SyncAll

Keeping Software Projects in Sync

Version 0.01

* + - 1. Introduction

SyncAll is a project that tries to help solving a well-known problem of software development with a new approach. The challenge is to keep the documentation in sync with the code base. Currently there are, as far I know, three approaches to achieve synchronity. Manual synchronisation, model driven development and document generation from code.

The disadvanages that made me look for improvement are the time consuming and boring work when doing manual synchronisation, while both model driven development and document generation have the disadvantage that they do not treat code and documentation as project parts of equal importance. Instead they focus on one and use a „best can“ approach for the other. Another important fact is, that the code or documentation generation process expresses the tool programmers opinion about what is „good code“ or „good documentation“. In contrast, I claim the decision of what is good code and what ist good documentation as a matter of the project development team.

Model driven development introduces a new set of data in a proprietary format that can not be easily browsed or modified with external tools. That makes every limit or bug of the tool a bigger issue.

Document generation from code has the disadvantage that code has to be made before documentation. This is considered to be against the rules of good software development, especially in safety critical environment.

Remark: I have had my experiences with model driven development with IBM/Rational's Rhapsody, and with document generation from code when using Doxygen. Although they are great tools both, the constraints expained above apply to both of them.

SyncAll´s aproach is derived from diff/merge tools. It considers codebase and documentation as two sets of data of their own, which have some data in common while containing other data that are unique to only one of them. SyncAll aims to keep shared data in sync while leaving unique data unchanged. SyncAll operates on top of a version control system. It analyzes the differences both between code base and documentation and also between the current versions of code base and documentation against their respective predecessors. The merge process is, as usual for merge tools, under control of the individual programmer. He may choose between automatic, semi-automatic and manual merge, and he is free to accept or to reject the tools proposals.

Project Principles

First and most important: SyncAll is an open source project. Even more, it is self referencing, using itself as a tutorial on how to use it. SyncAll tries to use open standards whereever possible. It uses the .svg and .docx formats, both specified in free, downloadable documents. Both are based on XML, also an open standard. For diagrams the UML standard is used.

SyncAll respects the claim mentioned above that the software developer shall always have the choice, and shall never be overruled by the tool.

SyncAll works transparent. It does not use hidden information as GUIDs or other meta information. In simple words, SyncAll works the same way as you do when you do your synchronisation work manually.

SyncAll is designed to be „easy join, easy leave“. Bring your Doxygen documented project and go on with SyncAll. If you are not conviced of SyncAlls features, return to other documentation or code generation systems.

SyncAll has a modular design that shall encourage volunteers to write adapters for different text processors, graphic tools, version control systems and programming languages.

SyncAll is a homegrown project and starts from scratch. Commited to the agile principles, we put focus on an early release with some use for the customer right from beginning, with features being added as the projects continues. The roadmap will be described later in this document, as the features will be described.

SyncAll by Example

As a first example we take the UML graphics editor needed for SyncAll. The editor shall be able to display basic shapes, text and UML components. The graphics shall be stored in .svg format.

Supported Enviroment

Currently the following components are supported:

Operating Systems

Ubuntu 14.04

Text Processors

Libre Office 4.2.4.2