FlexM: Designing a physical construction kit for 3d modeling

Markus Eng, Ken Camarata, Ellen Yi-Luen Do and Mark D Gross

FlexM: Designing a physical construction kit for 3d modeling

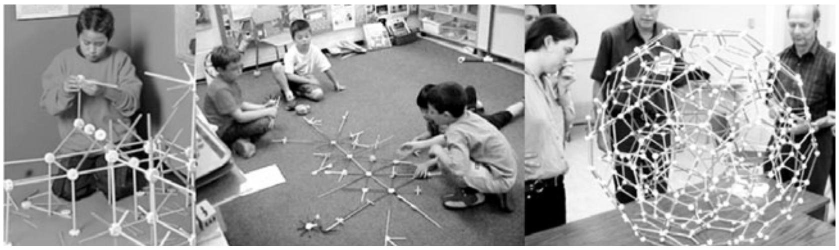
Markus Eng, Ken Camarata, Ellen Yi-Luen Do and Mark D Gross

We have designed a hub and strut kit that interfaces to a 3D graphics application. FlexM is a prototype flexible physical interface for manipulating and building 3D geometry. Using the FlexM hub and strut components, designers can build and explore 3D geometry with the ease of a toy and the power of a computer. The hubs transmit the model's topology and geometry to the computer, where the model is rendered on the screen in real time. The paper reports on the iterative development of several versions of the project.

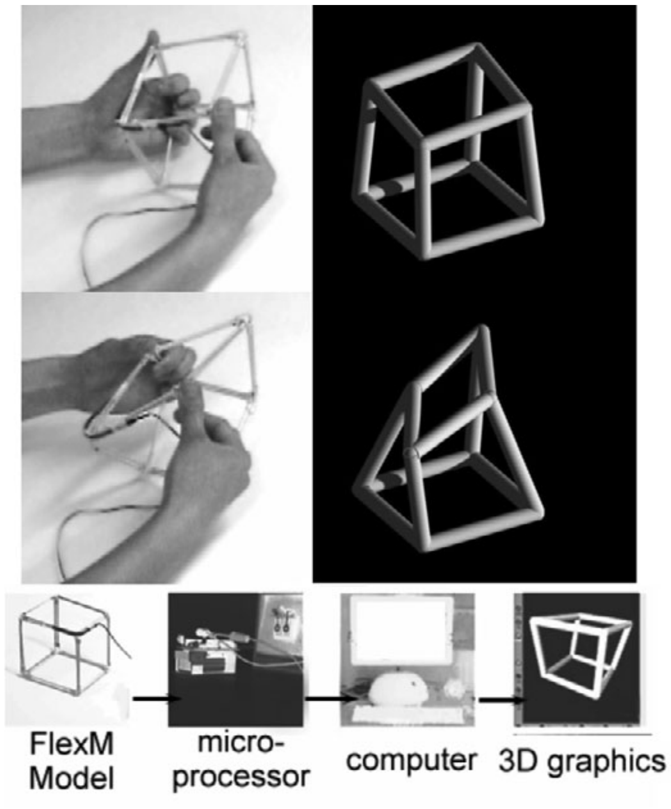
1. DESIGN CONCEPT

Designers often struggle with standard CAD software. The graphical user interfaces of screen-based modeling applications lack the direct tactile feedback of the form being manipulated. The software packages require an explicit commitment of action with complex commands and menu operations. Using a 3D graphics application, especially sophisticated ones such as 3D Studio or Maya demands an additional level of mental abstraction. In order to become adept in using software, the designer must understand the application's logic, and follow specific strategies of construction that bear little relation to how the modeled artifacts are actually to be made. To construct a cube on the computer requires the designer to click and drag the mouse, type dimensions, rotate and extrude figures, shifting through different views. These user-interface operations are very different from physical model building. Designers can more deeply understand 3D geometry and spatial hierarchy in a design when they can perceive it in a physical model.

Can the difficulties designers experience with CAD applications be improved by providing the haptics of physical construction? What if CAD possessed the attributes of physical construction toys, which are accessible, intuitive and fun? Construction kits, like K'nex^{TM}, LEGO^{TM} Technic, Magnetic Geomag^{TM} or Ramagon^{TM}, are easy to use, have flexible, moving pieces, and allow endless combinations for creativity [7]. Children and adults enjoy playing with construction toys because they facilitate the exploration of geometry, mechanics and kinematics. However, construction toys lack the ability to transfer the physical design into a digital format in order that the design can be further explored in greater depth with feedback or simulation that can further inform the design construction.


► Figure 1: Tinker Toys, K'nex, Zometools

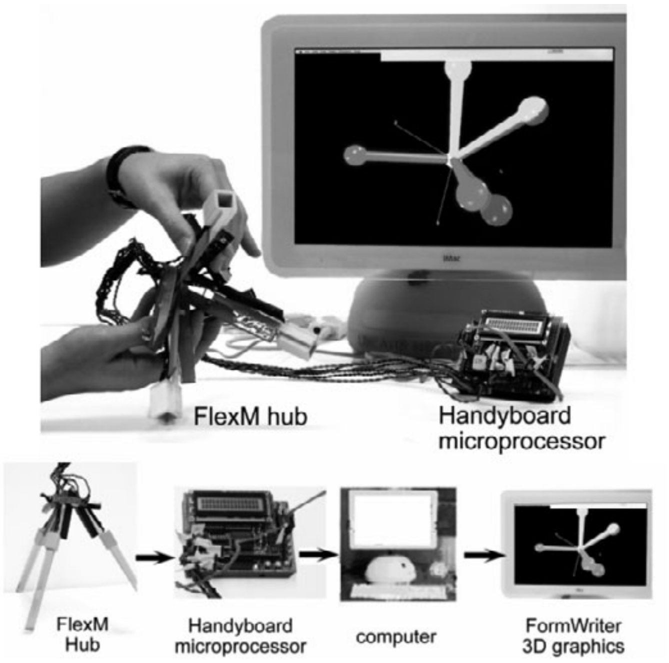
What if one had the option to build the digital model strictly with the physical components without touching a mouse or a keyboard? FlexM is a computationally enhanced construction kit of hubs and struts with which the designer can explore 3D manipulations with the spirit of a toy and the power of a computer. It belongs to the class of construction kits that includes Tinker Toys, K'nex and Zometools, that is, kits that consist of hubs and struts. With these basic pieces one can build a multitude of shapes (Figure 1). Because the hubs in these kits are rigid, they do not easily support forms that could be squished, twisted or deformed. The FlexM hub is flexible, allowing a greater range of movement than its toy counterparts do.



◀ Figure 2: Our first prototype: flexible model made from wooden sticks, surgical tubing, and a bend sensor driving the graphics display in VRML.

We envision the designer building a model with FlexM hubs and struts, manipulating the model, and seeing the digital version of the model change with real-time manipulations. The model can refer to a building's spatial composition, a simplified version of the building with key points in the model mapping to actual points on the building, or its structural frame. Or, the designer could use an existing digital model, and map key points to the FlexM model. The hubs have a flexible socket connection, so the designer can shear, twist, and rotate parts of the model. The model connects to a computer via a Handyboard microprocessor [1]), which transfers the model's topology and geometry to a receiving CAD or 3D graphics application. The graphics program uses these data to reconstruct and update the model. The application not only displays the model's changing form, but can also record the data for animation or separate modeling case scenarios.

Figure 2 illustrates the FlexM concept of a flexible physical model driving the graphics display. This was our first prototype built of sticks and surgical tubing with a bend sensor mounted at the corner. The computer graphics figures are generated in VRML. Figure 3 shows a more recent prototype of the FlexM flexible hub interfacing to the FormWriter 3D graphics program [2] via a Handyboard microprocessor. An image of the FlexM hub displays on the screen. As the designer squishes the angles at the hub, the hub displayed on the screen also squishes. The goal is to fabricate more hubs, so the designer could then build a cube with one hub at each corner. A 3D graphics application such as the FormWriter program would then update the cube model on the screen.



► Figure 3. FlexM with Handyboard and corresponding graphics display.

2. RELATED WORK

FlexM is about playing, building and visualizing. Play is a form of thinking with one's hands. Building is creating form, and visualizing is interpreting information. To elaborate on these goals, we review related work in three categories: traditional construction toys, computational construction kits and parametric based graphics.

Construction kits have been around in consumer society since the 1910's. A. C. Gilbert invented the Erector set in 1913 [3]. In the same year Pajeau conceived of Tinker Toys. Current kits, like LEGO^{TM} and K'nex^{TM} toys, as well as the Hoberman Sphere, offer dynamic, expanding joints. The Hoberman Sphere, a compact ball of interconnected, folded hinges, expands into a sphere, over four times its original size. Its transformation from a stellated polyhedron to a geodesic sphere encompasses both a technical and a toy-like appeal [4]. The Hoberman Expandagon Gro-Bot tumbler has flexible parts that can extend and expand a geometrical shaped "robot" to stand up and flip.

Computational construction kits are a new development, now possible because of the miniaturization of electronic devices. Gorbet and Orth's [5] Triangles is a construction kit of flat, plastic triangles, that interface to a computer. Each triangle tile corresponds to a different application, like an email program, or a personal calendar. The user activates the program through the tile face. The pieces have integrated, mechanical and electronic, magnetic connectors, which allow the user to build a variety of geometric forms that correspond to his suite of applications.

Early work in physical interaction with computational models dates to Negroponte's Architecture Machine Group [6]. Anderson et al.'s Computational Building Blocks [7] facilitates computer modeling with LEGO(tm) like blocks. This work expands on pioneering explorations of Aish [8], and subsequent research by Frazer [9], and Dewey and Patera [10]. Aish's Building Block System was a block set for interactively representing the structure and physical properties of the world. Frazer's 3D input devices, "Machine Readable Models" and "Intelligent Modeling Systems," enabled designers to build models that interface with software that can give design advice. Dewey and Patera developed processors to manipulate the geometry of 3D models. All these projects, however, lack a real-time interface for detecting moving pieces. Anderson et al's Computational Building Blocks are static pieces. Although Gorbet and Orth's Triangles have hinges, they assemble to make a static, rigid form.

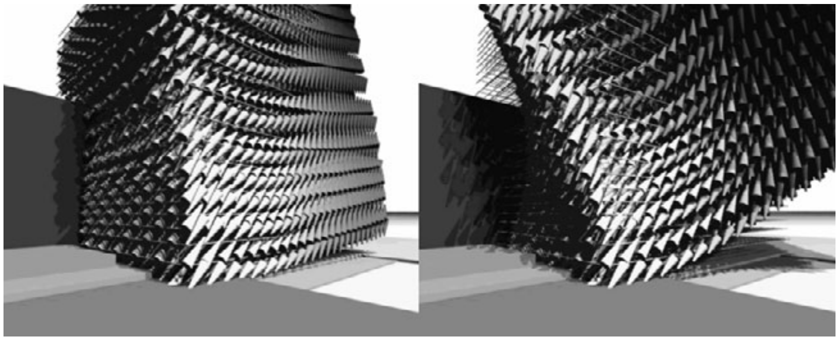
The mouse, joystick, trackpad, and pen are the usual hardware devices used to interact with a three-dimensional model on the screen of a desktop or laptop computer. A mouse, for example, encodes movement in the horizontal plane, and a joystick captures azimuth and bearing angles. They are deictic devices, used to indicate, select, and operate on objects on the screen. In contrast, the devices mentioned next, and FlexM, are embodiments of the models on the screen, providing a direct tangible mode of interaction with the models, rather than the indirect mode that a mouse or joystick affords.

Several projects track the movements of physical objects to generate animation. Monkey^{TM} is a specialized input device for virtual body animation [11]. It resembles a mechanical mannequin with articulated limbs. Instead of constructing a simulation of human animation and locomotion using a screen interface, the animator poses and moves the Monkey^{TM} to define the character's animation. Topobo is another project involving character animation [12]. It is a construction kit of articulating vertebra-like pieces for building assemblies with embedded kinetic memory. Its embedded memory records the angular movement at the joints. Users build a creature, move the model across a terrain, and then watch the model replay its movement from its embedded kinetic memory.

Similar to Topobo's mechanical widgets, Phidgets is a construction kit of physical computing widgets: sensors, motors, radio frequency ID readers, and a software interface for user interaction [13]. For example, users can use a motion sensor at a doorway to activate a light in the adjacent room to signal someone entering. Phidgets do not require knowledge of processors, or communication protocols. Their ease of use, modularity and ability to facilitate event-driven interaction make them a handy resource for prototyping tangible user interfaces.

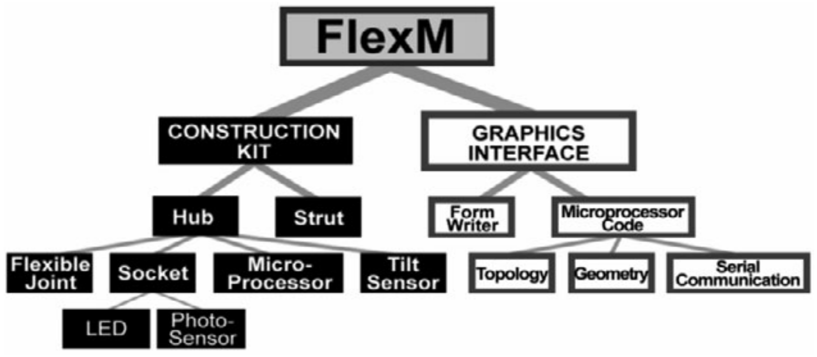
CUBIK is a tangible modeling interface to aid architects and designers in 3D modeling. It takes the form of a mechanical cube [14]. The designer manipulates dials on the cube's face to expand or contract the face's dimension. CUBIK's corresponding graphic user interface (GUI) displays in real-time how the cube is expanding or contracting. The communication between the GUI and CUBIK is bi-directional. The designer can manipulate the physical cube through the GUI, or change the cube's shape in the GUI via the mechanical cube. Both CUBIK and Monkey^{TM} have engaging physical interfaces, which encourage the spontaneous act of play.

► Figure 4: Parametric based graphics rendered in PovRay



One motivation for FlexM arose from our experience with the Persistence of Vision raytracing engine (POV-Ray). POV-Ray is a script-based, shareware 3D graphics application *(http://povray.org.)*. A user can create and manipulate complicated forms through an algorithm, macro or user defined function. Artists and designers can script visually complex, parametric designs (Figure 4). Yet, for most people, writing script is neither an easy nor an intuitive task. We want to make it easy for designers to create and manipulate 3D models using a physical interface. Therefore, we proposed a tangible interface that could shear, twist or distort the computer model, in place of learning the POV-Ray script.

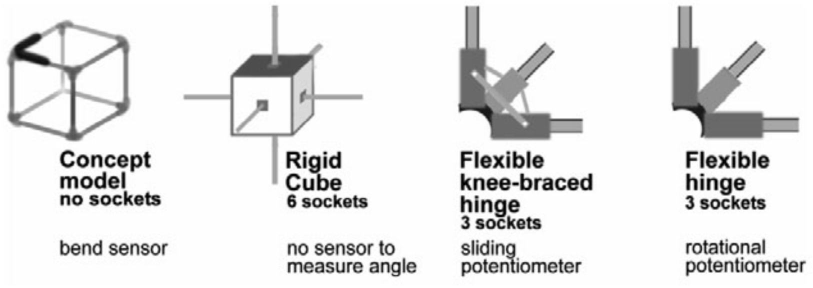
► Figure 5: FlexM components: hardware (left) and software (right).



3. INVESTIGATION

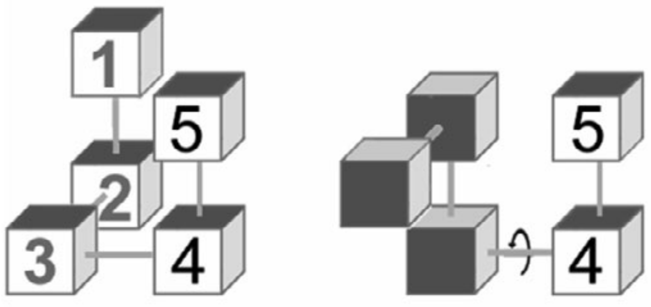
FlexM is a computationally enhanced construction kit [15] to interact with a 3D modeling program. The computational enhancements are the following: photosensors are used for determining topology; the Handyboard microprocessor reads, records and transmits the sensor data to the host computer; and the software that translates the data into a 3D model in the FormWriter graphics program. FlexM is composed of specially constructed hubs and struts that send the model's topology and geometry information to a graphics application on a host computer (Figure 5).

The development of FlexM evolved from concept model, to rigid cube, to the flexible hinge (Figure 6). Much of the design focuses on the flexible hub. It consists of flexible joints, sockets to connect the hubs with each other, tilt sensors for orientation, and LEDs and photosensors for communication. A microprocessor attached to each hub operates it and sends data to a desktop computer for display or further processing.



◀ Figure 6: Development of FlexM hub from concept model to the flexible hub.

In order to reconstruct the model digitally, the graphics application must capture the model's topology and geometry. The topology defines what components are connected to each other. The geometry comprises of the angles of their connections and the orientation of these angles. For example in Figure 7, both models have the same topology (cube 1 is connected to 2, etc.), but different geometries. In the model on the right the first three cubes are rotated 90 degrees counter-clockwise along the strut connecting cube 3 and 4. The FlexM hub components listed in Figure 4 would identify and send the model's topology and geometry. The next sections detail how each of the parts achieves this.

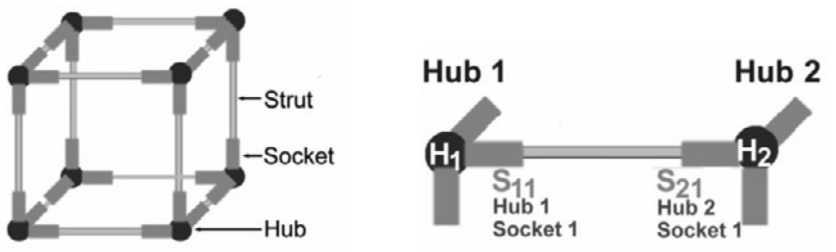


◀ Figure 7: Both models share the same typology, but their geometries differ.

3.1. TOPOLOGY

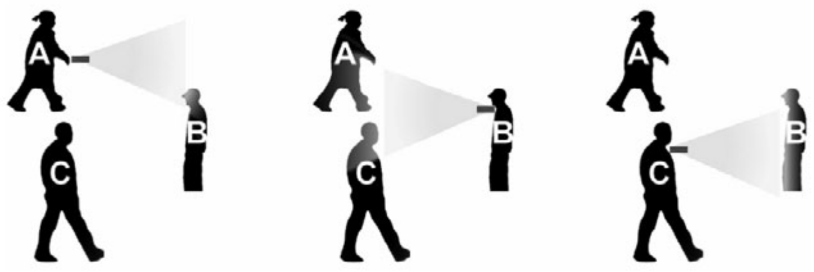
FlexM modules have two physical, components: hubs and struts. A model is a collection of n hubs (H_1, H_2, H_3....H_n), which are vertices on the model. For example, a model of a cube would have eight hubs, one for each vertex (Figure 8, left image). Each hub has a set of *m* sockets (S_{11}, S_{12}, S_{13}... S_m), equivalent to the connections on each hub. In the cube example, each of the eight hubs has three sockets or connection points. Two hubs are connected through a single strut, which make up an edge of the model. The cube has 12 edges (half the number of sockets, since it takes two connected sockets to define an edge). A strut connects to a socket on each pair of hubs (Figure 8, right image).

► Figure 8: (left) Definition of the parts, defining the model's topology. (right) Detail of Hub and Socket topology. Hub 1 connects to Hub 2, through Hub 1's Socket 1 and Hub 2's Socket 1.

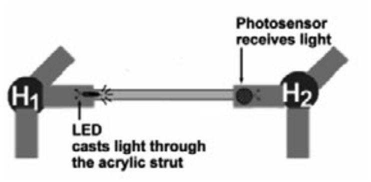


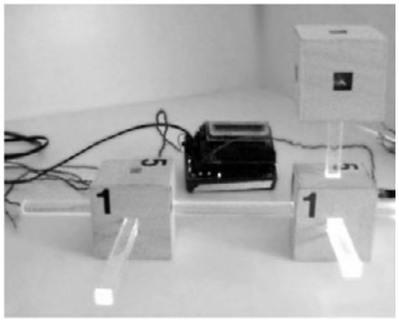
A method to identify the model's topology can be explained through the Flashlight Game. In Figure 9, player A points a flashlight. Player B sees the light and knows that it is A. In turn, B shines the flashlight. Likewise, A sees the light and knows that it is B. C also sees the light and knows that it is B. Finally, player C shines the flashlight, and B sees the light and knows it is C. This associative coupling of information defines the paired connections among the players.In the present FlexM prototype, each hub has three

► Figure 9: Flashlight game



sockets, which are both the physical and electronic connectors to the hubs via the struts. With Flashlight game metaphor, the LED and photosensor define the connection using light as the communication medium. Each socket has a high intensity LED and a photosensor. When switched on, the LED casts light from the socket through the acrylic strut to a photosensor in the socket of the attached hub (Figure 10). Each hub in turn flashes on, beaming light to the attached hubs. Each receiving hub checks if its photosensors can see the sending hub. This process produces a list of connections between the hubs (hub A connects to hub B's socket 1; hub B connects to hub C's socket 4; etc.). Because this prototype only allows one connector between two hubs, one need not include the socket of the "sending hub." The connection "hub A socket 1 connects to hub B socket 2" can be deduced from the pair "hub A connects to hub B socket 2" and "hub B connects to hub A socket 1." This list is the connectivity information an interfaced graphics program needs to construct the digital model.





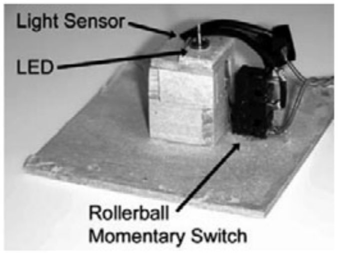
◀ Figure 10: (left) FlexM socket connection. (right) Early model of lit hub with one socket for each face. Hubs in this prototype were constructed as wooden cubes containing lights with acrylic struts.

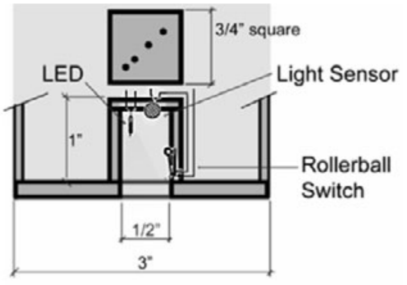
3.1.1 Socket Design

The socket design went through ongoing modifications during the development of three hub prototypes: rigid cube, flexible knee-braced hinge, and the flexible hinge (Figure 6). To simplify the geometry, the socket remained a square, which prevents the strut from rotating in the socket.

The first socket design incorporated a rollerball switch to identify when a strut was connected to the socket (Figure 11). This helped differentiate light readings from the lit LED and ambient lighting. (Section 3.1.2 discusses ambient light further.) The inserted strut engaged the switch, which activated the LED and photosensor. When the switch was off (no strut), no LED would flash on and no reading would be made from the photosensor. Despite these advantages, the rollerball switch was not used in later designs to simplify the circuitry and the construction, and because we resolved the ambient lighting issues by employing a stronger light source.

The second socket design simplified the hub wiring by reducing the number of LEDs. Three LEDs are placed in the center of the hub to create a single light source. In the back of each socket, an aperture permits light to

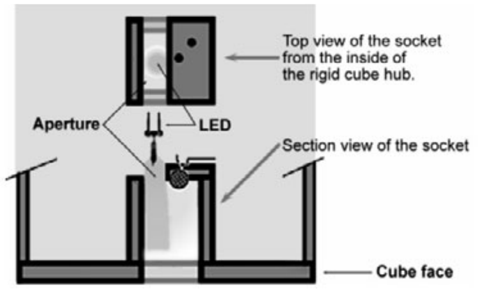


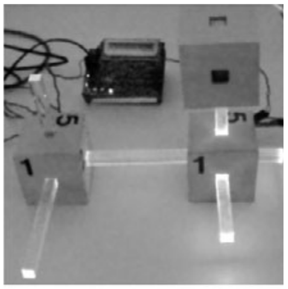


◀ Figure 11: (left) Inside face of hub cube with the socket components. (right) Corresponding section detail of socket.

shine from the center core of LEDs through opposite faces of the hub (Figure 12, left image). This approach reduces the number of LEDs from six per hub (one for each socket) to three per hub, resulting in fewer soldering points. However, the alternate design gave a reduced performance in light intensity (Figure 12, right image).

► Figure 12: (left) Section cut of cube face showing the LED shining through an aperture in the socket. (right) Photo shows the difference in light intensity between the cube with LEDs in the hub's core (left cube) and cubes with LEDs in each socket (right cubes).


Top view of the socket from the inside of the rigid cube hub.
Aperture
LED
Section view of the socket
Cube face

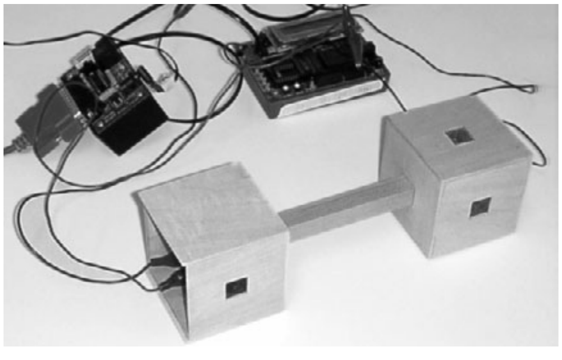


The drop in light intensity increased the range of photosensor readings, which complicated the programming. Light from the core LEDs was too weak to distinguish a photosensor reading from an ambient light reading. The decrease in light was due to LED orientation and wiring obstruction. It was also difficult to orient three LEDs to shine equal amounts of light to six socket apertures with wiring obstructing the line of sight. As a result, we decided to keep the original design of having six LEDs, one for each socket.

3.1.2 Strut Design

The strut serves two functions: (1) physical connection between hubs, and (2) the medium to channel light to the connecting hub. The struts act as connectors or edges between the hubs. The initial design was a square, hollow, wooden strut. The strut was square in section to prevent rotation between the hubs (Figure 13).

► Figure 13: Hubs connected by a hollow wooden strut.



The second iteration was the square, acrylic strut. Acrylic is an excellent medium for transmitting light. The difference in material did not influence the strut's function as a mechanical connector. Light studies we performed revealed that, the wood strut has a critical difference in transmitting light, compared to the acrylic one (Table 1). Wood has two advantages. First, ambient light does not interfere with the light sensor reading. Second, the light sensor readings from struts of varying length are easily discernible. In the hollow wood strut, light is absorbed by the inside wall, attenuating the intensity of light at the end of the strut. The differing light sensor values are discrete, which makes it possible to identify one length of a strut from another.

		PHOTOSENSOR READING	
		0=highest level of light	255=no light
		No Ambient Light	Ambient Light
WOOD STRUT	4-inch length	57	55
	6-inch length	99	96
ACRYLIC STRUT	2-inch length	17	12
	6-inch length	15	8

◀ Table 1: Strut material, strut length and ambient lighting affect the photosensor reading of a standard LED. Note that a high intensity LED yields a zero photosensor reading, the highest measurable amount.

The transparency of acrylic creates both disadvantages and advantages. The disadvantage is that it greatly reduces the attenuation of light which makes it possible to differentiate between strut lengths. Table 1 shows that light sensor measurements for the wood strut differ considerably between the two lengths from light attenuation. Because acrylic is clear, light attenuates only slightly, resulting in close readings between the two strut lengths. It would be difficult to differentiate between struts of varying lengths because the slight difference in light readings could be less than the error in the readings.

However, the transparent acrylic is excellent in transmitting a strong signal to the photosensor. Another advantage of using acrylic over wooden struts is the ease of fabrication. Acrylic struts are easier to make than wooden struts. Each wood strut required cutting, gluing and sanding four wooden strips to form the hollow, square strut. The acrylic strut can be cut to size from the square rod.

Ambient light introduces a noise issue for the acrylic strut. In lighting condition studies, we observed that, the acrylic strut collects more light from the room's fluorescent lighting than the sending LED. We resolved this by replacing the LED light with a high intensity LED. The high intensity LED exceeds the maximum threshold of the light sensor, which indicates the highest light reading.

In summary, acrylic struts with high intensity LED lights are an effective system to identify hub connectivity.

3.2. GEOMETRY

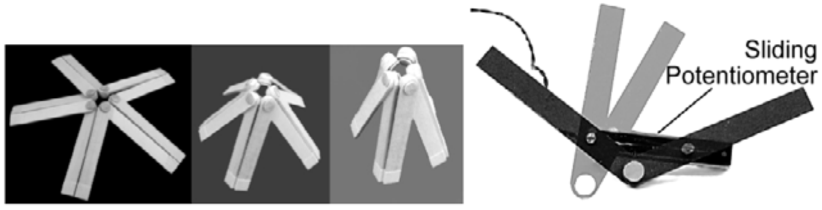
LEDs and photosensors identify the connections, but they do not measure angles of the connections, the geometry. We explored different sensors to measure the *angle* between the struts at the hub: bend sensor, sliding potentiometer and rotational potentiometer. However, these sensors only measure the angle, not the vector. One also needs the orientation of the angle. To resolve orientation we utilized tilt sensors to identify orientation.

3.2.1 Bend Sensor

The bend sensor was the first device we explored in our first FlexM hub prototype because of its ability to read an angle bend and its unobtrusiveness in structure (Figures 2). Two problems precluded using it in subsequent FlexM hub prototypes. The bend sensors gave inconsistent discrete readings and the sensor readings changed over time due to mechanical fatigue. Furthermore, the readings appeared inconsistent between multiple bend sensors, which required calibrating the bending angle to each sensor.

3.2.2 Sliding Potentiometer

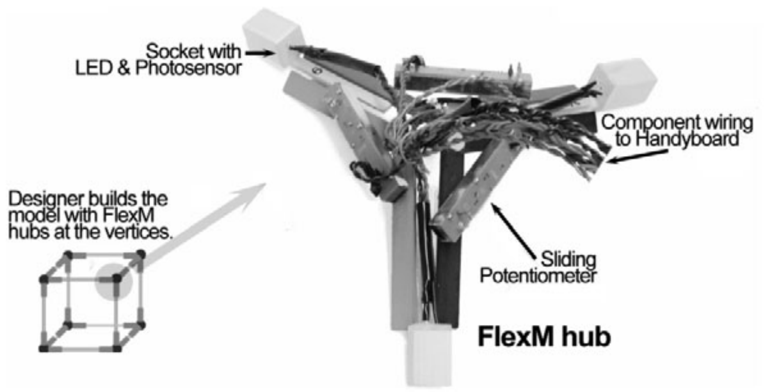
We next investigated using a sliding potentiometer to measure the strut angle at the hub. This implementation marks a shift in the hub design in measuring the angle between the socket arms. Earlier designs involved cubes with fixed sockets-this minimized variables in resolving connectivity. The hub design evolved from the rigid cube to the flexible hub with hinges. In the next prototype, we incorporated sliding potentiometers into the hinges (Figure 14).



► Figure 14: (left) Flexible hinge concept. (right) Sliding potentiometer measuring the angle bend.

Figure 14 shows the integration of the socket and its LED and photosensor with the sliding potentiometer at the hinge. The lower left diagram of the cube shows how the flexible hub fits in the model. Wires connect each hub to the Handyboard, which reads the sensors and activate the LEDs. The Handyboard transmits the model's configuration via a serial port to a computer. On the computer, a receiving 3D graphics application displays the model.

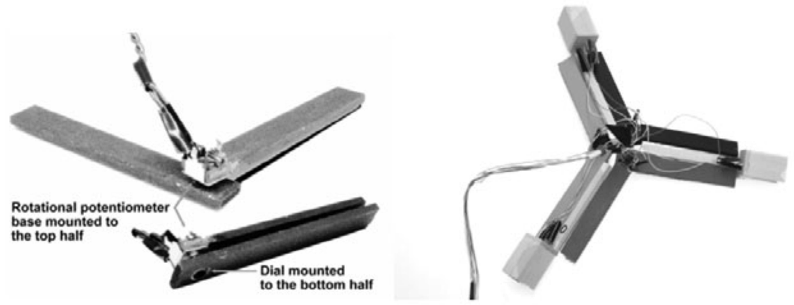
This hub has three sliding potentiometers, one for each hinge. Three hinges facilitate movement in three axes. Based mechanically on the hinge concept in Figure 14, each socket on the hub can move independently from the other. If the potentiometers were not so bulky, the hinge could even bend from convex angles to concave angles. Although the sliding potentiometer added to the sturdiness of the hinge by acting as a knee brace, it also added unwanted bulk.


Socket with LED & Photosensor
Component wiring to Handyboard
Designer builds the model with FlexM hubs at the vertices.
Sliding Potentiometer
FlexM hub

◀ Figure 15: Current hub design with sliding potentiometers in the hinges. The wiring connects the LEDs, photosensors and potentiometers to the HandyBoard.

3.2.3 Rotational Potentiometers

Rotational potentiometers solved the bulkiness of the sliding potentiometer. By allowing the potentiometer to be the mechanical hinge, we simplified the hinge design. Figure 16 (left) shows the basic hinge module with half of the hinge attached to the base of the sliding potentiometer, and the other half attached to its dial. The right image shows the entire flexible hub.


Rotational potentiometer base mounted to the top half
Dial mounted to the bottom half

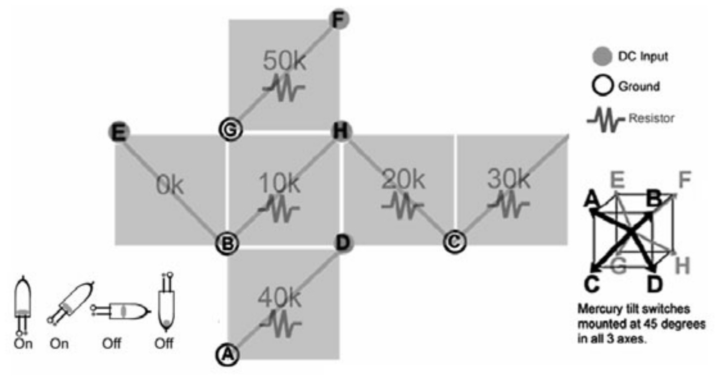
◀ Figure 16: (left) Rotational potentiometer mounted to a hinge. (right) Flexible hub with rotational potentiometers.

3.2.4 Mercury Tilt Sensors

Because the potentiometers that measure the hub geometry only measure the angle, not the vector, orientation information is absent. We have incorporated mercury tilt switches in determining the orientation of the hub. The tilt switch is a small tube with a drop of mercury in it. In the standard vertical position, the mercury settles to the bottom, closing the circuit (the ON position) (Figure 17, left image). Likewise, the circuit is open when the switch is upside down or sideways (the OFF position).

Figure 17 (center image) shows the connections for the mercury tilt switches, laid out as a folded cube. The lettered circles represent a tilt switch. The cube diagram in the lower right shows the orientation of the eight switches. In this configuration, only one pair of switches will be on in any of the six face orientations on the cube. The switches tilt 45 degrees in all three axes. There are three circuits: switches E, B, H and C; switches A and D; and switches G and F. Each pair of switches corresponds to a separate cube face (orientation). Because it takes two switches to close the circuit, only one pair of switches will return a finite resistance value. The computer knows which side is up based on the resistance value of the circuit, because each pair of switches has a different resistance value in the circuit. Because this wiring diagram is configured around a cube, it only identifies six discrete orientations. To measure continuous orientation values would require other techniques, for example using three gyroscopes to measure pitch, yaw and roll.

► Figure 17: Wiring diagram for mercury tilt switches.



3.3. PROGRAMMING FOR THE PHYSICAL MODEL INTERFACE

There are two programming components for FlexM: the physical model interface on the Handyboard and the graphics application on the host computer. The physical model interface collects the topology and geometry information and transmits them to the graphics application to render the digital model in real-time.

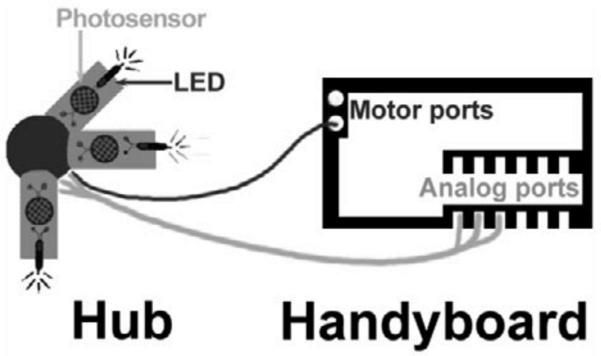
The goal of the programming for the physical model interface is to identify the model's topology and geometry. To illustrate the programming, we will walk though a simple example. First, we explain how the flexible hubs in the physical model connect electronically to the Handyboard. Then, we explain the Interactive-C program in the Handyboard.

3.3.1 Programming to identify the model's topology

In Figure 18, the three LEDs are wired together and connect to a single motor port on the Handyboard, which turns the lights on and off. The three photosensors, one for each socket, are wired separately and connect to separate analog ports, which read the resistance value from the sensors. This arrangement makes it possible to track the socket to the analog port. Likewise, the motor port maps to the hub. With this setup, the number of available motor and analog ports limits the number of hubs.

In order to identify the topology the program creates a table of values to express the connections: hub A connects to hub B at socket X. This is achieved by:

1. Turn on the light of the Sending Hub.
2. Poll all other hubs as Receiving Hubs and record the connections in the table.
3. Repeat steps 1-2 for all the other hubs.
4. Send the table of hub connections to the host computer.



◀ Figure 18: Three LEDs connect to a single motor port on the Handyboard. Each photosensor connects to its own analog port.

The Interactive-C program starts by turning on the LEDs in the first hub (hub 1). The program calls this hub the "sending hub" because it is the hub sending the light to the attached hubs.

While the LEDs in the first hub remain on, the program polls the photosensors on the next hub (hub 2) via the Handyboard's analog port for a resistance value. The program calls hub 2 the "receiving hub". The resistance values range from zero (greatest amount of light) to 255 (no light). The photosensors return a zero resistance value when the high intensity LEDs shine at them. The program cycles through the photosensors in the hub's sockets looking for zero values. When it encounters a photosensor value of zero, the program stops polling the other sockets on that hub, because two hubs can only connect through a single strut. The program records the connection information into a table. After checking all sockets on hub 2, it polls the photosensor values on hub 3, and all remaining hubs in the same fashion.

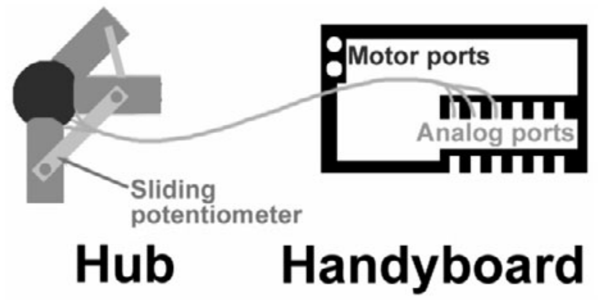
The program turns off the LEDs in hub 1 by setting the motor port value to zero. It goes to the next hub (hub 2) making it the "sending hub." All of the other hubs become "receiving hubs." The program searches through the receiving hubs for zero photosensor values, and records the connection information. The program loops through all the other hubs as "sending hubs" compile the topology information. Finally, the Interactive-C program sends the table of hub connections to the host computer via a serial port.

3.3.2 Programming for finding the model's geometry

The first part of identifying the geometry is measuring the angles at the hub. Although each socket is bound by two hinges, the socket angle measurement at the hubs is only one angle. The Interactive-C program has a table that correlates the sockets to the hinge: socket 1 relates to hinge A;

socket 2 relates to hinge B; socket 3 relates to hinge C; etc.

These angles are derived from sliding potentiometers or rotational potentiometers. Figure 19 shows the flexible hubs with the sliding potentiometers. Each potentiometer connects to the Handyboard through a separate analog port. The potentiometers return a resistance value between zero (smallest hinge angle) to around 180 (hinge angle of 180 degrees). The Interactive-C program converts the resistance value to their corresponding angle in the range of zero to 180 degrees. Because each hinge maps to a specific analog port, the Interactive-C program knows the angle of each hinge.



► Figure 19: Each sliding potentiometer connects to its own analog port on the Handyboard.

After the program sends the host computer the topology information (section 3.3.1), it sends the angle information in the format: socket x has angle A. The program steps are:

1. Send a zero value to signal the beginning of the list.
2. Send the angle information for all the hinges.
3. Repeat step one, to signal a new list.
4. Compare the old angle with the new angle for each hinge. If the angle has changed noticeably, send the new angle values.
5. Repeat steps 3 and 4.

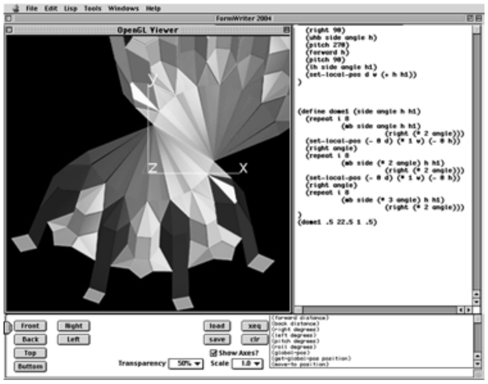
Step 4 reduces the amount of information transmitted to the host computer. Only changed angles are transmitted.

3.3.3 Graphics Application Programming

The graphics application interfacing with the FlexM model is FormWriter [2] a Lisp application for 3D geometry (Figure 20). FormWriter is a simple language for novice programmers to generate 3D geometry with turtle-geometry based commands. One could also use the programming language built in to a modeling application such as Maya's MEL or ArchiCAD's GDL.

4. DISCUSSION AND FUTURE WORK

Our project FlexM aims to bridge the gap between the tactile interaction of play and the 3D graphics application through a physical, flexible model. The FlexM hub and strut construction kit is the modeling interface between the designer and the computer. Physical model building is an important part of the traditional architectural design process [16] and the physical model has a significant meaning for architects [17]. The haptic feedback of the material and the kinesthetic sense of physical manipulation help architects understand and explore design possibilities.



```
  (right 90)
  (uhb side angle h)
  (pitch 270)
  (forward h)
  (pitch 90)
  (lh side angle h1)
  (set-local-pos d w (+ h h1))
)


(define dome1 (side angle h h1)
  (repeat i 8
            (mb side angle h h1)
                  (right (* 2 angle)))
  (set-local-pos (- 0 d) (* 1 w) (- 0 h))
  (right angle)
  (repeat i 8
            (mb side (* 2 angle) h h1)
                  (right (* 2 angle)))
  (set-local-pos (- 0 d) (* 1 w) (- 0 h))
  (right angle)
  (repeat i 8
            (mb side (* 3 angle) h h1)
                  (right (* 2 angle)))
)
(dome1 .5 22.5 1 .5)
```

```
(forward distance)
(back distance)
(right degrees)
(left degrees)
(pitch degrees)
(roll degrees)
(global-pos)
(get-global-pos position)
(move-to position)
```

◀ Figure 20: FormWriter graphics program.

As the past twenty years of papers in this area demonstrates, CAD and modeling software users have been employing these applications for design exploration (in some cases overcoming significant obstacles in the design of the software itself). Yet until the past few years in which rapid prototyping and fabrication technologies have entered the mix, the work has remained on screen, at best two-dimensional projections of three-dimensional space. FlexM shows the possibility of building computationally enhanced physical tools for design exploration. Using a construction kit a designer would build physical models, which in turn would generate a digital model. The digital model has some advantages over the physical one, for example, it can be used to test design scenarios using simulations to predict daylighting, structural stability, acoustics, solar heat gain, and other design performance criteria. A digital format also affords the ease of generating design alternatives.

4.1. FUTURE WORK & POTENTIAL CONSEQUENCES

We have built several prototypes. Each design iteration focused on a separate aspect. This section outlines how we will integrate the parts to build the complete kit.

The flexible hub will have three or more articulating arms, one for each socket connection. Each socket will have a high intensity LED and photosensors for identifying the model's topology. The sockets will attach to two hinges as shown in Figure 15. Rotational potentiometers serve as mechanical hinges and measure hinge angle.

The goal of future prototypes is to reduce production time and minimize the component size for ease of use. Past prototypes have been hand crafted with basswood, but the future hub will be fabricated out of plastic through rapid prototyping to reduce assembly time. Due to the wiring and the sliding potentiometers, the prototype appears cumbersome-more rigid than flexible. Future designs will incorporate a less bulky rotational potentiometer. Instead of dealing with a tangle of wires tethered to the Handyboard microprocessor and being limited by the number of analog and motor (pulse-width modulated power) ports provided by the Handyboard, we would explore using a smaller Basic Stamp microprocessor for each hub. Communication among hubs will be through radio frequency in place of the wiring to the Handyboard.

With FlexM, building a digital model with physical components can be easy and intuitive. Its struts can come in a range of lengths and materials to represent wood, steel, glass, etc. Current hub has interconnected hinges that serve as articulated arms at the end of which are sockets. Future work could include modifications of the hub and struts to accommodate different sizes and numbers of sockets, or different shapes of struts (triangle vs. square rod). Furthermore, the flexible hinges allow study and exploration in geometrical transformations.

The FlexM model could be applied to many domains. Using FlexM as an interface to a structural engineering program, an engineer investigating and manipulating a form could receive the benefit of analysis from the simulation on the screen. Designers could explore rigidity and structural integrity for curvilinear or non-orthogonal structures, such as those cladding systems described in the book "Twist&Build" [18]. It could be useful for medical applications and education. For example, one could map a FlexM model to a molecular structure to explore protein folding. To consider a wider perspective, FlexM could have implications for the construction industry, for integrative systems, and responsive buildings. An example could be using FlexM to interface with real time building performance simulation such as computational fluid dynamic analysis [19]. It could also interface with engineering applications that add the benefit of feedback and analysis of the physical model. Or it could become a control device that would direct the construction of self-assembling building components similar to the Espresso Blocks [20].

4.2. CONCLUSION

In conclusion, the FlexM construction kit follows in the toy-like spirit of other hub and strut construction kits. We argue for the necessity of physical models as part of the design process. As a result, FlexM is a tool with tangible interface that renders a digital version of the physical model in real-time. The translation of topological and geometric data turned out to be harder than we initially imagined. However, our research and exploration demonstrate the possibility to synchronously create an analogous digital model from the physical model building. Instead of moving the mouse in a two-dimensional surface to input coordinates, or shifting the joystick to control fly-throughs in a virtual space, we have taken "direct manipulation" to mean constructing and manipulating physical model components. The project is not yet at a point where the model can be used effectively for design. We have started a process that may eventually be expanded into a working system that could be useful. We plan to work toward a robust system for application and user testing. We hope more people join us to pursue this line of research of translating digital information from a physical construction system.

Acknowledgements

We thank Michael Philetus Weller for his enthusiasm and technical help. The optical signaling method for determining model topology was suggested by Don C. Witte. This research was supported in part by the National Science Foundation under Grants CCLI DUE-0127579 and ITR-0326054. The views and findings contained in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Portions of this paper were presented in the G-CAD Symposium at Carnegie Mellon University, or included in the Master Thesis by the first author [21]. The paper has benefited from a thorough and thoughtful critique by the anonymous IJAC reviewers, whom we thank.

Photo credits:

Figure 1, left. Tinker Toys *http://www.lhs.berkeley.edu/pfs/images/TinkerToys.jpg*

Figure 1, middle. K'nex *http://www.asmilan.org/teachers/mwhaley/hbody4.jpg*

Figure 1, *right. Zometool http://abstract.cs.washington.edu/~dougz/cgi-bin/photo.pl?*

References

1. Martin, F., *Robotic Explorations: A Hands-On Introduction to Engineering,* Upper Saddle River, New Jersey: Prentice-Hall, 2001.
2. Gross, M., FormWriter: A Little Programming Language for Generating Three Dimensional Form Algorithmically, in *CAAD Futures*: Eindhoven, NL, 2001, 577-588.
3. Watson, B., *The man who changed how boys and toys were made*. New York: Penguin, 2002.
4. Hoberman, C., Faltstrukturen für temporäre Gebäude (Temporary Unfolding Structures), in *Detail*, 1996. 36(8): p. 1184-1185.
5. Gorbet, M. and M. Orth, Triangles: Design of a Physical/Digital Construction Kit, in *Designing Interactive Systems (DIS-97)*. ACM, 1997, 125-128.
6. Negroponte, N., *The architecture machine; toward a more human environment.* Cambridge, Mass.: MIT Press, 1970.
7. Anderson, D., Frankel, J., Marks, J., Agarwala, A., Beardsley, P., Hodgins, J., Leigh, D., Ryall, K., Sullivan, E., Yedidia, J., Tangible Interaction + Graphical Interpretation: A New Approach to 3D Modeling, in *SIGGRAPH 2000*. ACM, 2000, 393-402.
8. Aish, R., *3D Input for CAAD Systems*. Computer-Aided Design, 1979. 11(2): p. 66-70.
9. Frazer, J., Frazer, J., and Frazer, P., *New developments in intelligent modelling*, in

Computer Graphics 81. Online Publications, 1981, 139-154.

10. Dewey, D. and A.T. Patera, *A. Geometry-defining processors for partial differential equations., in Special Purpose Computers*, B. Alder, Editor. Academic Press, 1988, 67-96.
11. Esposito, C., Paley, W. B., and Ong, J., Of mice and monkeys: A specialized input device for virtual body animation, in *Symposium on Interactive 3D Graphics*: Monterrey, 1995, 109-114.
12. Raffle, H., Parkes, A. and Ishii, H., Topobo: A constructive assembly system with kinetic memory, in *Human Factors in Computing (CHI) '04*. ACM, 2004.
13. Greenberg, S. and C. Fitchett, Phidegts: Easy development of physical interfaces through physical widgets, in *UIST 2001 Symposium on User Interface Software and Technology*. ACM, 2001, 209-218.
14. Lertsithichai, S. and M. Seegmiller, CUBIK: A bi-directional tangible modeling interface, in *Human Factors in Computing Systems, CHI 2002*, 2002, 756-757.
15. Wrensch, T. and M. Eisenberg, The programmable hinge: toward computationally enhanced crafts, in *UIST 1998*. ACM, 1998, 89-96.
16. Smith, A., *Architectural Model as Machine: A new view of models from antiquity to the present,*: Architectural Press, Oxford, 2004.
17. Moon, K., *Modeling Messages: The Architect and the Model*: Monacelli Press, 2005.
18. Vollers, K., *Twist&Build: creating non-orthogonal architecture*. Rotterdam, The Netherlands: 010 Publishers, 2001.
19. Malkawi, A.M. and R.S. Srinivasan, Interfacing with the real space and its performance, in International Journal of Architectural Computing, 2005. 3(1): p. 43-56.
20. Weller, M.P. Espresso Blocks, self-assembling building blocks, Master Thesis, University of Washington, Seattle, Washington.2003.
21. Eng, M. *FlexM: A Flexible Design Construction Toy*, Master Thesis, University of Washington, Seattle, Washington.2004.

Markus Eng
P.O. Box 159
Seattle, WA 98111-0159

markuse@myuw.net

Ken Camarata
KDF Architecture
1310 N. 16th Avenue
Yakima, WA 98902-1354

ken.camarata@gmail.com

Ellen Yi-Luen Do
PhD Program, College of Architecture
Georgia Institute of Technology
Atlanta, GA 30332-0155

ellendo@gatech.edu

Mark D Gross
CoDe Lab, Margaret Morrison 407
Carnegie Mellon University
Pittsburgh, PA 15213

mdgross@cmu.edu