Message from the MSR 2023 Mining Challenge Track Co-Chairs

The Mining Software Repositories (MSR) challenge started in 2006 and is designed to attract young researchers, including students to the MSR community. The 18th edition of the challenge at MSR 2023 focused on Global Software Supply Chain (GSSC) data, a giant dataset, and an accompanying World of Code (WoC) infrastructure that collects, curates, and cross-references data from nearly all public version control systems.

GSSC data version U that participants used during the challenge was collected based on updated and new repositories from GitHub, GitLab, Bitbucket, and dozens of other forges identified. This version contains 173M git repositories. over 3.1B commits, 12.6B trees, and 12.5B blobs and occupies over 250TB. Copying the entire dataset was thus no possible.. Instead, participants of the mining challenge used the Digital Archeology cluster that provides ample storage and computational resources to conduct their analysis or to conduct pre-filtering to select a subset of the dataset for their research contribution.Using this resource consequently requires training. For this, we held a kick-off session on October 27, 2022, via Zoom with approximately 35 participants registered. The kick-off started with a webinar explaining the basic structure of World of Code and a Q&A session. Participants then could present a project idea or a research question they would like to work on for the mining challenge and find collaborators. The entire session was recorded and shared. We also provided support via a Discord channel (83 participants) to address questions from the participants.

About the WoC Infrastructure

The primary data items in WoC (Level 1) are git objects retrieved from git repositories. These include commits (transactions representing changes to the source code), trees (representing folder structure), blobs (versions of the source files), and tags (specific commits identified as releases). Building on these data items WoC additionally provides access to cross-references or maps that connect primary git objects with projects, authors, files, and packages used in individual blobs, calculation of blobs created by a commit, and so forth (Level 2). WoC also provides access to curated data. The primary areas of curation include deforking of projects, aliasing author IDs, computing additional attributes for blobs by parsing file content to identify dependencies in 17 programming languages, and by inferring developer gender using leading commercial service (Level 3). Finally, WoC also provides summaries in a mongodb that includes various statistics concerning projects, authors, and APIs such as the activity rate, monthly activities, and core teams (Level 4).

For portability, ease of access, and to improve performance of operations sweeping the entire dataset, all (except the summary and raw level) datasets are also provided as flat files sorted and partitioned by key to facilitate the use of common Unix command line tools such as `grep`, `sed`, `awk`, `join`, `sort`, `uniq`. Thus, a common Unix tool-chain can be used to calculate transitive or more complicated relationships or conduct any other downstream analysis of WoC data.

Fast access to an arbitrary value by key for all cross-references is provided by the `getValues` Unix command created for WoC. This command follows Unix conventions by reading from standard input and writing to standard output with the type of map, for example, author to blob (a2b) or aliased author to blob (A2b) provided as the parameter. Queries involving under one million keys can be done via `getValues`, but larger queries can be more efficiently computed using the Unix `join` command on the corresponding flat files.

Challenge

The challenge was open-ended: participants could choose the research questions that they found most interesting. Our suggestion was to consider problems that are not centered on a specific project or a set

of projects but, instead, would exploit the completeness, curation, and cross-referencing capabilities of WoC.

1. WoC is designed to measure three types of software supply chains (code dependencies, code copying, and author-code knowledge transfer). All three pose unique risks and benefits. Participants could Investigate these risks and benefits. Software supply chains underlie many topical questions, such as vulnerabilities, code provenance, bill of materials, and many others.
2. Research questions that require to construct, sample, or analyze the global network of source code, APIs, people, and projects, and to filter subsets by time or content. Participants could, e.g. investigate where a particular piece of code came from, where and when a particular API was introduced and what projects or people use it, and what projects a particular developer worked on.
3. Determining the global context. A traditional MSR analysis tends to focus on a specific set of projects, as only project-specific data needs to be obtained. Often critical context of the elements in such datasets is lost, such as actions of developers, activities associated with the code, and usage of APIs *external to the specific set of projects*. WoC allows recovery and quantification of such global context.
4. Avoiding convenience sampling. Level-4 data provides detailed summaries of projects, APIs, and developers and could serve as a basis for selecting samples needed to conduct many kinds of natural experiments.
5. Exploiting curation at a global scale. The curation level in WoC solves common MSR headaches by aliasing author IDs and deforking projects based on shared commits.
6. Linking/enhancing the WoC dataset itself. Participants are encouraged to combine WoC with other data and include the code for the collection and linking of the external data, as well as suggestions on how this data could be permanently integrated into WoC.
7. Unlike a static database, WoC enables the reconstruction of past states of the entire open-source software. Many contemporary quality, lead time, effort, task prediction models need the ability to reconstruct past states to avoid the pervasive problem of “data leakage.”
8. The dataset provides ready-data for questions such as why and how developers decide to reuse pre-existing software, and which type of the supply chain they choose (technical dependencies, copy-based, or reuse of the the ideas)?
9. The ability to reconstruct past states of OSS allows finding answers to questions such as “how to produce a widely used framework or library?” or “how to reduce the risk of changes in the upstream projects and how to reduce the risk for downstream projects.”

We asked the participants to carefully consider any ethical implications that stem from using the WoC data and other data sources and explicitly discourage the public exposure of personally identifiable information.

Publications

This year, the Mining Challenge Track attracted 11 paper submissions. The papers went through a rigorous review process. Every submission was reviewed by three members of the program committee, and an electronic discussion was held for all papers. Based on the reviews and discussion, 5 papers were accepted for publication and presentation at the conference.

MSR Mining Challenge 2023 Co-Chairs
Audris Mockus, University of Tennessee, Knoxville, Tennessee
Alexander Nolte, University of Tartu, Tartu, Estonia
James Herbsleb, Carnegie Mellon University, Pittsburgh, Pennsylvania