

DIY Design for Social Robots

Cesar Vandevelde^{*1,4}, Francis wyffels^{†2}, Bram Vanderborght^{‡3,4}, and
Jelle Saldien^{§1,4}

¹Department of Industrial Systems Engineering and Product Design,
Ghent University campus Kortrijk

²Department of Electronics and Information Systems, Ghent
University

³Department of Mechanical Engineering, Vrije Universiteit Brussel

⁴Partner of Flanders Make

Abstract

One of the chief obstacles in achieving wider acceptance of robotics is that only experienced roboticists can develop robotics applications. If we want robots in our homes and offices, we need more tools and platforms that reduce the costs of prototyping robots them, in terms of both time and money. The open-source paradigm offers a potential solution to these key factors. However, creating open-source robotics hardware does not just mean making the design files available online. It is essential to design the hardware in such a way that it can be built and modified by non-expert users. In this article we summarize our experiences of four years of creating open-source robotics in academia that led to the social robot Ono and the Opsoro design toolkit for social robots. We detail our design approach, leveraging DIY-friendly techniques to create systems that, though complex, can be assembled and modified by novices. We describe four experiments, two focusing on the assembly of an open-source robot and two using our toolkit to create novel social robot embodiments. They show that the key elements to attract novices are the ability to build, hack and use a social robot platform at different levels of difficulty. We believe that the open-source approach holds much promise in robotics research, though this approach is not without its challenges. The main bottlenecks are: the lack of time for ancillary activities related to open-source, the

^{*}cesar.vandevelde@ugent.be

[†]francis.wyffels@ugent.be

[‡]bram.vanderborght@vub.ac.be

[§]jelle.saldien@ugent.be

difficulty of building communities around niche research topics and the challenge of consolidating open hardware approaches with traditional business models.

1 Introduction

Social interaction between humans and robots is currently the subject of much scientific research. This is perhaps unsurprising, as technology – both software and hardware – has evolved to the point where practical applications of robots in daily life become more feasible. Research has shown that the embodiment of a robot has a far-reaching impact on the way that robot is perceived by humans [1]. Goetz *et al.* [2] show that a robot’s appearance and demeanor can have a meaningful impact on the user’s willingness to cooperate with a robot. Li *et al.* [3] demonstrate the effect of appearance on the likability of a robot. Generally, it is also accepted that physical embodiment enhances a robot’s social presence [1], [4], and that tactile interaction is a key mode of interaction in HRI scenarios [5]. Bartneck *et al.* [6] argue that while physical embodiments are no better at expressing emotions than virtual agents, virtual agents do not have the same ability to interact with the physical world. For instance, the robot Travis [7] exploits this property to incorporate a smartphone in the embodiment design in a way that is meaningful for social interaction.

The appearance of a robot is important because it directly influences the user’s expectations about the robot’s behavior and mental state, and because human-robot interaction is enhanced by an attractive or interesting appearance [8], [9]. While some work is being done to explore the effects of embodiment design in human-robot interaction (e.g. [7], [10]–[12]), most experiments consider the appearance of the robot as an external constraint, focusing their experimentation on other aspects of interaction. The current state of affairs is that many different studies are being conducted using the same robots (e.g. Nao). This is understandable considering the downsides of building custom robots for an experiment, such as the monetary cost, the time effort, and the robustness and reliability of new prototype robots.

To address the current difficulties of designing custom social robots, we identify an open-source, DIY-friendly toolkit approach as one solution. Back in 2001, Hippel [13] demonstrated the potential of toolkits as a method for enabling user innovation, especially in niche applications and in “markets of one”. Within the field of human-computer interaction, the toolkit approach has already been widely adopted. Examples include reacTable [14], littleBits [15], and PumpSpark [16]. Within the field of CHI, many projects embrace

a DIY/hacking paradigm as part of the system [17], [18].

In recent years, we have also started to see the impact of DIY and open-source methodologies within robotics. To begin, there are open-source projects that serve as a building block intended to be integrated into a larger robotic platform. Examples include operating system ROS [19], the OpenHand manipulator [20], and TakkTile touch sensor [21]. Secondly, there are complete robots, the hardware design files of which have been released under an open-source license. Examples are the iCub humanoid [22] and the quadruped Oncilla [23]. It should be noted that while these robots are open-source, they are not necessarily designed with easy reproduction in mind. To illustrate this, the iCub robot contains many CNC-milled, as well as molded plastic parts, necessitating a very well equipped lab to copy the design.

More recently, new platforms have emerged that place a more explicit emphasis on community-driven modifications and development. For instance, the Poppy project [24] focuses on robot designs based on 3D printed components in conjunction with Dynamixel-brand smart servos. The use of 3D printing enables quick and accurate reproduction of parts, but also allows the designs to be altered quickly. Currently, the project offers three designs: a 6 DOF arm, a 13 DOF upper torso, and a 25 DOF humanoid robot. Finally, robot kits have seen a rise in popularity as a tool for STEM (Science – Technology – Engineering – Mathematics) education. The best-known example is probably Mindstorms, a closed platform based on the Lego construction toy. Thymio [25] is a recent example of an open-source robot designed for education. The platform consists of a mass-produced mobile unit that is to be extended by students using papercraft and Lego add-ons.



Figure 1: Social robot Ono



Figure 2: Ono internals with Opsoro modules

The work we describe in this article finds its basis in our experiences with

conventional social robots, mainly the robot Probo [26]. The experiments with Probo showed a substantial discrepancy between what the designers had envisioned as essential functionality, and the level of functionality that was actually used by the therapists. The robot had been conceived as an advanced research platform, of which only one copy would be built. As it turns out, many interaction experiments with children required only basic robot functionality [27], [28]. This insight prompted us to design a new, simplified social robot (shown in fig. 1), as we found that others were also dealing with similar issues. Our goals for the design of the Ono robot [29] were to create a robot that is inexpensive, reproducible, modular, easy to repair, and easy to transport. Many of these challenges were met by taking cues from contemporary DIY paradigms such as the Maker Movement [30] and the open-source hardware movement.

At a certain point in the development of Ono, we decided to integrate the technology behind the robot (fig. 2) into a DIY toolkit for the design of social robots focused on face-to-face communication. The platform – named Opsoro (Open Platform for Social Robotics) – allows non-experts to design, build and program new social robot embodiments [31]. The design files of the Opsoro toolkit, as well as the Ono robot are open-source, and can be found on GitHub¹.

In the design of our DIY toolkit, we identify the following challenges:

- **Openness:** Users should be free to modify any part of the system should they so desire. The effects on the barriers to modification should also be considered when making design decisions. Certain components or techniques (e.g. CNC milling) might lead to a better-performing design, but are less accessible to amateurs, thus hampering reproduction and adaptation.
- **Easy to build:** By building the system themselves, users become more experienced in the design and functioning of the robot. We expect that this experience will make users more confident in repairing and modifying the robot. To reach this goal, it is important to reduce the knowledge and skill requirements by making the designs easy to build, ultimately enabling a larger audience to build their own robots.
- **Low cost:** Cost is often an important barrier in the adoption of robotic systems. Many social robots are only affordable for large universities and research centers. Even then, the number of robots is often limited, hampering large-scale experiments. With open-source hardware, component costs can pose a barrier for replication and modification,

¹<https://github.com/OPSORO>

which, in turn, hampers the evolutionary process that drives open-source projects.

- **Community-oriented:** We want the platform to become a true, self-sustaining, open-source hardware project. Currently, most contributions originate from the original creators, though we hope to see more contributions coming from members of the community in the future. This is important because we cannot anticipate all potential uses of the platform, and because as researchers, we are limited in the amount of time we can invest in development work.

For now, HRI researchers are chosen as the primary target users of our platform. The toolkit is not yet developed to a degree where it is completely bug-free and feature-complete. Consequently, the research market is much more manageable for beta-testing the platform. Researchers are generally open to using an unfinished product. They usually have some technical background or have access to colleagues with a technical background, and they can be easily reached for additional support, if necessary. In this sense, researchers serve as the platform’s lead users. However, this researcher audience is very small, hampering the community-driven evolution process we hope to accomplish. We see the market of therapy, education, and pastime as potential future audiences to extend the number of users. While we have already had some preliminary success with users from this section of the public, more development is required before we can engage this audience in actively using our platform.

The Opsoro toolkit and the Poppy platform [24] have some similarities. Both platforms combine off-the-shelf components with 3D-printed custom parts, both use a small single-board computer, and both are controllable through a web-based interface. While both platforms can be used to study HRI, different aspects are emphasized. Poppy focuses on locomotion and body gestures, whereas Opsoro specifically targets face-to-face communication and emotions. Our platform also targets a different price range: an Opsoro robot costs roughly 550 EUR to build, whereas a Poppy torso retails at 5300 EUR. This is due to a number of factors: (1) we do not have actuated limbs, which allows the use of RC hobby servos instead of the more expensive Dynamixel motors, and (2) our custom parts are made using laser-cutting supplemented with low-cost FDM 3D-printing. These techniques are much cheaper than selective laser sintering.

The rest of this article is structured as follows: in section 2, we discuss the technical implementation of the platform. Next, section 3 expands upon the strategies we employ to achieve a DIY reproducible design. In section 4, we detail current dissemination efforts, including four experiments where

non-experts were tasked with building robots using the platform. Finally, in section 5, we conclude with a reflection on the current status of the project, as well as challenges of an open hardware approaches in robotics research.

2 System Architecture

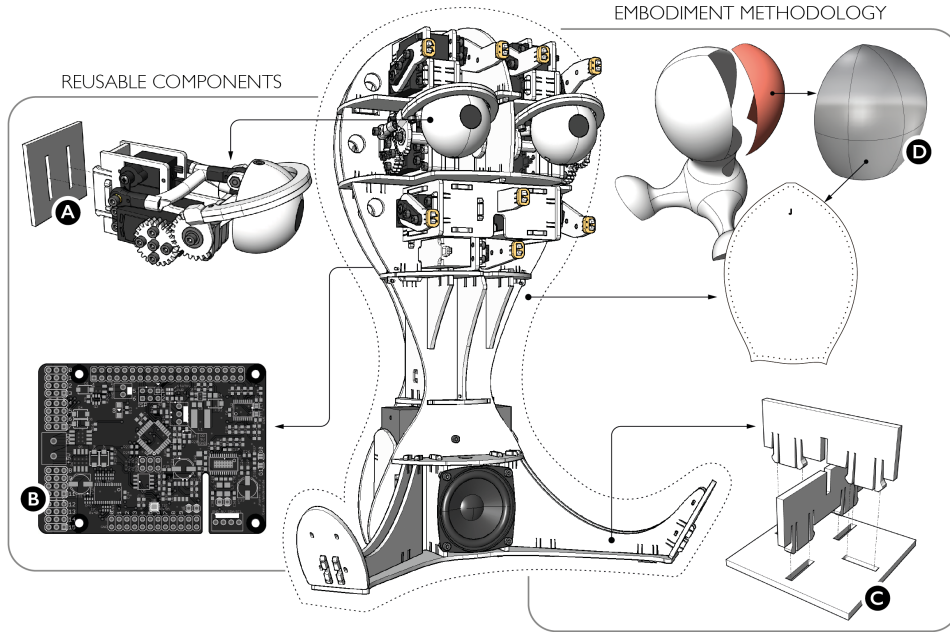


Figure 3: Architecture of the robot, showing (A) modules, (B) electronics, (C) skeleton, and (D) foam & skin

Designing a modular system requires careful balancing of two extremes. On the one hand, the system designer may choose to make modules completely multifunctional and generic, so that they can be used in nearly every situation. On the other hand, the designer can opt in favor of modules that are more specialized and prescriptive, resulting in a system that allows the user to achieve certain goals more quickly. Both approaches have merit, and the appropriate position in this spectrum is dependent on the intended use of the toolkit.

With our platform, we do not strive to design a system that is applicable for every type of robot. Instead, we focus on small-scale social robots, with a specific consideration for face-to-face communication. As a result, the components of our toolkit are specialized toward HRI applications, rather than aimed at general-purpose robotic applications. In the Opsoro system, we distinguish between two categories of components. The first group in-

cludes components that are largely the same between different social robots. These components can be used in most embodiment designs with nearly no modification.

The second group covers the parts of the robot that have a very large impact on the embodiment design of the robot. Rather than attempting to fulfill this role with generic, reusable components, we chose to incorporate a methodology centered around digital manufacturing techniques as part of our toolkit. The methodology gives users a step-by-step guideline to go from embodiment concept to custom-designed robot parts [31].

Figure 3 shows a high-level overview of the different components of an Opsoro robot. The left-hand side of the figure shows the reusable components of the system. This includes (A) the modules and (B) the robot’s electronics. The modules implement specific elements of facial features in self-contained building blocks. The eye module is shown in the figure. The toolkit also offers eyebrow, mouth, and joint modules, with more module types planned. The modules interface with the frame of the robot using laser-cut snap connectors

The electronics of Opsoro robots are comprised of a Raspberry Pi single-board computer paired with a custom daughterboard. The daughterboard was purpose-built to give the Raspberry Pi the ability to interface with different sensors and actuators, bringing the robot to life. The board can drive 16 RC hobby servos, one 5 W speaker, and a strip of addressable RGB LEDs. Sensing capabilities include 12 channels for capacitive touch sensors, as well as 4 generic analog inputs. The Raspberry Pi can interface directly with camera modules and USB devices, enabling even more extension options. The software of the toolkit is written in Python, and runs directly on the Raspberry Pi. Users can control the robot through a web-based interface, which is accessible through the robot’s built-in WiFi network. The software offers four programming options for custom behaviors. Ranked in order of increasing complexity, they include: (1) using built-in apps, (2) using a visual programming environment based on Blockly², (3) using Lua scripts, and (4) directly using the Python API.

The right-hand side of figure 3 depicts the embodiment design methodology, including (C) the design of the skeleton and (D) the design of the robot’s skin. The design methodology leverages digital manufacturing technology to enable rapid production of high-fidelity embodiments. The skeleton is made from intersecting pieces that fit together via interlocking cantilever snaps. The design is created using a 3D model of the outer design as a guide, and is produced from 3mm ABS sheet using a laser-cutter. Figure 3C shows the principle in detail.

²Blockly – <https://developers.google.com/blockly/>

The skin of the robot is comprised of a 2cm foam layer covered by flexible textile. The skin patterns are developed from the same 3D model as the skeleton. The foam shell is made from multiple laser-cut foam parts that are sewn together. A software tool is used to flatten 3D surfaces into 2D contours, taking care to minimize total distortion. The same steps are repeated for the fabric patterns, though different distortion parameters are used. Currently, the designs are made using standard CAD tools. However, the method is a prime candidate for automation due to the repetitive, formulaic nature of the process. We consider programming a custom design tool in the future.

3 Design Strategies

Ono and the Opsoro platform have gone through multiple design iterations, from which we gained different insights into the successes and failures of non-experts who design and build robots. The maker/hacker-centric approach lends itself well to quick design iterations due to the use of digital manufacturing techniques. A summary of this approach is given in this section. In our designs, we (1) use standardized components wherever appropriate and (2) manufacture all custom components using digital fabrication techniques. These two constraints serve to improve the reproducibility of the designs. We rely on digital manufacturing to produce components for our designs. Other solutions are to create designs that can be made using hand tools or designs that use only off-the-shelf components.

The first point is self-evident: using standard components is often cheaper and faster than creating a custom part for the same purpose. The second point deserves some elaboration. In the case of digital fabrication techniques, the information required to produce a part is contained within a digital file. Computer-controlled machines use this data to produce physical parts, requiring little skill or craftsmanship from the operator. This facilitates online sharing, lowers the barrier to copying, and offers a high degree of repeatability. Another intrinsic property of digital fabrication is that design complexity is (nearly) free: printing a complex object takes the same time as printing a simple object of the same volume. This property can be exploited to incorporate extra functionality in the geometry of a part.

Within digital fabrication technologies, we focus on laser cutting and low-end FDM 3D printing (i.e. RepRap [32] and derivatives) as these two technologies are commonly available through FabLabs or online services. These two techniques are complementary: 3D printers are well suited for producing small, complex, three-dimensional parts, whereas laser cutters are fast and

work well to produce larger, stronger parts. However, laser cutting is limited to flat parts. In our designs, the majority of custom parts are produced using laser cutting ($> 90\%$ by part count), supplemented with 3D-printed parts for complex mechanisms and structures. By taking advantage of digital manufacturing techniques, we can incorporate extra functionality in our custom part. This can lead to a reduced part count, simpler assembly, improved cable management, etc. Table 1 shows an overview of connections made possible through clever manipulations of custom component geometry.

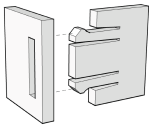
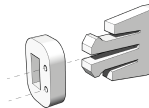
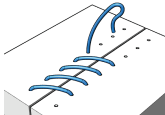
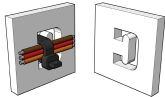

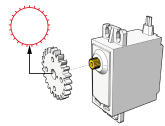
Figure	Name	Notes
	1. Snap connector ABS – ABS	A reversible cantilever snap is used extensively to make 90° T-shaped connections between two laser-cut parts. Used for connecting the different parts of the frame and for connecting modules to the frame.
	2. Textile snap ABS – Textile	A variant on connection 1 is used to attach the outer textile to the modules. A small laser-cut receiving part is sewn directly to the textile. This receiving part mates with its counterpart in the modules.
	3. Stitch pattern Foam – Foam	The foam padding layer of Ono is made from flat, laser-cut foam parts that are sewn together to form a three-dimensional shell. This 2D pattern is generated by flattening the 3D shape in software. The border of the foam parts is punctured by the laser at 1cm intervals, facilitating the manual sewing process.
	4. Zip tie anchor ABS – Cable	Feature to attach wiring to the frame using a zip tie. Previously, two parallel slots were used but this proved troublesome as zip ties needed to be inserted before assembly. Because of the “dog bone” shape, zip ties can be attached post-assembly.
	5. Nut trap 3D print – fastener	Hexagonal pockets are used in 3D printed parts to connect to fasteners. A hex nut is pressed into the pocket, after which something else can be attached to the printed part using a screw. This method is much more reliable than cutting threads directly into the printed part. The technique is also used extensively in 3D printed parts of the RepRap project.
	6. Servo spline ABS – servo	Hobby servos use a splined output shaft with triangular teeth to transfer torque to the output lever. While the dimensions of servos are standardized, those of the horns that come with servos are not. We solve this by cutting a radial pattern of lines in the circumference of a circular hole. This melts the plastic in such a way that the hole mates with triangular teeth of the spline. Cutting the triangular teeth directly does not work, as the geometry is too small.

Table 1: 3D printed and laser-cut connections in the Opsoro system.

Wherever possible, assembly information is embedded into the part geometry. Multiple methods can be used for this. To begin with, all laser-cut and 3D-printed parts are fitted with engraved annotations, indicating part numbers and orientation. This is useful to distinguish similar parts and helps when referring to a part in written documentation. Many parts are purposefully made asymmetric, so that they can only be assembled in one way.

Figure 4 shows how asymmetry can be used to enforce correct orientation.

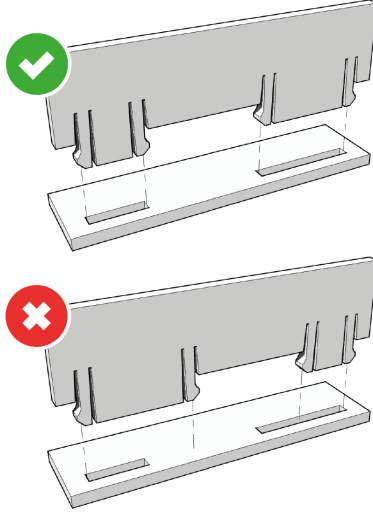


Figure 4: Using asymmetry to improve the assembly process.

Laser-cut parts are made from ABS sheet material with one textured and one smooth side. Though not a deliberate choice, the texture makes it very easy to distinguish between mirror parts. This is especially useful because laser-cut parts always have at least one plane of symmetry, parallel to the plastic sheet. Laser-cut parts of the same sub-assembly are also left connected to each other via small bridges, similar to a sprue tree of a model kit. Of course, the embedded information is not sufficient to document the assembly process completely. The main documentation is provided through a wiki, including photos, written instructions, and 3D models.

4 Experiments and Platform Dissemination

Creating successful open-source hardware necessitates more than merely making CAD files available. It is key to design with replication by others in mind in order to stimulate wider acceptance and adaptation. To stimulate this process, we organized a series of experiments in the form of workshops. The workshops serve a dual purpose: to test the design and assembly processes of the toolkit, and to kickstart a community by attracting potential users to the workshops. In our experiments, we investigate the assembly of “standard” Ono robots and the design of novel social robots using our toolkit. Details on the four experiments are summarized in table 2.

	Assembly		Design	
	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Description	workshop at UNN summer school	workshop at summer school on social HRI	studio at TEI conference	design course Illusion of Life
Scope	assemble 1 Ono	assemble 6 Onos	create robot using Opsoro + craft materials	create robot using Opsoro + CAD modeling
Soldering?	Yes	Yes	No	No
CAD modeling?	No	No	No	Yes
Behavior programming	standard apps	Lua scripting	Blockly	standard apps
Duration	1 day	1.5 days	1 day	1 semester
Participants	15	20	6	20
Instructors	2	2	2	2
Groups	1	6	2	10
Audience	social science students	HRI researchers	HCI researchers	industrial design students
When?	September 2014	August 2015	February 2015	Feb-May 2015

Table 2: Overview of experiments

The two assembly experiments involved participants who had little or no experience in constructing physical objects. Still, in both cases, participants were successful and assembled working robots in the allotted time. Follow-up interviews revealed that in both workshops, participants noted that they liked how the workshop taught them a variety of practical skills, such as soldering, in a short amount of time and applied to a realistic project. During the workshops, we provided participants with assembly instruction handouts. However, we found the most effective technique to teach the assembly process was to allow participants recreate the robot based on the example. The modular design also proved advantageous during the workshop: it stimulates parallelization because participants can organize themselves into groups focused on building one specific sub-assembly. Naturally, there are limits to this method: eventually subassemblies need to be joined together, at which point the groups are forced to converge.

In experiments 3 and 4, participants used the Opsoro modules to prototype new social robot designs. Both experiments had the same objective: to come up with an application for a social robot, to design a character around that task, and finally to create a working robot using the toolkit. However, there was a large difference in time scale: one day vs 3 months. Consequently, experiment 3 used quick-and-dirty prototyping techniques based on craft materials, whereas in experiment 4, students had the time and infrastructure to create high-fidelity prototypes. The results of the semester-long experiment are shown in figure 5. In both experiments, we saw that participants tried to build basic actuated limbs, though the hobby servos proved insufficient for this task, revealing a weakness in the current toolkit.

The longer time span of experiment 4 allowed more time to be devoted to the



Figure 5: The 10 robots designed during the student course.

concept, the character design, and the materialization of the embodiment. This was not possible in a one-day workshop. During the course assignment, inter-team collaboration was encouraged. Teams were explicitly allowed to exchange materials and module designs. This encouraged students to think about their designs in a more generalized sense: *“Is this module useful just for me or could it be used by others as well?”*. During the assignment, students also experimented with novel materials for the robot’s skin, including felt, vacuum-formed EVA foam, and rigid polystyrene. We had not thought of these techniques, and we think this illustrates the potential of community contributions. Further results of experiment 4 are detailed in [31].

Recently, we have had some success in attracting a small community of Opsoro users. This is mainly the result of addressing community-related factors, including organizing workshops, improving documentation, offering better software, and improving communication through social media. Currently, we know of 10 Opsoro robots “in the wild”. The Ono robot has also already been used by a third party in published HRI research [33]. We are also starting to receive the software contributions from outside users: developers have contributed three new apps to our software platform, as well as a binding to interface Opsoro with ROS [33]. The interface with ROS is interesting because it offers many reusable software building blocks, facilitating the process of prototyping with robot software.

5 Discussion & Conclusion

One challenge researchers often face is the lack of time for activities that do not directly lead to publishable results. However, successful open-source projects require many ancillary activities, including documentation, support, and community building. More so, to go from an experimental setup to an open-source hardware project requires a significant amount of time for development and documentation work. Another factor is that hardware created for research purposes tends to be very specific, resulting in a niche audience of users.

We have chosen to prepare to start a spin-off company around the Opsoro platform. The spin-off company would allow us to develop a stable platform with a sustainable community, as well as find ways of scaling the production of toolkits beyond the scope of FabLabs. Still, the commercialization process is not without its own challenges. One of the most difficult questions is balancing open-source practices with creating a sustainable business revenue. A number of companies, such as Arduino and Ultimaker, have had success with this, though the open-source hardware business model is new and each company requires a different approach.

During experiments, we have identified a number of weaknesses in the current platform. To begin, the toolkit has currently been used in “low-fidelity” (experiment 3) and “high-fidelity” (experiment 4) prototyping scenarios. However, as part of future work, we want to develop a middle-ground approach that is more durable and more attractive than cardboard embodiments, yet simpler to design than the custom CAD-based designs. A software tool to automatically generate laser-cutting plans from an embodiment design could be part of the solution. Secondly, the project’s documentation could be improved further. For instance, video documentation of the assembly process would make it much easier for an individual to independently copy the robot. Improving this aspect could encourage propagation of the project.

To summarize, in this paper we have detailed our experiences with open-source robot hardware in an academic context. We have described the strategies we used in the design of low-cost, DIY social robots, elaborating upon the design and production techniques we employed. We have also described the results of a series of experiments where novices build and modify these robots, as well as our efforts to disseminate the platform. Finally, we discussed the challenges we perceive for the proliferation of open-source robotics hardware to succeed. While some roadblocks remain, the open-source approach could facilitate and accelerate future robotics research, enabling studies that would otherwise have been too difficult, or indeed too expensive, to complete. We remain convinced that the open-source hardware paradigm

holds much potential in academic research, both within the field of robotics and beyond.

References

- [1] J. Wainer, D. J. Feil-Seifer, D. a. Shell, and M. J. Matarić, “The role of physical embodiment in human-robot interaction,” in *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, 2006, pp. 117–122.
- [2] J. Goetz, S. Kiesler, and A. Powers, “Matching robot appearance and behavior to tasks to improve human-robot cooperation,” in *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, 2003, pp. 55–60.
- [3] D. Li, P. L. P. Rau, and Y. Li, “A cross-cultural study: effect of robot appearance and task,” *International Journal of Social Robotics*, vol. 2, no. 2, pp. 175–186, 2010.
- [4] I. Leite, A. A. Pereira, C. Martinho, and A. Paiva, “Are emotional robots more fun to play with?” In *Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*, 2008, pp. 77–82.
- [5] K. M. Lee, Y. Jung, J. Kim, and S. R. Kim, “Are physically embodied social agents better than disembodied social agents?: the effects of physical embodiment, tactile interaction, and people’s loneliness in human-robot interaction,” *International Journal of Human Computer Studies*, vol. 64, no. 10, pp. 962–973, 2006.
- [6] C. Bartneck, J. Reichenbach, and A. Breemen, “In your face, robot! the influence of a character’s embodiment on how users perceive its emotional expressions,” in *Proceedings of the 4th Design and Emotion Conference*, Ankara, 2004.
- [7] G. Hoffman, “Dumb robots, smart phones: a case study of music listening companionship,” in *Proceedings of the 2012 IEEE International Workshop on Robot and Human Interactive Communication*, 2012, pp. 358–363.
- [8] I. Leite, C. Martinho, and A. Paiva, “Social robots for long-term interaction: a survey,” *International Journal of Social Robotics*, vol. 5, no. 2, pp. 291–308, 2013.

- [9] C. F. Disalvo, F. Gemperle, J. Forlizzi, and S. Kiesler, “All robots are not created equal: the design and perception of humanoid robot heads,” in *Proceedings of the 4th conference on designing interactive systems: Processes, practices, methods, and techniques*, 2002, pp. 321–326.
- [10] A. De Beir, H.-l. Cao, P. Gomez, G. Van De Perre, and B. Vanderborght, “Enhancing nao expression of emotions using pluggable eye-brows,” in *Proceedings of the 1st International Conference on Social Robots in Therapy and Education*, Almere, 2015, pp. 36–38.
- [11] A. De Beir and B. Vanderborght, “Evolutionary method for robot morphology : case study of social robot probot,” in *Proceedings of the 11th ACM/IEEE International Conference on Human Robot Interaction*, Christchurch, New Zealand, 2016, pp. 609–610.
- [12] J. K. Westlund, J. J. Lee, L. Plummer, *et al.*, “Tega: a social robot,” in *Proceedings of the 11th International Conference on Human-Robot Interaction*, 2016, p. 8370.
- [13] E. von Hippel, “Perspective: user toolkits for innovation,” *Journal of Product Innovation Management*, vol. 18, no. 4, pp. 247–257, 2001.
- [14] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner, “The reactable: exploring the synergy between live music performance and tabletop tangible interfaces,” in *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, 2007, pp. 139–146.
- [15] A. Bdeir, “Electronics as material : littlebits,” in *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, 2011, pp. 3–6.
- [16] P. H. Dietz, G. Reyes, and D. Kim, “The pumpspark fountain development kit,” in *Proceedings of the 2014 International Conference on Designing Interactive Systems*, New York, New York, USA: ACM Press, Jun. 2014, pp. 259–266.
- [17] D. A. Mellis, S. Jacoby, L. Buechley, H. Perner-wilson, and J. Qi, “Microcontrollers as material: crafting circuits with paper, conductive ink, electronic components, and an toolkit,” in *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, 2013, pp. 83–90.
- [18] A. Williams, A. Gibb, and D. Weekly, “Research with a hacker ethos: what diy means for tangible interaction research,” *Interactions*, vol. 19, no. 2, pp. 14–19, 2012.
- [19] M. Quigley, K. Conley, B. Gerkey, J. FAust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Mg, “Ros: an open-source robot operating system,” in *Open-source software workshop of the International Conference on Robotics and Automation*, Kobe, Japan, 2009.

- [20] R. R. Ma, L. U. Odhner, and A. M. Dollar, “A modular, open-source 3d printed underactuated hand,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 2737–2743.
- [21] Y. Tenzer, L. P. Jentoft, and R. D. Howe, “The feel of mems barometers: inexpensive and easily customized tactile array sensors,” *IEEE Robotics and Automation Magazine*, vol. 21, no. 3, pp. 89–95, 2014.
- [22] N. G. Tsagarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, F. Becchi, L. Righetti, A. J. Ijspeert, M. C. Carrozza, and D. G. Caldwell, “Icub: the design and realization of an open humanoid platform for cognitive and neuroscience research,” *Journal of Advanced Robotics*, vol. 21, no. 10, pp. 1151–1175, 2007.
- [23] A. Sproewitz, L. Kuechler, A. Tuleu, M. Ajallooeian, M. D’Haene, R. Moeckel, and A. J. Ijspeert, “Oncilla robot, a light-weight bio-inspired quadruped robot for fast locomotion in rough terrain,” in *Symposium on Adaptive Motion of Animals and Machines (AMAM2011)*, 2011, pp. 63–64.
- [24] M. Lapeyre, P. Rouanet, J. Grizou, S. Nguyen, F. Depraetre, A. Le Falher, and P.-y. Oudeyer, “Poppy project: open-source fabrication of 3d printed humanoid robot for science , education and art,” in *Digital Intelligence 2014*, Nantes, 2014, p. 6.
- [25] F. Riedo, P. Rétornaz, L. Bergeron, N. Nyffeler, and F. Mondada, “A two years informal learning experience using the thymio robot,” in *Advances in Autonomous Mini Robots*, Springer, 2012, pp. 37–48.
- [26] K. Goris, J. Saldien, B. Vanderborght, and D. Lefeber, “How to achieve the huggable behavior of the social robot probot? a reflection on the actuators,” *Mechatronics*, vol. 21, no. 3, pp. 490–500, 2011.
- [27] B. Vanderborght, R. Simut, J. Saldien, C. Pop, A. S. Rusu, S. Pintea, D. Lefeber, and D. O. David, “Using the social robot probot as a social story telling agent for children with asd,” *Interaction Studies*, vol. 13, no. 3, pp. 348–372, 2012.
- [28] C. A. Pop, R. Simut, S. Pintea, J. Saldien, A. Rusu, D. David, J. Vanderfaeillie, D. Lefeber, and B. Vanderborght, “Can the social robot probot help children with autism to identify situation-based emotions? a series of single case experiments,” *International Journal of Humanoid Robotics*, vol. 10, no. 3, p. 1 350 025, 2013.
- [29] C. Vandevelde, J. Saldien, C. Ciocci, and B. Vanderborght, “Systems overview of ono: a diy reproducible open source social robot,” in *Lecture Notes in Computer Science*, G. Herrmann, M. J. Pearson, A. Lenz, P. Bremner, A. Spiers, and L. U, Eds., vol. 8239, Springer International Publishing, 2013, pp. 311–320.

- [30] D. Dougherty, “The maker movement,” *Innovations*, vol. 7, no. 3, pp. 11–14, 2012.
- [31] C. Vandevelde and J. Saldien, “An open platform for the design of social robot embodiments for face-to-face communication,” in *Proceedings of the 11th International Conference on Human-Robot Interaction*, Christchurch, New Zealand, 2016.
- [32] R. Jones, P. Haufe, E. Sells, P. Iravani, V. Olliver, C. Palmer, and A. Bowyer, “Reprap – the replicating rapid prototyper,” *Robotica*, vol. 29, no. Special Issue 01, pp. 177–191, 2011.
- [33] I. Zubrycki and G. Granosik, “Understanding therapists’ needs and attitudes towards robotic support. the roboterapia project,” *International Journal of Social Robotics*, 2016.