HERMITS: Dynamically Reconfiguring the Interactivity of Self-Propelled TUIs with Mechanical Shell Add-ons

Ken Nakagaki[1] **Joanne Leong**[1] **Jordan L Tappa**[2] **João Wilbert**[1] **Hiroshi Ishii**[1]
[1]MIT Media Lab [2]Massachusetts Institute of Technology
Cambridge, MA
[1]{ken_n, joaleong, jwilbert, ishii}@media.mit.edu [2]jtapp@mit.edu

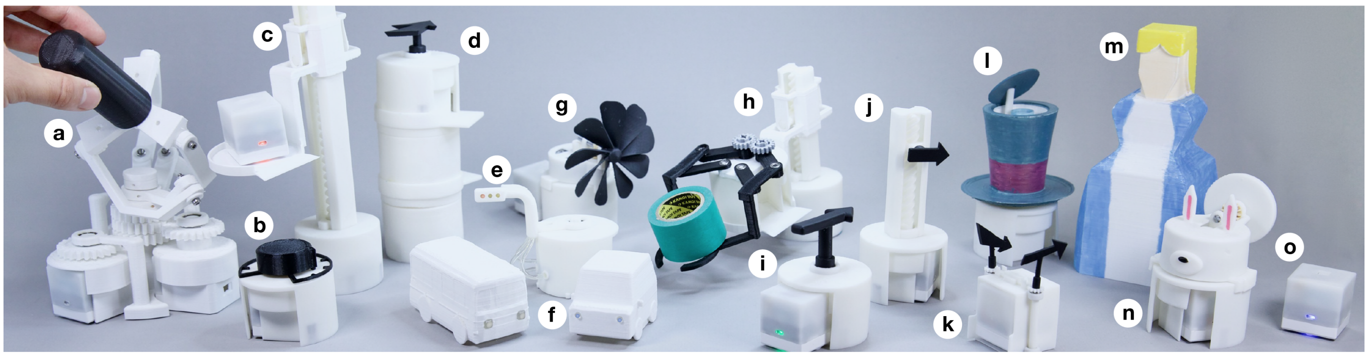


Figure 1. *HERMITS*: Mechanical shell add-ons for self-propelled TUIs (a. Joystick Shell, b. Knob Shell, c. Lift Shell, d. Rotation Accumulation Shell, e. Traffic Light Shell, f. Vehicle Shell, g. Fan Shell, h. Robotic Gripper Shell, i. Rotating Arrow Shell, j. Vertical Motion Shell, k. 2 DoF Rotation Shell, l. Mad Hatter Shell, m. Grown-up Alice Shell, n. Rabbit Shell, and o. Self-Propelled Robot.)

ABSTRACT
We introduce HERMITS, a modular interaction architecture for self-propelled Tangible User Interfaces (TUIs) that incorporates physical add-ons, referred to as mechanical shells. The mechanical shell add-ons are intended to be dynamically reconfigured by utilizing the locomotion capability of self-propelled TUIs (e.g. wheeled TUIs, swarm UIs). We developed a proof-of-concept system that demonstrates this novel architecture using two-wheeled robots and a variety of mechanical shell examples. These mechanical shell add-ons are passive physical attatchments that extend the primitive interactivities (e.g. shape, motion and light) of the self-propelled robots.

The paper proposes the architectural design, interactive functionality of HERMITS as well as design primitives for mechanical shells. The paper also introduces the prototype implementation that is based on an off-the-shelf robotic toy with a modified docking mechanism. A range of applications is demonstrated with the prototype to motivate the collective and dynamically reconfigurable capability of the modular architecture, such as an interactive mobility simulation, an adaptive home/desk environment, and a story-telling narrative. Lastly, we discuss the future research opportunity of HERMITS to enrich the interactivity and adaptability of actuated and shape changing TUIs.

UIST '20, October 20–23, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7514-6/20/10.
https://doi.org/10.1145/3379337.3415831

Author Keywords
Actuated Tangible User Interface; Swarm User Interface; Human Robot Interaction; Mechanical Shell

CCS Concepts
•**Human-centered computing** → **Interaction devices;**

INTRODUCTION
Actuated and shape changing tangible interfaces have been one of the main streams in the field of HCI to explore the role of dynamic actuation capabilities for interaction design [63, 8, 58, 61]. Through such work, researchers envision the ultimate goal of reconfigurable physical matter that is coupled with dynamic digital information, where the physical matter transforms into any shape that users wish and instruct (e.g. Radical Atoms [29], Programmable Matter [21, 81]).

Various implementation methods have been proposed to explore the enabling hardware platform that can adapt to a variety of interactions / applications, including pin-based shape displays [43, 18], actuated curve interfaces [52, 51], or swarm user interfaces [42]. While a variety of incremental efforts (e.g. resolution, additional modalities) have been made to improve the display and interaction quality of these platforms, they each have their own limitations on fidelity with regards to the shapes. This contradicts with their goal to be generic platforms that can be used for anything.

To overcome this limitation, we propose, HERMITS, a novel architectural design for self-propelled actuated tangible user interfaces (TUIs) to extend their interactions with interchangeable add-on modules which can be dynamically docked/undocked by the system. We define self-propelled TUIs as a physical interface with locomotion capabilities, that includes a system with one or more robotic devices which are previously explored widely [20, 42, 22]. The name for this project is inspired by hermit crabs, which are known to actively switch their shells throughout their lifetimes. Hence, we refer to the interchangeable physical add-ons in our system as *mechanical shells*.

In this paper, we define interaction design values of our approach for enriching the reconfigurablity and adaptability of self-propelled TUIs. While the extension of actuated TUIs with passive objects has been previously explored [68, 53, 78], our paper intends to explore a similar approach but particularly for self-propelled TUIs. By leveraging their locomotion capabilities, we introduce a novel approach to add dynamic/automated reconfiguration capabilities to actuated and shape changing TUIs.

Specifically, to explore this interaction design modality, we developed a proof-of-concept prototype. This system is based on two-wheeled self-propelled robots with an active docking mechanism, and a variety of mechanical shell add-ons that are designed to extend and convert the primitive interactivity of the devices. The docking hardware design (based on off-the-shelf robotic toys) as well as a scalable control architecture are introduced. Using the HERMITS prototype, we present applications that take advantage of the reconfigurability and adaptability of the interaction architecture. Finally, the paper discusses open research opportunities of this interaction architecture to contribute to future research into robotic, actuated and shape changing TUIs.

List of Contributions

- Introduction of the interaction architectural concept of HERMITS that augment the interactivity of self-propelled TUIs.
- Definition of the design space of HERMITS' mechanical shell add-ons for extending the interactivity.
- Implementation of the system of HERMITS based on an off-the-shelf robotic toy with added active docking capability.
- Demonstration of applications to validate the concept.
- Discussion of limitations and future work to define the open research space.

RELATED WORK

The concept and design of HERMITS are built upon previous work on reconfigurable tangible interfaces, extendable interface with physical attatchments, and self-propelled interfaces.

Reconfigurable Tangible Interfaces and Actuation

The functions of modularity and reconfigurability for Tangible User Interfaces (TUIs) have been investigated by HCI researchers [30] for TUIs to gain the capability of constructive assembly of digital model or functional artifacts with tangible and intuitive interaction [31, 23, 40, 45, 5, 60]. Recently, in the research stream of shape changing and actuated tangible interfaces [63, 8, 29, 61], there have been efforts in developing a modular architecture to improve the richness of shape representation and interaction capability of the actuated hardware [62, 49, 26, 80, 55]. Researchers explored the customizable actuated components to let users design actuated artifacts in a variety of configuration.

While these hardware design are mainly comprised of electric and computational components, the idea explored in this paper is a passive mechanicam attatchment for actuated tangible interfaces. Our approach intends to extend the interactivity of existing generic robotic devices with such passive components which doesn't require complex and costly computational electrical systems.

Extending Generic Interfaces with Passive Attachments

In HCI, there have also been approaches to extend the interactivity of generic devices/interfaces with passive objects and attachments. For example, optic interaction functionalities (display and optic sensing) were extended using optical fibers, 3D printed clear parts, or mirrored objects that act as optical guide [87, 4, 90]. Capacitive touch sensing surfaces, that are common in smartphones and laptop. were extended with passive objects containing conductive traces [36, 64, 92]. Other sensors within general purpose devices were extended to detect tangible input to attachable passive modules with use of microphones [41], accelerometers [88] or force sensitive surfaces [25]. As for commercial products, *Nintendo Labo* [56] smartly enables the tangible and embodied gaming experience by extending generic sensing capability of the gaming console using cardboard crafted attachments.

The use of passive attachments for extending interactivity of actuated tangible interface have also been explored in HCI. For example, pin-based shape display have been employed for assemblying passive magnetic blocks [68, 78], and actuating flexible crafts for storytelling and rapid prototyping [54, 15]. Among them, *TRANS-DOCK* [53] and *KineticBlocks* [68] introduced conversion of mechanical motion of pin-displays with passive mechanisms to enrich its dynamic interaction capabilities. Katakura et al, introduced extension and conversion of 3D printer's motion with mechanical attachments printed with the printer itself [34, 35]. These approaches fill the gap between users and generic actuated hardware to achieve enriched adaptable interactivity by using passive modular attachments.

Greatly inspired by such research, we intend to apply this approach to self-propelled TUIs. Self-propelled TUIs are tangible interfaces with locomotion capabilities. HERMITS leverages their capabilities and contributes to the research space through the novel exploration of self-docking functionalities as well as motion transmission designs specific to two-wheeled robotic devices.

Self-Propelled TUI and Swarm UI

TUIs with locomotion capabilities have been explored in-depth to explore a way to advance the classic tangible (graspable) interface paradigm [17, 82, 30] (where all the tangible pucks were passive). For example, tabletop actuated TUI has been explored with technical means of electro-magnet array [58, 76, 77] and self-propelled wheeled robotic system [20, 73] from 2000s. Moving across tabletop surfaces provided novel interaction opportunities with TUIs for bi-directional tangible / haptic interaction as well as the use of additional passive instruments [59].

More recently, with advancements in modular robots and the control of swarms in the field of robotics [66, 65, 67, 91], HCI researchers have expanded upon research into actuated tangible interfaces by introducing the concept of Swarm User Interface (SUI) [42]. These involve interfaces which leverage multiple self-propelled wheeled robots. By using a number of self-propelled actuated modules, researchers have investigated a way to configure the number of actuated pucks for information display [38], gestural interaction [37], haptic feedback [39], and education [57, 24]. SUIs have also been employed for activating passive modules including assembling blocks for rendering haptic proxies [93], and moving furniture and walls for rendering room-scale VR space [75]. To enhance the capability of SUI with reconfigurable interactive functionalities, HERMITS augment SUI with passive mechanical transmission attachments, and investigates the a new research space

For adding extra shape changing modality to SUIs, Suzuki et al. have explored with *ShapeBots* [79] and *RoomShift* [75] to add linear actuation modules. While some of the mechanical shell prototypes presented in HERMITS overlap with the interaction modalities for shape changing SUIs, our novel contribution is the possibility for self-propelled interfaces (including SUIs) to selectively attach and detach to modules, enabling more flexible functionalities. This is not possible with *ShapeBots*, which is constrained to its original mechanical design. To investigate this reconfigurability, we introduce a specific docking joint design for wheeled robots.

HERMITS

In this section, we outline the overall design of HERMITS as well as the benefits of using mechanical shells for interaction.

HERMITS is a concept and design for an interactive modular system for self-propelled tangible user interfaces. As shown in Figure 2, the components of the system include the (1) self-propelled TUIs (robots), (2) mechanical shells, (3) an interaction stage, and (4) a computer to control the system. In this paper, we refer to *robots* as self-propelled TUIs that activate mechanical shells in HERMITS, and we refer to the combination of one or more robots and a shell as a *unit*.

In the system, robots can selectively attach/detach to the mechanical shells by locomoting across the stage. While the robots themselves support generic interactions to users (as in [20, 42, 73]), a variety of mechanical shells allows the system to provide specific and reconfigurable interactivity to the users dynamically and on-demand.

Regarding the number of robots, some of the mechanical shells in HERMITS can be docked and controlled with only a single robot, while others need to be activated by multiple robots. Hence, it is not necessary that HERMITS have to have a swarm of robots. For example, one robot activating a variety of

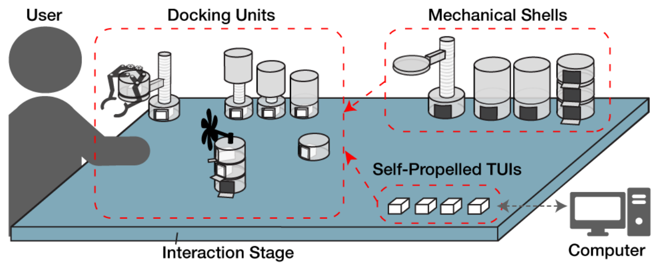


Figure 2. Overall design of HERMITS.

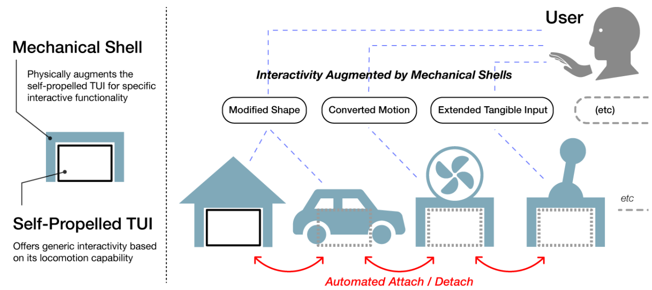


Figure 3. The concept of mechanical shell. A mechanical shell can be attached and detached to self-propelled TUI. The shell can thereby transform the interactivity of the device including shape, motion and other tangible I/O.

mechanical shells one after another is an architecture option in the design of HERMITS. As a design space, we also consider having a swarm of robots as with SUIs to be an exciting option. This opens the door to interactivity with a collective and pushes the versatility of the system.

Mechanical Shell: Definition and Interaction Design Roles

The mechanical shell is a critical component in the design of HERMITS. We define the mechanical shell as *"an external passive attachment to self-propelled TUIs that can be automatically attached or detached."*

The diagram on Figure 4 illustrates the primitive relationship between a self-propelled TUI and mechanical shells. The shell is an attachment that can modify, convert or extend the interactive functionalities of the actuated TUI. For example, mechanical shells can modify the interactive property of 'shape', convert the locomotive 'motion' via transmission mechanisms, or extend other I/O capabilities by taking advantage of installed sensors and actuators in the device. The mechanical shells can be designed with combinations of these tangible augmentation capabilities.

Mechanical shells can serve multiple purposes as shown in Figure 4. By reconfiguring its shape and converting mechanical I/O motion, the mechanical shells can enhance the dynamic physical affordances of the system and enrich its interaction space. For example, this enables the system to be used for tangible and haptic controllers (Figure 4a) as well as for shape representation (Figure 4b). Reconfiguring the shell with expressive and iconic shapes and motion would enhance its communication capability to users which can be great tool for storytelling (Figure 4c). Furthermore, the motion conversion capability allows the self-propelled TUIs to extend their

Figure 4. Overview of interaction design purposes and benefits of mechanical shells from instances of HERMITS (a. Tangible and Haptic Controllers, b. Data Physicalization / Tangible Representation, c. Storytelling with Expressive Shapes and Motion, and d. Extended Robotic Manipulator and Locomotion)

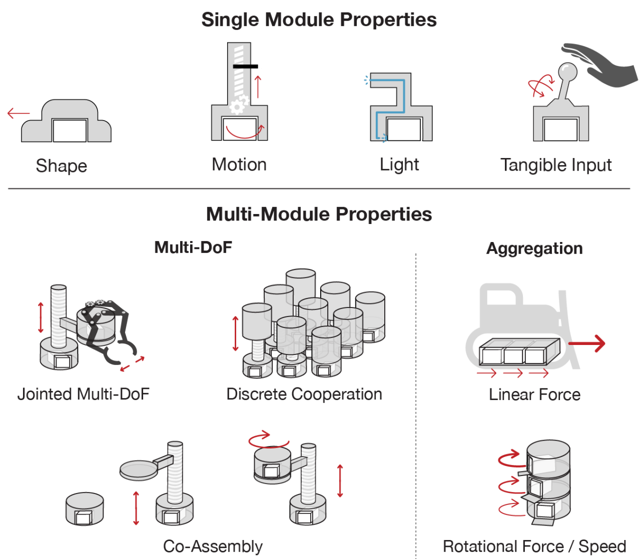


Figure 5. Design space for primitive properties of mechanical shells.

robotic functionalities to affect the physical environment, such as with the manipulator and locomotor units pictured in (Figure 4d). These will be highlighted by the application section of this paper. In summary, the benefit of the mechanical shell concept is not only that they can provide each capability described above, but that they allow for the system to dynamically adapt and be reconfigured to flexibly support interactions.

DESIGN SPACE OF MECHANICAL SHELLS

Many primitive interaction properties of self-propelled TUIs can be extended and converted using mechanical shells that are designed either for single or multiple robots. Figure 5 shows these primitive design properties which are demonstrated in mechanical shell examples in this paper.

Single Module Design Properties

The mechanical shells can extend or convert properties for a single robotic module:

Shape: The most primitive and simple property of a mechanical shell is its shape. With HERMITS, each robot is capable of changing its shape dynamically by exchanging the shell it 'wears.' As these shells can be prefabricated, this allows the robots to take on different high-fidelity shapes. This is in contrast to prior work into SUIs, which feature modules that share the same shapes [42].

Motion: The motion that each module can perform can be modified or extended with the use of a mechanical shell with embedded transmission mechanisms. While the robot of a SUI is capable of rotation and translation on a 2D plane, mechanical shells, for example, can convert lateral movement to vertical movement as in the vertical motion unit presented in (Figure 1j).

Additionally, the rotational motion generated by robots can be converted by shells to ramp up their speed or torque by using different gear ratios; the fan shell (Figure 1g) ramps up the speed of rotation to generate wind, while the walking shell (Figure 4d right) ramps up the torque allowing the unit to push and lift itsself up to traverse uneven surfaces.

Light: Mechanical shells can also augment optical properties. In HERMITS, the shells can redirect or forward the light from an LED embedded into the base of each robot to other custom locations via optic guiding materials (e.g. optical fibers), similar to [87, 4, 90]. In this way, the light can be repurposed for different modules, such as for the headlamps of a vehicle module, or the lights on a traffic light module (see Figure 1e and f).

Tangible Input: Mechanical shells can also extend the sensing capabilities of the system for interaction purposes. For example, tangible input to a joystick can be interpreted collectively by multiple robots' sensors in order to detect the rotation and direction of the joystick (Figure 1a).

Multi-Module Design Properties

As HERMITS encompasses the possibility for multiple robots to be used, taking inspiration from SUIs, there are also specific design properties pertaining to a collection of units. Mechanical shells can serve to combine multiple modules that can impact how they coordinate their movements:

Multi-DoF: A single shell can house and leverage several robots' individual actuation capabilities. For example, a robotic gripper shell on Figure 1h can be controlled with the use of multiple robots, enabling multiple degrees-of-freedom of motion. This is given the term *Jointed-DoF* (see Figure 5). Multiple degrees of freedom is also enabled through the *discrete cooperation* of units. In this manner, units can work together to present collective shapes, similar to [79], but with the added possibility to exchange the shell. *Co-assembly* also involves multiple degrees of freedom. The lift shell (Figure 1C) demonstrates this feature by extending the mechanism of the vertical moving shell such that it can lift other robots and help them dock into other mechanical shells.

Force and Speed Aggregation: A single shell can house multiple robots to attain a greater force than that generated by a single robot; the linear force of a dozer unit for instance features the aggregate force of three robots (Figure 4b). Furthermore, greater rotational speeds and forces can be generated when a shell allows multiple robots to be stacked and to rotate on-top of each other in the same direction (Figure 1d).

IMPLEMENTATION OF HERMITS

In this section, we introduce the implementation of our proof-of-concept prototype of HERMITS.

Overall System Design based on toio

The overall system of HERMITS is based on the off-the-shelf robotic toy system, *toio*^{TM} [14], developed by *Sony Interactive Entertainment (SIE)* that features two-wheeled robots (only sold in Japan [as of July 10th 2020] with price of approx. 40 USD). We use toio systems as the underlying self-propelled TUIs (referred to as *robots* in HERMITS). While the hardware is sold as a gaming console with dedicated game cartridges, SIE releases its API for maker and researcher communities to design interactions with the robots [12]. Based on their API, we have built a Raspberry Pi-based hierarchical control architecture as shown in Figure 6. A computer (either Windows or Mac) takes the central control of the system (based on Processing code), while a number of Raspberry Pi micro-controllers were used for connecting and controlling individual robots through Bluetooth (based on Python code). As for control specs, the Python program on Raspberry Pi operated at 100 fps and Processing programs on the computer operated at 30 fps.

The reason for taking this hierarchical control architecture is due to the limitation of the Bluetooth on a computer to connect to toio robots simultaneously. By using a Raspberry Pi, which is a relatively cheap computer with Bluetooth control and wired Ethernet connection capabilities, the control system can easily scale up to control many toio robots. The number of toio robots we have tested for robust connection per one Raspberry Pi is 5, and overall we have tested controlling up to 70 robots using 14 Raspberry Pis simultaneously. This software platform for the control architecture itself is useful for prototyping SUI interactions using off-the-shelf products as components.

A toio robot has an embedded downward-facing optical sensor (camera) to detect its position on a custom designed toio mat with printed patterns. This is similar to AnotoPen [16] technology. Toio mats feature printed patterns that encode absolute localization information so that a robot can detect its position and orientation when placed on the mat. The resolutions of position and orientation detection offered by the API were 1.42mm and 1 degree, respectively. Each mat has a thickness of 0.1 mm. Up to 12 mats can be tiled together to form a maximum area of 1260 x 1188 mm (*toio Mat for Developers* as in [13]). In our implementation, we cut the mats to cover an area of 610 x 920 mm to develop applications that operate within an arm's distance of a user. The remaining parts of the mat are used to augment some shells to enable closed-loop control for

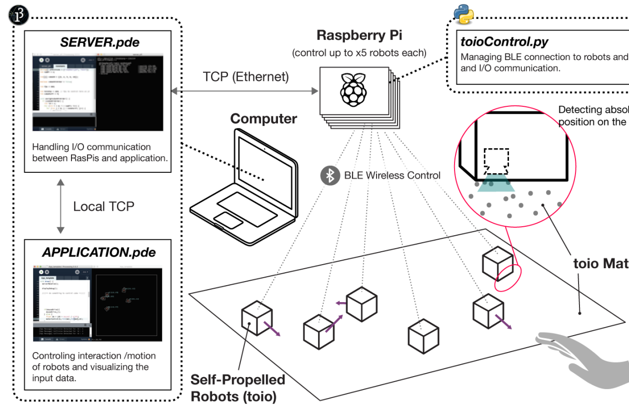


Figure 6. System overview and control architecture.

automated docking and for the detection of user-input. This will be elaborated upon in a later section of the paper.

toio modification for HERMITS

We have made three hardware modifications to the toio robots specifically to meet the design requirements of HERMITS as shown in Figure 7 and 8.

The first modification is for the active docking mechanism. In order to achieve the active docking capability needed for the HERMITS concept, a technique needed to be developed to connect the robots to passive mechanical shells. This was done by adding a micro-linear actuator to each toio. To avoid the need to add an additional microcontroller to each robot, we hacked the toios by repurposing their piezo control capabilities. The oscillation pin was soldered to the servo motor, such that generated signals could be used as a PWM signal to control the motor. With this, by using sound control functionality of toio API, the system was able to control the vertical pin up and down (7 mm stroke, 0.15 sec/cm speed, and 0.24 kg-cm torque) [2]. To contain this micro linear actuator, we replaced the original plastic container of the toio with a custom 3D printed case. The modified toios were 32 x 32 x 36 mm in dimension. As shown in Figure 7 and 8, this vertical pin is positioned at the center of rotation for a toio when two wheels were rotated in opposite directions.

The second modification is the addition of magnets for robust torque transmission to the mechanical shells. Our earlier prototype of HERMITS with original toio configuration revealed that the wheels can easily slip when actuating a mechanical shell, especially for the ones that require a relatively high torque. To increase the friction of the wheels, we added two disc-shaped neodymium magnets (6mm diameter, 1.5mm thickness) to the bottom of each toio with a 3D printed mount and placed a 1mm thick ferromagnetic iron sheet underneath the 0.1mm thick toio mat. This modification was inspired by other SUI hardware for increasing the horizontal force for haptic feedback [39] and for locomotion on a vertical plane [38]. With this, we increased the linear torque from approximately 0.8 N to 3 N. We measured the linear locomotive speed to be approximately 24.7cm/s.

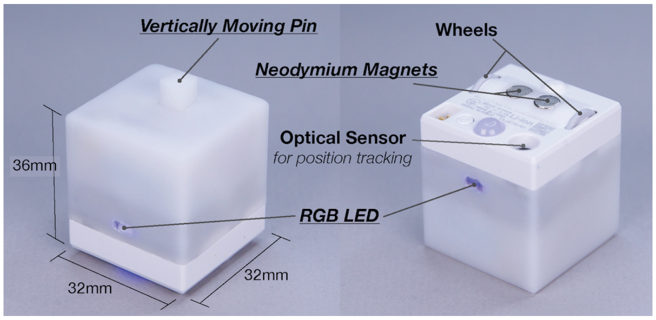


Figure 7. The wheeled robots developed for HERMITS based on toio, an off-the-shelf robotic toy. (Components with underlines are modified feature from original toio hardware.)

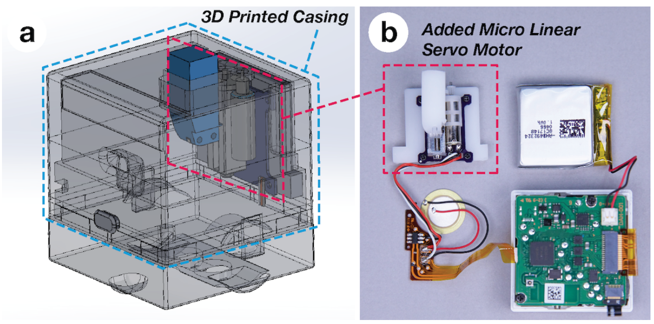


Figure 8. a. 3D model of the internal structure of toio with the additional Linear Servo Motor, b. Added Micro Linear Servo Motor soldered on Piezo signal wires.

Lastly, we made a modification to the LEDs for optic transmission to our shell prototypes. While the original toio robot had single RGB LED that could be controlled with the API, it faced downwards and was difficult to access by the mechanical shells for optic transmission. By tearing down the plastic casing, we noticed that we could place a hole on the front side of each toio for the internal LED light to pass through, such that it could be transmitted to shells (Figure 7).

Design of Mechanical Shells

Next, we describe the design of mechanical shell prototypes for HERMITS. We fabricated the mechanical shells mainly with FDM and SLA 3D printers. Additional parts including, bearings and magnets were embedded for some of the shell designs as detailed below.

Primitive Types of Mechanical Shell for Motion Transmission

We designed four basic primitive designs of mechanical shells to extend the motion of two-wheeled robots in different ways. These four primitives serve as the basis for the variety of mechanical shell prototypes featured in this paper. While each primitive was designed for a single robot, these primitives can be combined to achieve more complex mechanical shell designs that can be controlled by multiple robots.

These basic designs are categorized in how the available activating motion of actuated modules are transmitted. Figure 9 describes the mechanism in a simplified section-cut diagram, 3D appearance in the CAD, an example shell prototype, and specific characteristics for each basic shell design.

Simple (Static Shape) Shell: The most primitive deign of the mechanical shells is the one that modifies only the shapes of the robot. This shell simply contains a docking slot as well as lips on the bottom to hold the toio even if the units are picked up by users. While placed down on a mat, this design allows the robot to touch the platform for locomotion and position detection. This simple shell also became a basis for other modules with moving parts.

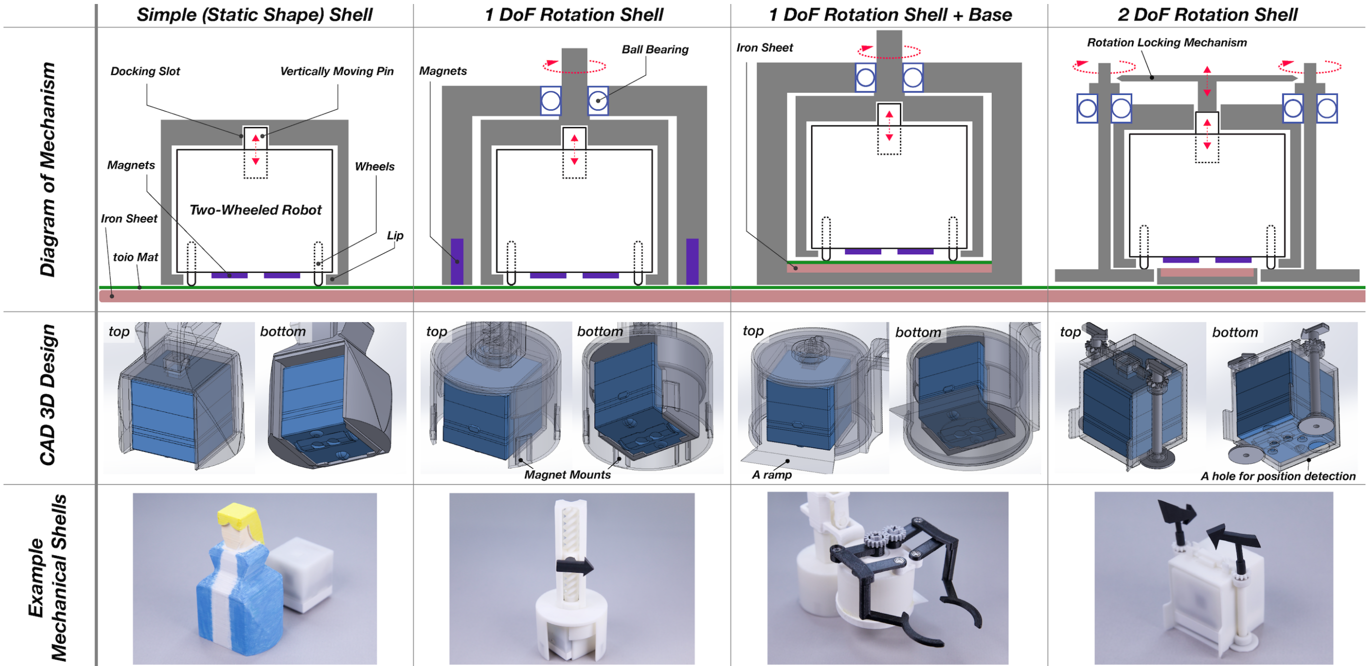
1 DoF Rotation Shell: This primitive shell is composed of two solid components (an internal and external part respectively) that are connected through a single ball-bearing. The external part is designed to house the internal one. The internal part has a similar design to the simple shell and can freely rotate within the external housing when the wheels of the robot are activated to spin in opposite directions. With this rotation, this shell can become a motor to activate the mechanism attached on top. Additionally, as the wheels are designed to be touching the platform surface, this shell design allows the unit to move across the surface when the wheels are controlled to move in the same direction.

1 DoF Rotation Shell with Base: This shell design incorporates an additional *base* under the 1 DoF rotation shell design. Unlike the 1 DoF rotation shell, the addition of the base means that the wheels always have a surface to drive against such that the unit can be actuated even when its lifted up. The base of this shell has a ramp such that a toio can smoothly drive into it and dock. The base contains an iron sheet to improve the traction of the wheels and a toio mat to enable the closed-loop control of the robot when it is inside this shell. Because this module can be lifted, it can, for example, be combined with a vertically moving module to act as a robotic gripper with z-axis control (Figure 1h). As the bottom base is enclosed in this shell design, the robot docked to this shell won't be able to detect the absolute position encoded on the mat. However, this can be resolved by combining this unit with another.

2 DoF Rotation Shell: Unlike the other shell primitives, the 2 DoF Rotation Shell utilizes the fact that each wheel of a toio can be independently controlled. This shell features two rotating plates that can each be spun by a wheel. These plates are locked to allow the toio to drive into the shell, and are unlocked to enable robotic actuation of the shell. The shell incorporates a special mechanism that can be triggered by the robot's vertically moving pin to control the locking and the unlocking of the plates. When the vertical docking pin is raised, the plates are free to rotate. When it is lowered, the plates are locked such that the robot can drive in or out of the shell. On the bottom of the shell, a hole was positioned so that the optical camera of the robot can continue to identify the position and orientation of the unit on the mat. However, the robot is not able to detect the degree of rotation of the rotating plates that it controls.

Mechanism Design Approach

Based on these simple primitive mechanical docking design, we have designed and implemented a variety of mechanisms



Characteristics	Simple (Static Shape) Shell	1 DoF Rotation Shell	1 DoF Rotation Shell + Base	2 DoF Rotation Shell
DoF of Motion to Activate	*0DoF*	***1DoF***	***1DoF***	***2DoF***
Movable on the Plane	***YES***	***YES***	*NO*	*NO*
Mat Position Detectable	***YES***	***YES***	NO	***YES***
Actuatable when Lifted	—	*NO*	**YES**	***YES***
Capable of Closed Loop Control	—	**YES**	**YES**	*NO*

Figure 9. Design and mechanism of four basic types of mechanical shells implemented for HERMITS, and their characteristics.

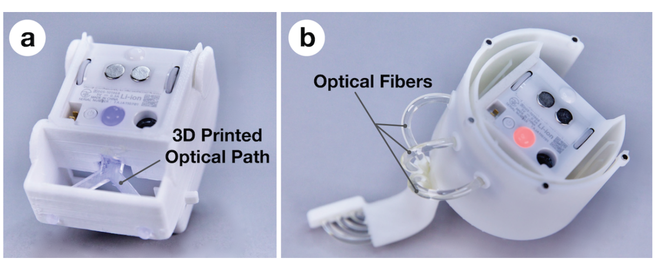


Figure 10. Examples of mechanical shells that extend the illumination of LED (a. 3D printed optical path for vehicle shell, b. three optical fibers for multi-channel output in traffic light shell.

to convert and translate the motion by the robots using *SOLIDWORKS*. Some of them were simply designed and 3D printed (e.g. the vertical motion shell on Figure 1j), while some other were composed with *LEGO* gears for quick prototyping and reliable, smooth and robust motion transmission (e.g. the fan shell on Figure 1g, and the walking shell on Figure 4d).

Optical Transmission

In addition to mechanism design, we integrated optical paths to transmit light from an RGB LED in each robot to different parts of the shells inspired by [87]. Figure 10a is an example of 3D printed transparent optical path designed to fit into a vehicle shell to enable head lamps (printed with transparent resin of Form3 SLA printer [19]). While our employed robots only had single RGB LED, using multiple optical paths with combination of motion enabled the shell to exhibit the light in multiple locations. Figure 10b presents the underline mechanism of traffic signal shell that incorporates three optical fibers for the robots to selectively illuminate by adjusting the rotation angle.

Tracking Mechanical Shells with AR Markers

While the toio robots were able to identify their own location on the toio mat, for us to explore and develop certain interactive functionalities (for automated docking and detecting controller inputs), we have implemented a way to track the location of mechanical shells through a computer vision technique. As shown in Figure 11a, we have designed AR marker modules to be magnetically attached on some of the mechanical shells. To track them, we placed a USB camera above the interaction stage facing down. Figure 11b shows the actual footage from the USB camera with graphically overlaid tracking information that is assigned to each AR marker associated to each mechanical shell. We chose to use detachable marker modules to quickly explore and demonstrate the concept of HERMITS. In the future, more advanced tracking techniques, such as using sensors underneath the interaction stage [84, 46, 64] should be incorporated for a robust, accurate, and compact system.

As shown in this figure, we programmed the software in the way that individual markers are associated with the relative docking position and orientation of the mechanical shells. With this, the system was able to control the robots movement

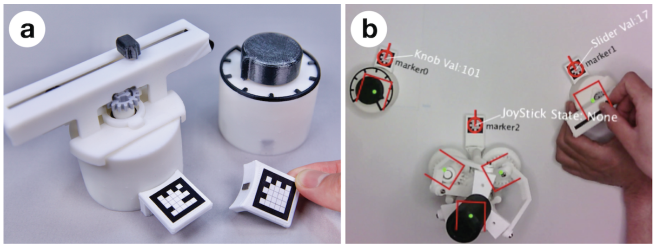


Figure 11. Tracking of mechanical shells (a. Magnetically attachable AR markers for tracking mechanical shells, b. the actual software view of the tracking.)

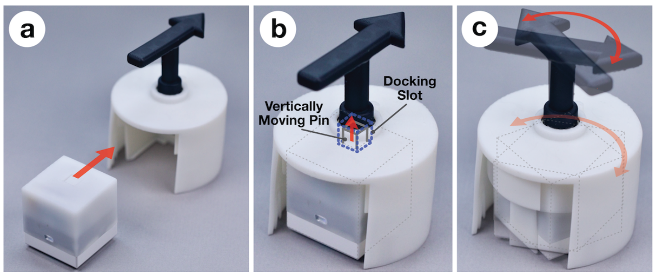


Figure 12. Docking process (a. A robot moving in from the entry point of a mechanical shell, b. a robot docking by moving its pin, c. the robot activating the mechanical shell.)

to be automatically docked to the associated mechanical shells dynamically. This marker tracking capability also allowed the system to capture controller inputs from the knob, slider and joy-stick shells by decoding the relative orientation between the robots and the mechanical shells.

Manual Control with JoyStick-based Controller

In addition to automated docking control, we have developed an Arduino-based controller with three joy-sticks to manually steer three robots simultaneously. As the prototyping process in this paper heavily focused on the iteration and fabrication of the different mechanical shell designs, this allowed us to quickly test and validate our prototypes of mechanical shells with the controller, without writing code for each shell.

Docking Procedure

As for the docking procedure, each mechanical shell has predefined entry points for each robot to move-in and dock (Figure 12a), similar to *RoomShift* [75]. After the robot is in the docking position, the robot actuates the vertically moving pin upwards to make the robust connection to the slot of the shell (Figure 12b), then it can switch to the activation/interaction mode (Figure 12c). The undocking procedure would be in the opposite process, while the robots have to adjust its orientation to secure the exit path when docked into mechanical shells with double layered shells (e.g. 1 DoF shell). The tracking of the mechanical shell as well as the robots are required to automatically moderate this procedure.

Opensourcing the Code and CAD

We opensource the following programs and design. We believe this can contribute to the community the following: A. a generic and scalable Swarm UI interaction platform based on off-the-shelf hardware (toio and Raspberry Pi), and B. a toio docking modification design as well as primitive designs for mechanical shells that other people can use to design their own [1].

- Python software for Raspberry Pi control
- Processing software (Server App + Basic Template App)
- SOLIDWORKS 3D model of the primitive shell designs and modified toio casing

APPLICATIONS

In order to demonstrate the strengths of HERMITS, we present it's applications to fulfill three different use-cases. We explain how the HERMITS's architecture and different types of mechanical shells are leveraged for each.

Interactive Mobility Simulation

TUIs have been utilized to facilitate multi-stakeholder interaction with urban mobility simulations. The tangibility of such tools encourages city governors and citizens to plan and discuss the future city together [1], while the output information remains only as projected graphics. As the technology and service of self-driving cars are drawing increasingly more attention recently, there are demands in tangible actuated tools to emulate the future of mobility. Tangible tools and real physical motion can enable users to gain greater intuition of the dynamic motion of a simulation, and also empowers users to intervene with the system, for instance, by intuitively moving a vehicle from one location to another.

We designed an application for HERMITS to support a traffic intersection simulation (see Figure 13 left). The actuated TUI enables vehicle motion to be simulated and for the location of different entities (e.g. buildings, traffic lights, as well as different vehicles, such as cars and busses) can be automatically adjusted to match the simulated data and view. They can simultaneously be controlled by users' for real-time tangible intervention with the simulation model. The mechanical shells in this application feature optic transmission of light and automated docking. The light is used to show the break-lamp of cars, as well as to control the color of the traffic signals at the intersection. This feature is critical to allow users to understand the intricate behaviour of the traffic at the intersection. The importance of automated docking to shells is evident in this application, as the types of vehicles in the simulation can be consistently changing and updating. Finally, the interaction stage helps to reveal and conceal different entities as the robots transform their identities by donning different shells. The interaction stage permits modules to switch between on-stage and back-stage locations. This allows for robots and shells that are not needed to be fully of out-of-view from the user (behind the wall on Figure 13e), to match the simulated data.

Since urban simulation data often features abstract metrics, HERMITS can also be used for complementary data physicalization. They can be used for interactive and tangible bar charts [32] for instance to indicate the amount of traffic on

[1] https://github.com/mitmedialab/HERMITS_UIST20

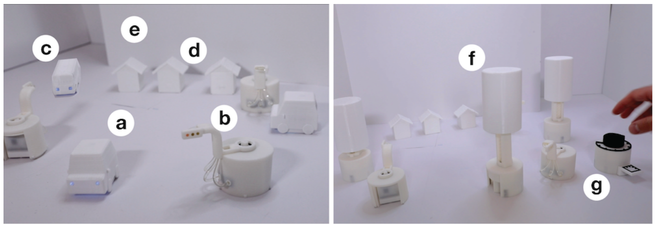


Figure 13. Use scenario of tangible interactive mobility simulation (a. car shell, b. traffic signal shell, c. bus shell, d. building shells, e. a wall to hide non-active shells and robots, f. bar chart shells, g. knob shell.)

each road (see Figure 13 right). HERMITS can also be leveraged to provide users with tangible controllers in the form of knobs and sliders to control the simulation parameters.

Adaptive Environment/Desktop

While the previous application largely demonstrated the use of HERMITS to interact with digital information, we believe that they can also be used for enriching physical environments in our everyday lives with their high level of reconfigurability and adaptability.

Figure 14 shows how HERMITS can contribute towards actuated physical desktop environments [69, 3]. HERMITS can sequentially switch the interaction mode by reconfiguring their mechanical shells. On this desktop, there are several shells including a fan shell for temperature adjustment, the dozer shell to provide strong haptic feedback for 'push' notifications, and two vertical motion shells to adapt the height of a display.

Beyond the desktop environment, we believe that the broad design space for mechanical shells can also be applied to dynamic physical environments in general [72, 85]. In a daily living environment, robotic vacuums are now commonly deployed. As this type of hardware share similar technical components with the wheeled system used in HERMITS, the idea of mechanical shells may give such robotic devices a way to handle extensively versatile tasks at home and in the office (e.g. reconfiguring furniture, climbing stairs, and moving obstacles away with grippers).

Storytelling

The strength of HERMITS is also highlighted through a storytelling application. Taking inspiration from *Alice in Wonderland*, we use modules to represent the scene where Alice chases after the rabbit that holds an iconic spinning watch (Figure 15a). The units are able to play out the chasing motion between the characters across the stage. The mechanical shells bring these characters to life by giving them high-fidelity forms that closely resemble the characters. Similar to the Interactive Mobility Simulation, the stage enables the robots to change their shells out-of-view from the user and return to the visible stage with their correct appearance to match the story (Figure 15b). This mimics how actors move on and off-stage to change costumes and play different characters to support the story. With this interchangeability function, Figure 15 c and d shows following scenes which represent the Mad Hatter shell dancing as a tea-cup weirdly appears from his hat, and Alice's body growing in size by switching to a larger shell. As

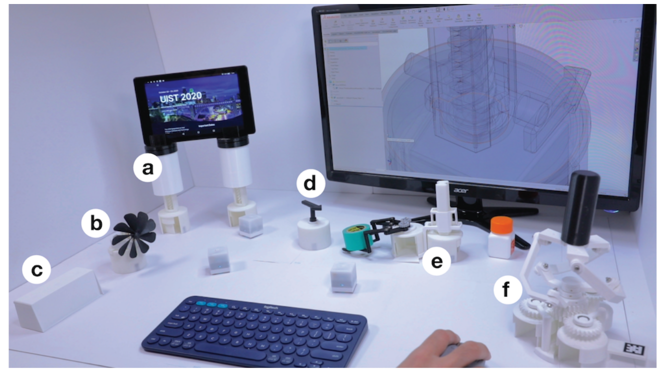


Figure 14. Adaptive desktop application. (a. Height adjusting screen shell, b.fan shell for cooling down, c. dozer shell for push notification, d. arrow shell to get users attention to specific direction, e. gripper shell to hand objects to users, and f. Joy-stick shell for controller to be used in specific software on the computer.)

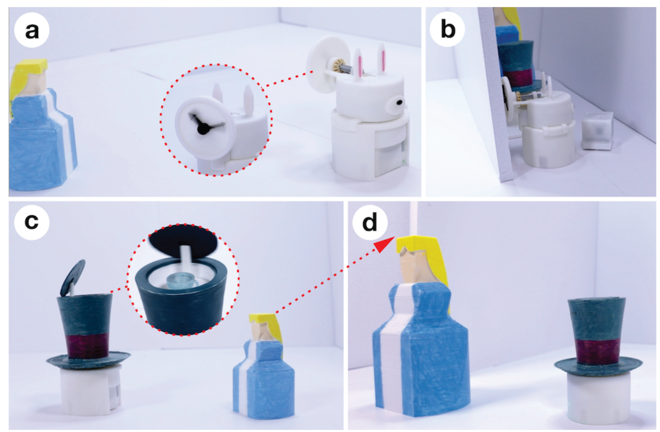


Figure 15. Demonstration of storytelling with *Alice in Wonderland* scenario (a. Alice chasing bunny holding spinning watch, b. robots changing the shell in the back-stage, c. Mad Hatter dancing in front of Alice, d. Alice with grown body by switching the shell.)

the characters can be controlled by users as well, this application highlights HERMITS' ability to be used as a tangible educational tool for creating animated objects [62, 54, 57, 24].

DISCUSSION AND FUTURE RESEARCH SPACE

In this section, we elaborate on future research opportunities based on the concept of HERMITS that extend the interactivity of self-propelled TUIs, and outline the current limitations of our implementation (see Figure 2).

Self-Propelled Device

Although we explored the use of wheeled robots as the underlying self-propelled TUI, a variety of other locomotive hardware can be utilized based on the idea of HERMITS. Such hardware may include levitating swarm interfaces (e.g. drones) [22, 6], actuated curve interfaces (e.g. serpentine robots) [52, 51], legged robots [71, 11] or wheeled pin-displays [70].

Accordingly, a number of properties of the robotic hardware can expand the interaction design possibilities. While HERMITS is based on hand-scale devices, we can explore sizes

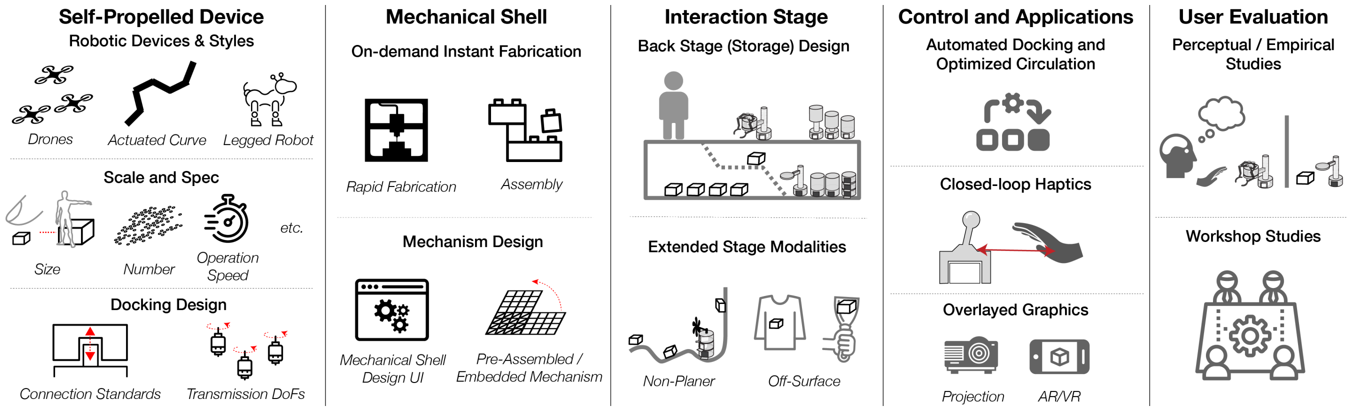


Figure 16. Future research opportunities.

to accommodate a range of scales from nano-scale robots for fingertip interactions [7] to vehicle-sized systems for bodily interactions [83]. Other design parameters may include the number of modules, the operation speed (for general locomotion and docking), etc. Additionally, for project HERMITS, there are research opportunities specific to the docking design. These opportunities include defining *connection standards* between different kinds of robots and shells, and exploring more motion transmission mechanisms to cater to different types of hardware.

Mechanical Shells
In HERMITS, the shells were prefabricated to meet the demands of the different scenarios, and were stored out of sight when not in use. However, we believe that in the future, instant or rapid fabrication methods [48, 89] as well as methods to rapidly assemble and disassemble [68, 78] structures can be useful additions to the overall system architecture. On a lower-level, further research into the design and fabrication of motion conversion mechanisms [28, 74, 9] is promising. This could improve the design process of shells for end-user applications, and enable them to be more compact and efficient.

Interaction Stage
In HERMITS, we demonstrated the idea for an interaction stage with on-stage and back-stage sections to accommodate the need for concealing shells that were not actively in use. Costume changes in theater and fashion shows inspired this design. Extending this metaphor, we can consider the way to store and circulate the shells in the background while users are interacting with the foreground devices. This may help to accelerate transitions. Taking it a step further, it may also be possible to circulate fabrication devices to facilitate on-demand construction and distribution of shells. Additionally, the design of the stage can also be extended to non-uniform or non-planar environments, such as on uneven or vertical surfaces. We can also consider stage design for wearable contexts [10, 33] or for handheld device use-cases [94].

Control and Applications
In HERMITS, we demonstrate a basic implementation for controlling and coordinating the movement of modules, including the use of AR markers to support automated docking. However, the control system can be refined for greater practicality, speed and robustness. Ultimately, the software system should adapt the number of modules that are present on-stage to match the users' needs, and coordinate the timing of movements (i.e. moving and docking) to facilitate an optimal user experience. Additionally, enabling bi-directional force control for individual shells [47, 50] would open up novel research possibilities in reconfigurable haptic feedback experiences. This requires precise mechanical design for the shells, high-speed control, and an adaptive PID control. With regards to applications, we see the possibility to combine the use of the system with graphical displays or projection mapping. This opens the possibility to use the system for data visualization as well as AR/VR [44, 70, 86, 93].

User Evaluation
Lastly, while this paper validates the novel approach to supporting new interactions with a proof-of-concept implementation [27], performing extended validations, such as user study-based evaluations, remains an opportunity. Future studies could include an empirical study to evaluate the affordances of our approach, which could cover how users perceive robots changing theirs shell back-stage and reappearing, or a workshop-based study to ask designers to create their own shells to explore a wider range of applications.

CONCLUSION
This paper proposed HERMITS, a novel approach for dynamic reconfigurable TUIs to augment the interactivity of self-propelled TUIs with passive attachments, named mechanical shells. We demonstrated this concept with a proof-of-concept implementation based on off-the-shelf wheeled robots. Our applications demonstrated the use of reconfigurability and adaptability in our approach to dynamically provide different interaction opportunities to users. While this paper rather focused on the variety of shell designs and the docking implementation, we outlined the future research directions beyond the hardware design, including possibilities to enhance the interaction environment and advance the computational control. We hope this paper inspires the field to open up a new way to fuse actuated computational devices with passive mechanical modules to enrich our digital and physical interactions with greater reconfigurability.

ACKNOWLEDGMENTS
We appreciate Dr. Alexis Andre's (from Sony CSL) sharing of knowledge and support with the toio hardware, which became the technical basis for this paper. We greatly thank John Vincent and Dr. Pietro Buttoro from the Ford Motor Company for their insightful inputs specifically related to the mobility simulation application. We thank the toio maker community in Japan for their inspiration and technical documentations which helped us to build the architecture introduced in this paper. We thank Dr. Artem Dementyev for his feedback on this paper. We also thank Hila Mor for the support in creating diagrams, and Jack Forman, Deema Qashat and the members of the Tangible Media Group for many fruitful discussions and their feedback. Lastly, we thank Zhipeng Liang, the first-author's room-mate and lab-mate, who co-built and shared a homelab. This enabled most of the fabrication and documentation for this research to be completed at home.

REFERENCES
[1] Luis Alonso, Yan Ryan Zhang, Arnaud Grignard, Ariel Noyman, Yasushi Sakai, Markus ElKatsha, Ronan Doorley, and Kent Larson. 2018. Cityscope: a data-driven interactive simulation tool for urban design. Use case volpe. In *International conference on complex systems*. Springer, 253–261.

[2] Amazon. *Micro Linear Servo Motor*. `https://www.amazon.com/Digital-Ultra-Linear-Aircraft-Tester/dp/B07KKCPC26` - accessed July-10-2020.

[3] Gilles Bailly, Sidharth Sahdev, Sylvain Malacria, and Thomas Pietrzak. 2016. LivingDesktop: Augmenting desktop workstation with actuated devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 5298–5310.

[4] Patrick Baudisch, Torsten Becker, and Frederik Rudeck. 2010. Lumino: tangible blocks for tabletop computers based on glass fiber bundles. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1165–1174.

[5] Ayah Bdeir. 2009. Electronics as material: littleBits. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*. 397–400.

[6] Sean Braley, Calvin Rubens, Timothy Merritt, and Roel Vertegaal. 2018. GridDrones: A Self-Levitating Physical Voxel Lattice for Interactive 3D Surface Deformations.. In *UIST*. 87–98.

[7] Adriano Cavalcanti, Bijan Shirinzadeh, Mingjun Zhang, and Luiz C Kretly. 2008. Nanorobot hardware architecture for medical defense. *Sensors* 8, 5 (2008), 2932–2958.

[8] Marcelo Coelho and Jamie Zigelbaum. 2011. Shape-changing interfaces. *Personal and Ubiquitous Computing* 15, 2 (2011), 161–173.

[9] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational design of mechanical characters. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.

[10] Artem Dementyev, Hsin-Liu Kao, Inrak Choi, Deborah Ajilo, Maggie Xu, Joseph A Paradiso, Chris Schmandt, and Sean Follmer. 2016. Rovables: Miniature on-body robots as mobile wearables. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 111–120.

[11] Boston Dynamics. *Spot*. `https://www.bostondynamics.com/spot` - accessed July-10-2020.

[12] Sony Interactive Entertainment. *toio™ Core Cube Specification (in Japanese)*. `https://toio.github.io/toio-spec/` - accessed July-10-2020.

[13] Sony Interactive Entertainment. *toio™ Mat for Developers (in Japanese)*. `https://toio.io/news/2020/20200423-1.html` - accessed July-10-2020.

[14] Sony Interactive Entertainment. *Toy Platform toio™*. `https://www.sony.net/SonyInfo/design/stories/toio/` - accessed July-10-2020.

[15] Aluna Everitt and Jason Alexander. 2017. PolySurface: a design approach for rapid prototyping of shape-changing displays using semi-solid surfaces. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. 1283–1294.

[16] Christer Fåhraeus, Johan Lindgren, and Stefan Burström. 2007. Electronic pen. (July 3 2007). US Patent 7,239,306.

[17] George W Fitzmaurice, Hiroshi Ishii, and William AS Buxton. 1995. Bricks: laying the foundations for graspable user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 442–449.

[18] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: dynamic physical affordances and constraints through shape and object actuation.. In *Uist*, Vol. 13. 2501988–2502032.

[19] FormLabs. *Clear Resin*. `https://formlabs.com/store/clear-resin/` - accessed July-10-2020.

[20] Phil Frei, Victor Su, Bakhtiar Mikhak, and Hiroshi Ishii. 2000. Curlybot: designing a new class of computational toys. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 129–136.

[21] Seth Copen Goldstein, Jason D Campbell, and Todd C Mowry. 2005. Programmable matter. *Computer* 38, 6 (2005), 99–101.

[22] Antonio Gomes, Calvin Rubens, Sean Braley, and Roel Vertegaal. 2016. Bitdrones: Towards using 3d nanocopter displays as interactive self-levitating programmable matter. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 770–780.

[23] Matthew G Gorbet, Maggie Orth, and Hiroshi Ishii. 1998. Triangles: tangible interface for manipulation and exploration of digital information topography. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 49–56.

[24] Pauline Gourlet, Mathieu Le Goc, and Sean Follmer. 2017. Revisiting Turtles and Termites: an Open-ended Interactive Physical Game with Multiple Robots. In *Proceedings of the 2017 Conference on Interaction Design and Children*. 679–682.

[25] Changyo Han, Ryo Takahashi, Yuchi Yahagi, and Takeshi Naemura. 2020. PneuModule: Using Inflatable Pin Arrays for Reconfigurable Physical Controls on Pressure-Sensitive Touch Surfaces. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–4.

[26] John Hardy, Christian Weichel, Faisal Taher, John Vidler, and Jason Alexander. 2015. Shapeclip: towards rapid prototyping with shape-changing displays for designers. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 19–28.

[27] Scott E Hudson and Jennifer Mankoff. 2014. Concepts, values, and methods for technical human–computer interaction research. In *Ways of Knowing in HCI*. Springer, 69–93.

[28] Alexandra Ion, Johannes Frohnhofen, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. 2016. Metamaterial mechanisms. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 529–539.

[29] Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. 2012. Radical atoms: beyond tangible bits, toward transformable materials. *interactions* 19, 1 (2012), 38–51.

[30] Hiroshi Ishii and Brygg Ullmer. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. 234–241.

[31] Alec Jacobson, Daniele Panozzo, Oliver Glauser, Cédric Pradalier, Otmar Hilliges, and Olga Sorkine-Hornung. 2014. Tangible and modular input device for character articulation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–12.

[32] Yvonne Jansen, Pierre Dragicevic, Petra Isenberg, Jason Alexander, Abhijit Karnik, Johan Kildal, Sriram Subramanian, and Kasper Hornbæk. 2015. Opportunities and challenges for data physicalization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3227–3236.

[33] Hsin-Liu Kao, Deborah Ajilo, Oksana Anilionyte, Artem Dementyev, Inrak Choi, Sean Follmer, and Chris Schmandt. 2017. Exploring interactions and perceptions of kinetic wearables. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. 391–396.

[34] Shohei Katakura, Yuto Kuroki, and Keita Watanabe. 2019. A 3D printer head as a robotic manipulator. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 535–548.

[35] Shohei Katakura and Keita Watanabe. 2018. Printmotion: Actuating printed objects using actuators equipped in a 3D printer. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings*. 137–139.

[36] Kunihiro Kato and Homei Miyashita. 2014. Extension sticker: a method for transferring external touch input using a striped pattern sticker. In *Proceedings of the adjunct publication of the 27th annual ACM symposium on User interface software and technology*. ACM, 59–60.

[37] Lawrence H Kim, Daniel Drew, Veronika Domova, and Sean Follmer. 2020. User-defined Swarm Robot Control. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. xx–xx.

[38] Lawrence H Kim and Sean Follmer. 2017. Ubiswarm: Ubiquitous robotic interfaces and investigation of abstract motion as a display. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–20.

[39] Lawrence H Kim and Sean Follmer. 2019. SwarmHaptics: Haptic Display with Swarm Robots. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[40] Yoshifumi Kitamura, Yuichi Itoh, and Fumio Kishino. 2001. Real-time 3D interaction with ActiveCube. In *CHI'01 extended abstracts on Human factors in computing systems*. 355–356.

[41] Gierad Laput, Eric Brockmeyer, Scott E Hudson, and Chris Harrison. 2015. Acoustruments: Passive, acoustically-driven, interactive controls for handheld devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2161–2170.

[42] Mathieu Le Goc, Lawrence H Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. 2016. Zooids: Building blocks for swarm user interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 97–109.

[43] Daniel Leithinger, Sean Follmer, Alex Olwal, and Hiroshi Ishii. 2015. Shape displays: Spatial interaction with dynamic physical form. *IEEE computer graphics and applications* 35, 5 (2015), 5–11.

[44] Daniel Leithinger, Sean Follmer, Alex Olwal, Samuel Luescher, Akimitsu Hogge, Jinha Lee, and Hiroshi Ishii. 2013. Sublimate: state-changing virtual and physical rendering to augment interaction with shape displays. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1441–1450.

[45] Joanne Leong, Florian Perteneder, Hans-Christian Jetter, and Michael Haller. 2017. What a Life! Building a Framework for Constructive Assemblies. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*. 57–66.

[46] Rong-Hao Liang, Kai-Yin Cheng, Liwei Chan, Chuan-Xhyuan Peng, Mike Y Chen, Rung-Huei Liang, De-Nian Yang, and Bing-Yu Chen. 2013. GaussBits: magnetic tangible bits for portable and occlusion-free near-surface interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1391–1400.

[47] Thomas H Massie, J Kenneth Salisbury, and others. 1994. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, Vol. 55. Chicago, IL, 295–300.

[48] Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. 2014. WirePrint: 3D printed previews for fast prototyping. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 273–280.

[49] 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*. 87–96.

[50] Ken Nakagaki, Daniel Fitzgerald, Zhiyao Ma, Luke Vink, Daniel Levine, and Hiroshi Ishii. 2019. inFORCE: Bi-directionalForce'Shape Display for Haptic Interaction. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 615–623.

[51] Ken Nakagaki, Sean Follmer, Artem Dementyev, Joseph A Paradiso, and Hiroshi Ishii. 2017. Designing Line-Based Shape-Changing Interfaces. *IEEE Pervasive Computing* 16, 4 (2017), 36–46.

[52] Ken Nakagaki, Sean Follmer, and Hiroshi Ishii. 2015. Lineform: Actuated curve interfaces for display, interaction, and constraint. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 333–339.

[53] Ken Nakagaki, Yingda Liu, Chloe Nelson-Arzuaga, and Hiroshi Ishii. 2020. TRANS-DOCK: Expanding the Interactivity of Pin-based Shape Displays by Docking Mechanical Transducers. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 131–142.

[54] Ken Nakagaki, Udayan Umapathi, Daniel Leithinger, and Hiroshi Ishii. 2017. AnimaStage: Hands-on Animated Craft on Pin-based Shape Displays. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. 1093–1097.

[55] Ryosuke Nakayama, Ryo Suzuki, Satoshi Nakamaru, Ryuma Niiyama, Yoshihiro Kawahara, and Yasuaki Kakehi. 2019. MorphIO: Entirely Soft Sensing and Actuation Modules for Programming Shape Changes through Tangible Interaction. In *Proceedings of the 2019 on Designing Interactive Systems Conference*. 975–986.

[56] Nintendo. *Nintendo Labo*. `https://labo.nintendo.com/` - accessed July-10-2020.

[57] Ayberk Özgür, Séverin Lemaignan, Wafa Johal, Maria Beltran, Manon Briod, Léa Pereyre, Francesco Mondada, and Pierre Dillenbourg. 2017. Cellulo: Versatile handheld robots for education. In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*. IEEE, 119–127.

[58] Gian Pangaro, Dan Maynes-Aminzade, and Hiroshi Ishii. 2002. The actuated workbench: computer-controlled actuation in tabletop tangible interfaces. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*. 181–190.

[59] James Patten and Hiroshi Ishii. 2007. Mechanical constraints as computational constraints in tabletop tangible interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 809–818.

[60] Florian Perteneder, Kathrin Probst, Joanne Leong, Sebastian Gassler, Christian Rendl, Patrick Parzer, Katharina Fluch, Sophie Gahleitner, Sean Follmer, Hideki Koike, and others. 2020. Foxels: Build Your Own Smart Furniture. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 111–122.

[61] Ivan Poupyrev, Tatsushi Nashida, and Makoto Okabe. 2007. Actuation and tangible user interfaces: the Vaucanson duck, robots, and shape displays. In *Proceedings of the 1st international conference on Tangible and embedded interaction*. 205–212.

[62] 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*. 647–654.

[63] Majken K Rasmussen, Esben W Pedersen, Marianne G Petersen, and Kasper Hornbæk. 2012. Shape-changing interfaces: a review of the design space and open research questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 735–744.

[64] Jun Rekimoto. 2002. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 113–120.

[65] John W Romanishin, Kyle Gilpin, and Daniela Rus. 2013. M-blocks: Momentum-driven, magnetic modular robots. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 4288–4295.

[66] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. 2014. Programmable self-assembly in a thousand-robot swarm. *Science* 345, 6198 (2014), 795–799.

[67] Daniela Rus. 1998. Self-reconfiguring robots. *IEEE Intelligent Systems and Their Applications* 13, 4 (1998), 2–4.

[68] Philipp Schoessler, Daniel Windham, Daniel Leithinger, Sean Follmer, and Hiroshi Ishii. 2015. Kinetic blocks: Actuated constructive assembly for interaction and display. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 341–349.

[69] Joon-Gi Shin, Eiji Onchi, Maria Jose Reyes, Junbong Song, Uichin Lee, Seung-Hee Lee, and Daniel Saakes. 2019. Slow Robots for Unobtrusive Posture Correction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–10.

[70] Alexa F Siu, Eric J Gonzalez, Shenli Yuan, Jason B Ginsberg, and Sean Follmer. 2018. Shapeshift: 2D spatial manipulation and self-actuation of tabletop shape displays for tangible and haptic interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[71] Sony. *aibo*. `https://us.aibo.com/` - accessed July-10-2020.

[72] Alexander Sprowitz, Soha Pouya, Stéphane Bonardi, Jesse Van Den Kieboom, Rico Mockel, Aude Billard, Pierre Dillenbourg, and Auke Jan Ijspeert. 2010. Roombots: reconfigurable robots for adaptive furniture. *IEEE Computational Intelligence Magazine* 5, 3 (2010), 20–32.

[73] James Patten Studio. *Thumbles - Robotic tabletop user interface platform*. `https://www.youtube.com/watch?v=SsD3uTUtPks` - accessed July-10-2020.

[74] Hiroshi Sugihara. *Ready to Crawl*. `https://www.youtube.com/watch?v=G0-lJKOgTuo` - accessed July-10-2020.

[75] Ryo Suzuki, Hooman Hedayati, Clement Zheng, James Bohn, Daniel Szafir, Ellen Yi-Luen Do, Mark D Gross, and Daniel Leithinger. 2020. RoomShift: Room-scale Dynamic Haptics for VR with Furniture-moving Swarm Robots. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. xx–xx.

[76] Ryo Suzuki, Jun Kato, Mark D Gross, and Tom Yeh. 2018. Reactile: Programming Swarm User Interfaces through Direct Physical Manipulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[77] Ryo Suzuki, Abigale Stangl, Mark D Gross, and Tom Yeh. 2017. FluxMarker: Enhancing Tactile Graphics with Dynamic Tactile Markers. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. 190–199.

[78] Ryo Suzuki, Junichi Yamaoka, Daniel Leithinger, Tom Yeh, Mark D Gross, Yoshihiro Kawahara, and Yasuaki Kakehi. 2018. Dynablock: Dynamic 3D Printing for Instant and Reconstructable Shape Formation. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 99–111.

[79] Ryo Suzuki, Clement Zheng, Yasuaki Kakehi, Tom Yeh, Ellen Yi-Luen Do, Mark D Gross, and Daniel Leithinger. 2019. ShapeBots: Shape-changing Swarm Robots. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 493–505.

[80] Yasaman Tahouni, Isabel PS Qamar, and Stefanie Mueller. 2020. NURBSforms: A Modular Shape-Changing Interface for Prototyping Curved Surfaces. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 403–409.

[81] Tommaso Toffoli and Norman Margolus. 1991. Programmable matter: concepts and realization. *Physica. D, Nonlinear phenomena* 47, 1-2 (1991), 263–272.

[82] Brygg Ullmer and Hiroshi Ishii. 1997. The metaDESK: models and prototypes for tangible user interfaces. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*. 223–232.

[83] Chris Urmson and others. 2008. Self-driving cars and the urban challenge. *IEEE Intelligent Systems* 23, 2 (2008), 66–68.

[84] Nicolas Villar, Daniel Cletheroe, Greg Saul, Christian Holz, Tim Regan, Oscar Salandin, Misha Sra, Hui-Shyong Yeo, William Field, and Haiyan Zhang. 2018. Project zanzibar: A portable and flexible tangible interaction platform. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[85] Luke Vink, Viirj Kan, Ken Nakagaki, Daniel Leithinger, Sean Follmer, Philipp Schoessler, Amit Zoran, and Hiroshi Ishii. 2015. Transform as adaptive and dynamic furniture. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. 183–183.

[86] Eric Whitmire, Hrvoje Benko, Christian Holz, Eyal Ofek, and Mike Sinclair. 2018. Haptic revolver: Touch, shear, texture, and shape rendering on a reconfigurable virtual reality controller. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[87] Karl Willis, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. 2012. Printed optics: 3D printing of embedded optical elements for interactive devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 589–598.

[88] Chang Xiao, Karl Bayer, Changxi Zheng, and Shree K Nayar. 2019. Vidgets: modular mechanical widgets for mobile devices. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

[89] Junichi Yamaoka and Yasuaki Kakehi. 2017. ProtoMold: An interactive vacuum forming system for rapid prototyping. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2106–2115.

[90] Yuko Yanagawa, Kaori Ikematsu, Chihiro Suga, Mana Sasagawa, Yasushi Matoba, and Itiro Siio. 2017. Anamorphicons: An Extended Display Utilizing a Cylindrical Mirror Widget. In *Proceedings of the 29th Australian Conference on Computer-Human Interaction*. 457–461.

[91] Mark Yim, David G Duff, and Kimon D Roufas. 2000. PolyBot: a modular reconfigurable robot. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, Vol. 1. IEEE, 514–520.

[92] Neng-Hao Yu, Sung-Sheng Tsai, I-C Hsiao, Dian-Je Tsai, Meng-Han Lee, Mike Y Chen, and Yi-Ping Hung. 2011. Clip-on gadgets: expanding multi-touch interaction area with unpowered tactile controls. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 367–372.

[93] Yiwei Zhao, Lawrence H Kim, Ye Wang, Mathieu Le Goc, and Sean Follmer. 2017. Robotic assembly of haptic proxy objects for tangible interaction and virtual reality. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. 82–91.

[94] Amit Zoran, Roy Shilkrot, Pragun Goyal, Pattie Maes, and Joseph A Paradiso. 2014. The wise chisel: The rise of the smart handheld tool. *IEEE Pervasive Computing* 13, 3 (2014), 48–57.