

Architectural Mining: The Other Side of the MDD

Grady Booch

IBM Thomas J. Watson Research Center
gbooch@us.ibm.com

Abstract. A back-of-the-envelope calculation suggests that - very, very conservatively - the world produces well over 33 billion lines of new or modified code every year. Curiously, the moment that code springs into being and is made manifest in a running system, it become legacy. The relentless accretion of code over months, years, even decades quickly turns every successful new project into a brownfield one. Although software has no mass, it does have weight, weight that can ossify any system by creating inertia to change and deadly creeping complexity. It requires energy to make such a system simple, and to intentionally apply that energy requires that one be able to reason about, understand, and visualize the system as built.

Considerable research and labor has been invested in model-driven development for the purposes of transforming models into running systems, and while these efforts have yielded some useful results they have not led to the revolution that some expected. Similarly, considerable work has been undertaken in static and dynamic analysis and design pattern discovery, and while they too have yielded some useful results, these efforts have been rather scattered. Still, we believe that there is much more that can be done. One of the explicit goals we made in the early years of the UML was that it be a language for reasoning about a system, and so in this presentation, we'll reexamine that early goal. In particular, we'll look at efforts to consider the other side of MDD, the mining of the architecture of an as-built system from its source code and its execution.