Detection of Software Clones

To the 2nd International Workshop on Detection of Software Clones in Amsterdam, The Netherlands
To the 1st International Workshop on Detection of Software Clones in Montreal, Canada

This page is meant to be a general repository and information centre for Detection of Software Clones. As nearly all web pages, it is under permanent construction.

Tool comparison experiment

Time schedule

28.01.2002 - 08.02.2002	First test experiments in order to find out technical problems and to test practicability of the experiment. Please note, the test ends on Friday, 8th February 2002 and only entries submitted till 20:00 GMT will be considered. Please send your submissions to bellon@informatik.uni-stuttgart.de.
18.02.2002 - 25.02.2002	Second test experiments in order to get our procedures straight. Please note, the test ends on Monday, 25th February 2002 and only entries submitted till 09:00 GMT will be considered. Please send your submissions to bellon@informatik.uni-stuttgart.de.
13.03.2002 - 08.04.2002	Main experiment, the results of which are used for the evaluation. Please note, the test ends on Monday, 8th April 2002 and only entries submitted till 09:00 GMT will be considered. Please send your submissions to bellon@informatik.uni-stuttgart.de.
02.10.2002	First International Workshop on Detection of Software Clones in Montreal, Canada. For more detailed information, please have a look at this page.

File format for submissions

We've now agreed on two possible file formats that can be used to submit your clones:

Format 1

BNF for format 1:

```
lineend   := '\n'
separator := '\t'
digit     := '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
digit0    := '0' | digit
number    := digit [digit0*]
filename1 := valid-filename
fromline1 := number
toline1   := number
filename2 := valid-filename
fromline2 := number
toline2   := number
clonetype := '0' | '1' | '2' | '3'
line      := filename1 separator fromline1 separator toline1
```

```
              filename2 separator fromline2 separator toline2
              separator clonetype lineend
file       := line [line*]
```

Example of valid submission file:

```
foo.c   12      56      bar.c   68      90      1
wom.c   34      50      bat.c   90      124     2
wom.c   69      100     bar.c   45      70      1
wom.c   69      100     foo.c   59      80      1
bar.c   45      70      foo.c   59      80      1
```

I have now created an Emacs mode for visualizing clone pairs in this "format 1" (or Clone Pair Format) mode. You can download it from below.

Format 2

BNF for format 2:

```
lineend   := '\n'
separator := '\t'
marker    := '§'
digit     := '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
digit0    := '0' | digit
number    := digit [digit0*]
filename  := valid-filename
fromline  := number
toline    := number
clonetype := '0' | '1' | '2' | '3'
line      := filename separator fromline separator toline lineend
clonelist := marker separator clonetype newline line line*
file      := clonelist [clonelist*] marker newline
```

Example of valid submission file:

```
§       1
foo.c   12      56
bar.c   68      90
§       2
wom.c   34      50
bat.c   90      124
§       1
wom.c   69      100
bar.c   45      70
foo.c   59      80
§
```

The two formats should be largely self-explanatory from the above. A few points:

- `fromline`/`toline` are both **inclusive**. I.e. `fromline=n` and `toline=m` represents `m-n+1` lines of code.
- `valid-filename` should be the path to the file, starting from the system/project directory (e.g. `spule/src/server/ClientConnection.java` is a valid-filename of the project `spule` in our test experiments).
- As we're only interested in clones in the main system, I'll drop every clone where one code snippet contains a filename without the `/src/` part in it, as this marks the main system.
- Please make sure that no clones are reported that are smaller than 6 lines of un-processed source code in size. I'll skip those when importing your data.

- The clones found in each system in our experiment should go into an individual single file. Please don't submit all clones for all systems in one file and please don't submit several files for one system (as we don't do cross-system clone detection in our experiment, it's no problem to separate clones of different systems into different files).

I've been asked to put a definition of the clone types here as well:

- **Type 1** is an exact copy. No modifications (except for whitespace and comments) are allowed.
- **Type 2** is a parametrized copy. Variable names and function calls have been renamed and/or types have been changed.
- **Type 3** is a copy with further modification. Statements have been added or removed.

Resources for the main experiment

We have decided to take 8 projects in the main experiments. 4 written in the C Programming Language and 4 written in the Java Programming Language. For every language, one project is under 30 KLOC, one project is between 30 and 100 KLOC, one project is between 100 and 200 KLOC and the last project is over 200 KLOC.

Concerning the C Programming Language projects, I have switched from using GNU/Linux/i386 as configuration to SunOS/spark. All sources for the C Programming Language have been tested with the Sun C Compiler's `-Xc` switch, which enforces strict ANSI C compliance.
Every C project has a file `check.sh` included in the archive which I used to test it. You can see the defines (e.g. `HAVE_CONFIG_H`) and include paths you have to use from those files. In addition you should define the following defines as the Sun C Compiler automatically defines them as well:

```
__sun
__unix
__SUNPRO_C = 400
__sparc
__BUILTIN_VA_ARG_INCR
__SVR4
__STDC__
```

I am not sure whether I can legally supply the system header files. In each archive is a file telling you the header files you need. If you don't have access to a SunOS 5.5.1 system and you need the headers, drop me an email.

For the Java projects, I provide jar archives of classes our chosen projects depend on. All projects depend on rt.jar. The other jars are listed below for each project.

`weltab.tar.gz`	Code for Weltab (System 1 in the C Programming Language)
`cook.tar.gz`	Code for Cook (System 2 in the C Programming Language)
`snns.tar.gz`	Code for SNNS (System 3 in the C Programming Language)
`postgresql.tar.gz`	Code for Postgresql (System 4 in the C Programming Language)
`netbeans-javadoc.tar.gz`	Code for NetBeans Javadoc (System 1 in the Java Programming Language) Jar files of classes imported by NetBeans Javadoc: `tools.jar` and `netbeans-support.jar`.
`eclipse-ant.tar.gz`	Code for Eclipse Ant (System 2 in the Java Programming Language)

	Jar files of classes imported by Eclipse Ant: `j2ee.jar`, `jakarta-oro-2.0.5.jar` and `log4j-core.jar`.
`eclipse-jdtcore.tar.gz`	Code for Eclipse JDTCore (System 3 in the Java Programming Language) Jar files of classes imported by Eclipse JDTCore: `ant.jar`, `jakarta-ant-1.4.1-optional.jar`, `resources.jar`, `runtime.jar` and `xerces.jar`.
`j2sdk1.4.0-javax-swing.tar.gz`	Code for Java 2 SDK 1.4.0 Swing components (System 4 in the Java Programming Language)

Results

I've put an ISO image of 160 MB in total containing all my original data here. Included are the sources to the original version of the evaluation program, my administrative scripts, the submissions of the participants, the email correspondence during the experiment etc.

Please note that submission data as well as the source code for the evaluation program have already been superceded. So, I advise you to download the ISO and then the additional updates from below if you need them.

For English speaking audience, there's an intermediate results report available in the archive `results.tar.gz`.

Here's the SQL data with all candidates and references:

`data_orig_2percent_clean.sql.gz`	Original data after submission deadline. No mappings present.
`data_orig_2percent_mapped.sql.gz`	Original data after submission deadline. With mappings from candidates to references.
`data_new_2percent_clean.sql.gz`	Data includes Jens Krinke's voluntary submission which was received after the experiment's deadline. No mappings present.
`data_new_2percent_mapped.sql.gz`	Data includes Jens Krinke's voluntary submission which was received after the experiment's deadline. With mappings from candidates to references.

Source code

The source code of the clone detection contest evaluation program (now renamed to `cloneval` because of a naming clash in our Bauhaus project) is available for separate download from here. It is currently version 1.0.2 and replaces the version supplied in the above ISO image.

Emacs mode for CPF files

Here you can download our Emacs mode for visualizing clones in CPF format. Installation and usage instructions are provided within the file itself.

Documents

This section consists of published papers of the individual competitors (in the order of receiption by us).

Ira Baxter	CloneDR_ICSM98_IEEE_Copyright.pdf
Toshihiro Kamiya	kamiya.ps
Jens Krinke	ast01.pdf
Matthias Rieger	Duploc_ICSM99_IEEECopyright.pdf
Stefan Bellon	Diploma thesis "Vergleich von Techniken zur Erkennung duplizierten Quellcodes"

Changelog

10.02.2006 The results of this study had major influences on the development of new tools, e.g. the Bauhaus Suite which now contains several different clone detection techniques and is available from Axivion.

27.10.2003 Uploaded separate data archives for original data and Jens Krinke's voluntary submission. Provide mapped and unmapped data.

15.09.2003 Updated Emacs mode cpf-mode.el (version 0.14) for viewing CPF files. Added horizontal/vertical splitting, highlight-all now opens the correct file and navigation next/prev displays the clonepair instantaneously.

08.09.2003 Updated Emacs mode cpf-mode.el (version 0.09) for viewing CPF files. Added sorting in CPF file and displaying all clones of one file.

05.09.2003 Updated Emacs mode cpf-mode.el (version 0.05) for viewing CPF files. More robust basepath setting and retaining original mode of source files.

21.08.2003 Added Emacs mode cpf-mode.el (version 0.02) for viewing CPF files.

06.05.2003 Added my diploma thesis (in German though) to the documents section.

22.04.2003 Updated `cloneval` to version 1.0.2.

10.04.2003 Added further directories with README and database scripts to source archive.

09.04.2003 Added separate archive with source code of `cloneval` version 1.0.1.

16.01.2003 Updated `results.tar.gz` archive to contain Jens Krinke's new data and added `data_2percent.sql.gz` which contains his new data (the ISO is still the old one, so to get the new data, grab the ISO and the `results.tar.gz` and copy the contents from the `results.tar.gz` over the ISO.

07.01.2003 Replaced my private email address with my work address and added changelog (based on my old CVS log information).

10.08.2002 Added results of main experiment.

13.03.2002 Added information for main experiment.

16.02.2002 Added information for second test experiment.

??.01.2002 Initial version.

Stefan Bellon (bellon@informatik.uni-stuttgart.de),
Last modified: Fri Oct 8 16:37:45 CEST 2004