A Timeline Summarization of Code Changes

Michael J. Decker
Bowling Green State University
Ohio, USA
mdecke@bgsu.edu

Christian D. Newman
Rochester Institute of Technology
New York, USA
cnewman@se.rit.edu

Michael L. Collard
The University of Akron
Ohio, USA
collard@uakron.edu

Drew T. Guarnera
Kent State University
Ohio, USA
dguarner@kent.edu

Jonathan I. Maletic
Kent State University
Ohio, USA
jmaletic@kent.edu

***Abstract*—A syntactic differencing tool (srcDiff) is used to present a summarization of the changes to a class occurring over a time line. An outline of the class is presented with the ability to drill down to individual members (methods and variables). The information is presented so that one can move to the next, or previous, version of the code and examine the changes that occur. The class summary view gives basic information such as the added, removed, or modified members. At the member level, a more detailed summarization of the changes is provided. At all levels, the version number, date, and author are provided.**

Keywords—Documentation, Syntactic differencing, Blame

I. Introduction

Developing and maintaining software is an iterative process of change. That is, we constantly modify (add to and delete from) a code base. In a system modified by a solitary developer, understanding these changes is generally straightforward. However, when scaled to real-world, large-scale systems that involve large teams of individuals located in different geographical locations, understanding what exactly changed in a code base is often problematic and requires substantial effort by developers. It is essential for developers to quickly and easily comprehend changes made to source code. This is critical when understanding the impact of changes, merging changes from multiple developers, conducting reviews of commit requests, and tracking down errors arising from changes.

Lexical differencing (diff) is the de facto standard for presenting changes to source code. It is widely integrated into IDEs and version-control systems (e.g., git, svn). Line-based differencing approaches are very flexible (i.e., can be applied to any text file) and efficient (i.e., scale well to large files), however, they understand nothing about the syntax of the source code being differenced. As such, it is often difficult to understand the output in the context of the language syntax.

An alternative to line-based approaches is syntactic differencing [1][2][3]. Syntactic differencing requires some type of tree comparison on the Abstract Syntax Tree (AST) of the source code. Pure tree/graph differencing is very computationally expensive and as such it is impractical to use on very large code bases as needed by today's developers. Additionally, just examining the changes in the context of the language syntax will not always produce a delta that adequately reflects the actual meaning of the change. Because of this, there are no commonly used tree-differencing tools being used by developers. The ultimate goal of our work is to produce a more understandable and readable delta than what is produced by current line-based and syntactic-differencing approaches. To this end, we developed a novel syntactic differencing approach and implementation named srcDiff [4] based on the srcML [5] infrastructure (www.srcML.org), and a syntactic difference output format (also called srcDiff) [6].

The work presented here uses srcDiff to produce change documentation, namely a timeline summary of the changes to a class. It uses the srcDiff presentation of changes, a summarization of the changes at class and method levels, and navigation of the commit history. This is produced as a webpage and renders in a browser. The change history is presented in full (entire file) or as a summary with only member names being displayed. One can then drill down to a specific method. The history can be navigated forward or backward at the class or member level.

II. Data Sources Used

The approach uses the source code files and version history (from git) of the file. The approach uses srcML and srcDiff.

III. Approach

The srcDiff format [6] extends the srcML format with the addition of four XML tags (`diff:common`, `diff:delete`, `diff:insert`, and `diff:ws`) to contain original and modified source code (i.e., any two versions) marked as the delta or the set of changes to the original source code (base version). Fig. 1 gives an example of the format, as produced by the srcDiff tool, showing both the original and modified source code (simplified) of the function `setImage` from KOffice revisions 1026809-1026810. The changes include the statement in the function `setImage` being wrapped with an if-statement and the data member `m_optionsWidget` being renamed to `m_options`. The srcDiff subsequently has a `diff:insert` tag around the if-statement and a `diff:common` tag around the contents. The text of the renamed identifier (marked in srcML with a name tag) has a `diff:delete` tag around the old text and a `diff:insert` around the new text. In addition, the tags have an attribute type

with the value replace to signify that the code is replaced (i.e., rename). If a modification is to an attribute of a srcML element, these values are versioned with "|", as in the attribute filename on the unit tag. Deleted/inserted whitespace is marked with `diff:ws` for easy processing/analysis. The srcDiff format completely preserves the original and modified srcML and therefore completely preserves both the original and modified source code (e.g., code, whitespace, comments, preprocessor). That is, the srcDiff format is a multi-version, single-document format that allows both the original and modified srcML/source-code to be extracted. As such, srcDiff supports both source-code change analysis, as well as, an efficient means of producing human-readable deltas.

Typically, syntactic-differencing methods support additional edits (e.g., update node value, move, etc). Because srcDiff marks up text directly (e.g., renamed identifier in Figure 1), it does not need a separate edit for an update. In srcDiff, moves are marked as a delete (moved from) and insert (moved to) tags with an attribute move and a unique identifier. Currently, srcDiff supports limited detection for moves within a file as part of the approach. The srcDiff tool is designed to efficiently produce the srcDiff format from any two versions of a source-code document, i.e., two files, directories, or repository versions. In contrast to other syntactic differencers, the code does not need to be syntactically complete, and changes to whitespace and comments are marked up. srcDiff also attempts to produce results on syntactically incorrect code. The tool is very scalable as it handles 1,000 commits/versions (all changed files) in under 5 minutes.

To generate the documentation, git-log is used to obtain a list of every non-merge commit to the specified class/file. Then, srcDiff is run on the original and modified version of the class/file individually for each commit. The output of each is then processed via highly-efficient SAX parsing to produce summary and change documentation as HTML pages. A separate main HTML page loads a summary/change HTML page as needed to show the desired documentation.

IV. Results

The results for the class XSSFWorkbook are available at http://www.sdml.cs.kent.edu/dysdoc3/diffdoc/.

All documentation for all commits of XSSFWorkbook is produced automatically via a command-line tool in about 90 seconds. Initially, the most current commit is shown. There are options to view with changes 1) the entire file (*Full*), just each member's signature (*Signature*), and only signature of changed members (*Changed*). Clicking on the signature of the class or member drills-down showing all code and changes to the code and also displays a textual summary of the changes. In a class textual summary, members can be clicked to drill-down further. The back button is used to exit one level of drill-down. At any point, the buttons previous and next are used to navigate the history, maintaining focus on the current entity displayed. When drilled-down, previous/next will be disabled if there is not a previous or subsequent version.

References

[1] S. Raghavan, R. Rohana, A. Podgurski, and V. Augustine, "Dex: A Semantic-Graph Differencing Tool for Studying Changes in Large Code Bases", 20th IEEE International Conference on Software Maintenance (ICSM'04), 2004, pp. 188–197.

[2] T. Apiwattanapong, A. Orso, and M. Harrold, "Diff: A differencing technique and tool for object-oriented programs", *Autom. Softw. Eng.*, vol. 14, pp. 3–36, Mar. 2007.

[3] B. Fluri, M. Wursch, M. Pinzger, and H. C. Gall, "Change Distilling:Tree Differencing for Fine-Grained Source Code Change Extraction", *IEEE Trans. Softw. Eng.*, vol. 33, pp. 725–743, 2007.

[4] Michael John Decker, "srcDiff: Syntactic Differencing to Support Software Maintenance and Evolution", Dissertation, Kent State University, 2017.

[5] M. L. Collard, M. Decker, and J. I. Maletic, "srcML: An Infrastructure for the Exploration, Analysis, and Manipulation of Source Code", 29th IEEE International Conference on Software Maintenance (ICSM) Tool Demonstration Track, 2013, pp. 1–4.

[6] J. I. Maletic and M. L. Collard, "Supporting Source Code Difference Analysis", IEEE International Conference on Software Maintenance (ICSM), 2004, pp. 210–219.

Original

```
void KisFilterOpSettings::setImage( KisImageSP image ) {
    m_optionsWidget->m_filterOption->setImage( image );
}
```

Modified

```
void KisFilterOpSettings::setImage( KisImageSP image ) {
    if (m_options) {
        m_options->m_filterOption->setImage( image );
    }
}
```

srcDiff

```
<unit xmlns="http://www.srcML.org/srcML/src" xmlns:cpp="http://www.srcML.org/srcML/cpp" \
xmlns:diff="http://www.srcML.org/srcDiff" revision="0.9.5" language="C++" filename="original.cpp|modified.cpp">
<function><type><name>void</name></type> \
<name><name>KisFilterOpSettings</name><operator>::</operator><name>setImage</name></name><parameter_list>( \
<parameter><decl><type><name>KisImageSP</name></type> <name>image</name></decl></parameter> )</parameter_list> <block>{
<diff:insert><diff:ws>    </diff:ws><if>if<diff:ws> </diff:ws><condition>(<expr><name>m_options</name></expr>)</condition> \
<then><diff:ws> </diff:ws><block>{<diff:ws>
    </diff:ws><diff:common>    <expr_stmt><expr><call><name><name><diff:delete type="replace">m_optionsWidget \
</diff:delete><diff:insert type="replace">m_options</diff:insert></name><operator>&gt;</operator><name>m_filterOption \
</name><operator>-&gt;</operator> <name>setImage</name></name><argument_list>( <argument><expr><name>image</name></expr> \
</argument> )</argument_list></call></expr>;</expr_stmt>
</diff:common><diff:ws>    </diff:ws>}</block></then></if><diff:ws>
</diff:ws></diff:insert>}</block></function></unit>
```

Fig. 1. Example source-code change. The top-left contains the original source code and the top-right contains the modified source code. The original and modified code contained within the srcDiff format is given. Deletions are highlighted in a light-red and with a line-though mark. Insertions are highlighted with green. All the original source code is placed in bold.