A Historical Dataset of Software Engineering Conferences

Bogdan Vasilescu, Alexander Serebrenik
Model-Driven Software Engineering,
Department of Mathematics and Computer Science,
Eindhoven University of Technology, The Netherlands
{*b.n.vasilescu, a.serebrenik*}*@tue.nl*

Tom Mens
Software Engineering Lab,
COMPLEXYS Research Institute,
University of Mons, Belgium
tom.mens@umons.ac.be

***Abstract*—The Mining Software Repositories community *typically* focuses on data from software configuration management tools, mailing lists, and bug tracking repositories to uncover interesting and actionable information about the evolution of software systems. However, the techniques employed and the challenges faced when mining are not restricted to these types of repositories. In this paper, we present an *atypical* dataset of software engineering conferences, containing historical data about the accepted papers and the composition of programme committees for eleven well-established conferences. The dataset (published on Github at https://github.com/tue-mdse/conferenceMetrics) can be used, e.g., by conference steering committees or programme committee chairs to assess their selection process and compare against other conferences in the field, or by prospective authors to decide in which conferences to publish.**

I. Introduction

Researchers in the Mining Software Repositories (MSR) community explore a range of software engineering questions using software repository data as the primary source of information. The mined repositories typically include software configuration management, mailing list, and bug tracking repositories. By mining repositories, researchers have accomplished, e.g., a better understanding of software systems and team dynamics, or improved mechanisms to predict and identify bugs [1], all important achievements for software engineering.

In this paper we are interested in the *health* of the software engineering research community as portrayed by software engineering conferences. To this end, we propose a historical dataset containing information about the accepted papers, the numbers of submissions and the composition of programme committees (PC) for eleven well-established conferences. Using this dataset, conference steering committees or programme committee chairs can assess their selection process (e.g., *"how much turnover do we have in the PC?"*) and compare against other conferences. Similarly, prospective authors can decide to which conferences to submit (e.g., *"how open is a particular conference to newcomers?"*). Furthermore, when constructing the dataset we had to resolve several technical problems, such as aliases referring to the same person, hence the dataset can also benchmark performance of identity merging algorithms developed for traditional software repositories [2]–[6]. Other potential use cases are discussed in Section IV-B.

The remainder of this paper is structured as follows: Section II gives an overview of the data, its provenance, and the schema used to store it; Section III presents the methodology used to gather the data; Section IV presents example use cases; Section V discusses limitations, and Section VI concludes.

II. Description of the data

A. Overview

Numerous software engineering conferences are organised every year. Some of them have a wide scope (e.g., ICSE), while others are focused on specific subdomains of software engineering (e.g., ICSM, MSR). Since conferences may exhibit different behaviour (e.g., in terms of turnover, or openness) depending on their scope, we have chosen to include in our dataset a mixture of both narrow- and wide-scoped conferences, as listed in Table I.

For each conference, we record the list of papers accepted each year together with their authors. Papers part of the main track are marked as such. Moreover, we record the list of PC members and the number of submissions received each year (both currently only for the main track). Except for MSR (started in 2004), the data covers a period of at least ten years, as can be seen in Table I. Finally, we record the impact of conference series, an accepted *prestige* measure.

Since we are integrating data from different sources, the names of authors and PC members are not necessarily consistent, while it is critical to know the identities of persons, e.g., if we wish to check for signs of inbreeding. Mike Godfrey is, for instance, also known as Michael Godfrey, Mike W. Godfrey, or Michael W. Godfrey. To match multiple aliases used by the same person we performed identity merging, described in more detail in Section III.

To foster replicability [7] and encourage contributions from the community, we have publicised the dataset on Github, at https://github.com/tue-mdse/conferenceMetrics, together with the tooling used to create and query it.

B. Provenance

For all considered conferences, the data about accepted papers and their authors was extracted from the DBLP records [8]. Data about the composition of the PC and number of submissions received was retrieved from the websites of

TABLE I
CONFERENCES INCLUDED IN THE DATASET (WIDE-SCOPED CONFERENCES IN **BOLDFACE**).

Acronym	Full name	First edition[1]	Last edition
ASE	IEEE/ACM International Conference on Automated Software Engineering	1994	2012
CSMR	European Conference on Software Maintenance and Reengineering	1997	2012
FASE	International Conference on Fundamental Approaches to Software Engineering	1998	2012
FSE	ACM SIGSOFT Symposium on the Foundations of Software Engineering	1993	2012
GPCE	Generative Programming and Component Engineering	2000	2012
ICPC	IEEE International Conference on Program Comprehension	1997	2012
ICSE	International Conference on Software Engineering	1994	2012
ICSM	IEEE International Conference on Software Maintenance	1994	2012
MSR	Working Conference on Mining Software Repositories	2004	2012
SCAM	International Working Conference on Source Code Analysis & Manipulation	2001	2012
WCRE	Working Conference on Reverse Engineering	1995	2012

[1] This is the edition for which we could reconstruct the composition of the PC. Data about accepted papers may go back further in time.

each conference and online proceedings volumes[2]. For earlier editions, we used the Wayback machine[3] to analyse no-longer-available websites as well as announcements posted by conference organisers in Usenet newsgroups.

The conference series impact factor is computed based on the Simple H-INdex Estimator (SHINE) [9], considering the entire period 2000 to 2012. For a given conference series, an impact of 40 means that it has 40 papers, each with at least 40 citations during this period. The earliest SHINE data available is from 2000. Since computation of the h-index can be inaccurate for recent years (due to late propagation of citation information), we use the entire available history of conference citations since 2000 until 2012.[4].

C. Schema

To store the data we use a MySQL database with the schema depicted in Figure 1. For each `person` we store an *id* and a unique *name* (after merging their different aliases). To increase compatibility with third-party usage of the database, we also normalise all person names by stripping accents and diacritics (e.g., Daniel Germán becomes Daniel German). Persons can author `papers`, as recorded in the many-to-many `authorship` relationship. Alternatively, persons can serve on programme committees, as recorded in the many-to-many `pc_membership` relationship. Currently, only programme committee members for the main tracks have been included.

An entry in the `papers` table contains an *id*, *year*, and *title*, the *conference_id* of the conference where the paper was published, the *pages* at which it appeared in the proceedings and the number of pages *num_pages*, a flag *main_track* signalling whether it was part of the main track or not (manually assigned), and the title *session_h2* (e.g., Technical Research) and subtitle *session_h3* (e.g., Fault Handling) of the session during which it was presented (if available in DBLP). Although generally not available for all conferences, we have chosen to include the session titles in the database whenever possible since these could potentially be used to automatically filter *interesting* papers (e.g., filter out industry track papers because they were selected for inclusion by a different PC, or mine the number of papers within a particular subtopic).

Finally, in the `conferences` table we store the *acronym* and *name* of each conference together with its *impact* factor (one value per conference), and in the `submissions` table we store the *number* of submissions received by conferences each *year* (again currently only for the main track).

III. EXTRACTION METHODOLOGY

The curation of this dataset was an incremental process, involving both automatic and manual steps. To ensure replicability and facilitate extensibility (cf. best practices in [7]), the linked Github repository contains both snapshots of the "raw" data (i.e., the data used as input for the various processing stages), as well as the Python scripts used during the process. All the files and data described below are part of the repository.

A. Data extraction

Data about the papers published at each conference was automatically extracted into CSV format from the internal backup files of DBLP. As opposed to the publicly-available XML data dump[5], the XML-like backup files also contain information about the session titles, as do the DBLP conference webpages[6], which could have been used instead. In contrast, the composition of the PCs per year was manually extracted (again into CSV format) from the websites of each conference, or old Calls for Papers retrieved from Usenet newsgroups. Extraction of submission counts and impact factors followed a similar manual process.

B. Identity merging

Since we are integrating data from different sources, inconsistencies with the names of authors and PC members can occur, both within each dataset as well as across datasets: the author names *may* contain inconsistencies since they

[2] Tao Xie also maintains a website with number of submissions and acceptance rates for software engineering conferences at http://people.engr.ncsu.edu/txie/seconferences.htm

[3] http://archive.org/web/web.php

[4] Note that the impact of younger conferences such as MSR or SCAM can thus be underestimated.

[5] http://dblp.uni-trier.de/xml/

[6] For example, http://www.informatik.uni-trier.de/%7Eley/db/conf/icse/icse2012.html

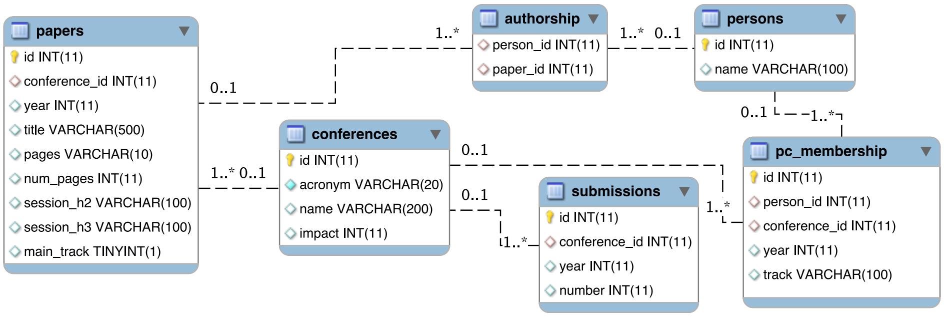

Fig. 1. Database schema.

originate from DBLP records of different editions of different conferences; the PC member names *are likely to* contain inconsistencies since they were retrieved from different websites altogether. Merging aliases associated to the same individual is a well-known problem in the MSR community [2]–[6]. We have chosen not to perform identity merging (disambiguation) fully automatically, since it is known that all existing approaches result in false positives and false negatives [6]. Instead, we pursue a semi-automatic approach, enhanced by manual validation of the results.

First, since DBLP already employs disambiguating mechanisms for its list of authors [10], we reverse engineer the list of known aliases for each author from the XML data dump. As a result we find, e.g., that Harald C. Gall is also known as Harald Gall, but Mike Godfrey and Michael W. Godfrey are still recorded as two different authors. Next, to aid with merging identities of PC members, we compute for each conference the union of yearly author sets (coming from DBLP) and the union of yearly PC member sets (coming from the different conference websites), then compare the latter to the former. The assumption here is that PCs are typically composed of (a large fraction of) established authors from within the respective communities, so we should be able to recognise the names of (most) PC members in the author lists. Whenever names of PC members do not match the authors list, we check for authors bearing the same last name (or variations thereof in cases where we suspect that stripping accents and diacritics might have affected the spelling, e.g., ü and u/ue). In case of manually validated candidate mergers, we incrementally update a lookup table with the *(alias, identity)* pairs. Otherwise, we repeat the comparison with the set of authors from all conferences considered, and ultimately the set of all authors from DBLP, until reaching a fixed point. Finally, using the list of DBLP aliases and the lookup table, we reconcile all the names of authors and PC members.

C. Model-driven database

To ensure flexibility and extensibility of the database, we follow a model-driven approach: we model the data as Python classes and use the SQLAlchemy [11] object relational mapper to generate the actual database. This facilitates co-evolution of the database schema with the data model, populating the database, and post-processing the query results.

IV. Example use cases

A. Conference "health" assessment

This dataset was initially intended for conference steering committees or programme committee chairs to assess their selection process, or for prospective authors to help decide to which conferences to submit their work. To facilitate such tasks, we have included a number of example metrics and queries in the repository[7], out of which we illustrate two. Both are computed using a sliding window, to better control for differences in conference age.

a) PC turnover: computed for conference c in year y with respect to k previous years (denoted $RNC(c,y,k)$), is the fraction of PC members for c in y that have never served on the PC between $y-k$ and $y-1$. Striking a balance between PC turnover and continuity is desirable. Inviting PC members from previous editions helps to ensure continuity and coherence. However, conferences are frequently subject to charters or guidelines that require PC renewal (e.g., the ACM SIGSOFT policy, applicable to ICSE, FSE and ASE, requires at least one-third of the PC members to change each year). Moreover, low turnover may impact variation in research approaches and negatively influence the scientific focus of a conference.

b) Inbreeding ratio: denoted $RAC(c,y,k)$, is the fraction of papers published at c in year y co-authored by PC members that served at least once between $y-k$ and y. Most software engineering conferences are based on a single-blind peer reviewing scheme, i.e., the reviewers know the names of the authors, but not vice versa. This may increase the risk of conferences becoming closed communities, and they may suffer to some extent from inbreeding [12], [13]. The evolution of $RAC(c,y,0)$, the fraction of papers co-authored by PC members each year, is depicted in Figure 2 (left).

Systä et al. [12] have observed negative linear correlation between $RAC(c,y,0)$ and $RNC(c,y,1)$ on a smaller sample of conferences: *"the less there is PC turnover, the greater is the proportion of PC papers among the accepted papers"*. We have observed a similar phenomenon on our dataset (Figure 2, right): $RAC(c,y,0)$ and $RNC(c,y,1)$ show a moderately-strong negative linear correlation $(r = -0.61, p < 2.2e-16)$.

[7] For a complete list see /metrics/metrics.md in the repository

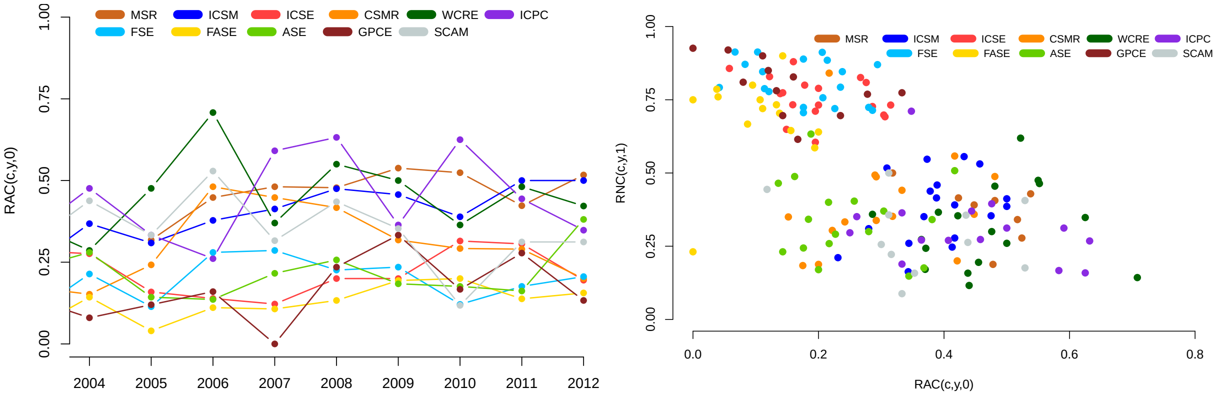

Fig. 2. *Left:* Evolution of the inbreeding ratio, since 2004. *Right:* Negative relation between $RAC(c, y, 0)$ and $RNC(c, y, 1)$.

B. Other applications

The dataset can also be used for other applications than its initial purpose. We can think of social network analysis (e.g., *"how does the collaboration structure vary between the different sub-communities in software engineering?"*, akin to [14]), gender issues (e.g., *"which software engineering sub-communities are more gender-unbalanced?"*, after inferring gender akin to [15]), recognition of community service (e.g., *"what is Alice's PC h-index?"*, the maximum number x of conferences such that Alice served at least x times on the PC of each, akin to [16]), specialisation of authors (e.g., *"which authors prefer to specialise in one subtopic rather than covering different ones?"*, cf. [17], [18]) or benchmarking identity merging algorithms (e.g., *"how well does my algorithm perform on this dataset in comparison to other approaches?"*) to name a few.

V. Limitations

In its current state, the dataset contains detailed information only about papers and PC members from the main track of each conference. To facilitate extensions, we have followed a model-driven approach when designing and populating the database (it is created automatically from a data model and CSV input files using an object relational mapper), and we include all the relevant tooling in the Github repository. However, extending the dataset remains challenging, since: (i) as opposed to authors and papers, composition of the PC cannot be automatically retrieved from DBLP records; (ii) names of authors and PC members are likely to be inconsistent, hence identity merging is required; (iii) track information cannot generally be inferred from DBLP records (but session titles and page numbers may help with filtering, when available). However, the most likely cause of data errors remains the identity merging process, since it cannot yet be performed reliably using automatic techniques, but instead requires subjective reasoning to decide which aliases to merge.

VI. Conclusions

We presented a database about the accepted papers and the composition of programme committees for eleven well-established software engineering conferences. The dataset can help conference steering committees or programme committee chairs to monitor the "health" of their conferences, and compare to other venues in the field. Although operating at meta level, curating this dataset faced traditional MSR challenges such as identity merging or replicability.

References

[1] A. Hassan, "The road ahead for mining software repositories," in *Frontiers of Software Maintenance, 2008. FoSM 2008.* IEEE, 2008, pp. 48–57.

[2] G. Robles and J. M. González-Barahona, "Developer identification methods for integrated data from various sources," in *MSR*. ACM, 2005.

[3] C. Bird, A. Gourley, P. T. Devanbu, M. Gertz, and A. Swaminathan, "Mining email social networks," in *MSR*. ACM, 2006, pp. 137–143.

[4] W. Poncin, A. Serebrenik, and M. G. J. van den Brand, "Process mining software repositories," in *CSMR*, T. Mens, Y. Kanellopoulos, and A. Winter, Eds. IEEE Computer Society, 2011, pp. 5–14.

[5] E. Kouters, B. Vasilescu, A. Serebrenik, and M. G. J. van den Brand, "Who's who in Gnome: Using LSA to merge software repository identities," in *ICSM*. IEEE Computer Society, 2012, pp. 592–595.

[6] M. Goeminne and T. Mens, "A comparison of identity merge algorithms for software repositories," *Science of Computer Programming*, 2011.

[7] G. Robles, "Replicating MSR: A study of the potential replicability of papers published in the mining software repositories proceedings," in *MSR*. IEEE, 2010, pp. 171–180.

[8] "The DBLP computer science bibliography," http://dblp.uni-trier.de, accessed February 2013.

[9] "Simple h-index estimation," http://shine.icomp.ufam.edu.br/index.php, accessed February 2013.

[10] M. Ley, "DBLP—some lessons learned," *PVLDB*, vol. 2, no. 2, pp. 1493–1500, 2009.

[11] R. Copeland, *Essential SQLAlchemy*. O'Reilly, 2008.

[12] T. Systä, M. Harsu, and K. Koskimies, "Inbreeding in software engineering conferences," http://www.cs.tut.fi/~tsysta/, accessed October 2012.

[13] J. Crowcroft, S. Keshav, and N. McKeown, "Viewpoint: Scaling the academic publication process to internet scale," *Commun. ACM*, vol. 52, no. 1, pp. 27–30, 2009.

[14] A. E. Hassan and R. C. Holt, "The small world of software reverse engineering," in *WCRE*. IEEE, 2004, pp. 278–283.

[15] B. Vasilescu, A. Capiluppi, and A. Serebrenik, "Gender, representation and online participation: A quantitative study of Stackoverflow," in *International Conference on Social Informatics*. ASE, 2012.

[16] A. Capiluppi, A. Serebrenik, and A. Youssef, "Developing an h-index for OSS developers," in *MSR*. IEEE, 2012, pp. 251–254.

[17] B. Vasilescu, A. Serebrenik, and M. G. J. van den Brand, "You can't control the unfamiliar: A study on the relations between aggregation techniques for software metrics," in *ICSM*. IEEE, 2011, pp. 313–322.

[18] A. Serebrenik and M. G. J. van den Brand, "Theil index for aggregation of software metrics values," in *ICSM*. IEEE, 2010, pp. 1–9.