



Free/Libre and Open Source Software Metrics



European Commission



Information Society Technologies

Sponsored through Framework Programme Sixth (Call 5) by

The FLOSSMetrics Consortium consists of: Universidad Rey Juan Carlos, University of Maastrich, Wirtschaftsuniversitaet Wien, Aristotle University of Thessaloniki, Conecta s.r.l., Zea Partners and Philips Medical Systems PMS Nederland B.V.

Document Information

Version: 1.0
Date : Sep 01, 07
 revision: 0

Owning Partner: URJC

Author(s):
 Carlos García Campos
 Gregorio Robles

Reviewer(s):
 Stefan Koch

To:
 CONSORTIUM

Purpose of distribution:
Initial Version: must create a table of content with Partner assignment and timeline

Printed on at

Status:

- Draft
- To be reviewed
- Proposal
- Final/Released

Confidentiality:

- Public - Intended for public use
- Restricted - Intended for FLOSSMETRICS consortium only
- Confidential - Intended for individual partner only

Deliverable ID:

D1.3

Title:

Repository Finder

License for distribution:

This work is licensed under a [Creative Commons Attribution-Share Alike 2.5 License](http://creativecommons.org/licenses/by-sa/2.5/).
 (The license can be found in <http://creativecommons.org/licenses/by-sa/2.5/>)



Repository Finder
Deliverable ID: D1.3

Page : 2 of 14

Version: 1.0
Date: Sep 01, 07


Status : Final
Confid : Public

Deliverable: D1.3

Title: Repository Finder

Executive Summary:

The Repository Finder described in this deliverable is a tool that tries to automatically find the usual repositories of information related to development tools for a given project. The techniques used by the Repository Finder tool are based on well-known and tested heuristics.

	<p style="text-align: center;">Repository Finder</p> <p style="text-align: center;">Deliverable ID: D1.3</p>	Page : 3 of 14
		Version: 1.0 Date: Sep 01, 07
		Status : Final Confid : Public

CHANGE LOG

Ver.	Date	Author	Description
0.1	10/07/2007	Carlos García Campos	Initial proposal
0.2	27/07/2007	Gregorio Robles	Review
1.0	27/08/2007	Stefan Koch	Review

APPLICABLE DOCUMENT LIST

Ref.	Title, author, source, date, status	Deliverable Identification

Contents

1	Introduction	5
1.1	Motivation	5
1.2	Objectives	5
2	Implementation Details	7
3	Installation and User Guide	11
3.1	Availability	11
3.2	Installation	11
3.3	User Guide	12

Chapter 1

Introduction

1.1 Motivation

Libre software projects usually offer large amounts of information publicly available on the Internet. But these data sources have to be identified first in order to be retrieved, and later on analyzed. As the FLOSSMETRICS project has to deal with thousands of libre software projects, an automatized technique has to be applied to discover these data sources. This deliverable discusses this system.

There are, at least, four data sources that are relevant for the analysis in the FLOSSMETRICS project:

- source code repositories
- mailing list archives
- bug tracking systems
- source code packages

Such data sources are indeed the input for the tools that will be used during the retrieval system process. Information of where these data sources may be found is in general not available at the beginning of the analysis process, when the only available information is usually the name of the project to be analyzed and the forge or community where such a project is hosted.

1.2 Objectives

The main goal of the Repository Finder is to provide a tool that automatically retrieves all the relevant data sources for a given project. It is important to

notice that such a tool provides URLs pointing to the relevant data, and not the data itself. Thus, for instance, it is an URL pointing to a subversion repository what this tool will provide instead of a checkout (i.e. downloading the sources to the local machine) of the repository. The output of this tool will serve as input for the retrieval system.

Repository Finder is, therefore, another piece of the retrieval system, which makes it possible that the whole analysis process is carried out in an automated way.

Chapter 2

Implementation Details

Octopus, the repository finder application, has been written in Python and has been designed to be flexible and extensible. The tool is able to support different kinds of forges, although its first version (*Octopus* 1.0) only supports GForge and GForge-based sites. Other forges can be supported easily and will be included in the future. In addition, *Octopus* is able to produce several kinds of output and, again, adding support for new output types is straightforward.

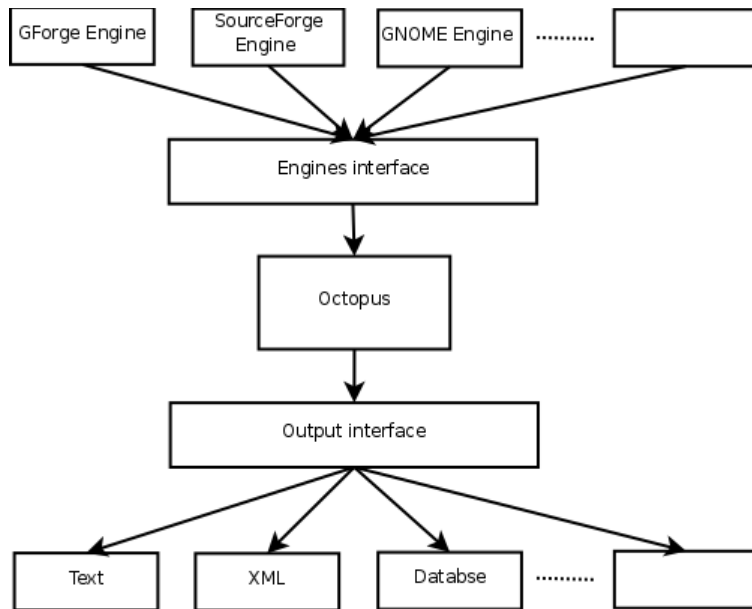


Figure 2.1: Octopus Architecture

The design architecture of *Octopus* is described in Figure 2.1. There is

an engine interface that abstracts *Octopus* from the different engine types. In the same way, an output interface allows *Octopus* to work transparently with several output types. The engines do not have to know any internals of the output types, as they just provide their information to *Octopus* through the engine interface. *Octopus* then uses the output interface to output such information.

Currently *Octopus* supports two engines: Gforge engine, which is able to analyze forges based on Gforge software, and SourceForge engine, which is able to analyze SourceForge; as well as two output types: text, which provides text plain output, and XML. Future versions will include new engines for projects that are not actually based on forges like GNOME or KDE. A new output device to be able to store the output in a database is also planned to be included in following releases of *Octopus*.

The Engine Interface

The engine interface allows *Octopus* to communicate with the engines without having to know which concrete engines are available. Every concrete engine is a plug-in, which means new engines can be included just by adding them to the engines directory. The engine interface is quite simple, consisting only of one method that every plug-in will have to implement. New methods might be added in the future if new kinds of analysis are needed. The implementation of this method in every plug-in consists of a simple HTML parsing of the project pages looking for URLs pointing to source code repositories, mailing list archives, bug tracking systems or source code packages.

The only method of the interface has two input arguments: *project* is the name of the project to be analyzed and *func* is a callback to be called when a new URL has been found for such a project. Such an URL could be pointing to a source code repository, a mailing list and so on.

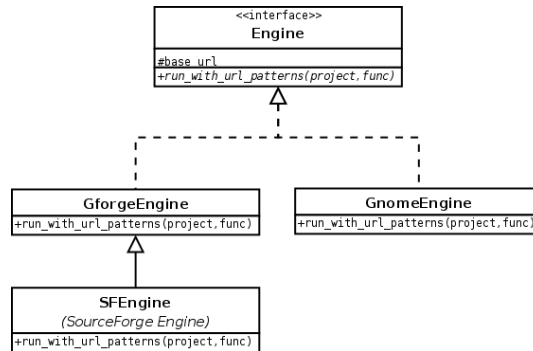


Figure 2.2: The Engine Interface

The OutputDevice Interface

The output interface is used by *Octopus* to provide the information in different output types. In the same way as the engine interface, different output devices are in fact plug-ins which implement the interface methods.

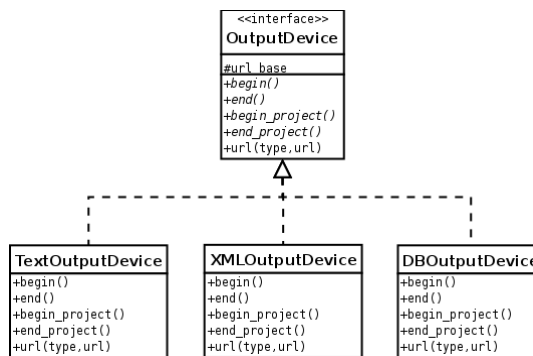


Figure 2.3: The OutputDevice Interface

This interface consists of five methods: *begin* and *end* are called at the beginning and end of the whole analysis respectively, while *begin_project* and

end_project are called for every project analyzed. Finally, the *url* method is called for every URL found during the process.

Chapter 3

Installation and User Guide

3.1 Availability

Octopus is free software distributed under the terms of the GNU GPL version 2 or any later version. It is hosted at the Morfeo Project forge where most of the GSyC/LibreSoft tools (such as CVSA_{na}Y or mlstats, both of which will be used by the Retrieval System) are publicly available.

The current version of *Octopus*, 1.0, can be downloaded from the Morfeo Project forge in the form of a source code package or a Debian package at <https://forge.morfeo-project.org/frs/download.php/234/octopus-1.0.all.deb> and <https://forge.morfeo-project.org/frs/download.php/233/octopus-1.0.tar.bz2> respectively.

3.2 Installation

- **Installing from sources**

The installation process of *Octopus* is the one in use for most of the UNIX applications. First the source package has to be uncompressed¹:

```
$ tar xvjf octopus-0.1.tar.bz2
```

A new directory will be created under the current working directory. An INSTALL file is included in this directory with generic information about how to install it. The next step consists of running the configure script:

```
$ cd octopus-0.1
$ ./configure
```

¹The latest source code package can be downloaded from https://forge.morfeo-project.org/frs/?group_id=33

The configure script can be run with different options in order to customize the installation, see INSTALL file or run `./configure --help` for further information. Once the configure script has finished without errors, *Octopus* can be build just by typing make:

```
$ make
```

At the end of the build process, *Octopus* is ready to be used from the sources directory. Optionally it can be installed in the system, so that it can be used from any directory like any other UNIX command. In order to install *Octopus* an addition final step has to be taken²:

```
# make install
```

At this point, *Octopus* should be ready to be used like any other command available in the system.

- **Installing as a Debian package**

There is also a Debian package available³ for installation in Debian or in any distribution based on Debian (Ubuntu, KUbuntu, GNU/Linux, Knoppix, etc.). Installing *Octopus* in Debian can be carried out in a single step. Once the Debian package has been downloaded it can be installed just by typing, as root or with superuser privileges, the following command:

```
# dpkg -i octopus_0.1_all.deb
```

It will be automatically installed by the Debian installer and thus ready to be used as any other command available in the system.

3.3 User Guide

The execution syntax of *Octopus* is very simple:

```
Usage: octopus [options] URL
```

```
Extract URLs pointing to code repositories, mailing lists, bug tracker  
systems and release packages from the given URL
```

²This last step has to be executed with root privileges in most of the cases, although it depends on the arguments used when running the configure script.

³The latest Debian package version can be downloaded from https://forge.morfeo-project.org/frs/?group_id=33

The following options are available:

- **-h, --help:** Print the usage message
- **-V, --version:** Show version
- **-t, --type:** The type of the project for the given URL. At the moment only GForge and SourceForge projects are supported, so these are the only valid options for this parameter.
- **--projects-file:** Path to a file containing the names of the projects to extract information from. If not provided standard input will be used
- **-o, --output-type:** The output type. Standard output and XML are supported in this moment. There are plans to support database output in following versions.

All of these options are optional except project type, although only GForge is supported by this first version. A list of projects can be analyzed by using a file where every line is a project name. If such a file is not provided the standard input will be used. Next, there are some examples of use with different output types.

- Getting information about the Galeon project hosted in SourceForge:

```
$ echo "galeon" | octopus -q -t sf http://sourceforge.net

Analyzing http://sourceforge.net

Project: galeon
[Code Repository] pserver:anonymous@galeon.cvs.sourceforge.net:/cvsroot/galeon
[Mailing List archive] http://lists.sourceforge.net/mailman/listinfo/galeon-announce
[Mailing List archive] http://lists.sourceforge.net/mailman/listinfo/galeon-devel
[Mailing List archive] http://lists.sourceforge.net/mailman/listinfo/galeon-user
[Release package] http://downloads.sourceforge.net/galeon/galeon-1.2.14.tar.bz2?modtime=1087516800&big_mirror=1
[Release package] http://downloads.sourceforge.net/galeon/galeon-1.2.14.tar.gz?modtime=1087516800&big_mirror=1
[Release package] http://downloads.sourceforge.net/galeon/galeon-1.2.13.tar.gz?modtime=1069545600&big_mirror=1
[Release package] http://downloads.sourceforge.net/galeon/galeon-1.2.12.tar.gz?modtime=1063497600&big_mirror=1
[Release package] http://downloads.sourceforge.net/galeon/galeon-2.0.3.tar.bz2?modtime=1158575516&big_mirror=1
[Release package] http://downloads.sourceforge.net/galeon/galeon-2.0.3.tar.gz?modtime=1158575529&big_mirror=1
[Release package] http://downloads.sourceforge.net/galeon/galeon-2.0.2.tar.bz2?modtime=1158417857&big_mirror=1
[Release package] http://downloads.sourceforge.net/galeon/galeon-2.0.2.tar.gz?modtime=1158417864&big_mirror=1
[Release package] http://downloads.sourceforge.net/galeon/galeon-2.0.1.tar.bz2?modtime=1140957696&big_mirror=1
[Release package] http://downloads.sourceforge.net/galeon/galeon-2.0.1.tar.gz?modtime=1140957706&big_mirror=1
(...)
```

Option `-q` (quiet) is used to avoid error messages that are shown when using *Octopus* with SourceForge due to the fact that SourceForge HTML code is not valid⁴.

⁴<http://validator.w3.org/check?uri=http://sourceforge.net>

- Getting information about a list of projects stored in a file (one line per project) hosted in SourceForge:

```
$ octopus -q -t sf --projects-file /path/to/file http://sourceforge.net
```

- Getting information about LibreSoft-tools hosted in Morfeo Forge (based on GForge) in XML format:

```
$ echo "libresoft-tools" | octopus -t gforge -o xml https://forge.morfeo-project.org
```

```
<?xml version="1.0"?>
<forge url="https://forge.morfeo-project.org">
  <project name="libresoft-tools">
    <repository url="https://svn.forge.morfeo-project.org/svn/libresoft-tools" />
    <mlist url="http://lists.morfeo-project.org/mailman/listinfo/libresoft-tools-commits"/>
    <mlist url="http://lists.morfeo-project.org/mailman/listinfo/libresoft-tools-devel"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=208&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=209&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=210&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=219&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=222&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=227&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=228&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=229&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=238&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=239&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=240&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=253&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=254&group_id=33&func=browse"/>
    <tracker url="https://forge.morfeo-project.org/tracker/?atid=271&group_id=33&func=browse"/>
    <package url="https://forge.morfeo-project.org/frs/download.php/234/octopus_1.0_all.deb"/>
    <package url="https://forge.morfeo-project.org/frs/download.php/233/octopus-1.0.tar.bz2"/>
    <package url="https://forge.morfeo-project.org/frs/download.php/232/octopus_0.1_all.deb"/>
    <package url="https://forge.morfeo-project.org/frs/download.php/230/octopus-0.1.tar.bz2"/>
    <package url="https://forge.morfeo-project.org/frs/download.php/229/wxflawfinder_0.3_all.deb"/>
    <package url="https://forge.morfeo-project.org/frs/download.php/228/wxflawfinder_0.3.tar.gz"/>
    <package url="https://forge.morfeo-project.org/frs/download.php/222/mlstats_0.3.2_all.deb"/>
    <package url="https://forge.morfeo-project.org/frs/download.php/221/mlstats-0.3.2.tar.gz"/>
    (...)
  </project>
</forge>
```