On the openness of digital platforms/ecosystems

Jose Teixeira
Turku School of Economics
University of Turku
Turku, Finland
jose.teixeira@utu.fi

ABSTRACT

A plenitude of technology is neither developed in-house nor simply outsourced in dyadic relationships. Instead, we are in a new age where technologies are developed by a networked community of actors and organizations, which base their relations dynamically to each other on a common interest. Such dynamic and networked complexity of technology development is often theoretical explored around the concept of platform, and more recently by employing the concept of ecosystem in an analogy to natural ecosystems. Following the success of open-source software, academics have long been examining openness in digital platforms/ecosystems; however most contributions take the perspective of a single stakeholder from the many that constitute a digital platform/ecosystem. Predominantly, they take the sole perspective of platform providers, those bundling hardware and software or more rarely, the perspective of third-party software developers developing valuable software 'apps' that add value to the overall platform. In this conceptual article, we grasp openness more holistically, both by acknowledging that openness means different things to different people and involve all stakeholders within the platforms/ecosystems. Towards the development of a theory of openness within digital settings, we propose six novel aspects of openness for enabling a greater understanding of the open-source software movement with a digital platforms/ecosystems perspective. Moreover, we invite scholars to reconsider the more predominating product-dominant logic in open-source software research to a more holistic logic embracing platforms and ecosystem thinking.

Categories and Subject Descriptors

D.9 [**Management**]: Programming teams; K.6.3 [**Software Management**]: Software development,Software process; H.0 [**Information Systems**]: GENERAL

General Terms

Economics, Management, Theory

 ACM 978-1-4503-3666-6/15/08…$15.00 DOI: http://dx.doi.org/10.1145/2788993.2789829

Keywords

OSS, FLOSS, Open-Source, Open-Hardware, Open-Data, Open-Access, Openness, Platforms, Ecosystems, Digital Infrastructures

1. ON OPENNESS AS GENERALLY ADDRESSED

Most of us already know that Google Android is open, that Apple uses a lot of open-source or that Nokia abandoned its open-source platforms when allying with Microsoft. When following the converging PC and mobile-devices industry, we can easily spot arguments on openness by vendors, analysts, journalists, software application (app) developers, etc. Openness traditionally refers to software products (e.g. Apache or MySQL), but this short paper handles the concept within platforms (i.e. technologies that enable the creation of third-party products and services) and ecosystems (i.e. networks of companies interacting with each other, directly and indirectly, to provide a broad array of products and services).

Academics have long been examining openness in digital platforms/ecosystems: Some investigate the open-source software phenomenon [6, 1]; some review the open-source strategies employed by different platform-vendors [33]; some benchmark the architectural openness of different platform stacks [2]; and many others take the view of software developers by investigating their perceptions of platform openness [10]. Sampling the multidisciplinary relevance of the open-source phenomenon, we observed that many prominent academic outlets, including *Research Policy*, *IEEE Network*, *IEEE Software*, *Management Science*, *Criticism* and the *Journal of the Association for Information Systems* [31, 24, 32, 5, 4, 14, 15], have published special issues on open-source software. Also evidencing the wide volume of research addressing the open-source phenomenon, many exaustive literature review articles pertaining the topic are available [27, 1, 9, 16, 11, 6, 28].

The phenomenon has drawn recurrent attention across a wide array of disciplines, from the more technical ones (i.e., Computer Science and Software Engineering) to more social-oriented ones (i.e., Management and Innovation studies). The vast and heterogeneous volume of research addressing the open-source software phenomenon has been tentatively explained by 1) its impact: open source software has an extensive impact on the economy and society; (2) its theoretical tension: the phenomenon deviates sharply from the existing theory in different fields; (3) its transparency: open source software has offered unprecedented access to research-data; (4) its communal reflexivity: the open-source

community frequently engage in a dialog on its functioning (it also has its own research community); and (5) its proximity with academia: processes in open source software resembles knowledge production in science (in many instances, open source software is an output of research processes) [30].

On the industry side, practitioners, which often deal with open vs. closed technological dilemmas, also provided valuable theoretical contributions by proposing a novel way of measuring openness in digital platforms/ecosystems by taking a governance point of view [13]. In the terms 'open-data' and 'open-content', the Open Knowledge high-networked organization also proposed a definition of "openness" in relation to data and content [20] which re-emphasizes the original four Free Software Freedoms initially proposed by Stallman [25, 26]. In medicine, Dr. A. Kogelnik founded the open-medicine institute pinpointing open-source principles which call for systems and information to be shared in a community-based, collaborative manner; the combination of community and technology are stated as the driving force behind the open-medicine approach to medicine [19]. Other pioneers of the open-medicine movement also explicitly call for more 'open-source' in cancer and drug-discovery research [18, 3]. After all, if open-source software reshaped science and the IT industry, why would other areas be immune to increased openness?

Even if openness is recurrently being addressed both by academics and practitioners, most contributions take the perspective of a single stakeholder from the many that constitute a digital platform/ecosystem. Predominantly, they take the sole perspective of platform providers, those bundling hardware and software [17, 29]; or more rarely, the perspective of third-party software developers developing valuable software apps that add value to the overall platform [23] If we want to be more inclusive and handle openness more holistically, we must start by acknowledging that openness means different things to different people [8] and involving all stakeholders within the platforms/ecosystems. By taking this more inclusive approach, as illustrated in Figure 1, we can obtain sharper insights on what is meant by openness and how we can measure it more consensually. This short paper aims to 'open minds' on openness by suggesting six different aspects of openness that emerge when investigating platforms/ecosystems with a more inclusive and holistic approach.

2. SIX DIFFERENT ASPECTS OF OPENNESS

First, openness is not just about software; it is also about hardware. A fully-open platform/ecosystem reveals itself across the overall platform stack from Silicon to Software (StoS): it starts with open chip-design, open-hardware and open-source software. For instance, Arduino and Raspberry Pi built up a great reputation as electronic prototyping platforms. They allow innovators to prototype ingenious combinations of open-hardware and open-source software, as evidenced by the emerging 3D printing industry. However, when innovators move from prototyping to production, the Arduino and Rasberry Pi chips are filled with ARM corporate proprietary Intellectual Property. The openness in hardware and the openness in software contrasts with the closedness in chip-design. Any innovator trying to sell products based on the above mentioned electronic prototype

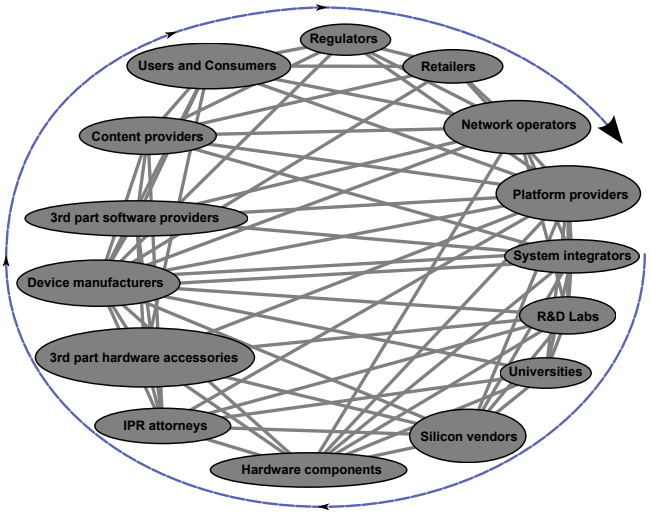

Figure 1: Assessing how different stakeholders construct openness in a platform/ecosystem

platforms is worse off than ARM corporate law attorneys. Holders of large patent portfolios, such as ARM, can freeze shipments and boost the legal expenses of innovators by simply filling DMCA take-down notices. In brief, within digital platforms/ecosystems **openness is not necessarily just about software**, it is about the overall platform stack.

Second, openness is not simply a function of the openness of its artifacts (hardware and software), but also a function of the way in which the same artifacts are developed. As in open-source software, openness in digital platforms/ecosystems is not just about the legal compliance with certain open-source software licenses [21], but also about the transparency and inclusiveness of its governance. For instance, the development transparency of Android is obscured by Google that keeps two source-code repositories where only a restricted set of partners have access to the newest version of Android's blueprints, while the public-domain is left with the older versions. Regarding inclusiveness, if third-party software developers were more included in the governance of digital platform/ecosystems, we would witness fewer defective-by-design features, and less planned-obsolescence protecting the business models from platform-core vendors. For instance, Apple has limited interest in supporting 3D acceleration or Flash technology in their mobile browsers in order to protect revenues from the Apple's App Store. They do not wish users to play games directly from developers' websites, but force users and developers to depend on Apple's own infrastructure for trading mobile games. A fully-open platform/ecosystem **is not just about the openness of its artifacts but also about the transparency and inclusiveness of its governance**.

Third, novel digital platforms/ecosystems incorporate new market features for the distribution of third-party complements (e.g. Apple's App Store and Google's Play). Openness is also visible in those third-party distribution markets. To be fully open platforms/ecosystems, they should implement **free-market** economic policies, with little or no protectionism of parts, and should reward successful innovators and

entrepreneurs. Current third-party developers stumble on closed markets where the ranking and featuring of apps is done in an opaque and inconsistent way. Platform-core vendors often feature and protect the apps of selected partners at the expense of bootstrap innovators and entrepreneurs that struggle to increase the visibility of their innovative apps without access to market intelligence data (solely in the hands of the platform-core owner) [23, 7].

Fourth, fully-open platforms/ecosystems **cannot be closed at side by legal mechanisms** such as end-user license agreements (EULA), anti-fragmentation agreements (AFA), non-disclosure agreements (NDA) or patents portfolios hindering the use of relevant technologies. Fully-open platforms/ecosystems must implement an open intellectual property regime where social norms and values lower the need for legal contracts and reduce the involvement of law attorneys. The use of patent portfolios, both blocking new entrants and restricting competition and in the mobile device industry, leads to paradoxical situations, such as the one where the HTC mobile-device maker is paying more royalties to Microsoft than to Google when selling its handsets based on Google's Android. Decision makers in fully-open platforms/ecosystems should invest more in R&D than intellectual property pooling.

Another characteristic of fully-open platforms/ecosystems regards its compliance with standards. They use, promote and influence standards that do not prohibit conforming implementations in an open-source way [22]. Standards implementations should be interoperable and publicly available by royalty-free terms at reasonable and non-discriminatory costs. All patents essential to implementation of such standards must be licensed under royalty-free terms for unrestricted use or, alternatively, be covered by a promise of non-assertion when employed by open-source software. To illustrate some of the antagonistic effects of mixing standards with patents, we can learn from the World Wide Web Consortium (W3C) codec-wars: With so many audio/video standards affected by many known software patents, the World W3C updated its HTML5 specs to support the Theora video and Vorbis audio open-source formats in a move to keep the Internet free from the danger of "submarine" patents issued by large companies. Submarine patents are patents whose issuance and publication are intentionally delayed. They allow applicants to promote standards protected by patents that stay "underwater" for long periods until they "emerge" and surprise the relevant markets. Overall, fully-open platforms/ecosystems **empower standards available to everyone, without patents** constraining their implementation in the public domain.

Finally, fully-open platforms/ecosystems **should not employ defective-by-design lock-in mechanisms** that technically or practically increase the switching-costs of users. For example, a few years ago it was very easy to switch mobile phones: one simply needed to copy their contacts to the SIM card and move them from one device to another. In the smartphone era, switching to a device from another vendor is highly impractical due to defective-by-design lock-in mechanisms: What would happen to the contacts, the messages, the music playlists, the downloaded maps, the internet-bookmarks, the installed apps and other data? What about the new vocabulary entries added to the device input-methods dictionaries? Are they open enough so that we can move them from one device to another? Is it technically possible? And practicable?

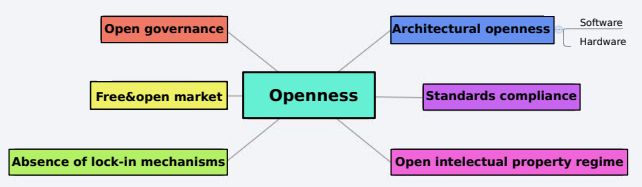

Figure 2: Assessing how different stakeholders construct openness in a platform/ecosystem

The Figure 2 visually agregates the proposed six aspects of openness regarding 1) architectural (hardware and software); 2) compliance with standards; 3) transparency and inclusiveness of governance; 4) free market policies rewarding innovation and entrepreneurhip ; 5) presence/absence of purposive lock-in mechanisms; and 6) an open regime of intellectual property.

3. A LIVELY EXAMPLE OF OPENNESSES

It might be that a fully-open platform/ecosystem is a utopia, an abstract and theoretical concept never implementable in practice. This viewpoint might argue that Google Android or Apple iOS are actually quite closed, but it also reveals one example of a champion in openness: The Milkymist [12], a video DJ computer-based platform/ecosystem initiated by a young French enthusiast. The hardware is open, the software is open, the community is open. Apparently they are not rich, they do not grab much media attention, but they innovate transparently and inclusively without patents, without R&D subsidies from governments or access to high-capital provided by banks. Their platform can be studied, modified and actually distributed by everyone interested in their technology while a sense of fairness is kept within their community.

Table 1: Why is Milkymist a champion in openness?

Aspect	Evidences
Architecture	LatticeMico32 core CPU under GPL open-source license SoC and FPGA blueprints under GPL open-source license Open-source software under GPL open-source license
Governance	Development infrastructure under public domain Community driven development Supporting derivative works (forks)
Market	No evidence of entrance restrictions No evidence of parts/stakeholders protectionism Public access to market-relevant community data
Intellectual property	Copyrights under an open-source approved license Absence of protective patents Absence of protective legal agreements
Standards compliance	Use of technical open-standards
Lock-in mechanisms	No evidence of defective-by-design users lock-in

4. CONCLUDING REMARKS

In conclusion, the echoes of the open-source software phenomenon are being heard outside its original software boundaries. Nowadays, we do not talk just about open-source, but also about open-data, open-hardware, open-platform, open-access, open-medicine, etc.. The openness of open-source software is demanded in other domains, however, with different meanings of openness. The six aspects of openness

introduced here should be useful to scholars and practitioners by enabling a greater understanding of the open-source software phenomenon outside the domain of pure software products. We conclude by remarking that we are not arguing for openness (i.e. the more the better) but exploring it in a digital platforms and ecosystems setting.

5. ACKNOWLEDGEMENTS

Besides the anonymous reviewers, I would like to thank early feedback on this paper by Jaana Nevalainen, Klaas-Jan Stol, Pádraig O'Leary, John Noll and Nea Kalleinen. This work was partially supported by the Fundação para a Ciência e a Tecnologia (grant SFRHBD615612009), Liikesivistysrahasto (grant 3-1815) and Marcus Wallenberg Säätiö (grant open-coopetition R&D management strategy).

6. REFERENCES

[1] A. Aksulu and M. Wade. A comprehensive review and synthesis of open source research. *Journal of the Association for Information Systems*, 11(11):576–656, 2010.

[2] M. Anvaari and S. Jansen. Evaluating architectural openness in mobile software platforms. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, pages 85–92. ACM, 2010.

[3] J. Bradner. Open-source cancer research | Talk Video | TED.com, 2011.

[4] A. Ceraso and J. Pruchnic. Introduction: Open source culture and aesthetics. *Criticism*, 53(3):337–375, 2011.

[5] K. Crowston and M. Wade. Introduction to jais special issue on empirical research on free/libre open source software. *Journal of the Association for Information Systems*, 11(11):20–27, 2010.

[6] K. Crowston, K. Wei, J. Howison, and A. Wiggins. Free/libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR)*, 44(2):7, 2012.

[7] R. Garg and R. Telang. Inferring app demand from publicly available data. *MIS Quarterly*, 37(4):1253–1264, 2013.

[8] A. M. Grubb and S. M. Easterbrook. On the lack of consensus over the meaning of openness: An empirical study. *PloS one*, 6(8):e23420, 2011.

[9] Ø. Hauge, C. Ayala, and R. Conradi. Adoption of open source software in software-intensive organizations–A systematic literature review. *Information and Software Technology*, 52(11):1133–1154, 2010.

[10] D. Hilkert, A. Benlian, M. Sarstedt, and T. Hess. Perceived software platform openness: the scale and its impact on developer satisfaction. 2011.

[11] M. Höst and A. Oručević-Alagić. A systematic review of research on open source software in commercial software product development. *Information and Software Technology*, 53(6):616–624, 2011.

[12] M. Labs. Milkymist one, 2014.

[13] L. Laffan. A new way of measuring openness: The open governance index. *Technology Innovation Management Review*, (January 2012: Open Source Business), 2012.

[14] Y.-D. Lin, R.-H. Hwang, G. Armitage, and V. Eramo. Guest editorial: Open source for networking: Protocol stacks. *Network, IEEE*, 28(2):2–5, 2014.

[15] Y.-D. Lin, R.-H. Hwang, G. Armitage, and V. Eramo. Open source for networking: Tools and applications [guest editorial]. *Network, IEEE*, 28(5):4–5, September 2014.

[16] J. Lindman. *Not Accidental Revolutionaries: Essays on Open Source Software Production and Organizational Change*. Aalto University, School of Economics, Department of Information and Service Economy, 2011.

[17] S. Q. Mian, J. Teixeira, and E. Koskivaara. Open-source software implications in the competitive mobile platforms market. In *Building the e-World Ecosystem*, pages 110–128. Springer, 2011.

[18] B. Munos. Can open-source R&D reinvigorate drug research? *Nature Reviews Drug Discovery*, 5(9):723–729, 2006.

[19] OMI. Overview | Open Medicine Institute, 2015.

[20] Open Knowledge. The Open Definition - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge, 2015.

[21] T. O'Reilly. Lessons from open-source software development. *Communications of the ACM*, 42(4):32–37, 1999.

[22] OSI. Open standards requirement for software | open source initiative, 2014.

[23] J. Salminen and J. Teixeira. Fool's gold? developer dilemmas in a closed mobile application market platform. In *Co-created Effective, Agile, and Trusted eServices*, pages 121–132. Springer, 2013.

[24] D. Spinellis and C. Szyperski. Guest editors' introduction: How is open source affecting software development? *IEEE Software*, 21(1):28–33, 2004.

[25] R. Stallman. What is the Free Software Foundation? *GNU's Bulletin*, 1(1), Feb. 1986.

[26] R. M. Stallman. What is free software. *Free Society: Selected Essays of*, 2002.

[27] K.-J. Stol and M. A. Babar. Reporting empirical research in open source software: the state of practice. In *Open Source Ecosystems: Diverse Communities Interacting*, pages 156–169. Springer, NYC, USA, 2009.

[28] J. Teixeira and A. Baiyere. Crafting a systematic literature review on open-source platforms. In *Open Source Software: Mobile Open Source Technologies*, pages 113–122. Springer, 2014.

[29] J. Teixeira and T. Lin. Collaboration in the open-source arena: The webkit case. In *Proceedings of the 52nd ACM conference on Computers and people research*, pages 121–129. ACM, 2014.

[30] G. von Krogh and S. Spaeth. The open source software phenomenon: Characteristics that promote research. *The Journal of Strategic Information Systems*, 16(3):236–253, 2007.

[31] G. von Krogh and E. Von Hippel. Special issue on open source software development. *Research Policy*, 32(7):1149–1157, 2003.

[32] G. Von Krogh and E. Von Hippel. The promise of research on open source software. *Management science*, 52(7):975–983, 2006.

[33] J. West. How open is open enough?: Melding proprietary and open source platform strategies. *Research policy*, 32(7):1259–1285, 2003.