

Project report on ic-depress git-scm plugin

authors: Sławomir Kapłoński, Tomasz Kuzemko

First milestone (17.03.2013)

About project

Our project is focused on getting process metrics [1] from GIT SCM. In literature about that topic [1] there are couple of metrics which can be collected from such repository's log:

- Number of Revisions (NR) - it represents number of revision of class in project,
- Number of Distinct Committers (NDC) - number of authors of class (developers who commits changes to such class),
- Number of modified lines (NML) - this metric counts lines which are added or removed in given commit in class,
- Is New (IN) - this metric show whether given class was existed in previous commit or not
- Number of Defects in Previous Version (NDPV) - it counts number of defects which were repaired in class during development of previous release.

Our plugin for KNIME which will be made within ic-depress framework should collect basic informations from every commit in git's log such as:

- name of class,
- extracted issue markers set,
- author of the change
- type of action (modified, deleted, etc)
- complete message from commit
- original path
- change date
- unique identifier

Project infrastructure

- github repository - each of the project participants has his own github account with a fork of ic-depress repository
- Knime Software Development Kit - it is Eclipse IDE with Knime plugin and JDK

Tools

- Eclipse for software development & debugging
- git - a decentralized version control system
- jgit - java class which is impementation of git in Java: <http://eclipse.org/jgit/>
- junit - unit testing tool for Java

Project development concept

1. First plugin version ("offline") will parse a string of text containing log information generated by external git program.
2. Second plugin version will not depend on external tools and will be probably based on jgit Java package (pure Java implementation of git).

```
893f71a008c813fe0daaa50fba184fed66e0230b
2013-03-04 22:31:08 +0100
Sławek Kapłoński
#12 Added methods for load, save and validating config values
893f71a008c813fe0daaa50fba184fed66e0230b
:100644 100644 04b0dbc45ff97b6107e799c482d26f5e3ef718b2
ebe5fda72fb37114706527fc40ac917592989ef9 M
ic-depress-scm-git/src/org/impressivocode/depress/scm/git/GitNodeDialog.java
:100644 100644 47fe5a10ca11d335ebaeecec50b22411c447ae3d
72b547c425ea32d3e6f87739ed9591f264290b53 M
ic-depress-scm-git/src/org/impressivocode/depress/scm/git/GitNodeFactory.java
:100644 100644 84099f19c9f1e10dc7861d96b68c68897dc3c64a
f3becca99d8328ddf6c40858bf9adde277c16aa6 M
ic-depress-scm-git/src/org/impressivocode/depress/scm/git/GitNodeModel.java

907348573f0b3f74a644e962128cc2c855610239
2013-03-04 22:29:44 +0100
Sławek Kapłoński
#12 Added class GitTableFactory with Input tables creating method. Methods which will
generate output table is not created yet
907348573f0b3f74a644e962128cc2c855610239
:000000 100644 0000000000000000000000000000000000000000000000000000000000000000
439cab3b49b75e5f09689c7ba2eba21feaa2f008 A
ic-depress-scm-git/src/org/impressivocode/depress/scm/git/GitTableFactory.java

b1f41ed6f2a284779eb6ccd1ad849207a4b67c42
2013-03-04 22:27:04 +0100
Sławek Kapłoński
#12 Fixed some informations about plugin in xml file
b1f41ed6f2a284779eb6ccd1ad849207a4b67c42
:100644 100644 858e6bda3fbfa25d5d4c3ad5cb16f68c5048ba03
ffa8addcc98154f8c84012394c50476e84c9f3bc M ic-depress-scm-git/META-INF/MANIFEST.MF
:100644 100644 44416b9dcc954ca0601491b1ccd9646911d28c94
```

```
859cca4dafa6e4a52e58ce30f833649793c0c54b M  
ic-depress-scm-git/src/org/impressivocode/depress/scm/git/GitNodeFactory.xml
```

A single text block (usually enclosed between two empty lines) equals to a single commit and has the following format:

1st line: sha-1 of the commit

2nd line: date of the commit (other formats are considered)

3rd line: author of the commit

Following lines all constitute the message of the commit, which is limited by a new line repeating the sha-1 of the commit. The sha-1 is not known before the commit is made, so it is reasonable to assume it will not appear within the message.

Following is a raw diff of the files touched by the commit, one line per file. Each line contains the following fields separated by space character:

- file permissions in octal form before commit
- file permissions in octal form after commit
- sha-1 of file content before commit
- sha-1 of file content after commit
- type of file operation:
 - A - added
 - C - copied
 - D - deleted
 - M - modified
 - R - renamed
 - T - file type changed (i.e. regular file to symlink)

From this data we are able to extract all necessary metrics. Through a straightforward process we are able to map them to fields in the output structure.

One problem which was early noticed is that from git we get data in commits, while the plugin is required to output one structure per file. For this reason we decided to construct the structure UID using the following format:

```
<commit-id>-<hash_of_file_path>
```

where hash_of_file_path is an ASCII hash of the original file path calculated using a hashing function (MD5, sha-1, etc. - to be decided later).

Online mode

For online mode we are planning on using jgit, which is a pure Java implementation of git. If we will be able to get the same format of git log using it, then it should be straightforward to add it in place of the user input text. Otherwise we would have to create a different data adapter.

Project workflow

A typical work cycle embraced by our team involves the following steps:

1. Pull changes from ic-depress github dev branch
2. Pull changes from other developer
3. Create new local git feature branch
4. Work on new feature, test, debug
5. Commit
6. Repeat last 2 steps until feature is finished
7. Merge to local dev branch
8. Push dev branch to own github

On project milestones a pull request is made to ic-depress repository.

Literature

- Marian Jureczko, Lech Madeyski, "A review of process metrics in defect prediction studies", *Metody Informatyki Stosowanej*, Volume 30, Issue 5, 2011, Pages 133-145, 2011
- Marco D'Ambros, Michele Lanza, Romain Robbes "An Extensive Comparison of Bug Prediction Approaches",