# Blossom: A Handcrafted Open-Source Robot

MICHAEL SUGUITAN and GUY HOFFMAN, Cornell University



Blossom is an open-source social robotics platform responding to three challenges in human-robot interaction (HRI) research: (1) Designing, manufacturing, and programming social robots requires a high level of technical knowledge; (2) social robot designs are fixed in appearance and movement capabilities, making them hard to adapt to a specific application; and (3) the use of rigid mechanisms and hard outer shells limits the robots' expressive capabilities. Addressing these challenges, Blossom aims at three design objectives: accessibility, flexibility, and expressiveness. The robot's mechanism can be quickly assembled and partially extended by end-users. Blossom's appearance is open-ended through handcrafted fabric exteriors created and customized by users. Smooth organic movements are achieved with tensile mechanisms, elastic components, and a soft exterior cover attached loosely to the body. Blossom's smartphone-based gesture generation requires neither programming nor character animation experience, allowing lay users to create their own behaviors. All elements in the design were conceived with a low barrier-of-entry in mind. The result is an accessible and customizable social robot for researchers. This article details the implementation of Blossom's design and demonstrates the platform's potential through four field deployment case studies.

CCS Concepts: • **Human-centered computing → Interaction design**; • **Computer systems organization → Robotics**;

Additional Key Words and Phrases: Robot design, craft, handcrafted, craft robotics, open-source, social robotics, research platform, soft robotics, toolkit, robot toolkit

**ACM Reference format:**
Michael Suguitan and Guy Hoffman. 2019. Blossom: A Handcrafted Open-Source Robot. *ACM Trans. Hum.-Robot Interact.* 8, 1, Article 2 (March 2019), 27 pages.
https://doi.org/10.1145/3310356

## 1 INTRODUCTION

The design of social robots with expressive capabilities is an active area of research in human-robot interaction (HRI). Researchers in HRI have been developing robots with expressive behaviors and collecting empirical evidence for the effects of these behaviors. Some of these robots use facial expressions [11–13, 23, 38] while others express their internal states through bodily gestures [33, 37, 39, 77] or other modalities [9, 57]. More recently, consumer electronics companies have also started to explore expressive social robots as commercial products [7, 34, 51].

Designing and building such a robot, however, requires extensive knowledge and resources in mechanical and electrical engineering. Similarly, designing and implementing the robot's

This work was supported in part by a grant from Google Creative Technologies.
Authors' addresses: M. Suguitan and G. Hoffman, Cornell University, 563 Upson Hall, Cornell University, Ithaca NY 14853, USA; emails: {mjs679, hoffman}@cornell.edu.

2573-9522/2019/03-ART2
https://doi.org/10.1145/3310356

Fig. 1. Three variations of Blossom with different embodiments and accessories. The robot on the left is knitted, and the two robots on the right are crocheted. The two robots on the left display swappable wooden ears and a number of attachable facial features, while the robot on the right features soft silicon arms as appendages.

expressive gestures and behaviors requires professional skills in computer science and 3D character animation. This makes robot building and programming inaccessible to a large swath of users.

This lack of accessibility limits the use of social robots for both researchers and end-users. For example, most researchers in HRI have a choice of one of a handful of programmable social robots, such as the Softbank Robotics NAO or Pepper robots, Philips's iCat, Rethink Robotics's Baxter, or the MyKeepon platform. These robots are subsequently exceedingly prevalent in the HRI literature, e.g., Refs [1, 10, 17, 29, 41, 43, 70, 79]. Each of these robots has a single outward appearance, which is overcome at times with adornments such as hats or other accessories [67]. Still, it is difficult to adapt the robot's appearance to the task at hand, rendering them inflexible with respect to specific applications and personalization.

The majority of social robots are also rigid in a more literal physical sense: Their exteriors are made of hard plastic or metal shells manufactured using additive and subtractive methods such as 3D printing, molding, and milling. These exteriors are fixed to direct or geared drive mechanisms and rigid linkages with fasteners such as bolts and adhesives to form solid connections. This mechanical rigidity restricts the robot's expressiveness and interactive capabilities. Rigid actuation mechanisms make it difficult to achieve smooth, organic movement without complicated software control or trajectory generation. Stiff direct linkage mechanisms also discourage physical interaction due to their hard tactile affordance and fear of damaging internal components.

In this work, we present Blossom, an open-source robotics platform, with the goal of addressing the issues identified above. Blossom is designed to allow researchers and end-users to imagine and build their own robot, enabling more flexible design possibilities in the robot's appearance, structure, and behaviors. This could increase adoption and help diversify HRI research. In addition, Blossom offers a mechanical design with compliant, organic movement in mind, to support expressiveness and interactivity.

Blossom thus attempts to achieve three design objectives: *accessibility*, *flexibility*, and *expressiveness*, implemented through the following design choices:

- The robot can be easily built by lay-users.
- It has a modifiable degree-of-freedom (DoF), but is still predictably expressive.
- It uses a tensile mechanical structure that affords smooth movements and safe interaction.
- Its appearance can be handcrafted with traditional crafts.

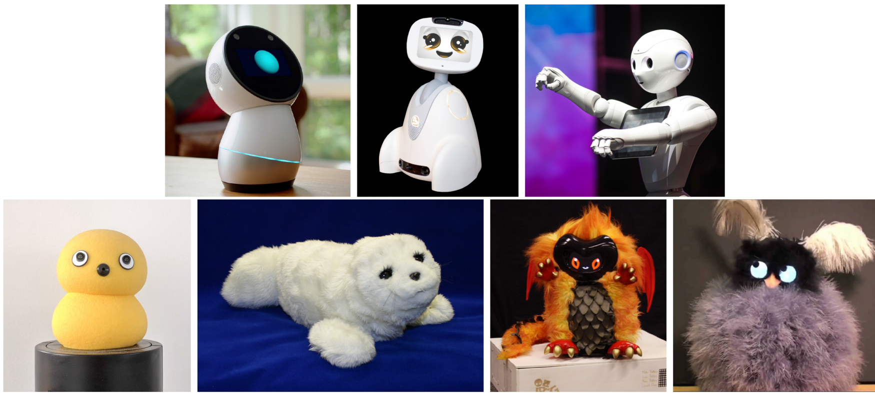

Fig. 2. Jibo, Buddy, and Pepper (top) are examples of social robots with design features related to consumer electronics devices. Keepon, Paro, DragonBot, and Tofu (bottom) exhibit softer and more zoomorphic embodiments.

— Both its mechanism and exterior can be made from readily available low-cost materials.
— New behaviors can be defined without requiring programming or computer animation skills.
— Behaviors are accessible through an open interface suitable for a broad range of applications.

Notably, Blossom is not a robotics kit in the same vein as LEGO Mindstorms$^{TM}$ or Meccano$^{TM}$, which differ in two important ways. First, these kits provide a widely open-ended design space that is not tailored to any specific application. In particular, they are not designed with expressive behavior or social interaction in mind. Second, these kits cater mostly to technically-oriented users and focus on the robot's construction rather than its use. In contrast, Blossom is socially-oriented, while still being easily customized by non-technical users, and focused on the end-user of the robot.

As a use case, we imagine a social science research group with limited engineering expertise but interest in a research question related to HRI. Researchers in this group should be able to quickly build, fashion, and use a Blossom robot and define behaviors specific to their application. Another scenario is that of a lay-user who is uninterested in engineering and programming but wants to build a social robot for their personal use with a particular appearance and set of behaviors.

In this article, we present Blossom's mechanical, electronics, and software implementation and detail the customizable exterior and behavior of the robot. To evaluate the design, we provide four case studies of field deployments where users implemented Blossom robots or interacted with them.

## 2 RELATED WORK

Blossom relates to the existing literature in social robot design, gesture generation, and open-source robotics and HCI construction kits.

### 2.1 Social Robot Design

Aesthetic designs for social robots range from product-like to organic. Jibo, Buddy, and Pepper (Figure 2 top)[1] are examples of robots with features akin to those of consumer electronics devices, such as straight lines, rounded edges, touch screens, and illuminated accents [19, 30, 36, 65]. On the other side of the spectrum are creature-like robots with a soft and compliant skin. Keepon

[1]Jibo, DragonBot, and Tofu images courtesy of MIT; Buddy image courtesy of Blue Frog Robotics; Pepper image courtesy of Shelly Levy-Tzedek; Paro image courtesy of AIST, Japan.

(Figure 2, bottom left, pictured here in its MyKeepon variant) was a pioneering design that provided a non-anthropomorphic flexible shape made of cast silicone [41]. The robot has been used extensively in research with children on the autism spectrum [42], but has also been used in other HRI scenarios such as learning [50], storytelling [49], and attention [2]. Other examples of soft social robot design include Paro, DragonBot, and Tofu (Figure 2, bottom), evoking a more zoomorphic aesthetic [71, 81, 82]. These robots enable the flexible deformation of their exterior, supporting organic-looking expressive effects such as folds, creases, and stretch-and-squash [46]. All of these examples are made of high-end components, materials, and fabrication processes, resulting in relatively expensive systems, although a low-cost variant of the Keepon robot was later developed under the name MyKeepon [16]. Common to all of the above-described robots is that they are available to researchers as fully manufactured designs with a fixed appearance. Some of them are custom one-off designs (such as the original Keepon, DragonBot, and Tofu), and some are commercially available products (such as MyKeepon, Cozmo, and Pepper). In HRI research, personalization has been found to positively affect user perception of a robot [48, 50, 76, 80] even if the robot is not specifically designed for social interaction. Blossom thus extends these prior design works by presenting a compliant robot that enables users to build and program their own robot and to customize its appearance.

The choice of materials also plays an important role in robot design. Appliance-like robots are generally made from rigid materials such as plastics or metals with smooth finishes. While the use of alternative and handcrafted materials has been emergent in other interactive technologies [56, 83], it has been less explored in social robotics. Open Platform for SOcial RObots (OPSORO) is an exception in that it uses fabrics in the design of its soft covers [80], similar to Blossom. Additional examples of alternative materials exist in hobbyist circles, such as TJBot, a single-DoF desktop robot, and Smartibot, a phone-controlled mobile robot, both constructed from cardboard [8, 22].

For actuation, most robots use rigid mechanisms and direct-drive motors to achieve movement. Smooth motions must thus be achieved through intricate control software and trajectory generation tailored to the robot's kinematics. Some have explored pneumatic actuators that can achieve smooth motion through mechanical design [66], but the pumps and compressors required to drive these systems are noisy and cumbersome. Another approach is to use tensile mechanisms to trade precise control for range and smoothness of motion. One example is the prototype robot Tofu, which has a head attached to a foam column with cables pulling on the head for actuation [82]. Another example is Probo, a robot with a tensile trunk [28].

Blossom combines aspects of the above-mentioned works resulting in a design that is flexible, inside and out. Blossom features an open-ended exterior meant to be customized by end-users through handcrafted materials, and its interior actuation mechanism uses compliant tensile components. This actuation mechanism is kinematically similar to Stewart platform mechanisms, which were used in the DragonBot [71] and Peeqo [72] robots. However, in contrast to those mechanisms, Blossom uses compliant components to achieve smooth and lifelike movement without requiring complicated software control. It also achieves a larger range of motion than Stewart platforms with only half the number of motors. Blossom's mechanism is most similar to that of the Tofu robot [82] but has a larger range of motion due to its free-floating platform (see below); it is also simpler to manufacture. In addition, Blossom's exterior cover and internal mechanisms are not affixed to each other, allowing for more expressive movement through slip and secondary action.

### 2.2 Robot Gesture Generation

Generating smooth and natural movements and gestures for social robots can be a lengthy and complicated process. Traditional methods for gesture generation are generally programmatic, require knowledge of the robot's kinematics, and are not accessible to novice users. Allowing users to create their own gestures affords a novel method of personalizing the robot and could help mitigate the novelty effect stemming from robot movements being repetitive and predictable.

In efforts to make robot gesture generation more accessible and intuitive, researchers have developed methods involving physical manipulation of the robot. Mirror puppeteering involves placing markers on parts of the robot and manipulating it in front of a camera to record movements [73]. Robots like Topobo and ChainFORM implement "kinetic memory," which allows gestures to be recorded by physically moving the robot's appendages using back-drivable motors with position encoders [54, 61]. Learning from Demonstration supplements either approach by having the user provide corrective demonstrations to iteratively teach the proper movements [6]. These demonstrations can be given either directly by manipulating the robot's actuators or indirectly by teleoperating the robot or using sensors attached to the user. All of these approaches are more intuitive than programmatic methods, but are either difficult to perform in real-time or rely on a known mapping between the demonstration source and the target robot's embodiment. In some cases, keyframes and interpolation can be used to complement the puppeteering activity [3], and the mapping between demonstrator and robot embodiments can be defined heuristically [4]. However, these limitations make it difficult to achieve high-quality expressive movements.

In contrast, Blossom allows lay-users to create gestures using a smartphone as a puppeteering interface. The robot's actuation mechanism kinematically resembles a free-floating platform and is controlled by mapping the orientation of the phone to that of the robot's head platform directly. This enables real-time exploration of four DoFs simultaneously, as well as real-time gesture recording.

### 2.3 Open-Source Robots

There are a few existing open-source robotics projects that allow users to build their own robot from openly accessible online data files. Robots like iCub, Poppy, and InMoov are examples of open-source platforms that have humanoid bodies and intricate mechanical and software designs [44, 45, 52]. Non-anthropomorphic open-source robots such as Hexy and TurtleBot are comparatively simpler [5, 26], owing to their more abstract embodiments. While the design of these robots is openly accessible, their appearances are largely fixed, and the systems require a high degree of technical knowledge to build, program, and use. Some of these robots can be programmed through visual block-based languages such as Scratch or Blockly [25, 63], but this programmatic approach does not support the authoring of new expressive gestures, making it less appropriate for social robotics applications.

While there have been many toolkits created for research in human-computer interaction (HCI) [47], there have been few that are specifically designed for constructing robots. Phybots is a toolkit that enables non-roboticists to quickly prototype and program a custom mobile robot [40], but is not designed for social interaction. CuddleBits are a series of do-it-yourself robots meant to be simple to design and build while maintaining a minimal level of expressiveness through a single DoF [14]. Among open-source robot platforms, OPSORO is the most socially-oriented. It is comprised of modular components representing different facial features and a customizable exterior cover that is made from soft materials [80]. This makes it more accessible and expressive than most other open-source robots. A semester-long deployment of the robot in a student design course produced several unique embodiments ranging from animals to the likeness of Albert Einstein. Blossom extends the foundations laid by the OPSORO project, going beyond facial expressions to full-body gestures, as well as enabling the authoring of behaviors without programming.

In summary, the Blossom platform offers a design that combines many features from previous robots described in the HRI literature, from Keepon and Dragonbot's soft exterior, to Tofu's tensile mechanism, to OSPORO's construction kit and fabric exterior approach. Blossom extends this

**Accessibility**

- Open-source design
- Easy to build and program
- Low-cost materials

**Flexibility**

- Customizable exterior
- Custom appendages
- User-defined behaviors
- OpenWoZ [32] API

**Expressiveness**

- Soft, organic appearance
- Tensile mechanism
- Smartphone gesture authoring

Fig. 3. Design objectives of the Blossom platform and features that address these objectives.

work by providing a construction kit for a full-body expressive robot, offering a larger range of motion with more secondary movement, a customizable DoF for appendages, and a way to author expressive gestures without programming or animation.

## 3 DESIGN OBJECTIVES

Blossom is designed to allow lay-users to create their own robot end-to-end, from building its structure, through the design of its appearance, to the authoring of new gestures and the combination of these gestures into behaviors. The design was driven by the following three objectives (Figure 3):

*Accessibility.* Users without technical knowledge should be able to contribute to all aspects of building and programming the robot.

*Flexibility.* The robot's design should allow end-users to alter aspects of its appearance, structure, and interactive capabilities.

*Expressiveness.* Despite the accessibility and flexibility of the robot's design, it should maintain a high degree of expressiveness in its appearance and movement. The movement should be smooth without relying on complicated control software.

## 4 IMPLEMENTATION

This section describes the technical implementation of Blossom in pursuit of the above objectives. It serves to enable the replication and extension of the technical aspects of the robot design. In overview, the robot's mechanical structure is made up of flat components that can be cut from sheets of wood or acrylic and uses snap and press fits to reduce the need for fasteners. It is actuated by a non-rigid tensile mechanism constructed from elastic components to achieve compliant, organic movement. One of the DoFs is open-ended and can be used to actuate custom appendages. The electrical design uses mostly snap connectors that do not require soldering and allows the robot to be either controlled by an external computer via USB or run untethered using an on-board battery-powered microcomputer. In both cases, an open Hypertext Transport Protocol (HTTP) application programming interface (API) allows remote control and programming of the robot's behaviors. The robot's gestures are authored using a smartphone-based puppeteering application, which can be recorded and played back in real time during operation, or saved on the robot to be triggered by the remote HTTP API.

### 4.1 Mechanics

Blossom's mechanical design is centered around a free-floating "head" platform, which is actuated using a tensile mechanism for power transmission (Figure 4). The head is suspended from the top of the central tower structure with rubber bands and is actuated by reeling in cables from the bottom of the tower. The design is related to the Stewart platform mechanism, which has been

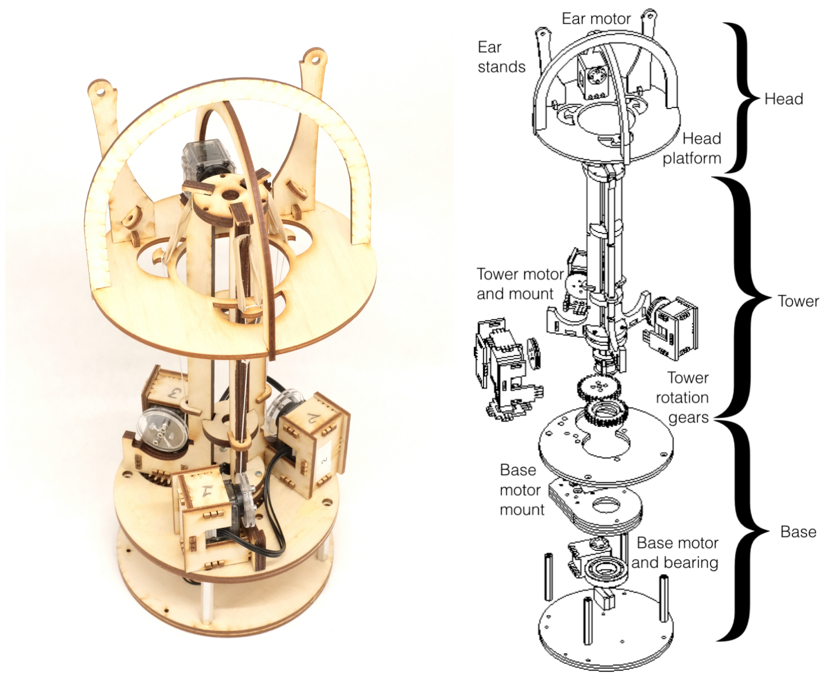


Fig. 4. The inner tensile actuation mechanism and an exploded view of the mechanism. The main expressive element is the head platform, which is suspended from a tower by rubber bands and actuated by cables driven by motors at the bottom of the tower. The tower itself is rotated by the base motor. As an example of an appendage, the head platform features ear stands and a motor for actuating the ears.

used in other social robots [69, 72], but Blossom's design is non-rigid and allows for a larger range of motion than a Stewart platform, all while reducing the number of motors from six to three. This is made possible through the variable lengths of the tensile components, whereas a Stewart platform is limited by the fixed lengths of its rigid linkages.

This actuation mechanism also bears similarity to the one used in the prototype Tofu robot [82]. While there is not much published information about the robot, it is described as also using an elastic element (a cylindrical foam core) to hold a head, which is actuated by cables. However, unlike the foam core used in Tofu to which the head and skin are rigidly attached, Blossom uses a free-floating head with elastic bands (Figure 7, left), as well as a freely moving exterior cover. This not only lowers the cost and difficulty of assembly, but also allows for larger range of motion that is accentuated by secondary motions. Additional movement is produced by a fourth motor in the base to rotate the tower assembly and a fifth motor on the head platform that actuates customizable appendages.

*4.1.1 Range of Motion.* Figure 5 shows examples of the head platform's range of motion. The gestures of the inner mechanism can be viewed as superpositions of several basic motion primitives: moving all the tower motors synchronously causes vertical translation, asynchronous motion results in pitching or rolling, and moving the base motor produces yawing. These fundamental motions are combined in timed sequences to create expressive gestures.

In addition to the increased range of motion, the tensile mechanism affords gestures that are smooth and organic-looking to an extent that would be challenging to replicate through software alone. The physical elasticity specifically supports several principles of animation [46, 64]. The

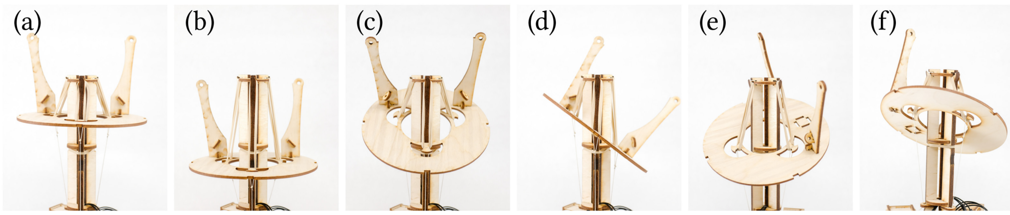


Fig. 5. Examples of the mechanism's range of motion. Vertical translation (a→b) and rotations (c, d) are combined to create more complex gestures (e, f).

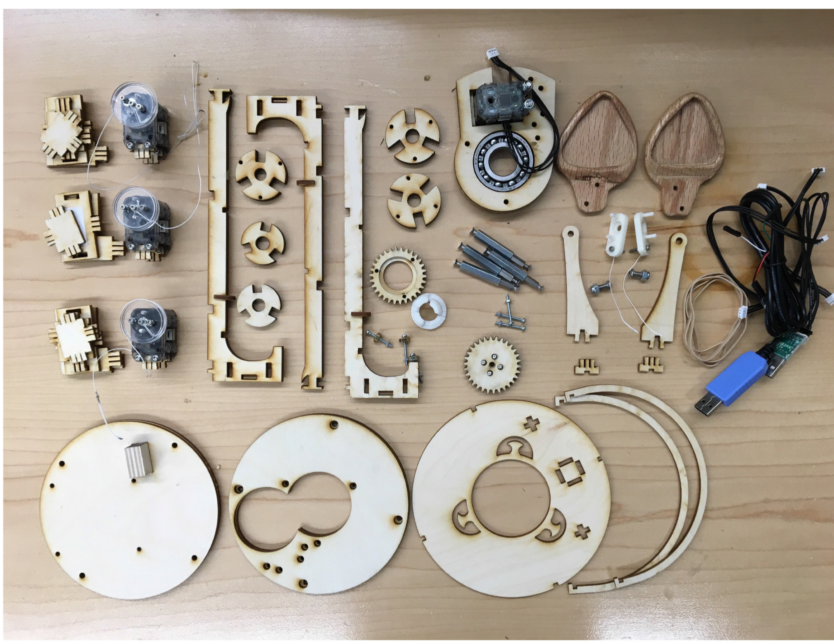

Fig. 6. Layout of the components used to assemble the mechanism.

cables and elastic bands provide a springiness that enables ease-in and ease-out in smooth arcs. The variable lengths of these components allow for greater exaggeration in motion. The momentum of the platform during quick movements elicits natural secondary motions such as overshoot and oscillation that would otherwise necessitate complex trajectory generation in motion planning software.

Given the novel mechanical structure of Blossom's design, we present a full derivation of the robot's forward and inverse kinematics in Appendix A.

*4.1.2 Fabrication.* Blossom's fabrication process relies almost exclusively on laser cutting, which has advantages over 3D printing for its reproducibility and speed, as well as for the affordance of low-cost, recyclable, and readily available materials such as wood and cardboard. The structure uses snap fits similar to OPSORO's design to reduce the amount of required hardware fasteners while being expandable with different appendages and motor configurations. Figure 6 shows all of the components needed to build one Blossom robot with ears as appendages. Figure 7 (right) shows the motor mount as an example of a snap-fit component.

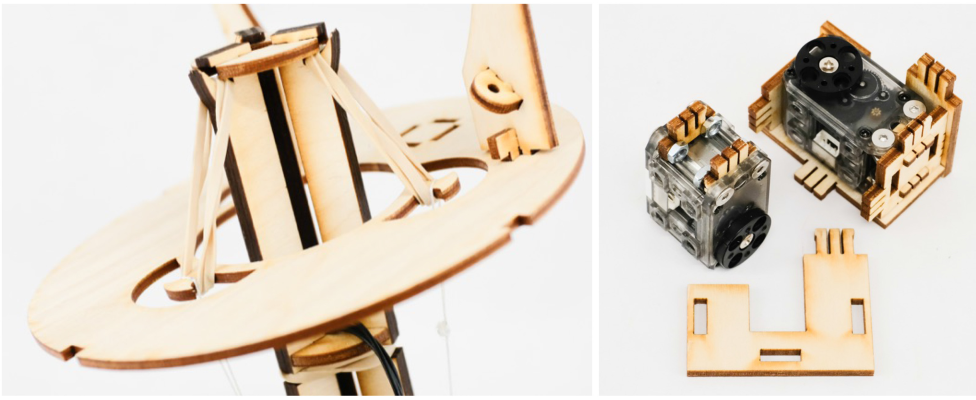

Fig. 7. Details of the compliant components (elastic bands and strings) used to suspend the head platform (left) and a snap-fit motor mount (right). Snap and press fits are used throughout the structure for ease of assembly and to reduce the amount of required hardware.

Table 1. Approximate Material and Component Costs for Blossom Components at Time of Writing

| **Part** | Motors | Motor cables | USB motor controller | Wood | Hardware | Bearing | **Total** |
|---|---|---|---|---|---|---|---|
| **Cost** | \$110 | \$30 | \$50 | <\$20 | <\$20 | \$8 | **<\$250** |

Table 1 lays out the approximate costs of materials to build the robot itself, excluding the small acrylic and 3D-printed parts and optional onboard microcomputer. The laser-cut components and 3D-printed parts were fabricated on an Epilog™ Helix 60W laser cutter and a Zortrax™ M200 3D printer, respectively. The motors used are all Dynamixel™ XL-320's and the USB motor controller is either a Xevelabs™ USB2AX USB-to-TTL interface or a Dynamixel™ U2D2. The hardware includes the nuts and bolts, string, and rubber bands required. The ease of construction and relatively low material cost make Blossom more accessible than other closed-source research platforms and commercial robots. Uninitiated users with no robotics experience were able to build a full robot from components in less than 2 hours. In terms of cost, it is slightly more expensive than the Anki Vector robot, a social robotics consumer product, which costs \$175 at time of writing, and significantly less expensive than a Softbank Robotics NAO robot, which is prized at circa \$8000 USD, or a Jibo robot at \$900.

### 4.2 Electronics

The electronics system also supports the design principle of accessibility by consisting of commercially-available components that use simple mechanical connectors, reducing the need for soldering.

Figure 8 shows the components of the robot's electronics system. The robot consists of five daisy-chained servo motors and a Raspberry Pi (RPi) microcomputer running the Linux operating system. The motors are controlled by the computer via a USB motor controller, which contains hardware to translate the USB protocol to Transistor-Transistor Logic (TTL) signals, and manages the half-duplex communication protocol of the servo motors.

The robot can be used in one of two modes: self-contained or externally controlled. In the self-contained mode, the motors are connected to the RPi with the motor controller. Both the RPi and

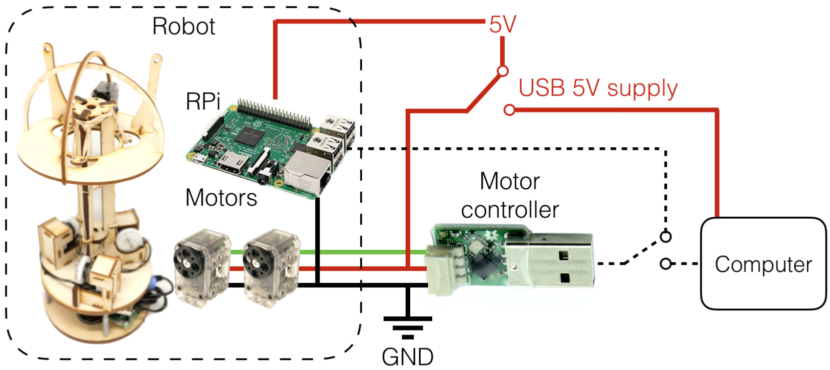


Fig. 8. Electronic component diagram. The robot can be used both in self-contained mode through an internal system-on-board, or controlled by an external computer. The motors within the robot are daisy-chained and thus only require one connection to the computer via the USB motor controller.

motors are powered by a 5-Volt (5V) power source such as a portable battery pack, but separate power connectors are used to prevent current overload on the logic components.

In the externally controlled mode, the RPi is unused and the motor controller is plugged into an external computer. Because power cannot be supplied through the motor controller and to prevent overcurrent on the computer's USB port, the motors must be powered from a separate 5V source such as an additional USB port or an external power supply.

### 4.3 Software

The software system of Blossom was designed in support of the flexibility and accessibility objectives. We chose to develop the software based on the Open Wizard-of-Oz (OpenWoZ) [32] framework, which runs as an HTTP server on the robot and allows open access to behaviors through a Universal Resource Identifier (URI)-based interface. This allows for application-level flexibility by exposing each of the robot's behaviors through a URI with parameters and thus enables driving the robot in various ways, from manual control triggers to programming its behavior in code. Other web-based robotics frameworks have been developed, including `rosbridge` [20], Robot Web Tools [78], the Robot Operating System (ROS) Transmission Control Protocol (TCP) [60], Raputya [35, 53], and standard protocol-based approaches [62]. We opted for OpenWoZ for three reasons: First, it has a lightweight footprint, as the full server and event-handling code is contained in a single python class and does not require installation of a large framework such as ROS. Second, the same software runs whether the robot is used in self-contained mode from the onboard RPi or externally controlled with a computer, and is compatible with major operating systems (Blossom runs on Linux, macOS, and Windows). Third, OpenWoZ is specifically tailored to trigger and modulate expressive gestures, with HRI research in mind.

The robot's software is made up of three main components (Figure 9): a motor control module and gesture library to command the motors as well as to store and play back authored movements (shaded light gray); an HTTP server, which listens to incoming requests and activates the appropriate gestures (shaded light gray); and the various user interfaces (UIs) for commanding the robot (shaded dark gray).

*4.3.1 Motor Control Module and Gesture Library.* The motor control is built on top of the PyPot motor control library [45], which abstracts the low-level serial communication for the servo motors to higher-level commands such as addressing motors and setting goal positions and speeds.

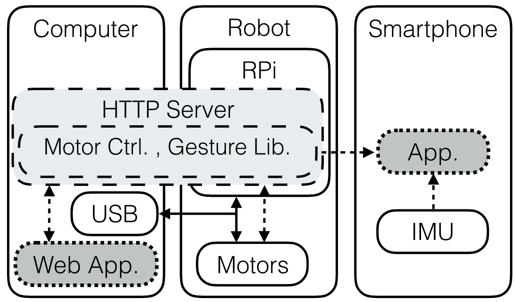


Fig. 9. Combined hardware and software diagram. Solid lines denote hardware, dashed lines and light gray shading denotes back-end software, and dotted lines with dark gray shading denote user interfaces. Physical connections are denoted by solid arrows and software communication is denoted by dashed arrows.

In the motor control module, robots are defined by the motors used and their respective ranges. The motors can be commanded directly, or controlled by executing gestures from a library. Gestures are stored as timed sequences of positions for each motor on the robot. The gestures can be played back with modulations to the speed, range, or posture.

*4.3.2 HTTP Server.* The control computer includes an HTTP server that is built on top of the open-source `flask` python server and enables Representational State Transfer ("RESTful") communication with the robot, allowing for the robot to be commanded from any device on the local network. This enables an open-ended method for interfacing with the robot and makes it easy to build WoZ interfaces, create custom applications that use sensor information, or communicate with existing web-based services or Internet-enabled devices.

The RESTful API receives the desired command or gesture and modulation parameters. For example, to play back a gesture titled "nodding" at 0.8 times the recorded speed and 1.4 times the amplitude of the original range of motion, the REST command would be `/s/nodding?speed=0.8&amp=1.4`. Examples of other functions include retrieving a list of available gestures and commanding the robot to a given position. This implementation follows the modular command structure of OpenWoZ [32] and affords flexible communication between the robot and clients built into user interfaces.

Additional behaviors can be added to the open-source HTTP server simply by defining a function and linking it to a RESTful command. Parameters are passed to the function as a URI string, and the custom behavior can parse the parameters. This is, for example, how different "breathing" and other programmatic idle behaviors are implemented.

*4.3.3 User Interfaces.* We demonstrate the flexibility afforded by the software architecture by presenting several methods we have developed to control the robot. In the simplest case, users can use the command line interface (CLI) on the terminal that started the robot HTTP server to trigger any command available to the RESTful API by simply typing in the REST URI. Beyond the CLI, we developed web and smartphone WoZ applications for high-level operation of the robot. A "soundboard" design enables the creation of buttons for triggering gestures or for gesture/modulation combinations (Figure 11(b)).

An additional web application is embedded in a web page (Figure 10). It allows Blossom to "react" to an online video as part of a research project in our laboratory, in which Blossom acts as a video-watching companion. The web page includes a video player and a Blockly interface for triggering gestures at specified timestamps and modulating them, allowing users to easily choreograph movement sequences to videos.

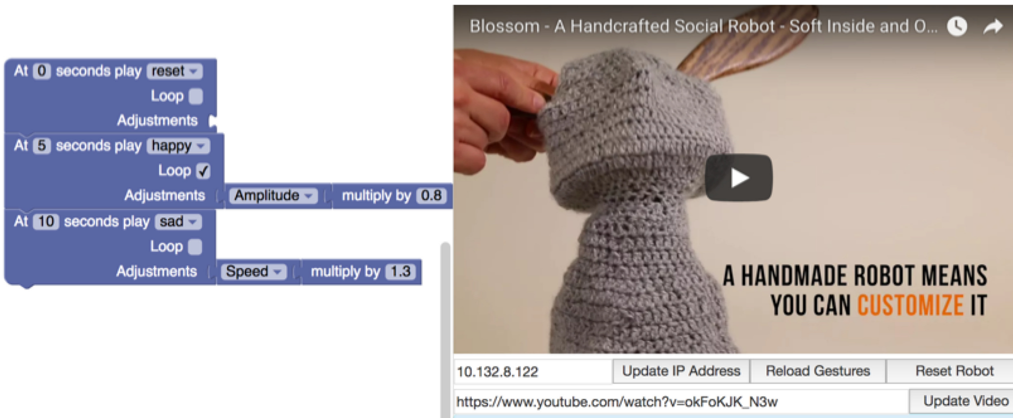


Fig. 10. The web application used to trigger gestures timed to a video. The Blockly interface is used to denote when to trigger gestures and how to modify playback speed, amplitude, posture, or looping. In this example, the robot resets at the beginning of the video, plays the "happy" gesture at 5 seconds at 0.8 times the original amplitude (range of movement) and loops until 10 seconds, at which time it plays the "sad" gesture sped up by a factor of 1.3.

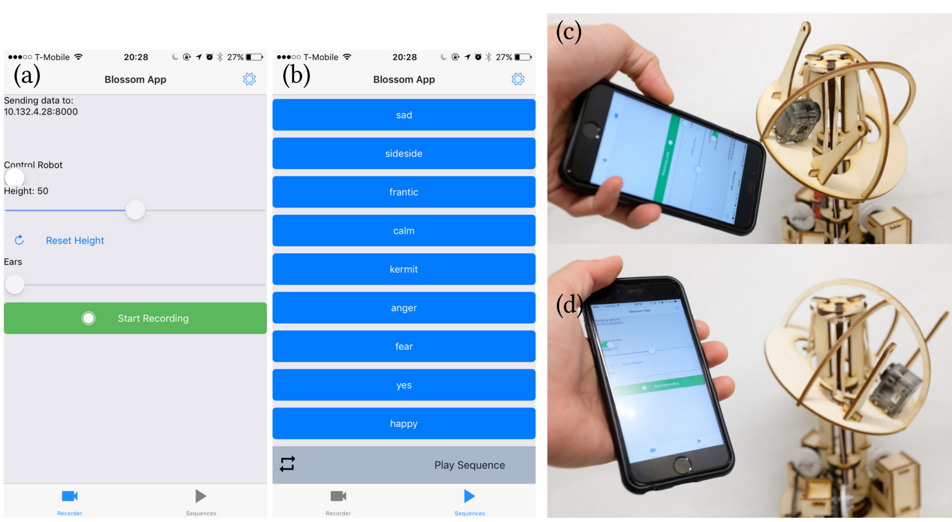


Fig. 11. Screenshots of the phone application for controlling the robot (a) and playing back gestures from within the application (b). The orientation of the phone is mapped to the orientation of the robot's head (c and d).

Beyond triggering and modulating gestures through a WoZ "soundboard," the mobile application (Figure 11) utilizes the phone as a puppeteering device to control the robot's expressive elements. Using smartphones as an input device supports the accessibility design objective by allowing lay-users to easily create behaviors for the robot without having to manually program its movements.

The puppeteering system is built on the React Native™ framework and leverages the smartphone's built-in inertial measurement unit (IMU) to map the phone's orientation to the orientation of the platform. Phone data (kinematic orientation, slider positions) is sent from the phone to the

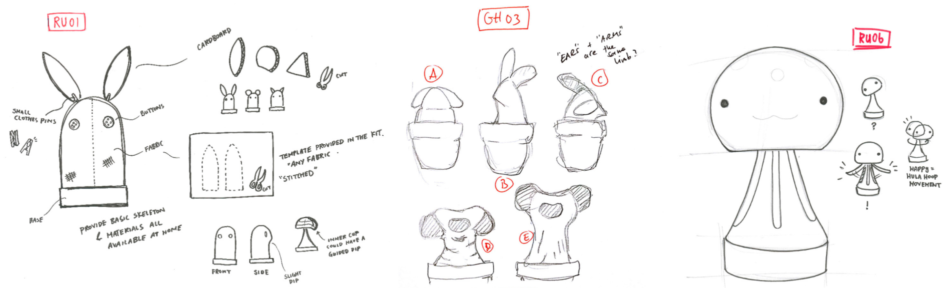


Fig. 12. Concept sketches exploring different embodiments and movements. The sketches show ideas for interchangeable exterior shapes and appendages meant to be hand-crafted by end-users.

robot using the same RESTful API as previously mentioned. The inverse kinematics of the robot as derived in Appendix A is used to determine the motor positions required to achieve a given orientation. Currently, the IMU only controls the 3D orientation of the head, but not the vertical offset of the platform's height. This is because integrating the IMU's raw accelerometer measurements at the current data rate (approximately 10Hz) would quickly result in sensor drift. To solve this, a slider adjusts the platform's resting height. Another slider controls the appendage motor. A mirror mode can be toggled to reflect the motion horizontally to make it easier to control the robot while it faces the user. Gestures can be recorded and played back within the application and can also be looped indefinitely to make idling motions such as breathing or looking around.

## 5 APPEARANCE

The robot's flexibility extends to its outer appearance design. Its exterior is created from soft fabrics that are not rigidly attached to the interior skeleton, and its appendages are interchangeable and in principle open to any tensile mechanism. Concept sketches from the ideation process of various exterior options are shown in Figure 12, illustrating the flexibility in the robot's appearance.

### 5.1 Soft Exterior

The soft woven exterior of the robot supports expressiveness in two ways: by augmenting the compliance of the internal mechanism through its bending and folding, and due to the fact that it flows freely over the structure. This helps the robot to appear more lifelike by accentuating the organic movement and providing mechanical flexibility to its exterior. Using traditional crafts rather than computer-aided design (CAD) and rigid manufacturing techniques also supports the design goal of accessibility by enabling a diverse user population to participate in robot-building.

Three examples of crocheted covers are shown in Figure 1, one in the likeness of a blue bunny clown, one in the shape of a gray mouse or cat, and the third modeled after a blue jellyfish. They are knit or crocheted out of wool. The blue-and-white design is constructed as a single pull-over piece; the exterior for the mouse design is also a single piece, but it is open at the top and closes with a button in the back of the head; the jellyfish cover is made of two pieces (one for the head and one for the lower body) that button together at the base of the head. The covers are designed to be loose-fitting to support the organic movement aesthetic and to not constrain the actuation mechanism.

### 5.2 Swappable Appendages

The robot's flexibility is further emphasized by its swappable and open-ended appendage mechanism. The head platform features an additional motor that can interface with various accessories

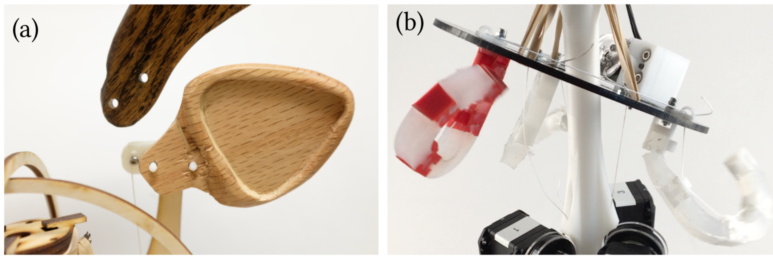


Fig. 13. Two examples of the swappable appendages: (a) two versions of pluggable wooden ears and (b) flexible silicon arms. Both appendages are actuated using the same tensile mechanism from the appendage motor mounted on the main head platform.

and appendages matched to different exterior designs. Control of the appendages is also tensile, with the motor reeling in a cable and either gravity or an elastic element restoring the DoF.

Figure 13(a) shows the mechanism for the ears. The ears attach to posts on a rotating hinge adapter with two hooks, allowing them to be easily interchanged. The hinge adapter itself is tethered to the accessory motor. The jellyfish configuration features flexible arms as shown in Figure 13(b). The arms are fabricated by first 3D printing a "skeleton" mold, which is then filled with Smooth-On EcoFlex™50 silicone. The rigid skeleton segments act as vertebrae with the silicone serving as ligaments that connect all of the segments. In both cases, gravity restores the DoF. We implemented two examples of swappable appendages, but in theory, any single tensile DoF could be added to the robot's design. In the prototyping phase of the robot's design, we explored tails and spinal spikes as additional DoFs.

## 6 CASE STUDIES

Evaluating toolkits in HCI research is a challenge despite the appreciation for the long-term impact that toolkits have had on HCI research. Ledo et al. surveyed evaluation techniques for HCI toolkits over two decades [47] and found that demonstrations and and observation of usage are the most commonly used practices. To evaluate Blossom vis-a-vis its design objectives, we have deployed the system in four demonstration and usage contexts and observed users' interaction with the system. The four case studies are: External HRI research groups that built and used the robot; users interacting with the robot in public exhibition settings; young children designing the robot's appearance and creating gestures for it; and a workshop in which middle school students built robots and authored gestures for them. These deployments provided feedback on the design and insight on how diverse users interact with the robot.

### 6.1 Providing the Design to External Research Groups

One of the main goals of the Blossom platform is to provide HRI researchers who are not engineers with an accessible social robotics platform. To that end, we provided Blossom prototypes to several external HRI research groups. We chose an evolutionary prototyping approach [15, 21, 27], which enables iterative reevalution of the design through increasingly mature developments of the system. In our case, we progressed from providing the user with the completely built system to letting them build the full robot from instructions only.

The first prototype was sent completely prebuilt to an industry research team studying robot companions for children with autism. The research team was able to set up the robot and control it in its untethered mode. They used it in technology demonstrations when meeting with therapists and user populations. This is the use-case of researcher-as-end-user, demonstrating the possibility to quickly set up a Blossom platform and use it in Wizard-of-Oz mode. This deployment was done before the smartphone gesture authoring system was completed; the researchers used a provided set of behaviors triggered from a web-based soundboard.

A second prototype of the robot was given to a university research team with limited engineering skills. The research group conducts field and laboratory studies related to public health. We provided the basic components (as laid out in Figure 6) and a repository with the assembly instructions and software library [74]. The group was able to successfully build the robot, install the software, and enlist the help of volunteers to crochet new covers. The group has since used the robot in a number of field studies. Their study participants used Blossom in a teleoperation scenario to communicate with each other by gesturing via the robot, using the smartphone control application. This is the use-case of researcher-as-builder, demonstrating the possibility of assembling a robot and customizing its appearance without engineering skills.

Two other prototypes were built by additional university research groups. Unlike the previous groups, we provided no components and gave the groups only a link to the repository containing the laser cutting design files, software, and instructions [74]. Apart from troubleshooting some minor software-related issues with one of the groups, both research teams were able to independently fabricate the robot. This is the full open-source use-case of Blossom, demonstrating the ability to build a new robot within days. This use-case indicates the possibility of scaling up the use of the system through digital distribution of code, CAD files, and instructions only.

### 6.2 Public Exhibitions

Throughout the development of Blossom, prototypes of the robot have been exhibited at several public events, including two technology fairs, one academic conference, and a collegiate project team showcase. These events had a diverse demographic of attendees, from lay-users to roboticists, and were opportunities to present Blossom to receive feedback on its design from a variety of populations. During these events, we showcased Blossom's movement and customizability and explained the motivation for the project. Users experienced the robot, but did not build a robot.

Participants responded positively to the robot's appearance, and several indicated that they would want to interact with it like a pet. At the project showcase, we showed different configurations and allowed participants to control Blossom with the phone. Though many found the controller to be somewhat difficult at the beginning, they found the interaction to be entertaining and would use Blossom to gesture to their friends. This bears similarity to the teleoperation use by the HRI research group described above. Along positive comments regarding the design, there were a few recurring questions and suggestions. A common question was whether Blossom could react to user input and whether it had sensors such as cameras or microphones. Others suggested interfacing Blossom with voice-based assistants to provide them with a physical embodiment. Many expressed interest in owning or building a Blossom robot.

### 6.3 Children's Science Day

We exhibited Blossom at a children's Science Day event where young children, approximately 4–8 years of age, could visit stations with various activities. For our activity, we had craft materials available for children to create accessories for Blossom. Children would then affix the accessories to Blossom and control the robot using a smartphone. Participants interacted with Blossom in different ways, with some staying at the booth for a long time crafting several accessories and others only interested in controlling the robot. There were some children who came in groups and took turns between crafting accessories and controlling; these groups sometimes collaborated by having the crafter ask the controller to move the robot to make it easier to attach an accessory.

Fig. 14. Children interacting with Blossom at the Science Day event (top) and examples of accessories created by participants (bottom).

This suggests that the Blossom platform can encourage collaborative design and the interaction of several users with a single robot.

Although we initially suggested creating ears, children branched off to make a wide range of different accessories, from appendages to facial features to jewelry. Most creations were simple single-layer shapes, but some designs were more elaborate and featured multiple layers and adornments (Figure 14). The diversity of accessories suggests the flexible design of the platform.

The ways that children controlled Blossom led to interesting observations regarding the smartphone as a controller. Users would often move the phone in exaggerated ways that Blossom would physically not be capable of achieving, such as turning completely upside down or twisting around over 360°. The children also had additional expectation of the robot's movement, such as making it locomote and jump. These expectations were emphasized by that fact that several children chose to make appendages such as legs and wings.

Adults were also interested in Blossom, from the project's application to its technical implementation. Some parents participated by making their own accessories while others helped their children control the robot more effectively. They commented on the project as relating art and technology, fitting in with the goals of Science, Technology, Engineering, Arts, and Math (STEAM) education. Adults also noted Blossom's unconventional appearance compared to traditional robot aesthetics.

Fig. 15. Examples of the embodiments created by the middle school students in the robot-building workshop.

### 6.4 Build-a-Blossom Workshop

In the most elaborate deployment, Blossom was used in an educational workshop for middle school students to learn about the skills involved in robotics engineering. The students had varying levels of technical experience, ranging from some programming and prototyping to very little exposure to coding or mechanical construction. The activity was to build and customize a Blossom robot, program its gestures, and choreograph its movements to a video of the students' choosing. There were six workshop sessions; each was approximately 80 minutes long and had 16–20 students that were divided into four groups. The total was 107 students in 24 groups. Lab members familiar with the construction and programming processes were present to provide assistance, but intervention was kept to a minimum and mainly involved guided troubleshooting.

Each group was provided a partially-assembled robot and the assembly instructions. We decided to provide some preassembly after the first round (four groups) of students spent most of the 80 minutes building the robot, leaving too little time for gesture generation. All subsequent 20 groups built the head platform from scratch, attached the ear mechanism, connected the tower to the base assembly, connected the motors to the robot, attached the head by hanging it from the tower, and wired the motor cables. A crocheted cover was included with each robot, as newly crafting a robot exterior was also not feasible in the time allotted.

We observed that in many cases, during the construction process, some students were building the inner structure, while other group members customized the cover with craft accessories. Figure 15 shows examples of some of the appearances created by workshop participants.

Once the robot was assembled and customized, students connected it via USB to a computer and authored movements using the smartphone application. Most groups designated one member with the mobile application to be the movement choreographer in charge of creating gestures. Students then imported the gestures into the web application and timed each movement, some with modulation, to the video chosen by the group. This resulted in a variety of choreographies with which the robot reacted to the student's videos. The videos themselves ranged from music videos to which the robot was made to dance, sometimes "dressed up" as the performing artist, to humorous videos with the robot reacting as an audience. Other examples included viral videos ("memes") where the robot was fashioned like one of the characters in the video, imitating the action on screen.

All of the 24 groups were able to successfully build and control the robot by the end of their session. The structures were mostly assembled correctly, except for the ear assembly, which often had cable routing issues. The programming process was largely error-free and some groups were able to produce fairly complex choreographies. However, similar to our observation at the children's Science Day event, many students tried to control the robot in impossible manners.

The vast majority of students were actively engaged throughout the sessions. We conducted brief informal question-and-answer sessions at the end of each meeting, where students were asked to say what their favorite and least favorite part of the workshop was. There was a wide variety of responses about the favorite part, with some students enjoying the craft more, and others preferring the mechanical construction or the gesture generation. Similar to the children's Science Day experience, this suggests that the Blossom platform allows students with different interests to be involved in some capacity. Others expressed satisfaction at being able to build and control a complete working robot in a short time. Several students, who admitted to being initially disinterested in robotics or intimidated by the topic, found themselves enjoying it due to the engagement of the activity and the relation to personally meaningful video content. In a post-survey asking students to rate the workshop from "useless" to "very valuable", 11 of the 13 respondents rated the workshop as "valuable" or "very valuable," with only one rating of "not very valuable" and one rating of "not sure."

### 6.5 Discussion and Insights

In our field deployments, a diverse population of users interacted with different aspects of the Blossom system, leading to a number of insights.

*6.5.1 Relation to Design Objectives.* The fact that external HRI research groups, some with no engineering background, were able to build the robot with little assistance and readily use it for their own research work is an encouraging signal that an accessible open-source platform can be a useful model for social robotics research and an alternative to procuring commercial social robotics platforms. However, at this point, the robots have not yet been used widely by the groups; therefore, the long-term usefulness of the system for research has not been validated. During each step of the evolutionary open-sourcing of Blossom, we made improvements to the mechanical design, assembly instructions, software interface, and installation procedure. This process illuminated the tight integration between the design of the system and the design of the fabrication, installation, and operating instructions when developing open-hardware systems [59].

The middle school workshop helped us evaluate several of Blossom's major design objectives. The fact that untrained students were able to mostly build and animate the robot within 80 minutes supports the accessibility of the platform's assembly and gesture authoring workflow. The variety of embodiments and their relation to the personal content choice of the students emphasized Blossom's flexibility. The complexity of the resulting choreographies indicates to us the robot's expressiveness. Similarly, showcasing Blossom at public exhibitions was useful in demonstrating its expressiveness and receiving feedback on the design. The largely positive comments regarding Blossom's appearance are encouraging and affirm that the design appeals to a wide audience.

*6.5.2 Interacting with Blossom.* Both the study participants of the academic researchers and the visitors to our public exhibitions frequently used the robot as a teleoperation "avatar" to communicate with their peers through the live-control mode of the smartphone application. The direct mapping affordance between the handheld device and the real-time reaction from the robot possibly encourages using Blossom as a gesture-extension of the self.

That said, in all of our deployments, we found that many users who attempted to control the robot tried to move it in ways that it was not capable of, such as turning it through a 360° rotation and flipping it upside-down. After some practice, users were able to use the phone to create gestures, but many had difficulty with accurately controlling the robot, and the mapping often remained unintuitive to users. This reveals a problematic mapping between the unconstrained motion of the phone and the limited range of the robot.

In both the Science Day exhibit and in the middle school workshop, we noted that the design of the robot supported participation by users with diverging interests, such as appearance design and gesture generation. We see this being a result of the design's open-endedness on many different levels, providing multiple points of contact for users to participate. This aspect of the Blossom system also seems to encourage collaboration.

*6.5.3 Open Design Issues.* The difficulties that middle school students had in the assembly process of the ears highlighted weak points in the design that should be rectified in future iterations, most notably the appendage module, which was prone to failure. The interconnectivity between the robot's control computer and the phone controller can also be streamlined, as it requires a somewhat elaborate configuration process.

Given the short duration of the evaluation deployments, none of them included building new appendages or fashioning new covers for the robot. Future evaluations on customizability should explore these features for alternative embodiments. Finally, many visitors were curious about the sensing capabilities of the robot, pointing out a gap in the current design.

## 7 FUTURE WORK

The field deployments of Blossom, together with our own experience in manufacturing and using Blossom, indicate several points in which the current design can be improved upon.

*Smartphone Control Mapping.* We found issues in the mapping between how users move the phone and the robot's motion limits. Possible solutions include better instructions and training to control the robot properly, a mechanical rig to place the phone into, which can enforce the robot's movement constraints, and methods for better mapping from the raw orientation detected by the smartphone's IMU sensor to the robot's pose. Relatedly, many users attempted to control the height of the platform by raising and lowering the phone; while it would be difficult to get accurate height control with the smartphone IMU sensor, usable height control should be explored, possibly by using filtering or predictive methods to alleviate drift.

*Sensing Capabilities.* Users commented that they wished Blossom had sensing capabilities. Incorporating sensors for the robot to react to external inputs should thus be considered. Implementing sensors on the robot itself may compromise its accessibility and handcrafted aesthetic, but simple sensors could afford richer functionality without being obtrusive. Many have interacted with Blossom by petting its head or calling to it, and components such as touch sensors or microphones could be implemented to provide more functionality. Another approach is to leverage sensors built into smartphones, such as the microphone or camera, to avoid adding complexity directly to the design of the robot itself [31].

*Intermediate Programming Language.* The Blockly interface is currently only used for triggering gestures to videos, but it could also be used as a mid-level programming method that is more versatile than the existing Wizard-of-Oz interfaces, while being still more accessible than a full programming language. Features such as motor control and conditional statements responding to external inputs could be useful to expand the current functionality.

*Lower Cost.* On the mechanical side, while wood is relatively inexpensive and is well-aligned with the handcrafted aesthetic of the robot, transitioning to an even cheaper material such as cardboard or paper could further improve its accessibility. The most expensive aspect of the current design are the high-end servo motors. They provide many advantages over standard servo motors, primarily velocity, acceleration control, and daisy-chaining, but are relatively expensive. Transitioning to standard hobby servos would significantly reduce the overall cost of the platform at the potential cost of ease-of-control and movement quality.

*Diverse Appendages.* Finally, as our evaluation did not delve into user design of new appendages, we would like to explore these to illustrate the platform's customizability. Flexible arms and dinosaur-like spikes were briefly explored, but the ear design has proven to be the most easy-to-use and expressive. Given the inclusion of limbs and wings among the accessories created at the children's science event, such alternative configurations should be explored in the future. Different appendages may also affect the robot's expressiveness by altering its DoFs and therefore its gesture capabilities.

## 8 CONCLUSION

This article presented Blossom, an open-source social robotics platform that incorporates untraditional mechanisms and materials. Building on previous work in social robot design including, among others, the compliant exterior of the Keepon robot, the tensile mechanism of the Tofu robot, and the snap-fit construction kit approach of OPSORO, Blossom was developed with the design objectives of accessibility, flexibility, and expressiveness.

Blossom is designed to be *accessible* by several means. First, it is comprised of inexpensive mechanical components that are readily fabricated using low-cost materials such as wood and cardboard. The mechanical setup relies heavily on snap and press fits without the need to use fasteners. The robot's ease of construction was evaluated by providing external HRI research groups with the assembly instructions and software. These groups have successfully built replicas of the robot and implemented them in their own applications. The accessibility objective was further evaluated in a workshop for middle school students to build a Blossom and program its behaviors. All 24 student teams were able to build and control the robot in a single 80-minute session, although we identified construction issues with the open-ended appendage DoF.

A smartphone-based puppeteering controller is presented as a novel approach to gesture authoring, meant to be more accessible compared to traditional programmatic or computer animation methods. It was evaluated at the above-mentioned middle school student workshop, as well as at a children's Science Day exhibit, where participants and visitors were able to create a range of behaviors for the robot without prior training. That said, users often tried to move the phone beyond the robot's range of motion, leading to unpredictable mapping between the phone's orientation and the robot's configuration. Furthermore, not all of the robot's DoFs can be puppeteered at the same time. While we believe that the smartphone controller provides an accessible way to author robot gestures, we have yet to conduct a systematic comparison between the presented gesture authoring system and traditional approaches, such as programming and 3D animation. This is left for future work.

Blossom is designed for *flexibility* through an open-ended design that allows users to customize its appearance and behaviors. The benefits of personalizing robot behaviors has been explored in HRI research [18, 48, 50, 58, 68] and some researchers have looked at allowing users to personalize a robot's outward appearance, usually through adornments [24, 76, 80]. This feature could also work in support of the so-called IKEA effect, which predicts that self-made creations are favorably valued over comparable mass-manufactured objects [55, 75]. Blossom allows exterior skin personalization as well as a limited customization of an appendage DoF. However, the rest of the kinematics is fixed in the design. This compromise is grounded in the tradeoff between customizability and predictable expressive movement. Using the appendage customization, we have demonstrated two distinct configurations, rigid ears pointing up and to the sides, and curling soft tentacles hanging from the sides of the robot. Enabling more kinematic customizability while still retaining expressive movement and the IMU-based puppeteering capabilities is an open challenge.

Blossom's design objective of *expressiveness* was attempted through the use of compliant materials in the internal actuation mechanism. This provides the design with physical flexibility and results in smooth motion without requiring complicated control through software. The use of soft exteriors that flow freely over the mechanism creates an organic skin-like effect, which further accentuates the looseness and lifelike nature of the robot. Additionally, incorporating alternative materials such as woven fabric exteriors and wood supports the robot's aesthetic expressiveness and could offer an alternative for social robots that may fit the domestic space better than existing choices of plastics and metals.

In our field deployments, we repeatedly observed that different participants chose to engage with different aspects of the Blossom platform. Some preferred the mechanical construction; others liked creating new exterior covers or adorning those with craft materials. Yet others were interested in generating precise movements for the robot or stringing them together to create behaviors. Blossom's design seems to have many entry points for engagement. We can thus identify two additional design outcomes of the proposed system, namely that it supports *diversity* in the robot's user population and possibilities for *collaboration*. This notion is anecdotal and post-hoc, and should be separately evaluated in future research. We are currently working with a number of partners to test Blossom in additional settings, including high schools and maker spaces.

The design process uncovered opportunities and challenges in the design space of open-source social robotics. We hope that the detailed description of the system can help invite a broader population to participate in designing, building, and using social robots. The smartphone-based gesture authoring approach could be used in other HRI systems where there is a need to quickly generate robot behaviors by nontechnical users.

A core contribution of this work is for the proposed design to benefit HRI researchers who are not engineers, enabling them to readily construct simple social robots, customize the robots' appearance to their needs, and use them in a teleoperated manner without additional software development. Such researchers can also add behaviors and link them to a Wizard-of-Oz "soundboard" to trigger these custom gestures without coding. Additional connections to external sensors and software systems are enabled through the HTTP API.

As social robotics becomes more ubiquitous, a family of soft handcrafted robots that are easy to build and control may suggest a fruitful design path for social HRI.

## APPENDIX

## A BLOSSOM KINEMATICS

The novel design of Blossom's internal mechanism requires custom kinematics for gesture generation and simulation. In this appendix, we detail the forward and inverse kinematics derivation for the robot.

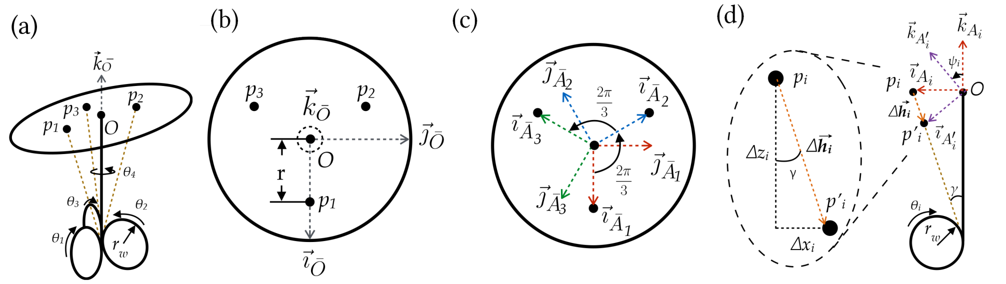


Fig. 16. Kinematic diagrams of the robot's inner mechanism. As shown in (a), the inertial reference frame $\bar{O} =< \vec{i}_{\bar{O}}, \vec{j}_{\bar{O}}, \vec{k}_{\bar{O}} >$ is defined with the origin $O$ at the center of the platform when at rest. The lines from the base of the tower to the attachment points $p_{1-3}$ represent the cable. The tower motors that actuate the platform are at the base of the tower and rotate the motor wheels of radius $r_w$ by angle $\theta_{1-3}$. The base motor located below the tower motors (not depicted) rotates the tower about the vertical axis by $\theta_4$. Top view diagrams in (b) show the locations of the attachment points. The frames $\bar{A}_1$, $\bar{A}_2$, and $\bar{A}_3$ depicted in (c) are aligned with the attachment points and rotate about the vertical $\vec{k}_{\bar{O}}$ axis shown in (b). The side view (d) shows the actuation mechanics of a single attachment point. The frame $\bar{A}'$ is aligned with $\bar{A}$ as it rotates about the vertical $\vec{k}_{\bar{O}}$ axis, but additionally rotates about the shared $\vec{j}_{\bar{A}} = \vec{j}_{\bar{A}'}$ axis out of the page. This results in the rotation from $\bar{A}$ to $\bar{A}'$ by the angle $\psi_i$. The displacement is approximated by $\Delta\vec{h}_i$ with components $\Delta x_i$ and $\Delta z_i$ in the $-\vec{i}_{\bar{A}}$ and $-\vec{k}_{\bar{A}}$ axes, respectively. The angle $\gamma$ is the angle between the vertical axis and the line formed by the cable when the platform is at rest.

## A.1 Forward Kinematics

Figure 16 shows an approximation of the inner mechanism. For simplification, the elastic bands are neglected and cables are assumed to be rigid links of variable length capable of both pushing and pulling the platform. The attachment points of the cables are denoted $p_{1-3}$. As shown in Figure 16(a), the tower motor wheels of radii $r_w$ rotate by $\theta_{1-3}$ and the base motor rotation about the vertical axis is denoted by $\theta_4$.

Top views in Figure 16(b) and (c) depict the locations of the attachment points and define the intermediate frames ($\bar{A}_1, \bar{A}_2, \bar{A}_3$). These intermediate frames are aligned with the attachment points $p_{1-3}$, respectively, and rotate about the vertical inertial axis, with all $\vec{k}$ axes shared: $\vec{k}_{\bar{O}} = \vec{k}_{\bar{A}_1} = \vec{k}_{\bar{A}_2} = \vec{k}_{\bar{A}_3}$, and shown as $\vec{k}_{\bar{O}}$ in Figure 16(b).

We are interested in the pose of the head platform given a set of motor angles $\theta_{1-4}$. Consider the movement of one of the attachment points, $p_i$, as depicted in Figure 16(d). The rotation of the motor wheel of radius $r_w$ by angle $\theta_i$ causes the cable to be pulled in by length $r_w\theta_i$. Denoting the angle between the vertical axis and the cable as $\gamma$, this shortening of the cable results in the displacement $\Delta\vec{h}_i$ of point $p_i$ from its resting position to the actuated point $p'_i$:

$$\Delta\vec{h}_i = -\Delta x_i \vec{i}_{\bar{J}} - \Delta z_i \vec{k}_{\bar{J}} = -r_w\theta_i \sin\gamma \vec{i}_{\bar{J}} - r_w\theta_i \cos\gamma \vec{k}_{\bar{J}} \tag{1}$$

A simplifying assumption is made that the attachment point moves along this line and that $\gamma$ remains constant. The resulting actuated reference frame $\bar{A}_i{}'$ is a rotation of the original $\bar{A}_1$ about the shared $\vec{j}_{\bar{A}_1} = \vec{j}_{\bar{A}_i{}'}$ axis out of the page. If we denote the vectors from $O$ to the resting position of the attachment point $p_i$ as $\vec{r}_i$, we get $\vec{r}_i = r\vec{i}_{A_i}$. After actuating motor $i$, we get the new vector from $O$ to $p_i$, $\vec{r}'_i$:

$$\vec{r}'_i = \vec{r}_i + \Delta\vec{h}_i = r\vec{i}_{A_i} + \Delta\vec{h}_i \tag{2}$$

These vectors need to further be transformed to the inertial frame by the planar rotation matrices of $\theta_4$ for $\bar{A_1}$ and of $\theta_4 + \frac{2\pi}{3}$ and $\theta_4 + \frac{4\pi}{3}$ for $\bar{A_2}$ and $\bar{A_3}$, respectively. The calculated positions of the attachment points can then be used to determine the resulting orientation of the platform.

To do so, we define unit normal vectors for the idle and transformed orientations as $\vec{N}$ and $\vec{N}'$, respectively. We take $\vec{N} = \vec{k}_{\bar{O}}$ to be simply pointing upward from $O$. The transformed vector $\vec{k}_{\bar{O}}$ can be calculated from a normalized cross product of the transformed attachment point vectors in the plane of the actuated platform.

$$\vec{N}' = \frac{(\vec{r}_1' - \vec{r}_2') \times (\vec{r}_1' - \vec{r}_3')}{|(\vec{r}_1' - \vec{r}_2') \times (\vec{r}_1' - \vec{r}_3')|} \tag{3}$$

The normal vector to the rotation plane $\vec{M}$ can be calculated and used to determine the quaternion rotation angle $\alpha$ and frame defined in $\vec{v}$.

$$\vec{M} = \frac{\vec{N} + \vec{N}'}{|\vec{N} + \vec{N}'|} \tag{4}$$

$$\alpha = \vec{M} \cdot \vec{N} \qquad \vec{v} = \vec{M} \times \vec{N} \tag{5}$$

$$\vec{q} = \begin{bmatrix} \alpha \\ \vec{v} \end{bmatrix} \tag{6}$$

This quaternion is then used to determine the change in orientation and the downward displacement is approximated using Horn's method. The resulting changes in position and orientation are superimposed to determine the final pose.

### A.2 Inverse Kinematics

Given the above forward kinematics solution, we can compute the head platform orientation given known motor positions. The same model can also be used to derive the inverse kinematics to calculate the required motor positions to achieve a desired final orientation of the platform. First, the Euler angles ($\psi, \theta$, and $\phi$ about the body $\vec{i}_{\bar{B}}-$, $\vec{j}_{\bar{B}}-$, and $\vec{k}_{\bar{B}}-$axes, respectively) of the desired orientation are used to derive the rotation matrix ${}^{\bar{O}}R^{\bar{B}}$ from the $\bar{B}$ frame in the final orientation to the inertial frame $\vec{O}$:

$${}^{\bar{O}}R^{\bar{B}} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - c\phi s\psi & s\psi s\phi + c\psi c\phi s\theta] \\ c\theta s\psi & c\psi c\phi + s\psi s\theta s\phi & c\phi s\psi s\theta - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{7}$$

The rotation matrix is used to transform the representations of the positions of the attachment points $\vec{r}_{p_i'}$ from the body frame $\vec{B}$ to the inertial frame $\vec{O}$:

$$\{\vec{r}_{p_i'}\}_{\bar{O}} = ^{\bar{O}} R^{\bar{B}}\{\vec{r}_{p_i'}\}_{\bar{B}} \tag{8}$$

From the initial ($\vec{r}_{p_i}$) and transformed ($\vec{r}_{p_i'}$) positions of the attachment points, the displacements $\Delta\vec{h}_i$ can be calculated by:

$$\Delta\vec{h}_i = \vec{r}_{p_i'} - \vec{r}_{p_i} \tag{9}$$

Given the known size of the motor wheel $r_w$, we can then calculate the angular motor displacement $\theta_i$:

$$|\Delta\vec{h}_i| = r_w\theta_i \rightarrow \theta_i = \frac{|\Delta\vec{h}_i|}{r_w} \tag{10}$$

## ACKNOWLEDGMENTS

We would like to thank our collaborators, Miguel de Andrés-Clavera, Rosa Uchima, and Gene Brutty for their input on the design of the robot and for project support. We would also like to thank Greg Holman, Harrison Chang, Michael Hu, Ankita Chandak, Antoine Klopocki, Miranda Jeffries, and Emma Cohn for their contributions to the project.

## REFERENCES

[1] Henny Admoni, Caroline Bank, Joshua Tan, Mariya Toneva, and Brian Scassellati. 2011. Robot gaze does not reflexively cue human attention. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 33.

[2] Henny Admoni, Bradley Hayes, David Feil-Seifer, Daniel Ullman, and Brian Scassellati. 2013. Are you looking at me?: Perception of robot attention is mediated by gaze type and group size. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction*. IEEE Press, 389–396.

[3] Baris Akgun, Maya Cakmak, Karl Jiang, and Andrea L. Thomaz. 2012. Keyframe-based learning from demonstration. *Int. J. Social Rob.* 4, 4 (2012), 343–355.

[4] Aris Alissandrakis, Chrystopher L. Nehaniv, and Kerstin Dautenhahn. 2007. Correspondence mapping induced state and action metrics for robotic imitation. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* 37, 2 (2007), 299–307.

[5] ArcBotics. 2016. Hexy the Hexapod. Retrieved from http://arcbotics.com/products/hexy/.

[6] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Rob. Auton. Syst.* 57, 5 (2009), 469–483.

[7] ASUS. 2018. Zenbo—Your Smart Little Companion. Retrieved from https://zenbo.asus.com/.

[8] Ross Atkin. 2018. Smartibot: The world's first A.I. enabled cardboard robot. Retrieved August 28, 2018 from https://www.kickstarter.com/projects/460355237/smartibot-the-worlds-first-ai-enabled-cardboard-ro.

[9] Kim Baraka, Ana Paiva, and Manuela Veloso. 2016. Expressive lights for revealing mobile service robot state. In *Robot 2015: Second Iberian Robotics Conference*. Springer, 107–119.

[10] Christoph Bartneck, Michel Van Der Hoek, Omar Mubin, and Abdullah Al Mahmud. 2007. Daisy, Daisy, give me your answer do!: Switching off a robot. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 217–222.

[11] Casey C. Bennett and Selma Šabanović. 2014. Deriving minimal features for human-like facial expressions in robotic faces. *Int. J. Social Rob.* 6, 3 (2014), 367–381.

[12] Cynthia L. Breazeal. 2004. *Designing Sociable Robots*. MIT press.

[13] Allison Bruce, Illah Nourbakhsh, and Reid Simmons. 2002. The role of expressiveness and attention in human-robot interaction. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'02)*, Vol. 4. IEEE, 4138–4142.

[14] Paul Bucci, Xi Laura Cang, Anasazi Valair, David Marino, Lucia Tseng, Merel Jung, Jussi Rantala, Oliver S. Schneider, and Karon E. MacLean. 2017. Sketching CuddleBits: Coupled prototyping of body and behaviour for an affective robot pet. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3681–3692.

[15] Reinhard Budde, Karlheinz Kautz, Karin Kuhlenkamp, and Heinz Züllighoven. 1992. Prototyping. In *Prototyping*. Springer, 33–46.

[16] Hoang-Long Cao, Greet Van de Perre, Ramona Simut, Cristina Pop, Andreea Peca, Dirk Lefeber, and Bram Vanderborght. 2014. Enhancing My Keepon robot: A simple and low-cost solution for robot platform in human-robot interaction studies. In *23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN'14)*. IEEE, 555–560.

[17] Ginevra Castellano, André Pereira, Iolanda Leite, Ana Paiva, and Peter W. McOwan. 2009. Detecting user engagement with a robot companion using task and social interaction-based features. In *Proceedings of the 2009 International Conference on Multimodal Interfaces*. ACM, 119–126.

[18] Caitlyn Clabaugh and Maja J. Matarić. 2016. Exploring elicitation frequency of learning-sensitive information by a robotic tutor for interactive personalization. In *25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN'16)*. IEEE, 968–973.

[19] Anki Corporation. 2017. Meet Cozmo. Retrieved August 28, 2018 from https://www.anki.com/en-us/cozmo.

[20] Christopher Crick, Graylin Jay, Sarah Osentoski, Benjamin Pitzer, and Odest Chadwicke Jenkins. 2017. Rosbridge: ROS for non-ROS users. In *Robotics Research*. Springer, 493–504.

[21] John Crinnion. 1991. *Evolutionary Systems Development: A Practical Guide to the Use of Prototyping within a Structured Systems Methodology*. Pitman London.

[22] Victor C. Dibia, Maryam Ashoori, Aaron Cox, and Justin D. Weisz. 2017. Tjbot: An open source DIY cardboard robot for programming cognitive systems. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 381–384.

[23] Carl F. DiSalvo, Francine Gemperle, Jodi Forlizzi, and Sara Kiesler. 2002. All robots are not created equal: The design and perception of humanoid robot heads. In *Proceedings of the 4th Conference on Designing Interactive Systems (DIS'02)*. ACM Press, New York, New York, 321–326. DOI: https://doi.org/10.1145/778712.778756
[24] Naomi T. Fitter, Yasmin Chowdhury, Elizabeth Cha, Leila Takayama, and Maja J. Matarić. 2018. Evaluating the effects of personalized appearance on telepresence robots for education. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 109–110.
[25] Neil Fraser. 2015. Ten things we've learned from Blockly. In *Blocks and Beyond Workshop (Blocks and Beyond)*. IEEE, 49–50.
[26] Willow Garage. 2011. Turtlebot. Retrieved from http://turtlebot.com/last visited, 11–25.
[27] V. Scott Gordon and James M. Bieman. 1995. Rapid prototyping: Lessons learned. *IEEE Software* 12, 1 (1995), 85–95.
[28] Kristof Goris, Jelle Saldien, Bram Vanderborght, and Dirk Lefeber. 2011. How to achieve the huggable behavior of the social robot Probo. A reflection on the actuators. *Mechatron.* 21, 3 (2011), 490–500.
[29] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jerome Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. 2009. Mechatronic design of NAO humanoid. *2009 IEEE International Conference on Robotics and Automation.* September 2015 (2009), 769–774. DOI: https://doi.org/10.1109/ROBOT.2009.5152516
[30] Rodolphe Hasselvander. 2017. Buddy: The Companion Robot Accessible to Everyone. Retrieved August 28, 2018 from http://www.bluefrogrobotics.com/en/buddy-your-companion-robot/.
[31] Guy Hoffman. 2012. Dumb robots, smart phones: A case study of music listening companionship. In *RO-MAN, 2012 IEEE*. IEEE, 358–363.
[32] Guy Hoffman. 2016. OpenWoZ: A runtime-configurable Wizard-of-Oz framework for human-robot interaction. In *2016 AAAI Spring Symposium Series*.
[33] Guy Hoffman and Wendy Ju. 2014. Designing robots with movement in mind. *J. Human-Robot Interact.* 3, 1 (2014), 89–122.
[34] Honda. 2018. Honda 3E. Retrieved from https://global.honda/innovation/CES/2018.html.
[35] Dominique Hunziker, Mohanarajah Gajamohan, Markus Waibel, and Raffaello D'Andrea. 2013. Rapyuta: The roboearth cloud engine. In *ICRA*. Citeseer, 438–444.
[36] Inc Jibo. 2015. Jibo. Retrieved September 18, 2017 from https://www.jibo.com/.
[37] Jinyung Jung, Seok-Hyung Bae, Joon Hyub Lee, and Myung-Suk Kim. 2013. Make it move: A movement design method of simple standing products based on systematic mapping of torso movements & product messages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1279–1288.
[38] Alisa Kalegina, Grace Schroeder, Aidan Allchin, Keara Berlin, and Maya Cakmak. 2018. Characterizing the design space of rendered robot faces. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 96–104.
[39] Michelle Karg, Ali-Akbar Samadani, Rob Gorbet, Kolja Kühnlenz, Jesse Hoey, and Dana Kulić. 2013. Body movements for affective expression: A survey of automatic recognition and generation. *IEEE Trans. Affective Comput.* 4, 4 (2013), 341–359.
[40] Jun Kato, Daisuke Sakamoto, and Takeo Igarashi. 2012. Phybots: A toolkit for making robotic things. In *Proceedings of the Designing Interactive Systems Conference*. ACM, 248–257.
[41] Hideki Kozima, Marek P. Michalowski, and Cocoro Nakagawa. 2009. Keepon. *Int. J. Social Rob.* 1, 1 (2009), 3–18.
[42] Hideki Kozima, Cocoro Nakagawa, and Yuriko Yasuda. 2005. Interactive robots for communication-care: A case-study in autism therapy. In *IEEE International Workshop on Robot and Human Interactive Communication (ROMAN'05)*. IEEE, 341–346.
[43] Jason Kulk, James Welsh, et al. 2008. A low power walk for the NAO robot. In *Proceedings of the 2008 Australasian Conference on Robotics & Automation (ACRA'08), J. Kim and R. Mahony, Eds.* 1–7.
[44] Gael Langevin. 2017. InMoov open-source 3D printed life-size robot. Retrieved August 28, 2018 from http://inmoov.fr/.
[45] Matthieu Lapeyre, Pierre Rouanet, Jonathan Grizou, Steve Nguyen, Fabien Depraetre, Alexandre Le Falher, and Pierre-Yves Oudeyer. 2014. Poppy project: Open-source fabrication of 3D printed humanoid robot for science, education and art. In *Digital Intelligence 2014*. 6.
[46] John Lasseter. 1987. Principles of traditional animation applied to 3D computer animation. In *ACM Siggraph Computer Graphics*, Vol. 21. ACM, 35–44.
[47] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation strategies for HCI toolkit research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 36.
[48] Min Kyung Lee, Jodi Forlizzi, Sara Kiesler, Paul Rybski, John Antanitis, and Sarun Savetsila. 2012. Personalization in HRI: A longitudinal field experiment. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 319–326.

[49] Iolanda Leite, Marissa McCoy, Monika Lohani, Daniel Ullman, Nicole Salomons, Charlene Stokes, Susan Rivers, and Brian Scassellati. 2015. Emotional storytelling in the classroom: Individual versus group interaction between children and robots. In *Proceedings of the 10th Annual ACM/IEEE International Conference on Human-Robot Interaction.* ACM, 75–82.

[50] Daniel Leyzberg, Samuel Spaulding, and Brian Scassellati. 2014. Personalizing robot tutors to individuals' learning differences. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction.* ACM, 423–430.

[51] LG. 2018. LG Exploring New Commercial Opportunities with Expanding Robot Portfolio. Retrieved from http://www.lg.com/au/ces2018/lg-exploring-new-commercial-opportunities-with-expanding-robot-portfolio.jsp.

[52] Giorgio Metta, Giulio Sandini, David Vernon, Lorenzo Natale, and Francesco Nori. 2008. The iCub humanoid robot: An open platform for research in embodied cognition. In *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems.* ACM, 50–56.

[53] Gajamohan Mohanarajah, Dominique Hunziker, Raffaello D'Andrea, and Markus Waibel. 2015. Rapyuta: A cloud robotics platform. *IEEE Transactions on Automation Science and Engineering* 12, 2 (2015), 481–493.

[54] Ken Nakagaki, Artem Dementyev, Sean Follmer, Joseph A. Paradiso, and Hiroshi Ishii. 2016. ChainFORM: A linear integrated modular hardware system for shape changing interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology.* ACM, 87–96.

[55] Michael I. Norton, Daniel Mochon, and Dan Ariely. 2012. The IKEA effect: When labor leads to love. *J. Consum. Psychol.* 22, 3 (2012), 453–460.

[56] Hyunjoo Oh, Mark D. Gross, and Michael Eisenberg. 2015. FoldMecha: Design for linkage-based paper toys. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology.* ACM, 91–92.

[57] Eunil Park and Jaeryoung Lee. 2014. I am a warm robot: The effects of temperature in physical human–robot interaction. *Robotica* 32, 1 (2014), 133–142.

[58] Goren Gordon, Samuel Spaulding, Jacqueline Kory Westlund, Jin Joo Lee, Luke Plummer, Marayna Martinez, Madhurima Das, and Cynthia Breazeal. 2016. Affective personalization of a social robot tutor for children's second language skills. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence.*

[59] Alison Powell. 2012. Democratizing production through open source knowledge: From open software to open hardware. *Media, Culture & Society* 34, 6 (2012), 691–708.

[60] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. 2009. ROS: An open-source robot operating system. In *ICRA Workshop on Open Source Software*, Vol. 3. Kobe, Japan, 5.

[61] Hayes Solos Raffle, Amanda J. Parkes, and Hiroshi Ishii. 2004. Topobo: A constructive assembly system with kinetic memory. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 647–654.

[62] Sekou L. Remy and M. Brian Blake. 2011. Distributed service-oriented robotics. *IEEE Internet Comput.* 2 (2011), 70–74.

[63] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009. Scratch: Programming for all. *Commun. ACM* 52, 11 (2009), 60–67.

[64] Tiago Ribeiro and Ana Paiva. 2012. The illusion of robotic life: Principles and practices of animation for robots. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction.* ACM, 383–390.

[65] SoftBank Robotics. 2014. Pepper, the humanoid robot from SoftBank Robotics. Retrieved August 18, 2018 from https://www.softbankrobotics.com/emea/en/robots/pepper.

[66] Daniela Rus and Michael T. Tolley. 2015. Design, fabrication and control of soft robots. *Nat.* 521, 7553 (2015), 467.

[67] Nicole Salomons, Michael van der Linden, Sarah Strohkorb Sebo, and Brian Scassellati. 2018. Humans conform to robots: Disambiguating trust, truth, and conformity. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction.* ACM, 187–195.

[68] Joe Saunders, Dag Sverre Syrdal, Kheng Lee Koay, Nathan Burke, and Kerstin Dautenhahn. 2016. "Teach me–show me"—End-user personalization of a smart home and companion robot. *IEEE Trans. Hum.-Mach. Syst.* 46, 1 (2016), 27–40.

[69] Adam Michael Setapen. 2012. *Creating Robotic Characters for Long-term Interaction.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[70] Syamimi Shamsuddin, Hanafiah Yussof, Luthffi Ismail, Fazah Akhtar Hanapiah, Salina Mohamed, Hanizah Ali Piah, and Nur Ismarrubie Zahari. 2012. Initial response of autistic children in human-robot interaction therapy with humanoid robot NAO. In *IEEE 8th International Colloquium on Signal Processing and its Applications (CSPA'12).* IEEE, 188–193.

[71] Elaine Short, Katelyn Swift-Spong, Jillian Greczek, Aditi Ramachandran, Alexandru Litoiu, Elena Corina Grigore, David Feil-Seifer, Samuel Shuster, Jin Joo Lee, Shaobo Huang, et al. 2014. How to train your DragonBot: Socially assistive robots for teaching children about nutrition through play. In *23rd IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN'14).* IEEE, 924–929.

[72] Abhishek Singh. 2016. Peeqo—The GIF Bot. Retrieved September 18, 2017 from https://imgur.com/a/ue4Ax.
[73] Ronit Slyper, Guy Hoffman, and Ariel Shamir. 2015. Mirror puppeteering: Animating toy robots in front of a webcam. In *Proceedings of the 9th International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 241–248.
[74] Michael Suguitan and Guy Hoffman. 2018. Blossom Public Repository. Retrieved from https://github.com/hrc2/blossom-public.
[75] Yuan Sun and S. Shyam Sundar. 2016. Psychological importance of human agency: How self-assembly affects user experience of robots. In *The 11th ACM/IEEE International Conference on Human-Robot Interaction*. IEEE Press, 189–196.
[76] JaYoung Sung, Rebecca E. Grinter, and Henrik I. Christensen. 2009. Pimp my Roomba: Designing for personalization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 193–196.
[77] Haodan Tan, John Tiab, Selma Šabanović, and Kasper Hornbæk. 2016. Happy moves, sad grooves: Using theories of biological motion and affect to design shape-changing interfaces. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, 1282–1293.
[78] Russell Toris, Julius Kammerl, David V. Lu, Jihoon Lee, Odest Chadwicke Jenkins, Sarah Osentoski, Mitchell Wills, and Sonia Chernova. 2015. Robot web tools: Efficient messaging for cloud robotics. In *IROS*. 4530–4537.
[79] Albert van Breemen, Xue Yan, and Bernt Meerbeek. 2005. iCat: An animated user-interface robot with personality. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 143–144.
[80] Cesar Vandevelde, Francis Wyffels, Bram Vanderborght, and Jelle Saldien. 2017. Do-it-yourself design for social robots: An open-source hardware platform to encourage innovation. *IEEE Rob. Autom. Mag.* 24, 1 (2017), 86–94.
[81] Kazuyoshi Wada and Takanori Shibata. 2007. Living with seal robots—Its sociopsychological and physiological influences on the elderly at a care house. *IEEE Trans. Rob.* 23, 5 (2007), 972–980.
[82] Ryan Wistort and Cynthia Breazeal. 2009. TOFU: A socially expressive robot character for child interaction. In *Proceedings of the 8th International Conference on Interaction Design and Children*. ACM, 292–293.
[83] Yunbo Zhang, Wei Gao, Luis Paredes, and Karthik Ramani. 2016. CardBoardiZer: Creatively customize, articulate and fold 3D mesh models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 897–907.

Received September 2018; revised December 2018; accepted January 2019