Calm Automaton: A DIY Toolkit for Ambient Displays

Minjoo Cho
MyDesign Lab, Department of
Industrial Design, KAIST
291 Daehak-ro, Yuseong-gu, Daejeon
34141, Republic of Korea
koopoo87@kaist.ac.kr

Daniel Saakes
MyDesign Lab, Department of
Industrial Design, KAIST
291 Daehak-ro, Yuseong-gu, Daejeon
34141, Republic of Korea

Authors version. To be published in CHI'17 Interactivity
https://doi.org/10.1145/3027063.3052968

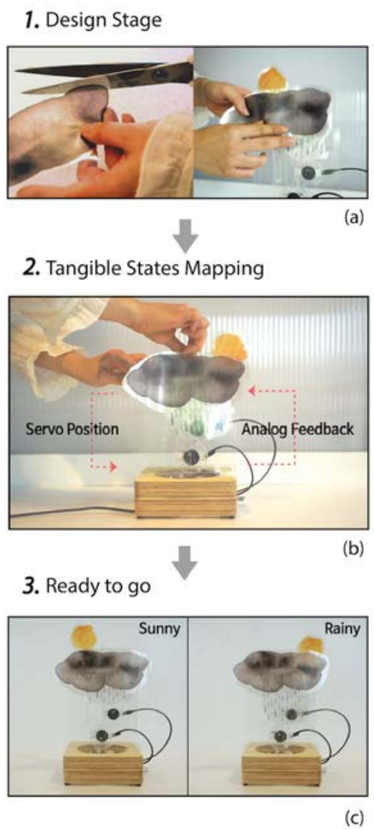

Figure 1: Design and mapping process with Calm Automaton (a) Cut and attach images to the motion modules, (b) Push or pull modules to map data states from the internet (c) Place at surrounding to make it move automatically

Abstract

The abundance of information technology in today's society results in "Alert Fatigue" due to the overwhelming number of alarms and notifications that attempt to grab our attention.

We introduce Calm Automaton, a customizable and programmable physical display that gently visualizes abstract data in a pleasing and meaningful way, without attracting attention. We extend the concept of calm technology with a DIY toolkit to make information and notifications comfortable, personal, and embedded in the periphery. We describe the design and implementation of the motion modules that make the automaton and report on the experience of people using these displays.

Author Keywords

Ambient display; calm technology; DIY toolkit; automata, tangible programming, IoT programming

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous;

Introduction

Mark Weiser and John Seely Brown [2] introduced the concept of "Calm Technology" 20 years ago to make technology both inform and encalm, by situating it in the "periphery" of our attention, or embedded in the environment. Since then, use of information technology in people's lives has increased dramatically thus making the topic of "Alert Fatigue" more relevant today.

To this end, several commercial and research products have been mapping abstract data into tangible or ambient displays. For instance, Ambient Orbs let users map stock market indices or weather forecasts to the color of a glowing sphere. Physikit [1] is a physical display that maps environment data, such as air quality, to a cube with an actuator that controls rotation or

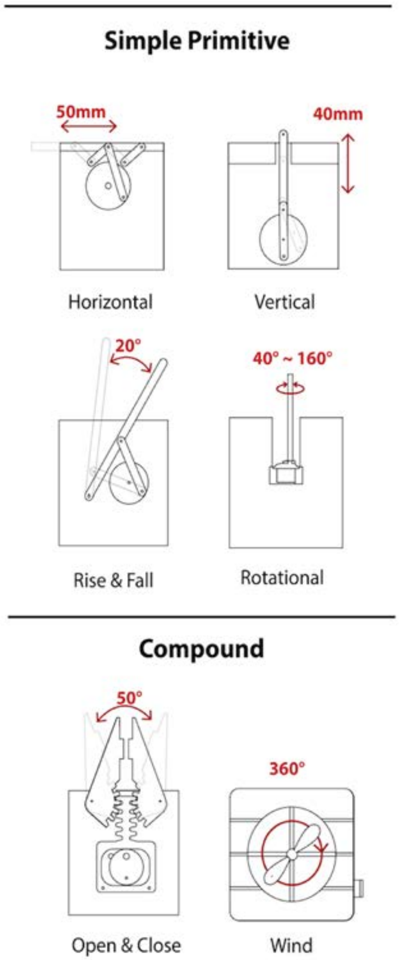

Figure 3: Motion Modules in Primitive Motions (horizontal, vertical, rotation) and Compound Motions (open & close)

airflow. Although these devices let users map data to actuators, they do not support the personalization of the display, nor let the user visualize personal data.

We were inspired by automatons, self-operating mechanics that move according to a predetermined sequence of operations, to allow for rich and personalized displays. The idea of "Calm Automatons" is to let the mechanism move using states and transitions based on information mapped by the user. For that, we build upon the variety of (robotic) construction toolkits such as RoBlocks, Cublets and Little Bits to let people build their own personal tangible display. With a teach-in paradigm, users map states to positions using direct manipulation and physical programming to design and program motion in robotic toolkits such as Topobo [4].

In a formative workshop with users we gathered ideas for "what to display?" and we discovered the need for visualizing personal data. For instance, an approaching deadline, or whether the boss is in his or her office. Hence, we build upon the related work in IoT to enable users to easily synchronize web data with peripheral objects such as IFTTT (If This Then That) [3], e.g. making use of existing API's to access calendar data or location services.

Calm Automaton Design

We designed Calm Automaton to fit in the living room and to be placed on a table or shelf. Shown in Figure 2, it consists of a small wooden block with slits to fit a stack of two dimensional motion modules.

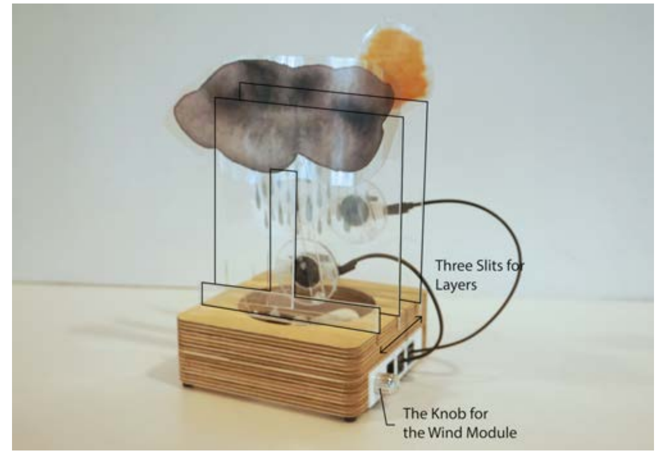

Figure 2: The Automaton consists of a wooden base that contains a Raspberry Pi wirelessly connected to the internet. Three slits on the top are for multiple "motion" layers.

Each of the motion modules is defined by a single primitive motion: simple linear ("horizontal" and "vertical"), "rise and fall", and "rotation": (around the vertical axis and on the modules plane) and "open and close", "wind" as compound motions. A "wind" module is embedded in the wooden body of the toolkit and consists of a fan for animating fabric and lightweight materials. (Figure 2 (b)).

The modules are made of thin transparent acrylic sheet material and fabricated with a laser cutter. A servomotor with analog position feedback is embedded in each module. The servo drives the module's motion, but the user can also position a module by hand to program a state. The modules have sufficient surface area for users to attach materials, such as cut-out

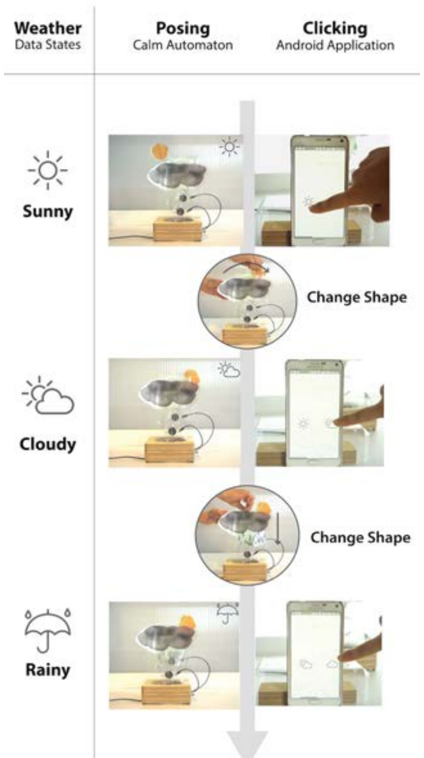

Figure 3: Users customize the animation by mapping states. A state is selected on a mobile phone application and programmed using a teach-in paradigm by manually moving the motion modules into position.

graphics, to both the static parts as well as the dynamic parts. Through layering multiple modules in the slits, a complex three-dimensional diorama can be easily created.

State mapping is performed through posing the modules by hand and a mobile application written in Processing for Android. On the smartphone, an information source is selected. For example, a weather forecast has the states "sun", "rain", and "cloudy", and a deadline has several states as defined by a countdown until the date/time is reached. Then, one by one, a state is selected on the phone and the modules are posed according to the state. Switching between states lets users preview transitions. Once states are mapped to poses, the automaton runs automatically and the software fetches information from the internet and interpolates the states accordingly.

Prototype

The Calm Automaton is implemented in Processing and runs on a Raspberry Pi v2. An Arduino (teensy 3.1) is connected through USB and drives the servo motors and reads the module positions on its analog inputs. The Raspberry Pi is connected to the Internet through a Wi-Fi adapter. In the prototype we provided weather (weather forecast, temperature, humidity, wind speed), finance (exchange rates) and deadline event synchronized with Google calendar API for customization.

The Processing for Android application on the smartphone provides the user interface and communicates with the Raspberry Pi using the OSC data protocol. The phone communicates the selected information source and sets the automaton in a specific mode: a "play mode" for use, and a "mapping mode" for each of the states. Entering the mapping mode, the Automaton transitions the motion modules to match the previous recorded state. Then, the Automaton listens for positional changes in the motion modules (made by the user) and stores new positions as an updated mapping for the state. When the user exits the mapping mode or application, the Calm Automaton turns back to auto-run mode , regularly fetches information from the Internet, and slowly actuates the servos accordingly.

User Study in the Wild

We made three Calm Automaton prototypes to perform a preliminary user study in the wild. Six students were recruited (3 males and 3 females, average age of 23.83 (SD = 3.49)). Participants made a personal Calm Automaton and used it for one week in either their dormitory or on their desk in their laboratory.

We sensitized participants with a probe package and diary to let them think about what kind of information is important for them. In a small design session, we first showed the motion modules and explained the concept. Then we gave them tools (printer, pen, scissors, glue) and paper, and asked them to make a personalized information display. Figure 4 illustrates automata examples created by the participants. They expressed the data by utilizing various personal metaphors. Some visualized the deadline event display with popular cartoon characters changing shape as the deadline approaches. One participant made the hair of a male character look like a bird, and let it fly away as the wind speed got stronger. Most participants used all three motion layers and mapped three or more states.

Figure 4: Automata examples designed by participants of the user study.

Participants experienced no difficulties in the data mapping process. However, they suggested several improvements. In addition to the dynamic layers we found the need for static layers. Some participants suggested smaller modules, so that they could position multiple modules in a slit. Other participants wanted to visualize data from multiple sources, such as notifying good weather conditions for exercising together with their exercise frequency.

Once the automaton was customized and programmed, participants took it home or to their office and used it for a week. We asked them to self-report the time when they interacted with the automaton.

During the week, all participants interacted with the Automaton and several participants updated and refined the mapping. The diary revealed that users felt close to their automata as frequently described during the post-interview session with the word “interact” or “commune”. We contribute this connectedness to the design process that required to reflect on personal experiences and preferences. In a post-use interview, some participants described their relationship as intimate as it can be used to purposely obscure personal data in shared spaces and make mappings only interpretable by themselves.

A limitation of the current prototype is the noise the servo makes, even when rotating very slowly. Occasionally, in silent situations, this inadvertently brought the Automaton to the center of attention.

Conclusion

We made a first step to make information both personal and available in the periphery. We suggest that in addition to publicly available information, such as weather forecasts, personal data such as calendar events or location based (parking spot free) data are also important information sources to visualize in the periphery.

Decorating and animating the Automatons seem to enable participants to make meaningful and personal mappings compared to existing solutions. An unexpected finding was the use of the Automaton to hide personal or sensitive information from co-workers while in plain sight.

Future automatons need to think of increasing the flexibility of both mapping, for instance multiple information sources, and animation, such as adding more modules per layer.

References

1. Houben, S., Golsteijn, C., Gallacher, S., Johnson, R., Bakker, S., Marquardt, N., … Rogers, Y. (2016). Physikit. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing* Systems (CHI ’16), 1608–1619. http://doi.acm.org/10.1145/2858036.2858059
2. Weiser, M., & Brown, J. S. (December 21, 1995). Designing Calm Technology. Retrieved December 1, 2016 from http://www.ubiq.com/hypertext/weiser/calmtech/calmtech.htm
3. Ur, B., Pak Yong Ho, M., Brawner, S., Lee, J., Mennicken, S., Picard, N., Schulze, D., Littman, M. L. (2016). Trigger-Action Programming in the Wild. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI ’16), 3227–3231. http://doi.acm.org/10.1145/2858036.2858556
4. Raffle, H. S., Parkes, A. J., & Ishii, H. (2004). Topobo. In *Proceedings of the 2004 conference on Human factors in computing systems* (CHI ’04), pp. 647–654. http://doi.acm.org/10.1145/985692.985774