The Journal of Peer Production
New perspectives on the implications of peer production for social change
Journal of Peer Production Issue 13: OPEN
http://peerproduction.net — ISSN 2213-5316

OPEN SOURCE BEYOND SOFTWARE: RE-INVENT OPEN DESIGN ON THE COMMON'S GROUND

Kosmas Gavras

The definition of open source hardware (OSH) is focused on electronic or mechanical hardware but less is known about the OSH subset that is neither electronic nor mechanical. In general, the OSH theoretical framework is summarised as the imposition of open source software's (OSS) licensing scheme on conventional design workflow, omitting the fact that the license is not the only difference between open source and proprietary code. Also, the OSH definitions overlook the broader context related to hardware's material nature as illustrated by the socio-technological implications of digital fabrication. Based on a hybrid quantitative-qualitative analysis of three case studies from the less examined OSH's subset, we manage to: i) demonstrate the limited applicability of current OSH definition and describe the points that lack specificity, ii) propose an optimized and detailed definition based on the pre-existing framework, and iii) reveal structural inconsistencies of OSH theory and highlight emergent practices that may constitute a hopeful future answer. As more and more researchers realize the catalytic potential of the open source methodology, but at the same time refer to the non-automatic process to turn open source everything, it will be indispensable to design a thoughtful and motivational OSH theory.

Keywords: Open Design, Open Source Hardware, Digital fabrication, Computational Design, Digital Commons

By Kosmas Gavras

INTRODUCTION

Open source software (OSS) has become a dominant mode of production in a number of areas such as server software, operating systems and scripting languages (Lerner and Tirole, 2005; Chesbrough and Appleyard, 2007). Since the last decade, several studies have focused to a wider applicability (von Krogh and von Hippel, 2006; Nuvolari and Rullani, 2007) of the OSS organizational and production model (von Hippel and von Krogh, 2003; Osterloh and Rota, 2007). Benkler (2002, 2006) draws from OSS and early P2P sharing networks, to describe a third mode of production, the Commons based peer production (CBPP), extending beyond software to open content such as Wikipedia and Openstreetmap among others. Additionally, an equally important direction is the extension of the open source model to the world of tangible objects (Raasch et al., 2009; Balka et al., 2009a, 2009b; Shirky, 2005).

Initially, a significant number of free and OSS theorists have objected to, or at least have been sceptical of, the openness parallelization between bits and atoms (Stallman, 1999; Raymond, 1999a; Maurer and Scotchmer, 2006; Ackermann, 2009). However, in the course of time the open source hardware's (OSH's) potential was realized, as among other advantages, the OSH will represent the only possibility to run freely OSS in the near future (Stallman, 2015). In the mean time a lot of major OSH projects have emerged. Indicatively, some of the most prominent approaches are located in the fields of: Electronics (Arduino), Mechanical (Farmhack, Ateliers Paysan, Open Source Ecology), Mechatronics (RepRap, OpenBionics), Non electronic – nor mechanical (Wikihouse, Opendesk, Openstructures, Open Architecture Network, Hexayurt).

Von Hippel makes a parallelization between OSS and open design (OD) as the immaterial phase of OSH: *"Hardware is becoming much more like software up to the point you actually fabricate an object."* (Von Hippel in Thompson, 2008).

Despite most of the OSH's theoretical background was transcribed from OSS's principles and licenses, the main theoretical stake persists. What is the "source code" of OSH? The profound answer that design is the "source code" of hardware is not solid and specific enough to establish an analytical framework (Vardouli and Buechley, 2014; Fuller and Haque, 2008). The purpose of the research is to provide a critical insight to existing OSH and OD theoretical definitions and describe potential new dimensions based on the following starting points:

- Current OSH's theoretical framework is based to two cumulatively applied assumptions. Firstly, it is assumed that design for hardware is what source code for software. Secondly, it is deducted that by applying to design, the same license's degrees of freedom that have been applied to OSS, then design will be turned to open source design or simply OD. While license principles are exogenous properties of source code, and thus easily transferable to other fields, it may be more fruitful to examine how intrinsic OSS's properties could be tranfered to hardware's design.
- Even if design as well as source code are essentially immaterial, we should always consider that design is not an end in itself but an integral input in the design-construction continuum. In this way, the theoretical foundation of OSH and OD should extend beyond the mere replication of OSS's model in order to reflect the digital fabrication revolution and the emergence of makerspaces.
- Hardware is a broad and generic description for diverse groups of objects while software is a much more coherent concept. Therefore, the design of a microcontroller may not share the same principles with the design of a piece of furniture. Consequently, it is meaningful to evaluate whether current OSH theoretical framework is applicable towards non-examined subsets of hardware.

In order to describe the "source code" of OSH, we will analyze a set of case studies against existing OSH definitions and in relation with the above-mentioned points. Specifically, we will highlight inefficiencies or inconsistencies of existing OSH theory in the given context and suggest ways to overcome the recognized limitations. The article is structured as follows: In the next section we describe the existing literature context that operationalizes the research framework. The stimulated research questions as well as the research approach and methodology are developed in the third section. Successively, the fourth section contains the case study analysis. The fifth section discusses the resulting limitations and outlines an emergent OSH and OD direction based on the instrumental potential of the case study research. The final section provides a conclusion of research results along with directions for future elaboration.

LITERATURE REVIEW

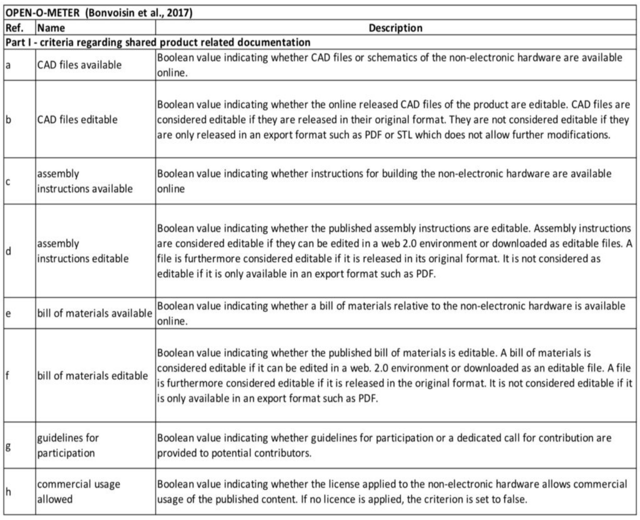
The multiple dimensions of OD

The term OD does not imply a single and cohesive meaning (De Mul, 2011; Vardouli and Buechley, 2014). In theory and practice it is often used interchangeably with different but frequently complementary meanings, such as the collaborative, cooperative or participatory design (Habraken, 1972; Kaspori, 2003; Salingaros et al., 2010; Manzini, 2015; Ratti and Claudel, 2015), the modular design (Fuller and Haque, 2008; Kostakis and Papachristou, 2014a), or the freely and globally shared design (Open Architecture Network, no date; Open Architecture License, 2016). Actually, all of those meanings represent major dimensions of an evolving OD definition in the framework of digital Commons and digital fabrication.

Analytically, the Design Global, Manufacture Local (DGML) model (Kostakis et al., 2015) highlights a new productive model with transformative socio-economic implications that is based on the convergence of globally shared OD Commons with desktop manufacturing. The voxel-based fabrication describes a modular design methodology along with an experimental manufacturing technique, which is based on the discourse between digital and analog materiality (Gershenfeld 2007, 2012; Gershenfeld et al., 2017; Kostakis and Papachristou, 2014a; Hiller and Lipson, 2009). At last, participatory design has a history that precedes the domination of the open source model. During the decade of 1970s the approaches of Habraken (1972), Friedman (1975) and Alexander et al. (1977) targeted to the user empowerment in the design process. Despite the aim for increasing user participation, there was always a common and clear separation line between what Manzini (2015) defines as expert and diffuse design. In the case of Habraken (1972) it was the distinction between the design of structure and infill that reflects the bipole of expert and user. Alexander et al. (1977) and Friedman (1975) constructed a pattern language and a computational design process respectively, as non-technical vocabularies for user participation. In general terms, the principle is common: the expert designs an almost closed system that the user interacts with. On the other hand, approaches that draw from the open source model (Kaspori, 2003; Salingaros et al., 2010) describe emphatically the demise of the expert design in the context of what Von Hippel (2017) coined as collaborative free innovation.

From OSS to OSH definition: the case of open-o-meter

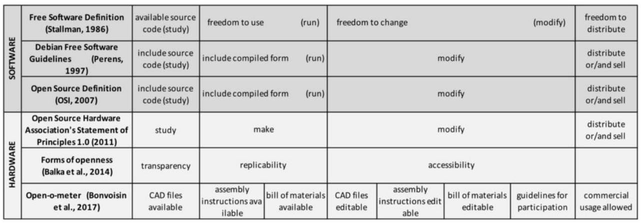
Despite the existence of wide literature on various partial aspects of OSH and OD, there are very few attempts to provide an integrated definition of openness in the sphere of hardware. In general, the most systematic and comprehensive quantitative method up to date regarding hardware's openness evaluation is the open-o-meter (OPEN!, 2018). Open-o-meter is a numeric scale, consisting of eight equivalent Boolean values which cumulatively express the total openness of OSH. Analytical documentation of the openness scale (Table 1) is presented by Bonvoisin et al. (2017). While the openness levels refer to the hardware, the evaluation criteria examine exclusively aspects of immaterial documentation or OD content. In this way open-o-meter is, beyond an evaluation tool, the theoretical foundation of OD as the open hardware's "source code".



OPEN-O-METER (Bonvoisin et al., 2017)		
Ref.	Name	Description
Part I - criteria regarding shared product related documentation		
a	CAD files available	Boolean value indicating whether CAD files or schematics of the non-electronic hardware are available online.
b	CAD files editable	Boolean value indicating whether the online released CAD files of the product are editable. CAD files are considered editable if they are released in their original format. They are not considered editable if they are only released in an export format such as PDF or STL which does not allow further modifications.
c	assembly instructions available	Boolean value indicating whether instructions for building the non-electronic hardware are available online
d	assembly instructions editable	Boolean value indicating whether the published assembly instructions are editable. Assembly instructions are considered editable if they can be edited in a web 2.0 environment or downloaded as editable files. A file is furthermore considered editable if it is released in its original format. It is not considered as editable if it is only available in an export format such as PDF.
e	bill of materials available	Boolean value indicating whether a bill of materials relative to the non-electronic hardware is available online.
f	bill of materials editable	Boolean value indicating whether the published bill of materials is editable. A bill of materials is considered editable if it can be edited in a web. 2.0 environment or downloaded as an editable file. A file is furthermore considered editable if it is released in the original format. It is not considered editable if it is only available in an export format such as PDF.
g	guidelines for participation	Boolean value indicating whether guidelines for participation or a dedicated call for contribution are provided to potential contributors.
h	commercial usage allowed	Boolean value indicating whether the license applied to the non-electronic hardware allows commercial usage of the published content. If no licence is applied, the criterion is set to false.

Table 1. Open-o-meter. OSH product characterization criteria (Bonvoisin et al., 2017).

In order to develop a critical stance on existing theoretical framework, it is important to trace open-o-meter history back (Table 2). Open-o-meter is a further specification of theoretical freedom principles (*study*, *make*, *modify*, *distribute*), and forms (*transparency*, *replicability*, *accessibility*) of openness, referenced in Open Source Hardware Association's Statement of Principles 1.0 (2011) and Balka et al. (2010, 2014) respectively. Successively, Open Source Hardware Association's principles are based to the Open Source Definition (Open Source Initiative [OSI], 2007).



SOFTWARE	**Free Software Definition (Stallman, 1986)**	available source code (study)	freedom to use (run)		freedom to change (modify)				freedom to distribute
	Debian Free Software Guidelines (Perens, 1997)	include source code (study)	include compiled form (run)		modify				distribute or/and sell
	Open Source Definition (OSI, 2007)	include source code (study)	include compiled form (run)		modify				distribute or/and sell
HARDWARE	**Open Source Hardware Association's Statement of Principles 1.0 (2011)**	study	make		modify				distribute or/and sell
	Forms of openness (Balka et al., 2014)	transparency	replicability		accessibility				
	Open-o-meter (Bonvoisin et al., 2017)	CAD files available	assembly instructions available	bill of materials available	CAD files editable	assembly instructions editable	bill of materials editable	guidelines for participation	commercial usage allowed

Table 2. OSH theoretical framework as an evolution from OSS definitions.

(abbreviations: Open Source Initiative [OSI])

Consequently, open-o-meter framework leans towards OSS rather than free and open source software's (FOSS's) political standpoint. As a proof of this tendency, open-o-meter defines commercial usability of design documentation and physical product, as one of the basic parameters of openness. In fact, there is a profound contradiction between commercial usability and the notion of openness, which is more evident at the immaterial side of the design-construction continuum. How can design documentation be commercially exploitable and freely editable and available concurrently? Some of the most known examples of FOSS (GNU/Linux, Apache server, Mozzila Firefox) thrive in the absence of commercial appropriation. Additionally, it is proved that total openness, as expressed by the sum of transparency, replicability and accessibility excluding commercial usability criterion, increases the level of contribution (Balka et al., 2014) even if some forms of openness play a greater role than others.

At this point, it should be stated that from an ideological point of view the author stands with the position of the Free Software Foundation (FSF), rather than that of OSI, regarding the social imperative of software (FSF, 2016). Since at the technical level, free software source code qualifies as open source code (Stallman, 2016) and vice versa, it could be supported that open source code is just one -the technical one- of the constituents of FOSS definition. In this sense, hereby the term open source code, OSS or open source design does not necessarily point to the OSI definition, unless otherwise explicitly stated. Additionally, taking into consideration the framework of this paper, OD is a commonly acceptable term, comparing with the limited usage of the term free design.

To get back on track, previous research works employing a quantitative assessment of hardware's openness have covered the field of electronic products (Balka et al., 2010, 2014), as well as the field of non-electronic but mechanical hardware (Bonvoisin et al., 2017). To the author's knowledge, there is no published quantitative study evaluating openness of the OSH subset that is neither electronic nor mechanical. This absence is explained by the initial lack of OSH practices being neither electronic nor mechanical. Due to the proximity of electronic hardware with OSS movement, the earlier OSH initiatives were concerning electronic devices (Gibb, 2014) or later mechanical (Vallance et al., 2001). It is characteristic that TAPR (2007) – one of the first open hardware licenses – had considered only electrical or mechanical artifacts as potentially open hardware.

To synopsize, the existing theoretical background of OSH is transcribed from OSS definitions and licensing schemes which are not structural properties of the source code. Even if the license profoundly affects to some extent the subsequent form and development of the code, it is not an inherent property of the code itself.

OSS's structural properties: the parameter of modularity

If the licensing schemes represent exogenous properties of open source code, which are the structural properties? According to Conway (1968:31), *"organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations"*. In other words, the structural properties of code or design mirror the structure of the organization that developed them. MacCormack et al. (2006, 2012) proved empirically the "mirroring hypothesis" in the field of software industry, highlighting source code's modularity. Modularity is defined as the decomposition of a product or design to independent modules. Analytically, the more distributed, informal and open organizations tend to produce the more modular codebases. On the contrary, the more centralized, hierarchical and closed organizations produce more monolithic software. As a summarizing outcome, OSS is structurally different from proprietary software in terms of modularity (MacCormack et al., 2006). The difference in modularity between the opposite ends is surprisingly large, reaching a factor of eight between software of similar size and function (MacCormack et al., 2012).

Therefore, there is a causal bidirectional mechanism between, on the one hand, the dispersed geographical distribution of open source communities, the open contribution spirit and the informal authority, and on the other hand, the emergent modular code's structure (MacCormack et al., 2012). The causal linkage is bidirectional as *ex ante* modular code's structure is prerequisite for attracting and facilitating a diverse and distributed community of developers which in turn enhance the overall modular effect in the course of time (Baldwin and Clark, 2006; MacCormack et al., 2012). Preliminary design choices are proved critical regarding the potential successful evolution of OSS projects. Baldwin and Clark (2002) explicated further the importance of modularity as a structural constituent of the open source model:

- "*Modularity makes complexity manageable*" through: (i) the decomposition of complex tasks to simpler, and (ii) the parallel processing of complex tasks in the direction of selecting the optimal solution;
- "*Modularity enables parallel work*" through the division of a monolithic system to independent modules or what Benkler (2006) named granularity; and
- "*Modularity is tolerant of uncertainty*". Code modules embed option values that render an overall code structure which is susceptible to change in potentially unforeseen ways. Modular structures can be modified and evolve over time at low cost comparing with monolithic data structures.

To summarize, modularity is a fundamental property with causal role at the open source model that can not be omitted when transferring the open source methodology to other fields, including OSH. As a conclusion, the freedoms to *study* and *modify* depend to a great extent in code's modularity.

Digital fabrication from a design perspective

While modularity is an important parameter of an evolving OD definition, we realize that it is meaningless to examine the OSH solely as an OSS offspring, but only in a wider perspective including the technological advancements related with hardware's material nature. For instance, the digital fabrication revolution makes feasible the distributed – even personal – manufacturing of a genealogy of customized objects, virtually at the same unit cost as if producing identical copies (Rifkin, 2014). According to Carson (2010), manufacturing is fading as a centralized, closed and capital-intensive activity. As a proof of concept, the digital fabrication revolution and hackerculture have already led to the emergence of a rapidly increasing global network of local makerspaces (Niaros et al., 2017), answering to the user's need for custom products (Von Hippel, 2005) among others. But manufacturing (being the second part in the design-construction continuum) depends to a great extent on the design input.

What kind of design will form the necessary data input to a localized, on-demand, and customized fabrication? A monolithic, closed, centrally produced design aiming to cover the greater homogeneous market segment of user needs, or a modular, open, globally produced design able to be modified at low cost to fit the local needs? Lakhani describes a user-powered design innovation, proactively forcing companies out of the product design space (2007). Consequently, the real potential of digital fabrication will be actualized more effectively in combination with the peer production, in proposing an alternative, holistic emergent productive model (Kostakis et al., 2013, 2015), rather than in re-organizing existing corporate infrastructure (Hermann et al., 2016) according to design principles that are primarily native to open source culture.

To sum up, modularity, beyond being a structural property of the open source mode of development, it is also related with the freedom to *make* in the socio-technological framework of digital fabrication.

RESEARCH DESIGN

Research Questions

Current OSH theoretical framework have tested and been tested against electronic products (Balka et al., 2010, 2014) and mechanical devices (Bonvoisin et al., 2017), but to the author's knowledge there is not any published study concerning physical products being neither electronic nor mechanical. There is preliminary evidence that current OSH theory can not migrate free ride from one kind of hardware to another. Both electronic and mechanical devices usually include stardardized, ready-made and commercially available components to a great extent. Off-the-shelf components can limit the design and customization potential which are inherent to the concept of user-developed hardware and digital fabrication. On the other hand, non electronic or mechanical hardware, ranging from a piece of furniture to a house, is characterized by higher degree of customization. Additionally, electronic circuits, machines and other hardware do not share a common design language. Electronic design is comprised of 2d layouts; mechanical design requires a kinetic 4d approach; whereas other hardware can be thoroughly described by a 3d model.

Bonvoisin et al. (2017) acknowledge that the broader applicability of open-o-meter is an open question. Beyond breadth, according to Bonvoisin et al. (2017) further research is required to provide an in depth qualitative specification of each openness criterion. The authors of open-o-meter followed a simplistic binary evaluation for each criterion that was practically imposed by the substantial sample size required for statistical analysis. Based on the literature review and the scope of the research, the following questions will be addressed (Figure 1):

RQ1a – theory's breadth: Is open-o-meter a valid quantitative tool and definition in the framework of neither electronic nor mechanical hardware?

RQ1b – theory's depth: If open-o-meter is not generically applicable, then which criteria fail to capture actual openness levels of the examined subset? How can these criteria be improved by further qualitative specification?

RQ2 – theory's context: How is modularity, which has not been expressed directly in open-o-meter, reflected in OSH practice? How can we describe a positive feedback loop between OSH's practice and theory, based on a new reading of openness in the context of digital fabrication and OSS's structure?

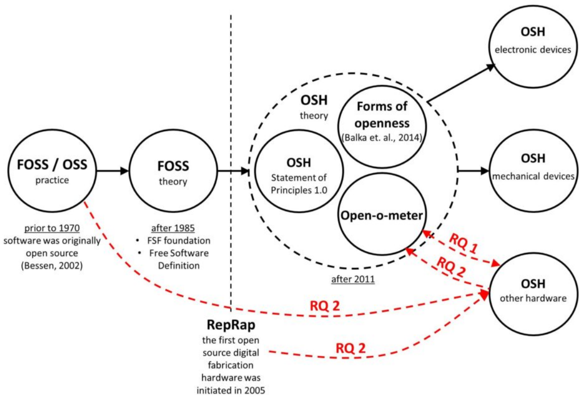


Figure 1. Research questions placement in the conceptual timeline diagram framed by the literature review.

(image processing: author, image source:author, abbreviations: Free and Open Source Software [FOSS], Open Source Software [OSS], Free Software

Foundation [FSF], Open Source Hardware [OSH])

Research Approach and Methodology

In order to address the research questions mentioned above, we employed an empirical quantitative and qualitative case study research.

Analytically, the hybrid quantitative-qualitative method refers to the first question (RQ1a) regarding open-o-meter theory's breadth, while following research questions will be tackled with exclusively qualitative approaches. Theory's breadth will be evaluated by the assessment of how quantified openness levels of the examined subset, actually correspond in practice to the more abstract and qualitatively approached freedoms of *study*, *make* and *modify* (Table 2). The quantitative method of openess' assessment is the open-o-meter (Table 1), as documented by Bonvoisin et al. (2017). Despite the fact that the original openness scale utilizes only binary value rating, the current research adopts the use of midpoint assessment, as a general measure for those cases where a variable pattern is discerned. Furthermore, the open-o-meter method followed in this article refers solely to part 1 of the original method, as the external and secondary criteria consisting the part 2 table, are mostly constrained by the case study selection process.

Empirical data required for the quantitative assessment and the qualitative inquiry are acquired from web-based documentation and databases. Indicatively, sources that have been used depending on each occasion are the following: websites of the product development communities, archived webpages, web-based repositories, github or other party development statistics, processing of CAD or other files, participation in developers forums and communication networks, and member initiation questionnaires among others. It is important to stress that data input to the quantitative assessment, is based on the "macroscopically" observed presense or absence (or variable pattern) of certain documentation as described by open-o-meter methodology, avoiding any qualitative characterization of the usefulness of each provided document. Data analysis has followed a combination of categorical aggregation and direct interpretation strategies, as described by Stake (1995).

The choice of case study research is based on the need for detailed examination of particular cases and their context. In this way, the shortlisted cases are intentionally not representative of the whole range of the OSH scape, but they are examined as collective – instrumental case studies (Stake, 1995), able to offer a new understanding of OD in the OSH framework. As the selection of cases is of critical importance in any type of case study research (Stake, 2003), the formal selection process and criteria actually reflect the author's strategy in the given context. Moreover, it is important to note that the aim of the collective quantitative case study research is not to concentrate on a comparative analysis per se, but to illustrate potentially different viewpoints to the research questions.

The selection[1] of case studies is based on an already existing online directory of OSH projects, maintained by Wikipedia (2018a) which is a prominent example of CBPP beyond software. Previous studies have employed either conventional internet search, using a snowball effect approach that was prone to be author biased or online directories built-up and biased by OSH projects' "owners". On the other hand, the selection of a Wikipedia directory ensures a minimum consensus about what is considered by the greater community as open source. The Wikipedia online directory is a dynamic, bottom-up archive of the OSH projects, the launch of which mostly coincides with the recent history of the OSH development[2]. The following selection criteria were applied to the initial list, in order to narrow the corpus of research, according to the research questions:

- The hardware is neither electronic nor mechanical. As the first two fields are already evaluated regarding openness by Balka et al. (2010, 2014) and Bonvoisin et al. (2017) respectively, it is crucial to define the complementary subset in order to test the broadness of the already set openness criteria. Moreover, the examined subset of the OSH is dictated by the background of the author in architectural design allowing an in-depth examination of qualitative properties of design process.

- Inactive communities are excluded. Evaluating inactive communities regarding openness is a task of limited value. Active communitites are considered those, who have commited a new structure upload or even a minor edit in already uploaded documentation, in the 9 months before the case selection took place. This criterion aims to exclude those communities that practically cease activities after producing the first or a few functional prototypes.
- Premature communities and projects are also excluded. An arbitrarily set time limit of 4 years from startup is considered as a reasonable time interval that most OSH communities should have acquired a certain level of maturity, if they are to do so. Another point of maturity is the production of at least one functional prototype product. Previous attempts (Bonvoisin et al, 2017) have taken into consideration immature communities among others, as shown by the fact that nearly 50% of the cases do not provide editable CAD files and almost 25% do not even publish CAD files. These results indicate either immature communities that could not be characterized as open source yet, or products that do not include any custom parts. In any case, the scope of this paper is not to study the evolution of openness in the course of development of OSH projects, but to evaluate the “source code” in relation to the discussed framework. At this point, we acknowledge the rare but possible limitation of omitting an early but open and well developed project.
- The hardware should feature a minimum level of complexity. An assumption is made that adequately complex products are considered those comprised of at least 5 parts. Web-based 3D printing repositories of nearly compact gimmicks, such as the Thingiverse, are usually characterized by an individual design approach. The rationale behind complexity constraint is that more complex objects force and require more collective product development. Peer production in its essence is a synergistic rather than an individual modus operandi.

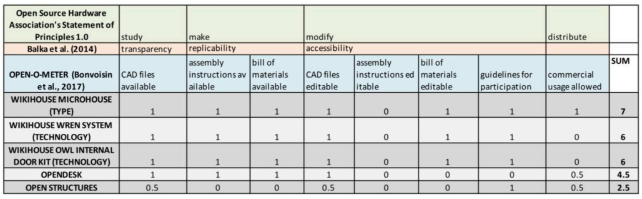
The OSH communities that satisfy the imposed criteria are: the Wikihouse (2018a), the Opendesk (2018) and the Openstructures (2018a). As denoted by their names, Wikihouse and Opendesk produce house and furniture designs respectively. Openstructures’ range starts from furniture and household equipment, ending potentially to houses or even bigger structures.

ASSESSMENT OF PRACTICES

Opendesk

Opendesk is a for-profit company, supporting an online platform that connects furniture designers with customers and local makers all over the world. It was founded in 2014, and part of the founders is part of the Wikihouse creators. Opendesk practice largely depends on the DGML model, as it is by definition a global platform for local making. Moreover, it is based completely on CNC manufacturing, as all designs are assembled from flat cut wooden profiles. Most of the furniture overpasses the assigned minimum complexity limit but features significantly less parts than other OSH projects, comprising from a few elements to some dozens of elements. Profit making comes from charging 30% of the manufacturing cost as a transaction fee every time a customer orders a piece of furniture. Apart from the transaction fee, there is a provision for a design fee which is calculated at 8% of the manufacturing cost. In any case, all of the furniture designs are freely available for personal fabrication and usage.

Opendesk project collects 4.5 out of 8 points in the openness scale (Table 3). Assembly instructions and bill of materials (BOM) are provided in non-editable formats, whereas contribution guide is a rather vague and closed process. Furthermore, licensing does not follow a uniform pattern, featuring a variety from copyleft to licenses allowing commercial appropriation. As the object of design is considerably smaller comparing to other OSH projects, the shortage of editable supporting documentation (assembly guide, BOM) is rendered less important than the absence of an open contribution process. The top-down enforced modus operandi of Opendesk is based on a standard and fixed relation between designer(s) and object of design, not formally allowing any collective optimization process which usually is very common to every CBPP and OSS project. Actually, modifications are theoretically feasible but for personal use only, as there is not any open platform or any even informal process for forking and then merging back into master branch. To conclude, Opendesk's one way workflow is almost identical to the conventional non-participatory design process, being characterized by a small scale closed design team or individual designer, retaining a practical ownership (above formal licensing) over artifact by indirectly controlling the potential modifications. Apparently, this is related to the nature of the for-profit organization and it seriously decreases the project's overall openness. It seems not rational to rate open contribution potential equally with the existence of editable BOM or other secondary documentation. Is it open a project that just freely reveals the "source code" of the hardware but restricts users and other designers from contributing?



Open Source Hardware Association's Statement of Principles 1.0	study	make		modify				distribute	
Balka et al. (2014)	transparency	replicability		accessibility					
OPEN-O-METER (Bonvoisin et al., 2017)	CAD files available	assembly instructions av ailable	bill of materials available	CAD files editable	assembly instructions ed itable	bill of materials editable	guidelines for participation	commercial usage allowed	SUM
WIKIHOUSE MICROHOUSE (TYPE)	1	1	1	1	0	1	1	1	7
WIKIHOUSE WREN SYSTEM (TECHNOLOGY)	1	1	1	1	0	1	1	0	6
WIKIHOUSE OWL INTERNAL DOOR KIT (TECHNOLOGY)	1	1	1	1	0	1	1	0	6
OPENDESK	1	1	1	1	0	0	0	0.5	4.5
OPEN STRUCTURES	0.5	0	0	0.5	0	0	1	0.5	2.5

Table 3. Openness evaluation

It could be supported that despite drawbacks, Opendesk still offers what is considered by open-o-meter the "source code" (editable CAD files and non-editable BOM and assembly instructions) which seems almost enough to *study*, *make* and *modify* the design individually. In contrast, a detailed examination of the shared "source code" will reveal that it is indirectly but efficiently restricting users and designers from altering the shared furniture designs even for personal use. The criterion regarding editable CAD files (Table 1) in practice seems a rather generic and abstract scheme. The limitation of the evaluation criterion does not lie only in the quality of CAD files, as suggested by Bonvoisin et al. (2017), but to the nature of the CAD files. To further the argument, it is required to make a clear distinction between design and fabrication CAD files. The actual shared "source code" of Opendesk is flat-cut drawings, intended for CNC manufacturing, which are provided in editable format. Fabrication files, even in native file format, are still a derivative of actual design files. It is self-evident that cut-out drawings (Figure 2) by themselves can not help neither to *study* nor to *modify* the design, but only to *make* it. There is no doubt that if personal resources were limitless, an expert designer could use the fabrication outline drawings and the assembly instructions to reverse-engineer the design files, but this can not be the case. Another important note, beyond the analytic distinction between design and fabrication CAD files, is that the cognitive process producing the latter from the former is of utmost importance regarding the OSH development, and definitely subject of openness evaluation. To conclude, community is not only restricted from providing optimization feedback, but it is generally constrained from *studying* and *modifying*.

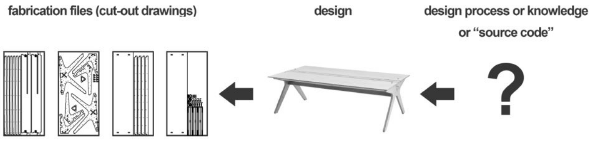

fabrication files (cut-out drawings)
design
design process or knowledge
or "source code"
?

Figure 2. Opendesk shared CAD files in relation to the design "source code".

(image processing: author, image source: https://www.opendesk.cc/, last visit: 01/07/2018)

Even the freedom to *make* should not be considered as a totally independent principle from *studying* and *modifying* an object. Specifically, the three principles are interconnected to a great extent in a synergistic manner as the full potential of *making* is achieved only if you are first of all able to *study* and *modify*. Otherwise, the DGML concept will be deducted to a simple model of distributed "mass production", negatively affecting the real potential of OD and digital fabrication. It is less meaningful and underutilized to use a CNC or a 3d printer just as a medium to manufacture locally identical copies of what is used to be massively designed and produced.

Openstructures

Openstructures began as a student project at Institute without Boundaries in 2006. In September of 2009, Thomas Lommee – designer and former member of the student team – organized an exhibition, showcasing the concept and some initial prototypes, as well as an open call for the collaborative development of the project. Openstructures is defined as an open modular system for building hardware – in contrast with current closed (proprietary) modular systems – which is inspired by the modularity of OSS (Openstructures, 2018b). The centerpiece of Openstructures system is OS grid, an openly shared geometrical grid, built up from the supersposition of a diagonal, polar and regular grid (Figure 3). The OS grid serves as the common compatibility base to design parts and their interfaces which then can be assembled in different combinations to form anything from furniture to houses. Another important design principle is the vertical organization of artifacts in *parts*, *components* and *structures* depending on the functional autonomy and relative position of each element in a posssible greater scheme. Specifically, *components* are made of *parts* and respectively *structures* are made of *components* in a recursive manner.

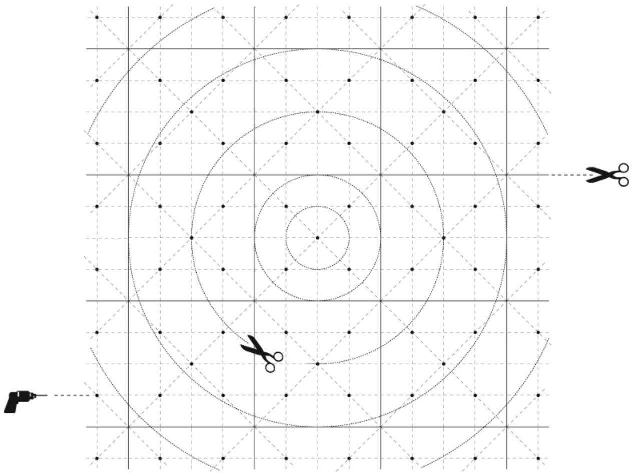


Figure 3. Openstructures modular grid made out of 4X4 cm square bold-lined cells.

(source: http://beta.openstructures.net/block_images/0000/0242/grid4.jpg?1268343038, last visit: 01/07/2018)

"The open modular system has the potential to:

- *generate flexible and dynamic puzzle structures rather than uniform modular entities*
- *introduce variety within modularity*
- *stimulate re-use cycles of various parts and components*
- *enable collaborative (and thus exponential) innovation within hardware construction."* (Openstructures, 2018b)

In contrast with the declared intentions, Openstructures is the less open project (Table 3) in the open-o-meter scale as the contribution guide is the only steadily provided documentation. Basic documentation, as CAD files whether editable or not, is provided occasionally. Supporting documentation, as BOM or assembly guide, is totally absent and licensing does not follow a specific pattern. From a first point of view, the freedoms to *study*, *make* and *modify* are only occasionally exercised.

Despite the lack of consistently open documentation, according to project's initiators, community is encouraged to participate either one or more of the following ways:

- *"Designing parts, components or structures according to the OpenStructures grid*
- *Trading designs online.*
- *Exchanging your experiences and ideas with others in order to improve the system".* (Openstructures, 2018b)

How can it be possible to contribute to an open invitation when there is not steadily provided any hardware's "source code"? A related important observation, regarding Openstructures' online platform, is that technically each contributor is permitted to *modify* only those elements that he has authored. Consequently, even in these cases that the design files are available and editable, the development is characterized by a conventional, individual and authoritative design workflow. On the other hand, in the CBPP framework, the collective optimization of existing knowledge modules is equally or even more important than making new modules. Digital Commons – with OSS being an eminent example – are structured more like work in progress (Raymond, 1999b), continuously optimizing according to new inputs, rather than as finished products. As a result, an OD platform without hierachical control of what – new design – is published, while being in the correct direction, it is not a per se sufficient condition to trigger the wealth of open source collaborative practices.

Beyond the actual technical and collaborative limitations of the platform, the Openstructures' most fundamental characteristic is the modular design development. Even if it is not explicitly mentioned, it seems that modular design in combination with the imposed classification to *parts*, *components* and *structures*, was indirectly a method of asynchronous collaboration and optimization. For instance, the process of re-using a *part* (module) from another designer to a new assembly turns the new assembly into a collaboratively produced object. In fact, a closer look to the repository reveals that most *structures* are composed of *parts* and/or *components* designed by the same author, practically eliminating the potentially emergent collaborative effect. Why does the design modularity not lead to re-use cycles in a collaborative innovation environment as declared?

Apart from the collaborative limitations imposed by the online platform and the cases of absolute documentation scarcity, the reason lies in the essence of modularity itself. What is different, when comparing the OS grid system with the voxel-based design described by 3d printing pioneers (Hiller and Lipson, 2009; Gershenfeld, 2007, 2012), is the interface design. The potentially innumerable joints that can be designed, based on the OS grid, render any interoperability between modules unfeasible. On top of that, the hierarchical division of elements (*parts*, *components*, *structures*) from an individual design perspective leads to an a posteriori modularization of a structure to *parts* and *components*, in an analytic rather than a synthetic manner, with limited re-usability.

In regard to the freedom to *make*, even if design files are occasionally shared, the fabrication files are never shared. This fact renders almost impossible for potential users to fabricate complex structures with little effort, efficiently precluding them from a feedback process. On the other hand, beta-testers and generally users are extremely valuable for digital Commons comparing even to developers base (Shirky, 2008). Raymond (1999b) strengthens the argument:

> *"# 6 Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging.*
>
> *The power of this effect is easy to underestimate. In fact, pretty well all of us in the*

> *open-source world drastically underestimated how well it would scale up with number of users and against system complexity, until Linus Torvalds showed us differently."*

Fabrication files' contribution is not limited to the freedom of *making*. Moreover, fabrication drawings are not derived from design files in a deterministic way as each specific design can be materialised in more than one way. To sum up, as the production of OSH does not culminate in the design documentation, design and fabrication files as well as the intermediary process are subject to *study* and *modify* beyond merely *making*.

Wikihouse

Wikihouse was initiated by a non-profit foundation in 2011 (Wikipedia, 2018b), claimed to be "*an open source project that re-invents the way we make homes*" (Wikihouse, 2018b), based on distributed digital manufacturing. Since then, architects, builders and users had constructed pilot wikihouses all over the world, inspired from the basic Wikihouse prototype called *microhouse*. *Microhouse* ensures by definition an advanced level of complexity far from the imposed cut-off limit. According to its declaration, Wikihouse draws a clear parallelization between OSS and architectural design as "*digital design allows every home to be designed as code; instantly customized to its site and user*" (2018a). Moreover, it is self-defined as a collaborative project: "*by everyone for everyone*" (Wikihouse, 2018a), following the DGML principle: "*share global, manufacture local*" (Wikihouse, 2018b).

Wikihouse's design knowledge is organized from smaller to larger elements in: *tools*, *technologies* and *types*. *Types* are ready-made building designs, hosting *technologies* as add-on systems which are assembled using *tools*. *Microhouse* is a one bed house design (Figure 4), currently being the only shared *type*. Available *technologies* include *WREN* chassis system and *OWL* internal door kit. The latter along with *tools* (mallet and a step unit) are components of limited interest and we will not analyze them further. On the other hand, *WREN* is an integral system for developing and sharing new Wikihouse *types* or modifying existing ones. Its main purpose is to automate fabrication analysis, subdivision and indexing required for the construction of 3-dimensional structural frame out of flat cut panels milled in CNC machines (Figure 5). Taking into consideration the distinction between design and fabrication CAD files as highlighted in the previous cases, *WREN* could be characterized as the intermediary process going from design to fabrication files. Openness levels will be assessed separately for *type* and *technologies* as they do not share the same documentation. Digital repositories of *type* and *technologies* are maintained at Github[3].

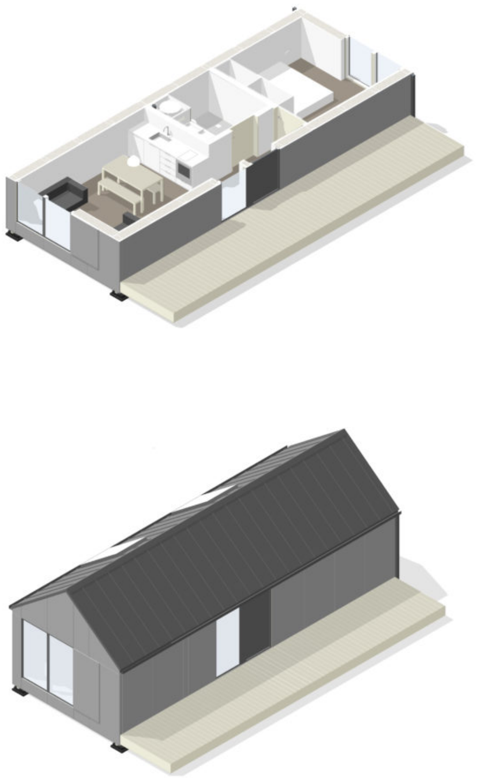


Figure 4. Microhouse type

(source:

https://github.com/wikihouseproject/Microhouse/blob/master/microhouse_0.5_isoFull.jpg, https://github.com/wikihouseproject/Microhouse/blob/master/microhouse_0.5_iso.jpg, last visit: 01/07/2018)

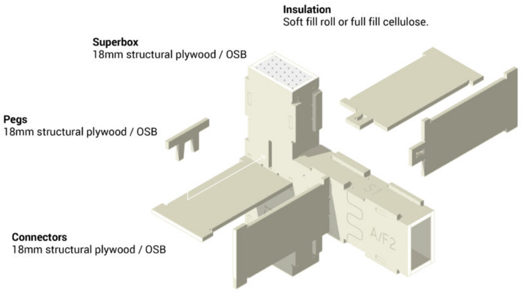

Insulation
Soft fill roll or full fill cellulose.
Superbox
18mm structural plywood / OSB
Pegs
18mm structural plywood / OSB
Connectors
18mm structural plywood / OSB
A/F2

Figure 5. WREN technology (source: https://github.com/wikihouseproject/Wren/raw/master/Images/Connectors-01.png, last visit: 01/07/2018)

The *Microhouse type* is evaluated as the most advanced project in the openness scale (Table 3). According to open-o-meter *microhouse* scores 7/8 points, the only drawback being the lack of assembly instructions in editable format which seems a minor deficiency comparing to the other criteria. In other words, if open-o-meter was a thorough and reliable method for the assessment of openness, then Wikihouse should be an almost perfectly open instance in-line with the principles of OSS development. Contrary to the expectations, *microhouse* development has considerable structural differences from the well-known OSS examples. Despite the fulfilment of participation guidelines criterion, the development of *microhouse type* remains largely at the control of a closed team mostly coinciding with Wikihouse founders. The *microhouse* repository at github[4] has 42 commits from only 3 unique contributors. The initial commit was made on the 25th of August 2016 and thereafter most of the commits refer to supplementary documentation rather than design files. In other words, the "source code" was uploaded to repository when the project was mature and most of the core files were already prepared by the founders. Since then, only minor edits have occurred. The above-mentioned characteristics are very common to the early stages of OSS development, before migrating to an open distributed and global mode of production. On the other hand, Wikihouse is a seven years old effort that could not be considered as an immature initiative. Consequently, a published contribution guide along with editable CAD files, BOM and assembly manual may not be enough to promote *accessibility* and the freedom to collaboratively *modify* an existing design.

The role of modularity in "*voluntary large scale parallel processing*" (Weber, 2004: 88) of a project is already described in the literature review as one dimension of its influence among others (Baldwin and Clark, 2002). If CAD files with the supporting documentation constitute the "source code" of OSH, then the knowledge contained in CAD files should be organized and built in a modularized way. GNU/Linux, Wikipedia and other known Commons could not have been developed to that magnitude, if they were not synthesized as an assemblage (DeLanda, 2006) of interconnected and semi-independent modules. On the contrary, CAD platforms are shaped to fit a much different mode of production, ranging from the sole designer to medium-sized dispersed but closed interdisciplinary teams. The first mode usually corresponds to an unstructured aggregation of 2-dimensional geometric data, while the second to a hierarchical relational model made out of preset 3-dimensional entities, which is known as *Building Information Modelling*. Whilst the uploaded design files of *microhouse* belong to the former data organizational structure, both are incompatible with an open source modular mode of development.

Beyond the collaborative role of modularity, another dimension is its tolerance of uncertainty. The case of Wikihouse, more than the other cases, highlights the need for adjustable design according to local or individual needs, faced in a globalized DGML methodology. It is oversimplistic to consider as realistic case the widespread application of OSH standard housing *types* all over the world, regardless how the number of *types*. Consequently, it is required to incorporate differentiation parameters in *type* design that will make easier the engagement of the community and the application of each *type* to different site, climate and materials among others. At this framework, it seems less arduous to design parametric housing *types*, than creating a new solid *type* for almost each one local instance. In the former case is required to design the change instead of the final object's properties. The global gestion leads on to a relational or procedural object definition rather than a categorical finished and unique artifact. It could not be overlooked that Wikipedia entries as well as OSS modules are defined in relation to other entries or modules resembling a distributed flexible network organization. Moreover, the design and execution of any OSS is procedural in its essence or in other words algorithmic. To summarize *microhouse* despite the nearly full rating in the openness scale, it is qualitatively less collaborative and customizable than expected in the placed framework, which can be attributed to the absence of design modularity.

WREN and *OWL technologies* score 6/8 (Table 3), as besides non-editable assembly instructions they lack license allowing commercial usage. Despite fundamentally disagreeing with rating commercial usability as an openness parameter, it is noted that *microhouse* is released under CC BY-SA 3.0 while *WREN* system under the copyleft MPL 2.0. But, *WREN* and *microhouse* share a more fundamental difference. *WREN* system is not designed in conventional CAD platforms, but it is created in a visual programming environment (Figure 6) which functions as an extension of a proprietary design platform. Visual programming as well as classic programming (scripting) is in-line with a relational or procedural definition of the design, being potentially more modular than conventional CAD if used as core of the design methodology. Consequently, CAD files are not necessarily the only design files, which was an axiomatic admission for open-o-meter evaluation framework.

Furthermore, it is interesting that both *type* and *technologies* have been accomplished and released under a proprietary and commercial design platform. It is generally accepted that OSS design platforms are not so common, and they have not approached yet the functionality of commercial platforms, possibly with the exception of Blender (Velkova, 2016) which is addressed more to visualization rather than design industry. Nonetheless, this is a highly restrictive fact regarding contribution potential, as it automatically excludes a great amount of the community. While design platform's openness is not part of open-o-meter evaluation, it certainly and severely affects the ability of a project to be characterized as open. To reformulate open-o-meter criteria regarding CAD files, they need not only to be in native file format but in open file format as well.

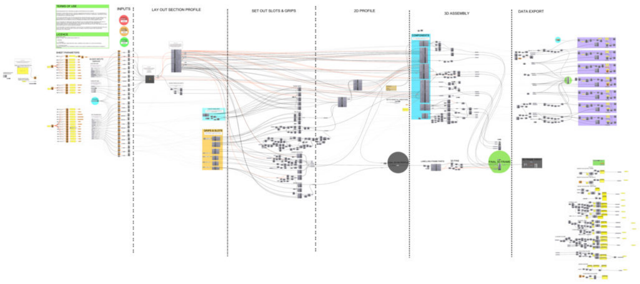


Figure 6. WREN system's source code in visual programming environment

(source: https://github.com/wikihouseproject/Wren/blob/master/WikiHouse_WREN_(v4.3).gh, last visit: 01/07/2018)

DISCUSSION

OSH theory's breadth

Open-o-meter has a double function in the field of the OSH. On the one hand, it is a tool for the quantitative assessment of practices' openness. On the other hand, it serves as an OSH programmatic definition, a kind of guide for startup open initiatives. While more and more researchers highlight the potential of the Common – as extensively demarcated by Vercellone et al. (2015) – to become a dominant (Rifkin, 2014) socio-economic paradigm, at the same time Shirky (2005) recognizes that open source model can not migrate free ride to any other domain. Consequently, the systematization of OSH theoretical specifications is crucial for the successful diffusion of open source principles in the material world.

The openness' analysis of the three case studies highlights major and minor limitations of the OSH theoretical framework as specified by open-o-meter. Minor limitations include over-simplification in the description of some of the evaluation criteria. Minor limitations can be painlessly overcomed by further in-depth specification and development of the already set criteria, without altering the quantitative nature of the open-o-meter. Major limitations come out from feed back loops between OSH and its constituent origins, revealing the absence or misconception of structural properties that characterize the open source model. The realization of the major limitations opens up a broader and more structural discussion regarding the social dimension, the transferability and the essence of OD as the hardware's "source code". Minor limitations are defined below and adressed in the next sub-section. The last sub-section discusses the major limitations of current theoretical framework and highlights an emerging OD direction based on the examined case studies and their context.

In general, among the examined case studies Opendesk and Wikihouse present the greater divergence between the quantitative assessment of openness and the respective qualitative evaluation of the freedoms to *study*, *make*, *modify* and *distribute*. It is noteworthy that Opendesk is rated 4.5/8 in the openness scale whilst the only actually provided freedom is to *make* identical copies of the shared design, practically without being able neither to *study* nor to *modify* the design before. On the contrary, Wikihouse is an almost perfectly (7/8) open example according to open-o-meter. Theoretically all of the described freedoms (*study*, *make*, *modify*, *distribute*) are unobstructed. But the closer look to collaboration platform's statistics revealed in practice poor collaborative development and limited number of contributors despite the project's maturity. In other words, the most successfully open case study is characterized more by a conventional design workflow and organization, rather than by a really open source mode of distributed development. To summarize, open-o-meter is proved to be a tool of limited applicability in the field of neither electronic nor mechanical hardware. In the direction of optimizing open-o-meter efficiency, some criteria, that profoundly need further specification, are the following:

The criteria concerning the availability and editability of CAD files are over-simplistic and incomprehensive. As a consequence, a strong tendency to produce positively biased results is observed in the examined subset. While the quality of the documentation is not easily quantifiable, the research showcased steps that can easily specify further these criteria.

The criterion of participation guidelines is a generic reference to open contribution not corresponding effectively to the wealth of the open source participatory development practices. In most practices open contribution rating is positively biased.

The criterion concerning OSH license's provision of commercial usability counteracts system's openness. The described interrelation is explicated early in the literature review and highlighted further in the case of Opendesk where the for-profit organization is branded as open but at the same time directly and indirectly obstructs open source bottom-up processes.

Apart from examining the openness criteria individually, it is equally or more important to explore their possible correlation. In terms of openness in the discussed framework, an important observation is that the freedom to *make* is thoroughly utilized when the user is first of all able to *study* and *modify* the design. Accordingly, the freedom to *study* is more thoroughly exercised when the user is able to *modify* the design or the piece of code. In other words, all of the four freedoms (*study*, *make*, *modify*, *distribute*) cumulatively account for more than the sum of each one freedom separately. Therefore the OSH freedoms interact in a synergetic way that is not reflected in open-o-meter methodology.

OSH theory's depth

The above-mentioned minor limitations of open-o-meter criteria regarding the examined subset can be addressed as follows:

a) CAD files available / editable.

The *WREN* case study reveals that CAD is not the only language of hardware's "source code". Hardware design apart from conventional digital drafting can also be parametric or computational design as expressed by visual programming and programming. The second point is that "source code", whether CAD or any other form, can be categorized further according to its content. All of the three case studies feature a clearly recognized division in design and fabrication files either are provided both, the one or the other. Fabrication files, which are derived from design files in a non-deterministic way, contain the information input required by either additive (3d printer) or substractive (CNC, laser cutter) digital fabrication media to manufacture hardware parts. But why are fabrication files so important as to be considered part of hardware's "source code"?

Undoubtedly, the hardware made completely out of standardized and off the shelf elements that are centrally and massively produced, does not need any fabrication files. On the contrary, the decision to analyze this specific OSH subset in the light of free innovation (Von Hippel, 2017) and digital fabrication revolution, is central to our research strategy. In this way, as the actual matter of research is the hardware, it seems self-evident that "source code" should refer to all the way up to the product's materialization. Fabrication files beyond being a substantial part of the freedom to *make*, they are important for one more reason. Fabrication files are directly related with parameters such as cost and environmental footprint among others that are extremely significant in a broader conception of OSH production. Consequently, given an evolving design practice, the fabrication file is the instant output of a knowledge intensive process which is object of continuous optimization from the community and certainly part of the product development.

Design and fabrication files as well as the intermediary process point to another associated limitation. How can we evaluate the openness of hardware based on shared CAD files or other type of design data without taking into account the openness of the design platform? An open design for *studying*, *making* and *modifying* that was shared in a native but proprietary format is a dead-end in itself. Design platform, whether the design is processed in conventional CAD, visual programming or classic programming, has to be open and developed as well as maintained in the Commons framework. The issue of proprietary platforms at the framework of netarchical capitalism was extensively described by Kostakis and Bauwens (2014b). We acknowledge the fact that currently open design platforms – especially beyond the conventional CAD – are scarce. This is partially attributed to the premature phase of the OSH in general, as well as to the nature and history of the – architectural – design itself regarding the intellectual property of the unique "handcrafted" design. In any case, it is relatively arguable to believe that OD platforms scarcity is subject to change in the near future. To conclude, CAD files criteria can be replaced by the following:

a.1) Design files available / editable.

a.2) Fabrication files available / editable.

a.3) Intermediary process from design to fabrication files available / editable.

a.4) Design platform – file format (FOSS compliant).

b) Guidelines for participation.

An open-o-meter's deficiency – that in different forms is discerned to most of the case studies – is that editable documents and a call for participation are not enough prerequisites to address the freedom to *modify* in its original full meaning. As the freedom to *modify* passed successively from OSS to OSH definitions, is gradually lost its collective dimension. Originally, the freedom to *modify* was addressing both the freedom to fork and modify a source code repository, as well as the freedom to merge modified code back to the master branch at your disposal. Instead of this rationale, the complete fulfilment of open-o-meter criteria can lead to an individual modus operandi that is so far from the collaborative intelligence of open source initiatives. An extreme example is Openstructures which is an aggregation of individually produced designs.

Even if participation guidelines or license allows directly or not all of the above mentioned freedoms to *modify*, none of them will ever occur without at least the required technical infrastructure. The respective infrastructure of OSS communities is an open and web-based collaboration platform that also serves for communication between contributors, version control, sharing and storing the source code. Similarly to the design platform, the open source character of the collaboration infrastructure is supporting the community and product's openness comparing to a proprietary platform. Additionally, the collaboration platform is a crucial point for supporting asynchronous collaboration which is so common to communities growing beyond spatial limitations.

Furthermore, the participation guidelines requirement does not define explicitly the aspect of unconditional participation which is fundamental characteristic of CBPP as demonstrated from OSS to Wikipedia paradigms. For instance, the Opendesk – which lost the point for participation guidelines – had issued a call for contributors in the past that is currently inactive. As the call was addressed to designers who were subject to further formal evaluation, it was a certainly closed procedure that formally would have passed open-o-meter generic evaluation. At this point, it is important to return to the OD dimensions and particularly participatory design to remind that contribution guide must be unconditional not only for the experts but for the users too.

To conclude it is suggested to replace the participation guidelines criterion with the more specific following:

b.1) Potential to fork repository.

b.2) Potential to merge back into repository.

b.3) Unconditional participation.

b.4) Collaboration platform (FOSS compliant).

c) Commercial usage allowed.

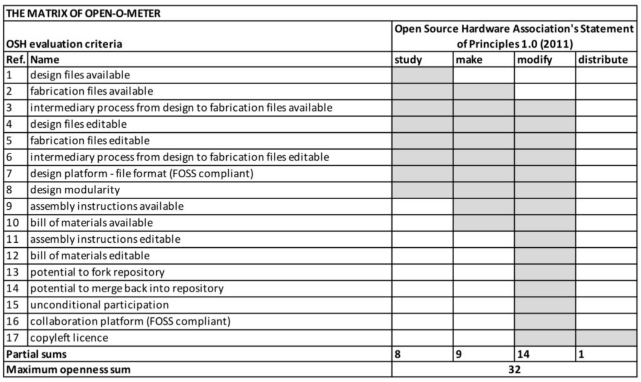
The case study analysis highlighted that OSH projects governed by for-profit organizations are characterized less open comparing to the non-profit communitites. Despite that case study research is not offered for inductive reasoning, this correlation can be further supported by the numerous OSS paradigms. The most interesting part is that the for-profit organizations employ a variety of direct and indirect techniques beyond licensing to limit bottom-up open practices. As the indirect techniques have been covered by further specification of openness criteria, what remains is a licensing strategy to replace the limitations and vagueness created by the commercial usability criterion. According to Vercellone et al. (2015) the mechanism of enclosure is what structurally differentiates cognitive capitalism from knowledge-intensive communities. On the other hand, it will be a paper by itself to analyze in-depth the different degrees and ways (from scratch, over derivative works, by selling exceptions or by parallel free distributon) that commercial usability as a mechanism of enclosure is incorporated to free and open source licenses. It is neither our scope nor it is feasible to initiate herein one more OSH license to face the raised issues. On the contrary, guided by the general concept of "*inappropriability*" (Vercellone et al., 2015: 62, 78), it could be simply supported that copyleft-minded licensing will certainly have a positive impact on hardware's openness. As the freedom to *run* mostly coincides with the freedom of *making* (executing the fabrication files), and the hardware's source code is actually immaterial, there are not so many differences to consider from OSS's to OSH's licensing.

d) Synergy – OSH freedoms' correlation

On the one hand, it is exposed by the examined case studies in the described context that the OSH freedoms (*study*, *make*, *modify*, *distribute*) are correlated to some extent but this is not easily and mathematically quantifiable. How can we quantify the contribution of the freedom to *modify* in the potential to *make* customized products comparing to standardized ones? On the other hand, it seems that the current open-o-meter scheme in which each OSH freedom corresponds to a completely different set of criteria is inaccurate. It is feasible from the case study research to define those criteria that contribute to more than one freedom. In this way, an indirect synergistic effect between the four freedoms will be approximately defined.

Specifically, the freedom to *study* is accomplished more efficiently when the "source code" is shared in editable format. While this argument actually depends on the nature of the "source code", it is generally arguable that trial and error is one of the basic methods for knowledge acquisition and in that way editable files add to the freedom of *studying*. The object of *study* exceeds the mere design files by including the fabrication ones as well as the intermediary process. Regarding the freedom to *make*, assembly instructions and BOM are insufficient means for making a custom object out of non-standardized parts. The availability of fabrication files is a minimum prerequisite of the *making* process. Additionally, the editable "source code" contributes even further to the freedom of *making* by freeing the customization of shared design according to local needs.

The following table synopsizes the described interrelations among the specified criteria:



THE MATRIX OF OPEN-O-METER					
OSH evaluation criteria		Open Source Hardware Association's Statement of Principles 1.0 (2011)			
Ref.	Name	study	make	modify	distribute
1	design files available	▨			
2	fabrication files available	▨	▨		
3	intermediary process from design to fabrication files available	▨	▨	▨	
4	design files editable	▨	▨	▨	
5	fabrication files editable	▨	▨	▨	
6	intermediary process from design to fabrication files editable	▨	▨	▨	
7	design platform - file format (FOSS compliant)	▨	▨	▨	
8	design modularity	▨	▨	▨	
9	assembly instructions available		▨	▨	
10	bill of materials available		▨	▨	
11	assembly instructions editable			▨	
12	bill of materials editable			▨	
13	potential to fork repository			▨	
14	potential to merge back into repository			▨	
15	unconditional participation			▨	
16	collaboration platform (FOSS compliant)			▨	
17	copyleft licence			▨	▨
Partial sums		8	9	14	1
Maximum openness sum		32			

Table 4. The matrix of open-o-meter (abbreviations: Free and Open Source Software [FOSS])

The criterion of design modularity is part of the next sub-section's discussion.

An emerging OD direction

As described in the literature review, modularity is a structural parameter of the open source mode of development and it is inextricably linked with the freedom to *study*, *modify* and *make*. Furthermore, modularity is what structurally differentiates open source code from proprietary code, Wikipedia from a conventional encyclopedia and so forth. As modularity is absent from OSH evaluation framework, it may be useful to analyze types of modularity that are observed in practice in order to theorize about them in a positive feedback loop. Generally, modularity can refer to different factors of OSH production such as: the community, the design, the physical product or even the procedure going from design to fabrication files. The types of modularity that have been observed in practice are the following:

- Openstructures features a geometric type of modularized design. Lower class objects (*parts*) and interfaces are designed, based on a fixed geometrical grid. Secondly, Openstructures features a type of hierarchical product segmentation as *components* are composed of *parts*, and successively *structures* are composed of *components*.

- Wikihouse potentially can be characterized by a non-hierarchical type of product modularity as *type* (house shell) can host other *technologies* (components) while not composed out of them. Potentiality is substantiated as currently the only provided component to be hosted is the *OWL* door-kit. However, the most promising part of Wikihouse, in terms of modularity, is the *WREN* system which is the intermediary process between design and fabrication files. *WREN* by its internal relational and procedural structure exemplifies the modularity in the closest form to the open source code modularity.

First of all, the fundamental principle of all modular systems is the standardization of the interface between modules. This is the basic difference between the lego-like natively modular design (Kostakis and Papachristou, 2014a) and Openstructures' modularized design. The freedom to design countless types of joints eliminates progressively the flexibility and reusability of the modules, leading to the solidification of hardware and design. Respectively, hierarchical product segmentation is incompatible within the open source mode of development and reveals a top-down rationale. The emergence of bottom-up assemblies between modules is more consistent with a truly open source community and design.

Furthermore, design modularity is not directly translated to product modularity. In other words design modules are not necessarily physical modules but modules of design knowledge. This fact is clearly demonstrated by *WREN*'s visual programming approach. Visual programming is positioned in the intersection of design and programming. Visual programming by its intrinsic algorithmic nature is more receptible to modularization. Traditional CAD, in contrast, is to great extent a monolithic sum of geometric and positional data that do not reveal the knowledge or the design process behind them. From the inception of CAD systems, researchers (Hanna et al., 1995) have attempted to propose more modular data structures that will be more susceptible to change and design re-usability. According to Terzidis (2006), CAD systems are a computerized version of the traditional hand drafting methodology, while the real potential in the utilization of computers in the design workflow lies in the computational design. Aish and Bredella (2017: 65) formulate the argument more explicitly: *"the increased use of parametric methods and scripting allow for the development of modelling and fabrication techniques, which in turn challenge the role of drawing as one of the main tools for conceptualising and realising architecture"*. Consequently, in the modularity framework, is it possible to articulate an OSH methodology out of conventional CAD drawings? In a similar comparison between the failure of collaborative writing of textbooks and the success of the collaborative writing of Wikipedia, Shirky (2005: 488) notes:

> *"Instead of assuming that Open Source methods are broadly applicable to the rest of the world, we can instead assume that that they are narrowly applicable, but so valuable that it is worth transforming other kinds of work, in order to take advantage of the tools and techniques pioneered here."*

During the last decade many architectural schools have integrated code learning to their curriculum (Papalexopoulos, 2011), going from a computerized analog drafting method to a computational workflow. Additionally, many individually produced design code snippets have been uploaded to proprietary platforms' repositories[5] due to the aforementioned lack of OD platforms. Even the little known controversy at 2011

(Davis, 2011; Piker, 2011) between the creator of the freely distributed plugin named *Kangaroo* and the geometry rationalization company *Evolute GmbH*, is apocalyptic of the shift of interest from the drawing to the code[6]. In this sense, *WREN* system may exemplify the future direction of OD in the OSH framework. Actually, *WREN* system concentrates the properties of what Papalexopoulos (2011) characterizes as Digital Design Commons:

> *"Digital Design Commons are pools of a multitude of micro- architecture problem solutions, a multitude of micro – syntaxes covering partial aspects of design, waiting to be actualized in larger design schemes.*
>
> *They also deny the unique and ultimate "form" in favor of a network's syntax. They tend to substitute the object's design with the design of networked multiplicities. Finally, they question the ubiquity of design as an end of work process, linking it to the (local) use value production."*

CONCLUSION

The aforementioned analysis shows that the OSH theoretical framework as specified by open-o-meter is not generally applicable to the whole range of OSH. In the examined subset of OSH, open-o-meter lacks specificity, which is easily revealed under the light of case study research but covered by large scale statistical analyses that have been performed before this paper. The over-simplification leads to positively biased results at the examined sample but the direction of the deviation can not be safely generalized for the full range of hardware. Intrinsic contradictions of open-o-meter are dependent on the political investment of open source movement: constituent of knowledge-based economy or cognitive capitalism. The author stands for the first position which is deeply rooted to the intrinsic properties of the open source code.

The matrix of open-o-meter is a suggestion to fine-tune the open-o-meter based on qualitative evidence without altering its quantitative and generic nature. However, the applicability of the matrix to the whole range of hardware is subject to future research. Furthermore, the research highlighted the hidden structural dimension of modularity in the OSH development and described an emergent type of design modularity beyond the much discussed geometric modularization. Also further investigation is required to establish an evaluation framework of modularity among different design media. This is a known limitation of the matrix of open-o-meter. Additionally, the proposed version of the matrix is delimited by the specific socio-technological research context of the digital fabrication and the digital Commons.

While OSS had long been a practice without theory, this condition is inverted when open source principles are applied outside their original domain. In this way, the articulation of a consistent and comprehensive OSH theory is of crucial importance in further OSH diffusion.

REFERENCES

Ackermann, John (2009) "Toward Open Source Hardware", University of Dayton Law Review, 34 (2), pp. 183–222.

Aish, Robert and Nathalie Bredella (2017) "The evolution of architectural computing: From Building Modelling to Design Computation", *Architectural Research Quarterly*, 21(1), pp. 65-73.

Alexander, Christopher, Ishikawa, Sara, Silverstein, Murray, Jacobson, Max, Fiksdahl-King, Ingrid and Shlomo Angel (1977) A Pattern Language: Towns, Buildings, Construction. New York: Oxford University Press.

Baldwin, Carliss Y. and Kim B. Clark (2002) "The Option Value of Modularity in Design: An Example from Design Rules, Volume 1: The Power of Modularity", Harvard NOM Working Paper, No. 02-13; Harvard Business School Working Paper, No.

02-078, <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=312404>, retrieved: 10 Jun 2018.

Baldwin, Carliss Y. and Kim B. Clark (2006) “The Architecture of Participation: Does Code Architecture Mitigate Free Riding in the Open Source Development Model?”, Management Science, 52, issue 7, pp. 1116-1127.

Balka, Kerstin, Raasch, Christina and Cornelius Herstatt (2009a) “Open source enters the world of atoms: A statistical analysis of open design”, First Monday, 14(11), <http://firstmonday.org/ojs/index.php/fm/article/view/2670/2366>, retrieved: 10 Jun 2018.

Balka, Kerstin, Raasch, Christina and Cornelius Herstatt (2009b) “Open Source beyond software: An empirical investigation of the open design phenomenon”, R&D Management Conference, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.595.3840&rep=rep1&type=pdf>, retrieved: 10 Jun 2018.

Balka, Kerstin, Raasch, Christina and Cornelius Herstatt (2010) “How Open is Open Source? – Software and Beyond”, *Creativity and Innovation Management*, 19/3, pp. 248-256.

Balka, Kerstin, Raasch, Christina and Cornelius Herstatt (2014) “The Effect of Selective Openness on Value Creation in User Innovation Communities”, *Journal of Product Innovation Management*, 31, pp. 392-407.

Benkler, Yochai (2002) “Coase’s Penguin, or, Linux and “The Nature of the Firm””, The Yale Law Journal, 112(3), pp. 369-446.

Benkler, Yochai (2006) The Wealth of Networks: How Social Production Transforms Markets and Freedom. New Haven: Yale University Press.

Bessen, James (2002) “What good is free software?” in R. W. Hahn (ed.) Government policy toward open source software. Washington, DC: Brookings Institution Press, pp. 12-33.

Bonvoisin, Jérémy, Mies, Robert, Boujut, Jean-François and Rainer Stark (2017) “What is the “Source” of Open Source Hardware?”, Journal of Open Hardware, 1(1): 5, pp. 1–18.

Carson, Kevin A. (2010) The Homebrew Industrial Revolution: A Low-Overhead Manifesto. Charleston, SC: BookSurge Publishing.

Chesbrough, Henry W. and Melissa M. Appleyard (2007) “Open Innovation and Strategy”, California Management Review, 50(1), pp.57-76.

Conway, E. Melvin (1968) “How do Committees Invent?”, Datamation, Volume 14 (5), pp.28-31.

Davis, Daniel (2011) “Patenting Geometry”, <http://www.danieldavis.com/patenting-geometry/>, retrieved: 01 Jul 2018.

De Mul, Jos (2011) “Redesigning Design” in Bas van Abel, Lucas Evers, Roel Klaassen and Peter Troxler (eds) Open Design Now. Amsterdam: BIS Publishers, pp. 34-41.

DeLanda, Manuel (2006) A New Philosophy of Society: Assemblage Theory and Social Complexity. London: Continuum.

Free Software Foundation (2016) “Why “Free Software” is better than “Open Source””, <https://www.gnu.org/philosophy/free-software-for-freedom.html>, retrieved: 10 Jun 2018.

Friedman, Yona (1975) Toward a Scientific Architecture. (transl. Lang Cynthia). Cambridge: MIT Press.

Fuller, Matthew and Usman Haque (2008) “Urban Versioning System 1.0” in Khan, Omar, Sholz, Trebor and Mark Shepard (eds.) Situated Technologies Pamphlet Series. New York: Architectural League of New York.

Gershenfeld, Neil (2007) FAB: The Coming Revolution on Your Desktop: From Personal

Computers to Personal Fabrication. New York: Basic Books.

Gershenfeld, Neil (2012) "How to make almost anything: the digital fabrication revolution", Foreign Affairs, 91(6), pp. 42–57, <http://cba.mit.edu/docs/papers/12.09.FA.pdf>, retrieved: 10 Jun 2018.

Gershenfeld, Neil, Gershenfeld, Alan and Joel Cutcher-Gershenfeld (2017) Designing Reality: How to Survive and Thrive in the Third Digital Revolution. New York, NY: Basic Books.

Gibb, Alicia (2014) Building Open Source Hardware: DIY Manufacturing for Hackers and Makers. Upper Saddle River, NJ: Addison-Wesley Professional.

Habraken, John (1972) Supports – An Alternative to Mass Housing. London: The Architectural Press.

Hanna, Paul J.R., Millar, Richard J. and John Hamilton Frazer (1995) "A CAD data structure to facilitate change", *Computing Systems in Engineering*, Vol 6, Issue 6, pp.511-519.

Hermann, Mario, Pentek, Tobias and Boris Otto (2016) "Design Principles for Industrie 4.0 Scenarios", *49th Hawaii International Conference on System Science*, IEEE Press, pp.3928-3937.

Hiller, Jonathan and Hod Lipson (2009) "Design and analysis of digital materials for physical 3D voxel printing", *Rapid Prototyping Journal,* 15/2, pp.137-149.

Kaspori, Dennis (2003) "A Communism of ideas: Towards an open-source architectural practice", *Archis*, Issue 3, pp. 13-18.

Kostakis, Vasilis, Fountouklis, Michail and Wolfgang Drechsler (2013) "Peer Production and Desktop Manufacturing: The Case of the Helix_T Wind Turbine Project", *Science, Technology, & Human Values*, Vol 38, Issue 6, pp. 773 – 800.

Kostakis, Vasilis and Marios Papachristou (2014a) "Commons-based peer production and digital fabrication: The case of a RepRap-based, Lego-built 3D printing-milling machine", *Telematics and Informatics*, 31(3), pp. 434–443.

Kostakis, Vasilis and Michel Bauwens (2014b) *Network Society and Future Scenarios for a Collaborative Economy*. London: Palgrave Macmillan.

Kostakis, Vasilis, Niaros, Vasilis, Dafermos, George and Michel Bauwens (2015) "Design Global, Manufacture Local: Exploring the Contours of an Emerging Productive Model", *Futures,* Volume 73, pp.126-135

Lakhani, Karim (2007) "Communities Driving Manufacturers Out of the Design Space", *The Future of Communities Blog*, <https://web.archive.org/web/20080602024947/http://www.futureofcommunities.com/2007/03/25/communities-driving-manufacturers-out-of-the-design-space>, retrieved: 10 Jun 2018.

Lerner, Josh and Jean Tirole (2005) "The Economics of Technology Sharing: Open Source and Beyond", *Journal of Economic Perspectives 19*, no. 2, pp.99–120. (Earlier version distributed as NBER Working Paper Series No. w10956.)

MacCormack, Alan, Rusnak, John and Carliss Baldwin (2006) "Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code", *Management Science,* Volume 52, No. 7, pp.1015-1030.

MacCormack, Alan, Baldwin, Carliss and John Rusnak (2012) "Exploring the duality between product and organizational architectures: A test of the "mirroring" hypothesis", *Research Policy,* Volume 41, pp.1309-1324.

Manzini, Ezio (2015) Design, when everybody designs: an introduction to design for social innovation. Cambridge, MA: The MIT Press.

Maurer, Stephen M. and Suzanne Scotchmer (2006) "Open Source Software: The new intellectual property paradigm", *NBER Working Paper Series,* No. 12148, <http://www.nber.org/papers/w12148>, retrieved: 10 Jun 2018.

Niaros, Vasilis, Kostakis, Vasilis and Wolfgang Drechsler (2017) "Making (in) the smart city: The emergence of makerspaces", *Telematics and Informatics*, Volume 34, Issue 7, pp. 1143-1152.

Nuvolari, Allesandro and Francesco Rullani (2007) "Curious Exceptions? Open Source Software and "Open" Technology" in Kirk St. Amant and Brian Still (eds.) *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives.* Hershey, New York: Information Science Reference, pp. 227-239.

Open Architecture License (2016) <https://github.com/dortheimer/open-architecture>, retrieved: 24 Oct 2018.

Open Architecture Network, <https://web.archive.org/web/20150315014820/http://openarchitecturenetwork.org/> (archived version), retrieved: 24 Oct 2018.

Open Source Hardware Association (2011) "Open Source Hardware (OSHW) Statement of Principles 1.0.", <https://freedomdefined.org/index.php?title=OSHW&oldid=9134>, retrieved: 10 Jun 2018.

Open Source Initiative (2007) "The Open Source Definition", <https://opensource.org/docs/osd >, retrieved: 10 Jun 2018.

OPEN! (2018) "Open-O-meter", <https://opensourcedesign.cc/wiki/index.php/Open-O-meter>, retrieved: 01 Jul 2018.

Opendesk (2018) <https://www.opendesk.cc/>, retrieved: 01 Jul 2018.

Openstructures (2018a) <http://openstructures.net/>, retrieved: 01 Jul 2018.

Openstructures (2018b) <http://beta.openstructures.net/pages/2>, retrieved: 01 Jul 2018.

Osterloh, Margit and Sandra Rota (2007) "Open Source Software development – Just another case of collective invention?", *Research Policy*, 36(2), pp. 157-171.

Papalexopoulos, Dimitris (2011) "Digital Design Commons", *Symposium: Computational Politics and Architecture: From the Digital Philosophy to the End of Work*, <http://atru.arch.ntua.gr/sites/atru/files/posts/50digital-design-commons/papalexopoulosensapm2011.pdf>, retrieved: 12 Nov 2017

Perens, Bruce (1997) "Debian's "Social Contract" with the Free Software Community", <https://lists.debian.org/debian-announce/1997/msg00017.html>, retrieved: 10 Jun 2018.

Piker, Daniel (2011) "Patents, Precedents and Geometry", <https://spacesymmetrystructure.wordpress.com/2011/09/03/patents-precedents-and-geometry/>, retrieved: 01 Jul 2018.

Raasch, Christina, Herstatt, Cornelius and Kerstin Balka (2009) "On the open design of tangible goods", *R&D Management*, 39, pp. 382-393.

Ratti, Carlo and Matthew Claudel (2015) *Open Source Architecture*. London: Thames & Hudson.

Raymond, Eric S. (1999a) "Afterword: Beyond Software?", <http://www.catb.org/esr/writings/homesteading/afterword/>, retrieved: 10 Jun 2018.

Raymond, Eric S. (1999b) "The Cathedral and the Bazaar", <http://www.catb.org/~esr/writings/cathedral-bazaar/>, retrieved: 12 Nov 2017

Rifkin, Jeremy (2014) *The Zero Marginal Cost Society: The internet of things, the collaborative* *commons, and the eclipse of capitalism*. New York: Palgrave Macmillan.

Salingaros, Nikos A., Caperna, Antonio, Bauwens, Michel, Brain, David, Duany, Andrés M., Mehaffy, Michael W., Mehta, Geeta, Mena-Quintero, Federico, Philibert-Petit, Ernesto, Rizzo, Agatino, Serafini, Stefano and Emanuele Strano (2010) *P2P Urbanism*. <http://zeta.math.utsa.edu/~yxk833/P2PURBANISM.pdf>, retrieved: 23 Oct 2018.

Shirky, Clay (2005) “Epilogue: Open Source outside the Domain of Software” in Joseph Feller, Brian Fitzgerald, Scott A. Hissam, and Karim R. Lakhani, *Perspectives on Free and Open Source Software*. Cambridge, MA: The MIT Press.

Shirky, Clay (2008) *Here comes everybody: the power of organizing without organizations*. New York: Penguin Press.

Stake, Robert (1995) *The Art of Case Study Research*. Thousand Oaks, CA: Sage.

Stake, Robert (2003) “Case studies” in Denzin, Norman K. and Yvonna S. Lincoln (eds.) Strategies of Qualitative Inquiry. Thousand Oaks, CA: Sage, pp. 134-164.

Stallman, Richard (1986) “What is the Free Software Foundation”, *GNU’s Bulletin*, 1 (1), pp. 8-9, <https://www.gnu.org/bulletins/bull1.txt>, retrieved: 10 Jun 2018.

Stallman, Richard (1999) “On “Free Hardware””, <https://www.linuxtoday.com/infrastructure/1999062200505NWLF>, retrieved: 10 Jun 2018.

Stallman, Richard (2015) “Why we need Free Digital Hardware Designs”, *Wired Magazine*, <https://www.wired.com/2015/03/need-free-digital-hardware-designs/>, retrieved: 10 Jun 2018.

Stallman, Richard (2016) “Why Open Source misses the point of Free Software”, <https://www.gnu.org/philosophy/open-source-misses-the-point.html>, retrieved: 10 Jun 2018.

TAPR (2007) “The TAPR Open Hardware License”, <https://www.tapr.org/TAPR_Open_Hardware_License_v1.0.pdf>, retrieved: 01 Jul 2018.

Terzidis, Kostas (2006) *Algorithmic Architecture*. Oxford: Elsevier Architectural Press.

Thompson, Clive (2008) “Build it. Share it. Profit. Can Open Source Hardware work?”, *Wired Magazine*, < https://www.wired.com/2008/10/ff-openmanufacturing/>, retrieved: 10 Jun 2018.

Vallance, Ryan, Kiani, Sepehr and Samir Nayfeh (2001) “Open design of manufacturing equipment”,

Proceedings of the CHIRP 1st International Conference on Agile, Reconfigurable Manufacturing, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.557.1285&rep=rep1&type=pdf>, retrieved: 10 Jun 2018.

Vardouli, Theodora and Leah Buechley (2014) “Open Source Architecture: An exploration of Source Code and Access in Architectural Design”, *Leonardo / ISAST*, 47 (1), pp. 51–55.

Velkova, Julia (2016) “Open cultural production and the online gift economy: The case of Blender”, *First Monday*, 21 (10), <http://firstmonday.org/ojs/index.php/fm/article/view/6944>, retrieved: 10 Jun 2018.

Vercellone, Carlo, Bria, Francesca, Fumagalli, Andrea, Gentilucci, Eleonora, Giuliani, Alfonso, Griziotti, Giorgio and Pierluigi Vattimo (2015) “Managing the commons in the knowledge economy”, <https://dcentproject.eu/wp-content/uploads/2015/07/D3.2-complete-ENG-v2.pdf>, retrieved: 25 Dec 2018.

Von Hippel, Eric (2005) *Democratizing Innovation*. Cambridge, MA: The MIT Press.

Von Hippel, Eric (2017) *Free Innovation*. Cambridge, MA: The MIT Press.

Von Hippel, Eric and Georg von Krogh (2003) “Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science”, *Organization Science 14*, 2, pp. 209-223.

Von Krogh, Georg and Eric Von Hippel (2006) “The Promise of Research on Open Source Software”, *Management Science*, 52(7), pp. 975-983.

Weber, Steven (2004) *The Success of Open Source*. Cambridge, MA: Harvard University Press.

Wikihouse (2018a) <https://wikihouse.cc/>, retrieved: 01 Jul 2018.

Wikihouse (2018b) <https://wikihouse.cc/about>, retrieved: 01 Jul 2018.

Wikipedia (2018a) “List of open-source hardware projects”, <https://en.wikipedia.org/wiki/List_of_open-source_hardware_projects>, retrieved: 17 Jun 2018.

Wikipedia (2018b) “Wikihouse”, <https://en.wikipedia.org/wiki/WikiHouse>, retrieved: 01 Jul 2018.

ENDNOTES

[1] The author consulted the Wikipedia list of OSH projects on the 17th of June 2018.

[2] https://xtools.wmflabs.org/articleinfo/en.wikipedia.org/List_of_open-source_hardware_projects The “List of open-source hardware projects” is a dynamic entry first edited in 2010, featuring 683 edits by 328 editors at the time of writing, most of them characterized as major edits. Last visit: 17/07/2018

[3] https://github.com/wikihouseproject/, last visit: 01/07/2018

[4] https://github.com/wikihouseproject/Microhouse/, last visit: 01/07/2018

[5] https://www.food4rhino.com/, last visit: 01/07/2018.

[6] It is worth noting that when this paper was under review, Opendesk launched customization services of existing designs based on algorithmic processes and parametric design. This is one more fact that augments the belief that a tranformative transition from drawing to code is taking place. The customization services were not openly available which is consistent with the precedent analysis and the initiative’s for-profit strategy. https://www.opendesk.cc/blog/parametric-design-and-tailoring, last visit: 23/12/2018.

APPENDIX

List of Acronyms

BOM	Bill of materials
CBPP	Commons-based Peer Production
DGML	Design Global Manufacture Local
FOSS	Free and Open Source Software
FSF	Free Software Foundation
OD	Open Design
OSH	Open Source Hardware
OSI	Open Source Initiative
OSS	Open Source Software