The idea that programs should clearly reflect the design decisions they embody has a long history. Higher-level languages, syntactic macros, domain-specific languages, and intentional programming are different approaches to this common goal.

Recent work from several areas, including aspect-oriented programming, has significantly advanced our ability to make code expressive. At the same time, it forces us to reconsider a number of basic assumptions, including what is a program, what is a module, what is a language, and what is an editor.

Gregor Kiczales is Professor of Computer Science at the University of British Columbia. His work is directed at enabling programmers to write programs that, as much as possible, look like their design. He has pursued this goal in a number of projects. He led the Xerox PARC teams that developed aspect-oriented programming and AspectJ. He is author, with Danny Bobrow and Jim des Rivieres of “The Art of the Metaobject Protocol”. He was one of the designers of the Common Lisp Object System (CLOS), and developed the standard PCL implementation of CLOS.