



## The Design of UDOO Boards: Contributing to the Appropriation of Digital Technology

This is the peer reviewed version of the following article:

*Original:*

Rizzo, A., Caporali, M., Conti, D., Montefoschi, F., Burrese, G., & Sinopoli, B. (2019). The Design of UDOO Boards: Contributing to the Appropriation of Digital Technology. FRONTIERS IN ICT, 6 [10.3389/fict.2019.00004].

*Availability:*

This version is available <http://hdl.handle.net/11365/1122111> since 2020-12-16T15:35:16Z

*Published:*

DOI:10.3389/fict.2019.00004

*Terms of use:*

Open Access

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. Works made available under a Creative Commons license can be used according to the terms and conditions of said license.

For all terms of use and more information see the publisher's website.

(Article begins on next page)



# The Design of UDOO Boards: Contributing to the Appropriation of Digital Technology

Antonio Rizzo<sup>1\*</sup>, Maurizio Caporali<sup>2</sup>, Daniele Conti<sup>3</sup>, Francesco Montefoschi<sup>1</sup>, Giovanni Burresti<sup>1</sup> and Bruno Sinopoli<sup>4,5</sup>

<sup>1</sup> Department of Social Science, University of Siena, Siena, Italy, <sup>2</sup> AIDILAB, Siena, Italy, <sup>3</sup> SECO S.p.A., Arezzo, Italy,

<sup>4</sup> Department of Electrical & Systems Engineering, Washington University in Saint Louis, Saint Louis, MO, United States,

<sup>5</sup> Department of Electrical & Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, United States

The domain of Human-Computer Interaction does not only concern the design of technology that is easy to use, useful, and fancy—it has to do with our role in shaping our environment, our ecological niche that today involves the whole earth. A key concept in the interaction between humans and computing resources is that of appropriation, originally proposed by Aleksei Nikolaevich Leontiev. In the present paper we will first review the concept of appropriation and will present bricolage as a key activity for fostering appropriation. Then we will present the Makers Movement as a socio-cultural movement relevant for the process of appropriation of digital technology. Finally, we will describe our approach and vision in the design of the UDOO, a single board computer, and of a specific developing environment, UAPPI, for enabling the appropriation through meaningful activities of digital technologies.

**Keywords:** appropriation, tinkering, open-source, makers, UDOO, App Inventor

The domain of Human-Computer Interaction does not concern only the design of easy to use, useful and fancy technology. It also relates to being human and to our role in shaping our environment, our ecological niche, which today involves our whole planet. Our relationship with the environment over the last 60 years has been characterized by an unprecedented acceleration. During this period, we have altered the Earth more than any other living species, and more than we did in the previous 10,000 years, i.e., from the invention and spread of agriculture until the mid-twentieth century (MacCready, 1999). Everything from industry, to medicine, to how we work and play has been fundamentally reshaped by the technologies produced in the second half of the twentieth century.

Digital technologies are transforming, distorting, or even making possible nearly every human activity. Yet the power of any computer lies in the manner in which it processes information, a manner very different from the way in which we cope with information. It is the coupling of human and computational artifacts that allows our crucial leverage over our environment. But why are interactions between human and technology so effective in bringing about changes in ourselves and the world around us? Activity Theory, with its roots in the seminal work of Vygotsky, Leontiev, Rubinshtein, and Luria, was among the first to acknowledge the role of artifacts in human's cultural evolution and in the development of peculiar psychological functions. The heuristic role of Activity Theory in explaining the design of new technologies is indeed evident in defining and exploring the ontological root of artifacts (Bannon and Bodker, 1991; Kuutti, 1991; Engstrom, 2000; Cole and Derry, 2005; Kaptelinin and Nardi, 2006; Stetsenko, 2008) more than in identifying methods and techniques for improving interactions with specific artifacts.

## OPEN ACCESS

### Edited by:

Carlos Duarte,  
Universidade de Lisboa, Portugal

### Reviewed by:

Kiran Jot Singh,  
Chandigarh University, India  
Massimo Mecella,  
Sapienza University of Rome, Italy

### \*Correspondence:

Antonio Rizzo  
rizzo@unisi.it

### Specialty section:

This article was submitted to  
Human-Media Interaction,  
a section of the journal  
Frontiers in ICT

**Received:** 08 September 2018

**Accepted:** 19 February 2019

**Published:** 18 March 2019

### Citation:

Rizzo A, Caporali M, Conti D, Montefoschi F, Burresti G and Sinopoli B (2019) The Design of UDOO Boards: Contributing to the Appropriation of Digital Technology. *Front. ICT* 6:4. doi: 10.3389/fict.2019.00004

In the following, we will use some heuristic principles of Activity Theory (Kaptelinin and Nardi, 2012) to describe the origin and vision of a recent enabling technology for the Makers Movement, the UDOO boards. The rationale for addressing the Makers Community as a target for our design process and related products is due to a three-fold link between the Movement and Activity Theory: (i) both encourage everyone to see themselves as producers, not just consumers of artifacts (e.g., to go beyond end-user development toward the invention of new products); (ii) both foster collaborative problem-solving and the sharing of creative work openly with others; and (iii) both claim that making is crucial to expressing and building our personal identities. Moreover, the Makers Movement is a growing worldwide undertaking that promises to play a sound and important role in the innovation of technology (Bajarin, 2014, <https://www.theatlantic.com/notes/all/2016/06/does-the-maker-movement-matter/486647/>).

To the aim of presenting the vision and mission of our set of technologies, we will first review the concept of *appropriation* initially proposed by Aleksei Nikolaevich Leontiev. Then we will present the activities of the Makers Movement as relevant for the *appropriation* of digital technology. Finally, we will describe our approach in the design of the UDOO boards hardware and software as promising resources for the *appropriation* of digital technology.

## THE APPROPRIATION OF MEANINGFUL RESOURCES

Leontiev's Activity Theory was, up to his time, the most developed and clear elaboration of the founding principle of the so-called Marxist theories of human psychology:

“it is precisely *the alteration of nature* by men, not solely nature as such, which is most essential and immediate basis of human thought, and it is in the measure that man has learned to change nature that his intelligence has increased.... The earth's surface, climate, vegetation, fauna, and the human beings themselves have infinitely changed, and all this owing to human activity, while the changes of nature in Germany which have occurred in this period of time without human interference are incalculably small” (Engels, 1883, emphasis in original).

Leontiev (1981) proposed two complementary processes to explain how human activity can achieve this new relationship with nature, Objectification and Appropriation:

“Through activity, human beings change their environments, and through that change they build up their own novel psychological functions. This relationship involves two directed, mutually complementary processes: objectification of the environment and appropriation” (Leontiev, 1981, p. 195).

More specifically, the concept of *appropriation* was introduced by Leontiev as a way to remark the role of cultural processes in respect to the biological one in the evolution of the human species:

“The fundamental difference between the process of adaptation in the proper sense of the term and the process of appropriation and mastering is that the process of biological adaptation is one of change of the organism's species characteristics and capabilities and its species behavior, whereas the process of appropriation or mastering is one that results in the individual's reproduction of historically formed human capacities and functions. That, it can be said, is the process by which man achieves in ontogenetic development that which is achieved in animals through heredity, viz., embodiment of the advances of the species' evolution in the characteristics of the individual” (Leontiev, 1981, p. 295–6).

Leontiev, as reported by Duarte (2004), also proposed three major features that characterize the process of *appropriation* that can be summarized as follows:

(i) It is an active process. The individual must perform a practical and/or cognitive activity that reproduce the social activity objectified in the social function of that object; (ii) It is a process of reproduction. The appropriation process must reproduce the human powers historically produced in the individual; and (iii) It is a socially mediated process. The appropriation process of the meaning of the cultural object is not one of solitary discovery. The individual learns this meaning in interaction with people.

Thus, in line with Vygotsky's general law of cultural development, Leontiev considers social processes at the root of cultural knowledge transmission by mean of generating new psychological functions in the individual:

“The main characteristic of the process of making-one's-own, “appropriation” or mastery... is that it creates for the person new capabilities, new psychological functions. In this it differs from the learning process of animals. When the latter is a result of individual adaptation of the species-specific behavior to the conditions of survival, making-one's-own is a process of reproduction in the characteristics of the individual of the historically emerged characteristics and capabilities of the human species.” (Leontiev, 1981, p. 420).

Nevertheless, the appropriation process for Leontiev was not confined to the transmission of historically acquired knowledge, it was also the basis for innovative way to use the “knowledge crystallized in the products of human activity.” Appropriation extends forth to the realm of external activity; the active person brings his or her own psychological world to bear upon the world of objects. This characteristic is what Leontiev named the “personality-sense” (Valsiner, 1998).

Mikhail Bakhtin elaborated more this crucial component of the appropriation process. Even though Bakhtin's target of investigation was literature (the novel, in particular), and not human psychology, while analyzing novels, he turned to wider issues of the role of language in social life:

“The word in language is half someone else's. It becomes “one's own” only when the speaker populates it with his own intention, his own accent, when he appropriates the word, adapting it to his own semantic and expressive intention. Prior to this moment of appropriation, the word does not exist in a neutral and impersonal language (it is not, after all, out of a dictionary that a speaker gets

his words!), but rather it exists in other people's mouths, in other people's contexts, serving other people's intentions: it is from there that one must take the word and make it one's own." (Bakhtin, 1981, pp. 293–294)

Appropriation, according to this view, takes place in the middle of heterogeneity of personal efforts by "social others." Words do not have meaning by themselves, are not uniform (not taken from a dictionary) but represent intentional, goal-oriented use of language for personal purposes (Tomasello, 2009). And what holds for words holds for any other social artifact. Indeed (Rabardel 2003; Beguin and Rabardel, 2000) proposes that tools have a double character: they contain components from the original design, and components from users' utilization schemes. The appropriation of artifacts is conceptualized by Rabardel as "instrumental genesis," a process which transforms the organization of the subject's activity, the tasks they self-attribute, and the artifact's characteristics. For Rabardel the *instrumental genesis* represents the continuation of design in usage. The central idea in the works of Rabardel and his group is that to understand the nature of artifacts we must separate design-for-use from design-in-use. Just as words become ours when we can go beyond other's intentions, artifacts become our own resources only with the design in use. A good appropriation process can be considered a landmark for successful design-for-use (cfr. Dix, 2007; Tchounikine, 2017).

A successful *appropriation* process is crucial for cultural evolution. Without a suitable *appropriation* of a given resource it could be very difficult to contribute to the innovation process within a given culture. But what is more perilous is that insofar as we do not go beyond the intentions potentially embodied in the artifacts, we are carrying out just other's intentions and finalities—we are at the service of the artifact at least as well as it is at our service, aside from enabling any tool constraints. Artifacts become part of our functional organs (Kaptelinin, 1996) and enable as well constrain humans to perceive and relate to the world through the artifact, even when we are not necessarily holding it in our hands (Bodker and Klokmoose, 2011). It is just this range of perceptions and relations through the functional organ that constitute the crucial issue. Artifacts are not neutral; they offer one reality at the expenses of others, one point of view among many, one possible system of categories among numerous others possible (Rizzo, 2015). Each technology offers its own opportunities for action, affordances, that make it easier to do some activities and harder to do others. The prime effect is that the easier activities get done and the harder ones are neglected. "The more successful and widespread the technology, the greater its impact upon the thought patterns of those who use it, and consequently, the greater its impact upon all of society. Technology is not neutral; it dominates." (Norman, 1993, p. 243).

Thus, if we use technology only within the range of perceptions and relations they come with, and we have no room for giving life to our own goals and intentions, we will just consolidate our role as consumers of technology. We are at the service of those who have the resources and knowledge for extending in the realm of objects their own intentions and goals. Today, many digital technologies impose constraints and

preconditions to their use. They hamper our ability to produce new meanings by transforming the material objects we interact with (e.g., the creation and use of artificial stimuli). They prevent us from *playing* with them; a crucial step in the appropriation process well exemplified in the seminal work of Vygotsky on children's play (1933/1982), who saw in play the seeds to the answer to Karl Marx and Friedrich Hegel' debate: Does the material determine the idea? Or does the idea determine the material? (Mecacci, 1979).

Vygotsky introduces his theory about children's play moving from a very early idea of affordance, that of Kurt Lewin's valence. Vygotsky quotes a study carried out by Lewin where it is shown how very young children, in the attempt to exploit the opportunities for action offered by a stone, exhibit a behavior that is strongly determined by the conditions in which the activity takes place. The behavior of these children, in attempting to use the stone as a seat, is bound in every action by the constraints the stone poses to their interaction (see <https://www.youtube.com/watch?v=3x4HWLMAwe8>).

For Vygotsky, the interaction exhibited by the children and the description provided by Kurt Lewin, is a clear illustration of the extent to which play can depart from situational constraints. He states:

"It is hard to imagine a greater contrast to Lewin's experiment showing the situational constraints on activity than what we observe in play. It is here that the child learns to act in a cognitive, rather than an externally visual, realm by relying on internal tendencies and motives and not on incentives supplied by external things... Lewin concludes that things dictate to the child what he must do: a door demands to be opened and closed, a staircase to be run up, a bell to be rung. In short, things have an inherent motivating force in respect to a very young child's actions and determine the child's behavior to such an extent that Lewin arrived at the notion of creating a psychological topology" (Vygotsky, 1978, p. 96).

As an alternative, for Vygotsky:

"...in play, things lose their motivating force. *The child sees one thing but acts differently in relation to what he sees...*in play the child creates the structure meaning/object, in which the semantic aspect—the meaning of the thing—dominates and determines his behavior. To a certain extent meaning is freed from the object with which it was directly fused before. I would say that in play a child concentrates on meaning severed from objects, but that it is not severed in real action with real objects." (Vygotsky, 1978, p. 96–97; emphasis in original).

However, in the production of new meaning the separation is not totally arbitrary:

"This is not to say that properties of things as such have no meaning. Any stick can be a horse, but, for example, a postcard can never be a horse for a child. Goethe's contention that in play anything can be anything for a child is incorrect. Of course, for adults who can make conscious use of symbols, a postcard can be a horse. If I want to show the location of something, I can put down a match and say, "This is a horse." And that would be enough.

For a child, it cannot be a horse: one must use a stick. Therefore, this is play, not symbolism. A symbol is a sign, but the stick is not the sign of a horse. Properties of things are retained, but their meaning is inverted, i.e., the idea becomes the central point. It can be said that in this structure things are moved from a dominating to a subordinate position.” (Vygotsky, 1978, p. 98).

Indeed, there is experimental evidence in pretending play with children to support that meaning is built not only in relation to the shape, color or other features of the objects involved in the activity but first of all in relation to the actions the object allows (Szokolsky, 2006). The feasible action patterns provide the cue for what an object is for, and in many cases you have to try it out, both if you are a child or an adult, to discover if they work—in pretense, function follow action. Pretense can be viewed as the unconventional use of an object in place of another in order to achieve a goal, as a dialog with objects that allow the production of pragmatic as well as epistemic actions (Kirsh and Maglio, 1994). More generally, pretending play is a privileged way of staying in touch with an object as well as stepping out of that objects to mentally modify it. To do this with digital technologies today is very difficult, insofar as for a child a poker card can become a smartphone but a smartphone with many probabilities will remain a smartphone.

This dialog with objects has been the main source of innovation in many cultures before science. In *The Savage Mind* (1962), the anthropologist Claude Lévi-Strauss used the French word *bricolage* to describe the characteristic mythological thought patterns of recruiting objects and signs already in existence for purposes that they were originally not meant for within a given culture. *Bricolage* is defined as the skill of using whatever materials are at hand and recombining them to create something new and meaningful. It involves a “dialogue with the materials and means of execution” (Lévi-Strauss, 1966, p.29). In such a dialogue, the materials may “suggest” adaptive courses of action, and the initial aim may be modified. Consequently, such acts of creation are not purely instrumental: the “Bricoleur” “speaks” not only with things... but also through the medium of things giving an account of his personality and life by the choices he makes between the limited possibilities. The “Bricoleur” may not ever complete his purpose but he always puts something of himself into it (Lévi-Strauss, 1966, p. 21, cfr. with Leontiev above). Given this, is it not surprising that Seymour Papert, the father of constructionism, when facing the critical questions of how the individual appropriates new technology, finds in Lévi-Strauss’s *bricolage* the key process. But what is more interesting is that Papert saw in the digital computer a privileged resource for carrying out *bricolage* activities:

“At the heart of the new possibilities for the appropriation of formal systems is the computational object, on the border between an abstract idea and a concrete physical object. In the simplest case, a computational object such as an icon moving on a computer screen can be defined by the most formal of rules and is thus a mathematical construct, but at the same time it is visible, almost tangible, and allows a sense of direct manipulation that only the encultured mathematician can feel in traditional formal systems. The computer has a theoretical vocation: it can make the

abstract concrete; it can bring formality down-to-earth.” (Turtle and Papert, 1990, p. 131).

In Papert’s vision children through *bricolage* activities, should be programming the computer rather than being programmed by it (Papert, 1980). However, Papert instead of sticking with the original “untranslatable French word *bricolage*” (Papert, 1993; p 143), he made a connection between the *bricoleur* and “the old-fashioned traveling tinker, the jack-of-all-trades who knocks on the door offering to fix whatever is broken.” From that moment forward this activity that merges imagination, inquiry, and goal-oriented behavior, and that exploits the interdependence between Mythological and Scientific thought (Kozulin, 1986) starts to be known as tinkering and loose some crucial aspects of the original meaning (see below). It turns out to be more and more associated to the jack-of-all-trades, to the do-it-yourself practice<sup>1</sup> and becomes the core practice of an important socio-cultural movement, the Maker Movement (Dougherty, 2012), a Movement that is not only important for the stress it poses on “*bricolage*” activities but first of all for the assonance it has with Activity Theory. Such an assonance makes this Movement an all the more relevant social phenomenon that can give live to Kaptelinin and Bannon’s vision of expanding the conceptual foundations of interaction design in the direction of “... not only helping designers create better products but also helping people themselves create better environments for their work, learning, and leisure activities.” (Kaptelinin and Bannon, 2012, p. 280).

But before introducing the Makers Movement let us sum up the heuristic principles we reported so far and that have driven our following design process: (A) Appropriation is a fundamental phenomenon of human cultural evolution. (B) Appropriation is characterized by: (i) practical activities that reproduce the social activity objectified in the social function of that object; (ii) the reproduction of the human powers historically produced in the individual with the establishment of the new relationship with his/her environment; and (iii) the social generation of meaning. (C) Appropriation is not limited to repetition of historically generated knowledge, rather it is a resource to give material form to our psychological life. (D) The passive reproduction of knowledge without much room to give voice to our inner world puts us at the service of the artifact as well as it is at our service. (E) A successful way to give life to the appropriation process is to rest on *bricolage* activities, an activity driven by spontaneous thought even when we have to face scientific concepts.

## THE MAKER MOVEMENT

The Maker Movement represents a growing movement of hobbyists, designers, hackers, and artists committed to creatively designing and building material objects for both playful and useful ends. The name and the idea of the Maker Movement can be traced to the 2005 founding of *Make* magazine and the first Maker Faire in 2006 (Martin, 2015). The key drivers have been

<sup>1</sup>In this process of moving from *Bricolage* to *Tinkering* with many chances played also a role the article written by the Nobel Prize Francis Jacob *Evolution and Tinkering on Science* (Jacob, 1977).

the introduction and the simplification of technologies like 3D printers, Arduino microcontrollers or the single board computer Raspberry Pi. These new technologies lowered the barrier to entry to tinkering with electronics, and people have begun to meet and play with these technologies in physical places like Fab Labs and Makerspaces, where they exchange knowledge under the umbrella of the open-source philosophy (Weber, 2004).

The Maker Movement has been associated with what has been named the Third Industrial Revolution (Anderson, 2013). The First Industrial Revolution, which took place from the 18th and 19th centuries, was brought about by the introduction of machines into the production cycle. This revolution changed everything. Living standards and education rose for millions of people. Global empires not only became possible, but also, for the first time, practical. Language changed, as words like job and work acquired their current-day meanings. Engels, who by the way was the first author not only to use the term repeatedly but also to endow it with the “full force of a revolutionary event” (Coleman, 1992, p. 3), saw the Industrial Revolution as a natural and essential phase in human evolution, one that would contribute to the benefit of mankind by developing a more educated, productive workforce better able to ultimately equalize the balance of power between it and the elites. Marx viewed industrialization as representing a further stage in the dehumanization of workers but at the same time as an essential stepping stone toward the revolutionary processes. In his preface to the first volume of *Das Kapital*, he wrote that “[t]he country that is more developed industrially only shows, to the less developed, the image of its own future.” If the feudal, agrarian society exploited the lower classes by enslaving them, the process of industrialization enabled the holders of capital to attain greater wealth through a more productive economic system, but always at the expense of the poor.

What was not clear at that time was that the Industrial Revolution was, first and foremost, a revolution in invention. As documented by Rosen (2010) it was not simply a huge increase in the number of new inventions but a radical transformation in the process of invention itself. The key leverages of Industrial Revolution, were not just gears, levers, and boilers, but *ideas* about gears, levers, and boilers:

“Before the eighteenth century, inventions were either created by those wealthy enough to do so as a leisure activity (or to patronize artisans to do so on their behalf), or they were kept secret for as long as possible. In England, a unique combination of law and circumstance gave artisans the incentive to invent, and in return obliged them to share the knowledge of their inventions.... Human character was changed, and changed forever, by seventeenth-century Britain’s insistence that ideas were a kind of property.” (Rosen, 2010, p. 23).

With the industrial revolution ideas become a fundamental asset of the economy of a nation (Bottomley, 2014). However, if the benefits of such industrialization were better products and accessible prices for a larger population, the cost was homogeneity in the production. The marketplace became increasingly dominated by the goods of mass production:

merchandises designed for everyone, with the resulting limited diversity and choice that implies (Anderson, 2013). The industrial production of standardized goods for mass consumption as well as the new organization of labor, pushed handmade production, at a small scale, toward the periphery of the social-economical system. This process created a market whose only aim is to support production and generate consumers. Altogether, the Makers’ Movement is seeking an alternative to being regarded as consumers, rejecting the idea that to buy is the privileged way to get something. The way in which Makers consider their relationship with technology and its evolution in respect to the society is well expressed in the Maker Movement Manifesto (Hatch, 2013). Below we report just three of the nine principles that inspire the movement plus the final gloss, from which it is easy to catch the assonance with Activity Theory:

**“MAKE:** Making is fundamental to what it means to be human. We must make, create, and express ourselves to feel whole. There is something unique about making physical things. These things are like little pieces of us and seem to embody portions of our souls.

**SHARE:** Sharing what you have made and what you know about making with others is the method by which a maker’s feeling of wholeness is achieved. You cannot make and not share.

**LEARN:** You must learn to make. You must always seek to learn more about your making. You may become a journeyman or master craftsman, but you will still learn, want to learn, and push yourself to learn new techniques, materials, and processes. Building a lifelong learning path ensures a rich and rewarding making life and, importantly, enables one to share.

*In the spirit of making, I strongly suggest that you take this manifesto, make changes to it, and make it your own. That is the point of making.*” (Hatch, 2013, p. 16–18, emphasis in original)

With the power of collective action, peer production, open source, and crowdsourcing, individual action is amplified to have extraordinary influence on the larger systems that surround us. The actions start at the smallest scale—with one person determined to make something happen—and can grow like wildfire to the extent our imaginations deem possible. The Makers Movement has the potential to turn more and more people into keeper, instead of just consumers, of the products of human labor. At the root of the Makers Movement there are just the two key factors that produced the Industrial Revolution: new affordable technologies for production and an explosion in the democratization of invention. Furthermore, most of the resources designed and produced for Makers goes just in the direction of supporting the transformation of “intrinsic practice.”—that is, transformations in human practices made possible by a new technology but envisioned not by the designer/producer but by the users of such technology (Kaptelinin and Bannon, 2012).

This state of affairs coupled with the design experience offered by the “Disappearing Computer” European Union initiative

(Streitz and Nixon, 2005), pushed us in the direction of envisioning the design of some new enabling technology to empower people in the process to appropriate digital technology and merge it with our everyday analog world.

## DESIGN FOR APPROPRIATION, THE VISION AND DESIGN OF UDOO BOARDS

### UDOO Vision

The UDOO vision can be traced back to the work carried out with Philips Design, Domus Academy and the University of Liegi in the design of an educational environment for children of the primary school for supporting the development of narrative competence (Rizzo et al., 2001; Decortis and Rizzo, 2002; Decortis, 2015). In designing such environment, we acknowledged the paramount need to be able to produce a seamless world between existing analog and new digital tools. This would have allowed children and teachers to go on with their continuous modification and transformation of their learning environment. Inspired by the work of Stewart Brand we named such vision the construction/deconstruction of environment (Rizzo and Rutgers, 2004). But it was very difficult even for the whole staff of engineers and computer scientists at Philips Design to apply such an approach. Digital technologies were not built for this. A few years later, the first digital technology designed for tinkering started to appear. At the Interaction Design Institute of Ivrea, Hernando Barragán began the development of Wiring, the grandfather of Arduino, the well-known open hardware/software microcontroller (Banzi, 2011). The tinkering process associated to Arduino was presented as a faster and more powerful way to build better prototypes; and in introducing tinkering Banzi grasped important aspects of the activity based on Bricolage:

“The Arduino philosophy is based on making designs rather than talking about them. It is a constant search for faster and more powerful ways to build better prototypes. We have explored many prototyping techniques and developed ways of thinking with our hands.

Classic engineering relies on a strict process for getting from A to B; the Arduino Way delights in the possibility of getting lost on the way and finding C instead.” (Banzi, 2011, p. 5)

Besides a facilitated way to interconnect with the physical world through sensors and actuators, the Arduino microcontroller came about with an Integrated Design Environment that allowed, and still allows, programming the microcontroller using a simplified C language. However, even though this makes it simpler and more affordable to get started interacting with the physical world through digital technologies, the main focus is introducing scientific concepts concerning coding and electronics and becoming acquainted with physical computing (O’Sullivan and Igoe, 2004). For example, the “Getting Started with Arduino” (Banzi, 2011) presents a hands-on introduction to coding and electronics by simple step-by-step tutorials for controlling LEDs and Motors by Light-sensors and Push-buttons. But unless you have already extended knowledge about

electronics and its history it is difficult to play with the *social activity objectified in the social function* of LEDs, push buttons, or the breadboard. And the same is true for the Arduino microcontroller itself.

Our goal was to facilitate the appropriation of digital technologies for people who were not necessarily studying electronics or coding. Our target was people who would prefer to have as “object” of their activity something closer to their everyday life issues and interests. For this it would be helpful to have an environment where much more mundane object-oriented activities could be carried out. This would allow our target groups to generate new feasible intentions that goes beyond the ones given, possibly since the start of the activity mediated by digital resources. Leontiev’s intuition that true appropriation starts with activities that can be contaminated by the personality-sense, requires that the object of the activity be meaningful and sensible for the people involved. And willy-nilly, with the Arduino resources, unless your desired “objects” are LEDs, motors, light sensors, push buttons and C Code, there is not much room for populating the playfield with people’s intentions that goes beyond the ones given in the tutorial.

But to continue our argument, let us be clear: The Arduino Way is a paramount step, it is a landmark in the process of appropriating digital technology, and we are just trying to build on top of this effort. We are critiquing with the aim to contribute, to complement the process started by Banzi and his colleagues. Working on projects involving physical computing, both for research and education, we had problems producing interactive prototypes that could be embedded in meaningful ways in the environments where they were supposed to work. Most of the time to achieve this we had to combine the Arduino microcontroller with a computer, with a web-service, or with an Android smartphone (Rizzo et al., 2009). We had to acknowledge that, if a computer can be considered an expressive medium that different people can make their own in their own way (Turkle and Papert, 1990), the same was not yet true for a microcontroller. And, the Arduino is a microcontroller, not a mini-computer. A microcontroller is just a small part of what makes a whole computer. Although the Arduino can be programmed with small applications written in C, it cannot run a full scale “operating system” and cannot interface with people exploiting the progress in human-computer interaction of current personal computers. The PC has evolved mostly for interacting with us and it is equipped with mature solutions that address straight away many of our needs and intentions but lacks in its real-time interaction with the physical world (O’Sullivan and Igoe, 2004). On the contrary, the microcontroller has been developed to interact with the physical world, but notwithstanding, the remarkable improvement produced by the Arduino team it remains a simple programmable controller with a simile C language (Kushner, 2011).

To go beyond the tinkering involved in getting started with Arduino, we needed to offer to our students a pool of resources that from the start, from the “hello world,” could afford the exploration and generation of new ways to use the available resources. In order to facilitate the appropriation of digital technologies for people who are not studying electronics

or coding, we had to think of something that could play a heuristic role in generating new feasible intentions. If we wanted to offer newcomers to the digital world an opportunity to appropriate computational resources that are stepping out from the virtual world of the computers and going into real activities objectified in the social function of objects (i.e., Internet of Things), we had to combine computers and microcontrollers in a seamless environment.

Thus, we thought of a board that could have the flexible management of Arduino sensors for interfacing with the physical world and the power and simplicity of interaction with the digital world of a full-fledged computer, thus to allow Tinkering with the capital T; a Tinkering made not for becoming just acquainted with physical computing but for playing with everyday life activities. We envisioned a single board that could extend the tinkering process promoted by the Arduino Way in the direction of true Bricolage activity resting on spontaneous of mythical thought by which our user “can reach brilliant unforeseen results on the intellectual plane” (Levi-Strauss, 1966, p. 17). We thought of a set of hardware and software resources that could allow a tinkering process closer to the original bricolage activity, an activity carried out for marrying spontaneous and scientific thought, for coupling imagination powered by everyday life experience with the inquiry about new digital resources that were not confined in the virtual world. A Board that could support the tinkering process at different levels of digital skills from newcomers up to engineers and designers of industrial products.

More specifically, our goals were as follows: design and produce resources that facilitate their same appropriation; make these resources relevant for objects of the activity that are meaningful for our target users, so that there is room for personality sense to express; organize the use of the resources around bricolage processes driven by spontaneous thinking at all levels, from newbie to professionals.

In the following we report on some phases of the design process concerning the UDOO boards, and about a special software UDOO App Inventor (UAPPI for short); which is the complementary software component for the integration of computational resources for tinkering at which UDOO aims.

To this aim, the UDOO hardware documentation is released under license Creative Commons CC BY-SA 3.0 and available on the UDOO web site (<http://www.udoo.org/other-resources>), and the full source code used to build UAPPI is available on GitHub (Montefoschi, 2015). The git repository contains also the whole development history, which documents how we built on top of the resulting outcome of previous research, and how we gradually modified the product following field tests and feedback from various experiences. The development history and the following description are produced so as to meet the set of criteria identified by Zimmerman et al. (2007): process, invention, relevance, and extensibility. We report below the main features of the board UDOO Quad, and provide an example of how we used feedback from the makers' community in the design process just after the release of the first prototype. Following this, we describe the hallmarks of the UAPPI Integrated Development Environment (IDE) and report

on how we contributed to the extension of MIT App Inventor in all three its parts (Component, Blocks and Devices) in order to manage within the same environment the processor and the microcontroller.

## Designing the Hardware of UDOO Quad

Following our Vision, the mission was to embed into a single board two different chips: a PC-like processor and an Arduino-compatible microcontroller, making simpler the overall management of connection and communication resources and their programming. In this way, we could aim at making available existing PC-based computational resources for managing the Arduino microcontroller (see UAPPI section below).

The first step was to define the technical specifications we wanted to implement on the board and what chips should be used. As for the main CPU, we decided to use an ARM Cortex-A9, the NPX i.MX 6. The rationale for such a choice was due to the availability of many different versions of this processor (single, dual, quad core), all which allow running Android or Linux open source operating systems. These processors are pin-to-pin compatible among them; in this way, with only one Printed Circuit Board (PCB), we could mount different core versions without changing the rest of the board. The features of the board related to the main CPU were defined so to be closer to a personal computer (see **Table 1**).

The interfaces for the most common communication protocols (i.e., I2C, SPI, CAN, UART) were also provided. Although they are not so common (or user accessible) in a personal computer, they are widely used in the embedded industry and facilitate the merge of single board computer and microcontrollers. Moreover, using such industrial components allows an easy and feasible transition from prototypes to industrial products. Finally, the exposure of GPIO pins allowed a quick and solid way to interconnect our CPU with the physical world.

The selection of the microcontroller was similarly influenced by the CPU. We wanted to avoid the use of level converters, and since the i.MX 6 CPU logic operates at 3V3, we choose a 3V3 compliant microcontroller also. We decided to use SAM3X chip, the standard one on the Arduino Due board, at that time the last member of the famous prototyping platform. In order to provide a great compatibility with Arduino IDE, we implemented all the circuitry of the Arduino Due board.

To allow users to easily access the i.MX 6 communication capabilities (like GPIO, I2C, SPI, etc.) from the microcontroller, we decided to connect them to the same pin headers of the Arduino pinout. In this way, the signal on a pin available on the external headers is connected both to the i.MX 6 processor and to the SAM3X microcontroller, a crucial feature for allowing the sharing of potentialities between the PC and the Arduino worlds.

After some weeks of work, the hardware team released the first schematics and the PCB design. This led to *Revision A*, the first prototype of the UDOO board (**Figure 1**).

At that point, we launched a crowdfunding campaign on Kickstarter. Crowdfunding differs from the traditional financing of new ventures in two important ways. First, funding is provided



**TABLE 1** | Technical specifications of the UDOO Quad Board and related specs for the Intel Galileo and the Raspberry Pi single board computer.

	UDOO Quad	Intel Galileo	Raspberry Pi B
CPU	NXP™ i.MX6 Quad	Intel Quark SoC X1000	BCM2835
Cores	4x ARM Cortex-A9 @ 1GHz	1 x86 @ 400 MHz	1x ARM @ 700MHz
Microcontroller	Atmel SAM3X8E (ARM Cortex-M3)	Embedded in SoC	–
RAM Memory	1GB DDR3 (Low Voltage)	256 MB DDR3	512 MB SDRAM
GPU	3D: Vivante GC2000 2D: Vivante GC355, Vivante GC320	–	Dual Core VideoCore IV® Multimedia Co-Processor
Video outputs	Full-HD HDMI, 18/24 bit LVDS interface	–	HDMI, Analog RCA, DSI
Camera	MIPI CSI	–	MIPI CSI
Storage	Micro SD Card, SATA, USB	8 MB NOR Flash, Micro SD card	Micro SD Card, USB
Connectivity	Gigabit Ethernet, Wi-Fi	10/100 Ethernet (PoE)	10/100 Ethernet
Audio	Headphone and Microphone 3.5 mm jacks, HDMI, SPDIF, I2S	–	3.5mm jack, HDMI, I2S
Communication	3x USB 2.0, 1x USB OTG, 1x USB-to-Serial, 4x UART, 3x SPI, 1x I2C, 4x PWM, 1x CAN Bus	1x USB 2.0, 1x USB 2.0 OTG, 1x UART, 1x mPCle, 1x I2C, 1x SPI	2x USB 2.0, 1x UART, 1x I2C, 1x SPI, 8x GPIO
Arduino Pinout	76x GPIOs, 12x Analog Inputs, 2x Analog Outputs (DAC)	20x GPIOs, 6x Analog Inputs	–
On-board Devices	RTC, Watchdog, Timer capture, Timer compare	RTC	–
Physical size	110 × 85 mm	72 × 123.8 mm	85 × 56 mm

by the relatively small contributions of many individuals over a fixed time limit (generally a few weeks). Second, potential donors are in most cases expert in the domains related to the product submitted to the campaign and prone to comment, suggest, and even propose design features. Platforms such as Kickstarter are not only a novel method for funding but also a new way to involve users and stakeholders in the prototyping phase. In the campaign, we immediately addressed the Makers community and the Kickstarter section dedicated to comments became a real forum where we discussed and elaborated with backers on features and potentialities of the

forthcoming products. We published videos and photos of the first prototypes, schematics and a detailed description of all the features implemented (Kickstarter – UDOO, 2013a,b). Soon after there was a spark of comments, many of which were questions or simply observations while others were suggestions and proposals.

Here a sample of the discussion available on the UDOO Kickstarter campaign:

*“User on April 10, 2013: This looks great, I love the idea of the two devices glued together, especially with both being full hardware implementations! The fact that it will be open hardware is what’s most exciting!”*

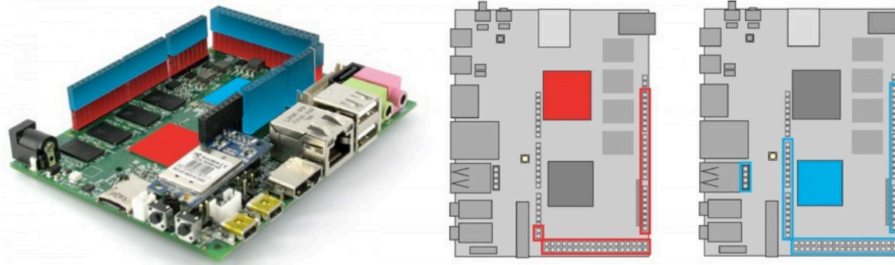
*Ian McMahon on April 16, 2013: Is it possible for the board to support I2S audio out? The raspberry pi initially was going to support it, but they cut that feature. I’d really like it to build a streaming audio target that can feed a high-end DAC an I2S or SPDIF stream.*

*Tejinder on April 10, 2013: Does the board support FIS over SATA so we can use a port multiplier to jbod or a hardware raid solution? You guys have already nailed 2 of the most important requirements of arm boards SATA and gigabit Ethernet, if FIS is supported a NAS would be so much easier.*

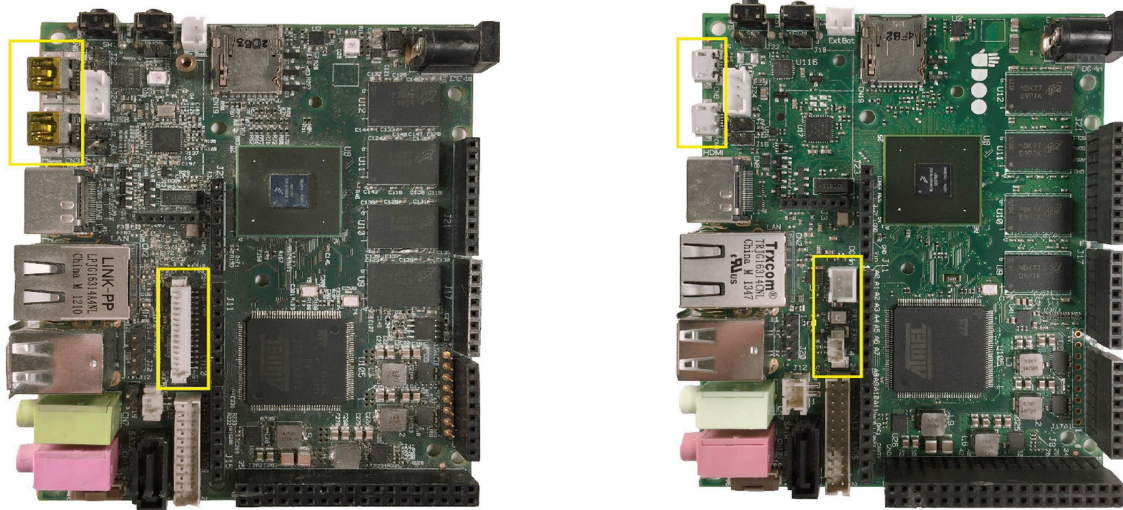
We carefully considered these suggestions and on May 9th 2013 we announced a redesign of the hardware as the first stretch goal of the campaign. The first board redesign consisted of using different USB connectors (micro instead of mini) and adding several new devices, available through pin headers: S/PDIF digital audio input and output, I2S/AC97/SSI digital audio multiplexer, FlexCAN, a second SD card (see **Figure 2**).

This kind of design approach would never have been possible without a crowdfunding platform like Kickstarter, which enables designers to showcase their prototypes to potential customers and receive their feedback in an open platform accessible to anyone. Thus, at the end of the Kickstarter campaign, we had for the first time to make available a single platform experimenting with an integrated solution between Arduino and a full computer, plus hundreds of remarks and suggestions supporting our vision and mission (and, not a secondary aspect, the already more than 4000 boards already sold). The hundreds of feedbacks received from the bakers was also a proof of how important is the sense of community within the Makers Movement and how meaningful is listening and supporting them in their everyday effort to play with the solutions we could offer. On this side we tried to learn from the Arduino and the Raspberry PI communities looking for the best practices and avoiding the pitfalls.

Moreover, in the year of UDOO launch, 2013, Intel launched Galileo, a board that integrated an Arduino microcontroller into a Linux Single Board Computer. Despite Galileo potentially could be the closest solution to our Vision, it presented two major downsides. The first issue concerns the hardware: Makers expect powerful hardware, in fact although the UDOO board was offered in different variants (Quad core, Dual core and Single core) the strict majority of the users requested the Quad core version. The technical characteristics of the Galileo board are lower than the ones of the Single core



**FIGURE 1** | The first UDOO board prototype, "Revision A." In red the i.MX6 processor with the related processing capabilities on the pin headers. In blue the SAM3X ones.



**FIGURE 2** | UDOO Quad Revision A (on the **Left**) and Revision D (on the **Right**). In the yellow squares are highlighted the components involved in the redesign.

UDOO (see technical comparison in **Table 1**). The second issue was about the community: Intel did not provide documentations nor tutorials, and did not release the technical details (internal registers, etc.) to allow users to prepare this material.

Finally a decisive difference resides in the target users and their related goals. Raspberry PI had "the goal of inspiring the next generation of programmers" (<https://www.zdnet.com/article/we-thought-wed-sell-1000-the-inside-story-of-the-raspberry-pi/>), targeting students to Computer Science degree with a low cost platform. Intel Galileo's target was not so clear, and the board looked like a demo board for their Quark X1000 SoC targeting people interested in experimenting with the emerging field of Internet of Things (<https://www.anandtech.com/show/7387/intel-announces-galileo-quark-based-arduino-compatible-developer-board>). But this declared aim clashed with the poor documentation provided and community support necessary for considering the board a valid resource for prototyping professional solution by start-ups or design studios (<https://hackaday.com/2017/06/>

[19/intel-discontinues-joule-galileo-and-edison-product-lines/](https://www.intel.com/content/www/us/en/programmable/development-hubs/19/intel-discontinues-joule-galileo-and-edison-product-lines/)).

This documentation lack has definitely a negative impact considering the fact that a full generation of embedded developers and programmers grown up with ARM architecture (<https://www.stuff.tv/features/arduino-galileo-and-quark-intels-war-arm-about-get-ugly>).

The UDOO target, on the other hand, is much wider, starting from novices to the digital world, passing through the Makers up to expert developers who build industrial prototypes. In the following we focus on the software resources we built for the newbies, but first we briefly report an example of the projects produced by people in our community.

Reports have been made on projects that utilize the main board characteristics: the cooperation between the main core CPU and the microcontroller unit, physical world interface, graphical, and computational power. Many of them were published on the platform Hackster.io.

One of these is UDOO Coffee. <https://www.hackster.io/doucoffee/doucoffee-machine-e47d65>

The UDOO Coffee project is based on a common coffee machine available in every store. It was modified using the UDOO board to enhance the user experience with

Android features like an advanced graphical interface, text to speech, and voice control. Android communicates with the microcontroller using the ADK protocol. The microcontroller is connected with the physical buttons of the coffee machine PCB to activate the pouring by relays. A control App was developed to represent a virtual barman face. Every time the user touches the touchscreen, the virtual barman asks some questions (text-to-speech) to understand the kind of coffee to pour and the sugar quantity. The user can then voice their preferences.

This project was the first stage of an industrial prototype made by high school students for answering an industrial project brief. The UDOO board allowed to easily interact and interface with an existing analog hardware reading voltages, controlling relays and connect with a pre-existing PCB. At the same time was possible to use lot of advanced services provided by Android—online and offline.

Another example is the UDOO Photobooth (<https://www.hackster.io/patsner/udoo-dslr-photobooth-1ab84c>).

This device (**Figure 4**) was produced by a Maker for his wedding Party. Turning on the UDOO the application starts automatically at boot and shows the main screen. The device is ready to work as soon as a camera is detected, then the physical big button of Photobooth starts blinking.

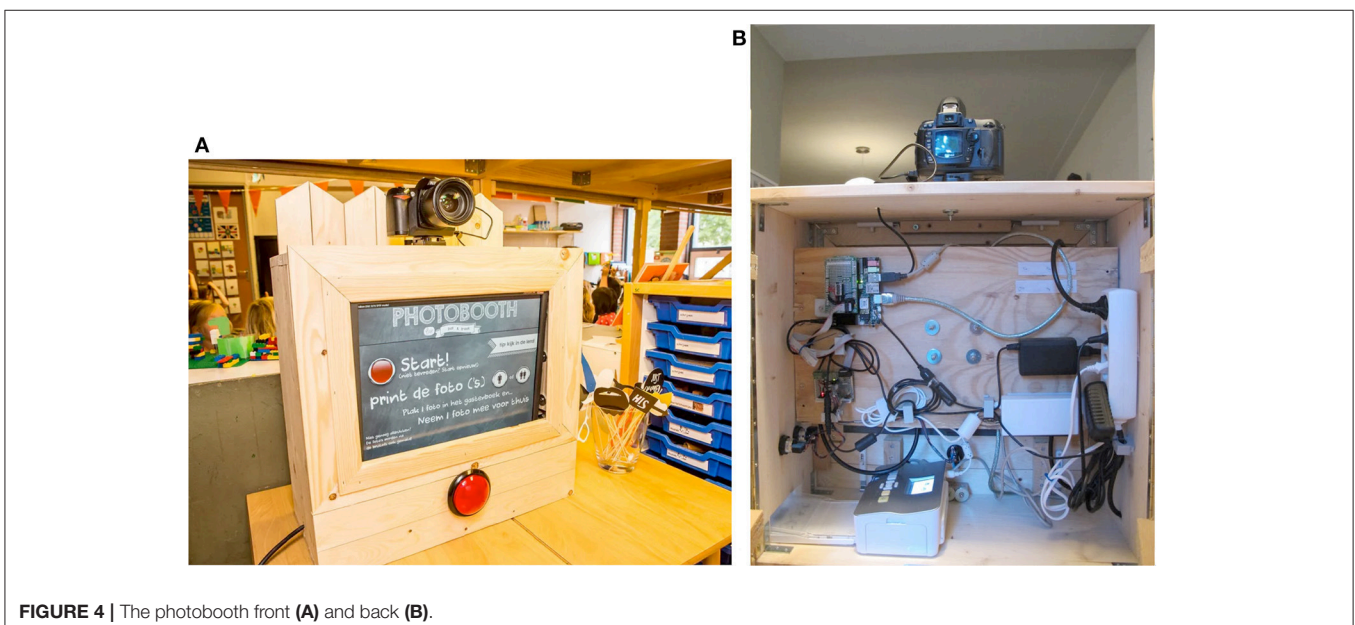
After the user push the button, the camera takes four pictures, transform them in black, and white and combines them in a nice composition showing them on the screen.

After that the user can decide to print them on a 10 × 15 photo using a Canon Dye-sub printer. In this project, the UDOO board interfaces with commercial equipment—a camera and a printer—to create a full working device.

The number of features available in the UDOO boards allows designers and developers to explore solutions, understand potentials and limits. Concepts can be turned in prototypes to be tested and evaluated. The UDOO prototype hardware can



**FIGURE 3** | The interactive coffee machine enabled by a UDOO Quad.



**FIGURE 4** | The photobooth front (A) and back (B).

be scaled and also used for the first stage of production. If the final product has success, the producer can switch to a customized industrial version of the UDOO board for the mass production with a reduced number of difference in hardware and software.

## UAPPI, the UDOO App Inventor Extension

The hardware was one side for promoting the appropriation of digital technology for newbie to the computational world. The other side was to provide easier to use and easier to learn software tools for managing the Arduino microcontroller. In order to ease the process of “design in use” (see above) and substantiate the appropriation process since the first encounter with the new hardware architecture a possible solutions would be a single developing environment that could integrate seamlessly microcontrollers into the world of personal computers. While the history of human-computer interaction is rich of solutions that step after step have extended the use of computer and made smoother their learning curve, the interaction with microcontroller has seen much less steps, with the design of

Arduino IDE as the biggest improvement. Thus, the simplest strategy was to try to rest on the intentional stances produced by the longest human-computer interaction history and extend them to the use of microcontroller. In this effort we exploited the well-established and successful experience of App Inventor.

App Inventor is an open source visual programming platform for creating mobile applications for Android-based devices. It was developed at Google Labs by a team led by Hal Abelson on sabbatical from MIT (Abelson, 2009). It can be considered the more advanced offspring of Papert’s effort to promote computational fluency for everyone. Like its cousin, Scratch (Resnick et al., 2009), the development does not require writing code in the traditional command line way. The look and behavior of the app is developed “visually,” using a series of building blocks for each intended component. But at difference with Scratch, App Inventor was designed to develop mobile applications exploiting Android devices sensors and web services that let users create apps with real life impact since their early steps (Wolber et al., 2015). App Inventor allows making a “hello world” that is a real App running on a smartphone. The App Inventor IDE has three key parts: the *Component Designer*, used to select components from the Palette for building the App and specify their properties; the *Blocks Editor*, used to specify how the components will behave (e.g., the coding of objects); and, an *Android device* on which the App can run (and be tested) even in the development phase.

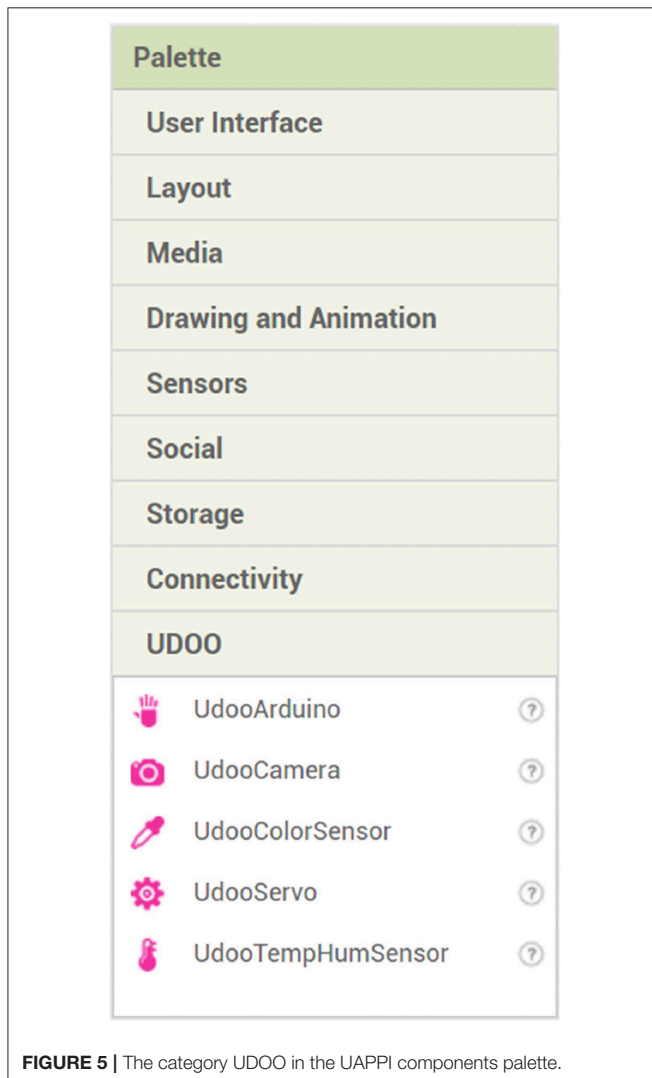
Thus, we went on designing an extension of App Inventor, UDOO App Inventor (UAPPI). UAPPI design and applications has been documented elsewhere (Rizzo et al., 2015, 2016), and it is accessible at the following link: <https://2-dot-uappi-151311.appspot.com/login/> using any gmail account. Here we want just to point up how we contributed to the three App Inventors parts: Components, Blocks and Devices, and briefly report on a latter development concerning a Machine Learning component.

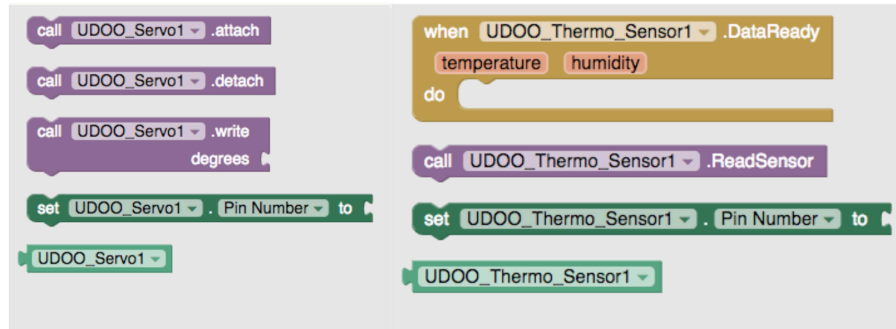
## Components

We produced a new category of components to the palette in the Component Designer named UDOO (see **Figure 3**). These components, thanks to the hardware architecture of the UDOO Board, let App Inventor users access the Arduino-side of the UDOO board, vastly simplifying the use of sensors and actuators. Component blocks mask the complexity of the C/C++ sensor libraries needed in the Arduino IDE.

UDOO’s components embody the App Inventor philosophy, simplifying the use of sensors for novices via plug-and-play blocks, and also providing flexibility for more experienced users. To this last aim, low-level calls to Arduino primitives (e.g., `digitalWrite`) can be used to develop applications with the same freedom that the Arduino IDE would provide.

The UDOO palette is composed of several components (e.g., `UdooArduino`, `UdooCamera`, `UdooColorSensor`, etc.). The `UdooArduino` establishes a connection with the Arduino-side of the board. The other components are used to access and operate Arduino sensors and actuators connected to the board. For instance, the `UdooServo` component can be used to control servo-motors (small motors able to rotate a shaft to a precise angular position). The `UdooTempHumSensor` component reads the data registers of two of the most used, lowest-cost digital





**FIGURE 6 |** The Servo component (Left) and the Thermo component (Right) with their methods.

temperature and humidity sensors, the DHT11 and DHT22<sup>2</sup> and so on. To design Android apps interacting with sensors and actuators, UAPPI users need only add those components to the project and call up the available methods, the signatures of which are often the same as in Arduino API. Having the same Arduino API helps users coming from the Arduino IDE world to develop Android apps using UAPPI, without having to learn a new interface from scratch. UDOO extensions meet both Arduino and App Inventor standards. Users can freely explore what to do with a component.

## Blocks

We developed the necessary Blocks for defining the behavior of the components. On the Blocks editor, a panel shows all the properties, methods and events for each component (no need to read API manuals) ensuring a gentle learning curve promoted by a potential easy match between user's intentions and opportunities for actions.

For example, the UdooServo components aims at controlling servo motors, DC motors embedding a position sensor, which are able to precisely control the angular position. This component has two user-configurable properties, Board and Pin Number. In case the user is designing a multi-board application, the first property allows to specify in which board the servo motor is connected to; the second one specifies in which Arduino pin the servo motor data signal is connected to.

This component exposes 3 methods in the Blocks panel: attach, detach, write (Figure 6). The first two should be called when the application is created and closed, in order to register the servo motor in the Arduino microcontroller. The most interesting method is write, which accepts an integer number of degrees as argument. When invoked, it sets the angular position of the servo motor to the desired value.

The UdooThermoSensor component is designed to work with temperature sensors, and, more in general, thermoigrometers. The properties are Board (with the same semantic as UdooServo), Pin Number (the Arduino pin number where the sensor data signal is connected to) and Sensor, which specifies the type of sensor. We currently support DHT11 and DHT22 sensors.

<sup>2</sup>How to Pick the Best Temperature Sensor for Your Arduino Project (2015). Available online at: <http://www.inorobotics.com/pick-best-temperature-sensor-arduino-project/>

This component has only one method, ReadSensor, which takes no arguments. When invoked, it reads from the sensor the temperature and eventually the humidity. This operation may take from a few milliseconds to a couple of seconds. In order to avoid locks in the GUI, we make the sensor values available to the developer in an event, DataReady, instead of directly returning the values from the method call. The DataReady event is fired with two arguments: temperature and humidity. The developer can then use those values in the application logic.

The same strategy oriented to the meet the everyday intentional stances of novices was used for the other components.

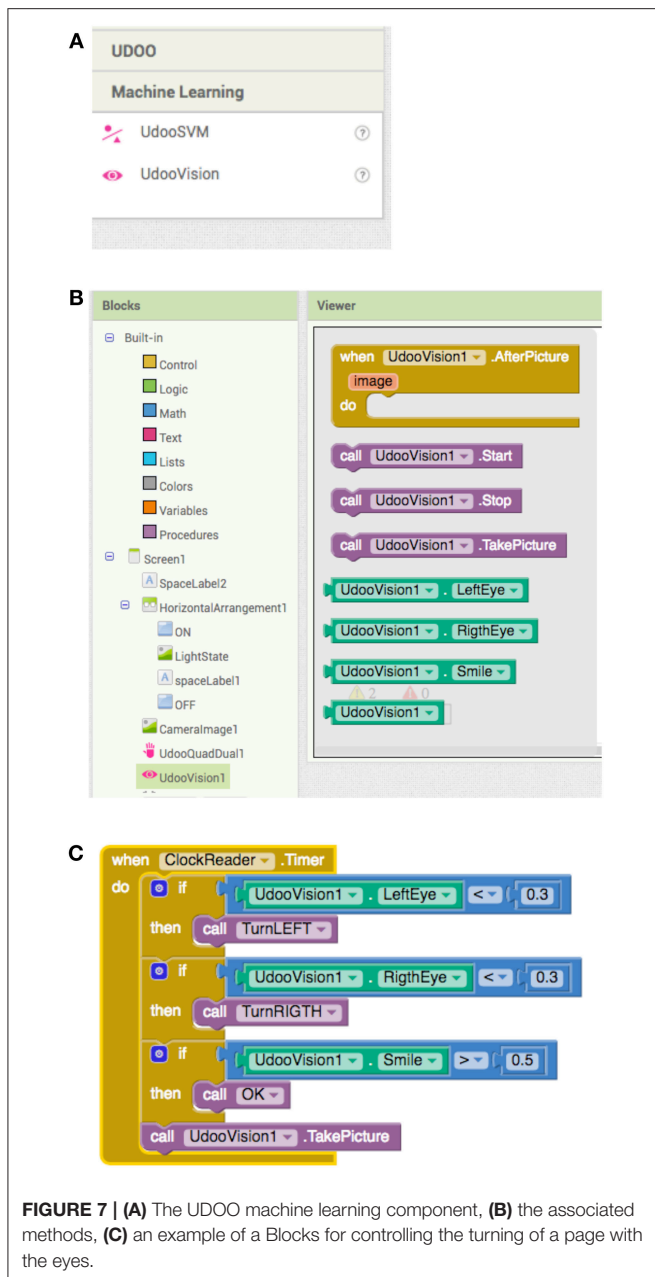
## Devices

We operated to make UDOO board as an Android Device that can run official Android Apps. The Android distribution for the UDOO board was designed to be as much compatible as possible with the standard Android OS that can be found in off-the-shelf phones and tables. Along the development the main focus was about binding the UDOO hardware to the Android Hardware, Abstraction Layers and Frameworks. In this way it has been possible to directly use the UDOO hardware by simply using the standard Android APIs. This concept is valid for all the hardware, the audio, the camera, etc. Even the Arduino microcontroller, which is not available on most Android-devices, was connected with the official Google's ADK.

Given the standardized software, the power of UDOO lives in the possibility of developing applications able to use both the traditional aspects (sound, 2D/3D graphics) and the cyber-physical ones, via the microcontroller. All of this can be done by using standard development tools, like Eclipse or Android Studio, or even MIT App Inventor.

## The Machine Learning Component

We have recently introduced a new component in the palette: UDOO ML (Figure 7), offering the opportunity to run Machine Learning models within UAPPI. Machine Learning (ML) is rapidly changing the landscape of digital technology. This new way to produce computational solutions is radically changing the capabilities of software and the interfaces people use to interact with that software, for this any attempt to contribute to the appropriation of digital technology should try to address such changing game resource.



Our first step in this direction has been to offer the opportunity to people using UDOO and UAPPI to play straight away with the models produced by machine learning algorithm. ML models can be uploaded in the UDOO Machine Learning directory and used as a special category of sensors or actuators. One of the first Model has been UDooVision (Figure 5). It is a model for face recognition that allows to detects faces in movies or picture. When a face is detected it is reported at a position with an associated size and orientation. After detection the frame with the face can be searched for landmarks such as the eyes and the mouth. In our first implementation we made available the possibility to control the state of the left eye, the right eye and the smile after a face has been detected (Figure 5). This can

allow, for example, to use the eyes and the smile as controllers for reading an ebook or other devices you want to change the conditions of interactions.

In summary, UAPPI is an integrated environment for programming that makes seamless the combination of a minicomputer running Android and an Arduino microcontroller. UDOO App Inventor gives novices the tools to develop Apps for physical computing since their early encounter with a programming environment. The synergy between the hardware (UDOO) and software (UAPPI) components turned out to be crucial for facilitating the set-up of activities that can be meaningful in the life of our target group.

## CONCLUSIONS

Appropriation is a key process for cultural evolution. Through appropriation we populate the resources we recruit from the environment with our own intentions and generate new ideas about products and solutions. The heuristic principles we used to promote the appropriation of digital technology came from the seminal work of Leontiev, Bakhtin, and Levi-Strauss and can be so summarized: focus on the intentional stance of people and provide at least some resources that can be recruited within activities that are meaningful for them, activities that can be contaminated by their personal sense. We are trying to do this bringing back Tinkering to the original idea of Bricolage. Tinkering, as introduced with the Arduino platform, is great if goals and intentions concern the *electronics* domain. Arduino contains a set of inputs and outputs that can be connected directly to actuators and sensors, and is easy to just jump straight into making something in that domain. However, although children, designer, and people in general are not necessarily interested in the basic of electronics, they are still often very interested in how their everyday life activities can be influenced by digital technology. Indeed, today much of the effort devoted to help people to become literate about digital technologies adopts a similar approach, aiming to introduce conceptual and material resources related to digital technology by allowing people to program games or play with robots, resting the potential interest of people, especially young people, in such kind of ludic activities (cf. Lye and Koh, 2014). But, our approach is a bit different, we are trying to facilitate the appropriation of digital technology from the earlier encounter up to prototyping industrial products that can easily scale to final products. And we are striving to design our open hardware and software accordingly.

It is hard to overestimate how important it is to allow the appropriation of digital technology. For a long time, we considered technology as neutral. It is often thought that is the use we make of technology is what makes technology good or bad. This view overlooks the fact that tools have not only a pragmatic role in our relationship with the world. They are not just at the service of human intentions. They also have an epistemic role, namely to evoke, induce, guide through their availability a given relationship with the world.

The epistemic role of instruments may be confined to a range of intentions and well-defined operations or can be

left open to the contingencies of human activity. This may seem a matter of little importance, but it is not. In the first case we are at the mercy of those who designed and created the technology and we are part of their “operation.” We are part of the product. In the second case, we can choose what and how to use, what to make and what to put into relationship with what else. We are not arguing that there should be no products aimed strictly for the purpose for which they have been designed to present a high degree of reliability in the related services. What we claim is that beyond this we should design for Tinkering/Bricolage, for hacking, for expressing ourselves beyond the horizon foreseen by product designers. In this we share the proposal put forward by Kaptelinin and Bannon (2012) that Interaction Design (or HCI) should expand beyond product design toward what they call Technology-Enhanced Activity Spaces (TEAS). TEAS are “spatially and temporally organized configuration of resources, including digital technologies, which enable an individual or a group to carry out one activity or several coordinated activities.” What is critical is how digital technology fits into these spaces. Insofar as we are not be able to appropriate digital technology, there will be a gap in these spaces that will hamper the design in use.

Today, the Makers Movement is the more relevant socio-technical phenomenon that is helping the new generations in the appropriation of digital technology. It is not by chance that the most important outcome of the Maker Movement is the (side) effect on education. Low cost electronic development platforms such as Arduino, Beaglebone, Raspberry PI, and the same UDOO are spreading their capabilities between appassionati, hobbyists, engineers and students. UDOO is our attempt to contribute to this phenomenon and at the same time to extend the domain of Interaction Design. We are designing both hardware and software tools so to make simpler and convenient for

novices to approach digital technologies as creators of their own solutions. The final aim is to allow people to join the new democratization of invention and to contribute to small as well big steps in the design of socio-technical innovation. We envision a future where the products, including the most sophisticated ones, could be marketed in different ways: “as they have been designed” and “just as you want them for you.” This will give the possibility to anyone to contribute to the innovation of any product—from home appliances to automotive devices—and be rewarded for their contributions. To this aim, technology (of any kind) should be designed for appropriation offering different levels of contributing to its and our evolution.

## AUTHOR CONTRIBUTIONS

AR, MC, and DC: conception of the board; DC: hardware designer; MC: product designer; FM and GB: software designer; AR and BS: literature review; AR, MF, GB, and BS: wrote sections of the manuscript. All authors contributed to manuscript revision, read and approved the submitted version.

## FUNDING

This research was partially supported by POR CREO FESR Toscana 2014–2020 CUP 7165.24052017.112000043.

## ACKNOWLEDGMENTS

We are thankful to our colleagues in SECO and Siena University who provided expertise that greatly assisted the R&D project.

We have to express our appreciation to the two reviewers for their insightful comments on the paper, as these comments led us to an improvement of the work.

## REFERENCES

- Abelson, H. (2009). *App Inventor for Android*. Google Research Blog.
- Anderson, C. (2013). *Makers*. Amsterdam: Nieuw.
- Bajarin, T. (2014). Why the maker movement is important to America's future. *Time. Com.* 19.
- Bakhtin, M. M. (1981). *The Dialogic Imagination: Four Essays*. Transl. by M. Holquist. C. Emerson. Austin, TX: University of Texas Press.
- Bannon, L., and Bodker, S. (1991). “Beyond the interface: encountering artefacts in use,” in *Designing Interaction: Psychology At The Human-Computer Interface*, ed J. Carroll (Cambridge: Cambridge University Press), 228–253.
- Banzi, M. (2011). *Getting Started With Arduino*. Cambridge: O'Reilly Media, Inc.
- Beguín, P., and Rabardel, P. (2000). Designing for instrument-mediated activity. *Scan. J. Informat. Syst.* 12, 173–190. Available online at: <https://aisel.aisnet.org/sjis/vol12/iss1/1>
- Bodker, S., and Klokmoose, C. N. (2011). The human-artifact model: an activity theoretical approach to artifact ecologies. *Hum. Comp. Interact.* 26, 315–371. doi: 10.1080/07370024.2011.626709
- Bottomley, S. (2014). *The British Patent System during the Industrial Revolution 1700–1852. From Privilege to Property, Vol. 28*. Cambridge: Cambridge University Press.
- Cole, M., and Derry, J. (2005). “We have met technology and it is us,” in *Intelligence and Technology: The Impact of Tools on the Nature and Development of Human Abilities*, eds R. J. Sternberg, and D. D. Preiss (New York, NY: Routledge), 209–228.
- Coleman, D. C. (1992). *Myth, History and the Industrial Revolution*. London: A&C Black.
- Decortis, F. (2015). *L'ergonomie Orientée Enfants, Concevoir pour le Développement*. Paris: PUF.
- Decortis, F., and Rizzo, A. (2002). New active tools for supporting narrative structures. *Personal Ubiquitous Comput.* 6, 416–429. doi: 10.1007/s007790200046
- Dix, A. (2007). “Designing for appropriation,” in *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but Not as we Know it, Vol. 2* (Swindon: BCS Learning & Development Ltd.), 27–30.
- Dougherty, D. (2012). The maker movement. *Innovations* 7, 11–14. doi: 10.1162/INOV\_a\_00135
- Duarte, N. (2004). Formação do indivíduo, consciência e alienação: o ser humano na psicologia de AN Leontiev. *Cad. Cedes* 24, 44–63. doi: 10.1590/S0101-32622004000100004
- Engels, F. (1883). *Dialectics of Nature*. New York, NY: LeoPard Books India.
- Engstrom, Y. (2000). Activity theory as a framework for analyzing and redesigning work. *Ergonomics* 43, 960–974. doi: 10.1080/001401300409143
- Hatch, M. (2013). *The Maker Movement Manifesto: Rules For Innovation In The New World of Crafters, Hackers, and Tinkerers*. New York, NY: McGraw Hill Professional.
- Jacob, F. (1977). Evolution and tinkering. *Science* 196, 1161–1166.

- Kaptelinin, V. (1996). "Activity theory: implications for human-computer interaction," in *Context and Consciousness: Activity Theory and Human-Computer Interaction* ed B. Nardi (Cambridge, MA: MIT Press).
- Kaptelinin, V., and Bannon, L. J. (2012). Interaction design beyond the product: Creating technology-enhanced activity spaces. *Hum. Comput. Interact.* 27, 277–309. doi: 10.1080/07370024.2011.646930
- Kaptelinin, V., and Nardi, B. (2012). *Activity theory in HCI: Fundamentals and Reflections. Synthesis Lectures Human-Centered Informatics*. Williston, VT: Morgan & Claypool
- Kaptelinin, V., and Nardi, B. A. (2006). *Acting With Technology: Activity Theory and Interaction Design*. Cambridge, MA: MIT Press.
- Kickstarter – UDOO (2013a). *Neo = Raspberry Pi + Arduino + Wi-Fi + BT 4.0 + Sensors*. Available online at: <https://www.kickstarter.com/projects/udoo/udoo-neo-raspberry-pi-arduino-wi-fi-bt-40-sensors>
- Kickstarter – UDOO (2013b). *Android Linux Arduino in a Tiny Single-Board Computer*. Available online at: <https://www.kickstarter.com/projects/udoo/udoo-android-linux-arduino-in-a-tiny-single-board>
- Kirsh, D., and Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cogn. Sci.* 18, 513–549.
- Kozulin, A. (1986). The concept of activity in soviet psychology: vygotsky, his disciples and critics. *Am. Psychol.* 41, 264.
- Kushner, D. (2011). "The making of arduino," in *IEEE Spectrum*, 26.
- Kuutti, K. (1991). "Activity theory and its applications to information systems research and development," in *Information Systems Research: Contemporary Approaches and Emergent Traditions*, ed H. E. Nissen (Amsterdam: Elsevier Science Publisher), 529–549.
- Leontiev, A. A. (1981). *Problems of the Development of the Mind*. Moscow: Progress Publishers.
- Lévi-Strauss, C. (1966). The scope of anthropology. *Cur. Anthropol.* 7, 112–123.
- Lye, S. Y., and Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: what is next for K-12?. *Comput. Hum. Behav.* 41, 51–61. doi: 10.1016/j.chb.2014.09.012
- MacCreedy, P. (1999). *An Ambivalent Luddite at a Technological Feast*. Designfax. Available online at: [http://www.designfax.net/~archives/0899/899trl\\_2.asp](http://www.designfax.net/~archives/0899/899trl_2.asp)
- Martin, L. (2015). The promise of the maker movement for education. *J. Pre-Coll. Eng. Edu. Res.* 5:4. doi: 10.7771/2157-9288.1099
- Mecacci, L. (1979). *Brain and History: The Relationship Between Neurophysiology and Psychology in Soviet Research*. New York, NY: Brunner-Routledge.
- Montefoschi, F. (2015). *GitHub - fmntf/appinventor-sources*: MIT App Inventor Public Open Source. Available online at: <https://github.com/fmntf/appinventor-sources>
- Norman, D. A. (1993). *Things That Make us Smart: Defending Human Attributes in the Age of the Machine*. New York, NY: Basic Books.
- O'Sullivan, D., and Igoe, T. (2004). *Physical Computing: Sensing and Controlling the Physical World With Computers*. Boston, MA: Course Technology Press.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Papert, S. (1993). *The children's Machine: Rethinking School in the Age of the Computer*. New York, NY: Basic books.
- Rabardel, P. (2003). From artefact to instrument. *Interact. Comput.* 15, 641–645. doi: 10.1016/S0953-5438(03)00056-0
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., .., and Kafai, Y. (2009). Scratch: programming for all. *Commun. ACM* 52, 60–67.
- Rizzo, A. (2015) "Les artefacts d'origine humaine," in *L'ergonomie orientée enfants*, eds F. Decortis (Paris: Presses Universitaires de France), 219–250.
- Rizzo, A., Burrelli, G., Montefoschi, F., Caporali, M., and Giorgi, R. (2016). Making IoT with UDOO. *IxD&A* 30, 95–112.
- Rizzo, A., Marti, P., Decortis, F., Moderini, C., and Rutgers, J. (2001). "The POGO story world," in *Handbook of Cognitive Task Design*, ed E. Hollnagel (London: Laurence Erlbaum), 577–602.
- Rizzo, A., Montefoschi, F., Ermini, S., and Burrelli, G. (2015). UDOO App Inventor: Introducing Novices to the Internet of Things. *Int. J. People Orient. Programm.* 4, 33–49. doi: 10.4018/IJPOP.2015010103
- Rizzo, A., Rubegni, E., Grönval, E., Caporali, M., and Alessandrini, A. (2009). The net in the park. *Knowl. Technol. Pol.* 22, 51–59. doi: 10.1007/s12130-009-9067-y
- Rizzo, A., and Rutgers, J. (2004). "The construction/deconstruction of ambient," in *CHI 2004 Workshop Lost in Ambient Intelligence?*, eds A. Nijholt, T. Rist, and K. Tuijnjenbreijer (Vienna).
- Rosen, W. (2010). *The Most Powerful Idea in the World: A Story of Steam, Industry, and Invention*. New York, NY: Random House.
- Stetsenko, A. (2008). From relational ontology to transformative activist stance on development and learning: expanding Vygotsky's (CHAT) project. *Cultural Stud. Sci. Edu.* 3, 471–491. doi: 10.1007/s11422-008-9111-3
- Streitz, N., and Nixon, P. (2005). The disappearing computer. *Commun. ACM* 48, 32–35.
- Szokolsky, A. (2006). "Object use in pretend play: symbolic or functional," in *Doing Things With Things: The Design and Use of Everyday Objects*, eds A. Costall, O. Dreier (Aldershot: Ashgate), 67–86.
- Tchounikine, P. (2017). Designing for appropriation: a theoretical account. *Hum. Comput. Interact.* 32, 155–195. doi: 10.1080/07370024.2016.1203263
- Tomasello, M. (2009). *The Cultural Origins of Human Cognition*. Cambridge, MA: Harvard University Press.
- Turkle, S., and Papert, S. (1990) Epistemological pluralism: styles and voices within the computer culture. *Signs* 16, 128–157.
- Valsiner, J. (1998). *The Guided Mind: A Sociogenetic Approach to Personality*. Cambridge, MA: Harvard University Press.
- Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Mental Process*. Cambridge, MA: Harvard University Press.
- Weber, S. (2004). *The success of open source, Vol. 368*. Cambridge, MA: Harvard University Press.
- Wolber, D., Abelson, H., and Friedman, M. (2015). Democratizing computing with app inventor. *GetMobile: Mob. Comput. Commun.* 18, 53–58. doi: 10.1145/2721914.2721935
- Zimmerman, J., Forlizzi, J., and Evenson, S. (2007). "Research through design as a method for interaction design research in HCI," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY: ACM), 493–502.

**Conflict of Interest Statement:** MC was employed by the company AIDILAB, and DC was employed by the company SECO.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2019 Rizzo, Caporali, Conti, Montefoschi, Burrelli and Sinopoli. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.