



Software Duplication Detector

Manage code duplication

Description

The Software Duplication Detector (SolidSDD) is a standalone application for detecting and managing source code duplication in software. It can be used to analyze large projects and detect code that has been cloned (e.g., via cut-n-paste operations) during development. Identifying such code fragments is of great importance for:

- **Facilitating development and maintenance;**
- **Reducing the application memory footprint.**

Duplicated code fragments make code updating error prone and time consuming. For example, in most cases, bug fixes are done in one instance of the duplicated code, and consequently, bugs are completely removed only after a number of test iterations. When the detection of such bugs happens after the software is released, each iteration can incur significant costs. Therefore it is very important that once bugs are detected, they are fixed in all the relevant places at the same time. To this end, one possible approach is to refactor duplicated fragments in unique code structures, hence limiting the number of relevant places to one. However, this might not be always possible (or desirable) given particular development constraints. In such a case, modifications have to be performed in all relevant places. In either situation, SolidSDD helps developers to automatically identify the duplicated code fragments, and hence, it accelerates the maintenance activities and reduces the risk of resurrecting bugs.

Reducing the application memory footprint is highly relevant for embedded applications where using smaller memories can lead to important cost and energy savings in the final product.

In addition to identifying the duplicated code fragments SolidSDD offers an intuitive graphical interface for assessing the similarity characteristics and the location of the duplicated code fragments in the code stack. This interface enables developers, architects and software managers to better manage the process of refactoring by assessing the required effort and establishing refactoring priorities.

SolidSDD is able to cope with variations in the duplicated code. Not only exact matches are identified but also cloned fragments that have been modified by renaming variables or by inserting/removing code.

Features

Fast and scalable

SolidSDD can analyze thousands of files in a matter of minutes*. It uses caches for preprocessed information, so the second time it runs even faster. In many scenarios it can be executed on the fly, enabling quick and timely assessment of code duplication (e.g., before committing code to a versioning repository).

*) The recorded end-to-end analysis time of the Azureus code stack (Java application with 3240 files 28MB) on a Intel® Core™2 Duo CPU @2.2 GHz with 4GB RAM was 75 seconds.

Tolerant to variations

SolidSDD does not look only for exactly duplicated code fragments. Most of the times when code is duplicated, developers perform some modifications such as renaming variables or inserting/deleting some code. SolidSDD is able to cope with such modifications, while still recognizing the modified piece of code as a duplicate. Additionally, SolidSDD pinpoints the actual differences in the code, so developers can easily assess the required refactoring effort.

Informative

SolidSDD can be used both by developers, architects and project managers. The detailed code views enable developers to precisely locate the duplicated code fragments; the visual overview enable architects to assess the architectural implications of duplications in order to initiate and prioritize refactoring; the aggregated duplication metrics enable managers to take informed decisions regarding maintenance and to trigger timely actions.

Configurable

The clone detection engine of SolidSDD can be configured for various detection strategies and accuracy levels, to suit specific analysis contexts and needs.

Easy to integrate

SolidSDD can be integrated in the build environment via a command line interface. In this way the clone detection operation can be executed in batch mode, for example together with the nightly builds.

Requirements

SolidSDD supports currently the analysis of software stacks written in the C, C++, C# or Java programming languages. Other languages can be supported on demand.

System requirements:

OS : MS Windows (2000, XP)

RAM : 2GB minimum. 4GB recommended

HDD : 100 MB minimum. The actual required available capacity depends on the size of the project. On average, this is ten times the disk size of the project

Contacts

In North America:

Vizlogix LLC

620 Driftwood Drive
Pittsburgh, PA 15238 USA

E-mail : info@vizlogix.com

HTTP : www.vizlogix.com

Tel. : (412) 828-2098

Mobile : (412) 996-4695

