



## Generative Programming Saves Development Effort

Richard J. Cichelli and Bob Harwick

Programmers write a lot of code they probably don't need to.

### Automating Program Development

Generative and component approaches have the potential to revolutionize software development in a similar way as automation and components revolutionized manufacturing. Generative Programming (developing programs that synthesize other programs), Component Engineering (raising the level of modularization and analysis in application design), and Domain-Specific Languages (elevating program specifications to compact domain-specific notations that are easier to write and maintain) are key technologies for automating program development.

*Fourth International Conference on  
Generative Programming and Component  
Engineering (GPCE'05)  
Sep 29 - Oct 1, 2005, Tallinn, Estonia*

We at SCS were able to replace a major module of our newspaper advertising system with a more functional solution for which the programmer wrote only 1/25 as many lines of code (LOC). The prior solution consisted of 112,000 LOC in C. The new version required the writing of only 4,200 LOC. How can other programming tasks be made over 25 times more efficient? Let me tell you the tale of this development effort.

Layout-8000™ is a major subsystem of our advertising system. It is a tool for automatically and semi-automatically figuring out where to place ads in an edition, a process called dummied. About 50% of U.S. daily newspaper

copies are dummied with Layout-8000.

I was responsible for Layout-80® development at the American Newspaper Publishers Association Research Institute (ANPA/RI). (The ANPA was the trade association for daily newspapers. It is now called the NAA.) When I left the ANPA in 1983, I licensed Layout-80, Layout-8000's predecessor, and made it a product of SCS.

From its inception, developing Layout-80 offered interesting opportunities to do computer science research. Dummied automatically requires GOFAI (Good Old Fashioned Artificial Intelligence). I wrote chess programs in Pascal in those days, so this seemed a natural extension. Widely delivering Layout-80 challenged the limited programming language support of the computers of the day.

For the ANPA board to support software distribution from its computer applications department (i.e., my department), the technology needed to be able to run on the computers newspapers owned. They had DEC PDP 11's, VAXes, IBM mainframes, HP3000s and a variety of other computers. Several operating systems were available on each of these.

Gannett sites ran DEC computers and Knight Ridder sites had Hewlett Packard systems. Neither company wanted the association to confer an advantage on the other.

We felt that best way to achieve portability was to use Fortran. Layout-80 was developed in Fortran 66. I also wanted to produce not just platform-independent code but quality software as well. The SFortran preprocessor from Volvo Flymotor allowed us to write structured programs that were translated into portable Fortran. Later, under SCS's ownership, the Pfort verifier was used to assure the portability of the generated Fortran.

### Software Consulting Services, LLC

630 Selvaggio Drive, Suite 420

Nazareth, PA 18064

Sales: 1-800-568-8006

Fax: 610-746-7900

E-mail: sales@newspapersystems.com

www.newspapersystems.com

Last update: February 19, 2009

**SCS BUILDS TRUSTED NEWSPAPER SYSTEMS.**



### John Backus

"Much of my work has come from being lazy. I didn't like writing programs, and so, when I was working on the IBM 701 (an early computer), writing programs for computing missile trajectories, I started work on a programming system to make it easier to write programs."

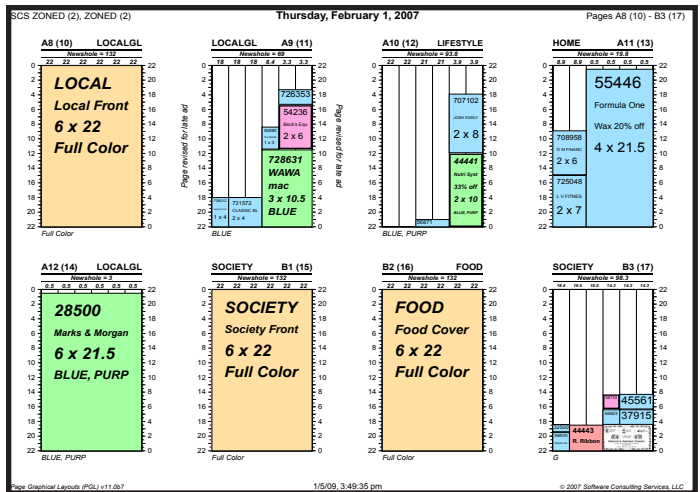
*John Backus (1924-2007)  
Principal creator of Fortran.*

[Fortran was the first highly successful high level programming language.]

While the community was skeptical that this new method could possibly out-perform hand-coding, it reduced the amount of programming statements necessary to operate a machine by a factor of 20, and quickly gained acceptance.

*From Wikipedia*

only was there a significant reduction in development time versus prior versions, NewPGL features enhanced functionality and improved presentation. It has, among its many advances, nicer typography and ad image inclusion.



### The Spice Formula Language.

The Formula Language is an algorithmic, application-oriented language whose internal variables can be declared with simple, structured, and abstract data types. The language allows application developers to focus on the business logic, user interface, and database design of the application instead of on the details of hardware and operating system platforms, input-output handling, and other low-level programming issues.

Unlike many programming languages that are compiled into machine code for a given hardware/OS platform, the formula language is compiled into a special byte code that is executed by a program known as the spice virtual machine. As long as the virtual machine runs on a given platform, any application written in the formula language runs on that platform.

Integrated with the formula language and built into the virtual machine are numerous subsystems.

As a side note, throughout the development of Layout-80 the two programmers, John McGrath and Kevin Weiner each wrote an average of 76 lines of deliverable SFortran code per day from design concept to first day of use in production.

There have been 11 major versions and hundreds of minor releases of Layout-8000 deployed over the last 29 years. The code base grew from 28,000 LOC when I licensed it to over 375,000. In all this code only a single routine of about 400 LOC was system dependent.

SCS ported Layout-8000 to new platforms like MS-DOS, Windows, Unix and Linux. In 2000, the SFortran code was retired and replaced with C. Starting in 2006, the C code is being systematically replaced with code in our domain-specific language (DSL) and tool set, collectively called **Spice**.

The rewritten Layout-8000 module mentioned above is called PGL, for Page Graphic Layout. NewPGL™ was developed with the latest generation of SCS development tools. Not



So how does Spice reduce programming effort and errors? We could call "reducing the amount of programming statements necessary to operate a machine by a factor of 20" as passing the Backus Test (BT). A technology which passes the BT would have a clear and measurable advantage over prior methods of writing software.

Very few computer scientists have claimed to have invented development tools that pass the BT, but some have. One is Paul Bassett. What Bassett advocates is using his Frame Technology to automate software development significantly. He is a provocative evangelist for Bassett Frames.

What's the relationship between the Spice DSL and Bassett Frames? The Spice formula language was engineered to support creating interactive database publishing applications, our bread and butter. Bassett Frames are a tool for writing generators for programs.

Newspapers can reduce costs and increase customer satisfaction by offering self-service portals for advertisers and subscribers. We needed to support server-centric, web-based applications. To succeed in programming for the web we needed the ability to easily and automatically compose the programs browsers executed.

Emitting HTML, JavaScript, CSS, PHP, etc. is simply making programs that write programs. Simple perhaps, but often really messy.

SCS's formula language isn't a small DSL, but an entire programming environment. It is a full third/fourth generation language with an embedded relational database processor, ODBC interface, forms handling system, reporting system, dialog manager, WYSIWYG text editor with language tools for spelling and political correctness checking, dictionary and thesaurus, composition engine, context sensitive help system, page dummy display routine, run-time expression evaluator, XML parser, etc. and the namespace glue to bind them all together. Couple these with a source-level run-time debugger and an execution-time profiler and you can begin to appreciate just how powerful Spice is.

**Paul Bassett writes on software engineering's chronic concerns and how to cure them:**

Since our industry's inception over 50 years ago, software systems have been notoriously late, over budget, and of mediocre quality. Already low, programmer productivity has actually worsened since 'object oriented' programming techniques went mainstream [According to project auditor Michael Mah of QSM Associates in a private communication to me.]

In what looks like a desperation move, organizations have been in a race to replace their expensive local software talent with cheaper offshore labour. Yes, this tactic can save money in the short run, but offshoring does nothing for our industry's chronic inability to deliver quality results in a timely fashion. Indeed, it could well make matters worse.

Clearly, something is wrong with this picture. **We don't import food from countries that till their fields with water buffaloes.** Why? Because modern agribusinesses have made the transition from being inefficient craft industries to producing cornucopias of low cost, high quality food. The software industry can and should do likewise. Why? Because, as we shall see, a key enabler - so-called frame technology (FT) with its associated processes and infrastructure - has been fulfilling this dream in diverse commercial settings for more than twenty years! What is wrong with this picture is our penchant for rejecting ideas that challenge deeply held core values, even when those ideas can cure chronic concerns...

*<http://www.stephenibaraki.com/cips/v46/bassett.html>*

Everything that we know of that is needed to make writing publishing applications easy is included in Spice.



We added macro processing, frame processing and string tag substitution to all the DSL environments to make development easier, especially development for the web.

With these technology changes in Spice, extraordinary advances were possible.

Frames really work as Bassett promises. With their introduction into Spice, programs developed for the web quickly grew in functionality and yet often got smaller in terms of lines of code. Automated frame processing replaced hand coding. Libraries of adaptable frame templates replaced manual cut-and-paste coding. The more the frame library grew, the faster new functionality could be implemented!

Best of all, Spice Patterns, a template or string tag substitution technology, allowed easy integration of much available Free Open Source Software (FOSS).

NewPGL makes particularly good use of FOSS. Its basic development technique consists of assembling components defined as Spice Patterns. To make an output page, one places page dummies on a sheet with so many across and down. Each page dummy has a name, section location, a set of zero or more ads, and so forth. To do the work, our programmer defined a pattern for an ad, then one for a page, and then the output sheet. Given a descriptor of the pages, ads and their locations from Layout-8000's dummifying routines, NewPGL takes the patterns and builds a program for Lout, a FOSS batch composition engine, to compose into output pages.

The advantages of Lout are that it is essentially a high-level, block-structured language that translates Lout documents into the PostScript graphics language.

This advantage can be a disadvantage as well. Writing a document for Lout is somewhat like writing a program. Spice Patterns allows SCS developers to use tools like Lout to create terse programs that provide quite complex behavior at a level of abstraction that is understandable.

### What is Lout?

Lout is a document formatting system designed and implemented by Jeffrey Kingston at the Basser Department of Computer Science, University of Sydney, Australia.

The system reads a high-level description of a document similar in style to LaTeX and produces a PostScript file which can be printed on most laser printers and graphic display devices. ...

Lout is inherently multilingual. Adding new languages is easy. The following languages are currently supported (in alphabetical order): Czech, Danish, Dutch, English, Finnish, French, German, Hungarian, Italian, Norwegian, Polish, Portuguese, Russian, Slovenian, Spanish, Swedish.

*From <http://lout.wiki.sourceforge.net/>*

### Conclusion

So now Spice is not just a DSL for database publishing applications but a powerful tool for generating programs and documents as well. We are achieving a much higher level of programmer productivity and system quality. Our Spice tools provide SCS with a unique and innovative development advantage.