The application of crowdsourcing and the Bazaar model to the development of library classifications: an assessment of the Open Shelves Classification

A aplicação do crowdsourcing e o modelo de Bazaar no desenvolvimento de classificações de bibliotecas: uma avaliação da Classificação Open Shelves

Daniel MARTÍNEZ-ÁVILA¹ iD 0000-0003-2236-553X

Rosa SAN SEGUNDO¹ iD 0000-0002-1449-8175

Abstract

This paper presents a discussion of crowdsourcing approaches to knowledge organization and more specifically to the development of classification schemes. It analyzes the case of Open Shelves Classification, a terminated project that was developed by the LibraryThing community following the "open source model", and assesses its outcomes from methodological and sociological points of view. Working with all the documentation of the project that is freely available, the text conducts an analysis of the project following the structure of the methodological lessons for open source (Bazaar model) presented by Eric Raymond in his seminal work "The Cathedral and the Bazaar" and complementing it with the discussion of the sociological aspects presented in "Homesteading the Noosphere" and other writings. The paper concludes with some recommendations for the success of open source projects and some possibilities of research for crowdsourcing projects and social epistemology.

Keywords: Classification. Crowdsourcing. Knowledge organization. Open shelves classification. Open source.

Resumo

O trabalho apresenta uma discussão sobre as abordagens de crowdsourcing *para organização do conhecimento e, mais especificamente, para o desenvolvimento de sistemas de classificação. Além de analisar o caso da* Open Shelves Classification, *um projeto finalizado e desenvolvido pela comunidade* LibraryThing *seguindo o "modelo de código aberto", e avaliar o resultado de pontos de vista metodológicos e sociológicos, o artigo trabalha com a documentação do projeto disponível, realiza uma análise do projeto seguindo a estrutura dos pontos metodológicos para projetos de código aberto (modelo Bazar) apresentados por Eric Raymond em seu trabalho seminal "A Catedral e o Bazaar" e complementa com a discussão dos aspectos sociológicos apresentados em "*Homesteading the Noosphere*" e outras literaturas. A conclusão deste trabalho inclui recomendações para o sucesso de projetos de código aberto e algumas possibilidades de pesquisa para projetos de* crowdsourcing *e epistemologia social.*

Palavras-chave: *Classificação.* Crowdsourcing. *Organização do conhecimento. Classificação* open shelves. *Código aberto.*

Introduction

Crowdsourcing is a "broad term referring to the collaboration of large numbers of distributed people on a common product, usually organized through information technology and generally invited through an open call" (Organisciak; Twidale, 2015). Jeff Howe, the journalist who coined the term in 2006, also said that crowdsourcing is "the application of Open Source principles to fields outside of software" (Howe, 2006, unpaginated). In his definition, Howe also emphasized the necessity of an open call. At the Annual Meeting of the Association for Information Science and Technology 2016 meeting in Copenhagen, Denmark, several scholars from different fields (Computer Science, Information Science, Economics, Communication Studies, Cultural Heritage) and from different countries organized a panel discussion on Crowdsourcing Approaches for Knowledge Organization Systems (Zhitomirsky-Geffet *et al.*, 2016). They discussed issues such as power (with mentions to Foucault's panopticon), control, trust, inter-contributor consensus, and others. Although they mainly focused on the construction of ontologies, the same arguments could be applied to the development of classification systems.

In fact, in July 2008, Tim Spalding, the owner of LibraryThing, called the community "to help build the Open Shelves Classification (OSC), "a free, 'humble' modern, open-source, crowd-sourced replacement for the Dewey Decimal System" (Spalding, 2008, online). A further explanation of the system revealed that the OSC would be collaboratively written and collaboratively assigned; it would also include progressive development, public-library focus, and statistical testing. The project was presented at the "2009 ALA Midwinter Meeting" (Spalding, 2009), the "Public Library Association blog" (Hill, 2009), and the "IFLA Classification and Indexing Section Newsletter". However, in 2010 the project was declared dormant (Spalding, 2010). The research question of this paper is: What went wrong from the methodological point of view of open source design? The paper aims to assess the OSC as a crowdsourcing project and to investigate the reasons behind this failure.

Characterization of the Open Shelves Classification as a crowdsourcing project

Following a taxonomy of crowdsourcing projects (according to their motivation, centrality, type of aggregation, director/beneficiary, type of work, and type of crowd) (Organisciak; Twidale, 2015), the OSC can be characterized as a project in which the contributors basically follow an intrinsic motivation, as they are volunteers with a primary motivator of interest in the topic and secondary motivators of reputation and feedback, as well as impression of change. As for its centrality, the OSC is a core crowdsourcing as the task involves the main goal of the project, *i.e.*, the development of the system. The type of aggregation of the project is iterative: contributions are used to develop a larger product, the classification system, and are permutations of a common work. Regarding the director of the crowdsourcing activities and beneficiary of the contributions, it is a mixture of sponsored and autonomous crowdsourcing. It is sponsored because there is an entity at the top soliciting the contributions: LibraryThing (and we should not forget that several companies were buying shares of LibraryThing at the time of the OSC announcement; AbeBooks had bought a 40% share of LibraryThing in 2006 and was acquired by Amazon. com in August 2008; in January 2009, Cambridge Information Group bought a minority share of LibraryThing, and their subsidiary Bowker became the official distributor to libraries).

On the other hand, it can also be considered an autonomous crowdsourcing because it serves the community itself, the community of LibraryThing users. It can be said that both the director and contributors benefit in a symbiosis. As for the type of work, it is generative because it develops a new intellectual product, a library classification; it is creative because it is the product of critical thinking; and it can also be considered subjective if we accept that the development of classification systems is not a neutral and value-free process. As for the type of crowd, it is internal as it is composed of contributors from the organization (the LibraryThing community), but also external because at some point it included external contributors. It is specialized because the contributors

were assumed to have some knowledge on classification and it is diverse because it included contributors from different countries and with different cultural backgrounds. In the announcement, the OSC expressed concerns about the copyright nature of the Dewey Decimal Classification, stating that the system should be free to use and change, as well as collaboratively written, reflecting some of the key concepts of the free software and open source philosophies.

The Bazaar's model as a methodology for crowdsourcing

Although originally it was not called free software, sharing code and sharing software was part of the history of computers, and was actually the way software was developed. However, in the early 1980s, companies started to hire the most prominent hackers from university laboratories and required them to sign nondisclosure agreements. Software was also distributed in binary form, marking the end of the culture of sharing code due to copyright restrictions. One should note there are several narratives of this history, and the one we follow in this text is Richard Stallman's "Free as in Freedom" and "Free Software, Free Society" (Stallman, 2012; 2015). Other authors such as Steven Levy (2010) in "Hackers" also recount these events. In 1983, Richard Stallman took an ethical decision and started the project of writing a free operating system compatible with Unix, called GNU (GNU is Not Unix). He left Massachusetts Institute of Technology (MIT) to be able to distribute it as free software, so they could not claim the work or impose their distribution terms: ("Leaving MIT as necessary so that MIT would not be able to interfere with distributing GNU as free software. If I had remained on the staff, MIT could have claimed to own the work, and could have imposed their own distribution terms, or even turned the work into a proprietary software package") Stallman 2015, p. 13).

Between 1984 and 1985 he finished the GNU Emacs text editor and released it as free software. According to Stallman, a program is free software if the program's users have the four essential freedoms: (a) The freedom to run the program as you wish, for any purpose (freedom 0); (b) The freedom to study how the program works and change it so it does your computing as you wish (freedom 1), for which access to the source code is a precondition; (c) The freedom to redistribute copies so you can help your neighbor (freedom 2); (d) The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes (access to the source code is a precondition for this outcome). To guarantee these freedoms, Richard Stallman came up with the concept of copyleft and a legal device in the form of a license: the General Public License (GPL), a contract based on copyright that states that some of the rights of the author are surrendered to the users and it guarantees that modified versions of the program are also free software. Copyleft "(very simply stated) is the rule that when redistributing the program, you cannot add restrictions to deny other people the central freedoms" (Stallman 2015, p. 5). Richard Stallman's concept of copyleft inspired Lawrence Lessig and the Creative Commons licenses. Indeed, Lawrence Lessig wrote the introduction to the first edition of Richard Stallman's book "Free software, free society".

The methodological implication of free software is that if anybody can use, study, change, and redistribute the code without legal restrictions, everybody could also use all the contributions in a single project. This is what Linus Torvalds did in the development of the kernel Linux (Torvalds; Diamonds, 2002). Eric Raymond realized that Torvalds' success had more to do with his methodology than any other aspect, and he studied the fundaments of the development of the Linux kernel in order to formulate a theoretical background for this revolutionary model in open source software engineering. The results of his work were formulated in 19 points in the now classic "The Cathedral and the Bazaar" (1997-2001). Although it was both criticized (Bezroukov, 1999; Kamp, 2012) and considered irrelevant from an ideological point of view (Stallman, 2015), "The Cathedral and the Bazaar" remains one of the most important works for open source software engineering and a crucial influence for many projects.

From a system design perspective, the "Bazaar model" corresponds to "an evolutionary model" (Brooks, 2010, p. 56), in opposition to other design models such as the rational model, the co-evolution model, the whirligig model, and the spiral model. The Bazaar model has also been described as a bottom-up model as opposed to

the predominant top-down "cathedral model" of development. Top-down design in programming was originally formalized by Niklaus Wirth (1971) and other seminal contributions to the topic include the works of Mills (1971) and Baker (1972). Establishing analogies between software engineering processes and architectural methods, Raymond (2001) also describes the cathedral method as a complex of individual developers working on a magnificent piece that will not be released or publicly evaluated for rework until it is totally finished. However, according to Deleuze and Guattari's (1988) explanation on how Gothic cathedrals were built by "compagnonnages", a cathedral might not be a correct example of rigid top down design, and that is why authors such as Miquel Vidal (2000) proposed that more accurate names would be "pyramid model" or "skyscraper model".

In the Bazaar model, anyone can contribute while following different agendas and approaches – as Raymond states "delegate everything you can, be open to the point of promiscuity". As the number of contributors grows, it implies on a revision of Brooks's Law that stated that as the number of programmers N rises, the work performed rises in the same proportion as N, linearly, but complexity and vulnerability to bugs rises in the proportion of N2 (Brooks, 1995). Linus Torvalds indicated that although this law was not completely false, it was not necessarily true either. Raymond himself does not claim that Brooks's Law contradicts his findings, which is explainable in combination with Hasler's Law: the costs of duplicated work tend to scale sub-quadratically with team size (Raymond, 2001, p. 220). However, some of these problems would also be overcome with Linus's Law, as Raymond (2001, p. 30) dubbed it: "Given enough eyeballs, all bugs are shallow".

The sociological aspects of the open source development model were explained in "Homesteading the noosphere" (Raymond, 2001). There, Raymond talks about a gift culture that can be related to the work of French sociologist Marcel Mauss (Mauss, 2002) (e.g., Bergquist; Ljungberg, 2001). He also talks about a reputation game and a peer review that have been likened to the academia and the work of Robert Merton (1973) (*e.g.*, Kelty 2010). Eric Raymond also dedicates one section of the work to the link of open source and the academia, in which he comments that one of the main differences between scholars and hackers is that criticism of their works in harsher to take for scholars.

As González-Barahona (2004a) suggested, the Bazaar model's methodology should not be applied exclusively for the development of free software, but also to many other environments, mainly those related to information processing and knowledge building, where public availability of internal information can be useful. That is not a new idea nor has it been proven unsuccessful, the most obvious and famous example being "Wikipedia, The Free Encyclopedia" (O'Reilly, 2005; Poe, 2006). In his entry on "Social Epistemology" for the Encyclopedia of Library and Information Sciences, Steve Fuller (2010) described Wikipedia as an example of "cutting-edge" social epistemology. Don Fallis (2008), in his study of Wikipedia's social epistemology, also characterized it as "essentially an open source project". González-Barahona also proposed other fields of application for the Bazaar model of development, such as dictionaries (González-Barahona, 2004c) and models of education (González-Barahona, 2004b), based on MIT's Open Course Ware initiative. Pekka Himanen (2001) also discussed the application of the Bazaar model to education calling it "The Net Academy". Other applications of the Bazaar model have been discussed in movies (Matellán-Olivera, 2004), research (Lemire, 2010; Gardler, 2013), and even government (Tkacz, 2013).

Eric Raymond has also shown concerns regarding the application of the Bazaar model to contexts other than software development, stating that:

> I am often asked if I believe the open-source model can be usefully applied to other kinds of goods than software […] A case in point; music and most books are not like software, because they don't generally need to be debugged or maintained. Without that requirement, the utility of peer review is much lower, and the rational incentives for some equivalent of open-sourcing therefore nearly vanish (Raymond 2001, p. 193).

However, debugging and maintenance are important for classification systems. Richard Stallman spoke about applying the free software model to other contexts, saying: "Software manuals must be free, for the same

reasons that software must be free […] The same arguments also make sense for other kinds of works of practical use – that is to say, works that embody useful knowledge, such as educational works and reference works. Wikipedia is the best-known example" (Stallman, 2015, p. 7) Arguably, this applies to classification systems too.

Regarding knowledge organization, Hope Olson and Dennis Ward stated that:

> The open source model is one likely to be very beneficial for development of new or modified subject access standards [*i.e., library classification schemes*]. Like the computing community, the library world has many practitioners who are both highly skilled and committed to their profession. One could envision cooperative efforts to build access for marginalized groups, and perhaps even more general standards. These standards would be controlled by communities and not by institutions focused on designs for compliance (Olson; Ward 2003, p. 55, italics by the author).

This might be considered an extension of what Himanen (2001) called the "Nethic," a hacker's drive which is defined by the values of activity and caring:

> Caring here means concern for others as an end in itself and a desire to rid the network society of the survival mentality that so easily results from its logic. This includes the goal of getting everybody to participate in the network and to benefit from it, to feel responsible for longer-term consequences of the network society, and to directly help those who have been left on the margins of survival (Himanen, 2001, p. 141).

Therefore, the OSC idea seems to be supported by both the classificationists and the hackers points of view. But did the OSC follow the proper method for an open source project as described by Eric Raymond?

Methodological Procedures

The present paper reports a thorough analysis of all the documentation of the OSC project (including forums, blogs, and websites) and the assessment of its outcome. Most statements are taken from the different contributions to the Open Shelves Classification project forum (LibraryThing, 2011) and related sources. Due to space constraints, not all URLs have been included here and they will be readily made available upon request. The paper discusses the application of the open source Bazaar model to this classification system, following the nineteen lessons of Raymond's "The Cathedral and the Bazaar" (2001), which can be seen as principles of open source development, as well as the sociological analysis of "Homesteading the Noosphere" and other writings.

Eric Raymond's "The Cathedral and the Bazaar" (2001) is composed of 19 guidelines that describe the Bazaar model and that had been followed in his previous Fetchmail project. These lessons intend to illustrate the aspects that made other open source projects successful in the past, as well as to serve as recommendations for future open source projects. The OSC project is assessed and discussed in this paper following the structure of these lessons.

Results and Discussion

Assessment of the OSC project following the Bazaar model guidelines

The first of Raymond's lesson is that: "Every good work of software starts by scratching a developer's personal itch." Raymond also states that "necessity is the mother of invention". In this sense, the OSC would be a response to Tim Spalding's personal itch, as it was necessary for LibraryThing given the inadequacy of the Dewey Decimal

Classification (DDC) in the following aspects: (1) it represented outdated thinking; and, more importantly, (2) it was a proprietary scheme that required paid licensing to use. The motivation of such criticism of the DDC, it seems to be theoretically justified according to the literature on critical knowledge organization over the last 40 years. Different studies have pointed out system inadequacies within each version of the DDC and related them to the prejudices of their times and of the DDC editors, which represent mainstream views and marginalize other communities (Foskett, 1971; 1984; Olson, 1998; 2001a; 2001b; 2002; 2007; Kublik *et al.*, 2003; Martínez-Ávila; Semidão; Ferreira, 2016; Mai, 2016). In accordance to the ethos of open source software, the DDC has also been criticized for its copyrighted proprietary nature (Ardito, 2003; Intner, 2004).

Raymond's second lesson is: "Good programmers know what to write. Great ones know what to rewrite (and reuse)." In the original/free tradition of software development, code reuse was a fundamental part of the project's progress. As Raymond pointed out:

> It's fairly clear that one cannot code from the ground up in Bazaar style. One can test, debug and improve in Bazaar style, but it would be very hard to originate a project in Bazaar mode. Linus didn't try it. I didn't either. Your nascent developer community needs to have something runnable and testable to play with (Raymond, 2001, p. 47).

Linus Torvalds reused ideas and code from Minix when developing Linux, Eric Raymond looked for an existing Post Office Protocol utility (fetchpop) as a development base for Fetchmail. However, the main difference between "reusing code" in a software project and in a library classification scheme is the way the "code" is made available. Although protected by copyright, the tables of DDC or of any other library system can be easily consulted. The legal limitation is on its use, modification, and redistribution. Thus, the question is if any system was used as a development base for the OSC, either as a theoretical influence for the structure or as a warrant to populate the classes. There was a section in the project's wiki that listed existing base schedules and included the Dewey Decimal Classification and the Universal Decimal Classification, Cutter Expansive Classification, Book Industry Standards and Communications (BISAC), and the Broad System of Ordering. Although they were not listed in the wiki, other systems mentioned in the forums were the British Book Industry Communication and Ranganathan's Colon Classification. As for the DDC, although the OSC was a response to its copyright nature, it was initially based upon its structure. This was later discarded in favor of a BISAC structure, a major reason for discontentment within the community and finally adopted for the new project of the Melvil Decimal System, once the OSC project was terminated.

The third lesson is: "Plan to throw one away; you will, anyhow." This is a concept is taken from Brooks's (1995) "The Mythical Man-Month". Raymond (2001, p. 25) stated: "you often don't really understand the problem until after the first time you implement a solution. The second time, maybe you know enough to do it right. So if you want to get it right, be ready to start over at least once". As for the OSC, there were several attempts to reboot the project, which was basically due to problems related to leadership. However, the classification system or the schedules were not enlisted to be thrown away, and when they were, the solution was a totally different project: The Melvil Decimal System.

The fourth lesson is: "If you have the right attitude, interesting problems will find you". Similarly, the collaborative attitude and philosophy of the LibraryThing community was one of its greatest assets. In this sense, problems related to copyright and the Dewey Decimal Classification found the LibraryThing community and hence the necessity of developing the Open Shelves Classification.

The next lesson is: "When you lose interest in a program, your last duty to it is to hand it off to a competent successor." Although it would be unfair to claim that Tim Spalding lost interest in the OSC, he did look for other leaders in the open call. As a founder of LibraryThing, his need to delegate leadership of this small project was quite understandable. In his call for successors, Tim Spalding said that he favored librarians with knowledge on cataloging and experience with the Book Industry Study Group, the institution in charge of BISAC. In August 2008,

two successors were appointed. One of them was Laena McCarthy, former Assistant Professor and Image Cataloger at the Pratt Institute in New York City, and the other one was David Conners, Digital Collections Librarian at Haverford College in Pennsylvania.

They were presented as facilitators. Even as Spalding was no longer officially in charge of the project, he never stopped contributing to the forums until the spring of 2009, when he bowed out. At this point, McCarthy took primary leadership, and between four and eight Pratt School of Information and Library Science graduate students started collaborating on the project. With the entrance of the students, Spalding, McCarthy, and Conners virtually disappeared from the forums. At this point, perhaps the community felt confused about leadership and in June 2009, David Conners called for leaders again on the OSC blog, which the community perceived as yet another problem of leadership and communication. Because the blog was not the usual channel of communication with the community, it gave the impression that they were looking for leaders outside the LibraryThing community. In September 2010, a forum user called Lunaphiles (probably a troll) appeared and self-appointed himself as the new project leader. He harshly criticized the open source philosophy as one of the main reasons for the project's failure and threatened the whole community with copyrighting their work. Lunaphiles new leadership became an authoritarian parody and was quickly rejected by the community. Spalding had to intervene, discrediting the new leadership and inviting the user to fork the dormant project. The user, in effect, created a new group that nobody joined.

Following Eric Raymond's sociological analysis of "Homesteading the Noosphere" we see that there are several aspects in this case deserving of further study. First, Spalding's leadership, which was well accepted by the community, followed what Raymond calls the benevolent-dictator model: "Typically, a benevolent-dictator organization evolves from an owner-maintainer organization as the founder attracts contributors" (Raymond, 2001, p. 101). Indeed, according to Raymond's conclusions, this model is the best for open source projects: "The culture's (and my own) understanding of large projects that don't follow a benevolent-dictator model is weak" (Raymond, 2001, p. 111). The relationship between the benevolent-dictator model and open source development was also pointed out by some comments in the forum. On the other hand, when McCarthy took over the leadership, technically, she and David Conners took the leadership over jointly, but, due to Conners's low participation, McCarthy was the de *facto* and only perceived leader, her model was much more authoritarian and uncommunicative than Spalding's, which was never well accepted by the community and determined the project's decline. According to Raymond, there are several typical ways to solve such problems in open source projects when they come up:

> Some very large projects discard the 'benevolent dictator' model entirely. One way to do this is turn the co-developers into a voting committee (as with Apache). Another [solution] is rotating dictatorship, in which control is occasionally passed from one member to another within a circle of senior co-developers; the Perl developers organize themselves this way (Raymond, 2001, p. 102).

Although a voting system was proposed by the community, and agreed to by McCarthy, it was never put into practice. Lunaphiles's leadership was never accepted by the community. Second, the way the user Lunaphiles announced the new ownership after an interval of inactivity might be considered a legitimate way to assume ownership in an open source project (Raymond, 2001, p. 75). However, this method is only considered legitimate if there are no objections among the community, and this was not the case. On this point, Spalding was right to censor him/her as a leader.

The next of Raymond's lesson says: "Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging". LibraryThing members are also contributors to social cataloging, so it can be said that they "debug" records. However, the community did not feel like co-developers except in the beginning of the process and although they constantly wanted to. This confusion about community status was

caused mainly by a redundancy of information channels, leadership communication problems, and a top-down hierarchical organization that gave the "lower classes" the impression that they were unimportant and unnecessary, which is contrary to open-source philosophy. Developers felt their contributions were not taken into consideration and they also complained about not having access to the data. The "code" was kept away in a Google document in order to prevent the wiki from becoming an "edit war". The existence of several uncoordinated channels – forums, the OSC blog, the "ClassifyMe" blog started by the students and only open to invited readers, the "LibraryThing Thing-ology" blog, Google documents, *etc.* – and different stratified profiles for editing discouraged collaboration and dissolved the co-developers' sense of community, causing dissatisfaction. In this sense, Raymond argued in favor of rejecting some of the traditional Brooks's roles such as the "code librarian" in open source projects as he suggests that when everybody can keep and edit the code, this generates a higher degree of involvement (Raymond, 2001, p. 221).

The seventh lesson is: "Release early. Release often. And listen to your customers". Here, Raymond states that "Linus was keeping his hacker/users constantly stimulated and rewarded – stimulated by the prospect of having an ego-satisfying piece of the action, rewarded by the sight of constant (even daily) improvement in their work". (Raymond, 2001, p. 30). For Torvalds, having the maximum ratio of person-hours on debugging and development was an even higher priority than the possible cost of instability in the code and user-base burnout if any serious bug proved intractable. However, rewards and stimulation through frequent releases and collaborator satisfaction was never done in the OSC project. The community felt that the releases were delayed and their suggestions were not considered.

The following lesson says: "Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone." Here Raymond introduces his probable most famous contribution, dubbed as Linus's Law: "Given enough eyeballs, all bugs are shallow," or in other words, every problem "will be transparent to somebody" (Raymond, 2001, p.32). This law has been rephrased by Jeff Dutky as "Debugging is parallelizable," Raymond, 2001, p. 226. The philosophy behind this principle is similar to the "Delphi effect" in organizations, which applied to software development means that if a large community of developers is involved in a project, when a problem arises, each member can contribute their best, thus making individual efforts less tedious, eliminating redundancy, and achieving overall results that are faster and better for the project. According to Raymond (2001, p. 31), this is "the core difference underlying the cathedral-builder and Bazaar styles".

Here the question is if "debugging" is easier for the OSC than for other classification systems. Each system has participatory devices in which the community can detect and suggest modifications. For instance, the Dewey Decimal Classification has a section in which the editors post papers discussing specific subject areas and give instructions for users to post comments (OCLC, 2020). This is a way in which the community can provide feedback in topics considered for revision. In the case of the Universal Decimal Classification, the UDC Consortium organizes biennial conferences in which research on classification is presented and it is said that it enables communication between developers and users (UDC Consortium, 2020). These conferences are similar to any other academic conference. In the case of BISAC, they consider requests from Book Industry Study Group members and 'the industry' (Martínez-Ávila, 2016), which, in practice, is the least participative and research-based revision process. In this vein, Birger Hjørland commented on the relation between research and the revision of classification systems:

> The practical work of maintaining and updating a given system (such as the UDC) is not, however, research in itself, but should be based on research. What could be recognized as research would be the publication of articles which carefully argue for the specific decisions made, considering empirical, logical, historical and pragmatic issues (Hjørland, 2012, p. 302).

For collaboratively-developed systems, the following comment on cumulative effort is also valid:

The possibility of a cumulative effort is related to how "arbitrary" decisions of classifications are. If one system, say, UDC, makes an arbitrary decision to place subject X under subject Y, then there is no reason why other systems, say DDC, should make use of the work done. If, on the other hand, the UDC based its decision on a mixture of empirical and theoretical considerations, and made good arguments for a specific solution, then there is a much better chance that such work can be reused for other purposes and that more cumulative results appeared (Hjørland, 2007, p. 10).

This idea is especially relevant in the context of social epistemology and aspects such as justification and authority.

The next lesson is: "Smart data structures and dumb code works a lot better than the other way around." Paraphrasing Brooks, "Show me your flowchart and conceal your tables, and I shall continue to be mystified. Show me your tables, and I won't usually need your flowchart; it'll be obvious" (Brooks, 1995, p. 102). In a classification system, the equivalent to data structures would be their analytic-synthetic features, auxiliaries, etc. There were no clear tables in the OSC during the first phase of the project (since the "top level categories" discussion lasted forever, with over 300 posts). On the other hand, the BISAC tables adopted as a basis for the second phase might arguably be considered far from smart. In addition, the long discussions about the hypothetical structure of the scheme did not reach any clear conclusions either. Thus, this point was not followed.

The tenth lesson says: "If you treat your beta-testers as if they're your most valuable resource, they will respond by becoming your most valuable resource." However, there were no beta-testers for the scheme since no libraries agreed to adopt the scheme in its early stages nor was it tested in the LibraryThing records.

The eleventh lesson is: "The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better". Here, Raymond (2001, p. 40) says that if you are humble about how much you owe to the contributions of others you will get a lot of recognition yourself. He uses the example of Linus Torvalds. However, as mentioned before, the perception among the OSC developers was that their ideas were not recognized or taken into consideration. Extrapolating this guideline to other library classification revision processes (and concerning the acknowledged problem of biases), the need for recognition and collaboration between editors and collaborators was a topic covered by Hope Olson who said:

We can not simply say that Dewey's pigeonholes have problems and tell the editors of the DDC to fix them. [...] Everybody must take some responsibility. Solutions would be more difficult if the DDC establishment were not receptive. The editors of the DDC and of other classifications regularly respond to concerns about bias at the same time that they endeavor to make their changes manageable for existing collections. However it is also crucial that individual librarians, libraries, associations, and researchers take responsibility. We need to look at the diversity of groups using libraries and apply a range of optional, partial, and local solutions. If we fail to take on the task of making materials accessible to different groups and cultures and well-represented to users outside of those cultures, then we are complicitous in the failures of our classifications (Olson 2001a, p. 121).

The next lesson says: "Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong". Raymond (2001, p. 40) states: "When you hit a wall in development [...] Perhaps the problem needs to be reframed". In this sense, the initial problem of the project was the inadequacy of the DDC, especially in relation to copyright. The initial solution was based on the use of a system alternative to the Dewey Decimal Classification such as BISAC. However, BISAC is even more biased and equally copyrighted. Once the project failed, the second and most innovative solution came in the form of the Melvil Decimal System (a free modification of the 1922 version of the Dewey Decimal Classification). However, this should not be considered a

part of the Open Shelves Classification project, but an independent and successful project by the LibraryThing community that started after the failure of the OSC project and followed a different philosophy and methodology.

The next lesson is: "Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away". This quotation is attributed to Antoine de Saint-Exupéry (Raymond 2001, p. 41), a writer, aviator, and aircraft designer. Basically, this point is a re-formulation of the KISS (Keep It Simple, Stupid) design principle that requires keeping it short and simple, and as Raymond expressed, "when your code is getting both better and simpler, that is when you know it's right" (Raymond, 2001, p. 42). It cannot be said that this principle was followed in the case of the OSC.

The next lesson is "Any tool should be useful in the expected way, but a truly great tool lends itself to uses you never expected". Since the scope of the classification was never clear, the concept of its expected usefulness could not be either. However, the OSC project seemed open and loose enough to be tested in any final application, although it was never acknowledged that the system had been used in any real way, making it impossible to evaluate its performance in different scenarios. The statistical testing concept announced in the presentation was never clear either, as it was only composed of some undeveloped ideas about lumpiness and overlapping with other systems.

The next lesson is: "When writing gateway software of any kind, take pains to disturb the data stream as little as possible – and never throw away information unless the recipient forces you to". If we assume that a classification scheme could be considered a gateway between users and information, then it would also be interesting to pay attention to what Raymond said concerning the Fetchmail project:

> Some European users bugged me into adding an option to limit the number of messages retrieved per session (so they can control costs from their expensive phone networks). I resisted this for a long time, and I'm still not entirely happy about it. But if you're writing for the world, you have to listen to your customers – this doesn't change just because they're not paying you in money (Raymond, 2001, p. 45).

Raymond introduces the concept of listening to all the communities that use the system, even if the final result suffers from a decrease in the overall quality. This reasoning is also in consonance with the ethical philosophy of Stallman's free software movement: freedom over technical superiority.

On the other hand, the key question here is: Is universality possible in classification? This is an important topic in knowledge organization that is also related to social epistemology. The "listen to the world" element that Raymond talks about does not necessarily mean a universalist view, but as Hope Olson claims, "to look toward solutions, rejecting the idea of universal solutions" (Olson, 2001a, p. 116), or, applied to theoretical aspects of classifications, "to include models that are radically different and to allow multiple models to coexist – separately or layered or even integrated with each other" (Olson, 2007, p. 522). Along these lines, several researchers with varying points of view consider the rejection of universality as positive for the communities (Martínez-Ávila *et al.*, 2015; Martínez-Ávila; Beak, 2016; Montenegro, 2019). Concerning the OSC project, several contributors also drew attention to the non-universality of the scheme and its US-centric bias. One example was British user Andy Leighton who complained repeatedly about the inappropriateness, outside of the US, of the term "African-American" as a synonym for "black". However, despite his efforts, Leighton's arguments and suggestions were systematically ignored by the coordinators.

The sixteenth lesson is "When your language is nowhere near Turing-complete, syntactic sugar can be your friend" (Raymond 2001, p. 46). To make an analogy with library marketing techniques, when confusion swamped the project during its first phase, OSC coordinators started looking at commercial-driven BISAC as a solution. As previously stated, the influence of BISAC's Dewey-free project on McCarthy was clear from the beginning. However,

at some point, even Spalding started to justify BISAC's "syntactic sugar" as a positive point for the project. The fact that libraries adopting BISAC perceived it as a simpler, more intuitive, and user-friendly system, can easily be considered the classificationist equivalent to the concept of "syntactic sugar". "In computer science, syntactic sugar is syntax within a programming language that is designed to make things easier to read or to express. It makes the language "sweeter" for human use: things can be expressed more clearly, more concisely, or in an alternative style that some may prefer". (Wikipedia, 2020, unpaginated).

The seventeenth lesson is: "A security system is only as secure as its secret. Beware of pseudo-secrets". In the beginning, there was no technical security concern in the OSC project. Regarding data integrity, the leaders claimed that the system code was not available to everybody in order to prevent "edit wars" and disorganization. This latter point, together with the communication problems and the contributions never taken into consideration, was perceived as a sign of disrespect to the community.

The next lesson is: "To solve an interesting problem, start by finding a problem that is interesting to you". Raymond (2001, p. 49) adds that "the best hacks start out as personal solutions to the author's everyday problems, and spread because the problem turns out to be typical for a large class of users". This has to do with the social context of open source and arguably also with social epistemology (Fuller 2010, p. 1). As for the OSC, while the community found many problems interesting, the project leaders seemed to lose interest, probably due to burnout. This situation meant that most problems went unsolved.

Finally, the last lesson is: "Provided the development coordinator has a medium at least as good as the Internet, and knows how to lead without coercion, many heads are inevitably better than one". Raymond (2001, p. 50) adds that "the really great hacks come from harnessing the attention and brainpower of entire communities". This is related to the discourse of the wisdom of the crowds. Raymond (2001, p. 52) also cites Russian anarchist Peter Kropotkin in relation to leadership and success: "the aim can be achieved only through the severe effort of many converging wills". In this sense, the open source development has been compared not only to anarchism but also to a free market, with analogies to science and the work of Robert Merton.

Conclusion

According to the seminal works on open source, key aspects for the success of a project are: (1) Legal aspects in the form of licensing to ensure the development of the project without limitations to studying, modifying, and redistributing code; (2) Communication tools, such as the internet, to allow fluid communications between leaders and the community; (3) A good community of developers that, of course, must be interested in the project and also able to contribute with meaningful knowledge; (4) A leader with good communication skills and who is able to recognize good design ideas by others; (5) A plausible promise at the announcement, including something runnable and testable so developers do not get frustrated. This is why is so important for an open source project to reuse code from previous projects instead of starting from scratch. In the case of the Open Shelves Classification, as we have seen, it did not meet the last two requirements.

In addition, some of the main problems detected in this analysis were: (1) An absence of clear goals for the system and project, including a lack of theoretical background on free software/open source principles and on pragmatist epistemology on the part of the leaders, that weakened the project and the entire community vision; (2) Inappropriate leadership, especially during the project's second phase, following an erroneous communication model and not adopting certain correct measures, even when they were proposed by the community; (3) Excessive slowness in the development process and releases, since every decision seemed to be overly discussed on the forums, and yet, the community's decisions were not taken into consideration by the coordinators. This problem caused discontentment among community members who did not see their contributions and reputation rewarded.

Finally; (4) No reuse of code in the beginning and a derivation to the BISAC scheme in the second stage. While the initial intention was to build the system from scratch, the whole OSC project derived to BISAC when the new leaders took over. This decision was rejected by the community, who had been told during first phase that BISAC was not a viable alternative to DDC because it was another proprietary system. At this point, the evident lack of goals took the project back to the initial problem, proving the OSC project had been unviable from its conception, and setting it into an indefinite state of dormancy.

However, the idea of a library classification based on open source methodology could have been successful if a different methodology had been properly followed. Indeed, the social and participative approach to knowledge organization has been identified as one of the key aspects of successful projects (Martínez-Ávila, 2015). Some further considerations for a library classification system following this model are: the rejection of universality as an assumption, accepting that a single system with the same code cannot be valid for everybody in the same way; the adoption of a more pragmatist view for system development, having a clear vision of which communities will use the system and the ways in which it could be useful for them; and, finally, the establishment of appropriate ways to develop based on the research and specific theoretical background of the domains of those communities. In the end, if a system is developed by the users, it must serve those users and allow them the freedom to decide the context for its use instead of adopting or allowing a third-party developer to base the system development on market warrant (Martínez-Ávila, 2016; Martínez-Ávila; Budd, 2017; Barité, 2018) or the developer's own interests. In this vein, the open source methodology of the OSC could be approached as an example of social epistemology in Knowledge Organization. Social epistemology looks at how social processes lead to the acquisition of knowledge (Shera, 1970; Goldman, 1999; Fallis, 2008) and Open Shelves Classification was a social system to create and acquire knowledge. In that sense, open questions for future research related to social epistemology (in its different views) in open source projects are: Who to trust when there are disagreements between contributors (based upon reliabilist principles)? What is the democratic role of the leader as a social keeper and facilitator of knowledge? While the first question relates to Alvin Goldman's kind of social epistemology, *i.e.*, what is true, the second question would relate more to Steve Fuller's position.

Contributors

D. MARTINEZ-AVILA contributed to the conception of the manuscript, data collection and analysis, and manuscript writing, revision, and approval. R. SAN SEGUNDO contributed to manuscript revision and approval.

References

Ardito, S. C. OCLC V: the Library Hotel. *Information Today*, v. 22, n. 11, p. 17-20, 2003.

Baker, F. T. *System quality through structured programming*. Gaithersburg: IBM corporation, 1972. Available from: https://dl.acm.org/doi/pdf/10.1145/147992.1480037. Accessed: Feb. 19 2020.

Barité, M. Literary warrant. *Knowledge Organization*, v. 45, n. 6, p. 517-536, 2018.

Bergquist, M.; Ljungberg, J. The power of gifts: organizing social relationships in open source communities. *Information Systems Journal*, v. 11, n. 4, p. 305-320, 2001.

Bezroukov, N. A second look at the Cathedral and Bazaar. *First Monday*, v. 4, n. 12, 1999. Available from: https://firstmonday.org/article/view/708/618. Accessed: Feb. 19 2020.

Brooks, F.P. *The mythical man-month*: essays on software engineering anniversary edition. Boston: Addison-Wesley, 1995.

Brooks, F. P. *The design of design*: essays from a computer scientist. Upper Saddle River: Addison-Wesley, 2010.

Deleuze, G.; Guattari, F. *A thousand plateaus*. London: Athlone, 1988.

Fallis, D. Toward an epistemology of wikipedia. *Journal of the American Society for Information Science and Technology*, v. 59, n. 10, p. 1662-74, 2008.

Foskett, A. C. Misogynists all: a study in critical classification. *Library Resources & Technical Services*, v. 15, n. 2, p. 117-121, 1970.

Foskett, A. C. Better dead than read: further studies in critical classification. *Library Resources & Technical Services*, v. 28, n. 4, p. 347-359, 1984.

Fuller, S. Social Epistemology. *In*: Bates, M. J.; Maack, M. N. *Encyclopedia of library and information sciences*. 3rd ed. Boca Raton: Taylor & Francis, 2010.

Gardler, R. Better academic research through open source collaboration, a guest post by Ross Gardler. *The Outercurve Foundation*, Wakefield, 2013. Available from: https://web.archive.org/web/20150922025914/http://www.outercurve.org/blog/2013/01/29/Better-Academic-Research-through-Open-Source-Colla/. Accessed: Feb.19 2020.

Goldman, A. I. *Knowledge in a social world*. New York: Oxford University Press, 1999.

González-Barahona, J. M. Con todo al aire. *In*: *Sobre software libre*: *compilación de ensayos sobre software libre*. Móstoles: Universidad Rey Juan Carlos, 2004a, p. 55-59.

González-Barahona, J. M. De cómo el conocimiento puede ser libre. *In*: *Sobre software libre*: compilación de ensayos sobre software libre. Móstoles: Universidad Rey Juan Carlos, 2004b, p. 73-77.

González-Barahona, J. M. El Diccionario de la real academia de la lengua. *In*: *Sobre software libre*: compilación de ensayos sobre software libre. Móstoles, Spain: Universidad Rey Juan Carlos, 2004c, p. 103-108.

Hill, N. New classification system for public libraries? *Plablog*, [*S.l.*], 2009. Available from: http://plablog.org/2009/01/new-classification-system-for-public-libraries.html. Accessed: Feb. 13 2020.

Himanen, P. *The hacker ethic and the spirit of the information age*. New York: Random House Trade, 2001.

Hjørland, B. Arguments for 'The Bibliographical Paradigm': some thoughts inspired by the new English edition of the UDC. *Information Research*, v. 12, n. 4, 2007. Available from: http://InformationR.net/ir/12-4/colis06.html. Accessed: Feb. 20 2020.

Hjørland, B. Is classification necessary after google? *Journal of Documentation*, v. 68, n. 3, p. 299-317, 2012.

Howe, J. Crowdsourcing: a definition. [S.l.]: Crowdsourcing, 2006. Available from: http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html. Accessed: Feb. 13 2020.

Intner, S. S. Contrasting views, conflicting ideas. *Technicalities*, v. 24, n. 1, p. 15-18, 2004.

Kamp, P.-H. A Generation lost in the bazaar: quality happens only when someone is responsible for it. *ACM Queue*, v. 10, n. 8, 2012. Available from: https://queue.acm.org/detail.cfm?id=2349257. Access on: Feb. 20 2020.

Kelty, C. M. Free Software/Free Science. *First Monday*, v. 6, n. 12, 2001. Doi: https://doi.org/10.5210/fm.v6i12.902.

Kublik, A *et al*. Adapting dominant classifications to particular contexts. *Cataloging & Classification Quarterly*, v. 37, n. 1/2, p. 13-31, 2003.

Lemire, D. Open sourcing your software hurts your competitiveness as a researcher? *Lemire blog*, Montreal, 2010. Available from: http://lemire.me/blog/archives/2010/02/10/open-sourcing-your-software-hurts-your-competitiveness-as-a-researcher/. Access on: Feb. 19 2020.

Levy, S. *Hackers*. Sebastopol: O'Reilly Media, 2010.

LibraryThing. Open shelves classification. *LibraryThing*, [*S.l.*], 2011. Available from: https://web.archive.org/web/20110113122944/http://www.librarything.com/wiki/index.php/Open_Shelves_Classification. Access on: Feb. 19 2020.

Mai, J.-E. Marginalization and exclusion: unraveling systemic bias in classification. *Knowledge Organization*, v. 43, n. 5, p. 324-330, 2016.

Martínez-Ávila, D. Knowledge organization in the intersection with information technologies. *Knowledge Organization*, v. 42, n. 7, p. 486-498, 2015.

Martínez-Ávila, D. BISAC: Book Industry Standards and Communications. *Knowledge Organization*, v. 43, n. 8, p. 655-662, 2016.

Martínez-Ávila, D.; Beak, J. Methods, Theoretical Frameworks and Hope for Knowledge Organization. *Knowledge Organization*, v. 43, n. 5, p. 358-366, 2016.

Martínez-Ávila, D.; Budd, J.M. Epistemic warrant for categorizational activities and the development of controlled vocabularies. *Journal of Documentation*, v. 73, n. 4, p. 700-715, 2017. Doi: https://doi.org/10.1108/JD-10-2016-0129.

Martínez-Ávila, D.; Ferreira, M.; Magro, J.L. Aplicación de la Teoría Crítica de Raza en la organización y representación del conocimiento. *Scire*, v. 21, n. 2, p. 27-33, 2015.

Martínez-Ávila, D.; Semidão, R.; Ferreira, M. Methodological Aspects of Critical Theories in Knowledge Organization. *Knowledge Organization*, v. 43, n. 2, p. 118-125, 2016.

Matellán-Olivera, V. Software, mentiras y cintas de vídeo. *In*: Matellán Olivera, V. *et al. Sobre software libre*: compilación de ensayos sobre software libre. Móstoles: Universidad Rey Juan Carlos, 2002. p. 17-20.

Mauss, M. *The gift*: the form and reason for exchange in archaic societies. London: Routledge, 2002.

Merton, R. The Sociology of science: theoretical and empirical investigations. Chicago: University of Chicago Press, 1973.

Mills, H. D. Top-down programming in large systems. *In*: Rustin, R. *Debugging techniques in large systems*. Englewood Cliffs: Prentice-Hall, 1971. p. 339-343.

Montenegro, M. Subverting the universality of metadata standards: the TK labels as a tool to promote Indigenous data sovereignty. *Journal of Documentation*, v. 75, n. 4, p. 731-749, 2019. Doi: http://dx.doi.org/10.1108/JD-08-2018-0124.

OCLC. *Discussion papers*. Dublin: OCLC, 2020. Available from: https://www.oclc.org/en/dewey/discussion.html. Accessed: Feb. 20 2020.

O'Reilly, T. What is web 2.0: design patterns and business models for the next generation of software. *O'Reilly*, Sebastopol, 2005. Available from: http://www.oreilly.com/pub/a//web2/archive/what-is-web-20.html. Accessed: Feb. 20 2020.

Olson, H. A. Mapping beyond dewey's boundaries: constructing classificatory space for marginalized knowledge domains. *Library Trends*, v. 47, n. 2, p. 233-251, 1998.

Olson, H. A. Sameness and difference: a cultural foundation of classification. *Library Resources & Technical Services*, v. 45, n. 3, p. 115-122, 2001a.

Olson, H. A. The power to name: representation in library catalogs. *Signs*: *Journal of Women in Culture and Society*, v. 26, n. 3, p.639-668, 2001b.

Olson, H. A. *The power to name*: *locating the limits of subject representation in libraries*. Dordrecht: Kluwer Academic, 2002.

Olson, H. A. How we construct subjects: a feminist analysis. *Library Trends*, v. 56, n. 2, p. 509-541, 2007.

Olson, H. A.; Ward, D. Mundane standards, everyday technologies, equitable access. *In*: International Federation of Library Associations and Institutions satellite meeting, 2001, Dublin. *Proceedings* [...]. München: K. G. Saur, 2003. p. 50-58.

Organisciak, P.; Twidale, M. Design facets of crowdsourcing. *In*: iConference, Newport Beach, 2015. *Proceedings* […]. Grandville: iSchools Inc., 2015. Available from: http://hdl.handle.net/2142/73640. Accessed: Feb. 20 2020.

Poe, M. The hive. *The Atlantic*, 2006. Available from: https://www.theatlantic.com/magazine/archive/2006/09/the-hive/305118/. Accessed: Feb. 20 2020.

Raymond, E. S. *The Cathedral and the Bazaar*: musings on linux and open source by an accidental revolutionary revised and expanded. Cambridge: O'Reilly, 2001.

Shera, J. *Sociological foundations of librarianship*. New York: Asia Publishing House, 1970.

Spalding, T. Build the open shelves classification. *LibraryThing*, [*S.l.*], 2008. Available from: http://www.librarything.com/thingology/2008/07/build-open-shelves-classification.php. Access on: Feb. 13 2020.

Spalding, T. Open shelves classification: first draft live and at ALA Midwinter, *LibraryThing*, [*S.l.*], 2009. Available from: http://www.librarything.com/blogs/thingology/2009/01/open-shelves-classification-first-draft-live-and-at-ala-midwinter/. Access on: Feb. 13 2020.

Spalding, T. Comment to "1. Lunaphiles the Great". *LibraryThing*, [*S.l.*], 2010. Available from: http://www.librarything.com/topic/99295#2217342. Access on: Feb. 13 2020.

Stallman, R. *Free as in freedom*: Richard Stallman's crusade for free software. Farnham: O'Reilly, 2002.

Stallman, R. *Free software free society*: selected essays of Richard M. Stallman. Boston: Free Software Foundation, 2015.

Tkacz, N. 'Openness' is the new magic word in politics – but should governments really be run like Wikipedia? *Aeon*, 2013. Available from: https://aeon.co/essays/bazaar-the-rise-and-rise-of-open-source-politics. Accessed: Feb. 20 2020.

Torvalds, L.; Diamond, D. *Just for fun*: the story of an accidental revolutionary. New York: Harper Collins, 2002.

UDC Consortium. International UDC Seminars. *UDC Consortium*, Hague, 2002. Available from: http://seminar.udcc.org/. Accessed: Feb. 20 2020.

Vidal, M. Cooperación sin mando: una introducción al software libre. Miguel Vidal. Madrid, ago. 2000. Available from: http://biblioweb.sindominio.net/telematica/softlibre/. Accessed: Feb. 13 2020.

Syntactic sugar. *In*: *Wikipedia*. San Francisco: Wikipedia, 2020. Available from: https://en.wikipedia.org/w/index.php?title=Syntactic_sugar&oldid=939802333. Accessed: Feb. 13 2020.

Wirth, N. Program development by stepwise refinement. *Communications of the ACM*, v. 14, n. 4, p. 221-227, 1971.

Zhitomirsky-Geffet, M. *et al*. T. Crowdsourcing approaches for knowledge organization systems: crowd collaboration or crowd work? *Proceedings of the Association for Information Science and Technology*, v. 53, p. 1-6, 2016. Doi: https://doi.org/10.1002/pra2.2016.14505301013.