Envisioning, designing, and implementing the user interface require a comprehensive understanding of interaction technologies. In this forum we scout trends and discuss new technologies with the potential to influence interaction design. **— Albrecht Schmidt, Editor**

Understanding and Researching Through Making—A Plea for Functional Prototypes

Albrecht Schmidt, University of Stuttgart

Over the past 20 years, our community has embraced low-fidelity prototypes. We see many researchers using paper prototyping, mock-ups, and sketches to explore their ideas. It is easy to do and there are many good reasons for low-fidelity prototyping [1]; however, in exploring new routes in human-computer interaction, this is only the first step. In my experience, low-fidelity prototypes are helpful in killing bad ideas early in the design process but are insufficient in validating ideas and concepts—in particular, new interaction technologies beyond the classical (touch)screen. Many researchers, though, stop at the easy-to-do low-fidelity prototype and do not move to the next level: functional prototypes. Different forms of prototyping can help narrow the search space for a solution in different phases in the process (Figure 1). It is important to understand that the type of prototype we use strongly affects what type of user interaction is created and what type of feedback is received, as already shown in [2].

In particular, when designing fundamentally new interaction devices and techniques in the context of ubiquitous and wearable computing, being able to try things out is essential to understanding the user experience. For tangible interfaces, the experience is most often linked to the functionality, and a mock-up, a Play-Doh implementation, or a paper prototype will prevent researchers from doing a useful evaluation. If SIGCHI papers are any indication, we seem to overlook this, getting easily excited about low-fidelity prototypes when they are well evaluated. The rationale is clear: A mock-up is much easier and cheaper (in time and cost) to make than a functional prototype. In addition, studies and experiments are often more complex and less clean with functional prototypes. Even though this creates more and deeper insights, it seems harder to publish this work—the community does not value the additional effort. Hence, one can publish more when not graduating to functional prototypes. But I think failing to leverage the power of functional prototyping limits the impact of HCI research. Functional prototypes allow one to explore how a new technology works and how it feels to use it. Additionally, the process of creating functional prototypes helps provide insights about the design space and the prototype's impact on the user.

Building functional devices is always more difficult and complex than making a mock-up with clay or sketching a paper prototype. However, I argue that with advances in electronics and embedded networked computing, the effort required for developing functional interactive devices has decreased. Creating a functional tangible interface or an interactive wearable device is now feasible in hours rather than in days or weeks. Involving users and customers in tweaking the prototype and changing the functionality while assessing it is now fairly simple. If the researcher is skilled, changing the sensors used or the response pattern of a motor, or adding an additional component can be done during a session together with the user—very similar to updating a paper prototype. And as the size and power consumption has gone down, even running studies has become much more feasible.

Finally, when researching new forms of interaction and developing new experiences for smart tangible objects, wearable computing, and smart environments, functional working prototypes are a powerful means of understanding the design space and of communicating with the user. Figure 2 depicts a design space for functional prototypes.

Insights

- → Making functional prototypes is a source of inspiration, understanding, and reflection.
- → The HCI community could increase its impact through making working prototypes that allow users to experience new forms of interactions.
- → Low-fidelity prototyping is only a first step—we should not stop there and, instead, move on to making experiences.

INSPIRATION, UNDERSTANDING, AND REFLECTION THROUGH MAKING

When building prototypes, we typically face the question of how much time and effort to invest. If we look at it purely from a user-centered evaluation perspective, we would aim for the least effort in prototyping. If we can assess a specific set of questions with a sketch or a paper prototype, then creating a complete screen design and an implementation of some feature may be a waste of time. However, in many of our projects, making was fundamental to our understanding and hugely inspirational, both as a process and in communication with users.

When implementing a functional prototype—even if the form factor may be far from the anticipated product—we consider many issues that would be overlooked if we only sketched it or acted it out. It typically takes longer, but this extra time for implementing functions helps us to reflect on the ideas and concepts. Creating functional physical prototypes often means choosing sensors, picking the right actuator, thinking of an approach for how to program it, and considering the implications of networking and power choices. In doing this, the researchers and developers are exposed to many options (e.g., a catalog of sensors or a website with actuators), hence triggering new ideas.

Our HCI community seems to largely overlook the value of making functional prototypes. In our experience, publishing functional working prototypes with limited evaluations is typically harder than publishing non-functional low-fidelity prototypes. Reviewers' expectations are much higher when presented with working prototypes. Though functional, these higher-fidelity prototypes are still far from the final vision; therefore, they're much easier to criticize than their non-functional counterparts. As reviewers, we should place more value on the insights and reflections gained by making and experiencing functionality.

WHAT TO BUILD?

As seen in Figure 2, there is no single typical functional prototype; rather, there is a prototype space. We have to decide what properties to design for, what functions to include, and what requirements must be met. These choices will affect how much time and effort one must spend on building the prototype and what we can learn from it. Hutchinson et al. [3] make a good argument for creating technology probes, in this case made functional by simple technical implementations.

The requirements we state for a functional prototype have a clear impact on how much effort will be needed to build it and on the prototype's potential uses. In our work, we have experienced that size and weight, connectivity, as well as battery life, play a major role—in the utility and versatility of the prototype and for the effort required to build it.

- *Size and weight.* For handheld objects and wearable devices, building a prototype in the anticipated form factor of the final product often massively increases the effort required. By relaxing this requirement, the making gets much easier. Consider prototyping a head-worn notification display to be included in glasses. Creating a prototype that consists of the display attached to glasses cabled to a board in a backpack is way easier and much faster to implement than the actual form factor. However, if you want to do an in-the-wild user study over several days, the backpack-based version will give you very different results from a fully integrated version.
- *Connectivity.* Many ideas for new devices envision wireless communication between an object and the environment. When prototyping, we make the decision whether to make the prototype wireless or to just attach a cable; if we make it wireless, we have to decide what protocol to use. Using some of the hardware described in the sidebar on the next page, making a wireless connection becomes really easy; however, taking shortcuts with the protocols (e.g., fixed pairing of devices, no discovery, etc.) can save much effort.
- *Battery life.* The time between recharging a device or a smart object is very relevant for products. Creating a clever power-management system takes a lot of effort. In prototyping, the basic approach is to enable the experience we want to evaluate. For example, if we want to study a new interaction device and we run sessions of three hours where users explore the functionality and utility, it is completely fine to create a device with a battery runtime of only three hours. This means the experimenter has to change the battery after each participant, but this does not affect what we learn. In contrast, if we have a device we give people for a long-term study to be used at home, the battery and its recharging needs become more of an issue.

The design of a functional prototype is strongly dependent on the research questions we want to ask. Typically, we go for the simplest

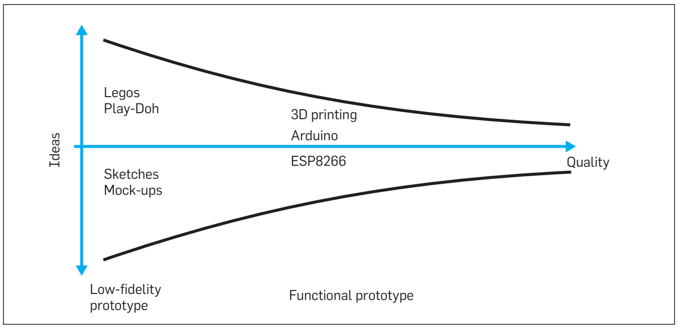


Figure 1. Low-fidelity prototyping helps to quickly narrow down ideas. Functional prototypes, however, are essential when researching tangible interaction, wearable computing devices, smart environments, and ubiquitous computing systems to validate the proposed benefit and to assess the user experience.

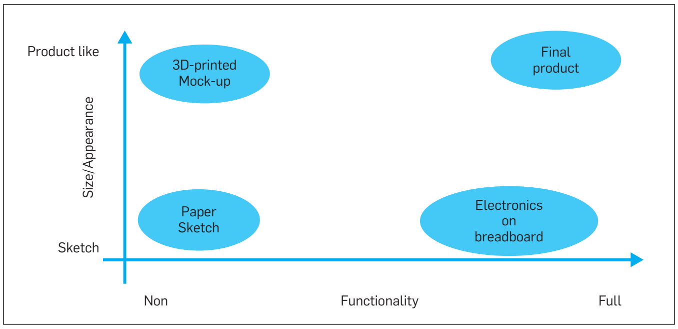


Figure 2. On the horizontal axis, we categorize prototypes based on how much of the desired functionality is implemented. On the vertical axis, we assess how closely the size and appearance resemble the target design. The lower left corner represents a non-functional low-fidelity prototype; the upper right corner, the final product.

implementation that allows us to assess these questions.

HOW TO CREATE FUNCTIONAL PROTOTYPES

What technologies to use for creating functional prototypes is an important decision, as it affects what we can build, what we can communicate and explore, and how much effort is required in the process. The following four basic approaches are common:

- *Repackaged* off-the-shelf devices
- *Systems* of standard hardware components
- *Add-ons* to common hardware components
- *Custom hardware* development.

Repackaging devices has a surprising effect on users. A mobile phone is no longer a mobile phone once it is packed into a different form factor, given a different affordance, and restricted to a specific software. Figure 3 shows an example of the Loupe [4], a functional augmented-reality prototype for exploring objects in an exhibition. Technically it is a phone in a wooden case that resembles a magnifying glass. (More details can be found in this video: https://vimeo.com/202213531.) Through the combination of small, powerful devices (e.g., mobile phones) and 3D printing, laser cutting, or CNC milling (widely available in fablabs), the hurdle to create such prototypes is low. In some cases, it makes sense to first remove the original housing. This makes the

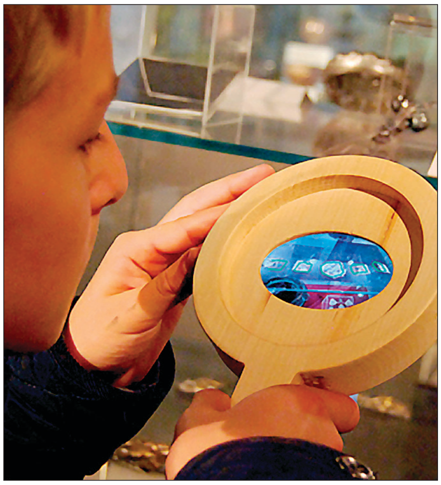


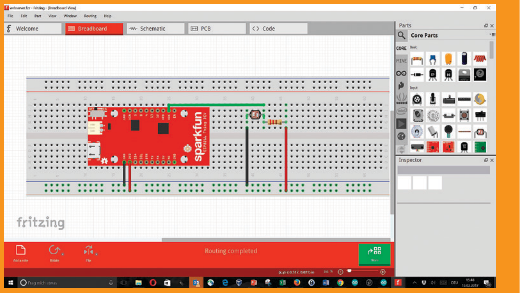
Figure 3. An example of a functional prototype using a newly repackaged mobile phone. In this example, an augmented-reality application on the phone is presented to the user as a magnifying glass.

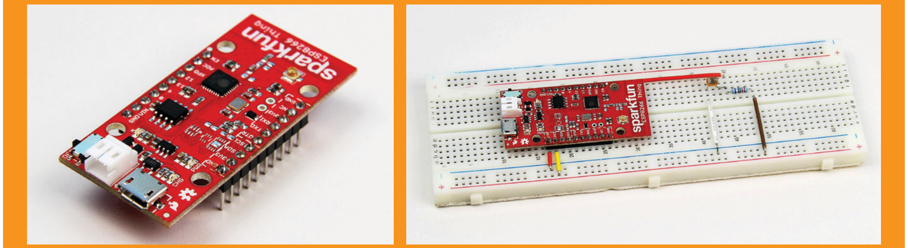
FIGURE 3 IMAGE BY WAAG SOCIETY, 2014, PROJECT-PROJECT, PHOTOS TAKEN DURING A TRAIL AUF MESCH-TECHNOLOGY AT THE NATIONAL MUSEUM OF HISTORY IN SOFIA, LICENSE: HTTP://CREATIVECOMMONS.ORG/LICENSES/BY-NC-SA/4.0/

TOOLS AND COMPONENTS TO GET STARTED

Here we show an example of a development environment (Fritzing) and a wireless computer module (ESP8266) that are good starting points for functional prototyping. I hope they make you curious to try; the threshold for entry is really low. Even for someone with no prior knowledge of electronics or embedded systems, it has become feasible to develop custom hardware for a specific task.

Fritzing (http://fritzing.org/) is a prototyping tool that allows you to create simple electronic circuits like those on a breadboard. Typically they are linked to a processing component such as an Arduino or ESP8266. The circuits can be built virtually, then programmed and exported to make real hardware. With little prior knowledge, one can design (and ship for production) a custom Arduino shield in a few hours.





The **ESP8266 module** is a microcontroller with a variety of input and output pins and with a Wi-Fi module (including a TCP/IP stack). It can be programmed with the Arduino IDE, and libraries are available to create a Web server or Web client with only a few lines.

This computer is available in different modules. The version pictured also offers USB. It is priced between $3 and $12. This is an all-inclusive module with wireless. (https://espressif.com/ https://openhomeautomation.net/getting-started-esp8266/)

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

// set SSID and password
const char* password = "aSecretWord";
const char* ssid = "myWLAN";

int analogPin = A0;    // analog pin for a sensor
int analogVal = 0;     // variable for storing the analog value
ESP8266WebServer server(80);  // the web server

void handleRequest() { // return a plain text page with text and analog value
  String returnStr = "hello world... A0 = " + analogRead(analogPin);
  server.send(200, "text/plain", returnStr);
}

void setup(void){
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  // link a path to a function that returns content
  server.on("/", handleRequest);
  server.begin();
  Serial.println("HTTP server started");
}

void loop(void){
  server.handleClient();
}
```

The source code shows the complete implementation of a Web server on the ESP8266 that replies to an HTTP-request with a plain-text page that includes a value read from the analog input.

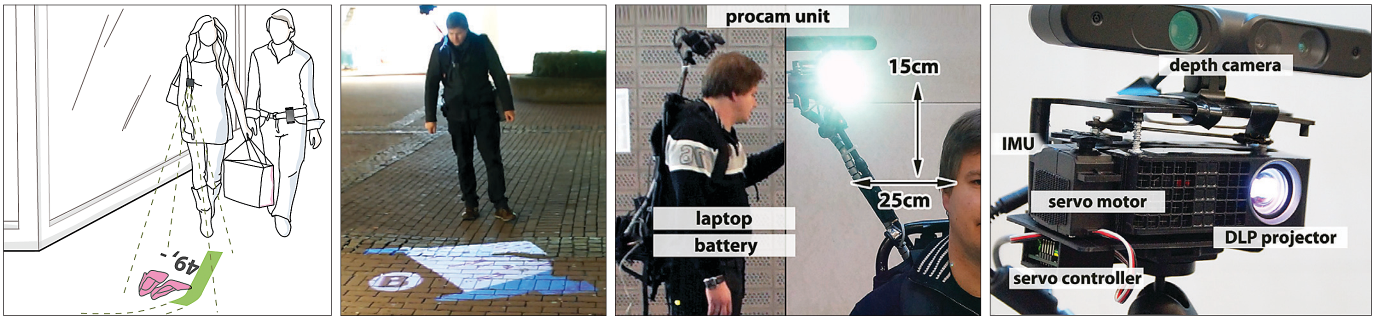


Figure 4. A wearable camera project system created based on standard components. The left image shows the vision of the system in the form factor of a small clip-on device. The remaining images show the actual functional prototype built on a backpack [5].

size slightly smaller and sometimes allows components to be redistributed, for example, taking a phone apart and separating the screen from the mainboard and battery to make it thinner. A major advantage of using off-the-shelf-devices is that they are easy to program and create content for, as when using HTML in a browser on a phone.

Building systems using standard hardware components is another option that also allows for easy programming and content provision. An example is shown in Figure 4, where a computer, a depth-sensor, and a projector are combined into a wearable prototype with a much bigger size than the anticipated product. Creating such prototypes is typically fast and requires no specific hardware skills. As the form factor is different, it is useful to assess research questions linked to the main function and less about the overall experience. Here, too, access to fablabs eases the creation.

In many cases, especially when creating prototypes that are in form and function close to the anticipated product, one has to develop custom hardware. This may be a complete custom hardware design for a specific device, but most often it is created using standard components for processing and communication. The custom-built aspect is then the sensing and actuation elements, as well as the shape of the prototype. To create such prototypes, basic hardware skills are required. An example is to use an Arduino Nano as a processing and data-acquisition core, a Bluetooth module for communication, a set of custom-designed sensors to detect interaction, and a motor as a specific output. The advantage of using a module (e.g., Arduino) as the core is that programming becomes much easier, and libraries likely already exist for the custom sensors and actuators. One has to keep in mind, though, that the size of a custom-designed device is typically much larger than that of a highly integrated product, such as a smart watch or a phone.

GET MAKING!

Functional prototyping offers great opportunities. It helps researchers and designers to better understand and communicate their ideas. Functional prototypes complement low-fidelity prototypes; they are, in a way, the next step after narrowing down the solution space with sketches, paper prototypes, and mock-ups.

Functional prototypes offer a deeper level of engagement with an idea or concept and help create an interactive user experience close to what is envisioned. There is a broad range of how prototypes can be created, from dressing commercially available devices into new casings to custom-developed hardware. With current advances in electronic components, in particular for processing, input/output, and communication, the bar has been massively lowered for custom-designed functional prototypes.

On a meta level, we as reviewers of academic papers should understand and reflect on the differences between prototypes when making decisions on whether to accept work to be published. No one would ask for product-like qualities from a mock-up or a paper prototype; however, we tend to expect this from a functional prototype and are hence less generous in their assessment. Motivating and incentivizing students and researchers to venture on to making could strongly increase the impact HCI has on industry.

Endnotes

1. Rettig, M. Prototyping for tiny fingers. *Comm. of the ACM 37*, 4 (1994), 21–27.
2. Van Buskirk, R. and Moroney, B.W. Extending prototyping. *IBM Systems Journal 42*, 4 (2003), 613-623.
3. Hutchinson, H., Mackay, W., Westerlund, B., Bederson, B.B., Druin, A., Plaisant, C., ... and Eiderbäck, B. Technology probes: Inspiring design for and with families. *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2003, 17–24.
4. van der Vaart, M. and Damala, A. Through the Loupe: Visitor engagement with a primarily text-based handheld AR application. *Proc. of Digital Heritage 2015*. Granada, 2015, 565–572. DOI: 10.1109/DigitalHeritage.2015.7419574
5. Winkler, C., Seifert, J., Dobbelstein, D., and Rukzio, E. Pervasive information through constant personal projection: The ambient mobile pervasive display (AMP-D). *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, 2014, 4117–4126. DOI: http://dx.doi.org/10.1145/2556288.2557365

Albrecht Schmidt is a professor of human-computer interaction and cognitive systems at the University of Stuttgart. His research interests are at the intersection of ubiquitous computing and human-computer interaction, including large-display systems, mobile and embedded interaction, and tools to augment the human mind. He has a Ph.D. from Lancaster University.
→ albrecht.schmidt@vis.uni-stuttgart.de

DOI: 10.1145/3058498 © 2017 ACM 1072-5520/17/05 $15.00