FOR IMMEDIATE RELEASE - from - *Visual Software International (VSI)*
Spring Lake, NJ, January 25, 2010 < www.VisiSoft.US >

A JERSEY SHORE COMPANY'S BREAKTHROUGH IN SOFTWARE

Since 1995, surveys on *changes in productivity* across multiple U.S. industries put the software industry dead last - with productivity actually declining. Yet, its counterpart in computer chips is the productivity improvement leader. After significant improvements in computer speeds, memory size, and other tools that aid software development, real productivity continues to fall. This is in spite of the fact that project sizes have gotten smaller because of management's concern for risk of failure.

Computer-Aided Design (CAD) tools are considered the major driver in chip productivity improvements. Computer scientists acknowledge the lack of such tools in software, which is still approached as a craft. Like other labor intensive environments, programmer concern for job security has caused resistance to innovation.

But the approach to building software appears to be headed for huge change. Since 1982, a small engineering firm on the Jersey Shore, Visual Software International (VSI), has been working on a totally new paradigm. It is driven by the need to develop large scale simulations of communication and control systems, simulations that must run very fast - on parallel processors under a single operating system. This led to a breakthrough - now called the "separation principle" - that separates data from instructions. Conceived by company founder Bill Cave, this approach allows one to track software module independence, and automatically allocate processors to processes at run-time on a large parallel processor.

More importantly, the separation principle provides the basis for engineering drawings of software, with a one-to-one mapping from the drawings to the code, a true form of *software architecture*. As in other fields, architecture is much more graphical than algebraic or textual. Whether designing machines, ships, or buildings, architects produce drawings. These drawings are not flow charts, nor "approximate" or suggestive, but precise engineering specifications used directly in production. Until now, drawings of software were abstractions to aid in design, but not of much use in the actual production or support mode. Just as important are the separate languages that define the data and instructions. They are read easily by non-programmers, a requirement for subject area experts validating complex models.

VSI has evolved this CAD environment over hundreds of successful software and simulation projects. Called *VisiSoft*®, it is a fully integrated CAD product for developing software. Prior to *VisiSoft*, software architecture did not exist, an observation that is immediately apparent upon seeing it. After using *VisiSoft*, one quickly draws the analogy between current programming approaches, and architects in other fields trying to produce designs without drawings. Recent articles about Microsoft's problems with its new Vista operating system bear this out. Quoting Cusumano[1], "We now know that the chaotic 'spaghetti' architecture of Windows ... was one of the major reasons for this gridlock. Making even small changes in one part of the product led to unpredictable and destabilizing consequences in other parts since most of the components were tied together in complex and unpredictable ways."

Programmers pride themselves in their ability to understand esoteric languages, and to create and maintain software directly in these languages. Now, with real architectural drawings and very readable languages, *VisiSoft* significantly cuts the time to produce software, and allows others to understand and upgrade that software. Independence of software modules makes reuse and sharing common. Modules can be copied and modified easily by people other than the original author. And, most important, productivity is up by whole number multiples, especially in the support phase of a complex software product's life-cycle. This dramatic breakthrough is verified easily by direct observation.

[1] Cusumano, Michael A., "What Road Ahead ...?," Communications of the ACM, July 2006, pg 21-23.