Invited Talk

Software Archeology and the Handbook of Software Architecture

Grady Booch
IBM Fellow and Chief Scientist, IBM Rational
gbooch@us.ibm.com

Abstract

Software is invisible to most of the world. Although individuals, organizations, and nations rely on a multitude of software-intensive systems every day, most software lives in the interstitial spaces of society, hidden from view except insofar as it does something tangible or useful. Furthermore, most interesting software is continuously evolving: it is never turned off, but rather is adapted, integrated, extended, and reused. As such, while greenfield software development persists, most software development is brownfield, requiring teams to conduct archeological digs in order to understand, reason about, and transmogrify those systems.

It is a sign of maturity for any given engineering discipline when we can name, study, and apply the patterns relevant to that domain. In civil engineering, one can study the fundamental elements of architecture in works that expose and compare common architectural styles. Similarly, in chemical engineering, mechanical engineering, electrical engineering, and now even genomic engineering, there exist libraries of common patterns that have proven themselves useful in practice. Unfortunately, no such architectural reference yet exists for software-intensive systems. Although the patterns community has pioneered the vocabulary of design patterns through the work of the Hillside Group and the Gang of Four (in their seminal book, Design Patterns), our industry has no parallel to the architecture handbooks found in more mature design disciplines. The primary goal of the Handbook of Software Architecture (http://www.booch.com/architecture) is to fill this void in software engineering by codifying the architecture of a large collection of interesting software-intensive systems, presenting them in a manner that exposes their essential pattern and that permits comparisons across domains and architectural styles.

In this presentation, we will examine the nature of software archeology and some of the systems under study in the Handbook.
Bio

Grady Booch is recognized internationally for his innovative work on software architecture and software engineering. A renowned visionary, he has devoted his life's work to improving the effectiveness of software developers worldwide. Grady served as Chief Scientist of Rational Software Corporation since its founding in 1981 and continues to serve in that capacity within IBM. Grady is one of the original authors of the Unified Modeling Language (UML) and was also one of the original developers of several of Rational's products. Grady has served as architect and architectural mentor for numerous complex software-intensive systems around the world in just about every domain imaginable.

Grady is the author of six best-selling books, including the UML Users Guide and the seminal Object-Oriented Analysis with Applications. He has published several hundred articles on software engineering, including papers published in the early '80s that originated the term and practice of object-oriented design (OOD), plus papers published in the early 2000's that originated the term and practice of collaborative development environments (CDE).

Grady is a member of the Association for Computing Machinery (ACM), the Institute of Electrical and Electronics Engineers (IEEE), the American Association for the Advancement of Science (AAAS), and Computer Professionals for Social Responsibility (CPSR). He is an IBM Fellow, an ACM Fellow, a World Technology Network Fellow, a Software Development Forum Visionary, a recipient of Dr. Dobb's Excellence in Programming award as well as three Jolt Awards, and generally just a really nice and gentle fellow. Grady was a founding board member of the Agile Alliance, the Hillside Group, and the Worldwide Institute of Software Architects, and now also serves on the advisory board of the International Association of Software Architecture. Additionally, Grady serves on the board of Iliff School of Theology. He is also a member of the IEEE Software editorial board. Grady helped establish work at the Computer History Museum for the preservation of classic software.

Grady received his bachelor of science from the United States Air Force Academy in 1977 and his master of science in electrical engineering from the University of California at Santa Barbara in 1979.

Grady lives in Colorado and Maui. His interests include reading, traveling, singing, and playing the harp.