Socio-Architectural Analysis in Kaiaulu

Leilani Reich, Nicholas Lee, Malia Liu

Sponsor: Rick Kazman, Shidler College of Business

Stakeholder/Mentor: Carlos Paradis, Ph.D. in Computer Science

ICS 496 Capstone Project – Spring 2023

https://github.com/sailuh/kaiaulu

What is Kaiaulu?

Kaiaulu is an R package that helps with understanding evolving software development communities, and the artifacts (gitlog, mailing list, files, etc.) which developers collaborate and communicate with.

Problem

- Create reproducible mining software repositories pipeline
- Expose assumptions to mitigate threats to validity in experimental design
- Minimal path to data

Objectives

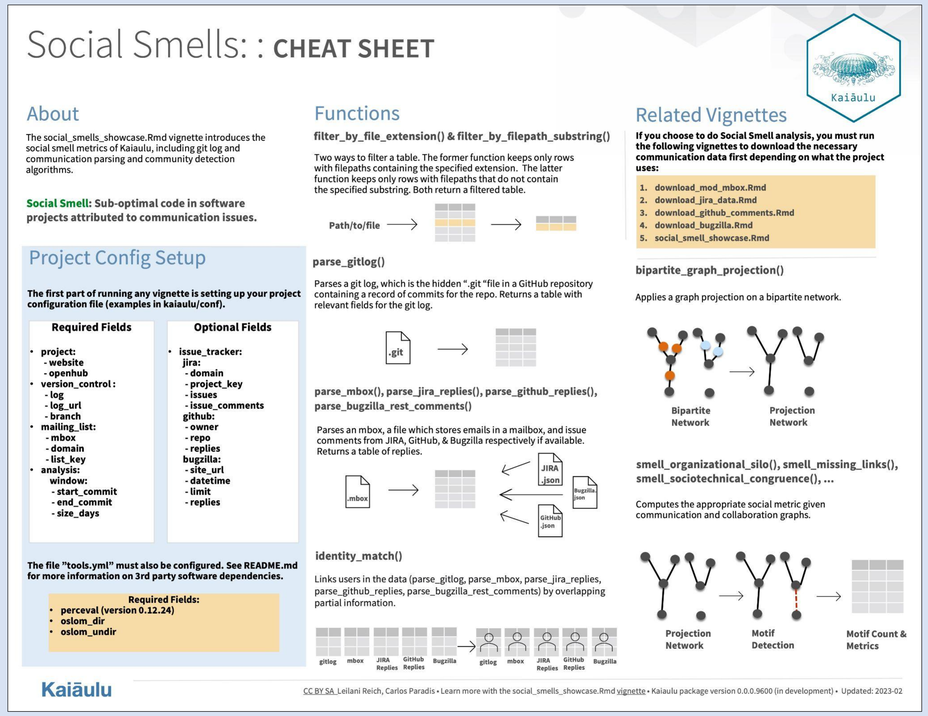
- Improve Kaiaulu usability via cheatsheets
- Better represent project communication network in mailing lists with inclusion of issue trackers like Bugzilla
- Define an API for DV8 to enable reproducible pipelines, notebooks, and visualizations

Methodology

- Project Management
  - Agile process
  - Github issues/pull requests/project board
- Roles
  - Malia - Technical/Documentation
  - Nico - Technical/Testing
  - Leilani - Technical/Communications/Meeting Scribe/ Requirements

Technologies Utilized

- **R Packages**
  - ***jsonlite, data.table, httr***
    - Retrieve, load, and read project data
  - ***testthat***
    - Unit testing
- **Third Party Tools**
  - ***Perceval***
    - Retrieve project data from Bugzilla sites
  - ***DV8***
    - Convert project data to design structure matrices



Social Smells: : **CHEAT SHEET**

Kaiāulu

About

The social_smells_showcase.Rmd vignette introduces the social smell metrics of Kaiaulu, including git log and communication parsing and community detection algorithms.

Social Smell: Sub-optimal code in software projects attributed to communication issues.

Project Config Setup

The first part of running any vignette is setting up your project configuration file (examples in kaiaulu/conf).

Required Fields

```
project:
  - website
  - openhub
version_control :
  - log
  - log_url
  - branch
mailing_list:
  - mbox
  - domain
  - list_key
analysis:
  window:
    - start_commit
    - end_commit
    - size_days
```

Optional Fields

```
issue_tracker:
  jira:
    - domain
    - project_key
    - issues
    - issue_comments
  github:
    - owner
    - repo
    - replies
  bugzilla:
    - site_url
    - datetime
    - limit
    - replies
```

The file "tools.yml" must also be configured. See README.md for more information on 3rd party software dependencies.

Required Fields:
- **perceval (version 0.12.24)**
- **oslom_dir**
- **oslom_undir**

Functions

filter_by_file_extension() & filter_by_filepath_substring()

Two ways to filter a table. The former function keeps only rows with filepaths containing the specified extension. The latter function keeps only rows with filepaths that do not contain the specified substring. Both return a filtered table.

Path/to/file →

parse_gitlog()

Parses a git log, which is the hidden ".git "file in a GitHub repository containing a record of commits for the repo. Returns a table with relevant fields for the git log.

.git →

parse_mbox(), parse_jira_replies(), parse_github_replies(), parse_bugzilla_rest_comments()

Parses an mbox, a file which stores emails in a mailbox, and issue comments from JIRA, GitHub, & Bugzilla respectively if available. Returns a table of replies.

.mbox → JIRA .json, Bugzilla .json, GitHub .json

identity_match()

Links users in the data (parse_gitlog, parse_mbox, parse_jira_replies, parse_github_replies, parse_bugzilla_rest_comments) by overlapping partial information.

gitlog, mbox, JIRA Replies, GitHub Replies, Bugzilla → gitlog, mbox, JIRA Replies, GitHub Replies, Bugzilla

Related Vignettes

If you choose to do Social Smell analysis, you must run the following vignettes to download the necessary communication data first depending on what the project uses:

1. **download_mod_mbox.Rmd**
2. **download_jira_data.Rmd**
3. **download_github_comments.Rmd**
4. **download_bugzilla.Rmd**
5. **social_smell_showcase.Rmd**

bipartite_graph_projection()

Applies a graph projection on a bipartite network.

Bipartite Network → Projection Network

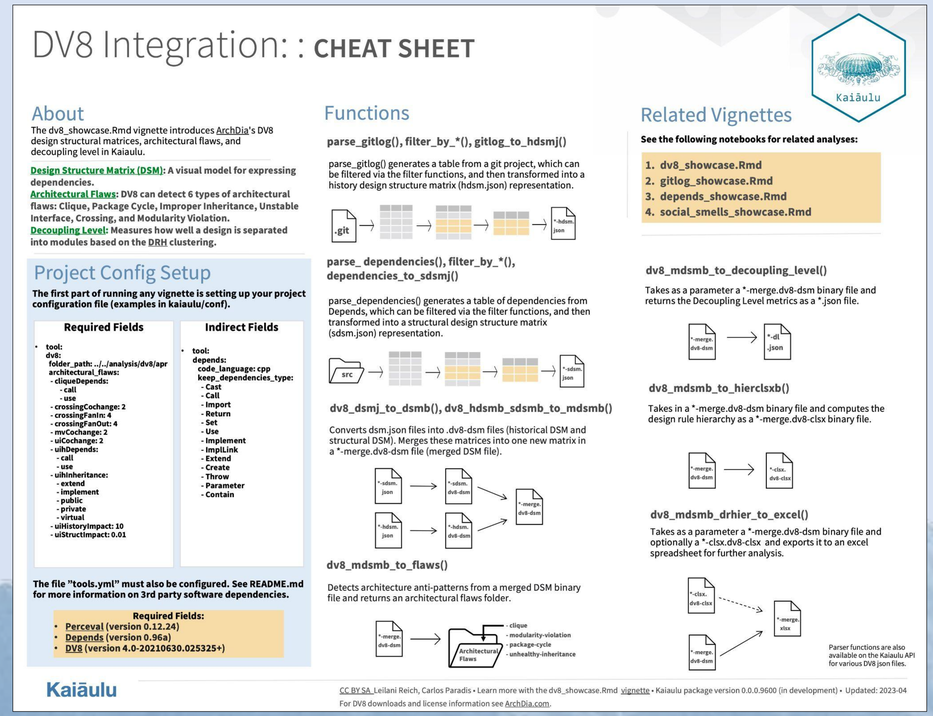
smell_organizational_silo(), smell_missing_links(), smell_sociotechnical_congruence(), ...

Computes the appropriate social metric given communication and collaboration graphs.

Projection Network → Motif Detection → Motif Count & Metrics

Kaiāulu

CC BY SA Leilani Reich, Carlos Paradis • Learn more with the social_smells_showcase.Rmd vignette • Kaiaulu package version 0.0.0.9600 (in development) • Updated: 2023-02



DV8 Integration: : **CHEAT SHEET**

Kaiāulu

About

The dv8_showcase.Rmd vignette introduces ArchDia's DV8 design structural matrices, architectural flaws, and decoupling level in Kaiaulu.

Design Structure Matrix (DSM): A visual model for expressing dependencies.
Architectural Flaws: DV8 can detect 6 types of architectural flaws: Clique, Package Cycle, Improper Inheritance, Unstable Interface, Crossing, and Modularity Violation.
Decoupling Level: Measures how well a design is separated into modules based on the DRH clustering.

Project Config Setup

The first part of running any vignette is setting up your project configuration file (examples in kaiaulu/conf).

Required Fields

```
tool:
  dv8:
    folder_paths: ../../analysis/dv8/apr
    architectural_flaws:
      - cliqueDepends:
          - call
          - use
      - crossingCochange: 2
      - crossingFanin: 4
      - crossingFanOut: 4
      - mvCochange: 2
      - uiCochange: 2
      - uihDepends:
          - call
          - use
      - uihInheritance:
          - extend
          - implement
          - public
          - private
          - virtual
      - uiHistoryImpact: 10
      - uiStructImpact: 0.01
```

Indirect Fields

```
tool:
  depends:
    code_language: cpp
    keep_dependencies_type:
      - Cast
      - Call
      - Import
      - Return
      - Set
      - Use
      - Implement
      - ImplLink
      - Extend
      - Create
      - Throw
      - Parameter
      - Contain
```

The file "tools.yml" must also be configured. See README.md for more information on 3rd party software dependencies.

Required Fields:
- **Perceval (version 0.12.24)**
- **Depends (version 0.96a)**
- **DV8 (version 4.0-20210630.025325+)**

Functions

parse_gitlog(), filter_by_*(), gitlog_to_hdsmj()

parse_gitlog() generates a table from a git project, which can be filtered via the filter functions, and then transformed into a history design structure matrix (hdsm.json) representation.

.git → *.hdsm.json

parse_ dependencies(), filter_by_*(), dependencies_to_sdsmj()

parse_dependencies() generates a table of dependencies from Depends, which can be filtered via the filter functions, and then transformed into a structural design structure matrix (sdsm.json) representation.

src → *.sdsm.json

dv8_dsmj_to_dsmb(), dv8_hdsmb_sdsmb_to_mdsmb()

Converts dsm.json files into .dv8-dsm files (historical DSM and structural DSM). Merges these matrices into one new matrix in a *-merge.dv8-dsm file (merged DSM file).

*.sdsm.json → *.sdsm.dv8-dsm; *.hdsm.json → *.hdsm.dv8-dsm; both → *.merge.dv8-dsm

dv8_mdsmb_to_flaws()

Detects architecture anti-patterns from a merged DSM binary file and returns an architectural flaws folder.

*.merge.dv8-dsm → Architectural Flaws: clique, modularity-violation, package-cycle, unhealthy-inheritance

Related Vignettes

See the following notebooks for related analyses:

1. **dv8_showcase.Rmd**
2. **gitlog_showcase.Rmd**
3. **depends_showcase.Rmd**
4. **social_smells_showcase.Rmd**

dv8_mdsmb_to_decoupling_level()

Takes as a parameter a *-merge.dv8-dsm binary file and returns the Decoupling Level metrics as a *.json file.

*.merge.dv8-dsm → *.dl.json

dv8_mdsmb_to_hierclsxb()

Takes in a *-merge.dv8-dsm binary file and computes the design rule hierarchy as a *-merge.dv8-clsx binary file.

*.merge.dv8-dsm → *.clsx.dv8-clsx

dv8_mdsmb_drhier_to_excel()

Takes as a parameter a *-merge.dv8-dsm binary file and optionally a *-clsx.dv8-clsx and exports it to an excel spreadsheet for further analysis.

*.clsx.dv8-clsx, *.merge.dv8-dsm → *.merge.xlsx

Parser functions are also available on the Kaiaulu API for various DV8 json files.

Kaiāulu

CC BY SA Leilani Reich, Carlos Paradis • Learn more with the dv8_showcase.Rmd vignette • Kaiaulu package version 0.0.0.9600 (in development) • Updated: 2023-04
For DV8 downloads and license information see ArchDia.com.

Tasks Accomplished

- **Documentation/Unit Testing**
  - Improvements to the Kaiaulu documentation (social smells notebook, function documentation, and wiki)
  - Created Kaiaulu cheatsheet
  - Added unit tests (integrated into GitHub actions) to various Kaiaulu modules
- **Bugzilla Wrapper/Crawler**
  - Bugzilla Wrapper (Perceval traditional & REST API backends)
    - Created downloader functions to call Perceval commands
    - Created parser functions to parse Bugzilla data
  - Bugzilla Crawler (REST API)
    - Created downloader functions to extract Bugzilla data directly from the REST API
    - Created parser functions to parse Bugzilla issues and comments data
- **DV8 Integration**
  - Created wrapper functions to call DV8 commands
  - Created parser functions and new data representations not available in DV8 for new analysis
  - Defined functionality to export any type of Kaiaulu graph to JSON for interoperability with tools like DV8

Challenges

- Getting familiar with the project's functions and documentation
- All team members relatively unfamiliar with the R language
- Moving away from Windows to Linux/OSX

Learnings

- Becoming adaptable between different OS
- How to be productive with the RStudio IDE
- Professional version control management with Git
- Software design analysis and flaws

Conclusion

Performing project analysis on repositories and issue trackers like Bugzilla can aid in tracking project dynamics, while analysis through DV8 can reveal architectural flaws and potential code relation problems. By analyzing data and identifying patterns, project managers can make informed decisions, mitigate risks, and ultimately improve project performance. Therefore, these newly integrated analysis capabilities are a valuable addition to Kaiaulu.