Towards "Skinnable" Physical Objects

First Author
AuthorCo, Inc.
123 Author Ave.
Authortown, PA 54321 USA
author1@anotherco.com

Second Author
AuthorCo, Inc.
123 Author Ave.
Authortown, PA 54321 USA
author2@anotherco.com

Third Author
AuthorCo, Inc.
123 Author Ave.
Authortown, PA 54321 USA
author3@anotherco.com

Abstract

Author Keywords

Optional section to be included in your final version.

ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous. See: `http://www.acm.org/about/class/1998/` for help using the ACM Classification system. Optional section to be included in your final version, but strongly encouraged.

Intro from proposal

The "Internet of Things" (IoT) is becoming an increasingly popular topic for commercial products. Devices such as the Nest learning thermostat, the Twine home automation and monitoring device, and the Kwikset Kēvo Bluetooth lock are popular consumer IoT products. At the core, each of these devices follow the same basic hardware and software pattern: sensors (such as a temperature sensor), actuators (such as LEDs or the motor to turn the lock), a radio (Bluetooth or WiFi), a processor (to control the device) and cloud-based software services. The basic hardware package of processor, radio, sensors and actuators is flexible enough to do many things [8]; however, IoT devices are limited to the form and functionality that the manufacturer has designed.

Another current area of intense interest is digital fabrication, which includes 3D printers, laser cutters, milling machines, and other similar equipment. Although these devices have only recently evolved from industrial machines into hobbyist equipment, they hold enormous promise for revolutionizing manufacturing. Writing about 3D printing in his popular-press book, Cornell professor Hod Lipson states [7],

> *3D printing gives regular people powerful new tools of design and production. People with modest bank accounts will acquire the same design and manufacturing power that was once the private reserve of professional designers and big manufacturing companies. In a 3D printed future world, people will make what they need, when and where they need it.*

There is an opportunity to combine these two new technologies to allow individuals to create their own personalized IoT devices. By making it easy to embed standardized hardware modules within fabricated objects and to add customized behaviors, users will be able to create their own, personal IoT devices. For example, a user might download a model of a bird and embed a thermometer IoT device inside it to track the temperature in her living room, and then a toy train to track the temperature in her son's room. Both of these devices do the same thing, but take very different forms. Another example is printing a customized wearable bracelet; one user might add an accelerometer to track steps while another might add a vibration motor to give alerts.

We borrow the idea of changing the form of an object while maintaining the same functionality from software; many applications support "skins" or themes that do just this. Rather than enhancing the usability of the software, usually skins and themes serve an aesthetic or ludic purpose. The challenge addressed in this research is that, unlike software, skinnable physical objects must meet requirements around manufacturability, size, cost, and ease of assembly.

As a motivating example, consider Jenny, a baseball fan. Her favorite team, the San Francisco Giants, has been in the news lately due to changes in team membership. She has the idea to print out the team's logo (a "S" and "F" overlaid on each other) on her 3D printer and place it on her kitchen counter; whenever breaking news about the Giants is posted on the ESPN sports news site, she wants the team's fight song to play briefly to alert her. She also wants a button on the logo that will post the message "I love the Giants!" to Twitter.

While this example is straightforward, realizing the idea is a difficult one for a non-technical individual. To start from scratch, Jenny must use a 3D modeling program to create the logo, ensure that it can be successfully printed, conceive of an electronic circuit that connects to the Internet and fits inside her logo, and fabricate and assemble these parts. Even if such skills were within Jenny's reach, the process would be quite time consuming and perhaps not worth the trouble.

The capabilities currently exist for Jenny to download a model of an object to print via online services such as Thingiverse[1]; to assemble pre-specified modular electronic circuits such as LittleBits[2]; and to configure the electronics to react to online events via a service like If This Then That[3] (IFTTT). However, to combine the circuit with the downloaded object in itself requires a non-trivial level

[1] `http://thingiverse.com`
[2] `http://littlebits.cc`
[3] `http://ifttt.com`

of technical skill; to further decompose the circuit, put parts of it in desired locations, fit the rest of the modules in where they will best fit, and to link those components together via wires, demands a high level of patience, experience, and technical know-how.

In this paper, we describe the initial steps for a system to enable easy end-user skinning for physical objects. Our system takes as input a modular circuit description, a 3D model file, and a specification of where certain of the modules (e.g., a button) should be placed relative to the model. It outputs a new model file modified to fit the components and a modified circuit layout showing where wires must be added to accomodate the physical structure.

Related Work

Tanenbaum et al. identified *playfulness*, *utility*, and *expressiveness* as central themes for individuals engaged in making things [16]. These themes will come into play more and more as digital fabrication technology reaches the home, enabling people to design and customize their own objects.

Long before digital fabrication equipment was commonly available, software allowed users to pursue playfulness, utility, and expressiveness by customizing the look and feel of many programs. For example, the famous Winamp audio player featured skins which could change the look and feel of the application, and the Windows operating system includes themes which allow the user to change color schemes and background images.

More recently, a number of research projects have focused on allowing users to design personalized objects for fabrication: chairs [13], toys [1, 19], and even visualizations [5, 15]. However, these examples all produce self-contained objects, not meant to work with other objects or incorporate extra functionality.

One of the most common ways to help non-experts design objects is to remove the necessity for users to learn complex computer-aided design (CAD) tools. Some research has employed sketch-based interfaces as a simpler method for users to create computer models, for example to sketch single-perspective views of three-dimensional polyhedral objects, which the computer then interprets and turns into a full 3D model [6, 9, 11, 12], or to work with two-dimensional sketches for cutting with a laser cutter or other 2D fabrication machine [2, 10, 11, 13]. These systems are generally intended for the fabrication of new objects, rather than to modify or work with existing ones.

Lots of focus on physical prototyping ([4], [3]); not as much on integrating electronics into actual things you might want to use.

Several systems exist that allow users to work with existing electronic components to embed them in fabricated forms. .NET Gadgeteer [17], a modular electronics system, provides 3D models of its various modules that can be used with computer-aided design (CAD) software to incorporate them into different designs. Weichel et al.'s Enclosed system [18] builds on this work with a simplified CAD system that uses the Gadgeteer modules as first-class components. NatCut [14] allows users to design a laser-cut electronics enclosure in 2D on an interactive tabletop, and to place physical Gadgeteer modules to indicate where they should go in the finished enclosure. These systems focus more on prototyping rather than creating a finished, personalized item for individual use, and require users to manually specify where each electronic module should be placed.

System Description

Rather than helping users with technical skills prototype various designs for interactive objects, the ultimate goal of our system is to allow a non-expert user to create finished objects by downloading a modular circuit description and "skinning" it with a (possibly unrelated) downloaded 3D model. We envision a user loading a machine-readable modular circuit diagram, an associated IFTTT recipe, and a 3D model into our system, interactively specifiying the locations on the model where interactive components should be placed, and receiving a new, fabricatible model file modified to support embedding the components, as well as instructions on how to assemble the new device.

explain IFFT recipes somewhere

In this paper, we present our initial work towards this goal. We address the challenge of placing modules inside a model without growing the model to extreme sizes. This is a variation on the packing problem—trying to pack items into a space as efficiently as possible. Our application introduces two additional constraints to the packing problem: modules that the user places at a specific position relative to the model (e.g., an LED for the eye of a bird), and that the modules must remain connected in a particular order to maintain the functional circuit.

For our initial implementation, we use LittleBits modules. LittleBits is a commercial electronics experimentation platform consisting of single-purpose electronics modules that attach to each other via magnets. LittleBits modules are ideal for people inexperienced in electronics because they cannot be put together incorrectly; that is, any physically-possible configuration, while not necessarily functional in the expected way, will not cause damage. There is a thriving community of LittleBits experimenters who post circuit diagrams to the company's website. The two drawbacks of LittleBits are size and expense: the average module is about 1x.8in (26x21mm), and prices range from US $8 (a button) to $60 (the Internet-connected "CloudBit"). However, it is not unreasonable to expect drops in both size and price, and our research will still apply.

In order to be able to arrange modules inside an arbitrary model, we assume that the modules can be connected directly together or via wire; even in the case of very small modules, wires will be desirable to locate different interaction elements at different places. To minimize both complexity and cost, our algorithm minimizes the number of wires necessary to assemble the circuit, balanced with the amount of scaling necessary to fit the modules in the given arrangement.

The preliminary implementation described in this paper requires hand-coding of the circuit and manual placement of the fixed-location circuit modules. The software then generates configurations C for the remaining modules, with the objective of minimizing the function

$$F(C) = wS + (1 - w)W$$

where S is the scale of the completed object, W is the number of extra wires required, and $w \in [0, 1]$ adjusts the weight between the two factors.

Our system represents the circuit as an undirected graph, with each module a node in the graph and the electrical connections between modules represented as edges. A property on each edge indicates whether it represents a direct connection or a connection via a wire, and a property on each node indicates whether the node is to be at

a fixed location or is free to mov
ical layout of the circuit, which
the user changes which modules are fixed.

Stan and Xiaojie fill in this part

Starting with a graph in which all edges are marked as wires, we mark edges as direct connections, generating all possible 2^n permutations. Each permutation has an associated W-cost which is the number of wires. We then sort the list by W-cost, and starting with the least expensive, attempt to place the modules inside the 3D model.

Our preliminary implementation operates in 2D, placing modules into a 2D outline of the model. At each attempted placement, it tests to ensure that the module does not intersect the wall of the model; if it does, the operation is aborted and the next graph in the list is tested. For the purposes of placement, the system treats directly-connected modules as single units. The software first places the units with fixed modules. Assuming these fit within the borders of the model, it then. . . .

Features:

- Represent the circuit as an undirected graph (currently input by hand in code).
- 2D packing to find the best arrangement.
- Minimize cost function of size, number of wires; something like $c = wS + (1 - w)W$ where S is size, W is number of wires, and w is relative weight between the two.
- Specify location of fixed modules (in code).
- Search all possible logical arrangments of wires/not wires.
- Output object for printing.

References

[1] Blauvelt, G., and Eisenberg, M. Computer aided design of mechanical automata: Engineering education for children. *International Conference on Education and Technology* (2006).

[2] Buxton, B. *Sketching User Experiences: Getting the Design Right and the Right Design*. Getting the Design Right and the Right Design. Morgan Kaufmann, July 2010.

[3] Hartmann, B., Klemmer, S., Bernstein, M., Abdulla, L., Burr, B., Robinson-Mosher, A., and Gee, J. Reflective physical prototyping through integrated design, test, and analysis. *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology* (Oct. 2006).

[4] Hodges, S., Taylor, S., Villar, N., Scott, J., and Helmes, J. Exploring physical prototyping techniques for functional devices using .NET gadgeteer. In *the 7th International Conference*, ACM Press (New York, New York, USA, 2013), 271–274.

[5] Khot, R. A., Hjorth, L., and Mueller, F. F. Understanding physical activity through 3D printed material artifacts. In *CHI'14: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press (New York, New York, USA, 2014), 3835–3844.

[6] Lau, M., Saul, G., Mitani, J., and Igarashi, T. Modeling-in-context: user design of complementary objects with a single photo. *EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling* (2010).

[7] Lipson, H., and Kurman, M. *Fabricated*. The New World of 3D Printing. John Wiley & Sons, Feb. 2013.

[8] Lyons, K., Nguyen, D. H., Seko, S., White, S., Ashbrook, D., and Profita, H. BitWear: a platform for small, connected, interactive devices. In *UIST '13 Adjunct: Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology*, ACM (Oct. 2013), 73–74.

[9] Masry, M., Kang, D., and Lipson, H. A freehand sketching interface for progressive construction of 3D objects. *Computers & Graphics 29*, 4 (Aug. 2005), 563–575.

[10] Mori, Y., Igarashi, T., and Mori, Y. Plushie: an interactive design system for plush toys. *ACM Transactions on Graphics (TOG) 26*, 3 (July 2007), 45.

[11] Oh, Y., Johnson, G., Gross, M., and Do, E. Y.-L. The Designosaur and the Furniture Factory. In *Design Computing and Cognition*. Springer Netherlands, Dordrecht, Jan. 2006, 123–140.

[12] Plumed, R., Pedro Company, Varley, P. A. C., and Martin, R. R. From sketches to CAM models: perceiving pockets and steps in single-view wireframe sketches of polyhedral shapes. In *UbiComp '13 Adjunct: Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, ACM Request Permissions (New York, New York, USA, Sept. 2013), 951–958.

[13] Saul, G., Lau, M., Mitani, J., and Igarashi, T. SketchChair: an all-in-one chair design system for end users. In *TEI '11: Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, ACM Request Permissions (New York, New York, USA, Jan. 2011), 73–80.

[14] Schneegass, S., Sahami Shirazi, A., Döring, T., Schmid, D., and Schmidt, A. NatCut: An Interactive Tangible Editor for Physical Object Fabrication. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, ACM Press (New York, New York, USA, 2014), 1441–1446.

[15] Swaminathan, S., Shi, C., Jansen, Y., Dragicevic, P., Oehlberg, L. A., and Fekete, J.-D. Supporting the design and fabrication of physical visualizations. In *CHI'14: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press (New York, New York, USA, 2014), 3845–3854.

[16] Tanenbaum, J. G., Williams, A. M., Desjardins, A., and Tanenbaum, K. Democratizing technology: pleasure, utility and expressiveness in DIY and maker practice. In *CHI '13: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Request Permissions (Apr. 2013).

[17] Villar, N., Scott, J., Hodges, S., Hammil, K., and Miller, C. .NET Gadgeteer: A Platform for Custom Devices. In *Pervasive Computing*. Springer Berlin Heidelberg, Jan. 2012, 216–233.

[18] Weichel, C., Lau, M., and Gellersen, H. Enclosed: a component-centric interface for designing prototype enclosures. In *TEI '13: Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, ACM Request Permissions (Feb. 2013).

[19] Zhu, L., Xu, W., Snyder, J., Liu, Y., Wang, G., and Guo, B. *Motion-guided mechanical toy modeling*. ACM Trans. Graph., 2012.