The Router of All Evil

Designerly Hacking a Network of One's Own

David Chatting
david.chatting@newcastle.ac.uk
Open Lab, Newcastle University
Newcastle upon Tyne, Tyne and Wear, UK

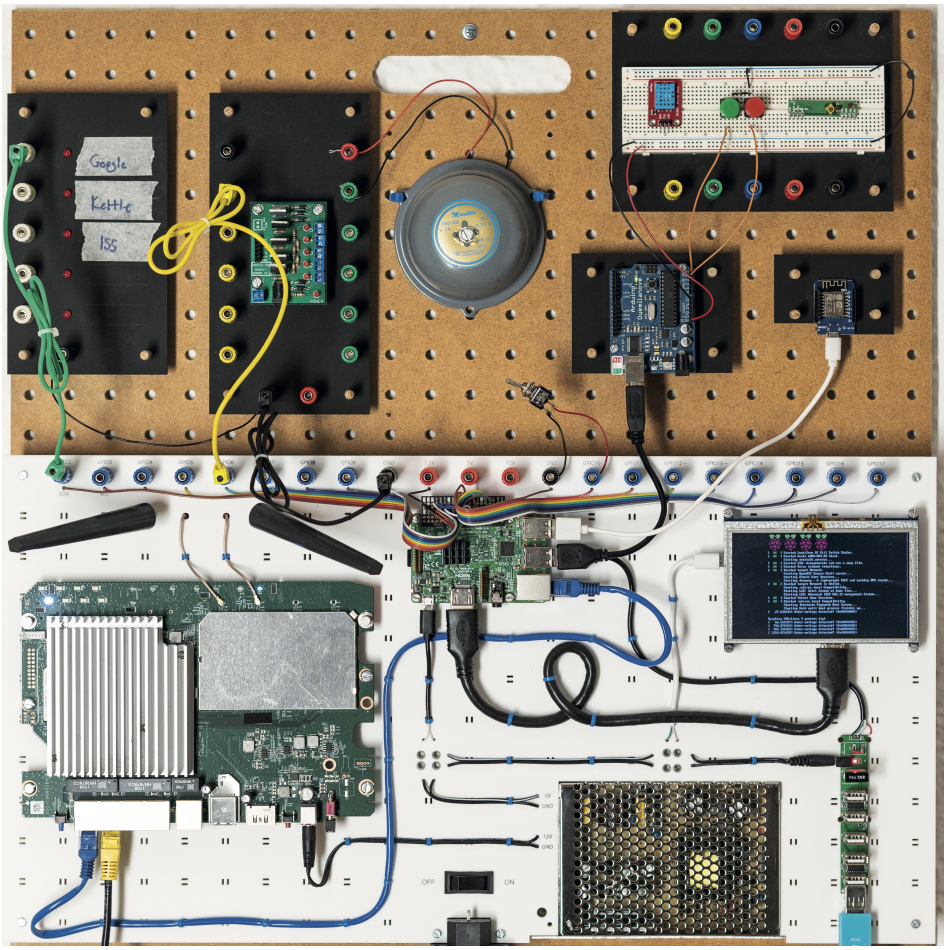

Figure 1: The Router of All Evil

ABSTRACT

This paper contributes a new design research artifact, *The Router of All Evil*. This is a Research Product that results from and scaffolds a Research Through Design exploration of the domestic Internet. The Router of All Evil is designed to reveal the technical possibility of the humble home router, to define *a network of one's own*, where homelife can unfold in creative, fulfilling and private ways, as an alternative to the prevalent corporate and surveillant logics of Silicon Valley's Internet of Things. Through a methodological contribution described as *Designerly Hacking*, this paper demonstrates how technical alternatives can first be revealed by hacking or breaking-up a system and then put back together for the use of a broader (designerly) public. To these ends, the Router of All Evil exemplifies *Pace Layer* design, where rapid design reconfigurations of hardware and software are purposefully afforded.

TEI '24, February 11–14, 2024, Cork, Ireland

ACM ISBN 979-8-4007-0402-4/24/02…$15.00
https://doi.org/10.1145/3623509.3633357

CCS CONCEPTS

• **Human-centered computing → HCI design and evaluation methods**.

KEYWORDS

Design Research, Research Through Design, autobiographical methods, hacking, networking

ACM Reference Format:
David Chatting. 2024. The Router of All Evil: Designerly Hacking a Network of One's Own. In *Eighteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '24), February 11–14, 2024, Cork, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3623509.3633357

1 INTRODUCTION

The *Router of All Evil* is a new design research artifact that emerges from a broader Research Through Design [17] exploration of the domestic Internet as a site of struggle for personal liberty – for *a network of one's own*; as an alternative to the prevalent corporate and surveillant logics of Silicon Valley's Internet of Things (IoT) [34]. The notion of a network of one's own is of course a play on Virginia Woolf's 1929 seminal feminist essay *A Room of One's Own* [33] and in making this parallel I am implying that there are creative and fulfilling ways for homelife to unfold when one owns the network, that goes beyond a simple notion of privacy [32]. As Woolf says, "*Even allowing a generous margin for symbolism, that five hundred a year stands for the power to contemplate, that a lock on the door means the power to think for oneself*" [33, p. 89][1].

This design research is in the tradition of the likes of Bill Gaver and Heather Martin's *Alternatives* workbook, which describes about twenty conceptual design proposals and in doing so critiques the design of domestic information appliances that, "*tend to represent a narrow range of cultural possibilities, reinforcing a simple dichotomy between work and play.*" [19, p. 209]. More recently, Audrey Desjardins and her co-authors report co-speculations with participants for alternative visions of the Internet of Things for non-stereotypical homes [10]. However, rather than developing alternative conceptual designs, my intention here is that alternatives be derived from demonstrable technical effects. This is achieved through a close technical practice described here as *Designerly Hacking*, which reveals some new technical possibility to design with, by hacking or breaking-up a system and then putting it back together in a new settled form for a broader (designerly, less technical) public. This is consistent both with Carl DiSalvo's *Adversarial Design* process of *revealing the hegemony* then *reconfiguring the remainder* [12], and with previous accounts of hacking in design research [6, 15, 20, 27]. The Router of All Evil is an example of such a settled designerly form, which puts together hacks that allow the behaviour of the home network (the devices, data and connections) to be managed at will; this reconfiguration of hardware and software is purposefully afforded through what is described as a *Pace Layer* design.

To overview the paper, I will first offer a definition of *Designerly Hacking*, before outlining my Research Through Design exploration of the domestic Internet, which was structured by a workshop series known as *Hack My House* and through which the *Router of All Evil* was produced. The process of designing the Router of All Evil is then considered in detail, which exemplifies the possibilities of Pace Layer design, with respect to the Research Product literature [2, 25]. To these ends, this paper contributes a method (Designerly Hacking), an artifact (the Router of All Evil), and a conceptual framework (Pace Layer design). Furthermore, this was explored in a necessarily autobiographical context. I close with a discussion reflecting on ways to approach Pace Layer design and on the reality of a network of one's own to be found in the humble home router. This goes considerably beyond a previously published overview of the router and Pace Layer design [5].

[1] £500 in 1929, is equivalent to around $50,000 in 2023.

2 DESIGNERLY HACKING

Designerly Hacking is a method that discloses new technical possibility in complex systems. It operates through close technical work that incorporates the products and methods of hacking but with designerly intent. In essence, it transforms hackerly forms into designerly forms to be manipulated in one's own design process or made public for the use of others. *Hackerly* here implies a set of technical competencies and aesthetic commitments, distinct from a (typical) designerly practice and distinct from a (typical) software engineering practice. Hacking (and so designerly hacking) has two phases of activity: breaking up and putting back together, having found some technical alternative. Crucially each phase consumes and produces different forms of knowledge and constructs different publics as they progress. This use of *designerly* is intended in the sense articulated by Nigel Cross [7]. There are some parallels with Agre's *Critical Technical Practice* [1] but with perhaps a less explicitly critical agenda and more as a means to seek technical alternatives. I have coined the term *Designerly Hacking* to describe this Research Through Design [17] exploration that enables a material engagement in the networked home by appropriating some of the practices and tools of hacking to disrupt the existing logics of the network.

3 PRIVATE WORK

In seeking to disclose design alternatives for a network of one's own through designerly hacking, a small network testbed would have been a convenient site, but it would have lacked the vital context and complexity of homelife that is the essence of the study. An autobiographical study of my own home network is then a natural choice; where access and permission are uniquely available for extended periods of time – here for over three years. Significantly, living alone I can reasonably give my express consent for the interception of my own network traffic, in ways that would not be possible in a participant's home or if I were sharing with others. However, there are some inherent methodological and social tensions in doing such autobiographical design research in one's own home (and working in intimate spaces in general), so this paper also seeks to contribute to this longstanding discourse [8, 9, 11, 15, 16, 23, 24].

More concretely, the intention of this exercise was to produce a series of self-directed private hacks, as software and hardware interventions in my home, that allow an exploration of prototyped forms, that then attempt to facilitate myself and others to make alternative design proposals for the network. However, this was

not altogether straightforward. Hacking is typically rather slow, concentrated, and antisocial; it is then a risky activity in which time invested can frequently not be rewarded and as a solitary endeavour, motivation can wax and wane, or one can become hopelessly lost in the detail. As Cally Gatehouse and I previously reported in first-person accounts of our private Research Through Design inquiries, hacking is a difficult activity to manage as part of a resource-limited directed inquiry [15]. In contrast, William Goddard and Robert Cercos describe their collaborative playful hacks as Research Through Design, where they were "*not driven by the expectation of research outcomes*" [20, p. 333]. It is evident that for Designerly Hacking to be a productive method it is likely to need some working framework.

4 COME HACK MY HOUSE?

In an attempt to mitigate some of the difficulties of private self-directed hacking I developed a series of public workshops known as Hack My House – the Router of All Evil was one of its many products. Hack My House was conceived as a lightweight workshop format, loosely structured as a series of irregular hackdays with a public consisting of a small group of trusted friends, who were invited to my house to come and make something playful with the products of my private hacking. I would not have been willing to open my home and my network to strangers. This gave my private hacks a series of deadlines, an audience and so a purpose. This section briefly outlines this framework for Designerly Hacking.

4.1 Participants

Over sixteen months and five events, nine friends (Dan, Andy, Cally, Kyle, Tom, Diego, Tim St, Tim Sw and Mike) came to hack my house, each with broadly designerly practices, working with software, electronics and networks, and with a mixture of academic and commercial experiences. Most returned on repeated occasions. Every iteration of the workshop had a slightly different flavour. On each occasion between four and six other people participated – some ran for the full day, some just for the afternoon, one for the whole weekend. I recognised early on that a critical role for me was as the host, responsible for curating materials to simulate the engagement and curiosities of those participating, as well as providing meals, snacks, cups of tea and coffee. Each day closed with a wide-ranging audio-recorded group discussion before we shared a take-away meal and some beers. By all reports, it was an enjoyable experience.

4.2 Ethics and Legality

From an ethical perspective, the Hack My House format presented few concerns. However, it was unusual in a number of respects that warrant some brief reflection. Firstly, being in a private space (a rented flat) I was mindful of the participants' safety, ensuring that safety certificates, insurance policies, and evacuation procedures were up to date. Secondly, being my own home, I considered my own safety and privacy; all the participants were already well-known to me professionally and socially, so a trusting relationship could be assumed. Thirdly, elements of hacking are illegal when practiced against a third party, but here the focus is my private home network for which I can grant permission. Under UK law and specifically the Investigatory Powers Act (IPA) of 2016[2] it is an offence to, "*intentionally intercept a communication [...] in the course of its transmission by means of a public or private telecommunication system*" (3.1). However, it is not an offence under 3.1 to "*intercept a communication in the course of its transmission by means of a private telecommunication system if the person, is a person with a right to control the operation or use of the system, or has the express or implied consent of such a person to carry out the interception.*" (3.2). That is to say, intercepting one's own local network traffic is legal in the UK. However, this required that systems external to my home network (like web servers) were not in the scope of our hacking and could not be improperly used. An application for ethical approval for my thesis programme of work, including these workshops and these legal considerations, was granted by my university before the event.

4.3 Not a Hackathon

While these hackdays might seem to share some features with Silicon Valley inspired hackathons or participatory design workshops [31], they were explicitly designed not to operate as such. The focus was not on the generation of design innovations or a co-design activity, but more obliquely to see how a designerly public responded to and motivated my private hacking and its products. The participant's designs were not intended as nuanced critical responses to the networked home per se, but instead to playfully surface ways of engaging with these materials through the experience and expertise of others – allowing me to draw on a wealth of technical and creative insights. This shares some intention with Tim Shaw and John Bowers' concept of public making [30] – indeed Tim participated in Hack My House #5. The longitudinal nature of Hack My House workshops allowed conversations to develop between us and engagements to change with the materials of the home and our interventions.

4.4 Preparation

My preparation for each workshop guided a new programme of designerly hacking, with an implied timetable and with some public accountability. This gave my work focus, as I found ways to make my network available to facilitate others to make rapid reconfigurations; in doing so I exposed, documented and curated the new technical potential that I was revealing through my ongoing private hacking. For the first workshop, I began to assemble a handbook for my house documenting all the networked stuff I had identified and the ways they might be hacked. My intention was to produce something close in spirit to a DIY manual for my home. For each device, I created a single printed page detailing serial numbers, firmware versions, network addresses, account details, APIs (Application Programming Interface), software tools and scripts, etc. While many of the devices had a network addressable interface, most were proprietary and while accessible not publicly documented by the manufacturer – so-called private APIs. However, a large online community reverse-engineer these devices, publishing the interfaces they find and the software control scripts they write under non-profit licenses on the Internet. In preparation for the first workshop, I installed and documented as many of these

[2] http://www.legislation.gov.uk/ukpga/2016/25/contents

scripts as I could find (and make work) on the network. To consolidate these found scripts, I hosted a web server on a Raspberry Pi and wrote endpoints to call these functions. An endpoint is a URL that specifies an action the server will take when presented with data of a specified type and format, containing the parameters required for its operation. In this way, control of disparate technologies like the Google Chromecast and IKEA Trådfri lights were addressable from a single interface on the Raspberry Pi. This was an early attempt to make a web *Home API*, a unified collection of software interfaces for the network. Knowing that many of the workshop's participants had professional experience with web technologies and tools, in particular with *JavaScript* and *Node.js*, made the choice of these web technologies straightforward — allowing easy integration of familiar software, workflows and practices. These preparations straightforwardly demonstrate a process of breaking-up and putting back together.

5 THE ROUTER OF ALL EVIL

The Router of All Evil (RoAE) was one of many products of my Hack My House activity, developed for and in response to the workshops. The RoAE is a WiFi router that serves my home network where the behaviour of connected devices can be reconfigured by making purposeful changes to the hardware and software of the router – in an attempt to assert a *network of one's own*. This section describes my Research Through Design chronologically, how the four major hardware iterations of the Router of All Evil were designed in dialogue with the five Hack My House workshops, concluding with the final iteration (see Figure 1). The artifact of the RoAE is also intended to contribute to design research discourses on alternative forms for the home WiFi router and home network [14, 26, 28].

5.1 Hack My House #1

The first Hack My House was a one-day workshop with four participants (Andy, Cally, Tom and Kyle). I had created a new WiFi network using a mini wireless router (GL-MT300N) attached by Ethernet to my ADSL modem, wanting to make an experimental space that was separate from my day-to-day network. The router was installed with the OpenWrt operating system, a popular Linux distribution allowing customisation of the routing software that might enable network messages to be intercepted and modified. A similar approach was taken by Kashmir Hill and Surya Mattu in their Gizmodo article *The House That Spied on Me* [22] to document the behaviour of Mattu's home IoT devices and the privacy-friendly Candle Smart Home project [29]. However, we soon demonstrated that the mini wireless router was incapable of serving my home network to resource-hungry hackers. Furthermore, while the possibilities of OpenWrt configuration had been available, nobody explored ways to modify the router, preferring instead more self-contained hacks using scripts running on their own laptops.

While the totality of workshop outcomes was rich and informative, the focus of this section is very specifically to recount those that guided the development of the RoAE. However to give a flavour of these outcomes, in the first workshop we collaborated to create a Rube Goldberg like IoT machine where a wireless doorbell triggered a picture to be taken (notionally at the front door) and then displayed on the television screen, whilst a synthesised text-to-speech voice instructed Alexa to make the sound of a barking dog. This proved a productive (and fun) collaborative way to proceed, where each of us took an element of the machine and created a trigger for the next stage.

5.2 Hack My House #2

The second Hack My House was an afternoon workshop with five participants (Dan, Andy, Cally, Kyle and Diego). The experience of the first workshop had made some clear technical demands and to address the throughput of the router I purchased a Linksys 1200 AC router with a considerably higher technical specification than the GL-MT300N. The Linksys router was also able to run OpenWrt, but the software installation required that the circuit board be removed from its case to reveal a serial connector required to transfer the new firmware; this was a literal opening of the black box to reveal a new possibility. In doing so, this process suggested an alternative way that the router could exist, open not only to software configuration but also open to hardware modification, being without its enclosure.

The enhanced specification of the new router and its Linux-based operating system also suggested that some of the functions previously served by the Raspberry Pi might now be run on the new router. However, the Raspberry Pi offered many new possibilities with its abundance of hardware (HDMI screen, audio, Bluetooth BLE, WiFi and GPIO) and readily available software packages. So instead, the router and the Raspberry Pi became tightly coupled but separate and this combination would later become known as the *Router of All Evil* (RoAE).

Before the second Hack My House there was a flurry of activity as I rapidly built three iterations of what would become the RoAE. In the first, the router and the Raspberry Pi were simply stacked with a 5" display into a compact semi-open unit, see Figure 2. Inside the Raspberry Pi GPIO was broken out on a bespoke PCB, there was a breadboard for electronic prototyping and a common power supply (on a second bespoke PCB) for the router and the Raspberry Pi. The router remained outside of its case and the circuit boards were attached to laser-cut acrylic plates. The top layer (the display) could be changed rapidly and superficially by software, the middle layer accepted hardware changes, and the bottom layer was the bedrock of the router.

The second iteration of the router disassembled the stack of layers and laid out these same components on a single laser-cut plywood plate, see Figure 3. The cables between the components were neatly stowed behind. While this exposed all the elements, the articulation between them was out of sight and the layout felt restrictive, rather than expansive and open to change. This led quickly to the third design.

The motivation of the third iteration of the router was then to give each circuit board adequate space and allow the cabling between them to be visible and so improve legibility, see Figure 4. I designed a laser-cut white acrylic cable board able to accept cable ties on a 1" pitch grid, which itself was mounted to an OSB (Oriented Strand Board) square plate (60 by 60 cm) – which was now large enough to warrant a handle. This version integrated an Arduino on a self-etched PCB where each of its input/output pins was broken out to a binding post. Sensors and actuators could then be assembled on the breadboard or other pegboard panels.

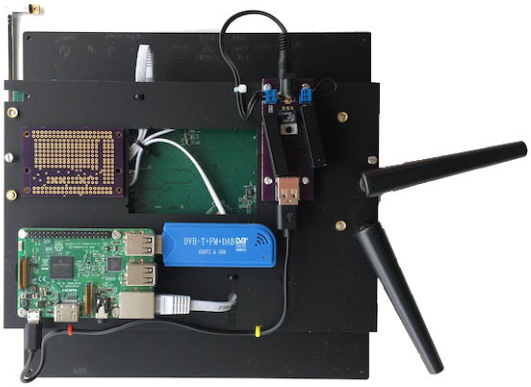

Figure 2: The first design iteration

Figure 3: The second design iteration

The PCB also allowed Raspberry Pi hats (expansion boards) to be plugged in. The Arduino was chosen for its GPIO, having more interfacing options than the Raspberry Pi; specifically it can read analogue sensors and operates at a 5-volt logic, rather than the less common 3.3-volt. The power supply PCB was similarly reproduced on a self-etched PCB. This was the iteration of the router used at Hack My House #2.

After the first Hack My House, I was looking to enable an easy way to make Rube Goldberg like machines of chains of loosely connected components – our doorbell machine had previously depended on relatively inflexible articulations. MQTT (Message Queuing Telemetry Transport) is a lightweight network protocol designed for IoT devices to publish small amounts of arbitrary data on a specified topic to a broker (a network server), other clients may then subscribe to this topic via the broker and will be pushed relevant messages instantaneously as they occur. This is then a simple way to connect one (or more) publishers of data to one (or more) subscribers, which may then in turn publish their own messages.

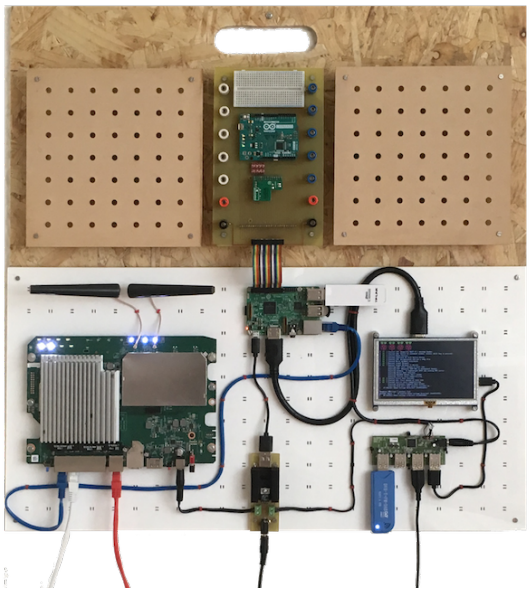

Figure 4: The third design iteration

Once I had installed an MQTT broker on the Router of All Evil's Raspberry Pi, this became the obvious choice for messaging on my network. MQTT is widely supported by good software libraries, significantly for JavaScript and Arduino, that create publishers and subscribers of messages in just a few lines of code. Using an inexpensive ESP8266 WiFi (Arduino compatible) module and humidity sensor, I quickly created a topic for environmental data. By attaching an old Wattson Energy Meter to a Raspberry Pi, I could then read and publish the real-time electricity usage of my home, making the activity of my high-current appliances legible, like the kettle and cooker. Finally, I made a topic for DNS (Domain Name System) requests made by devices to resolve Internet domain names to an IP address. A DNS request will always precede any device's interaction with a remote server, so this became a powerful mechanism to watch the network activity of every device in the home, in real-time.

By the time of Hack My House House #2, I had some new offers for the group including the third iteration of the router and the MQTT infrastructure. Kyle quickly dubbed the router as the *Router of All Evil*! A play on the biblical proverb that, "*the love of money is the root of all evil*". This stuck and seemed to communicate a little of the router's counter-cultural spirit.

The workshop itself was relatively short, just an afternoon, but in just a few hours, we created two interesting demonstrations. Firstly, an electrical consumption game that attempted to identify appliances (e.g. the cooker and the toaster) as they were turned on. This was achieved by watching the change in the power consumption that was reported as an MQTT topic. The second demonstration was a service that would read aloud, on the Chromecast speaker, any text that had been sent to an MQTT topic – this used the Google cloud-based text-to-speech service. The result had an appealing and playful flexibility, but perhaps more interestingly the (proper) use

of external cloud services sparked a debate about what services the home needs and needs not to provide for itself – given the overall objective was to assert a network of one's own.

5.3 Hack My House #3

Two months after the second workshop, I convened Hack My House #3, a two-day weekend workshop with four participants (Dan, Andy, Cally and Diego). In preparation, my efforts concentrated on ways to expose new software interfaces to the Router of All Evil, developing the Home API rather than making further hardware revisions. The API now allowed any device on the network to query any other device by various attributes, including by MAC address, vendor, and its open ports (indicative of the network services it exposes, like *ssh*, etc). Again, this required some close technical work using Unix command-line tools like *nmap* to transform these hackerly forms into something we could work with. The intention over the weekend was to use some of this potential to create some semi-permanent interventions in the space.

Early in the weekend Diego introduced us to and installed the *Node-RED* software on the Router of All Evil. Node-RED is a visual flow-based programming platform that runs in the browser and allows multiple sources of data to trigger events, including MQTT messages. It is based on Node.js and many packages exist for common IoT devices, for example, to control the Google Chromecast or to use the GPIO on a Raspberry Pi or an attached Arduino. We could now rapidly develop novel hardware inputs or outputs and integrate them with the router software. Node-RED seemed like the missing piece of the puzzle, allowing us to rapidly experiment with rules and integrate new hardware rapidly prototyped on the router.

The rest of our weekend was spent exploring the potential of Node-RED. Andy and Diego collaborated to make an MQTT button with an ESP8266 WiFi module and a breadboard to remotely control an LED attached to the GPIO on the router. We started to rewrite the Home API so that it also ran in Node-RED and wrote flows to send audio to the three Chromecast speakers around the house. Text-to-speech messages could now be directed to a specific room. Yet again, by the end of the weekend we could see yet more potential than we had begun with and wanted to do it again; the longer format had been altogether more relaxed and social. In contrast to previous workshops there was more focus on creating potentials than demonstrations, despite my intervention-focused design brief!

While there was a software focus to this workshop, it was apparent that there was an unresolvable technical problem with the Arduino prototyping board, the router was drawing too much current from the power supply, which was running very hot. Furthermore, despite its large size the physical prototyping area again felt restrictive – the pegboard panels being quite small. These considerations motivated the fourth and final hardware design.

5.4 Hack My House #4

Three months after the third workshop we convened for Hack My House #4 – a one-day workshop with five participants (Dan, Andy, Cally, Kyle and Tom). My framing was a little more explicit on this occasion, encouraging longer-term interventions, to "*build something for me to live with for the following week – something to surprise me, that will perhaps haunt the space!*". In preparation, I built what was to be the final iteration of the RoAE (Figures 1 and 5), which attempted to resolve the practical challenges of the previous versions and to embody more directly my notions of designing with layers of mutability, that I would later describe as Pace Layer design. Working from the bottom, the router transforms the infrastructurally delivered mains electricity via a new overly provisioned power unit into the 5 and 12-volt supplies that power both the router and the Raspberry Pi, with enough unprovisioned current for unspecified future electronic circuitry. The Raspberry Pi GPIO is now delivered directly to the edge of the cable board, via a row of labelled binding posts – with the power occupying the central posts. Binding posts were chosen as they allow both a connection by a plug and also by clamping a bared wire. The prototyping PCB was abandoned in favour of a more flexible arrangement of panels attached to the pegboard. The Arduino, a larger breadboard and a new ESP8266 WiFi module were now mounted on acrylic panels, held in place with wooden pegs. These first panels were primarily about making these components available and offering them some stability, to allow electronic prototypes to be rapidly explored. The LED and MOSFET panels (shown in Figure 5) would later be built for Hack My House #5.

Cally responded to the mystical dimension of the brief by building a crystal radio kit to interact with the home's Hertzian space. Tom created a visualisation of a WiFi beacon-based location system I had built – proving rather definitely that this approach did not work very well!

On the day the brief proved a little too ambitious – Kyle started to prototype a ghostly experience on my Kindle that would trigger events in the home but despite our best intentions the workshop didn't quite gel and nothing got haunted that hackday. Nonetheless, there was still a collective enthusiasm to do it at least once more.

5.5 Hack My House #5

Four months after the fourth workshop it was the final Hack My House, a one-day workshop with six participants (Andy, Cally, Diego, Tim Sw, Tim St and Mike). I wondered if we might work together to build something bigger and likely electronic. Following the relative failure of the previous workshop, it was clear some consolidation was required. To these ends, my preparation for Hack My House #5 was primarily to create and document a comprehensive Home API and secondly to extend the possibilities offered by the pegboard panels.

The task of designing a Home API had begun in preparation for the first workshop, to bring the first found hacks together in a common location. It evolved over the period of the workshops to accommodate the new technical possibilities being disclosed through hacking. In these final phases of work I was attempting to impose an order and design on these disparate technologies. This process suggested logical, but absent, new functions to develop and include. Likewise, through this process, common syntaxes and linguistic expectations emerged, as more functions were incorporated. The resulting Home API interface is itself intended as a designerly outcome that attempts to create a temporary settlement of the found and developed hacks, offering new desirable technical affordances, whilst obfuscating less desirable ones. At Hack My House #1 Andy

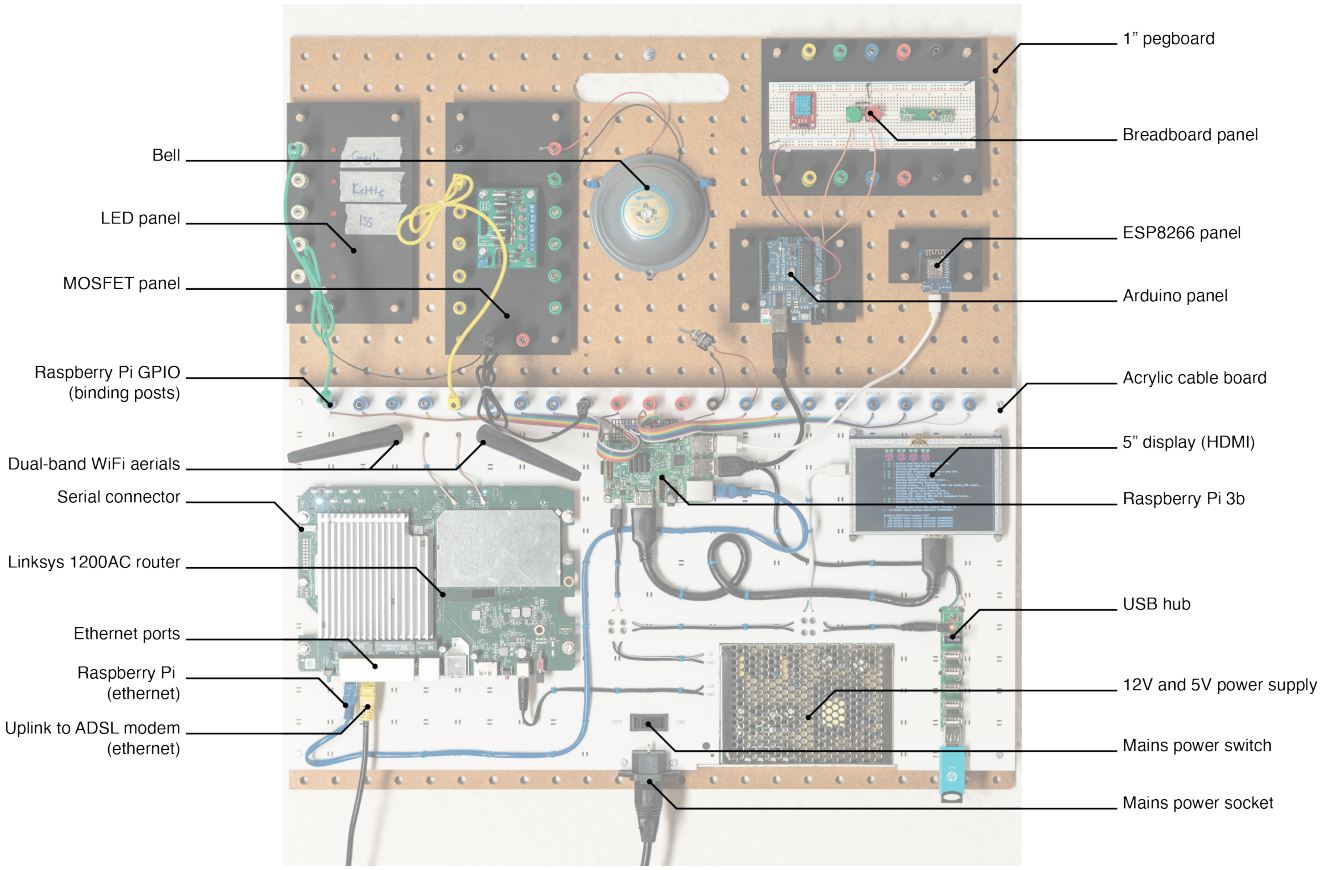

Figure 5: The Router of All Evil (annotated)

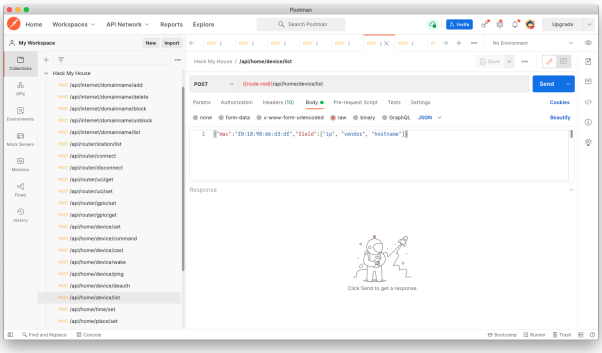

Figure 6: The Home API as a Postman collection

showed us the Postman tool for documenting and testing APIs; I added each new Home API endpoint to the Postman collection with a simple demonstration of its use, see Figure 6. For each workshop I then updated both the House Handbook with details of Home API and shared the latest Postman collection. This process of public documentation necessitated a period of reflection and invariably led to further edits of the API. The final Home API was implemented as a set of Node-RED flows and as such afforded inspection and editing.

Two new pegboard panels were introduced for Hack My House #5, an LED panel for indication and a MOSFET panel for operating high-power outputs. These were designed to directly interface with the Raspberry Pi's exposed GPIO binding posts – such that a patch wire could easily make a connection. Internally the binding posts on the panel were wired through to an LED via a series resistor such that they could be driven directly by the logic levels from the Raspberry Pi. In combination with Node-RED these panels allowed complex events to trigger simple outputs – for instance, an LED that lights every time the network makes a request of Google. The LED panel allowed an ad hoc annotation of these relationships using a simple handwritten label – reminiscent of the practice of writing a Scribble Strip on a mixing desk, where a piece of tape is stuck by a control with a note to temporarily associate it with a particular track or effect. The MOSFET panel could control five high power outputs from the Raspberry Pi's GPIO, each switching up to 60 volts. This panel mounted a commercially available MOSFET board whose inputs and outputs were again wired to binding posts for easy connection. This was a productive exercise with other panels being easily imagined; panels of switches, buttons and a mechanical stepper motor were planned but not realised.

At the workshop Mike and Tim St set about whimsically disconnecting my home network when my apartment was notionally shadowed by the passing ISS (International Space Station), an image

of which was to be displayed on my television screen to further my sense of awe. They developed a Node-RED flow to determine when this would be, showing this state using an LED on the panel. They considered this would happen infrequently and unpredictably enough (typically once or twice a week) to cause me a moment of reflection on each occasion. Their first conception of this powered off the router with a WiFi mains switch, but without the network it would then be impossible to automatically power it back on! Instead, they were able to use the Home API to list the devices on the network and disconnect each one in turn. Tim Sw brought along an electro-mechanical bell which was now straightforwardly wired to the router and script Node-RED flow written to ring on each request to a Google server made on the network, see Figure 7.

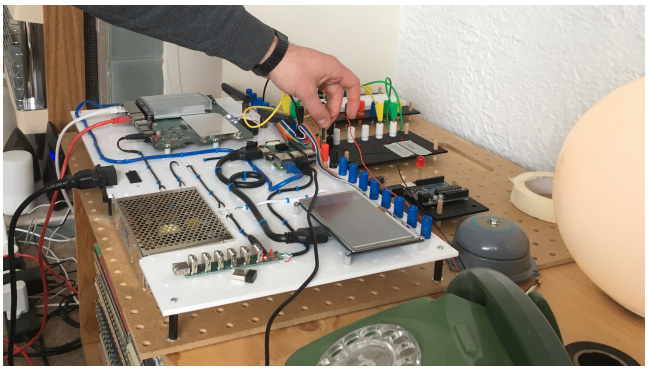

Figure 7: Hack My House #5

6 DISCUSSION

In drawing this multi-threaded work together, this section offers two points for discussion: a ways to understand the Router of All Evil as an example of what I term *Pace Layer Design* and some reflections on the reality of a network of one's own to be found in the humble home WiFi router.

6.1 Pace Layer Design

As described, the Router of All Evil is a commercial router that has been hacked open, then relaid out and then made stable. The final design principally has two pliable layers: the pegboard at the top presents the possibility of changing the hardware of the router, where components and panels can be easily reconfigured physically, while the white acrylic area at the bottom is where software can be changed via digital interfaces. The Raspberry Pi intermediates between the hardware layer and the router via its GPIO pins. These are made easily accessible by way of a row of labeled binding posts that allow a semi-permanent electrical connection to be made, without soldering. The router's firmware was replaced with OpenWrt, a popular Linux distribution that allows powerful configurations of the routing software. The Raspberry Pi was installed with Node-RED, a graphical programming environment that allows a variety of complex behaviors to be expressed and provides network interfaces like HTTP and MQTT. In this way, hardware and software layers are designed to accommodate reconfiguration.

The separation of the two areas demonstrates two ways of prototyping, one at the top in hardware and by software configuration below. While the hardware of the router has been opened up and unboxed, it has been laid out and made stable, and a new logic is suggested for it. The pegboard area affords a range of responses – some by the ad hoc attachment of components to the board, others through the development of new panels that formalise the hacks. Similarly, the router's software layer offered new ways of enacting change, notably through the development of the Home API, using the Node-RED environment.

The router's layers of hardware and software can be seen in terms of Stewart Brand's *Shearing Layers* [3] or more precisely in terms of his later *Pace Layer* model [4]. The Shearing Layers model offers a way to see the adaption of buildings through time, to see a complex system in struggle. Each layer (Site, Structure, Skin, Services, Space plan and Stuff) gains pace of change, from tens or hundreds of years to months or days. With each layer change is increasingly mutable with less work. Each shapes the possibilities of the adjacent layers as they shear against each other, as they resist or demand change from each other; e.g. the bright window that restricts the placement of the television screen. Brand frames this adaptation, through maintenance, as the system learning. A building that fails to learn, to adapt to use, can become precarious and ultimately fall. Brand has since proposed the Pace Layers model [4] to describe a durable social society with six layers from slow to fast: Nature, Culture, Governance, Infrastructure, Commerce and Fashion/Art. Each coexisting layer accommodates different rates of change but without the implied containment relationship between layers of the Shearing Layers. Pace Layers offer a way to understand complex dynamic systems, not least I argue, hardware and software prototypes.

Others have also applied the Shearing Layers to software/hardware systems. In their article *Big Ball of Mud*, Brian Foote and Joseph Yoder [13] consider the architecture of haphazardly developed software that rapidly evolves from quick-and-dirty code to deployed systems. They apply Brand's model to understand how code is maintained and adapted by multiple authors over time. This analysis highlights how some modules of code become established, whilst others are subject to continual modification – the software layer is itself structured. In Dan Hill's 2003 blog post *iPod and adaptive design* [21], he describes how after a firmware update, he had "*a whole new iPod*"; yet the integrated battery (that typically failed after 18 months of use) was not user replaceable and the physical aspects (control wheel, buttons and screen) were unchanging. Hill briefly explains how the Shearing Layers might inform a better adaptive design for connected electronic products. These ideas seem to naturally apply to prototyping and suggest a more deliberate approach to design where a complex system needs to learn.

Pace Layer prototyping, by my definition, is then a prototype where its form and function are not static or immutable, but instead respond to the environment in which it finds itself and the designer's emerging intentions; these are desirable qualities in a Research Through Design inquiry [18] and a Research Product [25]. Where these prototypes are changed by their encounters with the world, in ways legible to a designer, such that there is the opportunity for learning about that world – both by the designer and the prototype. In the context of a Research Through Design inquiry,

this can make legible some of the complexity of the system under study, whether technical or social.

While any electronic device (whether commercial devices, prototypes or Research Products) incorporating software and hardware might be seen through a Pace Layer lens, Pace Layer design implies that there are deliberate ways to design for change by virtue of the material (and immaterial) affordances that are chosen. For electronic stuff, these affordances are created by material enclosures and surfaces, the electronic hardware, the embedded software, data, and any subsequent interactional behaviour. Pace Layer design is then the intentional design of (especially electronic) stuff, that affords change or being changed through use.

The final design of the Router of All Evil is intended to be the clearest illustration of Pace Layer design, where the layers are most evidently to be seen. The commercial router has been unsettled in the hack by removing it from its case, exposing the circuit boards and allowing the original firmware update, but given a new stability through the backplate and new power supply. The addition of the Raspberry Pi and electronics prototyping area (including the pegboard panels) straightforwardly accepts hardware changes that can be easily recontextualised with a simple handwritten label. Immaterially the router is reconfigured in software, a process made easier through the design of the Home API and the use of Node-RED environment. Finally, the screen creates a surface on the router that can be changed up to 60 times a second.

6.2 A Network of One's Own?

My programme of design research, of which this paper describes a part, is an inquiry into the possibilities that are afforded when one has a private home network; it seeks to find practical ways to design alternatives that struggle with the corporate logics of Silicon Valley. A network of one's own technically implies that one has the visibility and control over the network, to determine which devices connect, what data is consumed and produced, and what connections are made to the Internet. To these ends, the designerly hacking that produced the Router of All Evil has demonstrated that there are alternative technical reconfigurations of the humble home WiFi router that achieve at least some of this autonomy.

Indeed, it becomes clear that without reconfiguration, the home WiFi router already defines a (relatively) private network (through walls you don't need to own), allowing the Internet to be brought into the home with a small (relatively inexpensive) router that makes a single connection to the wall. Each networked device in the home is typically connected through the router to request and receive data from the Internet. This affords some privacy and security, where firewalls can enforce access rules; partially hiding the home network from the outside world. However, this also exposes the home router as a point of surveillance from which to learn all about the home network and its use of the Internet. It is then a matter of asserting one's privacy through alternative configurations of this same technology, exemplified by the Router of all Evil.

The home WiFi router is perhaps the most prevalent pattern for the home network, at least in the UK. However, alternative technologies are to be found and are gaining popularity. There is a growing use of metropolitan-scale data networks (such as LoRaWAN and 5G), where there is no locally managed point of connection and all devices connect directly to the wide area network. If such metropolitan data networks continue to be adopted, without the intermediating home router, the private home network as we have come to know it, will cease to be.

7 CONCLUSION

This paper has contributed a new design research artifact, *The Router of All Evil* and described the process through which it was designed and built; a Research Through Design exploration of *a network of one's own* and a method described as *Designerly Hacking*. This has revealed some of the technical possibility of the humble home router to challenge the prevalent corporate and surveillant logics of Silicon Valley's Internet of Things. More broadly, this paper demonstrates how technical alternatives can first be revealed by hacking or breaking-up a system and then put back together for the use of a broader (designerly) public, in the context of the autobiographical Hack My House workshops. It has been argued that the Router of All Evil exemplifies *Pace Layer* design, where rapid design reconfigurations of hardware and software are purposefully afforded through material (and immaterial) design choices.

ACKNOWLEDGMENTS

I would like to express my gratitude to my friends and participants in the Hack My House workshops: Dan Foster-Smith, Andy Garbett, Cally Gatehouse, Kyle Montague, Tom Schofield, Diego Trujillo Pisanty, Tim Sargent, Tim Shaw and Mike Vanis. With thanks to Alexander Wilson for his photography. I am grateful for the guidance of Bill Gaver and Andy Boucher in developing my thesis work at Goldsmiths, from which this paper derives – this was supported by the Arts and Humanities Research Council (AHRC) Design Star Centre for Doctoral Training. My subsequent work at Newcastle University was supported by the UKRI-funded Centre for Digital Citizens (project number EP/T022582/1).

REFERENCES

[1] Philip E. Agre. 1997. *Computation and Human Experience.* Cambridge University Press. https://doi.org/10.1017/CBO9780511571169

[2] Andy Boucher. 2023. Research Products at Scale: Learnings from Designing Devices in Multiples of Ones, Tens, Hundreds and Thousands. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 177, 15 pages. https://doi.org/10.1145/3544548.3581540

[3] Stewart Brand. 1995. How Buildings Learn: what happens after they're built. *Penguin Books* 1 (1995), 720. https://doi.org/10.2307/990971 ISBN: 9781101562642.

[4] Stewart Brand. 2018. Pace Layering: How Complex Systems Learn and Keep Learning. *Journal of Design and Science* (2018). https://doi.org/10.21428/7f2e5f08

[5] David Chatting. 2023. Pace Layer Prototyping: How Prototypes Learn. *Interactions* 30, 2 (feb 2023), 14–15. https://doi.org/10.1145/3583127

[6] David Chatting, David S. Kirk, Abigail C. Durrant, Chris Elsden, Paulina Yurman, and Jo-Anne Bichard. 2017. Making Ritual Machines: The Mobile Phone as a Networked Material for Research Products. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 435–447. https://doi.org/10.1145/3025453.3025630

[7] Nigel Cross. 1982. Designerly ways of knowing. *Design Studies* 3, 4 (1982), 221–227. https://doi.org/10.1016/0142-694X(82)90040-0

[8] Sally Jo Cunningham and Matt Jones. 2005. Autoethnography: A Tool for Practice and Education. In *Proceedings of the 6th ACM SIGCHI New Zealand Chapter's International Conference on Computer-Human Interaction: Making CHI Natural* (Auckland, New Zealand) *(CHINZ '05)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/1073943.1073944

[9] Audrey Desjardins and Aubree Ball. 2018. Revealing Tensions in Autobiographical Design in HCI. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. Association for Computing Machinery, New York, NY, USA, 753–764. https://doi.org/10.1145/3196709.3196781

[10] Audrey Desjardins, Jeremy E. Viny, Cayla Key, and Nouela Johnston. 2019. Alternative Avenues for IoT: Designing with Non-Stereotypical Homes. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300581

[11] Audrey Desjardins and Ron Wakkary. 2016. Living In A Prototype: A Reconfigured Space. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5274–5285. https://doi.org/10.1145/2858036.2858261

[12] Carl DiSalvo. 2012. *Adversarial Design*. The MIT Press.

[13] Brian Foote and Joseph Yoder. 1997. Big ball of mud. *Pattern languages of program design* 217 (1997), 244–4695. http://www.laputan.org/mud/mud.html

[14] Joakim Formo. 2012. Ericsson's UX Lab & BERG explores the Internetworks of Things. https://web.archive.org/web/20130605052549/http://www.ericsson.com/uxblog/2012/05/ux-lab-x-berg-explores-iot/

[15] Cally Gatehouse and David Chatting. 2020. Inarticulate Devices: Critical Encounters with Network Technologies in Research Through Design. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (Eindhoven, Netherlands) *(DIS '20)*. Association for Computing Machinery, New York, NY, USA, 2119–2131. https://doi.org/10.1145/3357236.3395426

[16] William Gaver. 2006. The Video Window: My Life with a Ludic System. *Personal Ubiquitous Comput.* 10, 2–3 (jan 2006), 60–65. https://doi.org/10.1007/s00779-005-0002-2

[17] William Gaver. 2012. What Should We Expect from Research through Design?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 937–946. https://doi.org/10.1145/2207676.2208538

[18] William Gaver, Peter Gall Krogh, Andy Boucher, and David Chatting. 2022. Emergence as a Feature of Practice-Based Design Research. In *Proceedings of the 2022 ACM Designing Interactive Systems Conference* (Virtual Event, Australia) *(DIS '22)*. Association for Computing Machinery, New York, NY, USA, 517–526. https://doi.org/10.1145/3532106.3533524

[19] William Gaver and Heather Martin. 2000. Alternatives: Exploring Information Appliances through Conceptual Design Proposals. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. Association for Computing Machinery, New York, NY, USA, 209–216. https://doi.org/10.1145/332040.332433

[20] William Goddard and Robert Cercos. 2015. Playful Hacking within Research-through-Design. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction* (Parkville, VIC, Australia) *(OzCHI '15)*. Association for Computing Machinery, New York, NY, USA, 333–337. https://doi.org/10.1145/2838739.2838802

[21] Dan Hill. 2003. iPod and adaptive design. https://www.cityofsound.com/blog/2003/11/ipod_and_adapti.html

[22] Kashmir Hill and Surya Mattu. 2018. The House That Spied on Me. https://gizmodo.com/the-house-that-spied-on-me-1822429852

[23] Andrés Lucero, Audrey Desjardins, and Carman Neustaedter. 2021. *Longitudinal First-Person HCI Research Methods*. Springer International Publishing, Cham, 79–99. https://doi.org/10.1007/978-3-030-67322-2_5

[24] Carman Neustaedter and Phoebe Sengers. 2012. Autobiographical Design in HCI Research: Designing and Learning through Use-It-Yourself. In *Proceedings of the Designing Interactive Systems Conference* (Newcastle Upon Tyne, United Kingdom) *(DIS '12)*. Association for Computing Machinery, New York, NY, USA, 514–523. https://doi.org/10.1145/2317956.2318034

[25] William Odom, Ron Wakkary, Youn-kyung Lim, Audrey Desjardins, Bart Hengeveld, and Richard Banks. 2016. From Research Prototype to Research Product. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2549–2561. https://doi.org/10.1145/2858036.2858447

[26] Doenja Oogjes and Ron Wakkary. 2022. Weaving Stories: Toward Repertoires for Designing Things. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (, New Orleans, LA, USA,) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 98, 21 pages. https://doi.org/10.1145/3491102.3501901

[27] Daniela Petrelli, Nick Dulake, Mark Marshall, Matt Willox, Fabio Caparrelli, and Robin Goldberg. 2014. Prototyping Tangibles: Exploring Form and Interaction. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction* (Munich, Germany) *(TEI '14)*. Association for Computing Machinery, New York, NY, USA, 41–48. https://doi.org/10.1145/2540930.2540966

[28] James Pierce. 2016. Design Proposal for a Wireless Derouter: Speculatively Engaging Digitally Disconnected Space. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (Brisbane, QLD, Australia) *(DIS '16)*. Association for Computing Machinery, New York, NY, USA, 388–402. https://doi.org/10.1145/2901790.2901908

[29] Tijmen Schep. 2019. Candle - privacy friendly smart home. https://www.candlesmarthome.com/

[30] Tim Shaw and John Bowers. 2015. Public Making: Artistic Strategies for Working with Museum Collections, Technologies and Publics. In *Proceedings of the 21st International Symposium on Electronic Art.* (Vancouver, Canada) *(ISEA '15)*.

[31] Nick Taylor and Loraine Clarke. 2018. Everybody's Hacking: Participation and the Mainstreaming of Hackathons. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173746

[32] Richmond Y. Wong, Jason Caleb Valdez, Ashten Alexander, Ariel Chiang, Olivia Quesada, and James Pierce. 2023. Broadening Privacy and Surveillance: Eliciting Interconnected Values with a Scenarios Workbook on Smart Home Cameras. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference* (Pittsburgh, PA, USA) *(DIS '23)*. Association for Computing Machinery, New York, NY, USA, 1093–1113. https://doi.org/10.1145/3563657.3596012

[33] Virginia Woolf. 1929. *A Room of One' s Own*. Hogarth Press, London. ISSN: 01406736.

[34] Shoshana Zuboff. 2019. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. Profile.