Profiles of Schema Evolution in Free Open Source Software Projects

Panos Vassiliadis
Univ. Ioannina,
Ioannina, Greece
pvassil@cs.uoi.gr, 0000-0003-0085-6776

***Abstract*—In this paper, we present the findings of a large study of the evolution of the schema of 195 Free Open Source Software projects. We identify families of evolutionary behaviors, or taxa, in FOSS projects. A large percentage of the projects demonstrate very few, if any, actions of schema evolution. Two other taxa involve the evolution via focused actions, with either a single focused maintenance action, or a large percentage of evolution activity grouped in no more than a couple interventions. Schema evolution also involves moderate, and active evolution, with very different volumes of updates to the schema. To the best of our knowledge, this is the first study of this kind in the area of schema evolution, both in terms of presenting profiles of how schemata evolve, and, in terms of the dataset magnitude and the generalizability of the findings.**

***Index Terms*—Schema Evolution, Software Evolution**

I. Introduction

Much like traditional software, database schemata change over time. *Schema Evolution* means that tables and attributes can be added, deleted or renamed, and their data types and keys can be altered. Changing the schema of a database can incur a significant impact, as the surrounding code can be syntactically and semantically inconsistent with the new structure of the schema, leading to application failures or incomplete data delivered to end users. Despite this significance, and in sharp contrast to the study of software evolution and maintenance for years from the software engineering community, we know very little on how schemata evolve: are there any patterns, similarities, recurring behaviors, or even laws in the way schemata evolve? Understanding the mechanics of schema evolution is a piece of knowledge, currently absent from our body of knowledge as a data engineering community, that apart from enriching humanity's knowledge and moving from word-of-mouth impression to concrete evidence, can facilitate both the management of information systems on the practical side, and its scientific basis on the research side, by providing insights to different audiences: (a) software curators and developers who can benefit from the ability to (even coarsely) predict the tendency of a schema to evolve (which can be used for decision making, recourse allocation, software selection), and, (b) the academic community, who can have hard evidence on the suitability of the underlying data models for evolving applications, the potential hindrances to their usage and the needs of deeper education of students.

The goal of this paper is to expand our understanding of how relational schemata in the domain of Free Open Source Software (FOSS) projects evolve at the logical level, by introducing taxa of projects with similar schema evolution profiles. To this end, we have performed the largest ever - to our knowledge- study of schema evolution, by extracting, measuring and analyzing the history of 195 schemata of Free Open Source Software projects.

Research Questions: We pose three research questions to address the above goal.

- RQ1. Is schema evolution extensively present? Is schema evolution a process that frequently encountered, and if yes, to what measurable extent does it occur in terms of frequency and volume?
- RQ2. Are there consistent patterns in the lives of schemata – i.e., can we extract families, ("taxa" as in biology) of schemata, with respect to the way they evolve over time?
- RQ3. What are the quantitative characteristics of schema evolution and how do they perform for different taxa?

Means. To address the above research questions, we have performed the largest study of schema evolution ever performed in the literature (to the best of our knowledge) via a systematic collection of data for schemata in FOSS projects. Specifically, we collected 327 schema histories in Free Open Source Software projects, based on a principled collection method and with quality criteria to avoid dummy projects. Out of them, we identified 195 candidate whose history seemed to exhibit evolutionary characteristics, and for each such project, we automatically extracted schema histories from their git repository hosted in Github[1]. For each such history (which is a list of versions of the schema DDL file), we extracted the differences between subsequent versions and measurements in terms of timing, schema size, numbers of tables and attributes changed, coming up with specific measures of change to characterize the *heartbeat* of change of a schema and its characteristics.

Results. For the first time in the literature (to the best of our knowledge), we study this heartbeat of change, and, following

[1]All data, results, summary statistics and extra explanations are publicly available in https://bit.ly/3nMggEx (at Github). The site https://bit.ly/3ptnBJo, summarizes our research.

an iterative, qualitative process, we studied their evolution and grouped them in *taxa of evolution, i.e., families of schemata, which share similar evolution characteristics*. The taxa of schema evolution are: (1) completely *frozen* schema histories with zero change at the logical level; (2) *almost frozen* histories of very small change, typically with few intra-table attribute modifications; (3) almost frozen histories but with a single spike of change and almost no other change (*Focused Shot and Frozen*); (4) histories of *moderate evolution*, without spectacular changes, but rather small deltas spread throughout the life of a project; (5) projects with evolution similar to the moderate one but also with a pair of spikes on their activity (*Focused Shot and Low*); and (6) histories of *active* projects, typically with significant amount of change both as intra-table change and in terms of table generation and eviction.

Contribution. To the best of our knowledge, *this is the largest study ever performed, surpassing previous studies by at least an order of magnitude*. Apart from the sheer volume of the schemata studied, the principled method we followed for collecting the schemata, extracting their versions and performing the analysis, allows us to argue that *our results are* largely generalizable for Free Open Source Software. Secondly, we verify previous research that schema evolution is indeed evident; however, the examination of a large corpus of projects produces unshakable evidence for the first time, that its absence is way more omnipresent than its presence. This is a significant result as it establishes a deep problem in the way the relational part of information systems is linked to the overlying software. Third, this is the first time ever *that taxa of schema evolution are presented*. The identification of taxa provides us with *a fundamental tool for characterizing and forecasting the propensity of the DBA's of a certain database towards change*. Apart from the above, subsequent studies can benefit from *the experimental method, the nomenclature, the visualization and the analysis methods*.

Roadmap. In Section II, we present related work. In Section III, we discuss our experimental method, nomenclature, and threats to validity, in order to support the generalizability of our results. In Section IV, we discuss our findings concerning the taxa of schema evolution. In Section V we discuss the statistical support on the validity of the proposed taxa. We conclude in Section VI with a discussion of final thoughts.

II. Related Work

The study of schema evolution has taken several research paths. Apart from the engineering of schema evolution [1], [2] researchers have worked towards producing algebras of schema evolution operations (SMO's), to describe the history of schema changes in a semantically rich sequence of operations [3], [4], [5]. Apart from a study in the early '90s [6], it was only the proliferation of Free and Open Source Software (FOSS) that gave a momentum to the study of how schemata evolve. The original studies, [6] and 15 years later, [7], were focused on a single case study (a hospital database and mediawiki, respectively) and the quantification of changes in different categories, and the main finding is the significant dominance of expansion over deletion, as well as the different intensity of change at different tables. [8] follows along the same lines. [9] and [10] report that a large part of tables and modules of application code are not-synchronized at all times. The largest study so far has been [11] with ten open-source schemata studied. Again, the percentages of changes are reported, with *add table*, *add column* and *change column datatype* being the most populous. The lack of integrity constraints in several places (independently verified and explained later, in [12]), is also reported, along with the non-synchronization of application code and schema, as well as the presence of focused periods of change in the early life of the schemata. [13] shows that schemata grow over time with bursts of concentrated effort of growth and/or maintenance interrupting longer periods of calmness. [14] and [15] study patterns of *tables*, rather than *schemata*, best summarized by the *Electrolysis pattern*, named after the intense antithesis in the lives of dead and survivor tables: whereas dead tables are attracted to lives of short or medium duration and absence of schema update activity, survivors are mostly located at medium or high durations and the more active they are, the stronger they are attracted towards high durations. Later studies have moved towards the study of schema evolution in the realm of JSON, NoSQL databases [16], [17].

Still, data collection has been very hard to tackle, with no study exceeding a dozen projects. Compared to the previous work, our study (a) is the first one to *study the heartbeat* and provide *profiles of schema evolution* for the studied projects, and (b) is conducted over *one or two orders of magnitude higher in terms of studied projects*, via a *principled collection method*; resulting in a strong generalizability for the produced profiles for FOSS projects.

III. Experimental Setting

In this section, we discuss our data collection process, the artifacts and metrics that have been extracted from the fully automated processing of the schema histories, and the threats to validity of this study.

A. Data Collection

The goal of the data collection process was to collect data for a large number of projects with quality guarantees. Originally, we tried to work with GHTorrent [18], a well-known GitHub mining tool. Still, we were unable to usefully produce a data set with it, and, we resorted to one of its querying platforms, Google Cloud BigQuery[2]. Among its many datasets, BigQuery provides the GitHub Activity Data dataset in relational format, along with SQL facilities to query it. The GitHub Activity dataset is a 3TB+ dataset that contains a full snapshot of the contents and the commits of more than 2.8 million open source GitHub repositories. We queried the contents table for all file descriptions ending to a '.sql' suffix, in 2019-04-24 and 25, and obtained a collection of SQL file descriptions (to which we refer to as SQL-Collection, hereafter) for 133,029 repositories.

[2]https://cloud.google.com/bigquery/

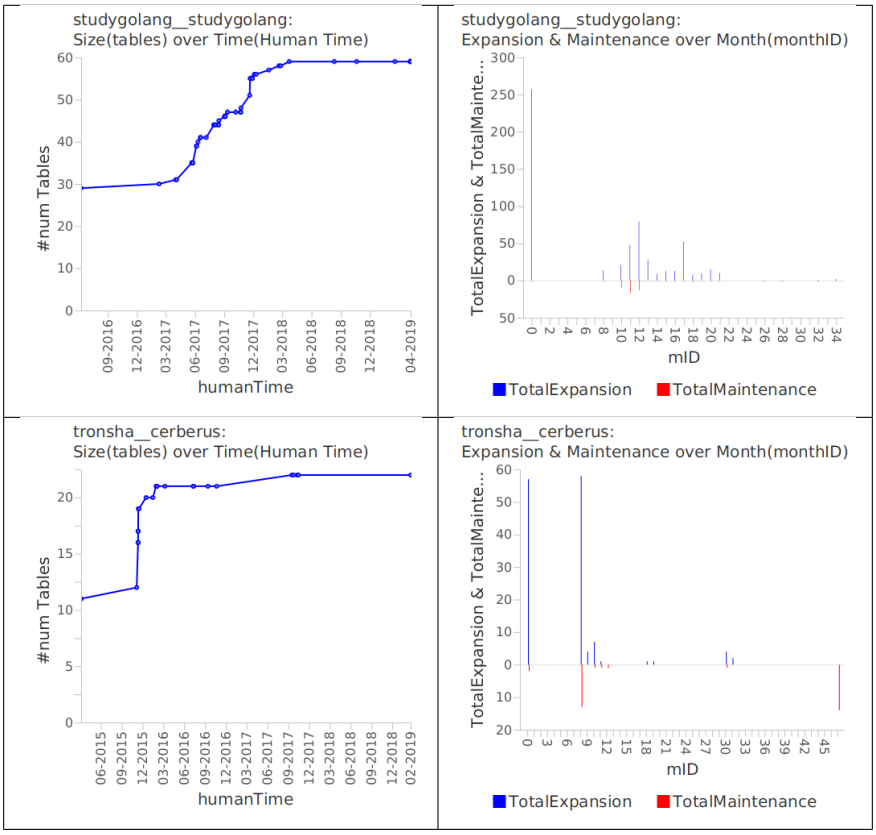


Figure 1. (Left) Schema size over time (#tables over human time, and, (Right) Expansion and Maintenance activity (#attributes affected) per month, for two active projects.

Since this number of files and repositories is extremely high to handle, we had to narrow it down via a principled selection method. To this end, we combined the SQL-collection that we obtained with another public dataset, available via BigQuery as the Libraries.io dataset[3]. Libraries.io is an opensource community monitoring and gathering metadata for over 2.7M unique open source packages from 3 source code repositories, namely GitHub, Gitlab and BitBucket. We have worked with the collection exported at 2018-12-22. The Libraries.io collection offers project metadata, including whether the project was an original project or a fork, its number of stars, watchers, etc.[4] We joined the two data sets over (a) their repository names and (b) the URL of their projects, taking care to include only Libraries.io projects which were (i) original repositories, (ii) with more than 0 stars and (iii) more than 1 contributor.

To alleviate the possibility of void projects, or repetitions of the same change in multiple files, the results where post-processed with several criteria:

- We excluded all results whose file descriptions included the terms 'test' or 'demo' or 'example' in the path.
- For all the cases where multiple vendors were supported, we chose MySQL as the DBMS to investigate (as the most popular DBMS in our collection).
- For all the cases where multiple SQL files were reported, we went through manual inspection, to identify candidates that could be reduced to a single DDL file with the table creation statements. Cases omitted included (i) several DDL scripts in a file-per-table mode, (ii) incremental maintenance of the schema, (iii) the Cartesian product of multiple vendors X different versions of the same schema for different languages (e.g., projects having different schemata for the combination of {English, French, ...} by {mysql, postgres, mssql, ...}).

The result of this post-processing was a data set of **365 FoSS schema histories**, which we refer to as the Lib-io dataset. For all these 365 projects we went on to clone them locally and extract their schema histories between 7 and 26 May 2019.

To remove erroneous or void files, a final post processing took place over the retrieved repositories. First, we removed 14 projects whose history extraction resulted in 0 versions (i.e., their file descriptions in Github Activity did not match their actual, downloaded .git). We also removed the commits with empty files, as well as the histories whose .sql files did not contain "CREATE TABLE" statements. This involved 24 projects. Out of the remaining **327 repositories**, we isolated **132 rigid projects with just one version** of the schema file, i.e., projects whose schema never changed. The number is

[3]https://libraries.io/about

[4]Among others, a GitHub project has (a) stars (i.e., someone considered it interesting and pressed the 'Star' button), (b) forks (i.e., a user copies the project in his own 'space' to work independently on it), (c) collaborators (users contributing to a project owned by someone else).

striking: 132 out of 327 is a vast 40% of projects without any schema evolution(!). Eventually, **we ended up with 195 non-rigid repositories** that were used for our subsequent analysis, and to which we refer as the Schema_Evo_2019 data set.

B. Nomenclature and Measurements

To address the diversity of nomenclature and measurements, in this section,we establish a reference nomenclature.

A *Schema History* is a list of *commits* (a.k.a. *versions*) of the same DDL file of a database schema, ordered over time. A *transition* from an older version i to its subsequent version $i+1$ occurs at the timepoint where version $i+1$ is committed, and potentially incurs changes in the schema. The *initial, originating version* of the history is called, as shorthand, $V0$. *Active commits* are the commits whose sum of updates (see next) exceeds zero. *Non-Active commits* involve changes in comments, directives to the DBMS, INSERT statements, indexing, and other changes that do not affect the logical capacity of the schema in terms of tables, attributes, data types or primary keys. The *Schema Update Period (SUP)* is the time span (in human time) between the first and the last commit of the schema file. This is a very different time interval than its superset, *Project Update Period (PUP)* that marks the start and end of project history.

For each transition of the schema history, our tool, Hecate, automatically computes several categories of measurements. First, it computes *timing* information, like the distance of the $i+1$ commit from V0 in days, and the running month and year. Second, it registers the *schema size* (no. of tables, attributes) of both the older and the subsequent version of the transition. Third, Hecate identifies and quantifies *updates* (all measured in attributes): attributes born with a new table, attributes injected into an existing table, attributes deleted with a removed table, attributes ejected from a surviving table, attributes having a changed data type, or a participation in a changed primary key. We measure as *Expansion* the sum of attributes born and injected, and as *Maintenance*, the sum of all the other categories. *Total Activity*, or simply *Activity*, of the schema, is the sum of Expansion and Maintenance (in what follows, remember that fundamental unit of measurement of change in our setting is the attribute, for all categories).

We define the *heartbeat* $H = \{c_i(e_i, m_i)\}$ of the schema as the ordered list of pairs (expansion, maintenance), one per commit, of the schema history. Due to the visual impression of the shape of the heartbeat (see Fig. 2), we refer to standing out commits with total activity strictly higher than 14 attributes as *"reeds"*, and commits with lower activity as *"turf"*. The reed limit was produced by taking all single-commit projects, sorting them by activity (producing a power-law like distribution)and splitting them at the 85% limit).

C. Threats to validity

From the very beginning, our goal was to be able to clearly specify the scope and generalization of our study.

Scope. *We are interested in the monitoring of the evolution of the logical-level relational schema for significant Free Open Source Software projects, hosted in GitHub.* We want to stress that, in the context of our deliberations, we are not covering or generalizing to proprietary schemata outside the FoSS domain. We do not cover conceptual or physical schemata. We are also restricted in relational schemata and not XML, JSON, or another format.

External Validity. The external validity refers to the possibility of generalizing the findings of a study to a broader context. We claim that our elicited repositories and their extracted history give a fairly representative view of schema evolution in FoSS projects.

First, the SQL-Collection data set includes the locations of schemata that are part of Free Open Source Projects (and not proprietary ones), available via GitHub. *Practically, the domain of search was all the .sql files of GitHub* reported at Github Activity and Libio. *In our opinion, this is also a very good representative of open source software overall, as GitHub is the main public repository for FoSS software*. We applied the restriction that the respective files end with a '.sql' suffix. It is possible that other suffixes, are used by developers. To the extent that this would be a non-recommended practice, we believe that the projects ending up in our study are valid candidates to be monitored as significant projects.

Second, the Lib-io data set is a restricted version of the SQL-collection data set with the schemata whose repository path was monitored by the public Libraries.io data set. *We applied the filter of more than one contributor, more than 0 stars and non-forking*. We believe this to be a fairly broad scope for original projects with a degree of significance (without implying, of course, that other projects are not significant).

Third, *the subsequent filtering, performed an extra quality check*. We believe that filtering out tests, examples and demos is not decreasing the value or validity of our approach. Although databases of these types have their value, monitoring their evolution, would not say much for the essence of a database supporting the regular operation of a software project. For the case of multi-vendor support for schemata, we are also confident that our choice to select only one vendor is the appropriate one, especially since we are studying logical-level changes. The only ambiguous situation, was the necessity to omit multi-file DDL declarations. This improves the precision of our study (as we are sure we get the correct history of the DDL statements) but reduces the recall.

Fourth, *the domains of the collected projects are diverse enough to support our external validity claim*. Specifically, the project domains include Content Management Systems, IoT Management on the cloud, Task Management Systems for operating systems, similarly for web services, Messaging Platforms, Systems for the management of Scientific Data, Web on-line stores, On-line Charging Systems (OCS), etc.

An issue of future investigation is also the commit habits of different projects: other teams commit small increments, other ones group changes in larger commits. Although this has an impact to the internal structure of changes, it does not impact the aggregate profile of a project. An extra issue of concern has to do with the non-linearity of git histories [19].

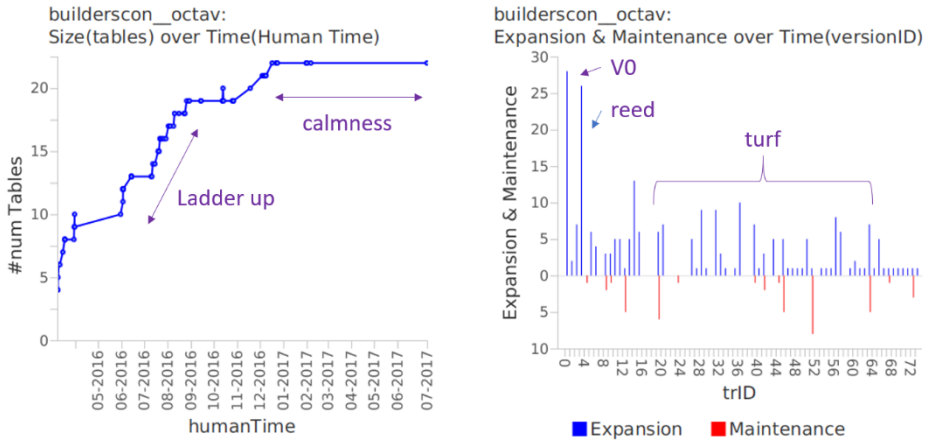


Figure 2. A reference example of schema and activity evolution via the builderscon_octav project: (Left) Schema size over time (#tables over human time): each dot on the line is a commit (observe that many of them do not affect schema size, as they are intra-table changes, or changes to documentation, data insertions etc., that do not affect the schema). The points are placed with respect to human time thus, they are not uniformly distributed. Observe that (a) the main part of the schema growth is a focused period tagged "ladder up" and (b) towards the end of the monitored period, commits are more infrequent and change is smaller. (Right) Expansion and Maintenance activity (#attributes affected) over transition ID (attn: not human time, but sequential id's of commits to the schema file). Blue bars above the x-axis measure expansion (attributes added to the schema), whereas read bars under the x-axis measure maintenance (attribute deletions, data type or PK changes). Observe how the change rates of the left and right parts are not isomorphic.

We investigate the entire schema history, whereas one might consider focusing on a single branch of the history.

Experimental Reliability. We tested our extraction scripts with OpenCart (the largest of our studied projects) for which we had a previous past extraction of its history, in 2016. The comparison produced an almost identical result, as only one commit out of 412 was missing from the GitHub history we extracted. We manually tested the histories of the retrieved files against the number of commits reported at GitHub for the respective file, for a random sample of 50 cases. In all cases there was an exact match. We also confirmed that the missing projects had also been removed from GitHub at the time of the cloning via a sample of 7 of them. Concerning our own software, we did extensive checks to our metrics computation tools. Overall, although bugs or omissions are still possible, we are quite confident with our software tool suite.

IV. Findings

A. Derivation and intuition of taxa of similar evolution profile

We have organized the studied schemata in families of evolutionary behavior, which we call *"taxa"*. *The derivation of the taxa was a (i) manual, (ii) qualitative and (iii) iterative process*. We automatically generated charts, in particular, the heartbeat and the schema size chart[5]. Manual inspection made apparent that there was a clear discrimination between *(a) completely frozen histories*, without any change to their logical schema, (b) a very large number of *almost frozen histories*, with very few commits, very small volume of change and mostly without change in the schema size of the project, (c) *moderately evolving projects*, mostly in terms of "regular" small changes and less in terms of schema growth, and, (d) *active* projects with significant change in the schema size, and typically, frequent active commits. Soon, our iterative, manual, visual inspection of the project charts and metrics revealed that we could further isolate two more taxa of *focused change*: a taxon of very few commits (i.e., mostly in the almost frozen category) but with a focused amount of change in a single commit and maybe a couple of commits of very small volume (i.e., the change was fundamentally focused in a single commit), and a taxon of -again- few commits, but also with a couple of reeds, and moderate to high change.

Once this manual, qualitative process was mature, with only a few grey-zone projects having an ambiguous label, we were able to extract a simple *classification tree of taxa* (see Fig. 3), with respect to active commits and activity. Table I demonstrates the initial qualitative intuition and the subsequent rule-based, quantitative definitions for the taxa. A Kruskal-Wallis analysis (see Section V), both for the entire set of the taxa, and for all pairwise comparisons, verifies the difference of the taxa in terms of active commits and total activity. In what follows, it is important to remember that *we have not selected just any random project, but rather, we intentionally restricted our scope to original, stared projects, where people were actually contributing effort to develop and maintain* (see Sec. III). *Overall, 65% of projects spanned more than 24 months and 77% more than a year.*

In Fig. 4 we present a summary of the statistical profiles of the different taxa with respect to their (a) cardinalities, and, (b) measures that characterize their evolutionary activity. Observe that all taxa come with a significant cardinality, and in any case, each of the taxa alone is several times larger than the corpus of each of the previous studies in the related work (that never surpassed a dozen of studied projects).

[5]Hecate at https://github.com/DAINTINESS-Group/Hecate and Heraclitus Fire at https://github.com/pvassil/HeraclitusFire, set up our toolset.

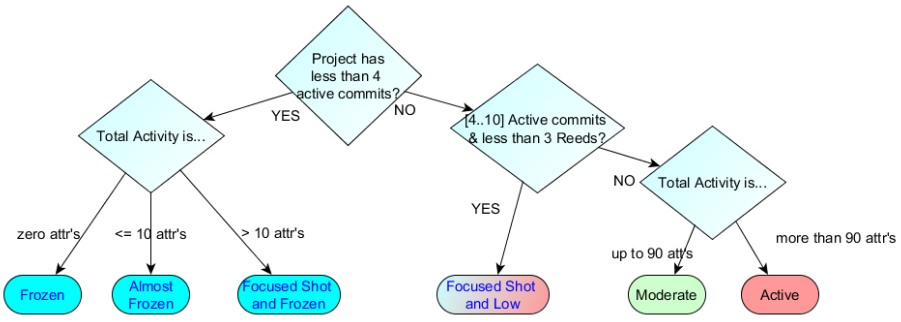


Figure 3. Taxa of Schema Evolution for FOSS Projects

	Frozen				Almost Frozen				Fshot n Frozen				Moderate				Fshot n Low				Active			
Count	34				65				25				29				20				22			
	min	**med**	*max*	*avg*	*min*	**med**	*max*	*avg*	*min*	**med**	*max*	*avg*	*min*	**med**	*max*	*avg*	*min*	**med**	*max*	*avg*	*min*	**med**	*max*	*avg*
Sch. Upd. Period (months)	1	**1**	69	*8.24*	1	**6**	99	*11.98*	1	**2**	46	*9.28*	1	**20**	100	*23.62*	1	**17.5**	57	*21.05*	1	**31**	100	*35.95*
TotalActivity	0	**0**	0	*0*	1	**3**	10	*3.62*	11	**23**	383	*45.64*	11	**23**	88	*30.0*	27	**71**	315	*105.15*	112	**254**	3485	*546.14*
#Commits	2	**2**	11	*3.18*	2	**3**	13	*3.83*	2	**4**	17	*4.56*	5	**10**	43	*13.52*	7	**10.5**	19	*11.55*	9	**36.5**	516	*77.36*
#Active Commits	0	**0**	0	*0*	1	**1**	3	*1.40*	1	**2**	3	*1.76*	4	**7**	22	*8.52*	4	**6.5**	10	*6.30*	7	**22**	232	*43.95*
#Reeds	0	**0**	0	*0*	0	**0**	0	*0*	0	**1**	3	*0.84*	0	**0**	2	*0.17*	1	**1**	2	*1.40*	1	**5.5**	31	*7.32*
Turf commits	0	**0**	0	*0*	1	**1**	3	*1.40*	0	**1**	3	*0.92*	4	**7**	22	*8.34*	2	**5**	9	*4.90*	0	**18.5**	207	*36.60*
Table Insertions	0	**0**	0	*0*	0	**0**	2	*0.26*	0	**2**	18	*2.48*	0	**2**	6	*2.14*	0	**4.5**	16	*6.70*	0	**24**	301	*5.23*
Table Deletions	0	**0**	0	*0*	0	**0**	1	*0.09*	0	**1**	45	*3.88*	0	**0**	4	*0.66*	0	**2.5**	15	*4.45*	0	**9**	214	*25.64*
#Tables@Start	1	**2**	227	*14.26*	1	**3**	68	*5.94*	1	**4**	47	*6.60*	1	**5**	65	*8.31*	2	**8**	26	*8.90*	2	**20**	61	*24.18*
#Tables@End	1	**2**	227	*14.26*	1	**3**	70	*6.11*	1	**5**	18	*5.80*	1	**6**	68	*9.79*	2	**10**	33	*11.15*	1	**22.5**	135	*33.77*

Figure 4. Measurements per Taxon (min, **median**, max, *avg;* Definition wherever appropriate).

B. The Frozen Land of Total Rigidity

"In a survey of 20 database administrators (DBAs) at three large companies in the Boston area, we found that ... DBAs try very hard not to change the schema when business conditions change, preferring to "make things work" without schema changes." [20] We have named this tendency as *gravitation to rigidity* in studies of smaller scale [13], [14], [15]. The most characteristic and undisputed finding of this study is the confirmation *with solid numbers over a very large dataset* of the above. Overall, *within the scope of FOSS projects, it is the absence of evolution of the logical level of the schema that demonstrates itself in large numbers, as opposed to its presence*. Out of the 327 repositories that we cloned, 132 (40%) had a single commit for their schema whatsoever, 34 (10%) had more than 1 commits, but zero changes at the logical-level schema, and 65 (20%) were almost frozen (with less than 4 active commits and 10 modified attributes). Overall, *70% of the projects, demonstrated total absence or very small presence of change.*

Interestingly, this absence of evolution is not a feature of abandoned projects, but rather an attitude of the developers: In terms of project duration (attn.: not schema update period), 68% of Frozen projects span more than 24 months, and 79% span more than 12 months. The respective numbers for Almost Frozen are 58% and 73%, respectively. The commits concerning the DDL file amounted to 6% and 5% of the total commits, respectively. Concerning change, the Almost Frozen category, which is the only one with some change, has a median of 3 commits, one of them active, a median of zero tables inserted and deleted (75% of projects having a flat schema line) and a median total change of 3 attributes.

Table I
INTUITION AND DEFINITION FOR THE TAXA OF SCHEMA EVOLUTION.

Taxon	Motivating Intuition and *Resulting Classification Definition* (*italics*)
History-less	Only 1 commit of the .sql file (we did not study them, due to lack of transitions)
Frozen	With history, but with *total activity of 0 changes & 0 active commits*
Almost Frozen	Very few commits and low change volume *Def: At most 3 active commits, change less or equal to 10 updated attributes*
Focused Shot & Frozen	Very few commits, focused change (not necess. small) in a single commit *Def.: At most 3 active commits, change more than 10 updated attributes (typically also involves a single reed)*
Moderate	Moderate rate of heartbeat (active commits), moderate volume of activity *Def.: None of the rest, total change less than 90 updated attributes*
Focused Shot & Low	A couple of reeds and a few active commits, focused (mod. - high) change *Def.: Between 4 and 10 active commits, no more than 2 reeds*
Active	Frequent rate of heartbeat (active commits), high volume of activity *Def.: None of the rest, total change more than 90 updated attributes*

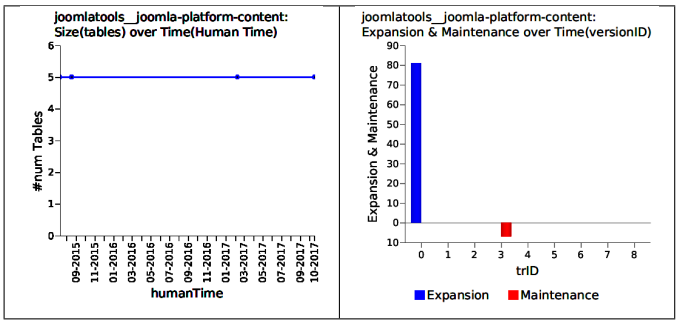


Figure 5. A typical example of an Almost Frozen schema: schema size over human time (left) and expansion vs maintenance over transitionId (right). There are 8 commits post the original version (mostly close to each other, thus overlapping each other on the left) and out of them, the only active commit involves the data type update of 3 attributes.

C. Hit and Freeze Evolution

There is a particular family of projects whose evolution demonstrates specific transitions with significantly higher amount of changes than the Almost Frozen taxon. The *Focused-Shot-and-Frozen* taxon is based around one or two such transitions, with often a single reed being practically the only change performed to the schema. The jRonak / Onlinejudge project in Fig. 6 depicts another form of schema evolution, again with a very restricted set of just a couple of active commits, and in this case, a small expansion of the set of tables of the schema.

This taxon's schemata do not change significantly and the amount of change is small in terms of attributes. In 36% of the projects, evolution involves attribute injections into existing tables (i.e., a flat schema line) and a small maintenance activity, all combined in a single active commit. In other words, a significant percentage of such projects simply focuses

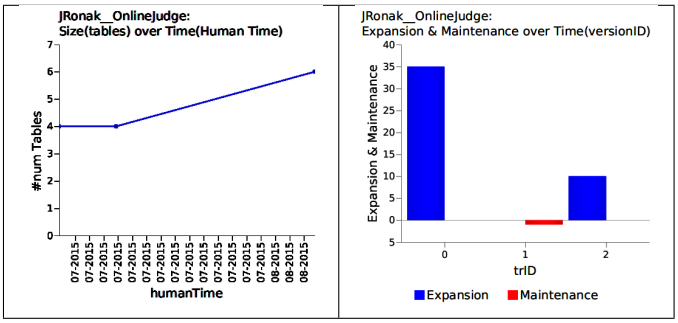


Figure 6. Example of a focused expansion of two tables

evolution in almost a single commit. 52% of the projects involve a single step-up in the schema line; on average the projects of the taxon insert 2 tables and remove 1 in their life.

Concerning project duration, out of the 25 projects, 9 lasted less than 1 year and another 6 less than 2 years. However, there are 11 projects (44%) which outlasted 2 years of project duration (PUP). In contrast, SUP has a median of 2 months and an average of 9 months. The commits concerning the DDL file amounted to 4% of the total commits.

D. The Timid Life of Moderate Evolution

Moderate evolution is the characteristic of 29 projects. There is a consistency in change, as the median Schema Update Period is 20 months with a median of 10 commits, 7 of them active, typically all of them turf (distinguishing the taxon from the subsequent ones), a median of two tables inserted and zero deleted and a median total change of 23 attributes. 65% of projects have a rise in the schema, 10% have a flat line and the rest of the projects have turbulent or dropping schema lines. In terms of project duration, 72% of the taxon's projects span more than 24 months, and 86% more than 12 months. The commits concerning the DDL file amounted to 5% of the total project commits.

E. Focused Change and Turf

Apart from a moderate and low-volume evolution of the schema, there is a distinct subcategory of not-insignificant heartbeat with focused change (via a couple of reeds) and a few extra active commits. We refer to this taxon as *Focused Shot and Low*. The taxon is characterized by one or two reeds and no more than 10 active commits. Change in this category comes to a large extent due to the "reeds" of focused activity, rather than the regular "turf-like" activity of small volume. Except for activity, the numbers are very similar to the moderate taxon: the median Schema Update Period is 17.5 months with a median of 10.5 commits, 6.5 of them active, with one reed, a median of 4.5 tables inserted and 2.5 deleted. This taxon is the second largest in activity volume, right after active projects, with a median total change rise to 71 attributes (significantly different from the previous taxa). In terms of project duration, 70% of the projects of this taxon span more than 24 months, and 75% span more than 12 months. The commits concerning the DDL file amounted to 6% of the total project commits.

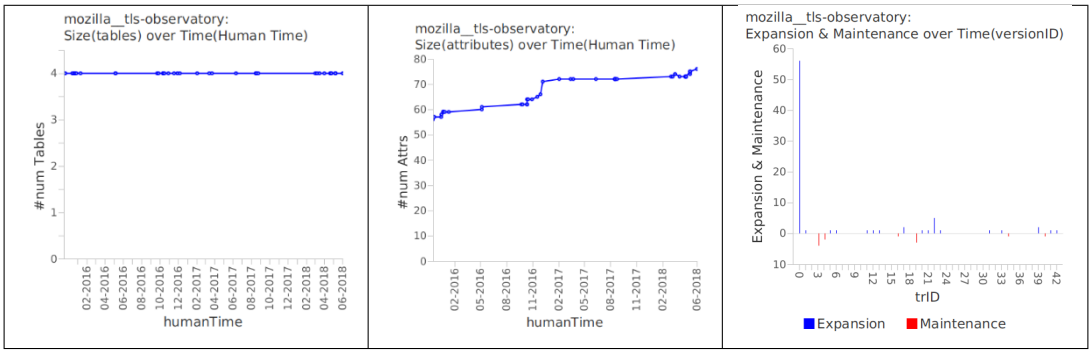


Figure 7. A typical example of a schema evolving with a moderate tempo: schema size over human time as tables (left), attributes (center) and heartbeat over transitionId for the mozilla/tls-observatory project. There are 43 commits (23 active) after the original version, mainly directed towards mild attribute injections, with different time density (as the middle figure demonstrates).

SUP in this category comes in two flavors: short and long schema update periods. In Fig. 8, the jasdel/harvester project has a very short SUP, with a couple of reeds and a two-step increase in the schema line. The TalkingData/OWL-v3 project is quite expressive of the family: the "reed" reaches 124 attributes of growth and 68 attributes of maintenance, practically including 90% of the project's post-V0 activity.

F. Schemata with high volumes of updates

Apart from the aforementioned, low-to-med activity taxa, there are also schemata that demonstrate *significant volume of updates*. The schemata of this taxon, although few, are the ones with the largest amount of tables involved, the largest SUP durations (31 months median), a heartbeat with a median of 36.5 commits, 22 active, 5.5 reeds and the rest turf, 24 tables inserted and 9 deleted and a median total activity of 254 attributes. In terms of project duration, 91% of the projects of this taxon span more than 24 months, and 95% span more than 12 months. The commits concerning the DDL file amounted to 6% of the total project commits. In other words, the behavior of the taxon demonstrates significantly higher volumes of change than the other taxa (Fig. 1, 2, 9).

It is important to note that in active projects, the heartbeat is not homogeneous. This has to do both with the frequency and the size of the change events. In terms of frequency, there are periods of systematic activity, with versions of small-to-medium size changes, periods of idleness, spikes of massive maintenance, growth and restructuring. The size of the schema is typically growing (50% of the cases with several steps, 9% with a single step), and, out of the 22 cases there are also 2 cases of flat schemata, 3 cases of massive drop of its size and 4 cases of turbulent evolution.

V. Taxa well-formedness

Are the taxa that we derived reasonable? Is it possible that two taxa actually hide the same behavior? To address these sanity-check questions, in this Section, we argue that our proposed taxa abide by three well-formedness criteria:

- Completeness:, i.e., covering all possible cases of activity behavior
- Disjointness: the characteristics of the different taxa are different, and each project can belong to exactly one taxon
- Internal Cohesion: within a taxon, the behavior of its projects is similar

It is easy to see that *Completeness* is covered both by the classification scheme of Fig. 3 at the logical level, and, as sanity check, by the projects at the instance level. Similarly, *Disjointness* is covered by mutual exclusion of the constraints of the classification scheme of Fig. 3, which dictate mutually disjoint taxa. Fig. 10 visually demonstrates why the rule-based classification makes sense, as the different taxa have a fairly small overlap.

Proving *Cohesion*, however, is not trivial. To this end, we compared the derived taxa both (a) as a set of taxa over the entire data set, and, (b) pairwise, on whether their statistical characteristics qualify them to be different. To this end, we assessed statistical significance of the taxa differences over (i) their number of active commits and (ii) their total activity.

We employed the Kruskal-Wallis test, in R, to test the differences of the defined taxa. The null hypothesis of the test is that the different taxa have the same median and thus the reported p-value is a measure on the rejectability of the null hypothesis. We excluded the totally frozen taxon, which is practically a special case of the Almost Frozen, from this analysis. The overall assessment of the Kruskal-Wallis test for the entire data set for the activity measurements produces a Kruskal-Wallis chi-squared = 178.22, df = 5, p-value $<$ 2.2e-16 and for Active Commits Kruskal-Wallis chi-squared = 175.27, df = 5, p-value $<$ 2.2e-16. In other words, it is extremely improbable that the taxa represent similar behaviors.

Note that our data are not normally distributed: Shapiro-Wilk normality test on total activity produces W = 0.24386 and a p-value $<$ 2.2e-16, i.e., it is extremely unlikely that activity data are normally distributed. Internally, within each taxon, the respective test revealed non-normality for all taxa for both active commits and total activity, with the exception of active commits for the case of Focused Shot and Low.

Extra statistical evidence. To reinforce the above conclusions, we performed extra statistical tests. First, we compared

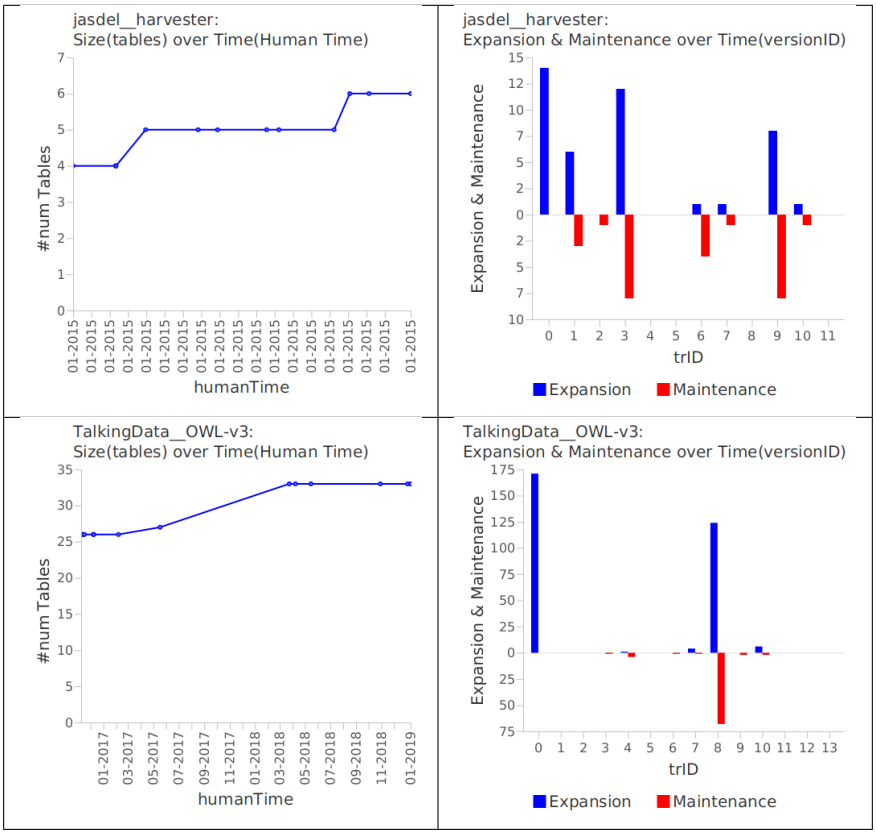


Figure 8. Examples of focused maintenance: a two-step schema increase accompanied with a few turf commits (top) and a very large reed, accompanied by very low change (bottom).

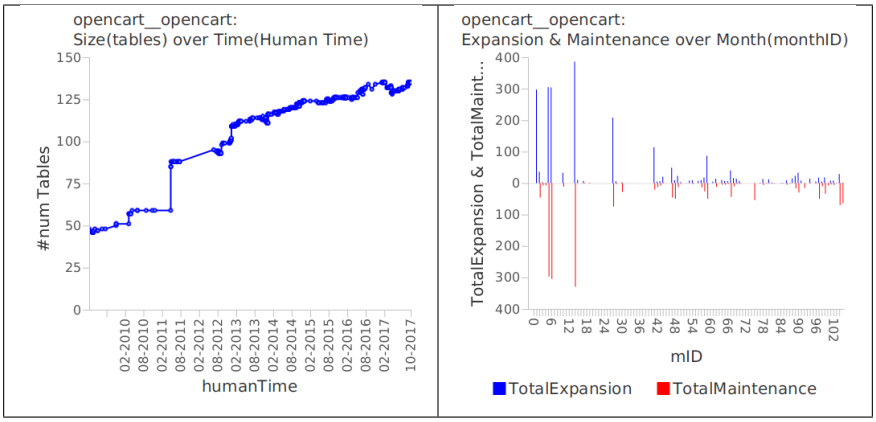


Figure 9. Example of high, systematic activity: the schema is being augmented over time via a systematic heartbeat that comes either with large spikes, or with constant turf and minor increases, without excluding periods of idleness. Observe that in this figure, the schema size is depicted over human time, whereas the heartbeat over the aggregated monthly changes. See Fig. 1 and 2, too, for more active schemata.

the taxa pairwise via a Kruskal-Wallis test. In Fig. 11, we report the p-values of the respective test: the lower left triangle refers to the active commits and the upper right triangle to the total activity. Assuming an acceptance threshold of 5%, *the test reveals that the differences between taxa are significant, with the exception of two cases*.

More concretely, the only cases where we cannot reject the null hypothesis are (a) the similarity of moderate with focused shot and frozen for their activity, and (b) the similarity of moderate with focused shot and low for their active commits. For both cases, however, the two involved taxa demonstrate significant difference in the other measures. Specifically:

- For the case of Moderate with Focused Shot and Frozen: whereas the overall number of affected attributes appears to be similar, the number of active commits is significantly different - in other words, whereas the Moderate schemata demonstrate a turf-oriented, more frequent evolution activity, the Focused Shot and Frozen schemata

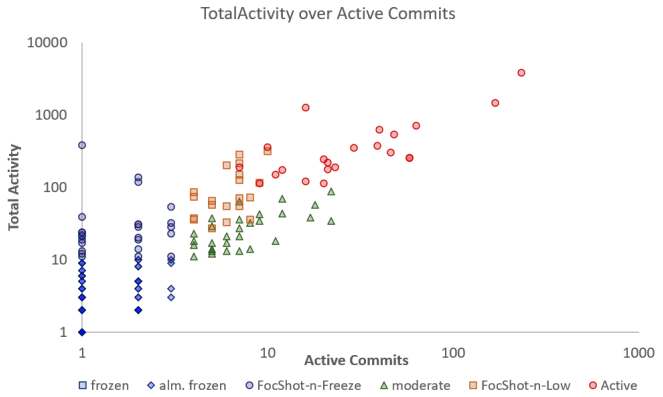


Figure 10. Project profiles in terms of active commits and activity. Frozen are not shown due to the logarithmic nature of the axes; almost frozen (blue diamonds) are lower left; focused shot & freeze (blue circles) are upper left; moderate (green triangles) occupy the center; focused shot & low (orange squares) complements moderate in the upper center with higher volumes of activity and moderate heartbeat; active projects (red circles) are at the upper right. Naturally, the borders are not completely separated; however, the original qualitative discrimination fits quite well with the rule-based discrimination of Fig. 3.

group a "similar" volume of activity in no more than 3 active commits. Thus, the difference lies in the active commit part.

- For the case of Moderate with Focused Shot and Low: whereas the overall number of affected attributes appears to be quite different, and the Focused Shot and Low schemata demonstrate significantly higher amounts of activity, the number of active commits is similar: in other words, the schemata of moderate activity lack the reeds of high activity that the Focused Shot and Low schemata demonstrate, and who drive their activity to larger heights. This resonates quite well with our intuition of treating Focused Shot and Low as a special case of Moderate evolution in terms of heartbeat, but whose reeds reach high activity volume.

In other words, we see that there is not a single pair of taxa that demonstrates similar behavior in both the number of active commits and total activity. Based, thus, on all the above discussion, we argue that the taxa that our analysis derived are pairwise different.

To further strengthen the statistical evidence, we also computed the quartiles of total activity and active commits for each taxon (Fig. 12). We also depicted these numbers in the double box plot shown in Fig. 13. The separation of the taxa demonstrates overlaps and "grey zones", however, the shape and placement of the boxes is quite reassuring. Specifically:

- The active taxon is very far from the rest. They are just 22 projects, but really far apart from the other taxa (see the legend of the Fig. 13 for details)
- The frozen taxa are quite close (also by definition) one to another. However, due to their larger populations, the separation can be justified: frozen with zero active commits and activity are by definition a taxon of its own. The most populous, Almost Frozen, is really close; however, we discriminate it from Frozen as having even such a small activity. Focused Shot and Frozen is also close both to Almost Frozen and to Moderate, however the shape of its heartbeat is quite different from both.

Is it possible that a taxon is not cohesive? The cohesion of the taxa is demonstrated by the double box plot of active commits vs activity, for the proposed taxa. The following observations are due:

- All taxa are heavily biased towards lower values, for both active commits and activity.
- The 3 most frozen taxa are really clustered in very cohesive boxes. This has to do both with the box and the whiskers of the box plot, with one exception: the 6 projects of the upper quartile of Focused Shot and Frozen, that spans from 31.5 attributes to 383. This should also be assessed under the prism of their population: there are 34 Frozen, 65 Almost Frozen (largest category and smallest distribution of all), 25 Focused Shot and Frozen projects, which means that there is almost an inverse relationship between population and surface of the box in the plot (remember that Moderate projects are 29, Focused Shot and Low are 20, and Active are 22 projects).
- The only box that spans a significant amount of surface in Fig. 13 is the taxon with the smallest population, Focused Shot and Low. However, there is no other taxon really, around the area of its high values, (there are only 3 Focused Shot and Frozen projects above 55 attributes of activity, and the entire taxon of Active, however at different areas of active commits, i.e., height in the chart).
- Finally, the Active taxon,with its just 22 projects of high activity, is very far apart from the rest. Both the location of the projects in the 2D area of Fig. 13, and its statistical tests, emphasize its separation from the other taxa.

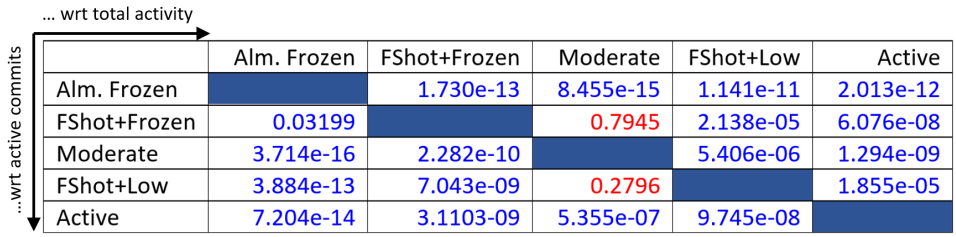
VI. Concluding Remarks

In this study, we attack the problem of understanding the characteristics of schema evolution by performing the largest empirical study ever performed in the domain of Free Open Source Software projects. Our contributions involve:

- A clear definition of the nomenclature and the important measures of the problem.
- The collection of a significantly large dataset, with a population of 195 studied projects (almost 20 times larger than the largest study in the literature).
- The study of the heartbeat of schema evolution as well as the identification of taxa of schema evolution (both for the first time ever), with taxa being distinct classes of archetypal behavior of a schema over its lifetime.
- The answering of several research questions around the nature of schema evolution, for the first time in the related literature (see next too).

Coming back to our original research questions, we can summarize our findings as follows.

RQ1. Is schema evolution present extensively? For a very large percentage of projects, schema evolution is practically



	Alm. Frozen	FShot+Frozen	Moderate	FShot+Low	Active
Alm. Frozen		1.730e-13	8.455e-15	1.141e-11	2.013e-12
FShot+Frozen	0.03199		0.7945	2.138e-05	6.076e-08
Moderate	3.714e-16	2.282e-10		5.406e-06	1.294e-09
FShot+Low	3.884e-13	7.043e-09	0.2796		1.855e-05
Active	7.204e-14	3.1103-09	5.355e-07	9.745e-08	

Figure 11. p-values of the Kruskal-Wallis test for the pairwise comparison of the taxa of our study: the lower left triangle refers to the active commits and the upper right triangle to the total activity values.

Active Commits	Alm. Frozen	FS_Frozen	Moderate	FS_Low	Active
MIN	1	1	4	4	7
Q1	1	1	5	5	15
Q2	1	2	7	6,5	22
Q3	2	2	10	7	50,5
MAX	3	3	22	10	232

Activity	Alm. Frozen	FS_Frozen	Moderate	FS_Low	Active
MIN	1	11	11	27	112
Q1	1	15,5	15	41,5	177
Q2	3	23	23	71	254
Q3	5	31,5	37,5	143	558,5
MAX	10	383	88	315	3845

Figure 12. Quartiles of activity and active commits for the different taxa.

absent. As already mentioned, out of the largest possible collection of 327 projects that we came up, 40% had no evolution whatsoever, 10% had different versions but no schema changes at the logical level and 20% were almost frozen.

We believe that our empirical evidence is important exactly because it refutes the traditional belief that schema evolution is extensive (also reported in the literature, e.g. [11]) and replaces it with a new perspective: *schema evolution is mostly absent from the typical Free Open Source Project, and emphatically present only in a small percentage of projects with an active profile of continuous schema maintenance*. The massive and widespread nature of this absence drives us to conjecture that there is a strong possibility that the idiosyncrasy of schema evolution is that this "absence" is not due to the lack of its necessity, but rather due to its difficulty (see also [20])!

RQ2. Are there archetypal patterns of schema lives? For the first time in the related literature, we *define and study the heartbeat* and *present patterns of schema evolution*. Frozen projects with no change whatsoever and Almost Frozen with few active commits and small change constitute 17% and 33% of the studied population – i.e., half the projects of the study. Focused Shot and Frozen projects with almost no activity other than a single spike of change arise to 13% and Focused Shot and Low projects, with a couple of high-volume reeds of evolution and less than 10 active commits overall another 10%. Projects of constant rate of schema maintenance involve (a) Moderate projects, with less than 90 attributes changed in their lifetime, and a fraction of 15% of the population, and, (b) Active ones, with frequent change and high volumes of it, amounting to an 11% of the population.

RQ3. What are the demonstrable properties of schema evolution, in terms of volume, frequency and important characteristics? One of our main contributions is the clear specification of important measures of evolution. Fig. 4 summarizes the most important of them. Volume of change is measured in affected attributes (as the universal unit of change) that can capture both table births and deaths, but also intra-table changes. The number of commits of the DDL file and active commits, that also include changes to it, is a demonstrator of the frequency of change. Reeds and turf commits characterize the density of change. Tables inserted and deleted, as well as tables at start and end characterize the resizing of the schema at a coarser level of detail. Specific questions around the behavior of these measures are also answered.

- *How is the amount and frequency of change demonstrated?* The frequency of change is really low. Out of the 195 projects studied, 124 (64%) have 0 - 3 active commits. The delta change in terms of tables is significantly small in almost all categories except for the active one (practically between zero and two tables in most categories). Deletions are extremely rare, in particular. Overall, with the exception of the active category, the change in the number of tables is quite small.
- *Is change mostly concentrated in few commits of focused change?* Focused commits of change do exist, and they are also apparent in both moderate and active projects, but also in low-activity projects (where although we do not count reeds, there are small-volume active commits that concentrate the small change of a project). However, focused change is not a recurring practice. Most taxa come with less than 2 reeds, and only active projects practically surpassing this number.

Open paths for research. As already mentioned in [20], the absence of industrial schema histories (and thus our reliance on FOSS projects only) makes the problem of acquiring publicly available schema histories from the industry quite an improbable scenario. In the absence of such data, however, we can continue research to test the existence of patterns at the table level, to extract the treatment of constraints (esp., foreign keys) in FOSS projects and to qualitatively study gravitation to rigidity at more depth.

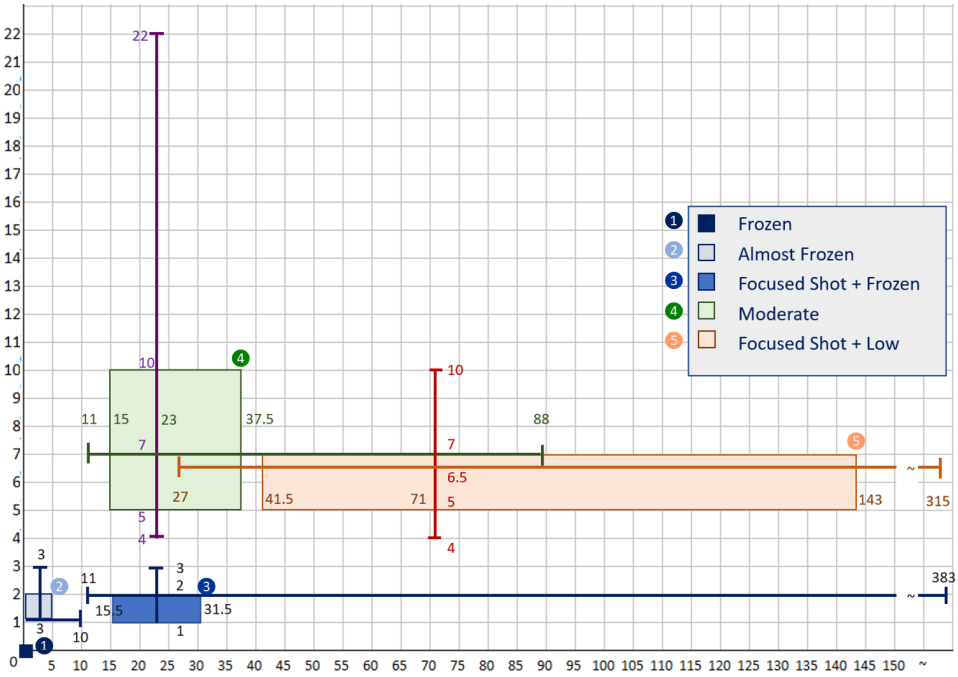


Figure 13. Double box plot for active commits and activity for the different taxa. The horizontal axis depicts the total activity (in number of affected attributes) and the vertical axis depicts the number of active commits. Each taxon has a rectangle with the Q1 and Q3 quartiles at its edges, for both dimensions. A cross formed by lines passing from the Q2 (median) for each dimension is also annotating the box of each taxon. The min and max values of each taxon for the respective dimension mark the limits of each line. For example, for the moderate taxon, activity has: min: 11, Q1: 15, Q2: 23, Q3: 37.5, max 88, and, active commits have: min: 4, Q1: 5, Q2: 7, Q3:10, Q4: 22. The active taxon is not shown, as its activity lies far away from the rest: Q1: 177, Q3: 5585 and its active commits with Q1: 15 and Q3: 50.5.

References

[1] M. Hartung, J. F. Terwilliger, and E. Rahm, "Recent advances in schema and ontology evolution," in *Schema Matching and Mapping*, ser. Data-Centric Systems and Applications, Z. Bellahsene, A. Bonifati, and E. Rahm, Eds. Springer, 2011, pp. 149–190.

[2] P. Manousis, P. Vassiliadis, A. V. Zarras, and G. Papastefanatos, "Schema evolution for databases and data warehouses," in *5th European Summer School on Business Intelligence , eBISS 2015*, ser. Lecture Notes in Business Information Processing, vol. 253. Springer, 2015, pp. 1–31.

[3] C. Curino, H. J. Moon, A. Deutsch, and C. Zaniolo, "Automating the database schema evolution process," *VLDB J.*, vol. 22, no. 1, pp. 73–98, 2013.

[4] K. Herrmann, H. Voigt, J. Rausch, A. Behrend, and W. Lehner, "Robust and simple database evolution," *Inf. Syst. Frontiers*, vol. 20, no. 1, pp. 45–61, 2018.

[5] R. E. Schuler and C. Kesselman, "A high-level user-oriented framework for database evolution," in *31st International Conference on Scientific and Statistical Database Management, SSDBM 2019, Santa Cruz, CA, USA, July 23-25, 2019*. ACM, 2019, pp. 157–168.

[6] D. Sjøberg, "Quantifying schema evolution," *Information and Software Technology*, vol. 35, no. 1, pp. 35–44, 1993.

[7] C. Curino, H. J. Moon, L. Tanca, and C. Zaniolo, "Schema evolution in wikipedia: toward a web information system benchmark," in *Proceedings of ICEIS 2008*, 2008.

[8] A. Cleve, M. Gobert, L. Meurice, J. Maes, and J. H. Weber, "Understanding database schema evolution: A case study," *Sci. Comput. Program.*, vol. 97, pp. 113–121, 2015.

[9] D.-Y. Lin and I. Neamtiu, "Collateral evolution of applications and databases," in *Joint Intl. Annual ERCIM Workshops on Principles of Software Evolution (IWPSE) and Software Evolution (Evol)*, 2009, pp. 31–40.

[10] S. Wu and I. Neamtiu, "Schema evolution analysis for embedded databases," in *2011 IEEE 27th International Conference on Data Engineering Workshops*, ser. ICDEW '11, 2011, pp. 151–156.

[11] D. Qiu, B. Li, and Z. Su, "An empirical analysis of the co-evolution of schema and code in database applications," in *2013 9th Joint Meeting on Foundations of Software Engineering*, ser. (ESEC/FSE), 2013, pp. 125–135.

[12] P. Vassiliadis, M. Kolozoff, M. Zerva, and A. V. Zarras, "Schema evolution and foreign keys: a study on usage, heartbeat of change and relationship of foreign keys to table activity," *Computing*, vol. 101, no. 10, pp. 1431–1456, 2019.

[13] I. Skoulis, P. Vassiliadis, and A. V. Zarras, "Growing up with stability: How open-source relational databases evolve," *Information Systems*, vol. 53, pp. 363–385, 2015.

[14] P. Vassiliadis, A. V. Zarras, and I. Skoulis, "Gravitating to rigidity: Patterns of schema evolution - and its absence - in the lives of tables," *Information Systems*, vol. 63, pp. 24–46, 2017.

[15] P. Vassiliadis and A. V. Zarras, "Schema evolution survival guide for tables: Avoid rigid childhood and you're en route to a quiet life," *Journal of Data Semantics*, vol. 6, no. 4, pp. 221–241, 2017.

[16] M. Klettke, H. Awolin, U. Störl, D. Müller, and S. Scherzinger, "Uncovering the evolution history of data lakes," in *IEEE International Conference on Big Data, BigData 2017, Boston,A, USA, December 11-14, 2017*. IEEE Computer Society, 2017, pp. 2462–2471.

[17] S. Scherzinger and S. Sidortschuck, "An empirical study on the design and evolution of nosql database schemas," *CoRR*, vol. abs/2003.00054, 2020. [Online]. Available: https://arxiv.org/abs/2003.00054

[18] G. Gousios, "The ghtorent dataset and tool suite," in *Proceedings of the 10th Working Conference on Mining Software Repositories, MSR '13, San Francisco, CA, USA, May 18-19, 2013*. IEEE Computer Society, 2013, pp. 233–236.

[19] C. Bird, P. C. Rigby, E. T. Barr, D. J. Hamilton, D. M. Germán, and P. T. Devanbu, "The promises and perils of mining git," in *6th International Working Conference on Mining Software Repositories, (MSR)*. IEEE Computer Society, 2009, pp. 1–10.

[20] M. Stonebraker, R. C. Fernandez, D. Deng, and M. L. Brodie, "Database decay and what to do about it," *Commun. ACM*, vol. 60, no. 1, p. 11, 2017. [Online]. Available: https://doi.org/10.1145/3014349