

Experience Report: Refactoring LogicBlox

by Spencer Rugaber, Hwa-You Hsu, R. E. K. Stirewalt and David Zook

This paper discusses the experiences of the authors in the task of refactoring a large system that had drifted from its original architectural design. The authors discuss some background research on refactoring and briefly list some steps that should be taken during an architectural refactoring. They describe the tools they used to perform this large refactoring which include:

- Lattix - a code analysis tool that relies on dependency structure matrices to show dependencies between modules
- Slick - a tool developed by the authors for analyzing C and C++ programs in order to derive information about the program's structure

The refactoring was done on a commercial platform called LogicBlox, a large multi-language system consisting of 1.5 MLOC which has been maintained and added to by multiple groups of developers.

To carry out the refactoring the authors first developed a conceptual architecture model of the system based on discussion with developers. They then used the tools described above to help target dependencies in need of refactoring. After coming up with a new architecture for the modules in question, the authors discussed it with the developers to get their feedback and to help discover any patterns that would be worthy of abstraction.

After their consultation with the developers, they ended up changing their original refactoring plan to better accommodate the developers' concerns. This resulted in a "less clean" version of the new architecture, but it was a necessary detour to make all parties comfortable with the proposed changes. The paper is concluded with a discussion on the lessons learned throughout the refactoring process.