Software Forking: creating a variant product by copying and modifying an existing product

Software forking is often considered an ad hoc, low cost alternative to principled product line development.

To maintain such forked products, developers often need to port an existing feature or bug-fix from one product variant to another.

Is forking a sustainable practice? -> look at 18 years of BSD product family history. Our study finds that maintaining forked projects involves significant effort of porting patches from other projects. And the porting rate does not necessarily decrease over time. A significant portion of active developers participate in porting challenges from peer projects. Ported changes are less defect prone than non-ported changes.

Repertoire is the first automated tool for detecting and ported edits with a high degree of accuracy.

Conclusion: Upkeep work of porting changes from peer projects is significant.

They call for new techniques to automate cross-system porting to reduce the maintenance of forked projects.

INTRODUCTION

It is common to fork or borrow features from similar software products.

Example: FreeBSD, OpenBSD, NetBSD (OpenSSH from SSH) (LibreOffice from OpenOffice)

Software forking is often considered an ad hoc, low-cost alternative to principled product line development

Forking has negative implications during software maintenance.

Porting bug fixes and ffeature implciations require development work

\*not all code clones across different projects undergo similar changes due to its evolution

Context: They observed and compared the content and edit operations of program patches for cloned content. More specifically, code changes whose commit messages indicate cross-system porting activities.

Domain: 18 years of version history of the BSD product family.

NetBSD and reeBSD were forked from BSD Lite in 1993 and OpenBSD was forked from NetBSD in 1995.

10.74%-15.52% of the BSD code lines were from ported projects.

Where porting patches occur periodically.

These ported changes are less defect prone (I assume by measuring when these lines of code are changed due to bugs).

26.12%-44.85% of active developers are involved in porting patches from other projects (a large number of them).

50% of patches are ported from 10-20 months, with 90% of all patches being ported within 66-81 months

\*Most porting occurs on less than 20% of the code (sub systems)

\*as much as 10-30% of code in large projects is ported

\*A study of 6,000 software projects found a general lack of uniqueness at 1 in 7 lines

\*Their repertoire program focuses on program patches from forked projects to gauge upkeep keeping these projects & their patches in sync. There is an emphasis on the amount of duplicated work due to multiple projects simultaneously pulling and customizing patches for their projects.

Up to 40% of code is shared between OpenBSD & other BSD projects (so these are a special case of projects that share an especially large amount of code).

The study examined 54, 14, and 30 released from FreeBSD, NetBSD, and OpenbSD (covering 18 years of development and parallel evolution history).

All three repositories use a CVS repo, so they used cvs diff to identify program patches.

40 token similarity within the repo was used as a minimum for identifying common lines of code.

Repertoire uses a bi-gram matching algorithm to find similar edit sequences (allows for slight variation for how editing is done on each project).

?what are recall values? Recall is high with a low token size and decreases with larger tokens. I’m assuming this is finding those matches again with the same program when using a different token size.

They changed their token size in increments of 10 for finding the best token size.

The program did not find contiguous lines of ported code, since this is less than 40 tokens.

14%, 16%, and 11% of the BSD project’s evolution is ported from other projects, where it takes an average of 734-944 days to port edits from other BSD projects.

50% of ported changes propagate to other projects within 3 releases.

GRAPH OF PORTED EDITS SHWOING MONTHS VS % PORTED EDITS

The authors acknowledge that their study on the BSD family may not generalize to other systems. The BSD is a very large, long surviving product family created by project forking who share massive portions of code.