "teach" anything with Topobo, but rather provided it as a "material" prototyping motion concepts in designs of transformable and deployable structures. Because the teacher's role was minimal, this study focuses on the one student's experiences (Ray) as self-taught with the system, and how it was reappropriated for applications that diverge from Topobo's usual purpose.

Putting Topobo to Use

During a studio session, student designers experimented with Topobo in an open-ended fashion As part of the course, students were using the Arduino [Ard07] programming environment to control sensors and actuators, so they were accustomed to the idea of embedding kinetic behavior physically into their models. However, these students were more comfortable working with physical materials like foam core or paper than with embedded technology. Topobo thus became part of their hands-on modelling and design processes to quickly and easily experiment with movement in their models.

Ray incorporated Topobo as part of his own learning and creative process. Following his experiences during the class session, Ray continued working with Topobo over the following six months, utilizing it in the design stages of his Master's thesis project. Ray's thesis work involved the design of a conceptual transformable opera house set on Potsdamerplatz in Berlin. The building morphs between two physical states, representing two alternate realities: one represents its form in the 1980's before the Berlin wall fell, and the second fictional state represents the building as imagined if the Germans had won WWII.

Ray's Conception of Topobo

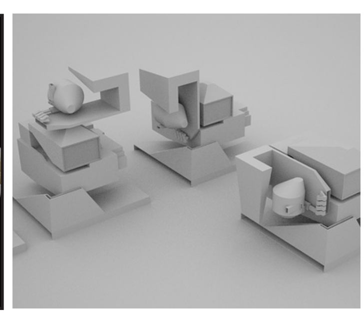
Ray used Topobo as a kinetic prototyping tool as part of the initial design phases for the project. He describes his process: "The most important part for Topobo for me architecturally has been toward

Ray explains his 'kinetic diagram' made with Topobo next to the final model of his thesis project

the use of diagrams. This model is a representation of some of the kinetic movements in the final project…I used it very early on in the project but as my building started becoming more spatial [modeled in detail & scale] the use for Topobo was eliminated. In the very first stage of a project,...Topobo was instantly these modular parts which I could bring into a kinetic state for discussion.

Ray used Topobo as one medium among many in which he communicated his design, with the most useful part for Topobo being early on in the research, "getting my kinetic idea across." When discussing the limitations of Topobo and why he had not continued to use it further along in his design process, Ray cited that he felt constrained by form factor, specifically the joints being a single degree of freedom which made his kinetic model bulky and spatially more complex as he had to offset each joint. As he continued with his design, however, he cited one wing of the building's mechanical design being directly inspired by this constraint, "[this area of joints] came about when I had to keep offsetting the Topobo and I noticed that the axis of rotation could be elongated." What began as a limitation became part of his design language.

Left, Ray's final thesis model and right, iterative joint models inspired by Topobo



Discussion

Topobo did not become part of Ray's more detailed design phases. While we had given him permission to modify the parts and embed them into his model, Ray preferred to begin 3-D modeling in a GUI as the next phase of his design process, "Physically I could take it apart and try to build a chip board model around it but that isn't the method I usually work in. I usually go straight to the computers, draw it in 3-D, send the file to the 3-D printer. It's just faster."

The advantages of the physicality and immediate access to kinetic behavior had now been outweighed with a more detailed oriented and familiar tool, 3-D modeling. However, the Topobo models Ray had made directly influenced many joineries in the final model. He found it useful to think about the design modularly, like Topobo, designing in segments and then connecting them with Lego-like attachments. It helped to work with a physical kinetic material first, when thinking about what would work mechanically in space before attempting to draw it on screen. The building took on a very toylike playful aspect to it, rare in architecture, which he felt may have come from his interactions with Topobo. Ray also used Topobo in one unexpected way, mapping the colors of the passives in different areas of his model to denote their spatial functionality, he described it as his 'legend.' The color mapping that began with Topobo continued into his 3-D onscreen model to become part of the design language in communicating the project.

Overall Findings

In addressing our original goals, we found it was not possible to analyze them separately; in every study, usage revealed interdependencies between context, age ranges and amount to time spent with the system. Together, these variables affected ways in which people worked with and conceived of the system.

Context of use

In all contexts - museum, classroom, after-school center, robotics center, graduate school - Topobo was regarded as a useful or provocative tool by the educators who worked with it. However, as a construction kit it seemed to excel in contexts that allowed for longer periods of engagement. Jane used it more as a play toy than a curriculum material. The museum asked to use it again, but in the context of a day-long activity. (They would like to use it in computer place, but in a more limited context, e.g. pre-built or somehow constrained in use.) Students and teachers in the after-school robotics center, who have more time to play with the technology, continue to work with it with success.

Time and Age

The idea of constructive learning or self-discovery came through in every context. As an open ended system the level of success with different age groups was directly determined by (a) the amount of time children spent with the system and to some degree (b) age. The longer kids may play with it, the younger they can be. When Mary used Topobo as a completely open-ended system, kindergartners (previously considered too young for such a complex system) engaged with it meaningfully if given enough time. Conversely, in the science museum, Topobo was used as a simple demonstration or inspirational piece (not at all an open-ended interaction with Topobo) with all ages, but visitors had only one or two minutes to engage with an idea. Somewhere in between we find Jane's example of providing her students with pre-built models, so that they might constrain their efforts on programming motion. Universally, less time to interact with the system required it be more constrained in scope.

Implications

Support for Educators

Perhaps the most consistent and salient message from educators themselves is that educators need prior experience with the system, to gain confidence in their ability to teach with it. Jane is a teacher who put a lot of time and effort into learning the system and developing a lesson plan so that she could confidently communicate and teach new ideas to her students. In contrast, Dale jumped right into a lot of exciting, but difficult concepts and ended up frustrating himself and his students. Clearly all teachers needed support, and creating one's own lessons is too difficult for teachers to improvise.

Educators all requested similar kinds of support: to be taught examples they could use in their teaching, but they must learn the underlying principles of the examples. Here, the format of the examples was not prescribed, but printed materials in the form of an instruction / activity book may have met many educators' needs. Such a booklet might be similar between a teacher's standard activity guide, but the computational aspect of tangibles requires a level of systems-thinking that is not often specified in teaching with static materials. Certain challenges will arise, such as representing dynamic information (like movement) using a static printed page. Perhaps the booklet would have a companion on-line component of animated examples.

Inspiring the Use of Toolkits

Many researchers like to develop "toolkits" that can be appropriated by teachers or students in a variety of ways. This contrasts with an interface designed to make a specific idea or application salient. For toolkits like Topobo, it seems especially important to provide educators with an inspirational example of an application scenario. Nearly everyone in our study was interested in making small robotic animals walk, and this provided both an emotional and a pedagogical "hook" to get people started thinking about and working with the system.

The inspirational scenario did not confine the range of ideas people explored with Topobo. Sonia and Mary saw Topobo as an entry to more general computing concepts like networking and communications; Jane compared the system to materials like straws and paperclips (suggesting a general view of it as a material rather than an application); Ray actually used it as a prototyping material in a unique context; Dale envisioned learning conic sections and logic with the system. These digressions from the inspirational example of walking robots encourage us that toolkits can be reappropriated (which allows a user to get more out of their investment in the tools), but we believe the inspirational example application (walking robots) was critical to engage people's interest in the first place.

Tangible Interfaces for Learning

Dale's conceptions of investigating DNA, parabolas and logic principles suggest that educators are seeking the things that tangibles are already working toward: a more transparent programming and control structure, the ability to physically play with math and science ideas, and putting in people's hands the dynamic simulations that are increasingly an important part of scientific teaching. Mary's observation that transparency allowed collaborative work further supports teachers' goals in constructivist education. In terms of this transparency, accessibility and ability to model dynamic processes, the tangibles paradigm seems an obvious fit to education.

Some Comments on Design

Topobo's highly refined physical design helped it succeed with a broad range of educators in such a hands-off manner because the parts were robust, reliable and approachable. However, the novelty of the system has both pros and cons: on one hand, its uniqueness invited people to explore and play with Topobo, catching people's attention in competitive environments like the science museum. But on the other hand, it is equally valuable to make tangibles seem "familiar" by referencing existing products and interactions. Familiarity allows the researcher to more quickly test the reactions and interactions of a seasoned user.

Summary

Our original goals set out to identify contexts for success of Topobo, the time period and evolution of children's engagement, how age range predicts experiences with Topobo, and contexts and approaches other educators will bring to the system. In addressing the original goals of our study, we found it was not possible to analyze them separately; in every study, usage revealed interdependencies between context, age ranges and amount to time spent with the system.

In all contexts - science museum, classrooms, after-school center, robotics center, graduate architecture school - Topobo was regarded as a useful or provocative tool by the educators who worked with it, and the idea of constructive learning or self-discovery came through in every context. However, as a construction kit it seemed to excel in contexts that allowed for longer periods of engagement. In general, younger children want and need more time with the system than older ones, and short interactions (with any age user) demanded more constrained activities. Perhaps the most consistent and salient message from educators themselves is that educators need prior experience with the system to gain confidence in their ability to teach with it, and would have liked more complete teaching support materials.

Educators' comments and use of Topobo demonstrated that they are seeking the things that tangibles are already working toward: a more transparent programming and control structure, the ability to physically play with math and science ideas, and the ability to put into people's hands the dynamic behaviors and simulations that are an increasingly important part of scientific teaching.

8 Climbing a mountain of ideas
Applying Multi-Layered Abstraction

How can children climb a mountain of ideas by playing with new kinds of interactive toys? And how can those toys be designed to coevolve with children, meeting children's interests and needs as they develop? With many existing computational construction kits, knowledge of physical modelling does not give a child many ideas about how to represent those ideas computationally. There are many layers of abstraction that separate the physical and computational realms. With Topobo, I try to leverage children's existing work and knowledge with the system to provide a foundation to play with more advanced ideas. I have designed the Topobo system to leverage a "multi-layered abstraction" that stages ideas so children can discover them as their personal interests and cognitive skills evolve.

Revisiting Bruner's theory

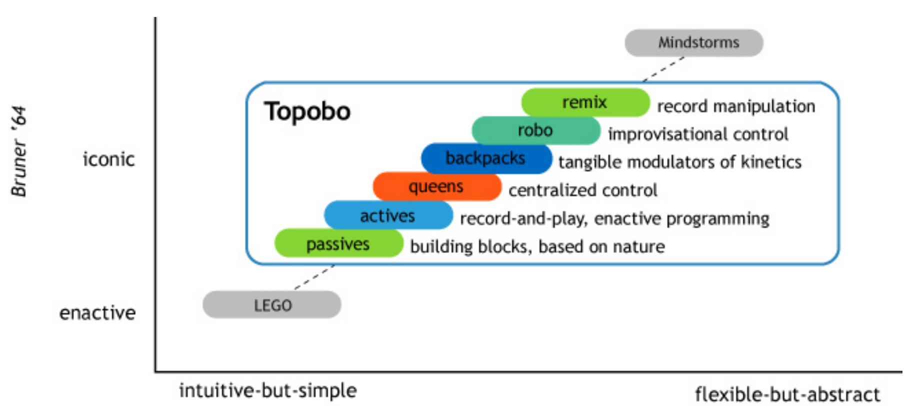
Manipulatives are about playing with physical objects to inspire children to play with ideas. The concept of "multi-layered abstraction" is concerned with the notion that a child's physical model-making inspires, or helps them construct mental models. How can a system be designed to support both? The answer, I think, centers on two related sets of questions. The first is functional: what does the system do, and what can I build? How can creating a simple physical or behavioral model give you ideas about creating a more complex physical or behavioral model. Put another way, what does playing with the system suggest or inspire you to pursue? The second sets of questions is conceptual: how do I understand the system? What ideas and concepts can I learn or understand by playing with the system? How does playing with simple ideas prepare to you play with more complex ones? In the design process, these two sets of questions cannot be separated—they are part of a system of interrelated design concerns.

Jerome Bruner's ideas about children's progression through enactive, iconic and symbolic representations of ideas underlie the pedagogy of the Topobo project, and I believe they provide a theoretical foundation for the development of other hybrid physical/digital construction kits. They might be summed up as:

1. Ideas can be rooted in enactive processes, like snapping together bricks, and physical programming. In the physical domain, the physical and digital processes are linked.

2. Children explore their ideas by representing them in iconic forms. Children can see their structures behave (both physical and computational) to demonstrate and evaluate their knowledge.

3. More abstract ideas are developed in the system as a symbolic "language" that has its own vocabulary, grammar, and structures.



Topobo components leverage "multi-layered abstraction" so kids can progress from concrete to abstract ideas without abandoning the tools (and ideas) they are already using.

How complex is Topobo? Complexity is layered into the system.

Topobo has 12-fold symmetry and can support free form or patterned building. Green Tetras can be connected flat into a pentagon (above), or notched together into a tetrahedral (right).

These structures - whether physical or computational - are flexible, e.g. capable of recursion and other advanced linguistic structures.

Enactive and iconic representation occurs through play with Topobo [Bru74] in the following ways: during model making and programming, the body's reflexes are physically engaged in the learning process. Building a model creates an iconic representation that a child may reflect upon. This reflection can then support children to classify and symbolize those ideas. Backpacks, Remix and Robo can help a child identify and play with an core system concept, and further reflect on their theories.

A sort of intellectual "dialogue" emerges between a curious student and the manipulative tool. In this dialogue, the student brings questions and biases, and the tool's design affords opportunities and suggestions. With Topobo, this dialogue might focus on the general question, "what can be learned by building with motion?"

Multi-Layered Abstraction

While I have designed individual Topobo components to seamlessly function with the entire system, I have often thought about different categories of components - and the associated kinds of models you can build with them - as "layers" of abstraction that range from simple to complex, and from concrete to conceptually abstract. I do not mean to imply that these layers are hierarchical or should be introduced to children individually or in a particular order, but rather that all users will progress from simple to more complex models. A child's progression with the system can in turn support their transition from concrete to abstract ideas. I describe the system as "layers" to provide a framework for different ways that Topobo might be used, and be thought about. To revisit the idea of a tangible language, I find it useful to think of Topobo components as an "alphabet" with which children can sculpt increasingly complex structures, behaviors and stories.

Layer 1: Basic Building — Nouns

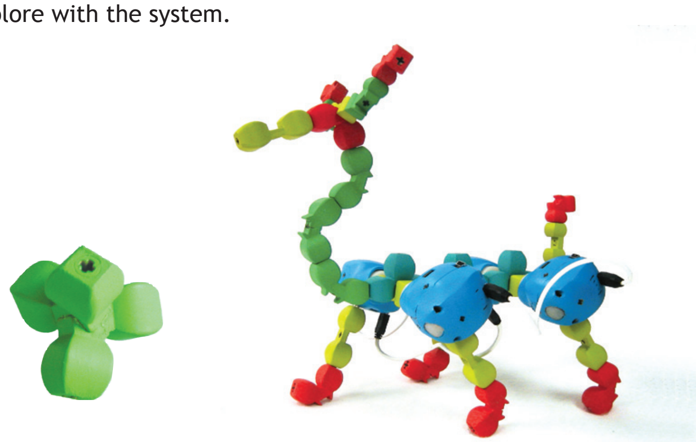
When children assemble Topobo components (the alphabet) into a character with an identity, they have made a "noun." What are the implications of creating nouns?

Until age 7 or 8, children cannot distinguish between photos of flat and 3-D objects. These children could begin a Topobo activity on a flat surface and then "fold up" their creation to become a 3-D object. For children who are becoming more adept with 3-D visualization, play with Topobo can convey some ideas about the 3-D crystal geometries that underlie the system design. These children can make discoveries about possible spatial loop constructions and learn about certain patterns in crystal growth (3-D geometrical regularity). The notch design is intended to help children begin to transition from flat drawing to 3-D branching, and the overall system is designed to encourage children to experiment with anthropomorphic types of forms.

Programming Topobo adds action – the realm of verbs – to the construction toy.

The physical system should function well on its own, having breadth and depth of possibilities. In our observations, some children played exclusively with the Passives, and some children introduced the system first as Passives only. As children and adults repeatedly play with Topobo, they discover more spatial patterns and develop more sophisticated ways to assemble the toy.

Layer 2: Building with Motion — Verbs
When children animate their creations, they give them action, the realm of verbs. Topobo programming is a simple and accessible way for children of many ages to experiment with kinematics. Children as young as four can successfully work with Topobo Actives, and compared to the Queen, the younger children found direct manipulation to be more "magical" and exciting than the Queen. Eighth graders, when faced with the task of making something walk, were challenged by the complexity of managing dynamic balance. Motion, like static building, has a range of complexities that the child can explore with the system.



A tetrahedral is a coomplex form. A griffin has complex movement. Topobo allows people to explore complexity in different ways.

Older children create more sophisticated structures. Two eighth graders designed a moose (left, and rebuilt below) and programmed it together to coordinate its 2 DOF walking motion.

Young children begin by programming a single active, for instance making something wave. Children struggle with the fact that Topobo often behaves differently on a table (reacting to gravity and its own momentum) than it does is one's hands (constrained by one's body). Young children's creations are sometimes large and sculptural explorations of passives with actives to make parts wiggle or move abstractly, without consideration of coordination between Actives' motions. Sometimes they have only a single Active, such as one Kindergartner's "no walking man" and another's walking "refrigerator." Kindergartners see Topobo as a synthesis of animals and machines.

As children develop with Topobo, they begin to investigate relative motion between actives. One second grader sets two actives, connected head-to-head by several passives, moving on a table. She programs and watches, programs and watches, investigating why they move differently on the table (rocking back and forth gently) than they did in her hands, moving wildly from side to side. She may begin to notice that they are moving one another in a balanced way because her creation is symmetrical.

Children leverage enactive knowledge as a foundation to work with Topobo Actives. When two boys struggle to make a four-legged Griffon walk, one boy gets on the floor and begin crawling. He investigates the movement of his limbs as he crawls, counting the sequence of his hands and knees on the floor. "I got it!" He gets up and teaches his friend how he crawled, and they work together to program the Griffon to have the same the timing in its legs. This boy's enactive knowledge - knowing how to crawl - was a concrete foundation for him to create a successful walking program for his Griffon.

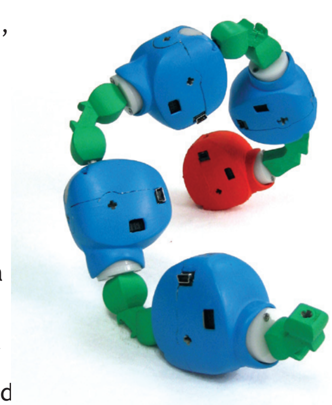
When children work with more actives, they investigate movement in multiple degrees of freedom, which requires attention to coordination between parts and system behaviors. In general, as the child grows, the types of structures and the types of motions can become more advanced as creations develop from abstract, sculptural creations, to (perhaps) more goal-oriented play activities like ambulatory motion. Scale introduces new problems to the child. Most children reported that using one Active was exciting and interesting, and adding more Actives (and their associated degrees of freedom) became much more complex. Understanding movement in multiple degrees of freedom is, in itself, a complicated problem that even adults often find difficult.

Layer 3: The Queen — Pattern and Repetition

Queens introduce temporally coordinated motions, and require that children begin to plan for interrelated forms and motions. The results of the Queen's behavior are even surprising to adults, so we believe there is much to be learned by using a Queen. In many structures, limbs will bump into each other when using a Queen, so children must begin to experiment with organizing their creations' forms in more planned ways. Queens force the child to begin to think about complex problems of spatial translation, such as creating a scissors motion with two legs of an animal. As mentioned in the earlier evaluation of the Queens, they also seem to require a different conceptualization of control, since their motion is based on absolute position rather than relative position (the way people usually think about direct Topobo manipulation). Therefore, using Queens requires more planning or experimentation to be successful.

Children can use Queens to play with flat and 3-D geometry, e.g. creating circles and helixes, and sophisticated coordinated motions in walking creatures. Children also use Queens as remote controllers. Children see the effects of their program as they create it, allowing them to debug their programs while they are being recorded. Queens are direct, because they are programmed by demonstration, but introduce an abstract idea to Topobo, the idea of "copy."

Small movements to the Queen (below) make this flat ring dramatically extend like a spring. This local/global relationship can give insights into the workings of a real spring.



Layer 4: Backpacks — Adverbs

Children who are adept with manipulating abstract ideas want to manipulate their recordings in different ways. Backpacks are like "adverbs" that modify a Topobo creation's action. Backpacks provide a layer of complexity for children who can begin to understand the roles of local motions in a globally moving system. Backpacks also physically instantiate computational variables. Because Backpacks modularize and

Backpacks can focus one's attention on local behaviors within a globally dynamic system. They are designed to be like physical functions, or "adverbs."

make tangible some of the fundamental invisible properties that affect robotic motion, Backpacks increase the complexity with which children can design and understand their creations.

Kim's experiments

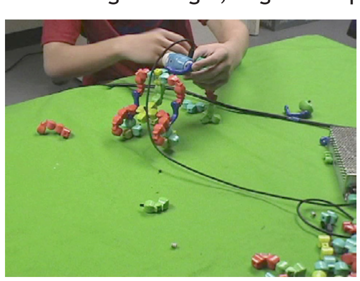
Speed is adjusted, and Kim expects her galloper to walk faster when she speeds up the motion. To her surprise, the galloper walks better when it is slower, because she has found a resonant speed for the creature to oscillate forward.

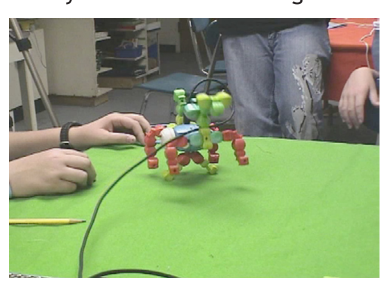
Playing with knobs can provide a foundation to play with sensors. Kim later builds a symmetrical walker, based on her galloper. She attaches a Time Delay Backpacks to one Active to change its phase relative to the other Active. By turning the knob forwards or backwards, kim notices she can make the creature walk forward or backwards. Building on this knowledge, she replaces the backpack with one that has light-sensing eyes, and makes the creature walk towards or away from light.

Hidden State

While some people have suggested a "syringe" metaphor in which information is "injected" into the Active, such a design leads to invisible information that must be remembered by the user (sometimes called "hidden state"). The physical embodiment of the function in the Backpack is intended to make Backpacks more accessible to younger children by making the idea of computational "state" tangible.

Although tangibility can improve the accessibility of the Backpacks to younger children, they may still be challenging for some kids because they all have abstract qualities. Time-delay Backpack introduces non-intuitive changes in temporal phenomena that are hard to visualize (perhaps because they are akin to a ratio, a comparison of two parts). Bigger-smaller Backpack can be confusing because as a motion gets larger, it gets "clipped" by the mechanical range of





Backpacks do not always create predictable movements. Children must experiment to develop an intuition for the rules underlying a creature's behavior.

the servo. Since the motion can only reach a finite size, in situations where the original motion is already large, the "larger" motion appears to be "faster" rather than "larger." The faster-slower Backpack results in surprising behavior also: a faster motion naturally exhibits low-pass filtering of the motion that causes some large motions to appear smaller: with faster motions, the motor cannot move fast enough to represent all of the recorded movements, and large motions can disappear. When children use all of the Backpacks, they are encouraged to formulate hypotheses for the unusual behaviors.

With the same motion, children can experiment with how Backpacks can affect a global behavior. A Queen leads to a circle. Adding a Bigger/Smaller Backpack creates a spiral. Using a time delay backpack forms a wave.

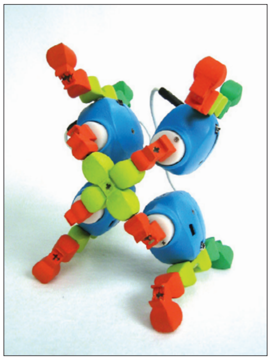
Queen + Local Backpack

Children who become more adept with understanding the outcomes of the Queen may begin to use it as a foundation for highly controlled motion. For example, a Queen can be used to create identical, coordinated motions in an ant, and then a Backpack can be added to the ant's thorax to shift its phase (time-delay) and amplitude relative to the rest of the creation. With some experimentation, highly tuned results can be created by coupling Queen motions with Backpack modifications.

Distributed Backpacks

When a Backpack is attached to a Queen, it behaves as a distributed Backpack. Like a game of "telephone" where a message is passed from one person to his immediate neighbor, this is an algorithmic behavior where the Backpack's effect increases each time the message is passed from one Active to the next. The distributed Backpacks are inspired by natural systems like waves and nautilus shells that change or grow as a result of local rules and interactions. Distributed Backpacks are intended to give the child some understanding of the nature of information behavior as it applies to concepts of growth and morphological change over time.

Understanding the distributed Backpack requires visualizing the spatial translation of the Actives, the effects of coordinated motions on a structure, and the effects of change in the network topology



A user created a novel form of robot locomotion with Topobo while trying to navigate on a slippery surface. This creation does cartwheels.

(which may be different than the physical topology of the creation). Therefore, I believe them to address some of the most conceptually advanced ideas embodied in Topobo.

Students can use distributed Backpacks to experiment with certain mathematical concepts related to series, growth, and wave motion. If a Queen is attached to a linear structure of Actives, gradual rotations to the Queen will cause the Actives to curl into a circle. With distributed behaviors, a time-delay Backpack can exhibit wave like motions in this same linear structure. The amplitude Backpack will cause this linear structure to curl into a flat nautilus spiral. Faster-slower can show harmonic resonance among Actives' motions.

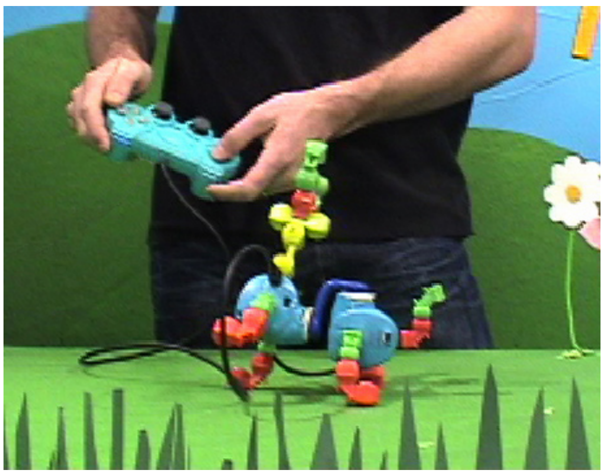
In less regular structures, distributed Backpacks can be used to coordinate interesting global behaviors of Actives. Bigger-smaller Backpack can be used in free-form ways to scale motion in a branching structure like a tree, where motions get bigger as they get closer to the limbs. Backpacks can be used in parallel to affect, for example, both the phase and speed of the motions in a complex structure like a caterpillar. Using multiple power cords, children can create a single creation that has sub networks that are governed by distributed Backpacks. For instance, a centipede might have one network controlling the oscillations of its body and another that controls the wave-like undulations of its many feet. Coordinating the two motions relative to each other could lead to a robust and interesting centipede robot.

Global and Local Backpacks

Global vs. Local Backpack can also be confusing for younger children, as there is a level of invisible information in the Backpack's internal state change. Further studies will be necessary to determine the age appropriateness for this feature. We may find that separate Backpacks, for example one "local amplitude Backpack" and one "global amplitude Backpack," are more accessible to younger children. While all of the above "problems" can be viewed as limitations to the design, they are also basic phenomena that are common to many different systems; these problems may turn out to be valuable lessons to learn.

Layer 5: Robo - Sentences and Paragraphs

Using Robo, children can sequence a number of actions together to create a story, similar to a writer's sentence or paragraph. Where the Backpacks allow a child to peer into the specific behaviors of a creation, and understand how invisible parameters can be adjusted to affect the local motion, Robo instead addresses issues of composition and control, and addresses how social motivations can encourage children to refine their learning through play.



With Robo, children can begin to construct stories and do performances with their Topobo creations.

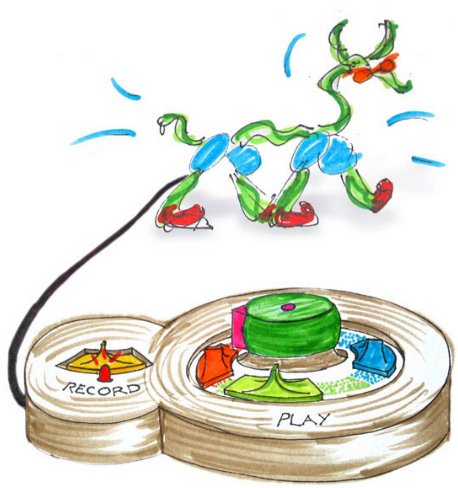
Robo also has an advantage as a remote control: Backpacks can inadvertently change the geometry and balance of a creature because they must be physically attached to it. Also, a creature must be picked up and interrupted to adjust a Backpack parameter. Robo allows children to tweak parameters like speed, scale and direction remotely, and they can more quickly debug the final effects on the robot's behavior.

Robo provides a greater degree of abstraction because it is a *remote* controller. Its form and design gives little indication of its functions, except to signify that it can be used as a controller. This simple fact focuses users on issues of control: how a creature be designed to produce a predictable and repeatable behavior? As discussed earlier, transitioning from simple-but-quirky behaviors to controllable ones requires a deep level of understanding from the designer, and Robo can support learning because it encourages both physical and mental stepping-back to reflect on one's work [Ack96; Ack99; Mar03].

Robo supports users to save multiple records, which may represent different moods (e.g. happy, sad, scared), or physical functions (e.g. walk forwards, backwards, defensive maneuvers) and encourages users to ascribe specific meaning to motions, and to design motions in response to desired meanings. Robo also encourages children to situate their creations within meaningful contexts like a competition or theater performance, which can support their affective relationships to their models. Papert called this "ego-syntonic" [Pap80] and the simple idea is that when children care about their work, they will learn it better.

Layer 6: Remix — Editing

With Remix children can permanently modify Topobo actions ("verbs") and encapsulate series of actions – whole sentences – as a single record. Thus, it can be regarded as an editing tool, one for meta-composition. Editing commands, like copy-and-paste are a hallmark of computers, and computational thinking. There is an opportunity to segment, organize and reuse pieces of compositions to make something new. We find it in nearly all of today's computer interfaces, from the word processor to image editor, and flexible segmentation and reorganization is also a hallmark of computer languages, in which procedures or processes can be manipulated in ways characteristic of language.



Remix concept sketch proposed a physical coupling between Remix and Topobo.

While Topobo Passives and Actives have their own spatial grammar, Remix adds a level of compositional grammar with which users can begin to structure their gestural programs. As discussed earlier, children can use Remix to copy records, concatenate several recordings, or permanently record modulations to a program's speed or direction. While the interface lacks the formal grammatical characteristics of discrete programming languages like Lisp or C, it adapts some of the real-time performance-based structures that have become powerful in music composition, e.g. record turntables and sampler/sequencers. I propose that Remix raises the ceiling of complexity with Topobo because it allows children to consider their gestural programs as data that can be saved, manipulated and organized.

Remix introduces abstract mappings of tokens to records, and allows users to edit and manipulate their gestural recordings.

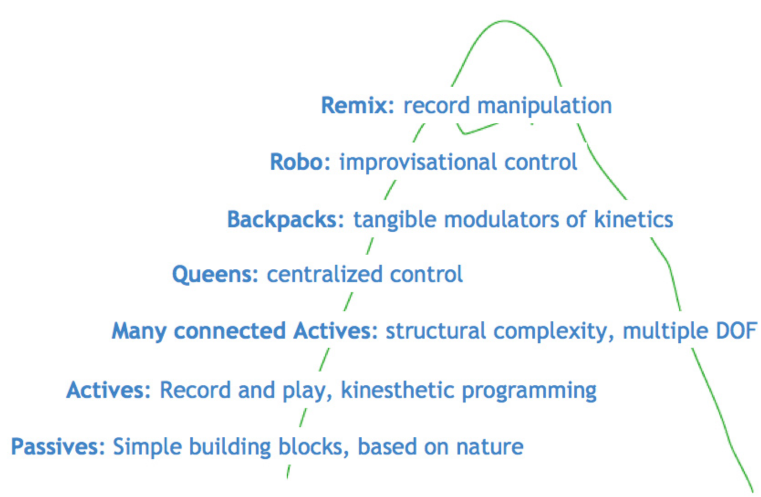
One key design criterion is that Remix builds on children's existing play patterns, and appeals to people's notion of the recordings as continuous (analog) elements.

In the tangible realm, I believe Remix already pushes the limit of complexity one would want to pursue. One reason is that physical interfaces become large and unwieldy when they become more physically complex. Another is that they cannot easily change their form or appearance to resolve ambiguity about what they may represent. That is, a user must remember that a Red Token means "walk backwards." Lastly, while Remix users found the tool to be flexible for record manipulation and seemed more able to refine their programs with it, the interface itself demanded a fair amount of a user's visual attention, distracting them from observing their Topobo model itself. As interfaces become more flexible (and more abstract), they compete with tangibles in a couple ways: first, they require a different mind-set, one that is based on visualization and abstraction of data. Second, since they demand one's visual attention, they draw one's gaze away from the tangible model.

Despite these limitations, all users in my studies reported they would not prefer a GUI to Remix because Remix was more consistent with the Topobo's tangibility, and because they liked not having to "use a computer." This illustrates that Remix did not seem limiting to users, and that they conceived (and related to) tangibles in a different way than to a GUI. This difference in mind-set is central to my thesis, since it supports the idea that even abstract controllers can fit within a tangible paradigm, by exposing processes and patterns that are already inherent in the tangible system.

Summary: Climbing a Mountain of ideas

In order to remain relevant to children at varying educational levels, I have designed Topobo to have depth, or complexity, in two different ways. Each element of the system is designed to be easy to use but, in itself, can represent a large range of simple and complicated ideas. For instance, the Passives alone allow children to build straight lines, flat shapes, 3-D branching structures, and 3-D spatial loops. While there is a range of discoveries to make with the Passives, coupled with the other system components, these elements combine to create countless more options for the child. With Backpacks, Remix and Robo, children can abstract the idea of gestural movement into data that can be adjusted and controlled. This design is intended to allow children to use Topobo to help them transition from concrete manipulation to mental manipulation of abstract ideas, using the interfaces to discover new ideas as they develop.



Climbing a mountain of ideas: How can interactive toys reveal new ideas through play, and coevolve with a child?

9 Beyond Tangibles
Raising the ceiling of complexity

This thesis has suggested a return to tangibility in digital manipulative design, both in a manipulative's interface, and in its control structure and programming. Our focus on tangibility has pervaded the system's design and the way we have used it, and we feel much can be learned through physical interaction with an actuated modeling system.

Nonetheless, bodies of work have explored how GUI software can engage children in learning and how coupling physical manipulatives and digital programming can support constructivist learning. This section will explore how both tangible and graphical extensions of Topobo can support children with diverse learning styles and cognitive levels, without sacrificing the core benefits of the tangible interface.

A Higher Ceiling? The limits of complexity with physical programming

A certain amount of complexity emerges from working with the dynamics of physical systems. Another type of complexity lies in the control of a machine, or in its program and control structure. This latter domain concerns computer programming, and while we consider recording motions to be "programming" Topobo, and using Backpacks, Remix and Robo to introduce advanced concepts related to programming, Topobo provides a very limited form of programming.

Topobo Programming?

Topobo inherited several traits from curlybot, including tactile programming-by-example and a simple one-button interface. While Topobo lacks the flexibility of a text based programming language like C or LISP, its coupling of physical construction and gestural programming does expand the possibilities that can be explored with programming by demonstration. Assembling several Actives in a single creation is equivalent to creating parallel programs that are linked in time, and the coupling of movement with a 3-D modeling system allows children to explore the surprising interactions between simple parallel programs and complex physical motions. Like traditional manipulatives, Topobo can be used in an unstructured way for play and discovery, or within a specific play activity the system can be used to teach ideas about physics (balance, mass, center of gravity) or about mathematical functions (series explored through Queens and Backpacks).

While the Topobo interface allows loops, some "object-oriented" control via Queens, and some functions and feedback loops via Backpacks, Remix and Robo, the interfaces' physicality limits the complexity of its control structure. One can create complex and coordinated motions with Topobo, but one cannot control them in a sophisticated and complex manner. This limits the types of activities one can do with Topobo, and thus limits its appeal to different children who are attracted to different kinds of activities. While basic building is fun for some kids, enabling different kinds of control structures could allow Topobo to support more kinds of learning and to engage more types of learners.

Floors, walls and ceilings

Papert stressed that a constructionist activity should have a low floor (easy to get started), and high ceiling (room to grow in complexity and abstraction). To this, Resnick has noted the benefit of "wide walls," or the ability to create many different kinds of models with a system. As argued by Frei [Fre00] and confirmed in our longitudinal user studies, Topobo – and perhaps tangibles in general – indeed have a low floor. Four year old children who are just beginning to work with manipulatives found Topobo to be accessible and engaging over long periods of time. However, what of the system's "walls" and "ceilings?" Perhaps Topobo's walls are wider than other manipulatives such as basic LEGO bricks or pattern blocks, because children can model a wide range of natural forms and behaviors with the system, but the system is not intended to capture the diverse range of activities that is a goal of symbolic programming languages. Compared to more general-purpose programming languages such as LOGO, Topobo has much more narrow walls. And what of the ceiling of complexity? Queens, Backpacks, Remix and Robo do indeed introduce abstract ideas, and let children play with rule-based behaviors, but the system lacks the flexibility that is characteristic of real symbolic systems like natural or synthetic languages, e.g. math or computer languages.

Benefits of a purely tangible interface

The question of how or whether to integrate GUI and TUI has pervaded research in Tangible Interfaces from its inception. Coupled with tangible interfaces, graphics can open new avenues of exploration with tangible interfaces. However, purely tangible interfaces can be elegant and compelling, despite their limitations.

In this thesis, I have focused on Topobo activities like creating walking creatures because these operations are representative of physical, bodily operations that are inherently hard to understand via a graphical representation. Topobo is successful at helping people understand these processes precisely because the interface is intimately connected to one’s body knowledge, the physical world, and one’s kinesthetic intelligence. Many people have commented that part of the magic of Topobo is the absence of a computer (screen + keyboard) in the interface. I think part of the basis for these feelings is that people appreciate the physicality and immediacy of their interactions. I believe some of these affordances could be lost if the focus of one’s attention turned from the physical phenomena to graphical representations or other expressions of a GUI environment.

Beyond Tangibles – Questions of Literacy

We might think of the jump from tangibles to symbolic systems like the jump from oral to written language. Oral language provides a foundation to think about what might be written, but successful writing requires learning a whole new set of skills (forming letters, proper syntax and grammar, etc.) and learning how to take spoken ideas and transform them into carefully-crafted words that have context, flow and structure that marks "good" writing. Thinking of this analogy, I notice that spoken language always precedes written, and knowing how to use one can tell you a lot about using the other one better. In designing tangible and symbolic systems, we might begin by thinking about what we can "speak" or "do" most naturally in the tangible domain, and then invent a symbolic system that captures these ideas. The jump from tangible to symbolic will be easiest and most powerful if the symbolic system is a refined characterization of ideas that can be explored tangibly.

When to integrate Topobo and a GUI

Topobo is designed to be scalable, and an added layer of complexity can come from Integrating a GUI with the system. This approach could open new avenues of discovery with Topobo and might allow people to use qualitative discoveries with the system as a basis to inform more formalized representations of a creation's dynamics. Furthermore, it could leverage the rich library of software that is designed to analyze and manipulate 3-D forms.

Compared to programming languages, the Topobo system (and Remix or Backpacks design in particular) is most similar to dataflow programming models like Max/MSP or Puredata, and is less similar to discrete models like Lisp. Dataflow languages are commonly considered to be easier to learn, but allow for less scalability and flexibility than discrete languages. I suspect the easiest way to transition from Topobo learning to a symbolic system would be through a GUI-based Topobo language that brings ideas from Topobo, Backpacks and Remix into a dataflow paradigm. If Topobo remained an *interface* for both physical input and output of data, rather than becoming a *display* for output only, the system would retain many of the benefits of tangibility while gaining some of the benefits of sophisticated computational models. This broad goal may have a number of applications.

Topobo + GUI may be better for older children

Coupling a GUI with Topobo may be an effective educational "bridge" for older children (11+ years) to transfer knowledge learned with Topobo to other fields of study. For example, much older students who are beginning to use symbolic math to understand dynamic systems may encounter the wave equation, and use Topobo to compare mathematically derived waves to wave motions that are created with the Queen and Time delay Backpack. If a GUI could represent the Topobo wave (and other behaviors) with symbolic math, children may be able to use Topobo lessons as a basis to learn this calculus lesson.

Similarly, by playing with the Queen and Amplitude Backpacks, children can create different kinds of spirals. Comparing the function of this distributed behavior to the mathematical expression of the spiral of Archimedes and the equiangular spiral could help children more deeply understand why these forms occur in nature. It can also help them to gain an appreciation for the relationships between these two representations.

Topobo + GUI for robotics design

Topobo could become a tool to study robotics. It is already a good tool to "sketch" robot ideas and to discover different kinds of robot locomotion. Adding a more refined layer of control via a GUI could allow people to further develop their robot "sketches." Much as graphic designers often scan and trace their pencil sketches, robotics designers may "scan" their Topobo creations with GUI software as a basis for a more refined design. This model should support two-way interactions, where editing can be done with the physical system or the graphical system and the two systems remain consistent.

One added benefit from such a system is that Topobo would become a 3-D physical display. A designer could use the system to rough out a motion with her hands, fine tune those motions on screen with mathematical models or best-fit examples, and then observe how the edited motions actually behave in the physical world.

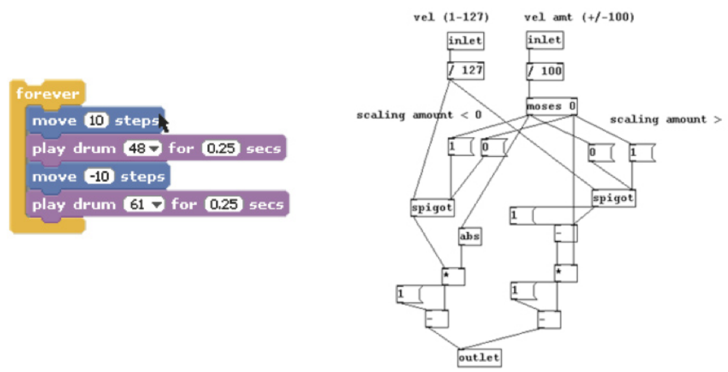
Topobo + GUI for motion capture and animation

Using the above infrastructure, a user could also use Topobo for human motion capture. One could attach Topobo to the body and capture one's movements with a PC. The captured data could be used to animate computer modeled characters or to drive a smaller scale version of a Topobo creation (with the creation directly mimicking the creator).

Conversely, one could experiment with using standard motion capture data derived from real animals' movements to drive a Topobo creation. This could lead to interesting discoveries as the user compares the compatibility between a real animal's movement data and a synthetic, robotic interpretation of that animal. Such a study could lead to a refinement of the robot, or insights into the dynamics of natural motion data.

Topobo + GUI for math and system behavior

Resnick [Res98] and others [Klo02] have evaluated how the programming language Starlogo can help kids to learn about the behavior of decentralized dynamic systems. By creating computer models of such systems and observing the graphical output of the programs, children ages 10 an older were able to develop intuitions for how the move-



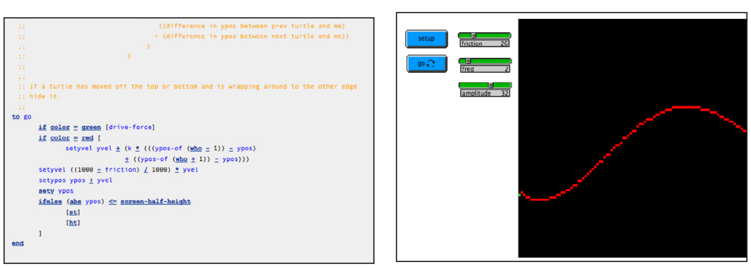
The Scratch procedural programming language (left) is imagined as a tool to specify backpack behaviors.

However, dataflow programming languages like Puredata (right) might require fewer conceptual shifts from the way Topobo and Backpacks already behave.

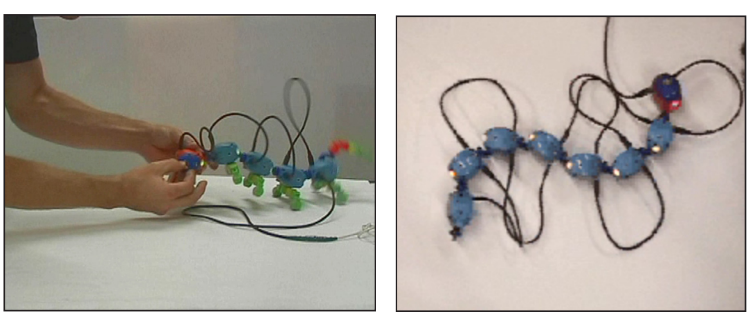
ments of a system's parts can lead to its global result. Because programming is a mathematical process, programming in Starlogo may have helped these children develop better symbolic math skills and an intuition for advanced ideas like the wave equation.

Topobo is a decentralized system, and could also be used to visualize the effects of programs created in a language like Starlogo. The Queens and Backpacks make some of these ideas tangible, and a GUI could add an additional level of control to the system. Such an interface can be imagined as an iconic programming language with distributed control like Starlogo and an iconic representation like Logoblocks [Log04]. A graphical output would mimic the physical state of the Actives, and Queens and Backpacks would be interpreted and represented as objects and functions. For example, if a child built a string of parts with a Queen and Bigger/Smaller Backpack, the screen based representation would mirror the state of the physical system. When a child recorded a motion with the Queen, the software would automatically construct mathematical equations and a control structure that represented the nature of the Queen and Backpack, including a timeline representation of the Queen's position in time.

Clicking on icons of successive Actives would represent progressively augmented versions of the Queen's motion. Children could experiment with editing the motion graphically and observe the output, or use this motion as a basis to graphically create new motions. Mathematical functions like a sine wave could be substituted for the physically input motion so children could compare the different behaviors. Alternatively, children could change the behavior of the Backpack — essentially reprogramming it — by editing the symbolic math that the software generated from the original physical model. Along these lines, Maeda suggested "Scratchpacks" with which children can create symbolic programs using the iconic Scratch language, which can be either injected or applied to a Topobo creation. This could give children insight into the design and function of the Backpacks as well as let them experiment with their own algorithms, learning how a global creation can be controlled or coordinated with local operations. Children could then observe the real effects of friction, gravity and material compliance on their physical systems, whether they are wave-induced strings of Actives or walking robots.



In Starlogo, wave behavior is programmed by creating the wave equation with a variant of LOGO.



With Topobo, children can create wave-like motions by combining Queens (same motion everywhere) with the Time Delay Backpack.

Balancing the physical and digital in Digital Manipulatives

One must ask when it is appropriate to add new features to a toy or tool. How much does the new feature add? Is something lost in the process of this change? Computational additions to Topobo in the form of GUI software should complement the system and retain the best qualities of the physical interface. In the same way that complexity like Queens and Backpacks can engage older students with Topobo, a GUI application could help older kids who are already learning programming to do more and learn more with Topobo. However, the activity would have to remain consistent with the system's foundations in using motion to broaden the extent of physical modeling.

I am not proposing that Topobo become a generalized programming tool, because I think it will be more successful if its applications specifically take advantage of the system's physical affordances. The robot design examples are intended to show how the control structures might be further developed without superseding the underlying nature of the system. These approaches could help kids transfer knowledge by building on the specific qualities that make Topobo unique (dynamic motion) and support the transition to abstraction in the process. Topobo is somewhat specific but remains open-ended. In this balance, the system suggests activities without prescribing discoveries.

Jumping to Symbolic Systems

Considering Experience

When should children jump to the more abstract symbolic systems? While there is no single answer to this question, I find two analogies to be helpful in considering how and when to make such a transition. First, Bruner cautions that if people begin problem solving with symbolic systems, they may lack enough concrete experience to debug their own models. In the case of modelling physical systems, I have already argued that beginning with tangible (not symbolic) systems can provide a foundation for more abstract modelling. So the answer here might be "not right away."

Considering Design Tools and Cycles

Second, if we think of the tools of a product designer, we find she uses a range of different tools at different stages of her design cycle. A product design may begin with a pencil sketch, transition to a clay model and then go through many iterations before being transferred to a computer. The computer is used for the last stages of design: quantitative analysis and design for manufacturing. Simpler materials like the pencil will be used throughout the design process, and the clay may be used while 3-D modeling the final form. Successive layers build upon the previous examples, and each material supports and informs the design process in a different way.

Anecdotal evidence of this idea was evident in a number of cases: Some children demonstrated that they were ready for a higher ceiling. Older children in Virnes' study, who were adept with symbolic programming languages like LEGO Mindstorms found Topobo to be interesting and fun, but sought more precise control and the ability to create more sophisticated behaviors (although these students did not have Backpacks, Remix or Robo). University level biology and robotics researcher Robert Full thought Topobo would be an excellent learning tool for his students to learn about the ways physical structure can play into a robot's control system (something he calls "preflexes") but saw the backpacks as an opportunity to apply AI techniques to Topobo's gesturally programmed motions. It seems both students and teachers see tangibles as a valuable way to get certain kinds of complex ideas roughed out, but when a student desires precision or more sophisticated computational behaviors, a more precise programming model applies.

Looking at Topobo as a part of a system of design tools, we can imagine the following ecology of tools and uses: Topobo may directly function as a educational tool that engages the design process; concepts are learned through the process of design. If we imagine Topobo, for a moment, as a design tool for creating a walking robot, the system may be used first to explore possible means for locomotion. When a model is deemed successful, the student may refine that motion using Queens, Backpacks, and Remix or experiment with fine adjustments to the creation's geometry to improve its gait. Compositions might be tested with Robo, and motions further refined. After this iterative process, the child might connect the creation to GUI software that allows the child to more finely tune the motions. Experimenting with this software could allow the child to explore symbolic mathematical models to better control the motion of the robot. The final output may be the Topobo creation itself, or the creation may serve as an example for another robot. The whole process can be a fun and rewarding learning experience through which the child designs and builds a creation and simultaneously develops ideas that inform her understanding of natural systems in the world around her.

Considering Age

The question of when to transition from tangible tools to more abstract ones must also consider the student's age. Both developmental theory [Pia76] and practical experience with children using LEGO/LOGO systems suggest that children who are able to manipulate abstract ideas (e.g. age 10+), will be able to benefit from the powerful qualities of symbolic systems. Symbolic systems may not be accessible or appropriate for younger children, and the jump might be most successful beyond a certain age. The question of how to jump will likely depend on the child. I have argued that Queens, Backpacks, Robo and Remix can provide a theoretical foundation for children to play with advanced ideas, and this could be elaborated by developing a Topobo-specific programming language that builds on these ideas in more powerful ways.

Like Papert's gears [Pap80], Topobo could become for some people both a modeling tool and a metaphor. The tool both is a medium through which to explore certain ideas and helps a child learn lessons that are used throughout life. If the lessons are general enough, a person may return to the tool throughout life to continue to play and experiment with a body of ideas. The tangible interface can spark memories and may become a resource that suggests different solutions to a person at different times in their life.

10 The Future of Play
Pursuing Kinetic Materials

This thesis has approached Topobo both from a design perspective, as a platform developed for generalized actuated modeling, and as a specific contribution to tangibles for learning that help children learn through interaction with physical objects. This section will propose design guidelines for tangibles, based on Bruner's framework. I will also consider issues that arise in pursuing a new class of media, which I call Kinetic Materials.

Interaction Design Guidelines — Extending Bruner's Framework

I have argued that Bruner's framework helps us understand:

(1) *Why* tangibles make learning certain (enactive) ideas more intuitive: some ideas – especially spatial, and body-scale kinetic ones – are best explored in the tangible domain, through enactive representations.

(2) *How* enactive learning can provide a foundation and stepping stone to develop more abstract and complex ideas: ideas are first explored through concrete, enactive representations, later visualized through iconic representations which are later stereotyped and understood in terms of more flexible, abstract representations.

(3) *When* (for what ages) different interface paradigms are likely to be most successful: younger children will be most successful working with enactive representations. As children mature, iconic ones are meaningful, and around age ten, children can intuitively grasp symbolic (language-based) representations. And all people may benefit from progressing through their learning in this order.

Through Topobo I have already provided examples of how a tangible system can scaffold learners from enactive through iconic and symbolic representations. Tangibles need not be criticized for being "too limited" since their limitations can be overcome.

Scaffolding: with Topobo, I transitioned between simple gestural programs to more abstract understanding of them using specialized interfaces and controllers. Other systems employ "stepping stones" to scaffold users to progress from simple-but-intuitive to precise-but-complex models. Just as Sandscape [Pip02] allows users to immediately transition from their physical model to a wireframe mesh 3-D computer model and Pico [Pat06] provides jigs as a means to define and compute parameters in a computational optimization model, we might invent a variety of systems where tangible tools correllate closely to a computational or symbolic process.

For example, an animation system could begin with drawing and gestural input, and provide simple tools to gesturally, or graphically, refine and compose those motions. For instance, how could one provide the intuitive interface of IO Brush, and the power of Flash animation?

Correlation: One guideline for interfaces that transition from intuitive-but-simple to complex-but-abstract is to provide a tight *correlation* (in concept or technical execution) between ideas that can be played with tangibly and ones that can be manipulated symbolically.

Staging Interactions: Another is to *stage interactions*: the UI will support users to begin their learning enactively, and be able to evaluate and later understand it iconically before needing to learn a symbolic system to describe and refine the behavior.

Intuitive Interfaces: Tangibles are often argued to be more intuitive that graphical interfaces. This should be true especially for children and novice users who do not have a conceptual foundation to manipulate some ideas using more abstract techniques, e.g. math or programming. According to the framework, we would also expect enactive representations to provide more intuitive beginnings for non-expert adults.

Kinetic Interfaces: where kinetic information can be modelled tangibly, such an interface can provide a more intuitive way to begin a model. Certain kinds of kinetics would need to be transformed to be "felt" and "touched," e.g. temperature variations or mechanical resonances in physical structures might be modelled in a tangible interface, and felt as slow-motion vibrations. 2-d or 3-d animations would be most easily composed with a tangible interface, and later refined with other techniques.

Spatial Interfaces: spatial information is most easily understood in 3-d space, in the physical world. For generations, sculpture, architecture, 3d design and planning has been understood first in 3-d materials before specifying it with more precise languages. For spatial topics in all disciplines, from drug design (requiring an understanding of protein receptor sites, dynamic protein folding, etc) to mechanical engineering (e.g. quickly modelling and feeling stresses, strains and resonances in a physical model of a bridge or building) rough and quick tangible modelling languages can provide an intuitive foundation for designers to begin their work.

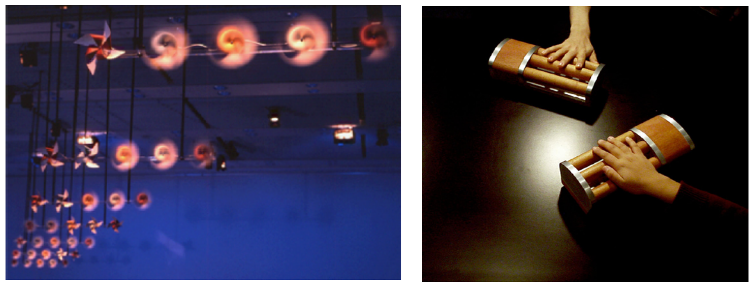
Emerging technologies: Actuated modeling

Beyond interaction design guidelines, Topobo may provide inspiration for future technologies. I propose an emerging category of actuated materials with which people can design a variety of tangible interfaces. Viewing Topobo as a modeling *material*, I will consider some potential applications that stem from past research in tangible interfaces.

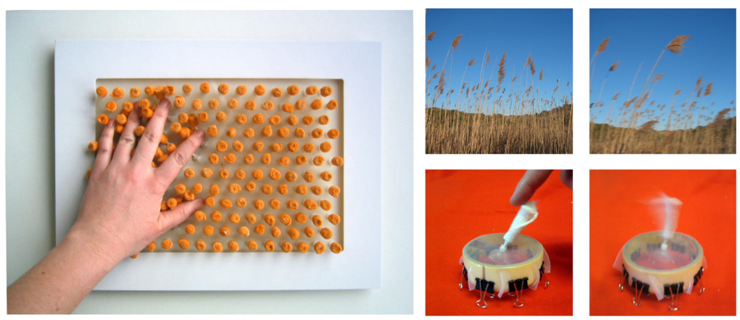
Movement as display

Movement is a natural means through which the physical world "displays" information. From one perspective, the development of tangible interfaces is similar to the development of motion graphics. The visual representation of information through 2-D images has progressed from static representation (paintings) to dynamic representation (motion pictures) to interactive dynamic representation (motion graphics). This might be described as a trend for the image to more authentically represent life. Where the image once captured a moment, film captures a temporal narrative and motion graphics give the narrative (or character, or object) a behavior, social context, or response to its environment.

Physical objects have a similar history. Where sculpture once captured a static moment in a physical form's existence (e.g. a Greek figurative statue), mechanized automata of the 18th - 20th c. gave those forms life. The trend in tangible interfaces to use objects' movement to represent both abstract information (e.g. pinwheels [Wis98]) and human intention (InTouch [Bra98]; curlybot [Fre00]; Super Cilia Skin [Raf03]) explores the potential for the object to reflect life and become an interactive part of a culture's social fabric.



Pinwheels and InTouch explored ways to use mechanical motion as a display.



Super Cilia Skin is inspired by wind swept grass and explores how motion in an array of hair like actuators can be used as a textural communication device.

Movement as Interface

All of these projects use mechanical movement as an interface. InTouch, a system of two sets of remotely coupled physical rollers on stationary bases, creates the illusion that two people, separated by a distance, are interacting with the same physical object [Bra98]. Pinwheels use the spinning of an array of these familiar objects to represent real-time internet data such as stock market activity or ocean waves [Wis98]. Super Cilia Skin explores how dynamic texture can be used both as a gestural input medium or as a kinetic display [Raf03].

The prevalence of mechanical movement as an interface leads one to raise the question whether mechanical movement is a fundamental quality of tangible interfaces. In contrasting tangible interfaces with the graphical interface and pixel [Ish97], Ishii implied that we do not yet know the fundamentals of display for tangible interfaces. For images it is color and light, modulated by an array of computer controlled pixels. Certainly color and light are a fundamental quality of TUIs. Is mechanical movement — and its many manifestations such as temperature (molecular movement) — another fundamental quality of TUIs?

An argument for new technologies

If mechanical movement is a core ingredient for TUIs, can a single material suit many needs? I believe one could imagine building InTouch, Pinwheels, Curlybot or Super Cilia Skin with a kinetic medium, similar to Topobo. One might use actuators that are smaller or that have continuous rotation, but the main idea is that a computer-controlled, scalable, actuated modeling system could be a display and interface for an entire class of tangible interfaces. If the material existed, would it enable further developments in TUIs? I believe it would support a growing class of TUIs that use shape or motion of physical objects as an interface. With the miniaturization of the actuators and the development of a GUI-based API, a tangible interface designer could use an actuated modeling system like Topobo to model a variety of compelling interfaces without struggling with months of customized hardware design and manufacture.

Communiclay

In order to prototype this concept, we have developed Communiclay, motorized robotic nodes that can communicate gestural manipulations over the internet. Communiclay builds on the Topobo platform, and allows users to attach their Topobo creations to a PC, open a Java application, and share their gestures with others in a multicast group. Several people may be sharing gestures at once, or two people can have an exclusive "conversation."

Communiclay is like a kinetic walkie-talkie: a user pushes a button on a Topobo node, and moves their creation around. All other creations on the network will mimic this motion until the sender ends the message, or someone else interrupts with a new gesture. While the system does not support full duplex communication, latency is very low, typically less than 40 ms. for users on the same subnet. Since two users' creations may be built differently (i.e. different number of nodes, different network topology), the software adapts to try to send data consistently to non-symmetrical creations.

Communiclay provides synchronous sharing of gestural manipulations of Topobo over the internet.

Communiclay allows Topobo hardware to adopt the functionality of InTouch, behaving as a remote haptic communication medium. People can sculpt an artistic object that is aesthetic, symbolic or ergonomic for them and use it as a basis to communicate touch with a friend. As Brave noted [Bra98], an non-figurative (abstract, generic) interface allows the users to ascribe their own meaning to the movement they are generating.

When people ascribe specific meaning to a kinetic creation, e.g. creating a pair of flowers, those flowers can become metaphors and signifiers for other meanings, e.g. laying down means "I'm tired." In this way, communiclay can become an ambient display.

Communiclay can also support remote learning with Topobo. For instance, if a novice is struggling to learn how to program a creation they can ask a remote expert to teach them by actually programming their creation over the internet, in real time. The novice can then kinesthetically feel the program and learn it through touch and enactive representations, providing a foundation to successfully recreate the program themselves.

Since the meaning of gesture alone can be ambiguous, the Communiclay software integrates text messaging and voice over IP to allow text or voice to complement the movement.

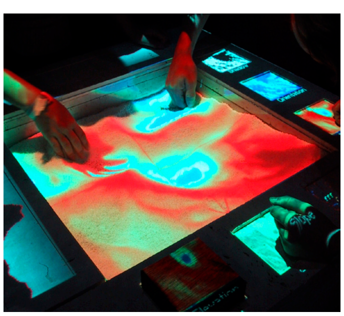
Communiclay allows people to integrate touch into their computer supported remote communications, and also provides an infrastructure to connect Topobo to a PC, supporting more sophisticated GUI-based control of the system.

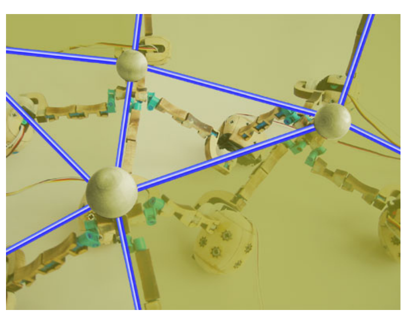
An argument for a new medium

While Communiclay is a functional and successful proof of concept, it suffers many limitations as a kinetic medium. What criteria should define the creation of a powerful kinetic medium? It is important to remember that designers are not engineers. In an overly general sense, engineers ask "how" and designers ask "why." Because designers are focused on the application of materials rather than their invention, designers create more and better designs when their materials are easy to use. An actuated modeling system could thus inform and facilitate the development of better TUIs in the future by making kinetic materials accessible to designers.

Of course, to achieve this goal a number of technical hurdles must be overcome. A main consideration is the size and behavior of the actuators. Motors are getting smaller, and the development of Topobo Actives that are no larger than a person's fingers would greatly improve the system's applicability to more generalized modeling. Such elements would be an improved modeling system and the actuators could get as small as LEGO bricks and remain useful. However, as actuators shrink another order of magnitude to the size of a pea, most people could no longer easily assemble the individual components. The material would need to behave more like a fabric that could be cut and sewn together. Shrinking still further, one imagines the long-sought "digital clay," that eludes the best materials scientists and engineers and lives in the realm of the movie industry's special effects departments (Terminator 2, for example). Such a material seems to be far on the horizon.

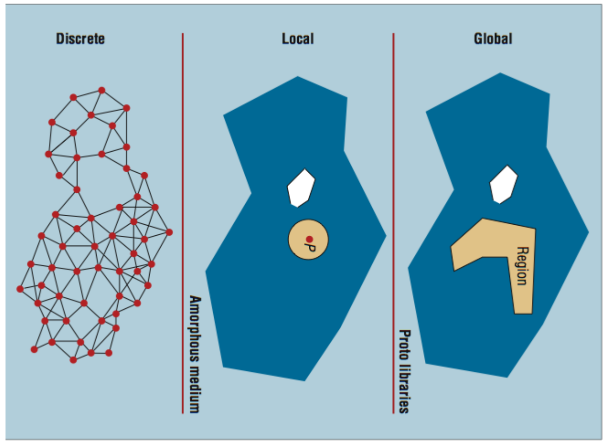
Nonetheless, much can be done with the large, simple and crude actuators that we use in Topobo. Topobo is not "digital clay" nor will it ever be, but it can become a platform to explore some ideas made accessible by kinetic modeling and provide some basis to fuel future research in these directions. As it becomes miniaturized, Topobo will become less





As scale shrinks, an actuated modeling system could be used for actuated surface mesh modeling. Here we compare Illuminating Clay to a Topobo mesh.

Proto decomposes self-managing systems into Discrete, Local and Global abstraction layers. This allows high level specification of distributed behaviors.



like a modular robotics system and more like a material. When it is coupled with an API that simplifies the interface to the physical system, many new and innovative tangible interfaces may be built with it.

Protobo: Programming a Distributed Kinetic Material

Questions of programming a kinetic material are the research of computer scientists who focus on amorphous computing, sensor networks and distributed operating systems. While physical programming would apply excellently to a kinetic material, at many scales (both temporal and spatial) we would require a different model. For example, how could we think about programming motion at the scale and speed of a protein? If the behavior cannot be transmitted through touch, how are we to apply it to the material?

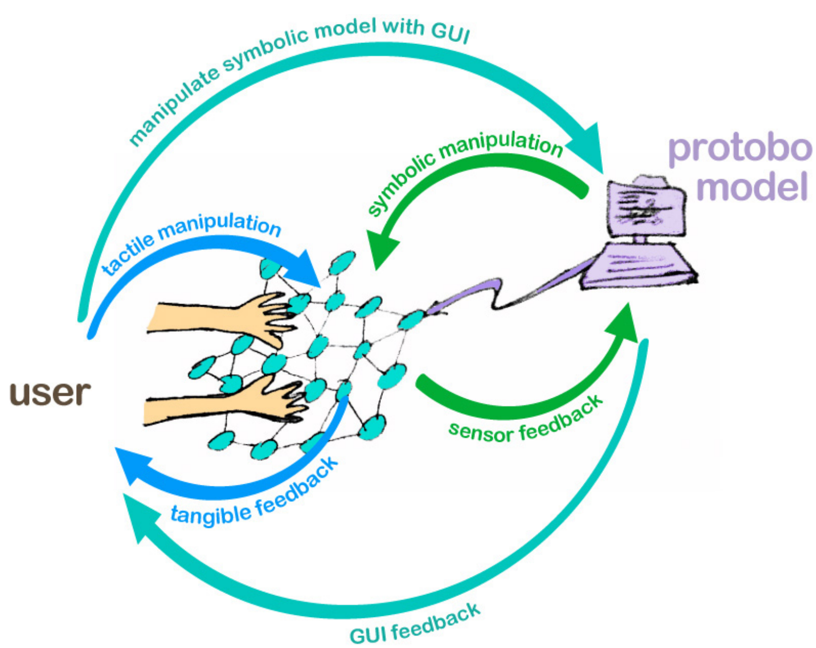
To program the behavior of a kinetic material we would like a programming language that allows us to describe high level global behaviors, which in turn is executed on a distributed programming language embedded in the material itself. The material should be able to "inherit" behaviors from its neighbors, so that if I add material to my object, the addition becomes part of the whole (both physically and computationally).

This vision motivates the *Protobo* project underway with Jonathan Bachrach, in which a high level lisp-syntax language called Proto has been developed to define the behavior of distributed robotics systems [Bac07]. Proto allows users to write simple composable scripts, simulate those scripts on a PC, compile them, and download them to be executed on a kinetic material.

Proto decouples self-management problems by decomposing self-managing systems into three abstraction layers: global, local, and discrete. Interactions between individual devices in the discrete layer emulate and amorphous medium. The local layer describes the behavior of points (e.g. Topobo nodes) in the medium, from which we build library code to allow description of the behavior of regions of the medium at the global layer [Bea06]

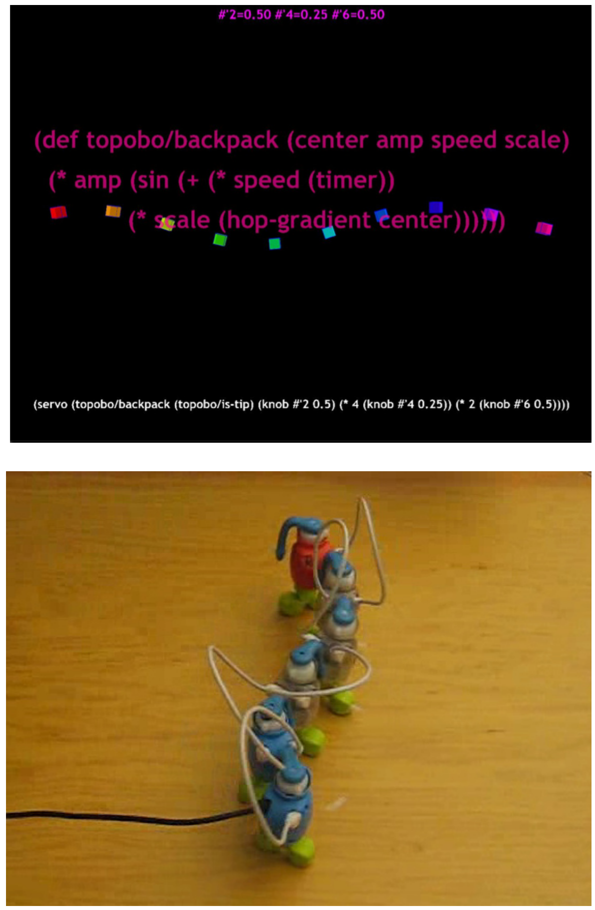
With Protobo we can demonstrate how complex distributed behaviors like those modelled in StarLogo can be understood in terms of interconnected kinetic objects. For example, with Proto we can recreate wave functions of physical phenomena like slime mold growth on Topobo. We can allow the user to create all of the Topobo, Queens, Backpacks, Remix and Robo functionality, and then play with composing those functions locally or globally, or distributing them across the network. With Protobo we can play with the same ideas children already explore when they have their hands on Topobo, but with the full flexibility of a functional programming language to define and refine the creation's behavior. Protobo can allow Topobo users to start at with a rough, gestural prototype and dig deep into behaviors that can be understood in terms of a composable programming language.

Protobo also allows us to use Topobo to prototype a *kinetic material* that can inherit parent behaviors as the system is reconfigured by



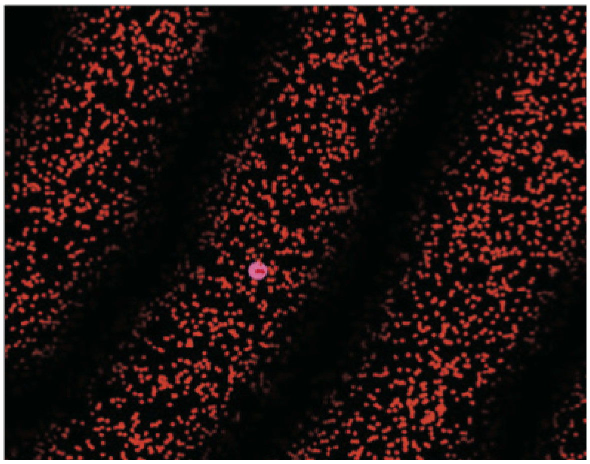
Cooperative interaction with a kinetic material: the kinetic material's behavior is affected by a combination of the user's tactile input and the Protobo symbolic model. The user will manipulate both the Proto program, via a GUI, and the kinetic material, via tactile gestures.

the user, where nodes are added or removed. We can simulate massively scaled systems, and foreshadow issues that may arrive when MEMS technology makes microactuators accessible to manipulate with one's hands. Protobo looks to a future of digital clay, where the physical medium can be manipulated and understood with metaphors people assume of today's inert physical media.



Left, a Topobo backpack behavior is scripted in Protobo and simulated on screen. Right, the same code is executed on a proto VM running on Topobo, generating a linear sine wave.

How will the next generation of materials engineers or roboticists transfer the learning they may have done in the tangible domain to engineering tasks at vastly different scales? Perhaps they will begin with hands-on modelling and discovery with Topobo, explore concepts of behavior and control with Queens, Backpacks, Remix and Robo, and then dive into Protobo to learn ways to classify and reformulate their theories in terms of a symbolic language. In an ideal scenario, Protobo would allow users to apply the language's



A plane-wave oscillator running on 10,000 simulated devices. The placement of source (yellow) and destination (magenta) markers in the devices' sensor field determines the wave's period and direction. In a kinetic mesh, the source and destination may be specified tangibly, e.g. by pressing buttons on nodes.

functionality to either gestural or synthesized movements, and fluidly transition back and forth between gestural programming and symbolic modelling. Discoveries made in one medium could be seamlessly integrated into the other.

While Protobo can be viewed as another step in the spiral of a student's progression from simple to complex, it marks a drastic jump, a phase transition. Since it introduces an entirely new paradigm to a distributed robotics system, students will need to re-imagine their theories in the terms of the proto language. At a minimum it would allow the principles learned via tangible programming to be leveraged to create behaviors for today's distributed robotics platforms (i.e. Topobo).

When atoms can dance

Our vision is that play with Topobo and Protobo will give children tools to approach the design and control for not just robotics, but our future's "kinetic materials." New tools – and especially new thinking – will be necessary to bring physical atoms to life, and Protobo aims to imbue on children (and adults) a sensibility and intuition for how such systems may be conceived, understood and used.

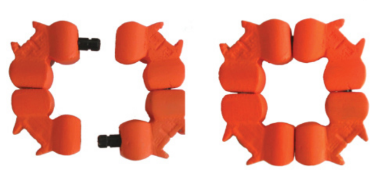
11 Conclusion
Coevolution of children and toys

Topobo is a tangible approach to learning advanced concepts related to biology, engineering and computers through playful invention and exploration with a dynamic modelling language. Instead of providing children with a keyboard, screen, and mouse, Topobo presents computation as familiar children's toys and encourages children to experiment and explore kinetic behavior with the push of a button and a flick of the wrist. Children began to sculpt with motion, and explore concepts related to biology, engineering, and computation as they developed and grew with the system. Topobo makes complex concepts accessible to children as young as 4 and Queens, Backpacks, Robo and Remix supported the inquiry of all users, from young children to expert, adult robotics engineers.

Topobo shows that toys can be designed to *coevolve* with children, to reveal salient ideas and relationships to children throughout their social-emotional and cognitive development. The toys themselves don't necessarily evolve or change, but rather the ways children approach them, and the kinds of things they do with them will change with a child's growth.

How can a single toy or system coevolve, allowing children to progress from simple-but-intuitive to flexible-but-abstract ideas? My strategy has been *Multi-Layered Abstraction,* through which children progress from hands-on (enactive) experiences to high-level abstract concepts. Following Bruner's theory [Bru04], children explore new ideas first in the enactive, tangible domain. They observe the effects of their work by observing their models, which are iconic representations. These models provide a foundation for children to reflect on their ideas and to develop abstract, symbolic representations of ideas that can be manipulated in more powerful and generalized ways. Topobo shows that it is possible to progress from concrete to abstract without giving up the tools and experiences you are already using to express and learn ideas.

A major problem in introducing computing (and embedded computing in particular) to kids stems from the disconnect between the physical and computational realms, or the "layers of abstraction" that separate them. This thesis presents a system that has eliminated some of the distance between computation and the "real world" while providing possibilities for truly sophisticated activities, whether they are intellectual, playful or physical.



Appendix A
Engineering Topobo

I overview the development of Topobo from early prototypes to mass produced toys, and look at the technical limitations in designing the system.

Structural Parts

"Structural parts" include the Passives, Active housings, and Backpack housings. (Remix employed a simple paper and electronics prototype and is not reviewed here.) They have been developed through dozens of iterative design stages that span multiple fabrication techniques. In general, the earliest techniques were fast to build but not extremely accurate. Later techniques required exponentially greater amounts of time and energy to implement, but the result is accurate, beautiful and manufacturable (repeatable) parts.

Flat Studies
The passive geometry is based on flat shapes, and I laser cut our original prototypes from 3/8" bass wood and glued LEGO connectors in to their ends. Bass wood is strong, light, affordable and aesthetically pleasing. However, it is soft and the notches compressed and wore out due to repeated connections. It also lacked a "finished" look that we sought for user studies.

3-D Studies
We developed a more three dimensional design for the passives in order to encourage users to think about the components as volumes rather than as flat puzzle pieces. The Active housing was designed to

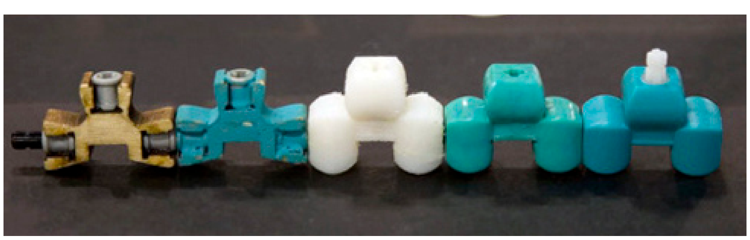
accommodate the servo, PCB, and LEGO connectors, and to be aesthetically consistent with the Passives. These parts were designed with sculpture materials such as clay, and various 3-D modeling environments. Final parts were manufactured with an FDM 3-D printer. The FDM produces ABS parts with fairly good dimensional accuracy and about 85% of the strength of molded ABS plastic. Although FDM prints are a close representation of injection molded parts, we were not able to hold snap fits with LEGO connectors on our 3-D prints, so we glued LEGO connectors into the 3-D prints. This gave us the "look and feel" of injection molded parts. The more finished quality of the parts allowed children in our studies to focus on the interaction design rather than handling fragile prototypes with many long wires. The children's feedback was also helpful in refining the design of the parts. For example, the students' difficulty in distinguishing the rotating connector on the Actives led to a redesign of the Active housing.

Hand Molded Parts

Hand molded parts were molded plastic, based on 3-D printed models. The passives were injection molded in ABS using a bench top press and epoxy/aluminum molds fabricated from 3-D wax prints. Passives were made in two pieces (split laterally) so that the assembled part is hollow. While a lateral weld seam causes snap-fit tolerances to be affected by assembly, through careful quality control the finished parts are dimensionally accurate, durable and have solid color (e.g. they are not susceptible to scratching). The Active housings were molded in 3 pieces in urethane resin with silicone molds. Since urethane is not durable enough for repeated insertions of LEGO connectors, LEGO plugs are in-molded in the urethane castings.

Final production parts

In all of our user studies, children would break our prototypes, so in



Five generations of passive parts are connected together.

order to distribute Topobo to larger audiences and over longer time periods, it was necessary to mass produce the basic system (Actives, Passives and Queens) using modern techniques. A two year collaboration with a Chinese toy manufacturer lead to injection molded ABS parts, with custom electronics, metal gear servo motor and integral acetal clutches in all of the passive components. We maintained the basic design of the prototypes, but rewrote all firmware for a more powerful processor, redesigned all of the electronic infrastructure and redesigned the parts to account for shrinkage, molding, and ease of manufacturing and assembly.

Mechanical and Electromechanical Engineering of Actives

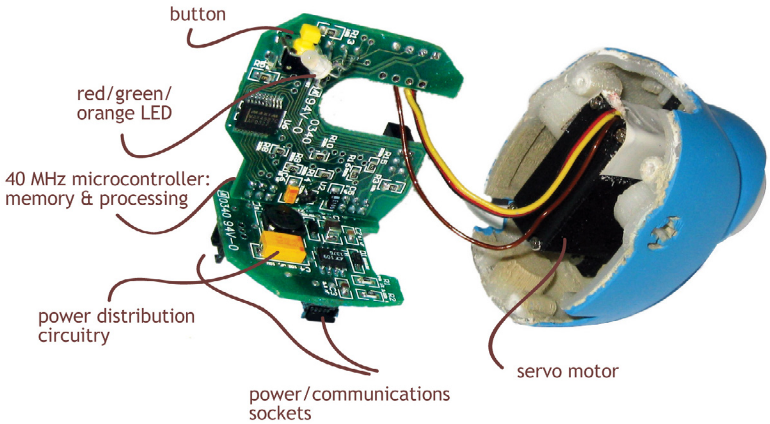
The engineering of the Actives is based loosely on modular robotics technology developed by Yim et al. at PARC [Yim00]. While I did not directly adopt any of Yim's designs, I considered his approaches during the design of the Actives and eventually adopted several similar approaches for scaling power. The PARC robots employ a hinge joint, but my geometrical studies focused on rotary motion. Therefore, one open question was whether to provide separate Actives for rotary and hinge type motions. Both are rotation, but users think about them differently when they build.

Joint Design

I chose a pivot joint for simplicity. It was very easy to connect my passives directly to the output shaft of the servos, and I sought to keep my mechanical design overhead to a minimum. To accommodate hinging joints, I designed a special passive called an "elbow" that allows an Active that is normally used as a pivot to be used as a hinge joint. A better system design might include a separate "hinge Active" for clarity, and such a part might turn at two collinear locations instead of one.

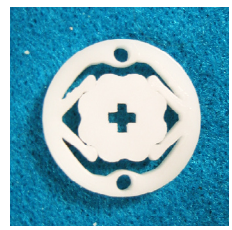
Motors

During record mode, the user back-drives the motor by turning the output shaft of the gear box. This is bad for the gears because they can break. The teeth on gears at late stages of a gearbox are often small and fragile, meant to be driven at a limited torque. When a gearbox is back-driven, small amounts of inertia in the motor core,



Custom electronics handles memory, processing, communications and power distribution.

plus friction in the gears themselves, are amplified by all stages of the gear train. Since Topobo requires motors that are both fast and strong (i.e. powerful) they require a sizable gear reduction, so back-driving the motor can break the gears. Ultimately, we designed our own servo for production that provides high output torque and has low input stiction.



An indexing clutch protects the servo from excessive torque. It feels different than normal recording if it slips. It also allows a position to be easily recovered.

Clutches

Clutches are integrated into all passives and the output shaft of the Active's servo motor. A clutch uses an arrangement of 4 spring arms and an indented ring so that it "clicks" through 45˚ increments, and is precision injection molded in acetal (delrin) resin.

I designed an indexing clutch over a slip/friction clutch for several reasons. I thought it would be easier to consistently manufacture an indexing clutch, I wanted the "feeling" of the clutch to be different than normal back driving of the servo, and the finite position of an indexing clutch could be easily recovered if the clutch slipped by accident. Some people use the clutch as a feature, in that once a passive is connected to the Active's clutch it does not need to be removed to be reoriented, it only needs to be "clicked" into the right place.

Compliance

Topobo benefits in several ways from slightly flexible connections. Kids can easily connect and reconfigure parts that do not fit perfectly, inaccuracies in motor calibration or gearbox backlash are inconsequential, and creations that are accidentally dropped or stepped on fall apart instead of shattering. The most rigid, and therefore most fragile, element of the system is the LEGO connectors. These small molded plastic pins break before any other part and have to be drilled out to be removed. This can be viewed as a flaw (they are poorly designed and should be stronger) or it can be viewed as a benefit, where the cheapest part in the system will fail before a more expensive one does.

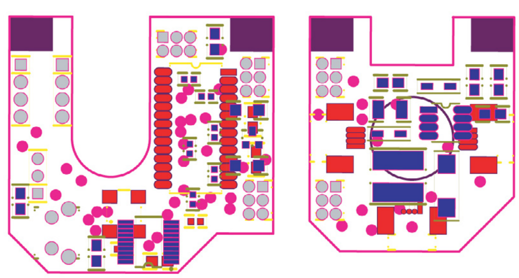
Electrical Engineering

The Actives' on-board custom electronics handles power distribution, memory, processing, and multichannel serial communications.

Power Distribution
Upon suggestion from former professor Paul Horowitz, early prototypes used an 18V power bus that is locally stepped down to 6V with a non-isolating buck converter and then is dropped to 5V with a linear regulator that powers the digital electronics. This minimizes the effects of power losses in the system, limits noise transfer between Actives and reduces current draws through our miniature connectors.

For production, we sought to reduce costs by running all electronics at 5V directly from the buck converter, and isolate the motor with a diode. This is a less flexible but effective design.

Processing
A 10 MIPS RISC microcontroller handles local memory, processing and network communications. At manufacture, a one-time calibration sequence measures the range of motion of the servo and correlates input and output position data. During record, the microcontroller reads the servo's internal potentiometer at 36 Hz using a 10 bit



Custom circuit boards are designed to fit around a variety of servos.

ADC and writes scaled 8 bit values to local memory. This gives us 34 seconds of record data at 3/4° output resolution, which is accurate compared to the backlash in the servo's 4 stage gearbox. The sensor is filtered by an RC low pass filter (f3db ~ 10 Hz) to remove high frequency noise. A custom peer-to-peer serial networking protocol transfers data between Actives at roughly 9600 BPS. Mini USB-b connectors and series resistors protect digital electronics during hot-swapping power/communications cables between Actives. Our early decision not to use batteries keeps Actives lighter and avoids the need to regularly maintain power sources.

Scalability

An engineering goal was to create a scalable system that could accommodate up to 100 Actives at once. So far, we have successfully tested the system with 50 Actives. The high voltage power bus facilitates scalability by limiting current requirements and noise transfer. In general, the peer-to-peer networking protocol is scalable both in software and in hardware. Compared to a multi drop bus such as RS485, the peer-to-peer arrangement is more fault tolerant to floating grounds that can occur at the ends of long chains of Actives because immediate neighbors will always have close relative power and ground levels. So far, we have not exceeded Topobo's limits of scalability, but as the number of Actives in a creation increases, we suspect the main bottleneck will be series resistance in long chains of Actives. Series resistance may either affect data transmissions (which is sensitive to floating grounds), or motor driving ability (which requires high startup currents).

Nonetheless, large structures do not always work as quickly and reliably as small ones. Topobo is susceptible to floating ground loops that can occur when people create large electrical rings of Actives. Large structures tend to work faster and more reliably if they are powered from multiple distributed points. If systems need to increase scalability, one approach is to use a higher voltage (24V - 48V) power bus and avoid network loops.

Software: Distributed Computation and Control

The autonomous functions of an Active include motor calibration, local recording and local playback. The remaining computation is devoted to a network communications protocol that is designed to be fault-tolerant and flexible. We expected children to arbitrarily create various network loops, push buttons in parallel, start recording with one button and stop with another, and do other "non standard" things with Topobo. Therefore, the system is designed to cause Actives to stay in synchronized states amidst any possible network topology, to easily incorporate new nodes that might be added to the network, and to easily forget nodes that are removed from the network. A number of people helped develop the firmware including several undergraduate researchers and colleague Josh Lifton. The system's stable and extensible firmware is largely Lifton's design, and I am grateful that he lent his expertise and experience with embedded networks to the project.

The major challenge in the firmware development was coordinating two time sensitive tasks, motor control and serial communications. While our servo requires a low duty cycle signal (about 36 Hz), it must be extremely consistent and is not fault tolerant, so motor control has priority over network communications.

Motor Control

The servo is driven by sending a 36 Hz TTL signal whose peak is 1-2 ms. long. Varying pulse widths correspond to absolute output positions measured from a potentiometer that is connected to the output shaft of the servo. Our microcontroller creates servo pulses using a two timers that change the duty cycle of the pulse based on 8 bit position values. No two servos are the same, so a valid range of pulse widths is established for each Active during a calibration sequence that is performed at time of manufacture.

Motor and Sensor Calibration

The calibration algorithm correlates input potentiometer readings from the servo to corresponding output pulse signals. The mechanical range of the servo is smaller than the electrical range of the pot, so we do not use the full range of the ADC. The calibration scheme first determines the absolute minimum and maximum potentiometer readings for the servo by overdriving the servo to the left and right mechanical stops while reading the ADC. A series of measured pulses then gradually drives the servo to the left and right stops while the ADC is concurrently read. When the ADC value matches the previously recorded minimum or maximum value, a minimum or maximum pulse width is recorded for the servo. These maximum and minimum pulse and ADC values are stored in EEPROM and all subsequent pulse widths are created along a linear scale between the minimum and maximum pulses. Similarly, all subsequent 10 bit ADC reads are linearly scaled to an 8 bit value between 0-254 before being stored in memory.

The calibration scheme is convenient for a number of reasons. It allows us to use the full range of the mechanical motion of each Active, get full resolution out of 8 bit storage registers in a data array used for position recording, and standardizes all positions readings across Actives. For instance, it is due to this standardization that the Queen is able to easily communicate a "copy" command despite significant inconsistencies among Actives' hardware.

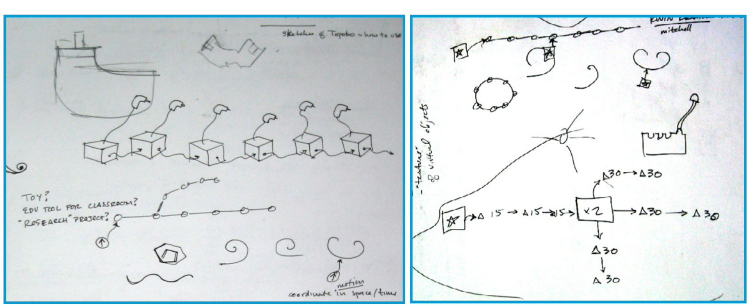
Record and Playback

During normal local recording, an Active will read its ADC at about 36 Hz and write values to an external EEPROM data array. Data is read and denormalized during playback. By writing to nonvolatile memory, programs can be recalled if an Active is unplugged, using double clicks, Remix or Robo.

Communications

Peer-to-peer communications are handled exclusively in software, giving us 4 channels of serial communications with data rates at around 9600 bits per second. The networking protocol, based on I2C, uses two wires for communication, generally used as "clock" and "data" that are by default pulled to Vcc with internal pull-up resistors. The protocol includes handshaking, parity checking and a hop count (time-to-live) to avoid re-sending messages in network loops.

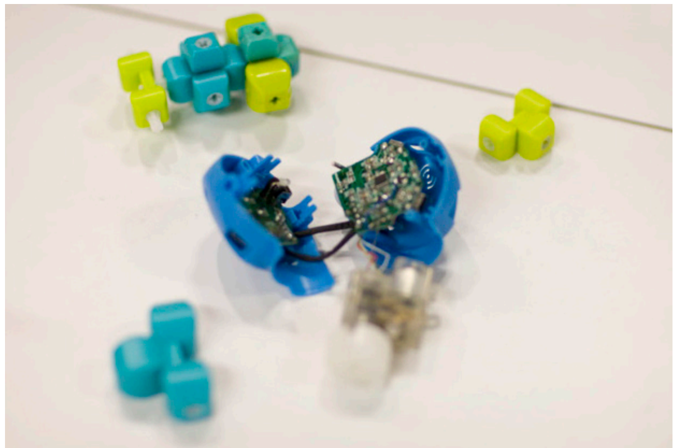
Due to complexity in handling message re-sending, collisions



Early sketches explored the benefits of a peer to peer network architecture.

Final production parts were reengineered to improve durability and reliability.

Topobo is the first mass-produced modular robotics system.



and other nonlinear behavior, Mike Fleder developed a system of message stacks and queues that handles incoming and outgoing messages for multiple channels of data. This system avoids network collisions using a "random backoff" in message timing (akin to ethernet) and greatly increases the system reliability. The trade-offs are memory and network speed.

Distributed communications

Topobo is a distributed system comprised of individual elements each with their own internal parameters (e.g. speed) that define their behavior. Topobo leverages distributed processing techniques from sensor networks and related research to accomplish many tasks, including temporal and state synchronization, backpack functions, and establishing flexible relationships between Remix, Robo, Backpacks and a Topobo creation. When Backpacks, Remix or Robo are added to a network, the "host" Active coordinates with the device and signals the Topobo network to change appropriate internal parameters (e.g. speed) that reside on individual Actives.

Limitations of the current design

Mechanical Connectors

While Topobo has been successful at fulfilling my original design criteria, it still has much room for improvement. One problem is that LEGO connectors sometimes break and get stuck in passives. Conversely, sometimes large structures fall apart.

Cables

Almost everyone who plays with Topobo asks if we are going to make the wires disappear. We decided early on not to attempt this engineering goal because it would require integrating the electrical and mechanical connectors in order to distribute power and communications channels. Furthermore, all passives would need to be "smart" in order to rout communications unless a wireless communications network were used. One advantage to using cables to connect Actives is that it helps children understand and visualize network topology.

Actives

The Actives are too large and not the best proportions for Topobo. Ideally, all joints would be actuated with the exception of notches, and there would be no Passives. This approach is not possible because the current Actives are too heavy, but future developments in actuator technology may facilitate this goal.

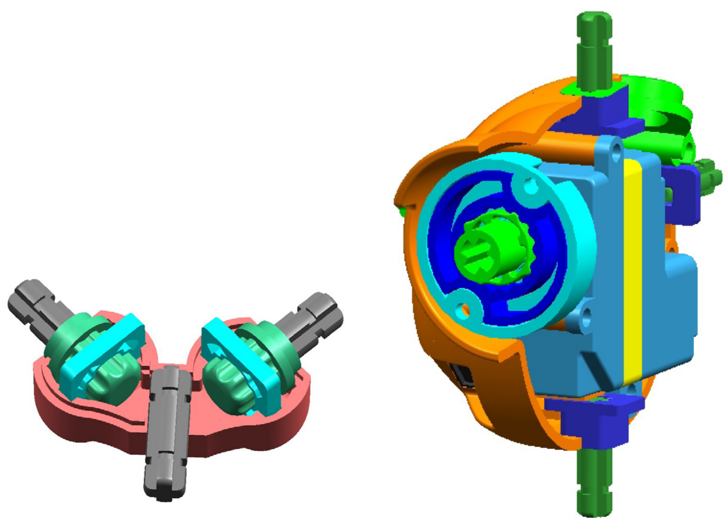
One major benefit of smaller and stronger actuators would be in mesh construction. As described earlier, meshes require looping structures for strength and stability. In order for meshes to be ergonomic, rings of Actives need to be small and flexible, which is not possible with the current implementation of Topobo.

The system's most major mechanical limitation that rotary motion is a very limited representation of flexible systems. Two- or three-DOF actuators would profoundly improve the types of structures that could be built and animated with Topobo. Linear actuators would also be a welcome addition and I hope that future developments in actuated modeling systems address this limitation.

From prototype to product

Despite the care that went into our prototypes, children reliably broke our hand-made parts. As part of the iCampus project, we received an educational outreach grant to get Topobo out of the lab - more permanently than user studies would allow. I used our modest grant to arrange production of parts with a Chinese manufacturer of electronic toys, with whom I had an existing relationship. They were interested in doing a research project, and together we set to redesigning the system for production.

Our process began by copying the prototypes with production issues in mind (manufacturability, assembly, technical issues such as shrinkage and mold design) and a focus on cost reduction. I took the opportunity to redesign aspects of the system, including the layout of usb plugs, increasing plugs from 3 to 4, upgrading the microcontroller, and improving reliability of the power scheme. A custom servo was designed that could reliably be backdriven, and clutches were added to passive parts. This process was arduous and incremental, and required two years of design, negotiation and production. It yielded steel injection molds rated to 1,000,000 cycles, with molding precision +/- 0.002", and the ability to order fully assembled, programmed and packaged "product." To my surprise, production Topobo is quite durable and allowed for the distribution to museums, teachers, and research collaborators to study the long term impact of the technology on a wide audience.



Production parts were reengineered from the ground up to optimize for manufacturing processes, ongoing R&D concerns, and durability in the field.

Future engineering of actuated modeling systems

Technological forecasts are almost always wrong. However, I have a few ideas how I might "do it differently next time," so here are a few thoughts for other actuated modeling systems.

Future actuated modeling systems will need to follow the dominant engineering paradigm "smaller, faster, cheaper." This especially applies to the mechanical components of the system. New actuators need higher strength to weight than modern servo motors. However,

The iCampus educational outreach project funded the mass production of Topobo, over a two year development cycle.

weak actuators may be useful because some loop structures can achieve strength with a large number of weak actuators.

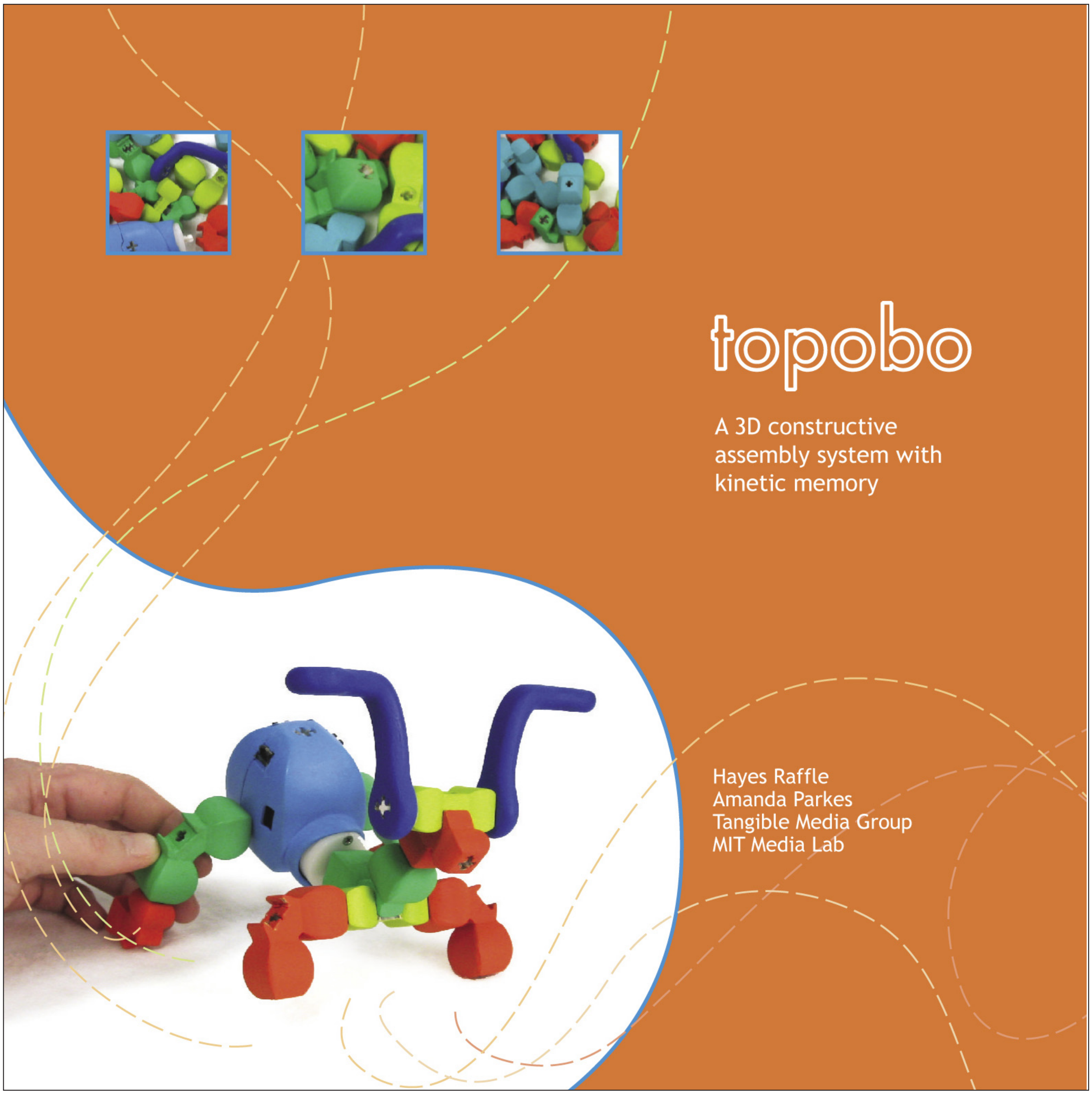
Researchers in "smart materials" are working on other approaches to actuation that are more similar to biological muscles that contract rather than rotate. Ideally, smart materials will serve as sensors or even generate power when they are manipulated. As Arthur Ganson suggested [Raffle, personal communication], if they are small enough, they might have binary states (e.g. short and long), and granularity would come from cascading many actuators in series. In any arbitrary manipulation, some actuators would be short and others would be long, giving a "smooth" overall effect.

These elements will need to be assembled. At the finger scale, one would use mechanical connectors that are electrically sensed so that the assembled structure could determine its overall shape. At a MEMS scale, such a system may use chemical interactions to communicate and establish physical topology. Such actuators would have to be self assembling.

The electrical element of such systems (if they are not superseded by chemical control structures) might be modeled on Butera's "paintable computers" [But02] and programmed in a language like Protobo, that use massively parallel computation and communication to process massive amounts of data. Such a system has the potential to be scalable, small, and effective for applications to engineering smart materials.

Appendix B
Topobo Brochure

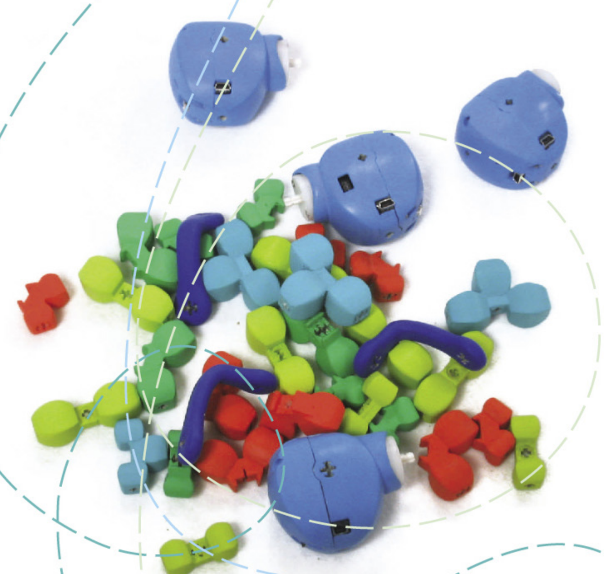
This brochure was originally designed for the ID Magazine Annual Design Review competition, for which we won second prize. Back-packs, Remix and Robo were later added to it, since it was such an effective communication tool. But people kept asking, where did you buy this new toy?


topobo
A 3D constructive
assembly system with
kinetic memory
Hayes Raffle
Amanda Parkes
Tangible Media Group
MIT Media Lab

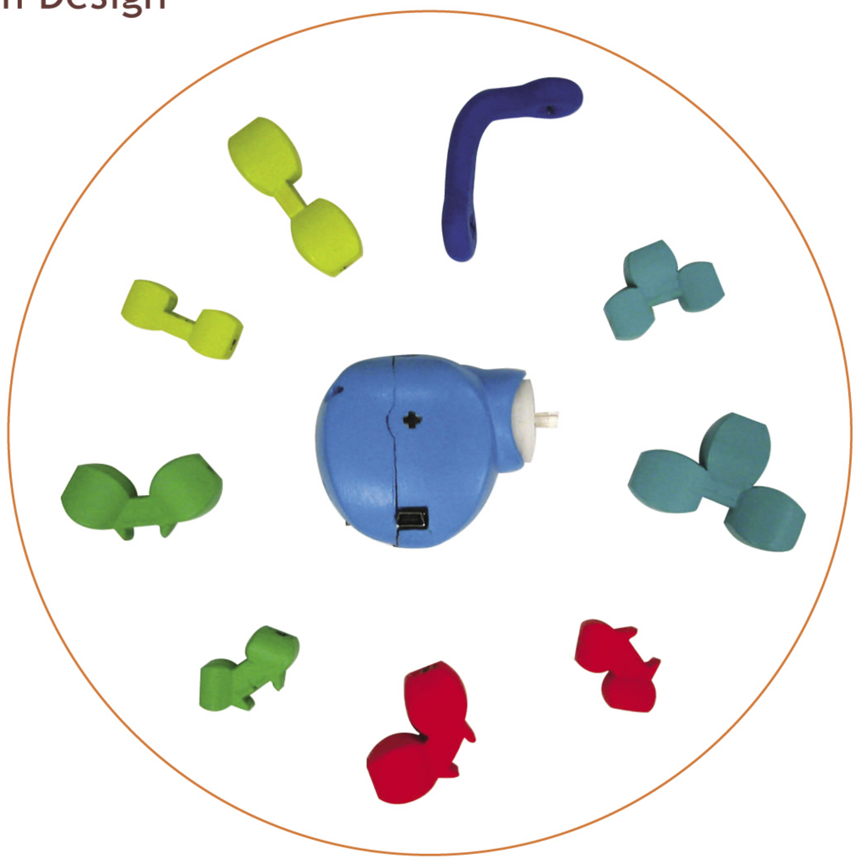
The Concept

What is it like to sculpt with motion?

Topobo combines the physical qualities of a building toy with gestural recording capability producing a means for dynamic expression with the press of a button and the flick of a wrist. Topobo works like an extension of the body, giving one's gestural fluency computation and memory.

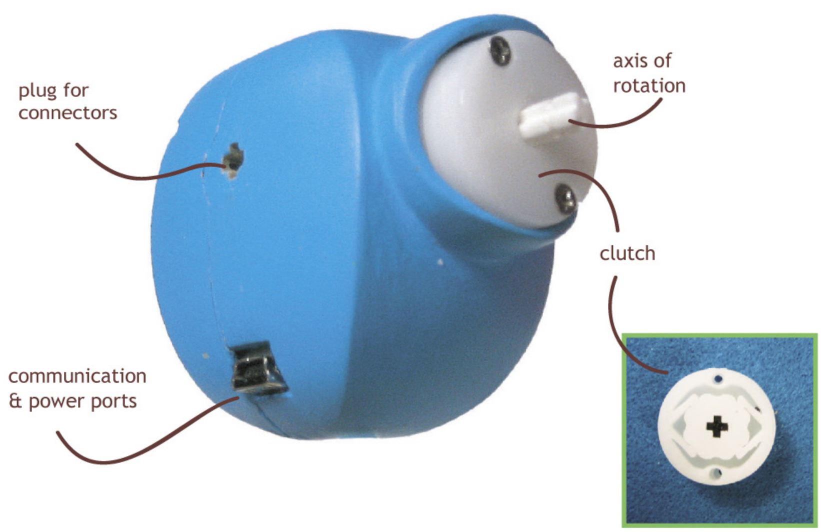


System Design

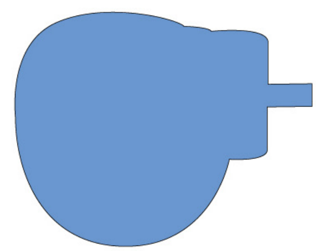


The Topobo system is comprised of ten primitives that are connected with Lego Technics® connectors. Nine of these primitives are called "Passive" because they form static connections. One motorized "Active" forms dynamic connections which allows the system to reproduce manipulations to a structure.

The Active

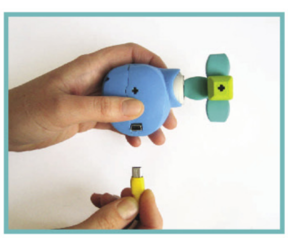


inside the clutch

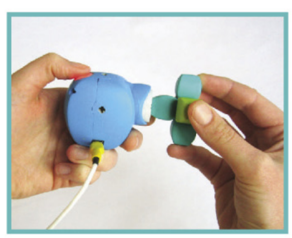


The Topobo Active uses a servo motor that has a built-in position sensor. A clutch protects the servo's gear train.

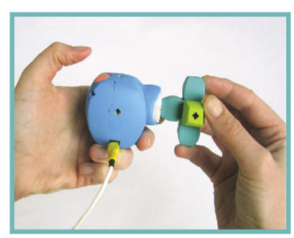
programming an active



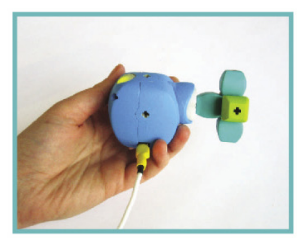
plug in the active



press the button to record

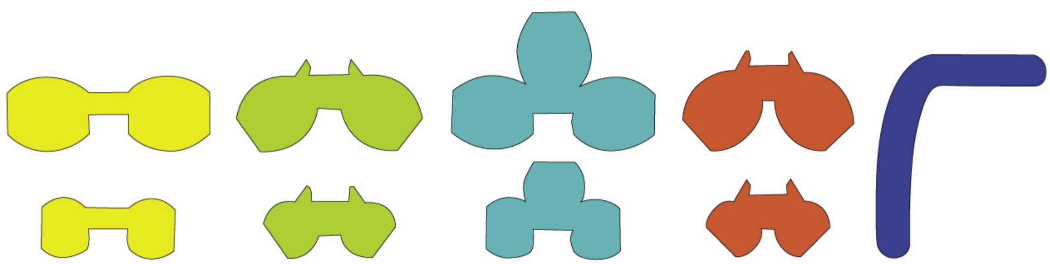


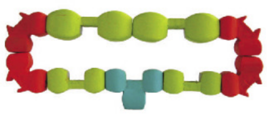
turn the axis with a motion

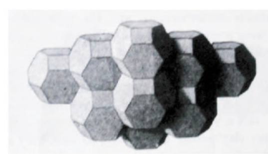


press the button for playback

The Passives

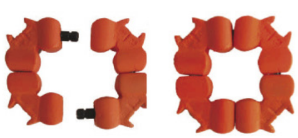


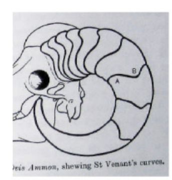


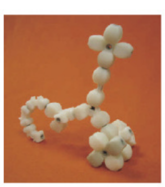


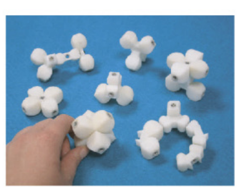
The Topobo passives can be connected end to end or with their central notches. The Topobo geometry allows for spatial loops

growth patterns & crystal structure inspired the geometry of topobo

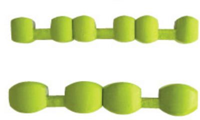




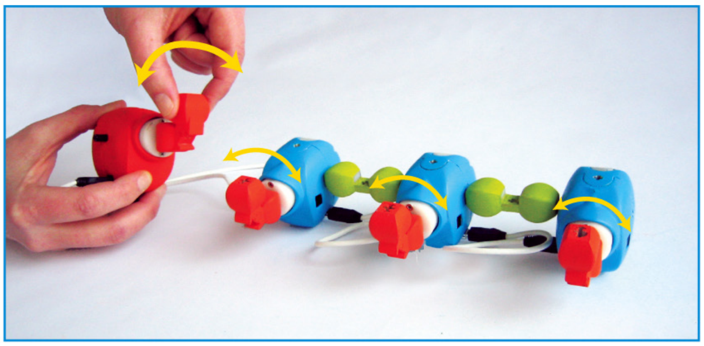




scaling 3:2

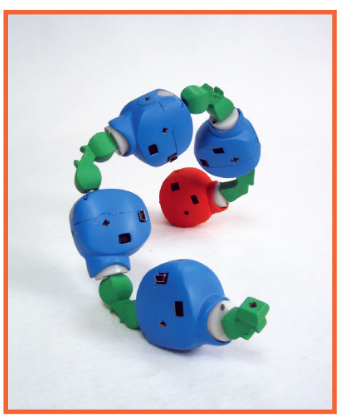


The Queen



Special orange actives called Queens can control many other Actives. In both record and playback, all motions made to a queen are mimicked by the other actives connected to the Queen.

centralized control



actives connected with tetrahedral passives create a spring-like helix when controlled by a queen.

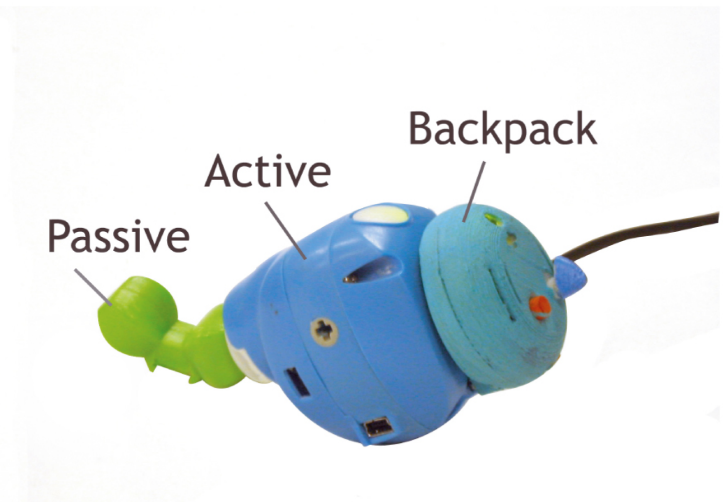
a linear sequence with a queen creates a circle

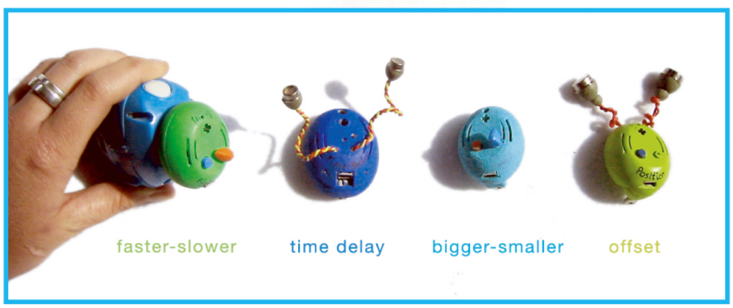
Queen with bigger-smaller backpack tells each active to increasingly scale the Queen's motion. a linear sequence creates a spiral

time delay backpack tells each active to wait before mimicking the Queen. a linear sequence creates a wave.

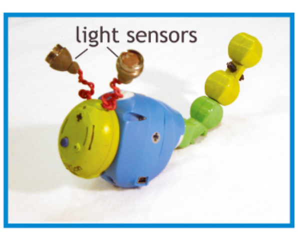
Backpacks

Backpacks can be attached to an Active to change the way a recorded motion will playback. By turning a knob or using sensors, children can adjust the *speed, timing, scale,* and *orientation* of playback.

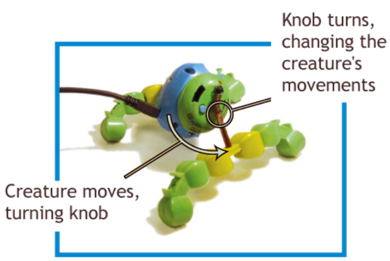




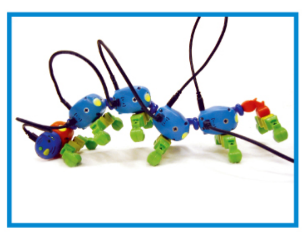
motion modulators



sensors replace the knob, and children can make creations that respond to ambient stimuli



its knob used as a joint, a backpack introduces the idea of feedback

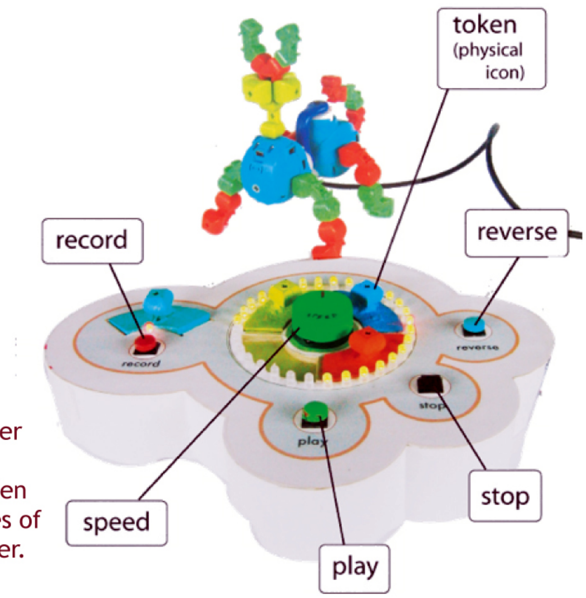


attached to a Queen, time-delay backpack lets children play with wave motions

Remix & Robo

global editing & control

Reaching beyond Topobo's direct record an play, we have created controllers for children to perform robotic narratives and have robot competitions.

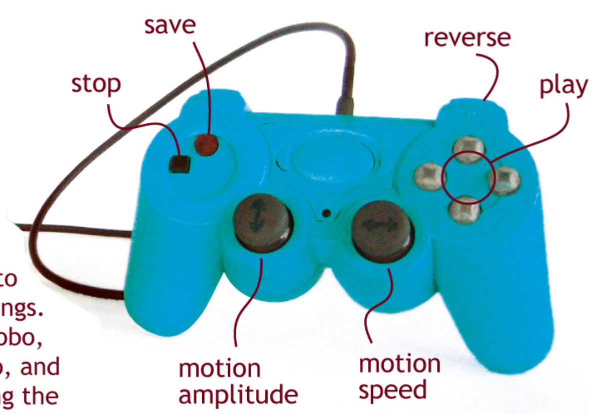


Remix

Remix is a tangible sampler-sequencer for capture and editing of motion recordings. A child can move a wooden block around to save a favorite series of footsteps or dance, and replay it later.

robot performances

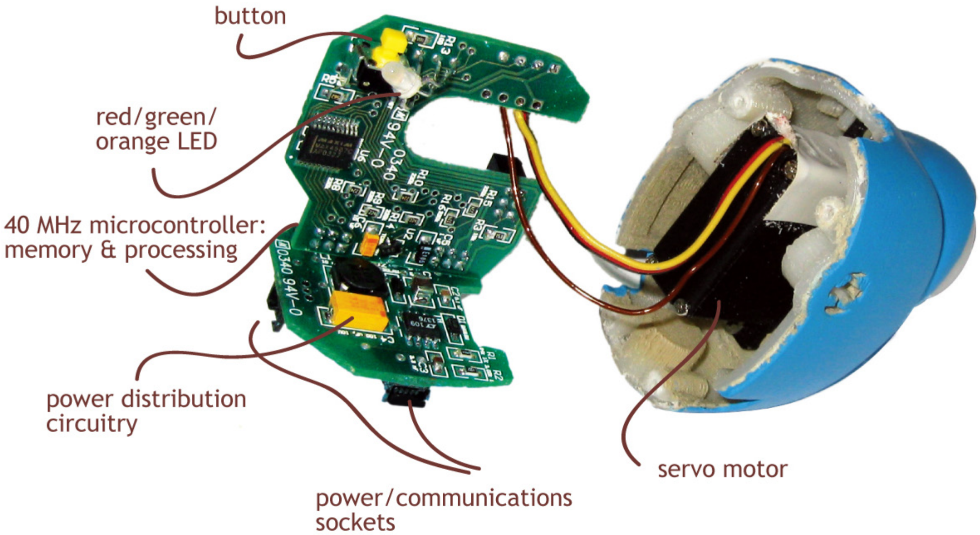
With Remix and Robo, children can make robotic puppet shows or have robot competitions.

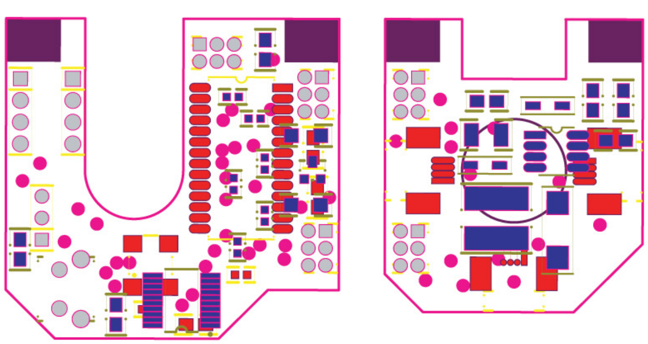


Robo

Robo is a modified game controller to capture and perform motion recordings. Children create recordings with Topobo, save the motion to a button on Robo, and then can replay the motion, changing the speed, scale or direction of playback.

Topobo Technology

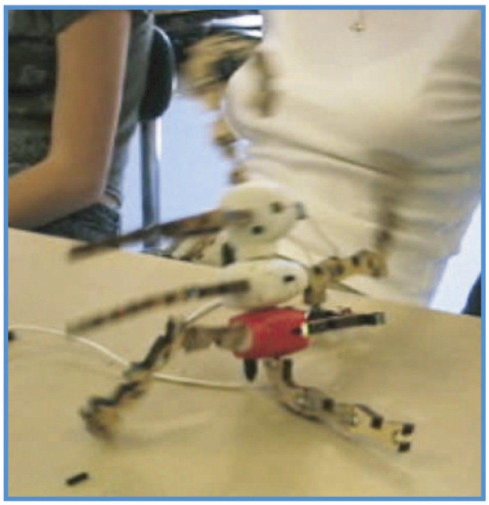




electronics design

Topobo is based on modular robotics technology. Electronics inside every Active handle local memory and processing and power distribution. A custom peer to peer serial network allows Actives to communicate with each other through small, white cables.

Topobo in action

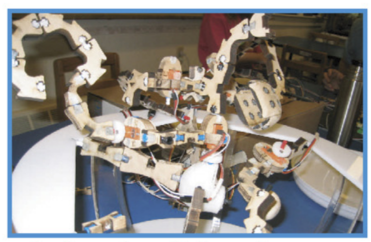


We found that Topobo can help students ages 7-13 to learn about:

- Balance
- Center of Mass/Center of Gravity
- Coordination
- Relative motion
- Movement with Multiple Degrees of Freedom
- Relationships between Local and Global Interactions

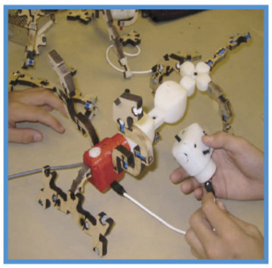
A tool for cooperative learning

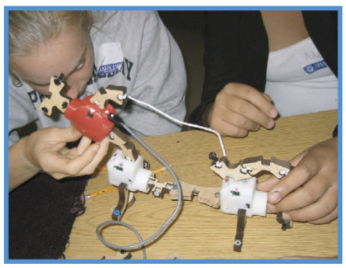
When we took Topobo in to the classroom, kindergartners, second graders and eighth graders cooperated to make Topobo creations. Younger kids told stories with topobo and did open-ended explorations. Older kids focused on trying to make things walk.

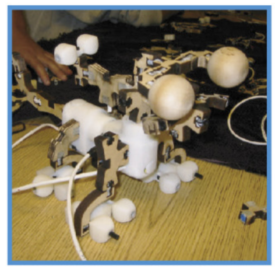


a 2nd grade collaboration

two 8th graders programming together

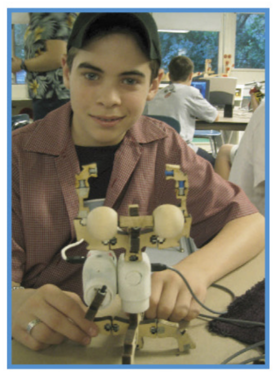




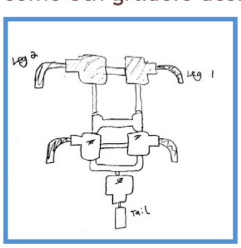


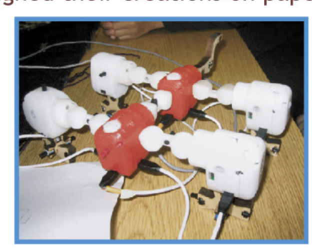
Animals & Machines

Kindergartners, second graders and eighth graders all related to Topobo models with their "familiar knowledge" about animals and machines. Metaphoric allusions to machines (robotics) and especially to animals ("the elephant," "the ant," "the scorpion," "the horse," "the no-walking man") were descriptive and salient. Many 8th grade students changed their creations based on their ideas about how animals and people move.

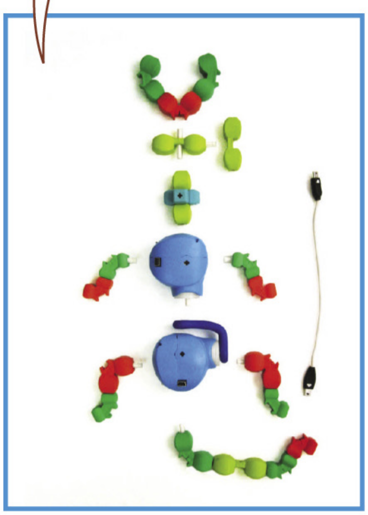


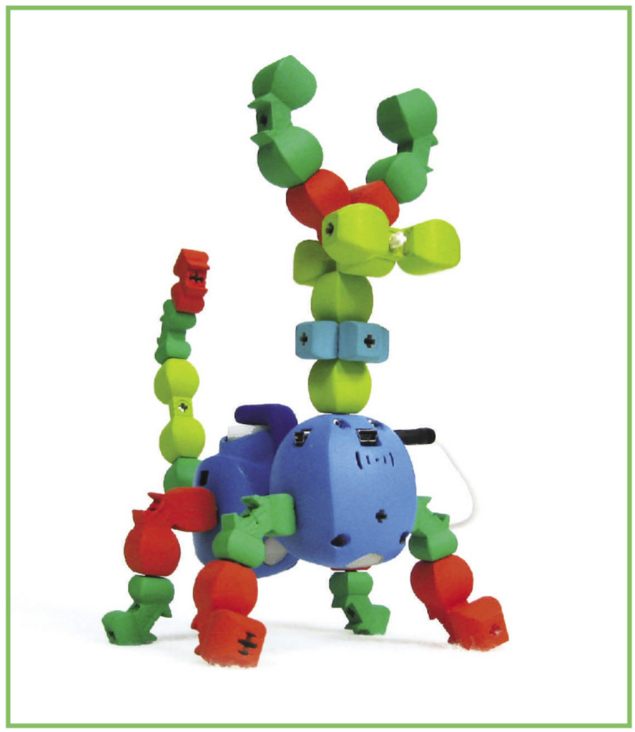
some 8th graders designed their creations on paper



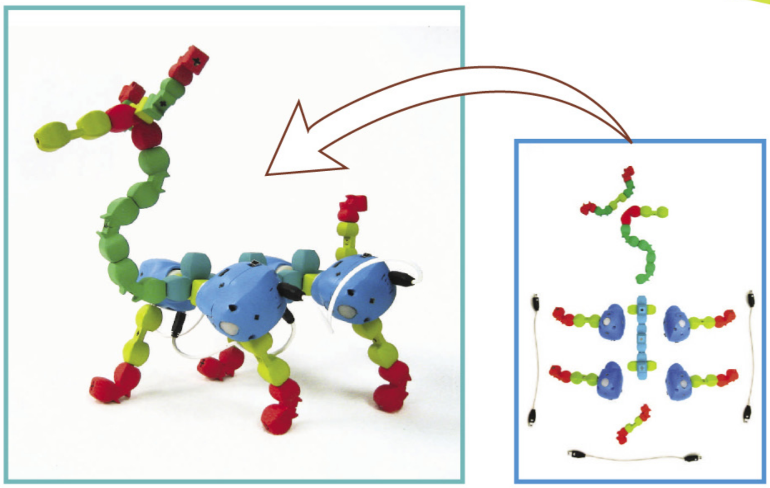


Topobo Creations

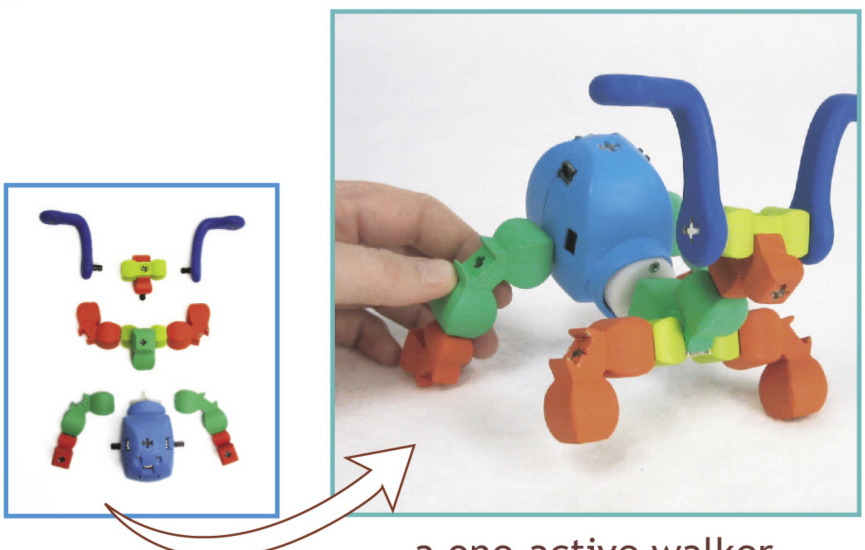




a moose

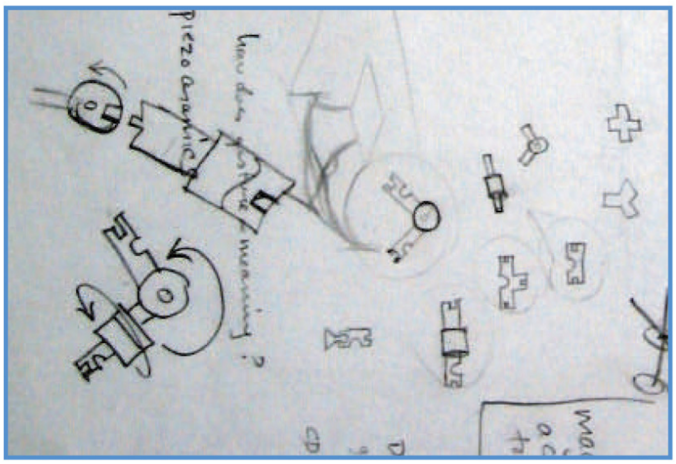


a griffin

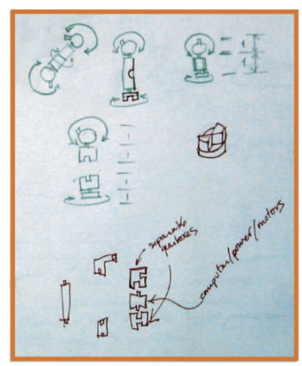


a one-active walker

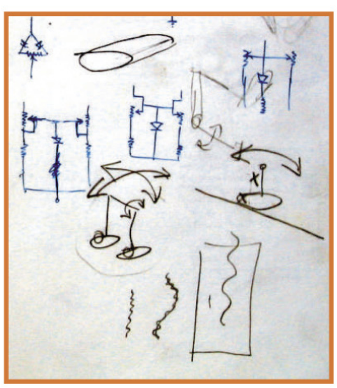
Concept & Early Prototypes



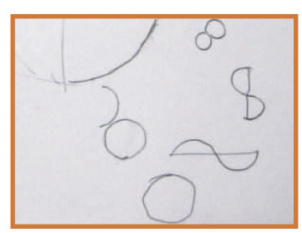
After choosing motors as our actuators, we did drawings to study spatial relationships possible with different arrangements of pivot joints. A branching system was an early choice for geometry.



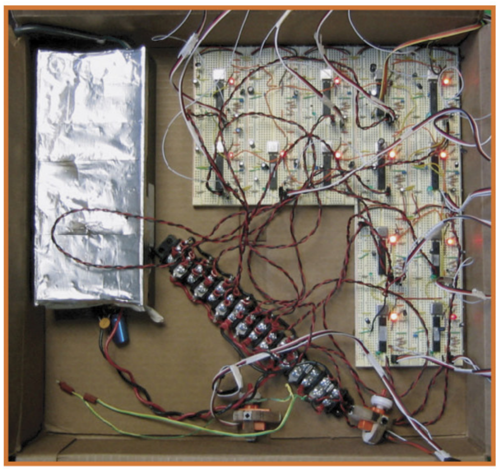
assembly drawings questioned the possibility of fitting motors and electronics in one designed system.



studies of passive walking toys and electronic systems compare concepts of feedback and emergence in mechanical and electronic systems.

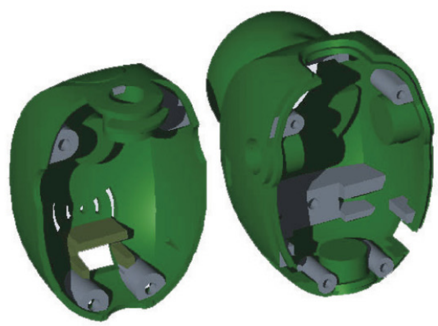


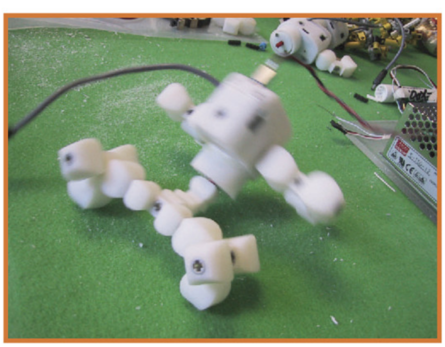
early studies of waves led to the time delay Queen.



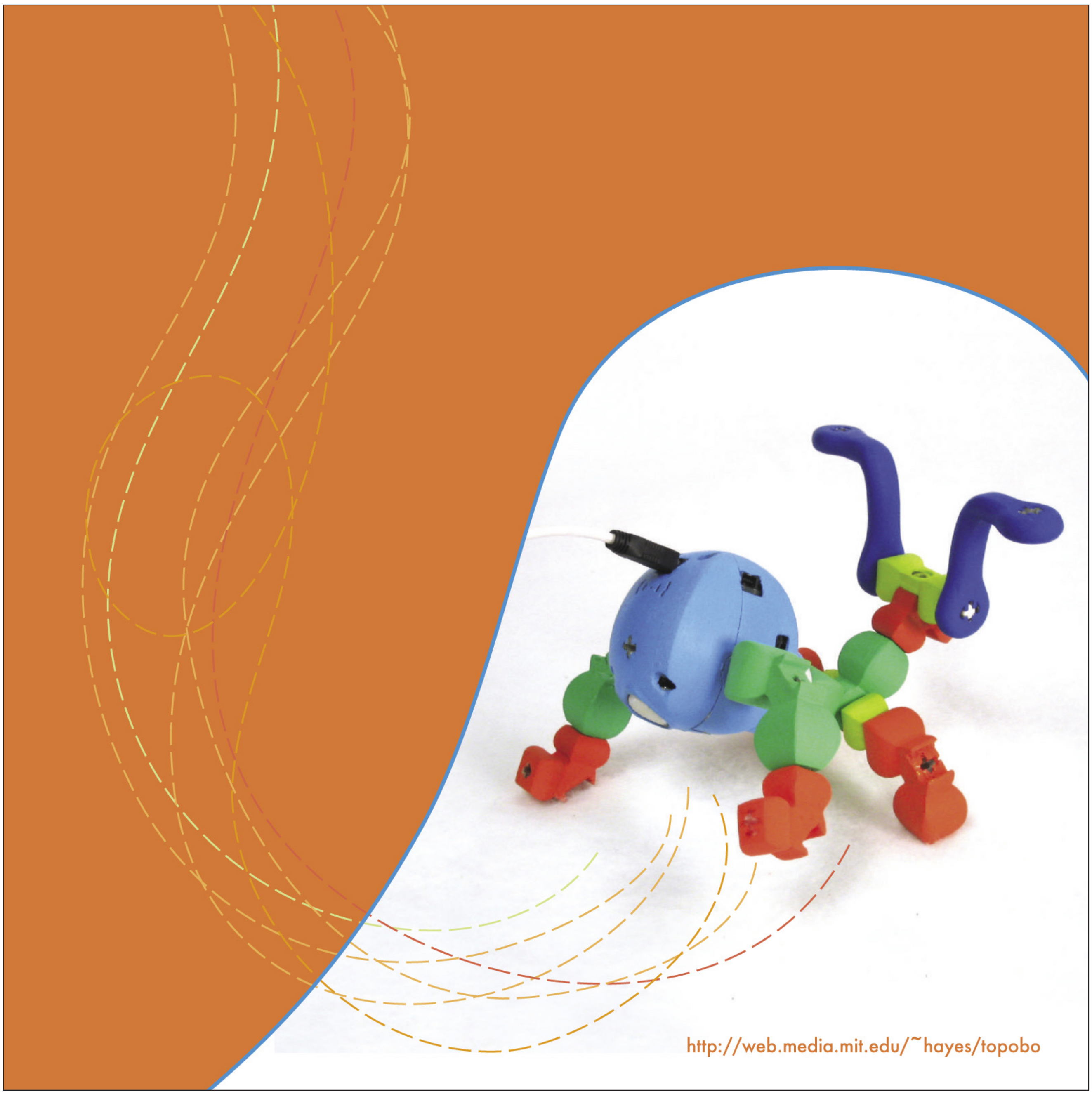
In order to avoid the "spaghetti" of wires we had with our prototypes, we needed to design a power distribution scheme to power all of the parts in a creation through one daisy-chained set of cables

Electronics design began on paper and progressed through many series of iterations from "breadboarded" electronics to printed circuit boards.





After the wooden prototypes, we modelled a plastic housing and printed plastic parts on an FDM 3D printer.


http://web.media.mit.edu/~hayes/topobo

13 References

[Ack96] Ackermann, E. "Enactive Representations in Learning: Pretense, Models, and Machines." In Bliss, J., Light, P. and Saljo, R. eds. *Learning Sites: Social and technological contexts for learning*, Elsevier, 1999, 144-154.

[Ack99] Ackermann, E. Perspective-taking and object construction: two keys to learning, in Kafai, Y. and Resnick, M. eds. *Constructionism in practice: designing, thinking, and learning in a digital world*, Lawrence Erlbaum, Mahwah, NJ, 1996, 25-35.

[Ann01] Ananny, M. *Telling Tales: A new way to encourage written literacy through oral language*. PhD Thesis, Media Lab, MIT: available http://web.media.mit.edu/~ananny/thesis.html [28th February, 2003], 2001, 165.

[Ann02] Ananny, M., Supporting Children's Collaborative Authoring: Practicing Written Literacy While Composing Oral Texts, in *Proceedings of Computer Support for Collaborative Learning,* (Boulder, Colorado, USA, 2002), Lawrence Erlbaum Associates, 595-596.

[Ard07] Arduino Programming Environment: http://www.arduino.cc/

[Bac07] New England Programming Language Seminar, TUFTS (2007-04-11), *Spatial Programming in Proto*.

[Bea06] Beal, J and Bachrach, J. Infrastructure for Engineered Emergence on Sensor/Actuator Networks. IEEE Intelligent Systems, (Vol. 21, No. 2) pp. 10-19, March/April 2006. http://hdl.handle.net/1721.1/32988

[Bel00] Bell, John. *Strings, Hands, Shadows: A Modern Puppet History*. Detroit Institute of Arts (2000).

[But02] Bill Butera. (2002) *Programming a Paintable Computer*. Ph.D thesis, Massachusetts Institute of Technology.

[Bra86] Braitenberg, V. (1986). *Vehicles: Experiments in Synthetic Psychology*. Cambridge: MIT Press. 1986.

[Bra03] Bratzel, B. (2003). Interviews with Barbara Bratzel, an elementary educator for over 15 years in a constructivist school of students ages 8-13.

[Bra98] Brave, S., Ishii, H. and Dahley, A. (1998). Tangible interfaces for remote collaboration and communication, *Proceedings of CSCW98*, 169-178. ACM Press.

[Bro97] Brosterman, N. (1997). *Inventing kindergarten*. New York, Harry N. Adams, Inc.

[Bru73] Bruner, J. (1973). Organization of early skilled action. In *Child Development* 44: 1-11. Chicago: University of Chicago Press.

[Cas01] Cassell, J. and Ryokai, K. (2001) "Making space for voice: Technologies to support children's fantasy and storytelling." *Personal technologies 5*(3): 203-224.

[Chi97] Chi, M. Why is self explaining an effective domain general learning activity? in Glaser, R. ed. *Advances in Instructional Psychology,* Lawrence Erlbaum Associates, 1997.

[Col01] Cole, M., and Cole, S. *The Development of Children, Fourth Edition*. New York, NY : Worth Publishers, 2001.

[Col98] Coleman, M., Ruina, A. (1998). An Uncontrolled Toy That Can Walk But Cannot Stand Still. *Physical Review Letters,* April. 80(16): 3658 - 3661.

[Dat07] Dataflow programming languages. http://en.wikipedia.org/wiki/Dataflow_language

[Duf98] Duff, Yim, et al. US Patents 6,454,624: Robotic toy with posable joints, 6,575,802: Robotic toy modular system with distributed program and 6,605,914: Robotic toy modular system.

[Exp07] Exploratorium Museum: http://www.exploratorium.edu

[Fre06] Fernaeus, Y. and Tholander, J. Finding Design Qualities in a Tangible Programming Space. *CHI 2006*.

[Fre00] Frei, Phil. (2000). *curlybot: Designing a new class of computational toys*. Master's thesis, Massachusetts Institute of Technology.

[Fre98] Frei, Su, Mikhak and Ishii. curlybot: Designing a New Class of Computational Toys. *Proc. CHI 2000*. ACM Press (2000).

[Ful98] Full, R.J., Autumn, K., Chung, J.I., Ahn, A., Rapid negotiation of rough terrain by the death-head cockroach. *American Zoologist*. 38:81A. 1998.

[Fur74] Furth, H. & Wachs, H. (1974). *Thinking goes to school; Piaget's theory in practice*. New York: Oxford University Press.

[Gar83] Gardner, H. (1983). *Frames of mind: The theory of multiple intelligences*. New York: Basic Books.

[Han04] Hancock, Marjorie R. (2004). *A celebration of literature and response*. Portsmouth NH, Heinemann.

[Hei92] Heise R. (2002). Programming Robots by Example. In Research Report No. 92/476/14, Department of Computer Science, University of Calgary.

[Ijs98] Ijspeert, A., Hallam, J. and Willshaw, D. From lampreys to salamanders: evolving neural controllers for swimming and walking. In *Fifth International Conference on Simulation of Adaptive Behavior,* pages 390-399, 1998.

[Ish97] Ishii, H. and Ullmer, B. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings on human factors in computing systems* (CHI1997), 234-241. ACM Press.

[Kam Kamimura, A., Kurokawa, H., Yoshida, E., Tomita, K., Kokaji, S, and Murata, S. Distributed Adaptive Locomotion by a Modular Robotic System, M-TRAN II. *Proceedings of IEEE/RSJ International conference on Intelligent Robots and Systems*. 2004. 2370-2377.

[Klo02] Klopfer, E., T. Um. (2002). Young adventurers - modeling of complex dynamic systems with elementary and middle school students. *International Conference on the Learning Sciences.*

[Kot99] Kotay, Rus, Vona, and McGray. (1999). The self-reconfiguring molecule: Design and control algorithms. Workshop on Algorithmic Foundations of Robotics.

[Lab75] Laban, R. (1975). Reprint, 1956. *Principles of dance and movement notation 2nd ed.* Boston: Plays, Inc.

[LEG07] LEGO Mindstorms. http://mindstorms.lego.com.

[Log04] Logoblocks. (2004). http://llk.media.mit.edu/projects/cricket/doc/help/logoblocks/startingwithlogoblocks.htm

[Mar03] Marshall, P., Price, S., and Rogers, Y. Conceptualising tangibles to support learning. *Proceedings of Interaction Design and Children*, Preston, England, July 1-3, pages 101-110. 2003.

[Mcn04] McNerney, T. S. 2004. From turtles to Tangible Programming Bricks: explorations in physical language design. Personal Ubiquitous Comput. 8, 5 (Sep. 2004), 326-337. DOI= http://dx.doi.org/10.1007/s00779-004-0295-6

[MAX07] MAX/msp. http://en.wikipedia.org/wiki/Max

[Mel94] Mellar, H. and Bliss, J. Introduction: modelling and education, in Mellar, H., Bliss, J., Boohan, R., Ogborn, J. and Tompsett, C. eds. *Learning with artificial worlds: computer-based modelling in the curriculum,* The Falmer Press, London, 1994, 1-7.

[Mon12] Montessori, M. *The Montessori Method*. Translated from 1912 original by Anne George. New York: Schocken Books (1964).

[Muy64] Muybridge, E., Man Walking on Inclined Plane: http://texaschapbookpress.com/muybridgeincline.htm

[o'Ma05] O'Malley, C. and Fraser, D. S. Literature Review in Learning with Tangible Technologies. *NESTA Futurelab Report 12* (2005).

[Pap80] Papert, S. (1980). *Mindstorms: Children computers and powerful ideas*. Cambridge, Massachusetts: Perseus Publishing.

[Par08] Parkes, A., Raffle, H., Ishii, H. Topobo in the Wild: Longitudinal Evaluations of Educators Appropriating a Tangible Interface. *Proceedings on Human Factors in Computing Systems* (CHI 08). ACM Press.

[Pat06] Patten, J., *Mechanical Constraints as Common Ground between People and Computers*, PhD Thesis, MIT Media Lab, 2006.

[Pip02] Piper, B., Ratti, C., and Ishii, H., Illuminating Clay: A 3-D Tangible Interface for Landscape Analysis, in *Proceedings of Conference on Human Factors in Computing Systems* (CHI '02), (Minneapolis, Minnesota, USA, April 20 - April 25, 2002), pp. 355-362.

[Pes03] Pestalozzi, H. (1803). *ABC der Anschauung, oder Anschauungs-Lehre der Massverhaltnisse*. Tubingen, Germany: J.G. Cotta.

[Pia52] Piaget, J. (1952). *The origins of intelligence in children, 2nd edition*. (M. Cook, Trans.). New York: International Universities Press.

[Pia76] Piaget, J. (1976). *The grasp of consciousness*. Cambridge: Harvard University Press.

[Raf02] Raffle, H. (2002). The art and design of Hayes Raffle. http://www.rafelandia.com.

[Raf04] Raffle, H., Parkes, A., Ishii, H. Topobo: A Constructive Assembly System with Kinetic Memory. In *Proceedings on Human Factors in Computing Systems* (CHI 2004), 869-877. ACM Press.

[Raf06] Raffle, H., Parkes, A., Lifton, J., Ishii, H. Beyond Record and Play: Backpacks—Tangible Modulators for Kinetic Behavior. *In Proceedings on Human Factors in Computing Systems* (CHI 2006), 427-437. ACM Press.

[Raf03] Raffle, H., Tichenor, J. and Joachim, M. (2003). Super Cilia Skin: An interactive membrane. In *Extended proceedings on human factors in computing systems* (CHI2003), 529-530. ACM Press.

[Raf07] Raffle, H., Yip, L., Ishii, H. Remix and Robo: Sampling, Sequencing and Real Time Control of a Tangible Robotic Construction System. In *Proceedings on Interaction Design and Children* (IDC 07). ACM Press

[Rao04] Rao, S. *Mechanical Vibrations, Fourth Edition*. Upper Saddle River, N.J. : Pearson/Prentice Hall, 2004.

[Res99] Resnick, M. (1999). Decentralized modeling and decentralized thinking. In Feurzeig, W. and Roberts, N. (Eds). *Modeling and simulation in precollege science and mathematics* (114-137). Springer: New York.

[Res98] Resnick, Martin, Berg, et al. (1998). Digital manipulatives: New toys to think with. In Paper Session, *Proceedings on human factors in computing systems* (CHI1998), 281-287. ACM Press.

[Rog97] Rogers, Yvonne and Bellotti, Victoria, Grounding blue-sky research: how can ethnography help?, *interactions*, v.4 n.3, p.58-63, May/June 1997.

[Rui04] Ruina, A. (2004). Passive dynamic walking at Cornell. http://tam.cornell.edu/~ruina/hplab/pdw.html

[Ryo04] Ryokai, K., Marti, S., Ishii, H. I/O Brush: Drawing with Everyday Objects as Ink, in *Proceedings of CHI 04.*

[San04] San Francisco Exploratorium (2004). http://www.exploratorium.edu

[Sch06] "Schweikardt, E. and Gross, M. roBlocks: A Robotic Construction Kit for Mathematics and Science Education. Proceedings of ICMI'06, November 2–4, 2006, Banff, Alberta, Canada."

[Sha02] Shankin, E. A. (2002). Art in the information age: Technology and conceptual art. In Michael Corris, ed., *Invisible College: Reconsidering "Conceptual Art."* Cambridge: Cambridge University Press.

[Sim94] Sims, K. Evolving Virtual Creatures. *Proceedings of SIGGRAPH 94* pp.15-22.

[Sod07] Sodaplay. http://www.sodaplay.com

[Suz93] Suzuki, H., and Kato, H. "AlgoBlock: A Tangible Programming Language." In *Proceedings of the 4th European Logo Conference,* August 1993. pp. 297-303.

[Tan79] Tanner W. ed. (1979). *Industrial Robots. Volume 1:* Fundamentals. Society of Manufacturing Engineers, Dearborn, MI.

[Th042] Thompson, D. (1942). *On Growth and Form: The Complete Revised Edition.* New York: Dover Publications Inc, 1992. Reprinted from the original Cambridge University Press publication, 1942.

[Ull99] Ullmer, B., and Ishii, H. (1999) MediaBlocks: Tangible interfaces for online media (video). In *Extended Abstracts of Conference on Human Factors in Computing Systems* (CHI1999). 31-32. ACM Press.

[Vor03] Vorenberg, Amy. (2003). Interviews with Amy Vorenberg, an elementary educator and school principla for over 15 years in a constructivist school of students ages 5-13.

[Vyg78] Vygotsky, L.S. (1978). *Mind in Society.* Cambridge: Harvard University Press.

[Wis98] Wisneski, C., Ishii, H., Dahley, A., Gorbet, M., Brave, S., Ullmer, B., Yarin, P. (1998). Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information. In *Proceedings of International Workshop on Cooperative Buildings* (CoBuild1998), 22-32. Darmstadt, Germany: Springer Press.

[Wye02] Wyeth, P., Purchase, H. (2002) Tangible Programming Elements for Young Children. In *Proceedings of Conference on Human Factors in Computing Systems* (CHI2002). 774-775. ACM Press.

[Yim00] Yim, Duff, Roufas. (2000). PolyBot: a Modular Reconfigurable Robot, *IEEE International Conference on Robotics and Automation,* San Francisco, CA.

[Zoo04] ZOOB Toys. http://www.infinitoy.com.

[Zuc05] Zuckerman O., Arida, S., and Resnick M. (2005). Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives. *Proceedings of CHI 2005*.