Topobo:
A 3-D Constructive Assembly System with Kinetic Memory

Hayes Solos Raffle

B.A. in Fine Arts (Sculpture), Yale University, May 1996

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on May 7, 2004
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the Massachusetts Institute of Technology

May 2004

Author
Hayes Solos Raffle
Program in Media Arts and Sciences
May 7, 2004

Certified by
Hiroshi Ishii
Associate Professor of Media Arts and Sciences
Program in Media Arts and Sciences
Thesis Supervisor

Ac
An
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Topobo:
A 3-D Constructive Assembly System with Kinetic Memory

Hayes Solos Raffle

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on May 7, 2004
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the Massachusetts Institute of Technology

Abstract

We introduce Topobo, 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, 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.

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 tangible 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.

Thesis Supervisor:
Professor Hiroshi Ishii
Associate Professor of Media Arts and Sciences
MIT Media Lab

Topobo:
A 3-D Constructive Assembly System with Kinetic Memory

Hayes Solos Raffle

The following people served as readers for this thesis:

Thesis Reader
Mitchel Resnick
Associate Professor of Media Arts and Sciences
MIT Media Laboratory

Thesis Reader
Arthur Ganson
Sculptor, Inventor and former Artist-in-Residence, MIT

Contents

Acknowledgements

Amanda Parkes and I have jointly developed Topobo over the past year and a half with the help of many colleagues and undergraduate researchers. When I write "we," I am generally referring to Amanda Parkes and myself, although "we" may also refer to any number of undergraduates and graduate students who helped engineer and build the system. I owe all of these people gratitude and thanks for their support.

I am particularly grateful for Hiroshi Ishii, who has provided the environment, people and support that has allowed me to be creative and accomplish my work. I would like to thank my readers Mitchel Resnick and Arthur Ganson for their guidance and support through the development of the system and through the development of the thesis. Both Mitchel and Arthur have provided inspiration for the work and have guided the project at critical junctures.

I have great admiration for Amanda Parkes, who has helped to develop Topobo from a nascent idea to a beautiful and functional system. Her persistence, hard work, design expertise, intuition and emotional support guided the project's growth.

Like the collaborative social interactions that it was designed to support, Topobo's development has been a collaborative and social process. Josh Lifton volunteered his expertise in programming distributed microcontroller systems to make the firmware clean and robust. As well as good code, I found in Josh a friend. A number of talented MIT undergraduates helped with engineering through the UROP program. Brian Mazzeo's early work developing hardware and software proved that the Topobo concept was viable. Jeremy Schwartz contributed an intuition for form and mechanical engineering in helping develop the physical system and Laura Yip spent countless hours developing good code. Our colleague Scooby propped up our efforts when we were at dead ends. Andy Lieserson has shown me the virtues of a focused young engineer in his development of robust electronics, and Wes Jin contributed ideas and patience to understanding motor interfacing. Elysa Wan has always lent a helping hand. Thank you all for your enthusiasm and dedicated work on the project.

The entire TMG team has played a role in this project. James Patten has offered advice and an occasional raised eyebrow helping to guide the system. Monzy has been supportive of the conceptual potential for the project, and Gian Pangaro has been instrumental in helping me to learn electronics. Media Lab colleague Ben Recht contributed his inspiration to use peer-to-peer algorithms to control a scalable system. Of course nothing would happen at TMG without the group's social dynamics, and I am grateful to Kimiko, Chad, Dan, Kazue and Brad. On a daily basis, Lisa Lieberson kept the project moving forward.

Bakhtiar Mikhak supported exposing our early prototypes to "real kids" and convinced me to be willing to share my transitional work. He has also offered countless ideas to help advance the social implications of the project. Cristobal Garcia collaborated on early studies with young children and showed me a sociologist's perspective. His friendship and guidance has been refreshing. Jonathan Bacharach coined the name for this project, and many professional educators have supported the development of Topobo by arranging classroom visits, supporting our interactions with students, and offering feedback and advice.

For inspiration, I would like to thank Phil Frei for inventing curlybot and introducing me to TMG. Michael Grey introduced me to the design of building toys many years ago and taught me how to develop an idea.

There was a great deal of support from sponsors and friends of the Media Lab. In particular, the LEGO group has shown consistent interest and support for the project. I would like to thank them for their enthusiasm and fellowship over the past year. I would also like to thank the members of the TTT consortium who have supported our work in TMG and have made Topobo possible. Special thanks go out to my friend James Tichenor for always making me laugh and think hard about the real meaning of my work, and to Mitch Joachim for going crazy over my crazy ideas.

And most importantly, I would like to thank my family. Thank you Rachel for making this all worth while.

May 2004
Cambridge, Massachusetts

This thesis is dedicated to Rachel.
You have been my greatest teacher and supporter
over the past two years, and I love you.

Topobo:

A 3-D Constructive Assembly System with Kinetic Memory

Prologue

This thesis began with a simple question: how can one sculpt with motion? From this basis, Amanda Parkes and I developed Topobo. Topobo grew from seeds planted in the fields of sculpture and toy design and grew to integrate physical programming with educational manipulative tools to present a new kind of computerized tool for the elementary and middle school classroom. In these pages, I document both Topobo's design and development, and evaluate the system as a "digital manipulative" to make new ideas about motion accessible to kids through hands-on modeling. In the end, I'll come full circle and return to the idea of sculpting with motion, considering how this idea might facilitate the development of new tangible interfaces in the future. But first, let's address another question: why sculpt with motion? We'll get started with a bit of history about tangible interfaces and digital manipulatives.

1 Introduction

Tangible user interfaces (TUIs) take advantage of skills people have developed through working with physical objects.

This thesis explores how an actutaed modeling material can contribute to this idea.

The Tangible Media Group at the MIT Media Lab conducts research in "Tangible Interfaces." We are pursuing a vision to improve people's access to computers by creating computational media that take advantage of skills people have developed through working with physical objects [Ish97]. Contrasted with the familiar GUI (graphical user interface with a screen, "windows," keyboard and mouse) these platforms and digitally enhanced objects aim to bridge the intangible world of digital information with the physical world. One hallmark of tangible interfaces is the coupling, or coincidence, of input and output space. One of the more promising recent directions of our research is the use of actuation to represent change and enforce constraints in physical, computational interfaces. This thesis will begin to explore how actuation can be coupled with physical modeling (manipulative) materials to allow simple computational processes to represent complex phenomena.

Modeling is a theme for kids with educational manipulatives and for adults with computer modeling. How about tangible modeling with computers?

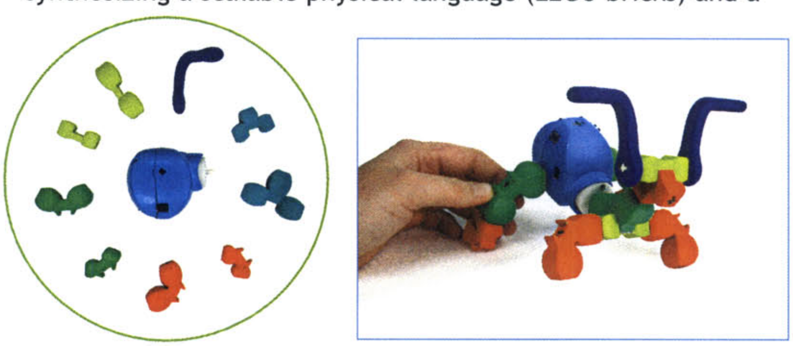
A major theme in the development of tangible interfaces has been to couple physical materials and computational models to create and control simulations [Ish97]. This is not surprising since both computer simulation and manipulative materials can be seen as systems for creating models of the world. The addition of computation to physical model making and simulation may be a natural progression in the field of human-computer interaction (HCI).

Model making is a prevalent activity in education that ranges from kindergartners' experimentation with wooden blocks to college physics students' computer simulation of material stress and strain. Model making allows rapid experimentation with a system to understand its limitations. Tangible interfaces present a unique opportunity to create a computationally augmented, actuated physical modeling system that takes advantage of the editability of computer data and the physical immediacy of a tangible model. Such an interface could help increase understanding of physical systems because the model is physical and its relationship to its environment is not simulated.

While such materials can be meaningful to both children and adults, this thesis will focus on the child's experience with an actuated modeling system for a number of reasons. There is a rich history of the development of modeling materials to help children learn about the world. Constructivist educators and researchers in "educational manipulatives" believe that children have an important need to learn about the world through working with physical materials. By playing with manipulative materials like blocks, building toys and construction kits, children can learn many complex patterns and ideas that support an intuitive understanding of the natural world and inform their later development of abstract thought [Bro97].

Recent efforts to couple computer programming and educational manipulatives have sought to introduce systems concepts to younger children. "Digital Manipulatives" embed computation in familiar children's toys to make accessible concepts that educators currently consider to be "too advanced" for children at a certain age [Res98]. Digital manipulatives can be viewed as a convergence of the educational manipulative tradition [Bro97], and Tangible Interfaces, which are designed to give physical form to digital information [Ish97]. Digital Manipulatives are intended to introduce complex ideas through hands-on modeling with integrated hardware and software systems. For example, Resnick et al. embedded a programmability into a building toy, synthesizing a scalable physical language (LEGO bricks) and a

Digital Manipulatives are educational manipulative tools that use computation to allow children to experiment with complex ideas. Coupling digital manipulatives with tangible interfaces can make the programming activities more like physical construction activities.



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

scalable computational language (LOGO). This work led to the development of the LEGO Mindstorms line that is successful in many schools today [Res98]. Topobo builds on these trends in digital manipulatives and introduces themes from tangible interfaces to make ideas about kinematic systems accessible to children.

1.1Topobo

Topobo integrates physical and motion modeling via tangible programming.

"Topobo" comes from "topology" and "robotics."

Topobo is a 3-D actuated modeling system that can be assembled into various configurations and then physically programmed to move. Topobo is akin to building toys such as LEGO® and ZOOB®. The difference is an introduction of motorized "Active" components with embedded kinetic memory. Topobo – for *topology* and *robotics* – includes "Passives" (static parts) and "Actives" (networkable, motorized parts) that can be snapped together to form models of animals, regular geometries, or abstract shapes. Topobo is different than existing robotic assembly systems such as LEGO Mindstorms in that the computation is distributed into the physical modeling system: the processes of physical and digital (or program) construction are both physical activities.

"Queens" allow one motorized part to control many motorized parts.

To use Topobo, a child builds a creation of their choosing and connects the Actives with small cables. To record a movement, the child presses a button on an Active, twists and moves the creation to program a sequence of behaviors, and then presses the button again. The creation immediately goes into playback mode, which repeatedly replays the child's input until the button is pressed a third time.

Usually Topobo is programmed by direct manipulation, where each Active synchronously records its own motion. However, Topobo has special Actives called "Queens" that control an entire network of Actives. This introduces the concept of centralized control and many possibilities.

Backpacks are physical parts that modify a recorded motion.

Special pieces called "Backpacks" can be attached to an Active to modify its playback motion. Backpacks behave differently when they are attached to a Queen: their modifications grow as they are passed from one Active to another. The design section of this thesis will explain how Backpacks can allow a user to use physical programming to physically edit motions, create sophisticated motions in large networks of Actives with simple rules, and provide "conditional" behaviors that allow creations to react to their environments.

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

Topobo was designed to facilitate learning about dynamic kinematic systems through both individual discovery and cooperative social interactions. Through several research studies with over 80 students ages 5-13 we found that Topobo is engaging for children at multiple cognitive levels, and can help children develop intuitions for how balance, leverage and gravity affect kinematic systems like ambulatory robots. Children's focused iteration with many surprising creations suggests that the simplicity of the Topobo interface encourages playful expression and rapid experimentation with different kinds of structures and motions. I will report the details of our experiments in the Evaluations section of this thesis. Since Topobo was designed to introduce new fields of knowledge to young children, the thesis will include a thorough design overview that includes a set of possible play scenarios for Topobo to highlight different educational concepts.

Kids ages 5-13 played with Topobo and learned about the complex behaviors that result from simple motions.

1.2 Thesis Overview

In the following pages, I will place equal emphasis on the design and educational implications for Topobo. Topobo was designed to be a multi-purpose actuated modeling material, but this thesis will focus on digital manipulatives as one possible application of the material. The design will reveal that the system foundation is general but extensions to the system specifically address children's education. My evaluations and discussion of Topobo's educational implications target a child's constructivist education.

Topobo was designed to be a general purpose kinetic modeling medium. We'll talk about the design of the system, but mainly look at educational applications.

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 1: Motivation to create the work. How I got from sculpture to tangible interfaces.

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 2: Educational foundation and related robotics work.

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

Ch 3: Design studies explored a variety of approaches and enabling technologies.

Ch 4 & 5: The system design, including its evolution and the current system and how we built it.

Ch. 4 Topobo System Design describes the current system, including the design principles, the development of different versions of Topobo, and an explanation of how you can use the many different Topobo components. System Design concludes with a critique of some of Topobo's limitations.

Ch. 5 Topobo Engineering describes how we engineered and built the system. This includes, mechanical, electromechanical and electrical engineering as well as a qualitative description of the firmware.

Ch 6 & 7: What did kids do with Topobo, and what can they learn from it?

Ch. 6 Studies with Children report our findings from studies in school classrooms with students ages 5-13. This section reports our findings and some design changes prompted by the interactions.

Ch. 7 Educational Implications describes the domains of knowledge children can explore with Topobo. I explain how the system is designed to reveal complexity to children in a leveled manner, and may support their growth through Piagetian stages of development.

Ch 8: How can the system be further developed to support more kids at more ages?

Ch. 8 Future Work with Topobo as a Digital Manipulative will consider extensions to the system, including GUI-based approaches. The general focus will be a discussion of how to balance the physical and digital elements of a digital manipulative.

Ch 9: What else could you do with Topobo? Could it be a sculpting medium to design various tangible interfaces?

Ch. 9 A TUI Material? will look ahead to the future of actuated modeling. I will argue that movement is a central quality of tangible interfaces and suggest that further developments of Topobo could support the popularization of tangible interface design.

Appendix A: How might you build it in the future?

Appendix A briefly considers design approaches and technologies to support future actuated modeling systems.

Appendix B: Brochure.

Appendix B documents the Topobo brochure.

2 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.

I conceived of Topobo as an artist and toy designer who was thinking about educational toys.

2.1 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 is a building toy I helped develop that inspired Topobo. Zoob is also based on dynamic modeling.

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].

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

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:

> *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!*

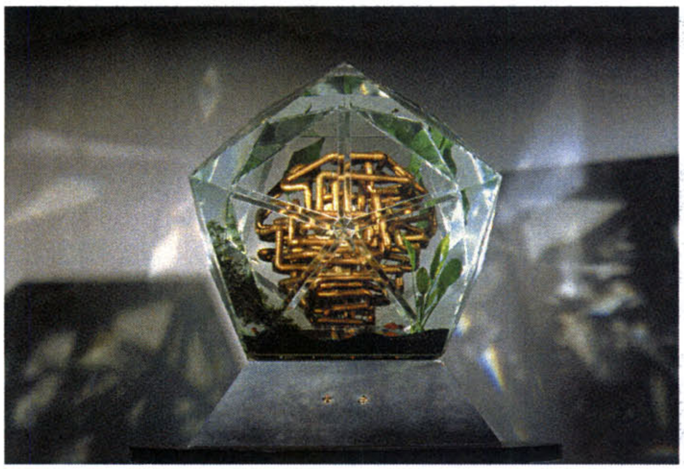
After graduating from Yale, I worked for several years helping design, produce and market the product, addressing a variety of issues from manufacturing to marketing to visual communication of the system's dynamics via the printed page.

2.2 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].

As an artist, I explored other interactive tools to understand dynamic systems.

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 move-

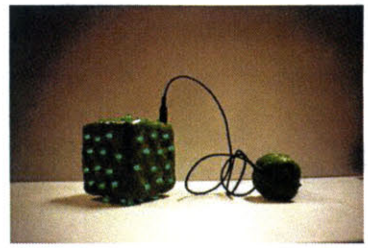


Biosphere is an earth metaphor. The owner has to manage the technology that supports the system, or it will die.

ment. 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 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 sculptural creations that responded to their environments with both local and global feedback pat-

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



terns, 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.

Walkers are based on passive dynamic robots. They walk downhill, powered by gravity. I was exploring how a simple system can lead to complex behavior.

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 oscil-

Walkers inspired Topobo activities in ambulatory locomotion.

lations 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 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.

2.3 The educational museum experience

Like my art practice, educational museums foster learning and discovery through experimentation with machines that have specially designed behaviors.

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.

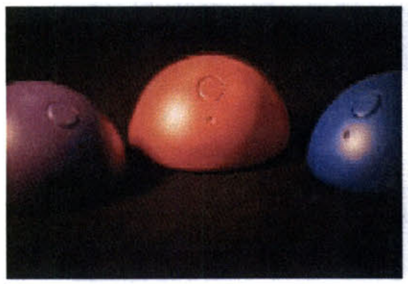
The Exploratorium pioneered what is becoming a popular idea – the interactive science installation – and is a rich source of many exhibits that fill today's children's science museums. Their work informed my motivation to explore concepts like dynamics through physical experimentation with machines and, 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.

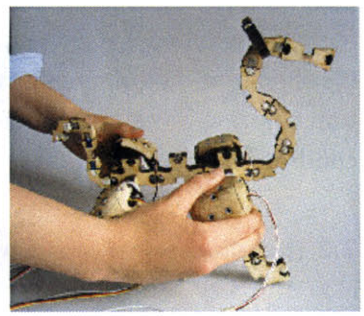
A general deficit in applications of computing technology, evidenced at the Exploratorium and in the limited use of computers in school classrooms everywhere, led me to the Tangible Media Group. Topobo is designed to target both the type of informal learning that takes place in the science museum and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

2.4 Tangible media

Curlybot explores programming as a gestural activity.

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





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

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.

The current system inherits elements from Zoob, curlybot and the Walkers.

2.5 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.

Actuated modeling could be used for a lot of things, but we're only looking at educational manipulatives in this thesis.

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.

Topobo could help kids learn about how robots and animals walk. This is a hard and fun behavior to understand.

3 Background and Related Work: Education, Tangible Media and Robotics

We are focusing on kids because they may be able to learn more than adults through physical interactivity.

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.

Tangible interfaces are especially relevant to kids.

We'll talk about robotics also.

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.

3.1 An Educational Basis for Tangible Media

Kinesthesia is one's knowledge of one's body in its surroundings.

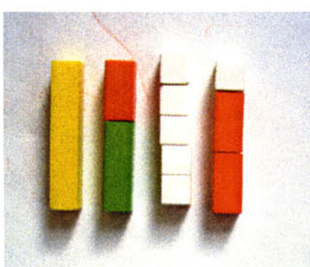
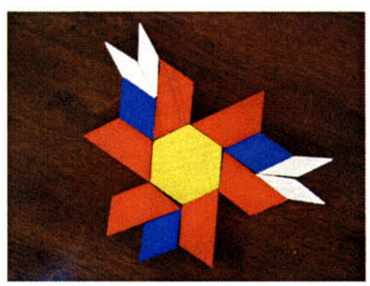
Kinesthesia and Learning
Touch is a central aspect of learning, and the study of kinesthesia focuses on the individual's movement and interaction with physical objects as a means of learning. Researchers in education, developmental psychological and cognitive sci-

ences 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].

According to Piaget, physical interaction with the world is especially important for young children.

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

Some people may be especially adept with their bodies and physical objects. These people are kinesthetically intelligent.

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

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

Educational manipulatives are tools that children use to learn ideas by constructing physical models of them.

Constructivist theorists study this type of learning, in which a child constructs his own knowledge.

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.

Educational manipulatives are related to Frederick Froebel's kindergarten gifts.

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].

Pestalozzi influenced Friedrich Froebel who created the first kindergarten by the year 1837. Froebel's kindergarten was filled

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

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 built on this trend and did much work with young children.

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].

Piaget continued some of this trend by providing an epistemological foundation for these educational ideas. He developed his famous "child's stages of knowledge development" by constructing a particular progression from the concrete to the abstract: children must 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.

Piaget developed theories that supported these educational practices.

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 worked with Piaget and studied how computers could be modeling materials to support children's education.

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.

This led to the LOGO programming language, and several variants like Starlogo.

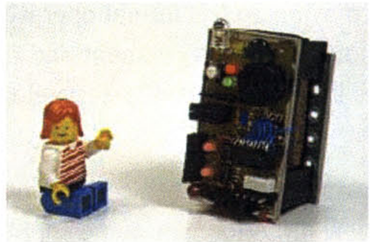
Digital Manipulatives

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 allow kids to understand certain systems concepts. As Resnick argues, "children, by playing and building with these new manipulatives, can gain a deeper understand-

Digital Manipulatives attempt to reintroduce physical construction to computer-based educational activites.

ing 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].

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



Digital Manipulatives and Tangible Interfaces

With most digital manipulatives, the programming uses a GUI.

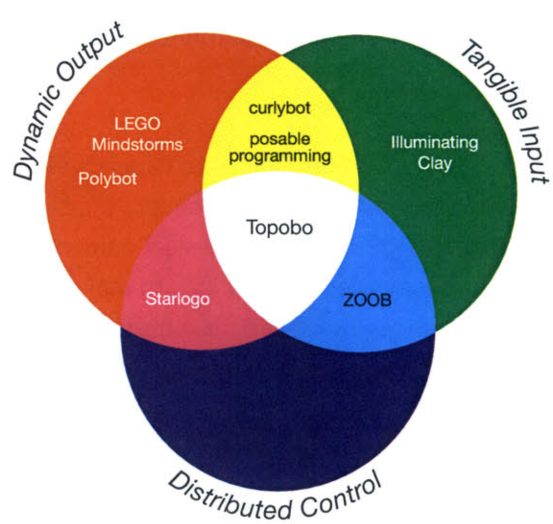
Curlybot explored how tangible interfaces could facilitate tactile programming, and get rid of the GUI.

Resnick's original examples of digital manipulatives prescribed 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 solution to this asymmetry. 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."

But curlybot lacks a constructive assembly activity.

Curlybot lacks the physical "construction" activities that are so common and valuable with educational manipulatives. One of Topobo's contributions to digital manipulatives is its integration of physical and program construction activities, presenting them both as tangible, constructive processes.

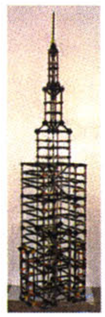
Topobo combines tangible programming and distributed computation to explore dynamic systems.



Building Toys

Digital Manipulatives 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 Logs® and ZOOB® in toys stores and in classrooms is evidence of our culture's appreciation for educational manipulatives.

Building toys are similar, in spirir, to educational manipulatives.



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

From one perspective, Topobo is a new member of the building toy heritage. 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.

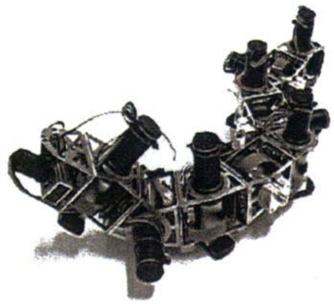
Through informal play and discovery, different building toys can teach children different ideas.

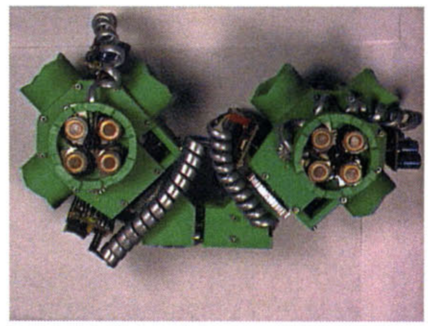
Topobo also facilitates explorations in topology in a different manner than Zoob. While ZOOB was intended to convey some aspects of the nonlinear nature of information behavior, it does not make information behavior manipulable. Topobo is designed to make certain systems concepts more clear with the Queens and Backpacks. 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).

Topobo's topology and ergonomics are based on Zoob. Topobo lacks the spatial flexibility of Zoob, but highlights the benaviors of moving structures and ideas related to information systems.

3.2 An application of robotics to actuated modeling: Related robotics research

This chapter now departs from an investigation of educational theories and tools to describe some of the robotics technology that allowed us to couple tangible programming and physical output. This might be thought of as "functionally related work" although it is no less important than the conceptual and educational foundation for Topobo. Robotics research was integral in the system's conceptualization and execution. It allowed Topobo to step off the drawing board into the physical world.





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

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 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 geom-

The main difference is intent. Modular robots are designed to perform tasks people can not do. Topobo is designed to be a medium for thinking, experimentation and discovery.

etry. 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.

The Polybot designers thought about toys.

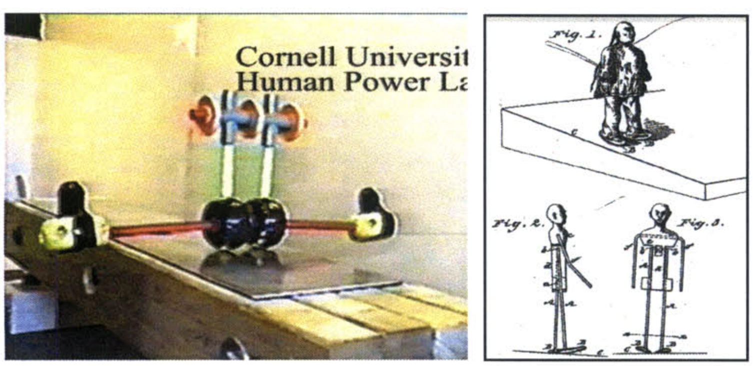
Programming by Demonstration

Like curlybot, 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."

Topobo is "programmed by demonstration," an old idea.

Passive Dynamic Robots

As I mentioned in the Motivation section, walking robots constructed with Topobo share physical simplicity and local-global



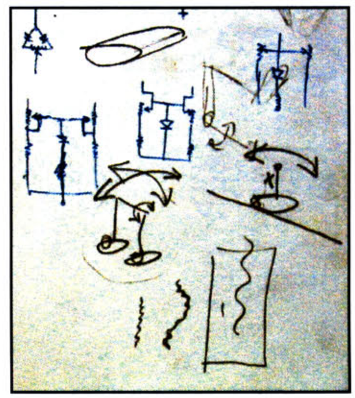
This tinkertoy passive dynamic robot is similar to the 19c walking toy. The difference is that psssive dynamic robots are only stable when they are moving.

Topobo creations encounter similar problems as passive dynamic robots, and may use similar dynamics to be successful walkers.

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.

4 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.



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

4.1 What is the Meaning of Motion?

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.

4.2 A Spatial Language

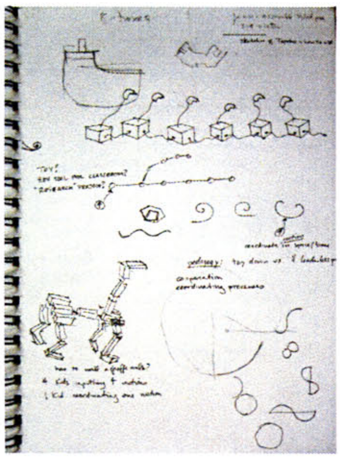
Topobo is a sort of "spatial language" with primitives and a grammar of usage.

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.

Coincident I/O

Coincident input and output was a means to test mesh modeling, and an end for educational toy design.

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.



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.

Preliminary Design Constraints

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. In addition to this criterion, 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.

4.3 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.

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.

Motors were a compromise because they do not mimic nature.

Physical Asymmetry

My early studies included dozens of sketches and models of modular, meshing systems. I used various materials including Zoob units, cut and glued LEGO bricks, cardboard and tape.

Mechanical studies revealed that an asymmetrical actuator is versatile.

Many of these studies 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.

4.4 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.

There are different ways to physically input motion. One way is to backdrive a motor and sense its movements.

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.

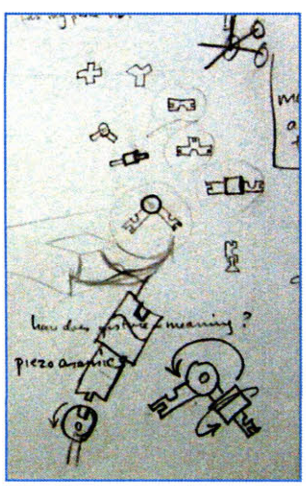
Another way is to backdrive a clutch and sense its movements. This can be more versatile.

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.

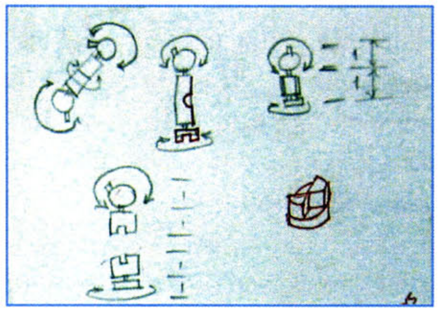
We backdrive a motor because it is easier.

4.5 A System Geometry

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



Actives and Passives
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



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

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.

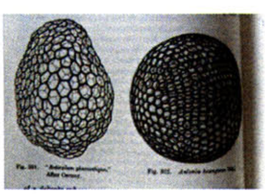
Actives were motorized, and Passives were not. The combination would allow various structures to be built and actuated.

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].

One finds different kinds of branching structures in nature.

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.



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.

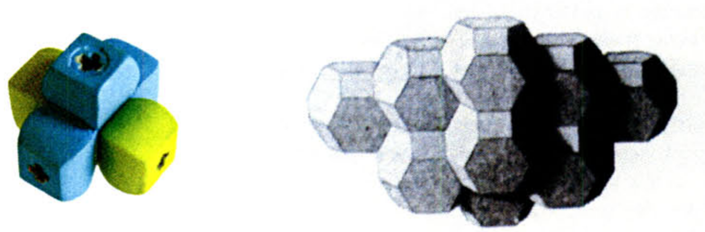
By contrast, smaller scale isomeric structures use spatial looping and weaving of structural members to achieve strength. For instance, the inside of a bird's hollow wing reveals how spatially distributed structural members can create a strong and flexible bone. This approach was better suited to my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

This is different than trees, which change scale to remain strong as they grow larger.

Rotary Motion: Crystals, Strength and Flexibility

The system geometry is designed to form spatial meshes and loops.

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



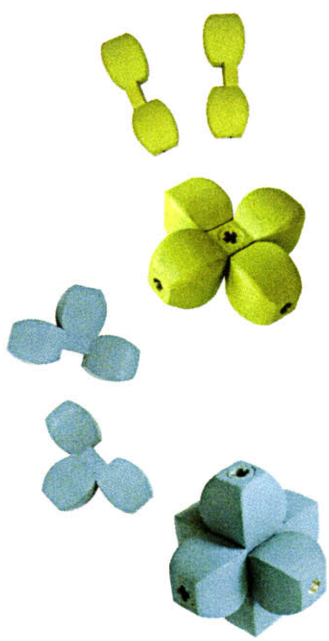
The system geometry is based on cubic and tetrahedral crystals. Here, Topobo is compared to rhombic dodecahedral crystals.

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.

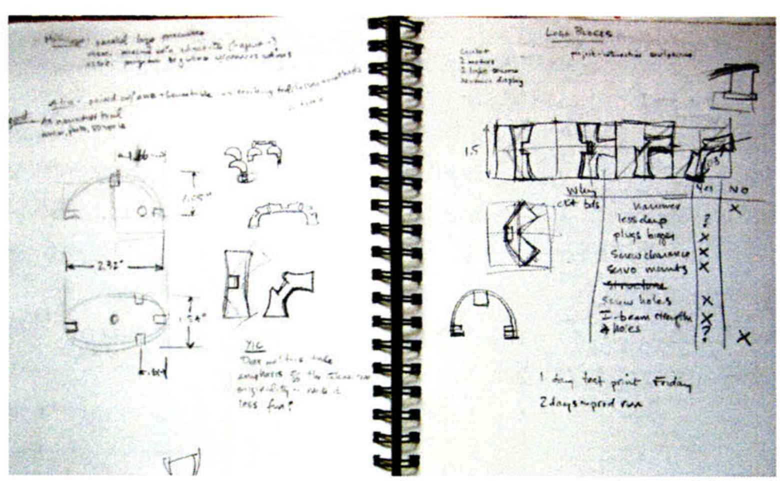
I chose these forms because they are easy to visualize and can be spatially manipulated.

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

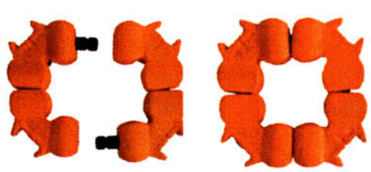


Notches allow passives to combine to form secondary parts. Eight notched passives can create 36 different shapes.



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 can also help people think about 3D shapes.



Four small 90°s form a square ring.

5 How you use it: Topobo System Design

We'll talk about how the UI design of the system evolved. The system components are designed to provide layers of complexity to support learning at multiple levels.

In this section I will describe how a number of design principles informed the creation of the current Topobo system. My discussion of the system's interaction design will include a qualitative description of the system's components, functionalities, and some envisioned uses. The envisioned uses are established from simple to complex. This structure is intended to suggest that Topobo can convey a range of ideas to the maturing child as she transitions through multiple stages of cognitive and emotional development.

5.1 Design Principles

Topobo adheres to seven design principles:

- *Be accessible, yet sophisticated*
- *Be meaningful even if the power is turned off*
- *Be expressive*
- *Support exploration of specific ideas*
- *Engage multiple senses*
- *Be scalable*
- *Be robust*

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.

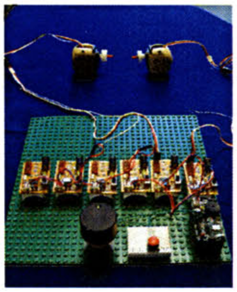
Be expressive – Design multifunction parts that give people latitude for their own personal explorations.

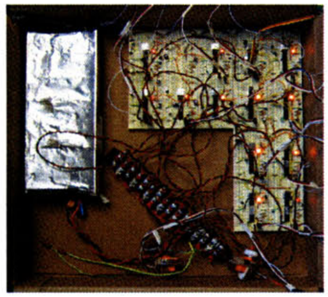
Support exploration of specific ideas – Make certain ideas salient so that people explore them through their activities.

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."



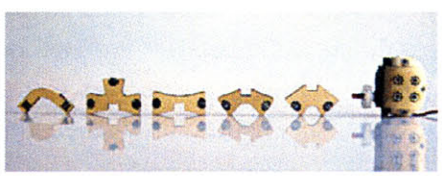


We built two prototypes to test the system design and concept. The cricket prototype tested the Queen and the breadboarded version tested a scalable, distributed electronic architecutre.

5.2 Proof of Concept

The early design studies led to the development of the current system geometry and a proof of concept using Cricket microcontrollers 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.



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

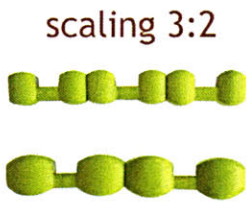
5.3 The Current System

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.

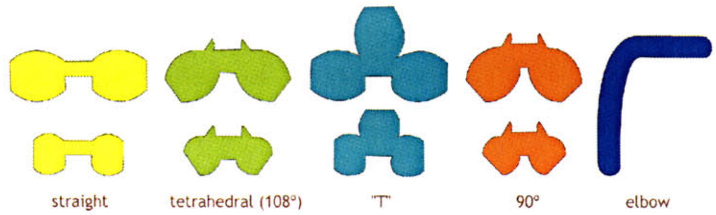
The current system is similar to this early prototype, but it has 10 parts now: 1 Active and 9 passives.

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.

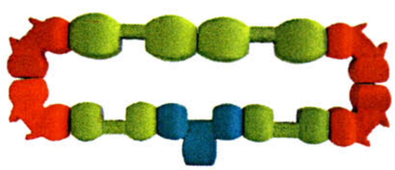


The passives come in two sizes with a 3:2 scale ratio that is based on the fibonacci ratio found in natural structures like plants and skeletons.



The "elbow" (offset 90°) comes in one size. The "straight," "T," 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.

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



The passives are also designed to be aesthetically consistent with the system's goals; they are intended to be individually beautiful and to assemble to create unified-looking creations. By basing the segmentation of the passives on Brancusi's *Endless Column*, the passives are intended to appear organic but regular, and their square cross section is intended to imply the possible 90° orientations of their connections.

The aesthetic design is based on Brancusi's Endless Column. Parts are color coded by shape.

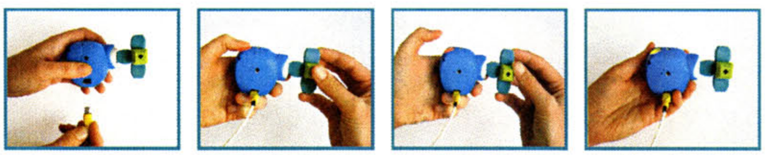
We color coded the parts to help people distinguish the different shapes and to encourage them to build playful, unified looking creations. A palette of tonally consistent cool colors (blues and greens) is offset by one warm accent color (orange-red) to give characters a visual "pop." We assigned the accent color (orange-red) to the 90° components because these parts often are most often used to terminate limbs on animals. People can build animals with unified looking bodies and accented feet, ears, or tails. The goal was to avoid "polka dot" creations while keeping the creations visually playful.

Colors are tonally consistent to allow people to build unified looking creations with a few accents.

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. Double-clicking the button will recall the last recorded motion.

programming an active



Programming an Active

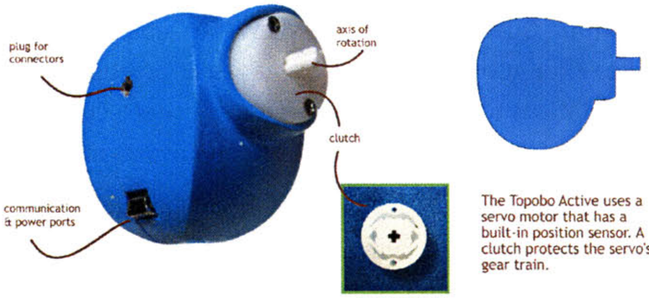
In a creation with many Actives, all of the Actives will record and playback at the same time. For example, if a child makes a circular ring of Actives, pressing a button on one of the Actives then sets all of the Actives in the structure to be in recording mode. The child may then move the circular 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 its 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.

In a creation with many Actives, all Actives will record and playback at the same time.

This can allow a child to build a complicated structure like a tank tread and program the whole structure with a single button press.

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 mechani-

The Active



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

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

The button is backlit with a colored light. The servo rotates 170°.

One button toggles through three states: record, plaback and stop.

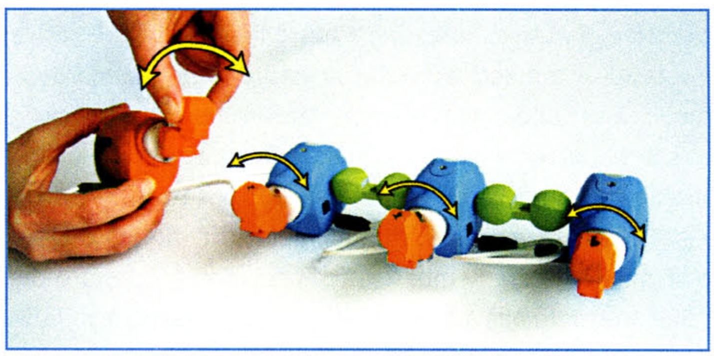
This simple interface has several benefits. Complex motions are easy to create, and they are natural looking because they are a reflection of one's bodily movements.

cal 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 and 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].

Queens: 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



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.

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.

A string of parts controlled by a Queen can curl into regular shapes like a circle or a spring-like helix (jpage 81).

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.

... or be used to explore spatial translation, as with the legs of an animal.

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.

... or be used as a remote control so that the child can debug motions during the recording mode.

Motion Editing and Control

If a child creates a motion in a creation and wants to change that motion, she must stop playback and then re-record the entire motion. The basic Topobo system did not originally allow the user to edit the playback motion. To provide a means to edit motions, we have introduced an editing feature to Active functionality and a series of "physical functions" called Backpacks.

Motions can be edited in two ways, by overwriting old motions or by using special parts called Backpacks.

The simplest way to edit playback motions is to rewrite a portion of the playback motion. Normally, if a child presses and releases a button during playback, the entire creation will stop moving. However, if the child presses and holds down an Active's button during playback, the Active will record new motions until the button is released. After the button is released, the Active will return to playback, displaying the resulting combination of the original recording and the re-recorded motion.

Segments of playback motions can be overwritten with new data by holding down the button during playback. The old motions are overwritten with new motions until the button is released.

This scheme introduces a complexity: normally, all Actives record and play back at the same time, resulting in all Actives displaying synchronized loops during playback. However, what happens if a child has recorded a motion that lasts longer than the original recording? In this case, the resulting playback motion could be longer than the playback motion of the other Actives in the creation. Loop lengths, and motions will no longer remain synchronized.

In structures with many Actives, this can raise some problems maintaining consistent loop lengths.

5.4 Physical Functions: A Backpack

Backpacks — physical parts with a button and a knob — can be snapped onto an Active to augment the phase, amplitude, and frequency of playback motions. The are described using familiar words, where phase is called "Time Delay," frequency is called "Faster/Slower" and amplitude is called "Bigger/Smaller." They can be used in three different ways called

Backpacks can be attached to an Active to change the phase, amplitude or frequency of its playback motions.

"local," "global," and "peer-to-peer," affecting the playback motions in strikingly different ways. The default mode is "local," while pushing the button on the Backpack will enable "global" mode. Attaching a Backpack to a Queen and pushing the button enables "peer-to-peer" mode.

Local Backpack

The local Backpacks allow local editing of playback motions. For example a "Faster/Slower Backpack" will make the playback motion of an Active faster or slower, depending on where the Backpack's knob is turned to. This might be used to make a dog's tail wag faster. Frequency is visualized with a pulsing LED on the Active whose frequency changes depending on the setting of the Backpack's knob. The backpack pulses its LED at a fixed rate that corresponds to no change in frequency.

A Backpack normally affects only the Active it is attached to.

Backpacks can be used to fine tune a creation's motion.

The Time Delay (phase) 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 phase Backpack to its tail and turn the knob on the Backpack to make the tail wag in sync with the head's shaking. Time Delay Backpacks can also be used to finely tune the relative motions in an animal's body, to make the animal walk more effectively. Time Delay is visually expressed with a simple LED pulse: at the beginning of each loop, the LED momentarily changes from green to orange. Normally, all Actives begin their loops at the same time and flash orange in sync. If a Time Delay Backpack is employed, the user will see the orange flash at a different time.

The LEDs on the Actives help display what the Backpacks are doing.

The Bigger/Smaller (amplitude) Backpack scales the recorded motion of an Active. Motions are scaled relative to the start position of the recording and, since Actives rotate only 170°, amplified motions can get "clipped" during playback.

Global Backpack

Backpacks are outfitted with a button that allows them to toggle through states. The default state is Local Backpack. Pressing the button will toggle to the secondary state, "Global Backpack." A Global Backpack identically affects all Actives in a creation with the Backpack's setting. For instance, the Frequency Backpack will cause all Actives in the creation to move faster or slower. In Global mode, all Actives will pulse their LEDs to reflect the changed speed of their playback motions, and turning the knob on the Backpack will cause the motions, and LED pulses to speed up or slow down. Similarly, Amplitude Backpack will scale the motions of all Actives in a creation. Since phase shift is a temporally relative function, the Time Delay Backpack does not have a Global function.

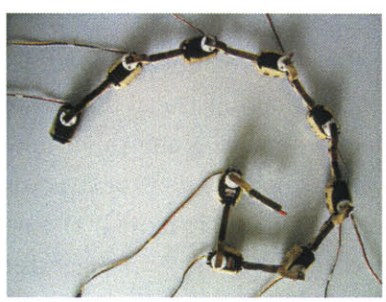
Pressing the button on a Backpack makes it affect every Active in a structure.

Peer-to-Peer Backpack

If a Backpack is attached to a Queen, by default it will affect the motion of the Queen, and thus synchronously affect all Actives that are controlled by the Queen.

However, if the user pushes the button on a Backpack that is attached to a Queen, it behaves as a Peer to Peer Backpack. The relative amplitude, frequency or phase shift of peer-to-peer functions is based upon the network topology, where the change increases every time the message is passed from one Active to the next. Topobo is designed with a peer-to-peer network, and the peer-to-peer Backpacks explore the behaviors of progressive mathematical series as expressed through motion. This enables dramatically different behaviors of structures of Actives, compared to the normal Queen.

If a Backpack is attached to a Queen, it's affect depends on how the Actives in the creation are wired together. The change becomes more pronounced on Actives that are farther from 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 Bigger/Smaller Backpack transforms a normal Queen into a 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.

Peer-to-peer Time Delay Backpack can create waves.

Time delay Backpack transforms a Queen into a Time Delay Queen. Actives connected to the Time Delay Queen mimic 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. Users can turn the Backpack's knob during playback to experiment with the relationship between loop length and time delay.

Finally, the Faster/Slower Backpack causes a Queen to speed up or slow 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.

Peer-to-peer Faster/Slower Backpack can exhibit resonance.

The Faster/Slower Queen introduces an interesting complexity: during record, Queens normally cause all Actives to behave the same way they will during playback. However, Actives cannot play "faster" during record mode since this would require knowing the future. If a child turns the Backpack to "faster," he will only see LEDs on neighboring Actives flash faster during record while motions remain synchronous. Motions will speed up only upon transition to playback mode.

Backpack as Physically Programmed "Conditional"

A number of eighth grade students who participated in our user study commented that they would like to have sensors with Topobo, or some means for their Topobo creations to respond to their environment. Indeed, feedback opens important conceptual and functional opportunities but is difficult to implement without a graphical or text based interface. Any feedback system implies a control structure, and few models for complex control structures exist that employ only physical programming by demonstration. 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, but the Backpacks afford a different approach to physically programmed conditional behaviors.

What about feedback, so Topobo creations have a sense of their environment?

Backpack feedback works like this. During normal use, if the knob on the Backpack is turned during playback, the motion changes in real time. For feedback, the knob is outfitted with a LEGO connector making it compatible with Topobo passives and is affixed with a torsion spring that forces it to return to a predefined position after it is turned. Thus, imagine the leg of a scorpion attached to the knob of a Bigger/Smaller Backpack that is modifying the motion of that leg. As the scorpion walks and shifts its weight onto that foot, changes in the scorpion's center of gravity place more weight on this foot, deflecting the Backpack and causing the foot to step more lightly. The scorpion might thus be designed to walk over irregular terrain and might be developed to step out of holes in the ground.

By connecting physical parts to the knob on a backpack, a child can use the backpack as a variable touch sensor. This can provide a simple form of feedback.

Other Conditional Behaviors

This example could be a model for future work to develop physical feedback techniques that can be used and understood without the need for a GUI. One simple example might be to use an infrared detector in place of the Backpack's knob. The detector might sense proximity of nearby objects to allow a creation to respond to its environment. If other Topobo creations wore Backpacks that emitted a certain IR frequency, creations could change their behavior based on "social" interactions. While modifications such as phase, amplitude and frequency may seem simple and limited, they can be useful. One exploratory creation could change the direction it walked based on a single phase shift.

New Backpacks could be designed to provide a Topobo creation with better senses.

5.5 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.

Topobo connectors break.

Wires

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. There is one advantage to using cables to connect Actives: when using the peer-to-peer Backpack, it is easy to visualize and reconfigure the network topology. However, in an ideal design Topobo would not have cables.

Wireless would be better.

Actives

The Actives should be smaller and more finger-like in form. Better actuators like polymer muscles could facilitate this.

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. A major incremental improvement to the system would be to design Actives with more oblong proportions so that they are more similar in shape to the Passives.

This will be important for meshes that have many actuators.

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.

2 or 3 DOF actuators would also help with meshes.

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.

Saving Programs

What about recalling saved motions? The last program can be replayed with a double click, but a more flexible approach would be useful.

Currently, the last stored program can be recalled by double clicking a button. However, saving and replaying older programs, either using Backpacks or another interface technique, would be beneficial. Anyone who has enjoyed building models understands that people like to keep successful creations, and saving motions would be a necessary element of a Topobo creation to save and later recall.

Control Structure

Some older kids wanted a more sophisticated control structure.

While the Backpacks present one example of a feedback loop, in general Topobo lacks a sophisticated control structure to model "intelligent" behaviors. This was a conscious design decision made to allow people to focus on mastering the basics of processes like kinematic locomotion, but a more scalable control structure would be an interesting addition for more experienced users.

6 How we built it: Topobo Engineering

This section reviews the technical implementation of the current system. A qualitative overview of the engineering decisions will give a more complete sense of the technical limitations in designing the system.

6.1 Structural Parts

"Structural parts" include the Passives, Active housings, and Backpack housings. 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

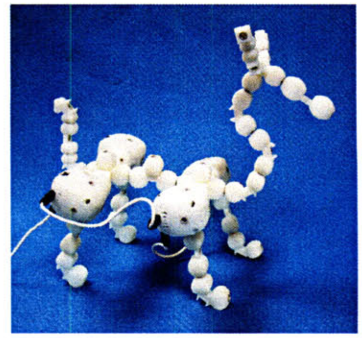
The passive geometry is based on flat shapes, and I laser cut our original prototypes from 3/8" bass wood and glued LEGO connectors in to their ends. Bass wood is strong, light, affordable and aesthetically pleasing. However, it is soft and the notches compressed and wore out due to repeated connections.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

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

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

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.



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.

Final Molded Parts

Our final parts are molded plastic, based on 3-D printed models. The passives are injection molded in ABS using a bench top press and epoxy/aluminum molds fabricated from 3-D wax prints. Passives are 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 are 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 parts are custom molded. Motor housings are cast in urethane and passives are injection molded in ABS

6.2 Mechanical and Electromechanical Engineering of Actives

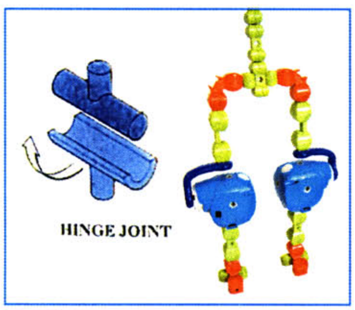
Electronics design is loosely based on the Polybot modular robots.

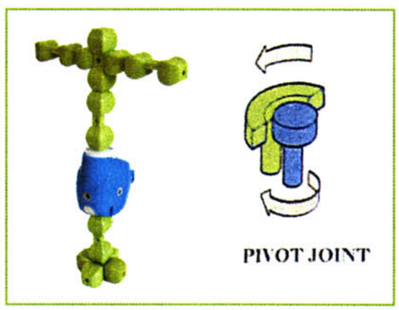
The engineering of the Actives is based loosely on modular robotics technology developed by Yim et al. at PARC [Yim00]. While I did not directly adopt any of Yim's designs, I considered his approaches during the design of the Actives and eventually adopted several similar approaches for scaling power. 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.

Obvious applications of servos are pivot and hinge (elbow) joints. We use a pivot joint, but adding the Elbow passive allows it to act like a hinge.





Actives naturally provide a pivot joint. The Elbow connectors allow Actives to behave like a hinge joint.

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, 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.

We backdrive motors during record mode. This not work with most gear motors.

Since engineering data is not available for the back driving torque of a motor and no manufacturer would advise our search, we empirically found a HiTec hobby servo that met our needs. This servo has a low enough gear reduction that we can back-drive easily (60:1), the metal gears resist breaking, and the motor is strong enough for many of our applications. A

We use hobby servos: medium quality, but good enough.

thorough engineering test of back drivability of motors, or a re-engineering of the sensing approach for Topobo (such as using a clutch and external sensor) could facilitate the use of higher quality coreless motors in future designs.

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

Despite the quality of our gears, they are still fragile. Accidentally dropping a Topobo creation or recording a violently fast motion can strip the gears, so I created an indexing clutch to attach to the output shaft of the servo. This clutch, made of laser cut 1/8" Delrin® sheet, was empirically designed to have enough tension not to slip during normal use, but to clutch before gears could break. It uses an arrangement of 4 spring arms and an indented ring so that it "clicks" through 45° increments.

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.

Flexible connectors require lower manufacturing tolerances and limit breakage.

They are also easier to use.

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.

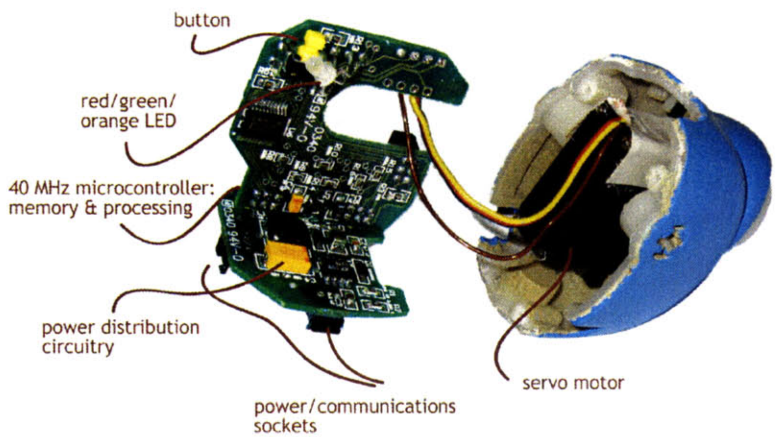
6.3 Electrical Engineering

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

Power Distribution

Upon suggestion from former professor Paul Horowitz, we use 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.

All Actives are identical and autonomous. They only need power to function.



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

Processing

A 40 MHz PIC 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 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 57000 BPS. Mini USB-b connectors and specialized Maxim line drivers 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

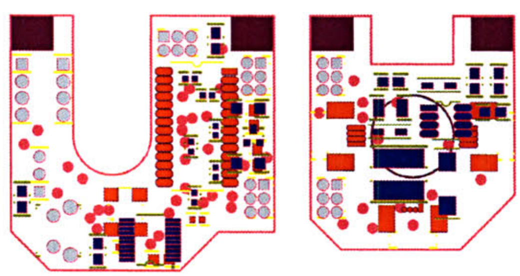
An engineering goal was to create a scalable system that could accommodate up to 100 Actives at once. So far, we have successfully tested the system with 30 Actives. (Our bottle neck is manufacturing — we don't have more parts.) The high voltage power bus facilitates scalability by limiting current requirements 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).

Topobo is designed to be scalable up to 100 parts.

Peer to peer communications helps with scalability.

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 future systems need to increase scalability, one approach may be to use a higher voltage (24V - 48V) power bus.

Large structures work better with multiple power points.



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

6.4 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 Josh's work, and I am grateful that he lent his expertise and experience with embedded networks to the project.

A microcontroller mostly handles motor control and serial communications.

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.

These are both time sensitive tasks, but motors have priority.

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.

PID circuitry on the servos requires a low frequency control signal.

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

A calibration scheme measures the mechanical and electrical characteristics of each servo so that they all appear to behave identically.

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.

Calibration allows the Queens to work well and limits network traffic.

Record and Playback

During normal local recording, an Active will read its ADC at about 36 Hz and write values to a 1 Kb data array. When playback is initiated (or when the array is full) the data is copied to nonvolatile flash memory and is then passed as an argument to the playback function, which simply uses the calibration results and recorded position data to recreate a series of servo pulses at 36 Hz. This gives us a maximum of about 34 seconds of recording time. One improvement to this scheme is to record at half the rate and linearly interpolate values during playback using a simple average. This approach has been proven to work and gives the user over a minute of recording time. By writing to flash memory, programs can be recalled if an Active is temporarily unplugged.

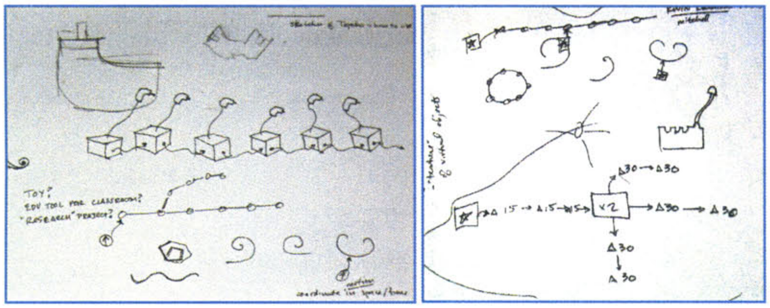
We have 34 seconds of recording time.

Communications

Peer-to-peer communications are handled exclusively in software, giving us 4 channels of serial communications with data rates at around 57000 bits per second. The networking protocol uses two wires for communication, generally used as "clock" and "data" that are by default pulled to Vcc with 220K pull-up resistors. In the program's main loop, an Active will routinely poll for messages on all channels. If an Active wants to send a message, it will pull the clock line low and wait for the data line to be pulled low by the neighbor. If no neighbor is present, the channel will time out and the Active will check the next channel. If the handshaking is returned (the receiver pulls the data line low), the sender will begin clocking data at a predefined rate. Bytes are transmitted with a parity bit and arrive in 1-5 byte packets handled by a software data buffer. An error

A custom software serial communication protocol uses a clock and data line.

in parity will cause the receiver to request the bad bytes be resent. An Active can only send or receive on one channel at one time, as we have no hardware buffers. Messages arrive with a message type (denoting a type of state change, for instance) an argument, and a message ID. Message ID's are used to prevent propagation of a message in a network loop: if the same message type and ID is received twice, the message is ignored .



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

Message Types

The most common messages are state changes telling an Active to record, playback, or stop, and synchronization signals. Other message types are Backpack messages and Queen messages that include a position signal. When a message is received correctly, it is immediately sent to all communications channels except the channel the message was received on. After a message is propagated, it is processed.

Most messages are short

Synchronization

One problem with an asynchronous system such as Topobo is that sometimes Actives record messages of slightly different lengths, causing them to eventually get out of phase during their looping playback. To keep Actives synchronized during playback, they all communicate their loop start to their neighbors. If an Active receives a loop start signal and its own array pointer is near its loop start, the Active's array pointer will jump to its loop start. This is similar to the way people clap in unison by listening to the cadence of their neighbors and subtly correcting their own cadence.

During playback, Actives stay synchronized by listening to the cadence of their neighbors.

6.5 Backpack Engineering

The Backpacks are made of a single PCB with two power/ communications ports, a button and a potentiometer housed in a plastic case. Their engineering is similar to the Actives with a few notable exceptions. Backpacks have no servo, and thus require much less power. This allowed us to power the Backpacks with a linear regulator instead of the more complex switching regulator used on the Actives. We also do not use a buffered line driver, and instead rely on the mechanical design of the USB connectors to protect I/O lines during hot swapping. Backpacks use the same PIC as the Actives and implement the standard Topobo communications protocol, allowing them to process and rout messages through their two I/O ports. Unlike the Actives, one of the ports is a "male" plug, allowing the Backpack to connect directly to an Active without the need for an additional cable.

A Backpack is built similarly to an Active.

Backpack Communications

When a Backpack is attached to an Active, it will announce to that Active that it is present and pass its Backpack identity and pot value to that Active. It will then send an occasional (5 Hz) "I'm still here" message to the Active to denote that it is still attached. Any changes is the ADC, or any received messages will be passed to the Active at normal data rates. If an Active ceases to hear the Backpack's "I'm still here" message, the Active's internal "Backpack timer" will time out, and the Active assumes that the Backpack is no longer present.

A Backpack can be attached or removed at any time and the Active will know.

State Changes

Some Backpacks have a button that allows them to transition from a "local" to a "global" Backpack. The button press simply changes the internal state of the Backpack, which then sends state and modifiers to the Active. All of the Actives are preprogrammed to handle all variations of Backpack message types.

7 Studies 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. These studies introduced various versions of Topobo prototypes. Kindergarten and second grade students played with wooden parts, breadboard electronics and a cricket microcontroller version of Topobo. Eighth graders used wood and plastic prototypes with embedded electronics that are similar to the current hardware. The Backpacks were built too late to include in these studies, and I hope to assess their effectiveness at a later date.

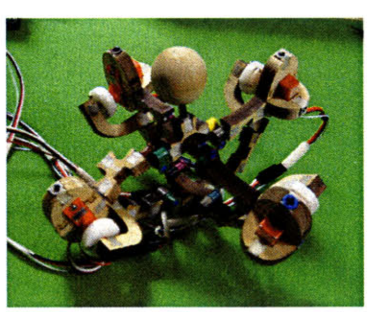
We evaluated various versions of the system with children ages 5-13. We have not evaluated the Backpacks yet.

I conducted the studies with younger students with the help and support of Cristobal Garcia, who made insightful contributions to the children's learning processes. Amanda Parkes and I worked with eighth graders, and she contributed insightful analysis of these children's different learning styles.

7.1 Kindergarten and Second Grade Studies

In order to evaluate the technical features, design principles and educational goals of the early Topobo prototype, Cristobal Garcia and I did a preliminary field observation at an elementary school in Boston. We spent two hours each in a second grade class and a kindergarten class playing with Topobo. The envi-

Kindergarteners and second graders played with the wooden version that used crickets and breadboarded electronics.



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.

Younger children at the school were encouraged to play with manipulatives rather than computers.

ronments of the classrooms, especially the kindergarten, were Montessori-inspired. The classrooms featured simulations of sea world, sofas, different manipulative toys, a kitchen where kids usually cook, paintings, aquariums, and a little playground filled with sand and dinosaurs, among others things. Only one computer was present in each classroom and it was strictly for teacher use.

We worked intimately with small groups of children.

Each of us 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. The rest of the classes were doing other activities with the teachers while we were introducing Topobo.

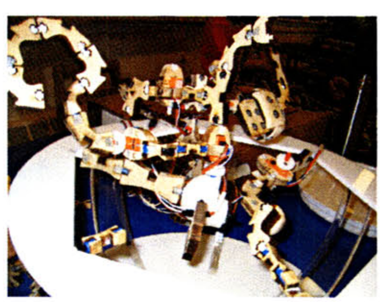
Topobo was viewed as a manipulative rather than as a computer system.

** Note: While older school children who are more adept with abstract manipulation routinely use a computer lab, these classrooms did not have student computers because the teachers want to continue to emphasize manipulatives with children this age. Children can, however, use the computer lab by request. Topobo may have been welcome in this environment because its interface is more similar to manipulatives than to a PC/GUI. The teachers seemed not to view Topobo as "computers."*

7.2 A Case Study with Dave

Dave is in second grade and, according to his teachers, he is usually very impatient with things and other kids in his environment. He was in one of our groups of field observation. We introduced Topobo to his group by comparing a walking Topobo creation to ourselves walking. When Dave came to one of the tables where we were sitting and manipulating Topobo, he immediately got engaged with Topobo. 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.

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



Dave was working to create his own walking animal. Drawing on the earlier examples of walking animals that we showed him, Dave 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 Topobo Queens 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? Dave began manipulating Topobo in new ways to try to produce movement, feedback, global-local interaction and walking. While he was not successful in creating a walking animal, the process of physically "debugging" his creation may have given Dave new insights to kinematic systems.

Dave's breakdown situation suggested he was very engaged with the system. Such sitations can lead to new conceptualizations.

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. While it is too early to know what Dave may have learned from his interactions with Topobo, his deep engagement with the system suggests that Topobo was appropriate for his developmental level. 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 [Mon12].

Our support was helpful for Dave, and teacher guided support may be important for certain types of Topobo activities.

Dave's engagement with his work suggests that Topobo was appropriate for his developmental level.

7.3 Studies with Early Adolescents

Later studies 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.

Eighth graders who routinely work with LEGO Robolab tested Topobo.

Our first evaluation session introduced the system. Using a preliminary worksheet, students described different types of motion related to their bodies based on 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.

Kids started with free exploration of the system.

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

We observed two styles of work. "Active iteration" involved regularly testing small ideas and incrementally building on successful pieces.

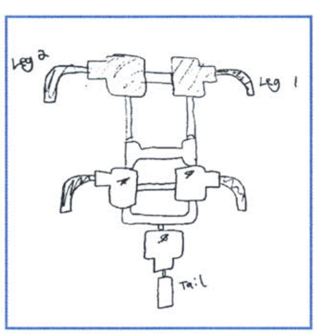
"Compartmentalized designers" would build a complex model before programming a motion. They were usually unsuccessful with walking.

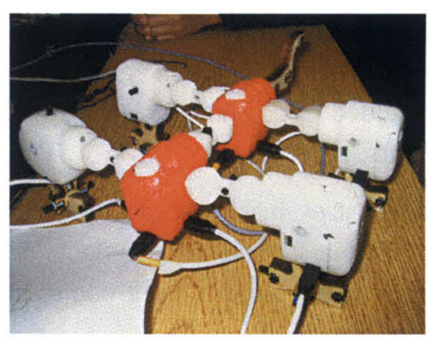
The second evaluation session a week later focused on a task to construct a "walking creature." Students first planned and drew their creature and then tried to build it and make it walk. We observed two different methods of design. The first method involved "active iteration" during the creative process. 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 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.

Successful creations were very different than kids' original designs.

Compartmentalizers would adopt active iteration.

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.





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

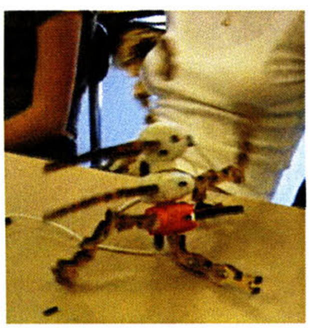
These findings show that an interface design should support active iteration by allowing users to test the behaviors of interdependent processes as they develop designs. Users often need to test many ideas to incrementally develop a successful design. Students who initially compartmentalized the design 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.

Interfaces should support active iteration.

Evaluation of Queen functionality

Our evaluation of the Queen was 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,

Queens were difficult for some kids. They seem to require a different mental model than normal Actives.



Eighth graders test their walking creations.

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.

A software bug skewed these findings.

This study showed us that the Queen needed further engineering and design refinement. A minor bug can be an obstacle to

Learning to use Queens may require more scaffolding.

learning if it causes greatly unexpected output. It also showed that in future interactions, Queens may require more scaffolding than direct manipulation with Topobo.

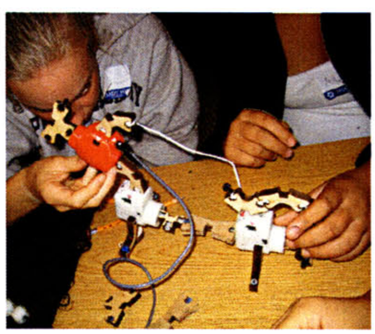
7.4 Animals and Machines

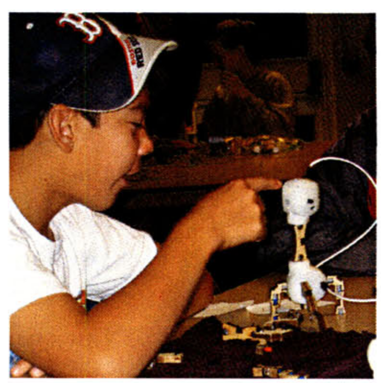
Kindergartners, second graders and eighth graders all related to Topobo models with their "familiar knowledge" about animals and machines. Metaphoric allusions to machines (robotics) and especially to animals ("the elephant," "the ant," "the scorpion," "the horse," "the no-walking man") were descriptive 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.

Children compared Topobo motions to the familiar movements of their bodies.

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.

Play with Topobo appeared to be a body syntonic activity,





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

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

Could Topobo become a "transitional object for some children?"

moving into the world of the abstract [Pap80]. We hope that further research will help us evaluate this hypothesis.

7.5 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.

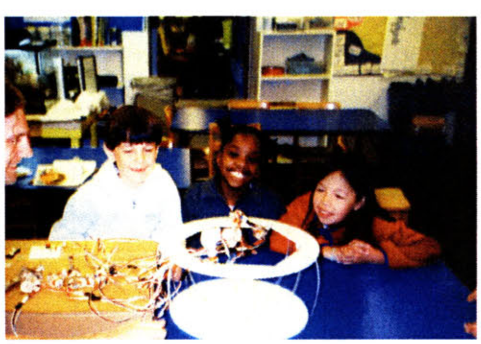
All kids had initial experiences of discovery.

Kindergartners generally programmed only one Active at a time, although many collaborated on group sculptures. 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 7-11 who are in the process of developing abstract mental models, but still experiment very freely, may benefit most from Topobo.

Children ages 7-11 who are in the process of developing abstract mental models, but still experiment very freely, may benefit most from Topobo.

We tested Topobo with a wide age range to evaluate its capacity to be both accessible and complex to children at widely



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

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.

Students at multiple developmental levels effectively collaborate ... Topobo may support learning during cognitive transitions.

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, and are dependent on adult or peer 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.

7.6 Design Changes prompted by the studies

The work with the children revealed several shortcomings of the design.

Kids told us how to make the system better, either explicitly or implicitly through their difficulties with the system.

- Children had difficulty distinguishing the rotary connector on the Actives. All connectors appeared like they should move.
- Some kids wanted sensors or feedback.
- Some kids wanted to experiment with recording the motions of individual actives one at a time, and then playing them all back together. We may implement this function in the future.
- Queens were confusing, perhaps because the absolute position of the Actives' output shafts could not be determined.
- The rotational limits of the Actives were not clear
- Distinguishing tetras and 90° parts was difficult.
- Friction was a major issue for locomotion, and many children worked on cloth because parts were too slippery on the table top surface.

We have tried to address most of these problems with design changes. The output shafts of the Actives has been redesigned so that it is visually distinct from the other connectors. It is a different color. Also, the mechanical stops on the output shaft are now exposed and integrated into the housing design so that it is (a) obvious what the mechanical limits of the Active are and (b) the absolute position of the output shaft can be quickly seen. Our most current parts are color coded to help children distinguish the different shaped parts. We have not addressed issues of friction, although some people have suggested rubber "feet" and other such solutions that may be interesting additions to the system. We hope that future user studies can help determine if the children will use the Backpacks as "sensors" for feedback and control.

We tried to address most of the major problems with the current system.

8 Educational Implications

Topobo introduces new ideas that aren't in a curriculum. Is this because they're not important, or because there has been no tool to teach them?

We think it's the latter. The ideas are basic, fun and exciting for kids.

Schools might pay the high price for Topbo if it can help them teach many different ideas to different aged kids.

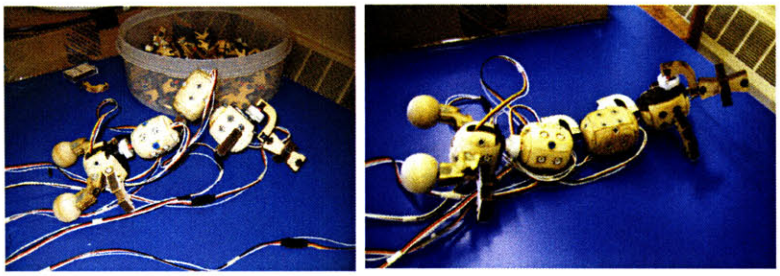
Our early studies, in which children have worked to create walking robots, have addressed Topobo's applicability as a digital manipulative to help children learn about the roles of balance, leverage and gravity in kinematic systems. While children learn about these ideas through their own body-knowledge, they do not have tools to experiment with, and thus abstract them. This absence of tools and related educational activities raised a "chicken and egg" question stated by some elementary educators [Bra03; Vor03] with whom we worked: are such concepts not taught because the knowledge is not valued, or is it because teachers have no suitable tools to teach them?

Educators' enthusiasm with the toy encouraged us that kinematics concepts are relevant to their students' curriculums, despite the fact that the concepts are not currently taught to these younger children. This section of this thesis presents Topobo as an open-ended system for exploring a range of "advanced" ideas related to physics and math education.

Topobo is designed to model certain types of natural structures and processes, and the following activities and applications are designed to reveal a number of patterns and processes found in nature. Some of this knowledge is intended to be discovered through informal play experiences, while other ideas are intended to be scaffolded, or introduced to students with the developmentally appropriate support of a parent or teacher.

Although Topobo is expensive, it may be useful to educators if they can repeatedly apply it to many different students and activities. Our preliminary observations that children ages 7 and 13 found Topobo to be complex, challenging *and* accessible encourages us that the system may be widely applicable to different age ranges, and possibly to different educational domains. This chapter will thus conclude with a description of how varying levels of complexity are designed into the system, so that Topobo can make a body of knowledge both exciting and challenging to a growing, maturing child.

8.1 Learning with Manipulatives



Play with a time delay Queen lead to this undulating caterpillar.

The activities and ideas outlined in this chapter are inspired by Froebel's kindergarten gifts and modern trends in educational manipulatives [Bro97; Res98]. As discussed in the Background chapter, these activities could find a home in a constructivist classroom where students are encouraged to construct their own knowledge through manipulation of familiar, but specially designed physical objects. There is an element of kinesthetic learning that occurs through play with Topobo [Gar82]. In kinesthetic activities, the body's reflexes are physically engaged in the learning process, and a sort of intellectual "dialogue" can emerge 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?"

Play with Topobo is a kinesthetic activity that focuses on the question "what can be learned by building with motion?"

8.2 Domains of Knowledge

Through our preliminary studies, we found that Topobo can help students ages 5-13 to learn about several educational concepts. Although the Backpacks have not yet been evaluated, this subsection will include examples of ideas that children may explore with them in the future.

Topobo helped kids experiment with a number of ideas related to how leverage, balance and gravity affect kinematic systems.

Balance: Experimentation with balance can lead students to discover relationships between a creation's center of mass and its stability. Children can begin to notice these relationships when their creations move and unexpectedly fall over. Learning how to control falling can lead to an understanding of familiar dynamic processes such as walking.

Balance: things fall over

Center of Mass/Gravity: as creations move, their balance changes.

Center of Mass/Center of Gravity: When objects move, their center of gravity changes. Because Topobo creations move on their own, children can transfer their knowledge about their bodies' center of gravity to develop an understanding for balancing Topobo creations. For example, 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.

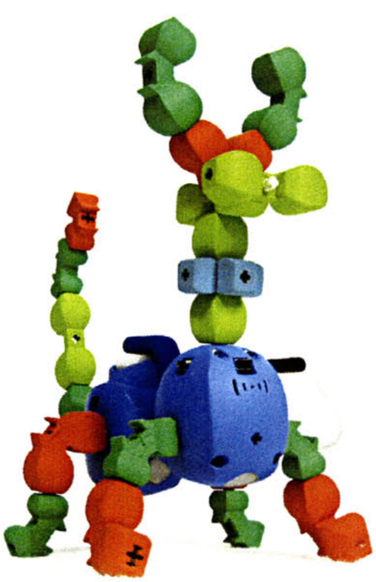
Coordination: through collaboration or with Queens.

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 provides children with a tangible means to understand how coordinated movements can change a whole system.

Relative Motion: movements change among different physical constraints.

Relative motion: Movements change among different physical constraints. For instance, a Topobo creation often seems to move differently when one is programming it and it is responding to one's hands, and when it is in playback and is responding to gravity and friction. In one's hands Topobo can also draw one's attention to ideas of relative motion. For instance, 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. When he tried holding a shaking end, and was amazed to see his entire creation wave wildly back and forth. While he did not appear to understand the reason for the different behaviors, the contrasting motions drew his attention to the idea that movements in a connected system are relative to one's frame of reference.

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



Movement with Multiple Degrees of Freedom: A Topobo Active provides motion in one degree of freedom (DOF). 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. With continued access to Topobo, I predict that the students would begin to internalize the process and understand how to organize motion in multiple degrees of freedom.

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

Local/Global interactions: With Queens and Backpacks.

Micro/Macro: Topobo Queens synchronize motions in time, and require that the motions be identical. The interface invites a child to experiment with the many ways a single gestural twist can lead to a recognizable global motion. If a child creates a "spring" out of a helix of parts, a very small local rotational motion, spread throughout the length of the helix, would cause it to extend dramatically. Such an exercise might give students an opportunity to wonder if a real spring is made of very small parts, and if it is, what might happen to those parts when a real spring extends.

Micro/Macro: structures with rule-based motions can become models for other systems.

Accretion: By incrementally changing the amplitude of one's gesture with a Queen that is wearing a bigger-smaller Backpack, concepts of accretion and decay can be demonstrated. For instance, if a long string of parts is built and programmed with a Queen alone, it will curl into a circle. However, with the

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.

bigger-smaller Backpack, the original input motion is scaled larger at each sequential Active in the line; the string will curl into a spiral. Such an exercise can be used to help students to understand the equiangular spiral, as seen in snail shells, nautilus shells and sunflowers. More broadly, Queens with Backpacks can help people understand the idea of math functions.

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

Topobo has 12-fold symmetry and can support free form or patterned building.

Phase shift: A Queen with a time-delay Backpack can demonstrate concepts related to phase shift. For instance, real waves can be described as a series of identical oscillations that are offset in time. If a string of Topobo Actives is programmed to do a simple oscillation with a propagation delay, the string will playback with a motion like an oscillating wave. As well as using this model to think about the local-global behavior of real waves, a child can use these undulating movements to experiment with familiar terrestrial motions like snake or caterpillar locomotion.

8.3 Complexity

One goal in the design of any educational tool is to have a "low floor" (easy to learn) and a "high ceiling" (lots of room for cognitive growth). It is too early to know if Topobo has both a low floor and a high ceiling, but our preliminary results that both second graders and eighth graders thought Topobo was designed for their age range were encouraging to us. Evidence that their teachers were enthusiastic about the activities also encouraged future applications for Topobo across this wide age range.

While I have designed individual Topobo components to seamlessly function with the entire system, I have often thought about different categories of components as "layers" of complexity that range from straightforward 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 hope to provide a framework to describe activities and different ways that Topobo might be used.

Layer 1: Basic Building (without power)
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.

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

Repeated play leads to repeated discoveries.

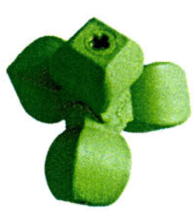
Layer 2: Building with Motion

Normal Topobo programming is a simple and accessible way for children of many ages to experiment with kinematics. Compared to the Queen, the younger children found direct manipu-

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.

lation to be more "magical" and exciting than the Queen. Some kindergartners showed facility using the one-button interface and were able to focus on motion when using it. 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.

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.



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

As children build larger creations with more Actives, coordination also becomes a limiting factor. Since children can usually only grasp 2 Actives at a time, in creations with more than two Actives children must either cooperate with other kids or create interconnected geometries that are self-constraining, like rings or loops.

Layer 3: Using Backpacks

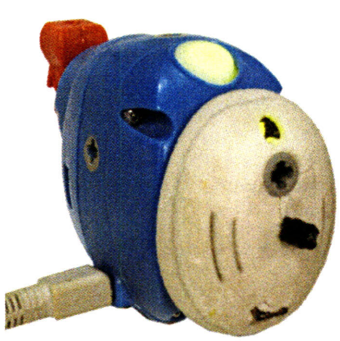
Although the Backpacks have not been evaluated, they were designed to provide a layer of complexity for children who can begin to understand the roles of local motions in a globally moving system. The Backpacks are physical objects that embody a change in an Active's function, such as frequency (faster-slower), amplitude (bigger-smaller) or phase shift (time-delay). When a Backpack is removed, the change in function disappears. In effect, Backpacks physically instantiate computational functions.

Backpacks can lead to surprising behaviors.

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. Bigger-smaller Backpack can be confusing because as a motion gets larger, it gets "clipped" by the mechanical range of 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.

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



Global vs. Local Backpack may also be confusing for younger children, as there is a level of invisible information in the Backpack's internal state change. While the creative use of LEDs on Actives may make these state changes clear, 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.

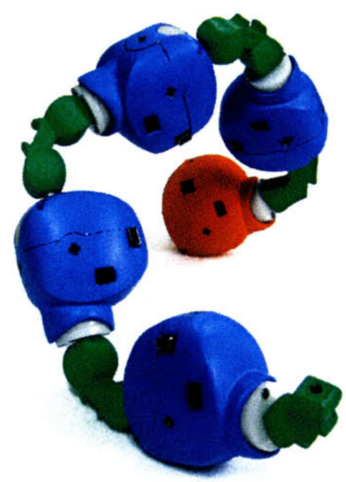
Layer 4: The Queen

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.

Queens demand that children think about the absolute positions of the Actives' output shafts. This is more abstract than normal record and playback, during which one thinks about relative positions of Actives.

Queens can also be used to explore spatial patterns, such as the similarities and differences between a flat pattern like a circle, and a 3-D pattern like a helix. By coupling patterned motion (with the Queen) and patterned form (with the Topobo system geometry), children can begin to discover some common forms found in nature.

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 5: 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.

Layer 6: Peer-to-Peer Backpacks

When a Backpack is attached to a Queen, it behaves as a peer-to-peer 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 peer-to-peer Backpacks are inspired by natural systems like waves and nautilus shells that change or grow as a result of local rules and interactions. Peer-to-peer 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.

A Queen can be used to create a base motion, and Backpacks can be used to edit the motions of individual Actives.

Understanding the peer-to-peer 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 (which may be different than the physical topology of the creation). Therefore, I believe them to be the most conceptually advanced component of Topobo.

Peer to peer Backpacks are abstract and complex. They may be used to think about certain calculus problems like the wave equation, or series expansions like the fibonacci series.

Students can use peer-to-peer 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 peer-to-peer 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.

Peer to peer Backpacks may be used to think about certain calculus problems like the wave equation, or series expansions like the fibonacci series.

In less regular structures, peer-to-peer 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 peer-to-peer 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.

8.4 Summary

Topobo is designed to reveal to students a range of patterns and processes found in nature. By layering different functionalities within the system, Topobo is intended to be meaningful to people of varying ages. Through empirical discovery and guided exploration, students ages 5 and older can use Topobo to begin to experiment with kinematics concepts like balance, leverage and center of mass. As children grow older and begin to master abstract conceptualizations, they can use Topobo to experiment with mathematical ideas related to series and distributed algorithms like the wave equation. Although children cannot use Topobo as a tool to work with the advanced symbolic math that describes such systems (since Topobo parts do not directly represent numbers and operations), it can help them develop a qualitative foundation to understand how such systems behave.

A system must be easy enough to learn that it is rewarding. A sufficiently complex system does not get boring very quickly.

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. This design is intended to allow children to use Topobo to help them transition from concrete manipulation to mental manipulation of abstract ideas.

Complexity is layered into the system in two ways: each aspect of the system can be used in many ways, and additional layers of functionality add complexity.

9 Future Work with Topobo as a digital manipulative

How can Topobo become a better digital manipulative? First, it needs to get into the hands of educators.

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.

I have argued for a return to tangibility in digital manipulative design. However, a GUI may benefit Topobo in certain ways that we will explore.

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.

9.1 A Higher Ceiling? The limits of complexity with physical programming

Topobo does not have a complex control structure.

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, this is 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).

Topobo does make certain contributions to the breadth of tangible programming.

While the Topobo interface allows loops, some "object-oriented" control via Queens, and some functions and feedback loops via Backpacks, the system's 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.

However, these contributions are not very scalable.

Storytelling

Storytelling with multiple characters is hard to do with Topobo because storytelling requires saving and organizing sets of motions into a narrative, and the current system only allows a single recording to be played back. Since storytelling is a fun and effective way for children to learn a variety of important ideas [Han04], a natural question is how to extend Topobo to support narrative structures. Saving and recalling motions in arbitrary sequences to compose a narrative could be approached in a number of different ways.

Storytelling could engage different kinds of learners. It would require a way to save, organize and recall motions as a narrative.

One might imagine a physical "timeline dock" that is designed after common programs like Flash® and iMovie® and can be connected to a Topobo unit. Before a child programs a creation's motion, a block is inserted into a "recording" dock. This block is then associated with the program [Ull99]. A child may associate multiple programs with multiple blocks, and then arrange the blocks on a "playback timeline." A spring-loaded timeline pointer can be pulled to the beginning of the story, and as it scrolls across the timeline during playback, the various programs are played. Pauses between motions could

One way is with "phicons," physical blocks that are associated with motions and organized on a timeline dock.

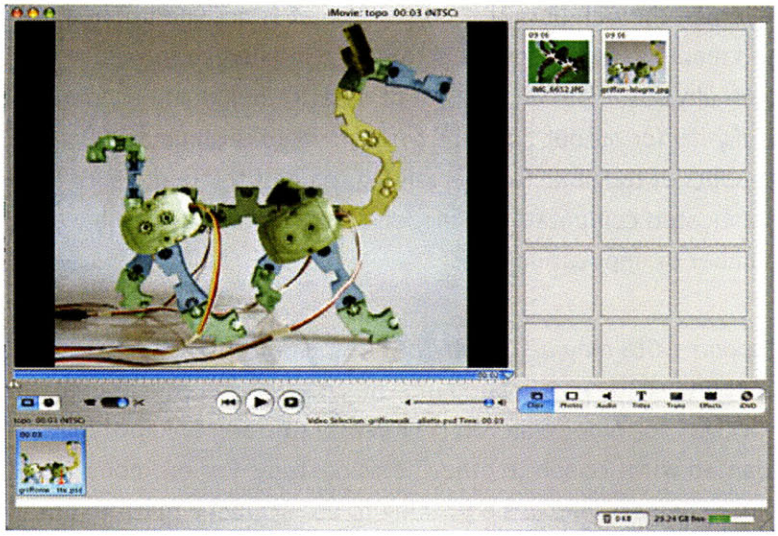
be represented with blank space in between recordings. In this way, a child could record multiple motions and then spatially arrange them to playback in a certain order. In order to coordinate multiple distinct creations (characters), a number of these docks might be stacked above each other so that they play along the same timeline.

For younger children, another interesting approach for storytelling may be to apply models presented by Cassell and Ryokai with Storymat [Cas01]. In Storymat, location is a trigger for computational events to occur. One might imagine a Storymat rolled out as a play area for Topobo creations. Different programs might be linked to different physical locations on the

Following Storymat [Cas01], Topobo creations could recall motions based on physical location on a mat. This could support storytelling or games in which a creation is programmed to walk from one location to the next.

mat, so that children could program a series of motions that would replay when their creation was on a different part of the mat. Children might decorate the mat to represent different places in which their creation would perform different movements. Physical location on Storymat could thus become a tool to facilitate children's story authoring and storytelling with Topobo. As a goal-oriented activity, children might also try to program creations to move themselves from one location to another, as a sort of hop-scotch where the creation adopts a new program at each destination.

Another way to facilitate storytelling activities is to couple Topobo with a GUI. Screen based icons could represent certain activities, such as storing motions, organizing stories, and sharing creations and motions over a distance. This approach could facilitate the first example of narrative using a timeline structure with graphical icons instead of physical blocks. Furthermore, transfer to a GUI would allow a child to use other functions already available on a home PC. For instance, a child might email a picture of his favorite Topobo creation to a friend along with the program, so that his friend can build and run the creation, too.



Topobo could be connected to a PC so that children could use an interface like iMovie® to compose narratives with their creations.

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.

Although I have argued for a return to tangibility in digital manipulative design, a GUI may benefit Topobo in certain ways that we will explore.

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.

When to integrate Topobo and a GUI

However, the system 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

How do you introduce a GUI without losing the benefits of interacting with tangible interfaces?

A GUI could add new functionality but Topobo would have to remain an interface: a tool for both input and output.

leverage the rich library of software that is designed to analyze and manipulate 3-D forms. 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 could help older children compare peer to peer models using Queens and Backpacks to the mathematical representations of similar ideas.

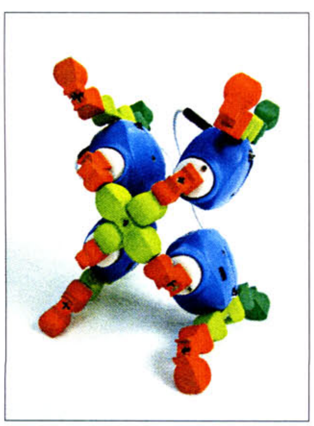
Topobo + GUI may be better for older children

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.

Topobo + GUI could support robot design if movements could be more highly controlled.

Similarly, by playing with the Queen and Amplitude Backpacks, children can create different kinds of spirals. Comparing the function of this peer-to-peer 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.

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



Topobo + GUI for robotics design

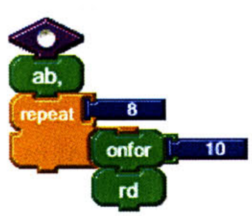
Topobo could become a tool to study robotics. It is already a good tool to "sketch" robot ideas and to discover different kinds of robot locomotion. Adding a more refined layer of control via a GUI could allow people to further develop their robot "sketches." Much as graphic designers often scan and 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.

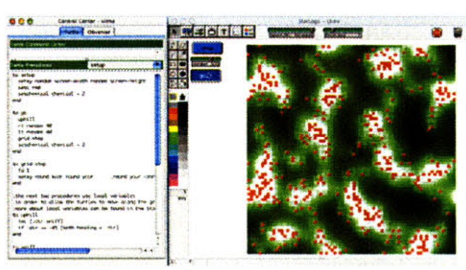
One added benefit from such a system is that Topobo would become a 3-D physical display. A designer could use the system to rough out a motion with her hands, fine tune those motions on screen with mathematical models or best-fit examples, and then observe how the edited motions actually behave in the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





Kids can learn certain ideas through programming. Can Topobo + GUI help children learn about the mathematical representations of patterns discovered through play withTopobo?

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

To remain an interface (rather than a display), Topobo+GUI would require bidirectional consistency, where changes to either the digital or physical system were immediately reflected in the rest of the system.

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.

Gestural motions could be compared to mathematical functions like a sine wave.

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. 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.

One benefit is that Topobo would show how real-world properties like gravity and friction affect ideal motions designed with a mathematical model.

9.2 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.

Features should add to the system without obscuring the system's core benefits. If the focus turns to the GUI, is something lost?

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 storytelling applications and 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.

Topobo should not become a general purpose programming tool.

If building leads to learning, is learning like design?
Designers use different media for different stages of a design's development. 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.

An interface, like a design tool, may be best suited only for a certain stage of design.

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 and Backpacks, or experiment with fine adjustments to the creation's geometry to improve its gait. 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.

As a design tool for robot design, Topobo is good for sketching.

Adding a GUI could help transfer sketches to other platforms.

... or could be used to refine a creation. The end may be the creation itself or a model upon which to base another robot.

... or the goal could be to play, have fun, and learn new ideas.

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.

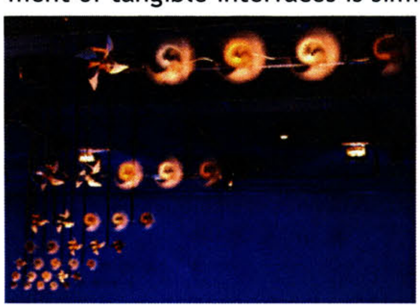
10 A TUI Material? Looking Ahead

This thesis has approached Topobo both from a design perspective, as a platform developed for generalized actuated modeling, and as a specific contribution to digital manipulatives that help children learn through interaction with physical objects. This section will consider the future of Topobo as a more generalized actuated modeling system, or as a *material*. Topobo builds on trends in tangible interfaces and may provide a foundation for a future actuated material with which people can design a variety of tangible interfaces. By viewing Topobo as a modeling material, I will consider some potential applications that stem from past research in tangible interfaces.

Could Topobo apply to more than digital manipulatives? We conclude by reconsidering the idea of a multipurpose actuated modeling material.

10.1 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




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

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.

Computers allow design and art to come alive and react to their environments.

Paintings, the film, then interactive motion graphics.

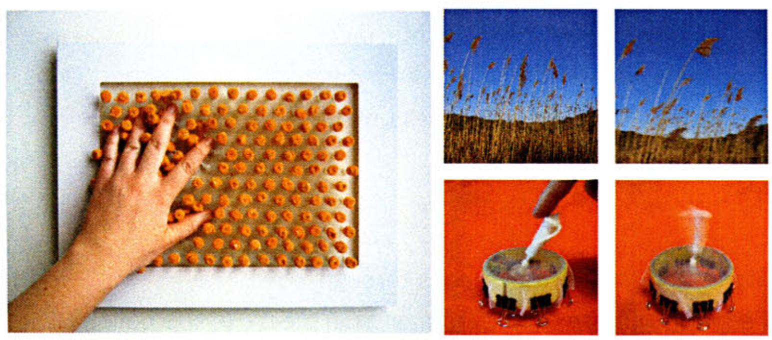
Sculpture, then mechanical automata, then tangible interfaces?

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.

10.2 Movement as Interface

TUIs use mechanical motion for input as well as output.

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



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.

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].

Is mechanical movement a fundamental quality of tangible interfaces?

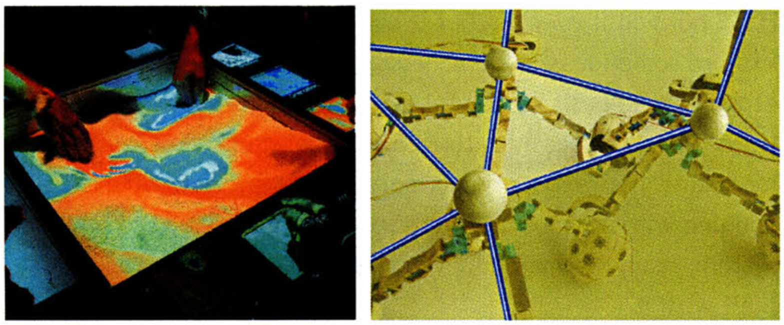
The prevalence of mechanical movement as an interface leads one to raise the question whether mechanical movement is a fundamental quality of tangible interfaces. In contrasting tangible interfaces with the graphical interface and pixel [Ish97], Ishii implied that we do not yet know the fundamentals 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?

10.3 Actuated modeling

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 variant of 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

If yes, can an actuated modeling material enable the development of TUIs?



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

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.

A good material would enable designers to create new TUIs.

An argument for a new 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.

Smaller actuators are better.

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 applicabil-

At a certain point, smaller actuators are not individually manipulable, and the mesh is more like a fabric. "Digital Clay" is much farther off.

ity 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.

We can still do a lot with big actuators. Designing good applications is the limiting factor, not the technology.

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 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.

11 Appendix A: 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 ideas and suggestions for other actuated modeling systems.

A few ideas how to build it better next time.

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

Smaller actuators have to be organized carefully so that they are powerful.

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.

Smaller linear actuators could have binary states.

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.

Binary actuators could apply to MEMS. They would have to be self-assembling

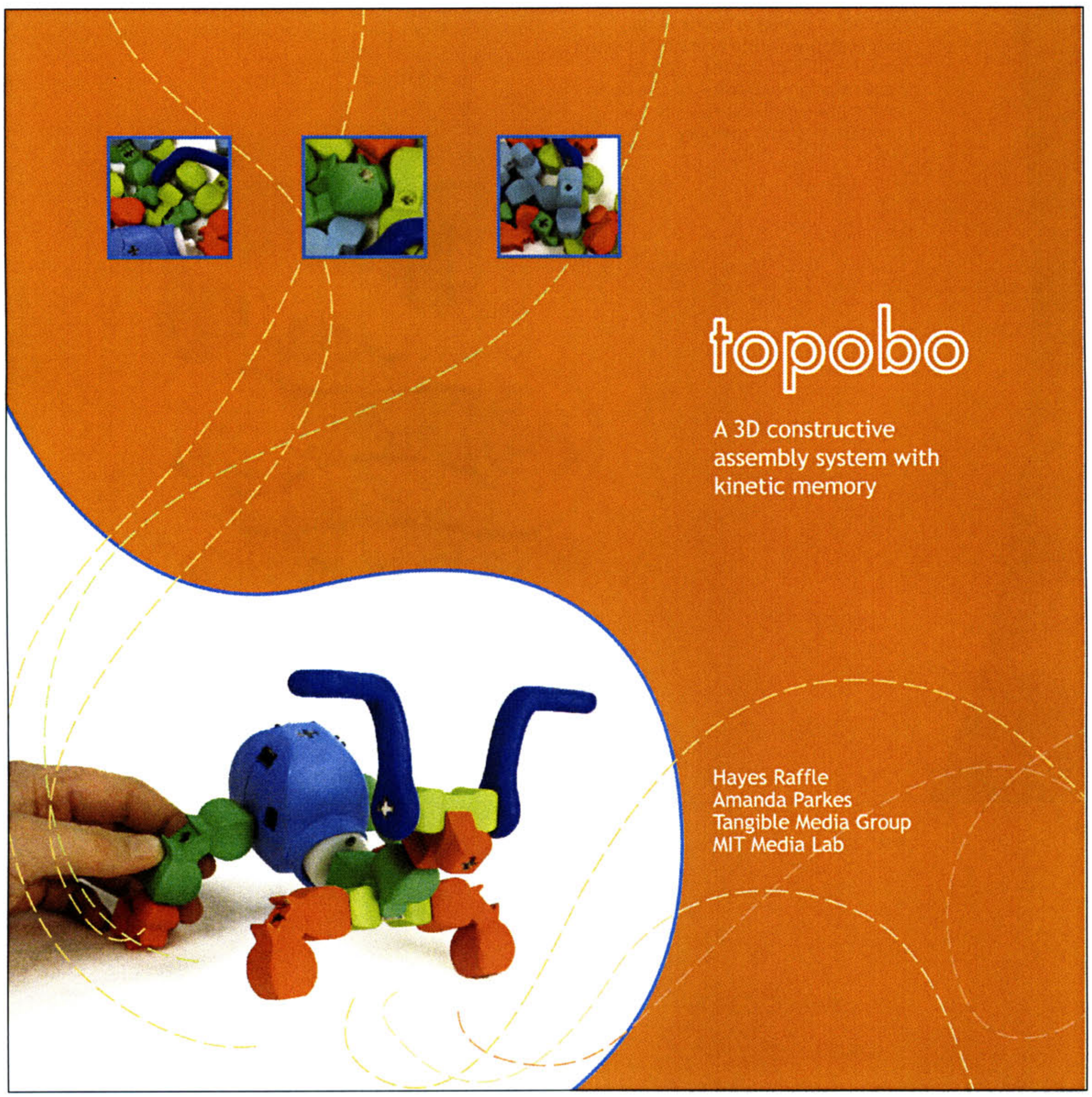
The electrical element of such systems (if they are not superseded by chemical control structures) might be modeled on Butera's "paintable computers" [But02] that use massively

Paintable computers may provide a model for smaller electrically driven systems.

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.

12 Appendix B: Topobo Brochure

Amanda Parkes and I originally designed this brochure and an accompanying package for submission to the I.D. Magazine design competition. We later modified the brochure to this version, to communicate Topobo to a non-academic audience.



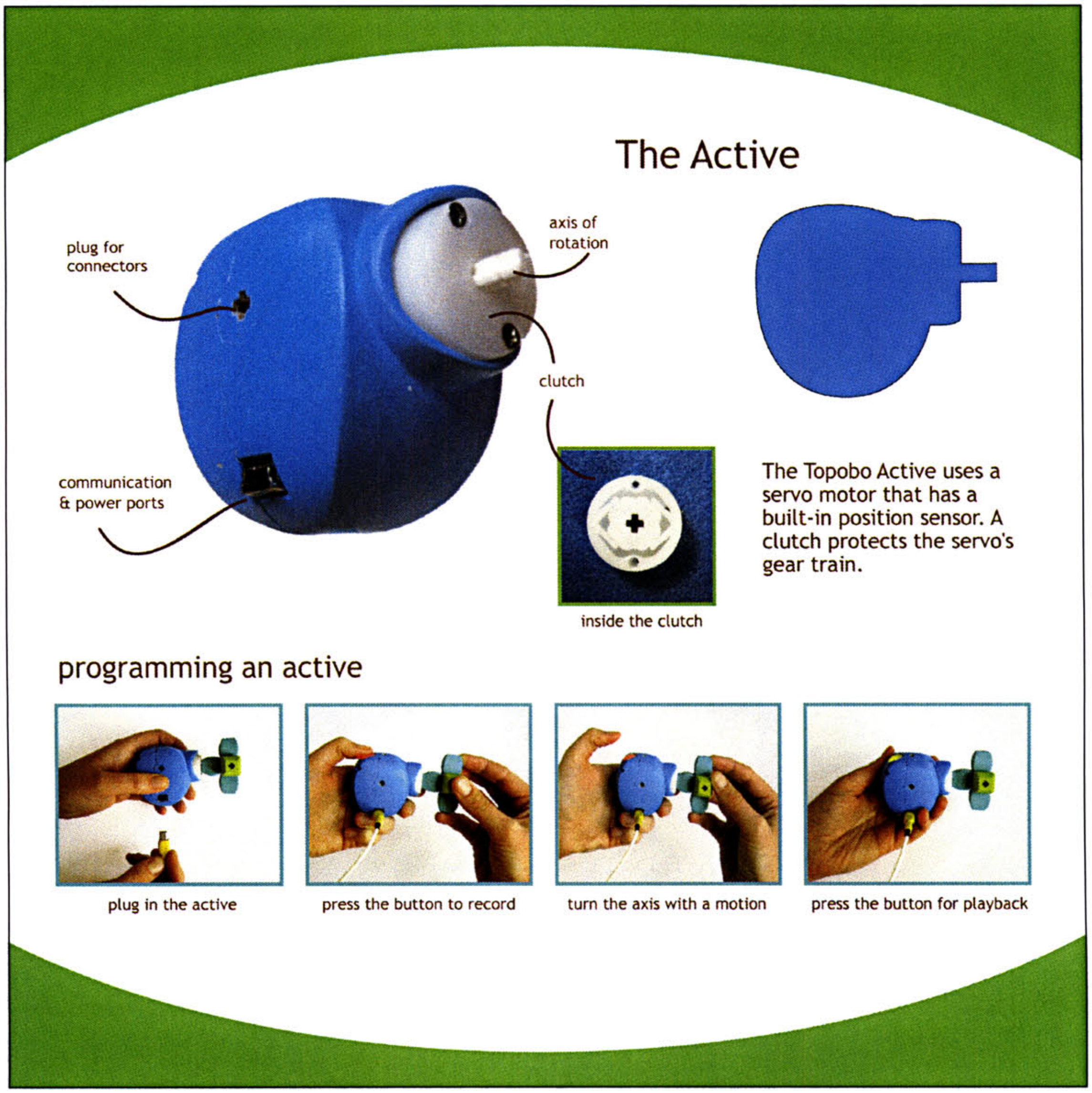
The Concept

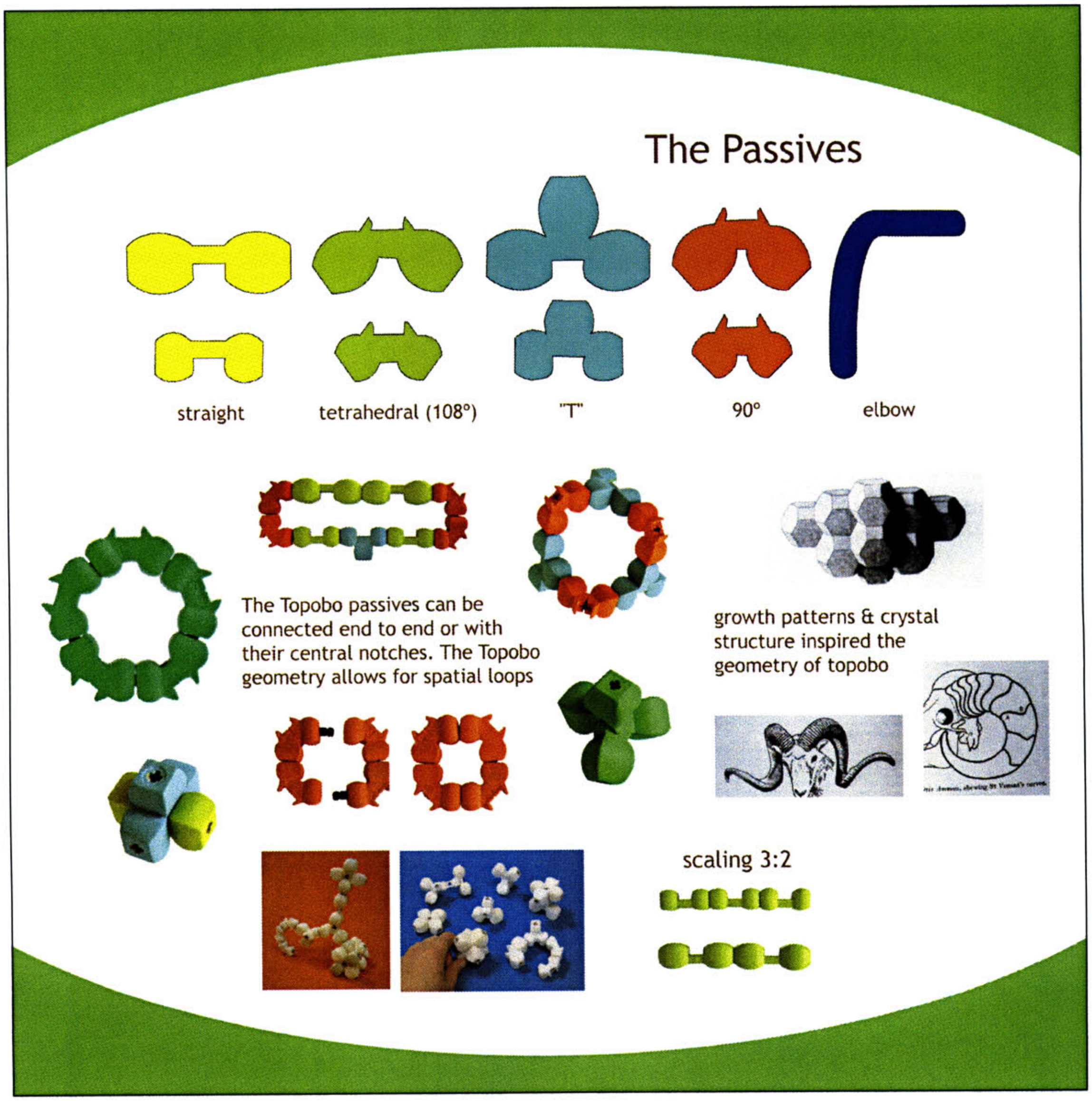
What is it like to sculpt with motion?

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.

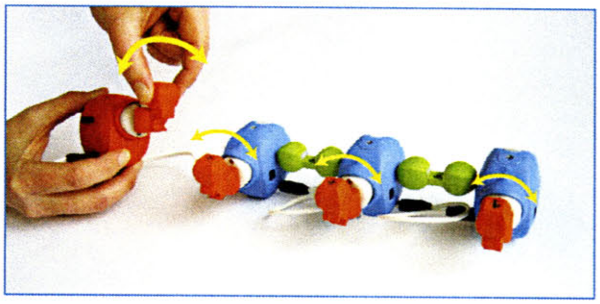
System Design

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
plug for connectors
axis of rotation
clutch
communication & power ports
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

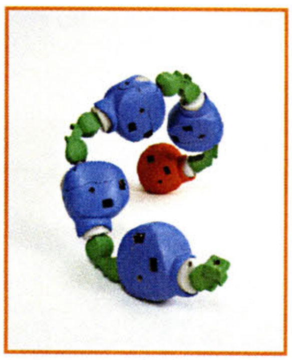

The Passives
straight
tetrahedral (108°)
"T"
90°
elbow
The Topobo passives can be connected end to end or with their central notches. The Topobo geometry allows for spatial loops
growth patterns & crystal structure inspired the geometry of topobo
scaling 3:2

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



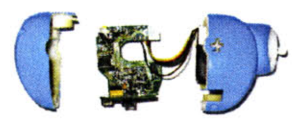
actives connected with tetrahedral passives create a spring-like helix when controlled by a queen.

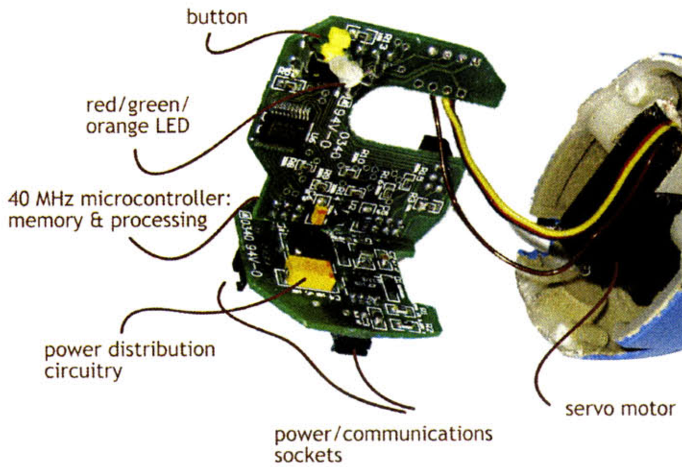
a linear sequence with a queen creates a circle

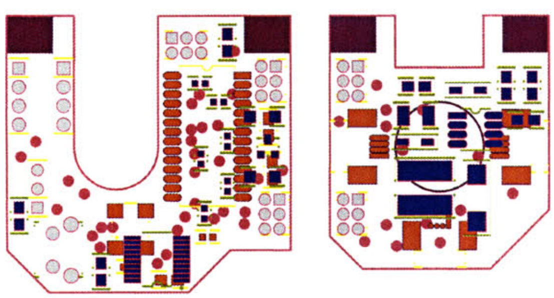
a special "decay Queen" tells each active to increasingly scale the Queen's motion. a linear sequence creates a spiral

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

Topobo Technology



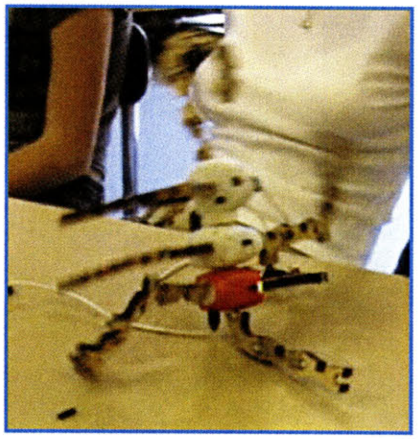


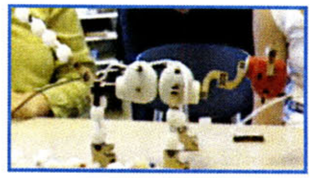


electronics design

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



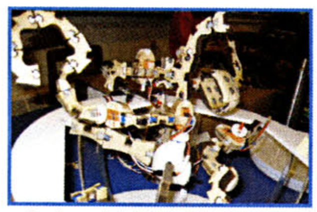


A tool for cooperative learning

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.

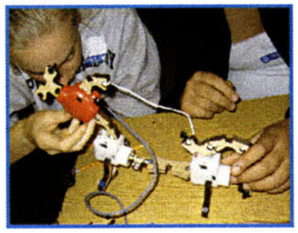
We found that Topobo can help students ages 7-13 to learn about:

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



a 2nd grade collaboration

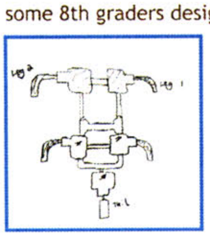
two 8th graders programming together

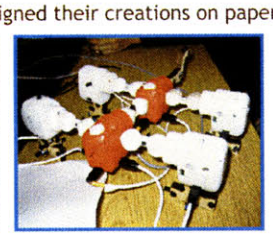


Animals & Machines

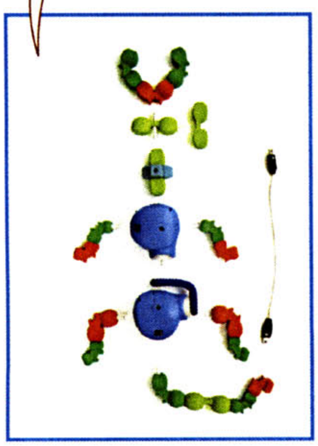
Kindergartners, second graders and eighth graders all related to Topobo models with their "familiar knowledge" about animals and machines. Metaphoric allusions to machines (robotics) and especially to animals ("the elephant," "the ant," "the scorpion," "the horse," "the no-walking man") were descriptive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

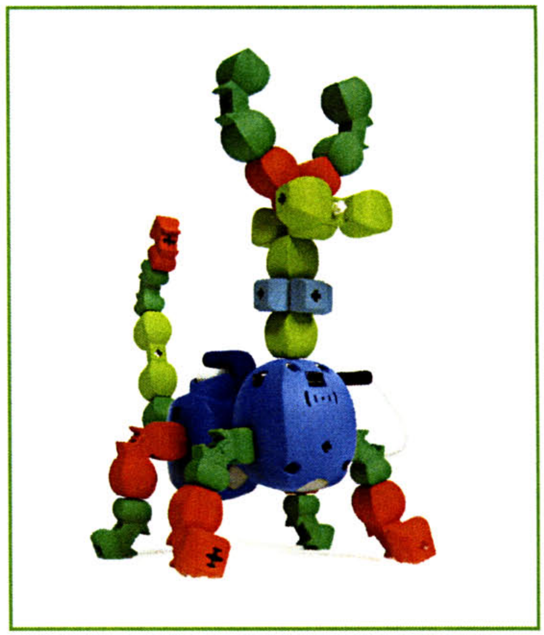
some 8th graders designed their creations on paper



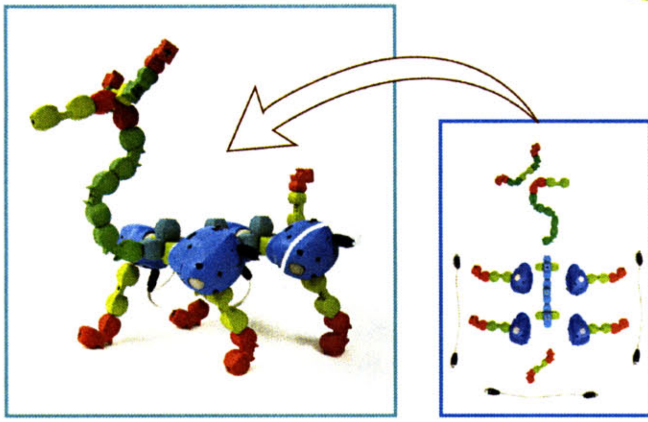


Topobo Creations

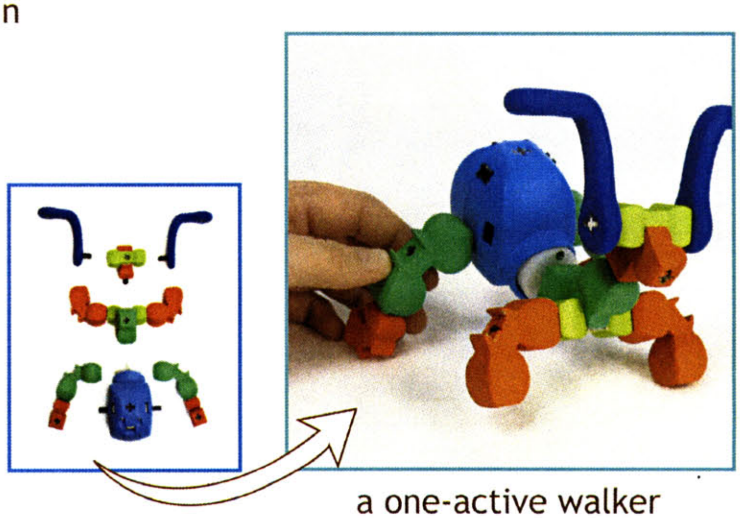




a moose

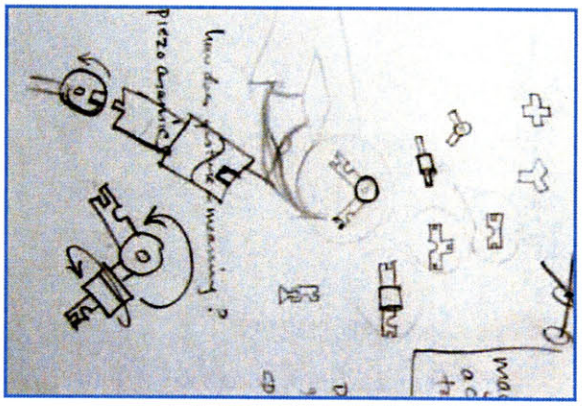


a griffin

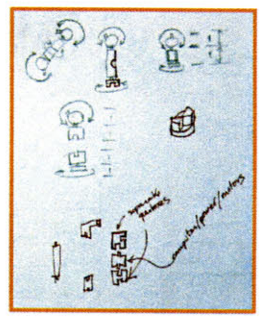


a one-active walker

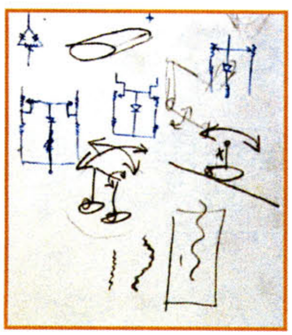
Concept & Early Prototypes



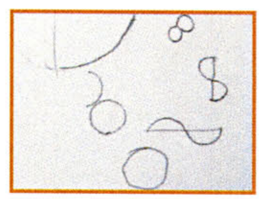
After choosing motors as our actuators, we did drawings to study spatial relationships possible with different arrangements of pivot joints. A branching system was an early choice for geometry.



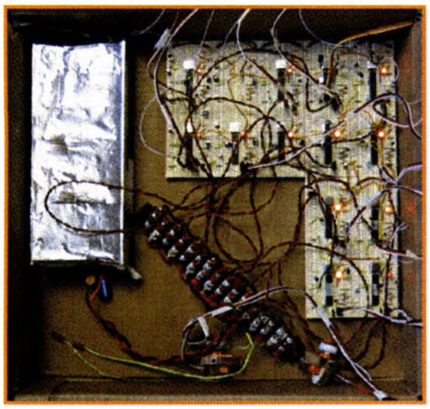
assembly drawings questioned the possibility of fitting motors and electronics in one designed system.



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

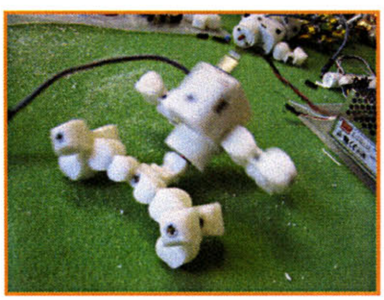


early studies of waves led to the time delay Queen.



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

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

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

[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.

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

[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.

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

[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.

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

[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.

[Ish99] 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.

[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.

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

[Mon12] Montessori, M. (1912). *The montessori method.* New York: Frederick Stokes Co.

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

[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.

[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.

[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.

[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.

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

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

[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.

[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.