

Automated Transformation of Legacy Systems

This paper argues for an increased focus on automating transformation processes based on the statistic that, on average, a well trained programmer can transform about 160 LOC per day. To transform an entire system at that rate would be too costly. The author breaks down the process of reengineering a legacy system into four steps:

- Assessment - captures the legacy system's As-Is state by extracting properties of the existing system's design, and simultaneously generating detailed documentation of the system
- Transformation - provides transformed software that is compiler-ready and testable at the unit level
- Refactoring - reengineers the new system to improve system architecture and performance
- Web enablement - facilitates migration of the new system to the web environment by transforming the legacy application to Java (this step is optional)

The innovative part of the above process is that the transformation step uses a suite of artificial intelligence technology tools to automate 99 percent of it's work. Most of what is left for manual transformation should just be what truly requires a human decision maker. The author makes the point that while high levels of automation are achievable for transformation tasks there will always be tasks that are manually intensive. Nevertheless, anything that can be automated should be.

Unfortunately, the paper did not go into how the AI performs it's tasks but I assume that is because this comes from a private company dealing with defense software. Also, the techniques described were applied to languages to convert them to C++ - it is unclear whether they would work if trying to perform a transformation to the same language. Regardless, these techniques would probably be overkill for reengineering ISVs.