What Does an Ontology Engineering Community Look Like? A Systematic Analysis of the schema.org Community

Samantha Kanza[1], Alex Stolz[2], Martin Hepp[2], and Elena Simperl[1]

University of Southampton
sk11g08@soton.ac.uk, e.simperl@soton.ac.uk
Universitaet der Bundeswehr Munich
alex.stolz@unibw.de, mhepp@computer.org

Abstract. We present a systematic analysis of participation and interactions within the community behind schema.org, one of the largest and most relevant ontology engineering projects in recent times. Previous work conducted in this space has focused on ontology collaboration tools, and the roles that different contributors play within these projects. This paper takes a broader view and looks at the entire life cycle of the collaborative process to gain insights into how new functionality is proposed and accepted, and how contributors engage with one another based on real-world data. The analysis resulted in several findings. First, the collaborative ontology engineering roles identified in previous studies with a much stronger link to ontology editors apply to community interaction contexts as well. In the same time, the participation inequality is less pronounced than the $90 - 9 - 1$ rule for Internet communities. In addition, schema.org seems to facilitate a form of collaboration that is friendly towards newcomers, whose concerns receive as much attention from the community as those of their longer-serving peers.

Keywords: collaborative ontology engineering, GitHub, schema.org, community analysis, social computing, mixed methods

1 Introduction

Creating an ontology is a complex process. It requires an understanding of the relevant domain, the technicalities of ontology engineering, and an ability and willingness to collaborate with others, often across disciplinary boundaries, to agree on what the ontology should cover and how. The Semantic Web community has built an impressive repertoire of methodologies, methods and tools to assist in this process [22]. Over a decade after the first influential papers in collaborative ontology engineering were published [18, 24], it is broadly acknowledged that, for ontologies to unfold their benefits and be economically feasible, they must be developed and maintained by a community, using systems that support the technical, social and participatory aspects of the process. Groupwork platforms such as MediaWiki, GitHub and Quora are broadly used for similar tasks

in software development and are increasingly adopted for ontology engineering projects. They enable ontology stakeholders to ask questions, exchange ideas, and discuss modelling decisions; helping the community to form and thrive.

Analysing ontology engineering communities helps us understand how ontologies are built; whom they represent (and whom not); whether the community follows specific processes and if proposed methodologies work; how to improve group performance; and what tool support is needed in specific situations. Several studies in the ontology engineering literature illustrate this, including aspects such as: users collaborative roles [15, 30]; how people use collaborative ontology editors [6, 21, 31]; or what tool features enable collaboration [5, 22]. Our paper contributes to this field of research by analysing the activities and interactions of the schema.org community. Many consider schema.org [8] as one of the most successful collaborative Semantic Web projects of all times, alongside DBpedia, Wikidata and the Linked open Data Cloud. Founded by the four major Web search engines, it is home to a large community that follows an open participatory approach to develop and maintain Web vocabularies used by over 10 million websites[1]. The community is supported by two main tools: a GitHub repository (tracking issues, making vocabulary versions publicly available), and a public mailing list (for day-to-day discussions). The aim of this study is to gain an understanding of the community make-up in terms of topics, contribution types, and engagement levels, using publicly available data from these two platforms.[2] We broke down the analysis into the following aspects:

Topic prevalence : The topics discussed across GitHub and the community-group public mailing-list help us understand whether the platforms are used as the community managers intended and whether additional tools are needed to support specific topic-centric community activities [34].

Popular topics : We define popularity by the level of engagement a topic attracts from the community via comments and replies. These metrics signal areas of interest, which may require better documentation, process and tool support, or the intervention of community managers [9].

Participation distribution : Online communities tend to be governed by the "90-9-1" rule, meaning that around 10% of the users contribute 90% of the work [16]; schema.org will be tested to see if it conforms to this pattern.

Typical user profiles : We aim to identify common user behavior patterns based on participation characteristics. They can be used to tailor community management towards certain sub-communities to improve participation inequality and improve group performance [11].

Our actual understanding of building and maintaining successful ontology engineering communities remains limited. schema.org seems to have some of the answers and our analysis tries to translate them into observable characteristics,

[1] http://schema.org/

[2] Whilst many other social channels host schema.org-related discussions (e.g. Quora or StackExchange), this paper focuses on the platforms that offer designated collaboration and community support. See also http://schema.org/ [Accessed on 4/1/18].

which can be applied to other ontology engineering projects. While our methods cannot claim to establish a causal link between any of them and schema.org's success; theory and studies in the broader space of online communities, e.g. [25], support our approach, and our study sheds light on how GitHub is used as a tool to facilitate the evolution of Web vocabularies, complementing previous works such as [1, 17]. The paper is structured as follows. We give a brief overview of collaborative ontology engineering and discuss the main findings of related studies, including previous empirical work on collaboration with respect to ontologies and GitHub (Section 2). We then describe our methodology and data sources in Section 3, followed by our main results of the four analysis areas (Section 4), and a discussion of their implications and the study limitations (Section 4.5). We finish with interim conclusions and proposed future work (Section 5).

2 Related Work

The process of people using technology to work collaboratively, otherwise known as Computer Supported Cooperative Work (CSCW) [20], has been extensively studied. In this section, we focus on previous research on two specific areas in this vast literature space: (i) collaboration around ontologies (or related artifacts e.g. schemas, vocabularies, knowledge bases); and (ii) collaboration via GitHub.

2.1 Ontology Collaboration

Collaborative ontology engineering involves multiple individuals or organisations communicating, often remotely, to create an ontology. A significant amount of the early literature in this area looked at the steps that need to be carried out collaboratively [22], and at the tool support required in each step [21]. Several tools have been developed over the years for this purpose, from OntoEdit [24], Swoop [10] and Semantic MediaWiki [12] to WebProtégé [27] and Neologism [2]. Directly relevant to our study, several newer ontology editors specifically link to GitHub to leverage its teamwork and version control features [1, 17].

As more ontology projects have been set up, researchers have begun to investigate collaborative ontology engineering empirically [23]. Initially, most work involved small user or case studies that aimed to validate or collect feedback on a specific methodology or tool [18]. As the field advanced, researchers had access to growing amounts of experimental and observational data, allowing them to expand their research questions. Walk et al. [32] analysed change log patterns of four collaborative ontology engineering projects, concluding that participants played different roles in the collaborative process. They identified four roles: administrators; moderators; gardeners (who focus on syntax errors and maintaining the ontology); and users (who frequently interact either to collaborate or revert each others changes over the same set of classes). They also concluded that the way people approach these edits depends on the hierarchical structure of the ontology. These findings align with Falconer et al's earlier study [5], who distinguished among the following roles: administrators; domain experts; and content

editors (who typically make the most edits). Wang et al. [33] also investigated ontology editor change logs to try and predict user changes based on previous contributions, concluding that further work was needed to factor in different ontology life-cycle stages and participant roles, despite some prediction success. Gil et al. looked at contributions and editing patterns in 230 semantic wikis [7], noting that only a small subset of users create properties and that further work is needed in order to understand how and if editing restrictions in the wikis may have affected the observed editing patterns. Also related to MediaWiki, Müller-Birn et al. [15] clustered editing activities and determined that Wikidata has a stronger focus on peer production rather than ontology engineering, with a large share of editors specializing on specific types of contributions.

Studies like these offer valuable insights into typologies of ontology engineering contributions, levels of engagement with specific technical features, common tool usage or ontology editing patterns. Our work complements them by taking an overarching view of the collaborative process, focusing more on how interactions within the ontology community are instigated. Our main data sources are interactions of community members carried out via emails and GitHub, rather than system logs, which capture less immediate forms of collaboration. Our work thus aligns with earlier efforts such as DILIGENT [18], which annotated structured discussions using Rhetorical Structure Theory (RST) concepts to automatically detect inconsistencies and resolve conflicts, and Cicero [3], a Semantic MediaWiki extension focusing on decision support through discussions. While these previous works have not been extensively tested, they share similar aims to our study. Comparatively, we use observational data from emails and GitHub discussions from a real-world, successful ontology community, schema.org.

2.2 Collaborative Coding with GitHub

GitHub[3] is an open-source code repository that facilitates collaboration and revision control in technical projects. Developers can create project branches to work on different functionality, push and pull code updates, and clone repositories for direct usage or re-purposing. GitHub is more than a distributed version control system (DVCS). It contains advanced functionality of a social network platform: users have public profiles detailing their involvements in GitHub projects. Developers can have discussions about project progress and raise and comment on project issues; which can be branched into different threads [14]. As GitHub-style platforms entered the mainstream, researchers started analysing how they are used and which factors make projects and communities successful. Earlier work by Duncheneaut [4] hypothesized that group open-source projects involve complex social structures. Forums like GitHub with developer profiles mean that users can view other users' profiles and judge their coding abilities from their publicly available code before working with them. Similarly, new developers looking to join collaborative projects can assess the skills and contributions of other members from previous releases to estimate expected contribution [26]. Behavior

[3] https://github.com/

and expectations adapt as newcomers become more familiar with a project. They may start by joining group discussions or building up a presence for themselves [28]. Their involvement evolves, both in volume and type of contribution and similar to other participatory platforms, a share of the community is made of "lurkers", people who observe projects without actively participating [26].

2.3 Collaborative Ontology Engineering using GitHub

In recent years, ontology engineers have started to use DVCS in general, and GitHub in particular to collaborate. Related research mostly focuses on new GitHub-enabled ontology tools [1, 17]. To the best of our knowledge, studies evaluating the use of GitHub in collaborative ontology engineering projects have yet to emerge. Additionally, schema.org is a broader collaborative venture than the case studies from the literature, with multiple ontologies being edited, documented and discussed. This paper focuses on analysing these discussions.

3 Data & Methods

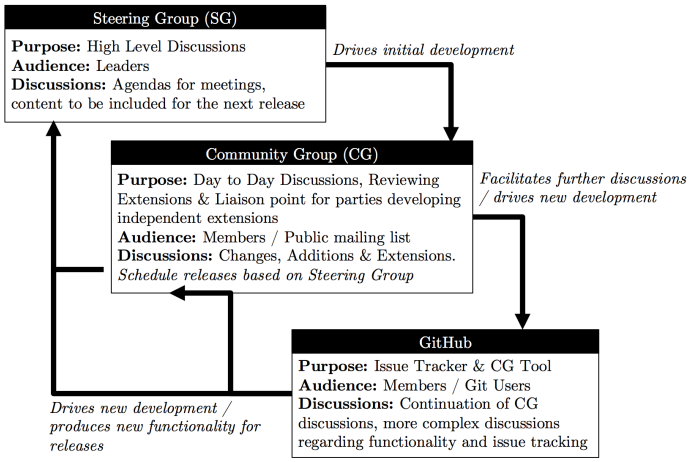

Fig. 1. Collaborative Workflow of schema.org

In our analysis we started from the three collaboration tools that were mentioned on the schema.org website: (i) the steering group;[4]; (ii) the community group; [5]; and (iii) the GitHub repository.[6] The collaborative workflow of schema.org, based on the details given on the project website, is illustrated in Figure 1. Anonymised data and analysis code are available on GitHub at https://github.com/samikanza/schema-datasets.

[4] https://groups.google.com/forum/#!forum/schema-org-sg
[5] https://www.w3.org/community/schemaorg/
[6] https://github.com/schemaorg/schemaorg

3.1 Data

The steering group is a small group of individuals, appointed by the sponsors of schema.org, in order to coordinate the development of the vocabulary. Most of its discussions are public. The respective Google group is a forum for high-level discussions and is not meant for "heavy traffic"[7]. Unsurprisingly, it has the least content, totaling 32 group discussion topics that make up 112 posts overall. A bulk of the topics (28 topics with 81 posts) were made in 2015, with one post in 2016 and two in 2017. Emails tend to focus on planning and scheduling releases.

The community group is a W3C forum for day-to-day discussions[8]. The group promotes GitHub as main community platform, especially for bug reports, and details how to raise and manage issues. Nevertheless, discussions also happen on the @public-schemaorg mailing list, with 1506 messages and 313 email threads from 263 different authors recorded thus far. GitHub is used to organize technical project elements. From the GitHub API we extracted 736 issues, 227 users who raised issues, and 406 users who commented on issues, totaling 483 unique users. Discussions range from adding new functionality and fixing bugs to general organization. We also noted some crossover between the two groups, where some GitHub issues and email subjects had the same titles. These threads had either been moved to GitHub for further discussion, or resulted in GitHub issues being raised. Given the sparseness of the steering group, we mainly used the community group mailing list and GitHub in our analysis.

3.2 Methods

To analyse the data we used a mixed-methods approach comprising of: (i) iterative thematic coding [19], to elicit discussion topics; and (ii) concrete methods, to compute topic popularity, levels of contribution, and engagement.

Topic Prevalence: GitHub issues and mailing list emails were thematically coded to assess **topic prevalence**. Python scripts were written using PyGithub[9] to extract schema.org issues and issue comments from the GitHub API, which were manually coded to identify topics. We started from the list of categories from Walk et al [32] that highlighted four ontology-centric activities: editing, adding, organizing and fixing content and formalized them into these topics: **Modification**; **Extension**; **Organization**; and **Bug**. Additionally, for issues that did not fit into these topics, or were considered off topic, we added an **Other/Off Topic** category. During coding, we identified six new topics (listed below). The topic descriptions were formalized to ensure consistency, and subsequently used to code the community group emails. Emails were inspected by thread, and each thread was coded according to its overall theme; for any that did not fit into the existing categories, they were inspected for common themes to form new topics, or were deemed **Other/Off Topic**. After coding the emails, we added

[7] https://groups.google.com/forum/#!forum/schema-org-sg
[8] https://www.w3.org/community/schemaorg/how-we-work/
[9] https://github.com/PyGithub/PyGithub

a **Release** category; as noted earlier, the community group is also used to plan and schedule releases. Finally, completely random samples of 10% of both corpora were re-coded and checked for consistency reasons after ensuring agreement between the authors of the defined categories. The final topic list consists of:

- **Release** - Discussing new release versions.
- **Extension** - Proposing additional functionality.
- **Clarification** - How something should be used/implemented or if it exists.
- **Modification** - Proposing a change to existing functionality.
- **Bug** - Detailing a bug or a fix for a bug.
- **Use by consumers** - How the schemas could be used by consumers.
- **schema.org website** - About the schema.org website.
- **Github use** - About how GitHub should be used in this project.
- **Organisation** - About general organisation.
- **Investigate Technology** - About investigating a new technology.
- **Documentation** - Adding or improving or editing documentation.
- **Other/Off Topic** - Irrelevant or didn't fit into the other categories.

Topic Popularity: We computed several descriptive statistics: number of replies in relevant email threads; number of comments on GitHub issues; mean/median number of responses on the mailing list and on GitHub; and percentage of topical conversations with no comments. We also inspected a random sample of 10% of unanswered and off topic messages (emails, issues) to identify common themes.

Participation Distribution: Participation is defined as number of emails sent and replied to, or number of issues raised and commented on. We sorted participants by participation level and assessed the overlap among the top 10 participants in the community group and on GitHub. We compared email addresses and GitHub usernames to determine when a participant on the mailing list was the same as a GitHub user (email addresses were inspected manually, and GitHub user ids were extracted using PyGithub). There were some instances in the community group where users were sending emails using different email addresses, in these cases we merged the user totals.

Typical User Profiles: Participants were categorised into profiles according to several dimensions: how active they are in the community, whether they initiate new conversations (by raising new issues or starting a new email thread) or whether they contribute to existing conversations (through replies and comments).

4 Results & Discussion

In this section we aim to answer the four research questions by presenting the results and discussing the main themes that emerged from the analysis.

4.1 Topic Prevalence

Table 1 illustrates how the 313 community group email threads and the 736 GitHub issues were categorised via thematic coding.

Topic	**Community Group**	**GitHub**
Extension	21.9%	38.9%
Clarification	41.7%	19.8%
Modification	1.8%	16.2%
Bug	5.5%	10.3%
Documentation	0.6%	4.6%
Organisation	8.5%	3.5%
schema.org website	0.9%	2.6%
Use by consumers	2.4%	2.5%
GitHub use	0.6%	0.5%
Investigate technology	2.1%	0.1%
Other/Off Topic	9.7%	1.0%
Release	4.3%	0.0%

Table 1. Topics discussed by the schema.org community

The community group focuses on clarifications and extension-based discussions (41.6% and 21.8%). Additionally, it hosts some organisational messages (8.5%), which is in line with the aims of the group. On GitHub, the community focuses mostly on extensions (38.8%), followed in roughly equal measure by modification or clarification issues (19.8% and 16.1%). This suggests that GitHub is used more to propose new functionality or changes to existing functionality, whereas the mailing list is used for clarifications of existing work.

We note that participants require an account to raise issues on GitHub, and might be hence more willing to make initial queries via a public mailing list. Further on, GitHub has stronger focus towards raising bugs (10.3% vs. 5.4% at almost double the number of posts), which fits with its purpose. The GitHub corpus also has much fewer off topic messages than the community group, where some of these messages involve unsubscription requests or irrelevant questions, both of which also lend themselves more to a public mailing list discussion. Finally, the community group is clearly the place to talk about scheduling and organisation of new releases, as intended by the community managers.

4.2 Topic Popularity

Figure 2 shows how many responses emails and issues in each topic category received. The types of topics the community engages with most via email are: extensions, clarifications, and releases; as well as organisational and off topic matters. Extensions and clarifications are also popular on GitHub, alongside modifications and bugs. The remaining topics, which are to a certain extent peripheral to the use of a revision control system, receive considerably less attention from the community on this platform. However, topic prevalence seems to have

a bearing on the topic popularity; to refer back to Table 1, the most popular topics in both instances are the ones that also showed the highest prevalence.

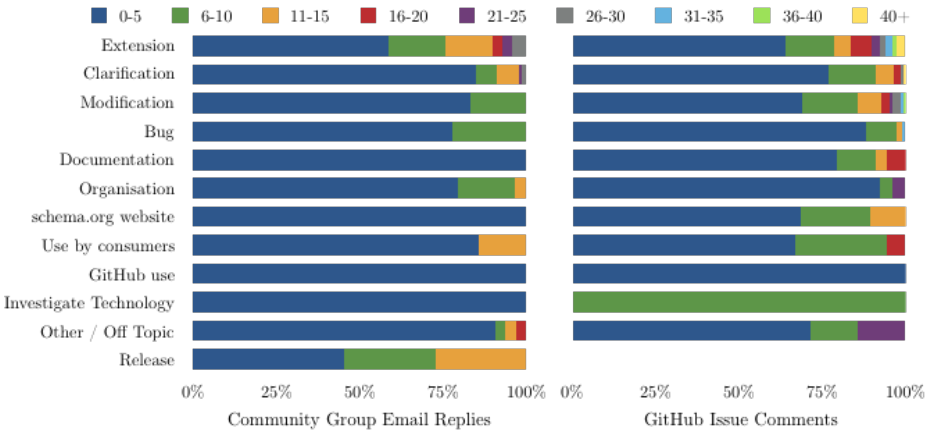

Fig. 2. Share of Responses to community group emails and GitHub issues per topic

Topic	**Community Group**			**GitHub**		
	Mean	Median	No Replies	Mean	Median	No Comments
Extension	6.4	4	27.1%	8.0	3	23.4%
Clarification	3.5	2	22.8%	4.3	2	32.2%
Modification	1	0	83.3%	5.2	2	24.4%
Bug	2.3	1	27.8%	2.4	1	43.4%
Documentation	2	2	0.0%	3.7	2	26.5%
Organisation	2.7	2	44.8%	3.0	2	19.2%
schema.org website	2	1	0.0%	4.1	2	15.8%
Use by consumers	3	1	14.3%	3.9	3.5	16.7%
GitHub use	1	1	50.0%	0.8	0	75.0%
Investigate technology	0.7	0	66.7%	6.7	0	0.0%
Other/Off Topic	2.1	0	53.1%	4.9	1	42.9%
Release	6.4	7	9.1%	N/A	N/A	N/A

Table 2. Mean/Median averages of responses per topic, and % of no responses

The difference between some topics' mean and median values (Table 2) shows that although some topics receive more engagement than others, the response level varies significantly. The median is mostly similar or lower for the community group, and consistently lower for GitHub, meaning that over 50% of these topics receive two or less responses (the results show that GitHub issues typically receive more comments than community group emails receive replies). It is also worth noting that, even popular topics typically receive between $0-5$ responses and as many as 43% of bug issues and 32% of clarification issues remained unanswered. This may point to lack of resources or incentives, or other deficiencies in the community organisation. Of 313 email threads, 33% received

no replies, while on GitHub of 736 issues 7% received no comments. Overall 40% of interactions received minimal engagement. The lack of engagement with **GitHub use** is less concerning, as this category only contains four issues.

We then looked in more detail at a random sample of 10% of the emails and issues that received no responses. The email sample had 10 emails: three extension emails that seemingly did not receive a response because the conversation was continued on GitHub; three other/off topic messages that were spam, an unsubscription request and a questionnaire link respectively; two detailed modifications which we believe did not merit replies; and two others for which the reason for lack of response was unclear to us. The GitHub sample included 20 issues: 55% were relatively new (dated September 2017 onwards), so it is possible that group members will reply in due course; 20% had been self assigned to the user who created the issue in the first place (all by two of the main GitHub users), therefore may not have necessitated a response. A further 15% were referenced in other, more descriptive issues which the community commented upon, suggesting that participants may have moved any potential discussion to those issues elsewhere; the final 10% of issues detailed how a certain fix had been made and linked to the appropriate commit, which could also justify a lack of response, as they did not include questions or other elements for discussion.

With respect to the off topic subjects, we were interested in learning more about which interactions attracted community participation despite their nature. 60% of the relevant emails were roughly equally split into: unsubscription requests (which would not occur in GitHub as users have control over leaving groups) and emails promoting surveys or courses which arguably did not merit a response. There were also two off topic political emails, and the rest were questions relating to schema.org that were more 'other' than 'off topic' and subsequently received responses. On GitHub, only 7 issues were classified as **Other/Off Topic**: three were uncommented (one had a blank issue body and a one word title, and the other had a one word title and issue body); the other four that were attended to either asked for advice about similar areas to schema.org or made suggestions regarding other technologies.

4.3 Participation Distribution

Figure 3 shows the participation level between the community group and GitHub. The group had 264 unique active users: 73 participants who only started email threads; 100 who only replied to them; and 91 who did both. The GitHub repository had 483 unique active users, which included 77 users who just raised issues, 256 who just commented on issues, and 150 who did both. The four graphs show similar contribution patterns, with 10% of users responsible for 80% of all contributions, which is a more balanced than the Nielsen norm [16], suggesting that there must be a higher proportion of users who contribute on a minimal, but significant enough level to influence these metrics. Finally, we note that in each case, there is one member that participates on a significantly higher level than other users; this will be elaborated on further when we analyse the most prolific members of the community (Table 3).

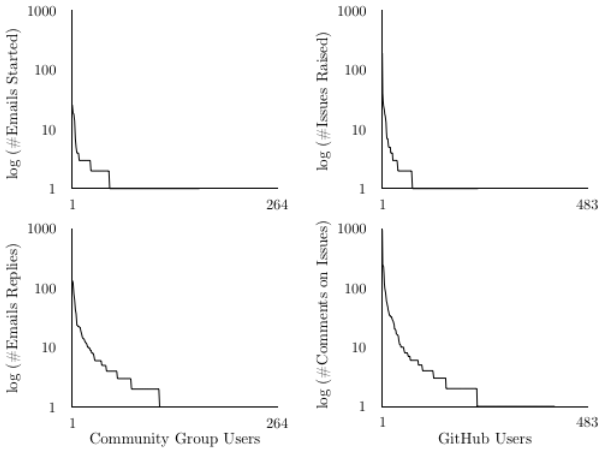

Fig. 3. Participation in the community group and GitHub

Highest Contributor	Community Group		GitHub	
	Emails Sent	Email Replies	Issues Raised	Comments
1st Contributor	User #1 (26)	User #1 (135)	User #1 (193)	User #1 (949)
2nd Contributor	User #2 (19)	User #2 (124)	User #10 (41)	User #2 (309)
3rd Contributor	User #3 (18)	User #10 (81)	User #12 (30)	User #9 (245)
4th Contributor	User #4 (14)	User #3 (64)	User #2 (25)	User #10 (227)
5th Contributor	User #5 (7)	User #9 (44)	User #13 (23)	User #12 (173)
6th Contributor	User #6 (5)	User #4 (37)	User #14 (21)	User #11 (118)
7th Contributor	User #7 (4)	User #11 (24)	User #15 (18)	User #16 (99)
8th Contributor	User #8 (4)	User #7 (23)	User #9 (17)	User #17 (98)
9th Contributor	User #9 (4)	User #8 (23)	User #5 (15)	User #18 (81)
10th Contributor	User #10 (3)	User #5 (22)	User #7 (13)	User #19 (60)

Table 3. Number of contributions of top 10 users, anonymised

We then analysed the top 10 contributors on each platform (community group: users who sent/replied to the most emails, GitHub: users who raised/ commented on the most issues) to establish any overlap. The top contributor on both platforms is the same. In the community group, nine users achieve top 10 ranks regarding starting or contributing to threads. on GitHub, top participation is less concentrated, with only five users being very active in both raising and commenting on issues. Seven individuals appear across all four leaderboards, illustrating the inequality in participation. In Section 4.4, we will dig deeper into the types of contributions different categories of users are responsible for.

4.4 Typical User Profiles

From our analysis so far, we have identified several roles:

- **Leaders**: Actively start and engage in discussions.
- **Broadcasters**: Actively start discussions, but rarely reply to them.
- **Followers**: Rarely start discussions, but actively reply to them.
- **Lurkers**: Rarely start or reply to discussions.

To categorise users we averaged the number of emails sent, emails replied to, issues raised, and issue comments. In the community group, active users have started at least one email thread and replied to at least four emails; and GitHub raised at least three issues, and commented on at least 15 issues. About 10% of the community group members and 9% of GitHub users fall into the leaders category; including one user who is top of each category, suggesting high interest and commitment. Roughly 2% of the mailing list contributors, and 8% of GitHub users are broadcasters, suggesting that community group members are less likely to initiate new discussions than GitHub users. This is somewhat unexpected, as public mailing lists have lower entry barriers than GitHub, as noted earlier. It does, however, fit with users preferring to hold lengthier discussions on GitHub, evidenced by some conversations beginning via email and then continuing on GitHub. Approximately 7% of both user bases are followers; showing a level of reactive engagement, where users are happy to discuss and express opinions on existing issues, but are less likely to or less comfortable in raising their own. In some instances, there are several followers who engage more with some issues than the leaders. The most common users are lurkers (community group: 75%, GitHub: 82%), suggesting either lurking until certain issue come to their attention to raise or discuss; or engaging episodically.

4.5 Discussion

Studying collaboration cross-sectionally reveals interesting differences. Both community channels appear to function under their intended purposes. However, this does not mean that certain areas could not still be improved. The unanswered topic sample suggested that engaging less with some topics is practical, due to conversations moving from the mailing list to GitHub, or because the topics do not necessitate a reply. However, both groups feature a fairly large share of unattended posts on core topics; and while our manual inspection shed some light into why they were left unanswered, further analysis is needed to understand the effects of this lack of engagement on participation, especially for newcomers [13], and across the four profiles discussed earlier. The popular GitHub topics map well to the ontology editing roles identified by Walk et al. [30] (administrators; moderators; gardeners; fixing content). Longitudinal studies should explore these parallels to establish how participation levels and discussions on community platforms impact ontology-centric activities. The participation distribution is less unequal than elsewhere, though a clear group of top contributors could be identified across the two corpora. Meanwhile, a fair share of activity seems to be generated by people outside the core of the initiative; contacting the community via the mailing list, or to a lesser extent via GitHub. The percentage of lurkers on GitHub is higher than reported in literature [29]. Overall, this speaks for an attractive community, with a more egalitarian participation of distribution, that could do more to onboard some members of the community, for example in the follower and broadcasting categories introduced earlier, in particular to resolve clarification or bug questions. Additionally, a more in-depth analysis is required to assess the importance of posts in each topic category; the current study con-

siders them all equal, but the small samples we inspected painted a much more reassuring picture than the initial metrics suggested. So far, all metrics we calculated can be meaningfully interpreted only in the context of related literature. They prompt discussions around what makes a successful ontology engineering community, which requires exploration. Collaborative ontology engineering literature sometimes touches upon the quality of the created ontology, but appears to lack an understanding of healthy, purposeful participation in ontology projects.

4.6 Limitations and Threats to Validity

Our data is subject to the following limitations. Matching users between the two datasets was not an exact science and we did not always have enough information to confidently disambiguate between the different accounts. Furthermore, the PyGithub scripts used to analyse the GitHub data do not facilitate extracting all engagement types: one can extract the number of comments per issue, but not necessarily other types of engagement such as user assignment history. We have also not considered other activities such as watching issues. Finally, our analysis could include other data sources, such as Quora or StackExchange to enrich the findings and employ methods that dig deeper into content of the posts.

5 Conclusions and Future Work

Our main conclusions for the analysed areas are: Topic prevalence shows that GitHub is used more to propose creating or editing functionality, whereas the mailing list is used more for clarifications. We found that topic popularity reflects topic prevalence, in that the most popular topics are typically ones that initiated the most new discussions. Overall, participation distribution is less unequal than expected with 10% of the users performing 80% of the work. A majority of the users across both platforms were lurkers, who rarely started new discussions or engaged with others in the group, and around 10% of the users can be classified as leaders, where a small core rank very highly in any form of participation.

There are a number of avenues for future work such as calculating how many of the extensions suggested by new users are developed and included in releases, and to what degree a user's social standing within schema.org influences this. Other avenues could include performing further qualitative analysis of emails and issues to understand the different social dynamics within these groups to see if they differ across groups. It would also be valuable to study how engagement adapts over time such that recommendations can be made to improve engagement and understand where it peaks and dips. Finally, we could look at how issues get assigned and solved on GitHub; who is in charge of the assigning; how many are self assigned, and how long any issues take to get solved (if at all).

6 Acknowledgements

This work has been conducted in the context of the Data Stories Project: EPSRC (EP/P025676/1) and the WDAqua Project: (Marie Skodowska-Curie Grant Agreement No 642795).

References

1. Alobaid, A., Garijo, D., Poveda-Villalón, M., Pérez, I.S., Corcho, O.: OnToology, A Tool for Collaborative Development of Ontologies. In: ICBO (2015)
2. Basca, C., Corlosquet, S., Cyganiak, R., Fernández, S., Schandl, T.: Neologism: Easy Vocabulary Publishing (2008)
3. Dellschaft, K., Engelbrecht, H., Barreto, J.M., Rutenbeck, S., Staab, S.: Cicero: Tracking Design Rationale in Collaborative Ontology Engineering. In: European Semantic Web Conference. pp. 782–786. Springer (2008)
4. Ducheneaut, N.: Socialization in an Open Source Software Community: A Socio-Technical Analysis. Computer Supported Cooperative Work (CSCW) 14(4), 323–368 (2005)
5. Falconer, S., Tudorache, T., Noy, N.F.: An Analysis of Collaborative Patterns in Large-Scale Ontology Development Projects. In: Proceedings of the 6th International Conference on Knowledge capture. pp. 25–32. ACM (2011)
6. Gil, Y., Knight, A., Zhang, K., Zhang, L., Sethi, R.: An Initial Analysis of Semantic Wikis. In: Proceedings of the Companion Publication of the 2013 International Conference on Intelligent User Interfaces Companion. pp. 109–110. IUI '13 Companion, ACM (2013)
7. Gil, Y., Ratnakar, V.: Knowledge Capture in the Wild: A Perspective from Semantic Wiki Communities. In: Proceedings of the Seventh International Conference on Knowledge Capture. pp. 49–56. K-CAP '13, ACM (2013)
8. Guha, R.V., Brickley, D., Macbeth, S.: Schema. org: Evolution of structured data on the web. Communications of the ACM 59(2), 44–51 (2016)
9. Jamali, S., Rangwala, H.: Digging digg: Comment Mining, Popularity Prediction, and Social Network Analysis. In: Web Information Systems and Mining, 2009. WISM 2009. International Conference on. pp. 32–38. IEEE (2009)
10. Kalyanpur, A., Parsia, B., Sirin, E., Grau, B.C., Hendler, J.: Swoop: A Web Ontology Editing Browser. Web Semantics: Science, Services and Agents on the World Wide Web 4(2), 144–153 (2006)
11. Kraut, R.E., Resnick, P., Kiesler, S., Burke, M., Chen, Y., Kittur, N., Konstan, J., Ren, Y., Riedl, J.: Building Successful Online Communities: Evidence-Based Social Design. Mit Press (2012)
12. Krötzsch, M., Vrandecic, D., Völkel, M.: Semantic Mediawiki. In: International Semantic Web Conference. vol. 4273, pp. 935–942. Springer (2006)
13. Lave, J., Wenger, E.: Legitimate Peripheral Participation in Communities of Practice. Supporting lifelong learning 1, 111–126 (2002)
14. Lima, A., Rossi, L., Musolesi, M.: Coding Together at Scale: GitHub as a Collaborative Social Network. In: ICWSM (2014)
15. Müller-Birn, C., Karran, B., Lehmann, J., Luczak-Rösch, M.: Peer-production System or Collaborative Ontology Engineering Effort: What is Wikidata? In: Proceedings of the 11th International Symposium on Open Collaboration. p. 20. ACM (2015)
16. Nielsen, J.: Participation Inequality: Encouraging More Users to Contribute (2006)
17. Petersen, N., Coskun, G., Lange, C.: TurtleEditor: An Ontology-Aware Web-Editor for Collaborative Ontology Development. In: Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on. pp. 183–186. IEEE (2016)
18. Pinto, H.S., Staab, S., Tempich, C.: DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies. In: Proceedings of the 16th European Conference on Artificial Intelligence. pp. 393–397. IOS Press (2004)

19. Pope, C., Ziebland, S., Mays, N., et al.: Analysing Qualitative Data. Bmj 320(7227), 114–116 (2000)
20. Schmidt, K., Bannon, L.: Taking CSCW Seriously. Computer Supported Cooperative Work (CSCW) 1(1), 7–40 (1992)
21. Schober, D., Malone, J., Stevens, R.: Practical Experiences in Concurrent, Collaborative Ontology Building using Collaborative Protégé. ICBO p. 147 (2009)
22. Simperl, E., Luczak-Rösch, M.: Collaborative Ontology Engineering: A Survey. The Knowledge Engineering Review 29(1), 101–131 (2014)
23. Strohmaier, M., Walk, S., Pöschko, J., Lamprecht, D., Tudorache, T., Nyulas, C., Musen, M.A., Noy, N.F.: How Ontologies are made: Studying the Hidden Social Dynamics behind Collaborative Ontology Engineering Projects. Web Semantics: Science, Services and Agents on the World Wide Web 20, 18–34 (2013)
24. Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D.: OntoEdit: Collaborative Ontology Development for the Semantic Web. The Semantic Web-ISWC 2002 pp. 221–235 (2002)
25. Tinati, R., Van Kleek, M., Simperl, E., Luczak-Rösch, M., Simpson, R., Shadbolt, N.: Designing for Citizen Data Analysis: A Cross-Sectional Case Study of a Multi-Domain Citizen Science Platform. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. pp. 4069–4078. CHI '15, ACM (2015)
26. Tsay, J., Dabbish, L., Herbsleb, J.: Influence of Social and Technical Factors for Evaluating Contribution in GitHub. In: Proceedings of the 36th International Conference on Software Engineering. pp. 356–366. ACM (2014)
27. Tudorache, T., Nyulas, C., Noy, N.F., Musen, M.A.: WebProtégé: A Collaborative Ontology Editor and Knowledge Acquisition Tool for the Web. Semantic web 4(1), 89–99 (2013)
28. Von Krogh, G., Spaeth, S., Lakhani, K.R.: Community, Joining, and Specialization in Open Source Software Innovation: A Case Study. Research Policy 32(7), 1217–1241 (2003)
29. Wagstrom, P., Jergensen, C., Sarma, A.: Roles in a Networked Software Development Ecosystem: A Case Study in Github (2012)
30. Walk, S., Esín-Noboa, L., Helic, D., Strohmaier, M., Musen, M.A.: How Users Explore Ontologies on the Web: A Study of NCBO's BioPortal Usage Logs. In: Proceedings of the 26th International Conference on World Wide Web. pp. 775–784. International World Wide Web Conferences Steering Committee (2017)
31. Walk, S., Singer, P., Noboa, L.E., Tudorache, T., Musen, M.A., Strohmaier, M.: Understanding how Users edit Ontologies: Comparing Hypotheses about Four Real-World Projects. In: International Semantic Web Conference. pp. 551–568. Springer (2015)
32. Walk, S., Singer, P., Strohmaier, M., Tudorache, T., Musen, M.A., Noy, N.F.: Discovering Beaten Paths in Collaborative Ontology-Engineering Projects using Markov Chains. Journal of Biomedical Informatics 51, 254–271 (2014)
33. Wang, H., Tudorache, T., Dou, D., Noy, N.F., Musen, M.A.: Analysis of User Editing Patterns in Ontology Development Projects. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 470–487. Springer (2013)
34. Yang, M.C., Rim, H.C.: Identifying Interesting Twitter Contents using Topical Analysis. Expert Systems with Applications 41(9), 4330–4336 (2014)