



QUR Q-zine

[Index](#)
[Editorial](#)
[Mailing lists](#)
[Mission](#)
[About](#)
[Dr. Unix](#)

[Going Dutch](#)
[Le Coin](#)
[Français](#)

[Planned](#)
[Proposed](#)

[SANE 2004!](#)
[NLUUG](#)
[WURLUG](#)

[Portaloo](#)

Comal has been set free

Jos Visser



23-09-2002

License

This article is Copyright (c) 2002 by Jos Visser. You are granted an unlimited license to copy and publish this article in any form you see fit, provided that you do so in its (the article's) entirety, make no changes and do not remove the copyright or this license.

Link

The home page of the OpenComal project is <http://www.josvisser.nl/opencomal>.

Intro

In a well lit room in a non-descriptive suburb somewhere in the western world, a crowd of twenty to thirty people gathers. They're a mixed bunch, some teens, some twenties, but most are in their mid-to-late thirties or even older (nearing their first mid-life crisis). The moods are also mixed. Whereas most some seem to be content, even happy, to be here, others are reserved and act withdrawn. The presence of a small camera team from the local cable television network seems to have little influence on the crowd.

At the clock of eight the cameras are switched on. Their lights permeate every little corner of the hall. A man in a dark blue suit walks to the stage and asks for attention. A silence ripples through the crowd. He talks in a warm and trust-inspiring voice, thanks everyone for their attendance and says he like to extend a warm welcome (on behalf of all present) to a new member. He points to a man in his mid-thirties who is sitting near the side of the hall. The man stands up. Suddenly the light in the hall dims and a spotlight is switched on. Someone hands him a wireless microphone. His bright blue eyes wander through the room but see nothing (remember the spotlight?). Small drops of sweat appear on his face. He moves his hand through his short blond hair. Then, with a flick of his thumb that betrays experience, he switches on the microphone and says: "Hi, I'm Jos Visser, and I'm a Comalcoholic...."

The real intro (although somewhat long winded)

Sometimes I can feel very lonely. And never more so than when I talk about Comal and people go: "Comal? What is that?". It seems strange that even though in the 1980's I felt part of a strong Comal movement I almost seldomly (let's call it "never" for practical purposes) run into someone who knows what Comal is. And in case you wonder, my standard answer today is: "The language in which I learnt how to program decently".

Way back when I was young I spent my high school days at the "S.G. De Krimpenerwaard" (which today is called "Krimpenerwaard College"). Being an outsider these school days were not particularly happy, but yet I feel very fortunate to have been taught by a set of unique teachers who tried to inspire love for their scholarly fields in their pupils (much of which effort was wasted on my peers, something I could not for the life of me understand....).

Even though for most of my early teens I had no clue what they were about I had always been

interested in computers. Then, in around 1982 the micro computer phenomenon took off and I subsequently bought a Sinclair ZX-80, which was quickly replaced by a ZX-81, which was somewhat less quickly replaced with a BBC Microcomputer. For those too young to remember, these micro computers were fairly basic, in fact so basic that they were programmed in Basic. The micros had no real operating system but booted directly in the Basic interpreter.

So I taught myself programming in Basic, dooming self in the process (according to the late professor E. Dijkstra). And then, when the limits of Basic were being reached, I went on to master assembler (Z80 and 6502). This was before hard disks, the Internet, modems and bulletin boards. The micro computer movement was kept alive through magazines and local user groups.

And suddenly some real micro computers appeared in school: RC700 computers running the CP/M "operating system". One was appointed to the school administration but two (or maybe three, memory starts fading at my age :-), were to be used to support (math and IT) teaching. Leading these early IT-efforts at our school were the math teachers Dick Klingens (also sub-principal and in charge of creating the school's teaching schedule) and Jan Meerhof. There are probably no two people who influenced my programming career as much as these great teachers.

Dick Klingens introduced the Comal language to our school. He was abhorred by the bad programming practices that were promoted and kept alive by the very limited Basic implementations that were in wide spread use. He subsequently went on a crusade to promote **Structured Programming**. The GOTO statement was officially excommunicated and lucid programming became the norm: clear algorithms, meaningful variable names and procedures and functions that could be called by name and with parameters became compulsory. A group of enthusiastic pupils followed his lead and somewhat of a Comal movement was formed. A Comal newsletter was published and Dick Klingens was in touch with fellow Comal users in Germany, Denmark, England, Ireland and the US of A.

For me, structured programming proved to be a breakthrough in my perception and understanding of programming. I started applying these same techniques in Basic and assembler (at home), and wrote many a structured program for nothing more than the sheer fun of writing beautiful code. Comal spread, and implementations appeared for the Commodore 64 (the famous Comal cartridge) and even for my beloved BBC Microcomputer (the Comal ROM by Acornsoft). I went on to study computer science and picked up Pascal, COBOL, C and 68k assembler, graduating in 1988 on programming language and compiler design. In the meanwhile, the Intel personal computer with MsDos had become popular, and UniComal provided an excellent Comal implementation for that platform.

The meaning of Comal

For non-Comalites, now might be a nice time to digress a bit on the nature of this programming language.

Comal started out as a modification of Basic by Danish teacher Borge Christensen. A small team led by him hacked away at the code of a Basic interpreter, adding features that would allow structured programming. Thus the COMmon Algorithmic Language was born (there are several nice and more detailed articles about this phase of Comal available on the web).

Comal was intended as an alternative to Basic, especially with regard to learning people how to program in a structured fashion. It provides a simple and forgiving user interface that is easy to master but still allows the development of quite complex programs. If you want to define its capabilities in one sentence "A crossover between Basic and Pascal, with the best features of both and none of the drawbacks of either" would do quite nicely (in fact, I like this definition I just came up with so much that I immediately pasted it on OpenComal's home page :-).

A look at a short Comal program immediately betrays its origins but also gives you a clue about its structured programming features:

```
10 INPUT "A number please : " : n#
```

```

20 PRINT n#;
30 IF isprime#(n#) THEN
40   PRINT " is a prime number!"
50 ELSE
60   PRINT " is not a prime number..."
70 ENDIF
80
90 FUNC isprime#(n#) CLOSED
100  FOR f#:=2 TO n# DIV 2 DO
110   IF n# MOD f#=0 THEN RETURN 0
150  ENDFOR f#
160  RETURN 1
170 ENDFUNC isprime#

```

This simple program tries to assert whether a given number is a prime number (not using the best algorithm, I am sure). Running it gives the following dialog:

```

$ run
A number please : 8
8 is not a prime number...
$ run
A number please : 23
23 is a prime number!

```

Comal supports you (amongst other things) with the following features:

- Line numbers are used for editing only. There are no statements that refer to a line number.
- Program listing formatting and indentation is done automatically by the Comal interpreter.
- Cosmetic keyword (like THEN and DO) need not be input.
- Only syntactically correct lines can be input in the program.
- A programs does not run unless its structure is correct (all control structures (IF, REPEAT, WHILE, PROC, FUNC, TRAP etc.) closed and no reference to non-existing procedures or functions).

And apart from that it provides all features of a modern programming language with respect to parameter passing, local variables, arrays and so forth. Comal does not support pointers.

I like Comal

It might be needless to say at this point, but I really like Comal. Even though at this point I "speak" more than ten programming languages more or less fluently (amongst which Comal, Perl, shell script, Pascal, C, C++, Java, various assemblers, Forth, Basic and Javascript) I am still of the conviction that Comal is a very good language to start programming with or to write a program in. You would not write an operating system in it, nor a compiler, a GUI library, a device driver or a database management system, but there is a large class of problems that can be comfortably attacked with Comal.

And with respect to teaching programming to beginners: I have taught countless courses on various programming languages (C, C++, Java, Javascript, assembler, shell script, Perl) and it is crystal clear to me that programming starts and ends with having a mindset that is capable of dividing a problem in a series of steps that work toward its resolution (i.e., an algorithm). I probably have a difference of opinion on this with object orientation gurus who claim that all programming starts with converting the problem to a model of interacting objects, but in my experience this step is impossible for people who do not have any idea about what an algorithm is and how computers can be programmed to implement such an algorithm. There is no doubt in my mind that my exposure to Comal helped me pick up on Pascal and C with unnerving speed.

OpenComal (part I)

Then in 1992 my career was put in hibernation because I had quit my job to move into international IT contracting. Unfortunately, at that almost exact same time, the whole IT market slumped and I

found myself sitting at home without a contract (a sort of unplanned sabbatical). It was a very relaxing time: I slept a lot, read books as if there was no tomorrow, obtained my second scuba diving star and watched the 1992 Summer Olympic Games on TV.

However, it seems almost impossible for me to stay away from the keyboard. Additionally, my brain has this habit of going into high creative gear when I have nothing else to think about (i.e. during sabbaticals and long vacations). So, in order to keep myself sane (all this creative IQ needs to go somewhere) I decided to create a free and portable Comal interpreter.

This is exactly the pointless sort of hobby project that keeps hackers like me going. I guess it is almost impossible for a normal person to fathom the therapeutical value of carefree hacking away for no other reason than to create code. My development platform at that time was an IBM PC XT with Turbo C. I took the Acornsoft Comal manuals and started to implement a Comal interpreter with all the features of Acornsoft Comal plus various extensions that I thought I remembered from Comal-80 or that I thought were useful (and easy to implement :-).

In order to make OpenComal highly portable the entire thing was programmed in K&R C with all platform dependent routines (e.g. for memory allocation, keyboard input, screen management et cetera) externalised in a separate module with well-defined entry points). I hacked away and hacked away, and pretty soon I had written a very well working portable Comal interpreter with a bunch of test programs.

And then (after sitting at home for almost four months and with money running out) I started my own consultancy company in the Netherlands and quickly found a bunch of contracts teaching Unix courses at Hewlett-Packard. OpenComal went into the fridge for a while.

OpenComal (part II)

In the next couple of years I worked on OpenComal quite irregularly. The interpreter was basically finished although a number of small bugs persisted in crashing it every once in a while. I was getting involved in Unix quite heavily, and to test the portability of the interpreter I ported OpenComal to Linux, FreeBSD, HP-UX and SunOS, which proved to be a quite simple exercise. However the real work in porting OpenComal to a new OS is to write a good line editing routine (for Comal program input and handling the INPUT statement). This is not particularly difficult, but not particularly interesting either and I never really took the time to finish these ports properly (using "gets()" for line input, which leads to a sad and user unfriendly interface).

I did undertake some other small efforts in these years like switching to ANSI C (converting to using C function prototypes immediately solved some of the bugs) and some other small maintenance.

OpenComal (part III)

And then the year 2002 came round and I realised that due to some (let's call them "interesting") events in my personal life I was in more dire need of a break than any white man in history. So on August 1st I moved to Denia (Spain) and started a (this time planned) sabbatical.

Early in 2002 I had been invited to host the end-of-conference quiz at the Linux Kongress in Cologne early September (I do quizzes like this regularly at conferences and I'm also an amateur standup comedian and improvisation actor). As quiz master (speaker) one is treated to the full VIP treatment, which includes access to the entire conference. Also present was my friend Dr. Moshe Bar who was giving a presentation on OpenMosix (really mind boggling stuff by the way) and I suddenly remembered that I had this quite interesting body of code lying around and that now might be nice time to release it to the general public.

And so two days later I fired up my laptop and started hacking away at OpenComal once more. I fixed some small bugs, cleaned up the code, gave some serious effort to the Linux system dependent routines (introducing ncurses to do the screen/keyboard management), patched the Makefile, wrote a

README, formatted the sources (GNU indent rulez!), stuck the GNU General Public License to it, renamed the interpreter to OpenComal (it was called something else previously, but that is not important right now :-)) and put it up on the web (on josvisser.nl and Freshmeat).

I then decided to drop a line to Dick Klingens and tell him what I was up to. He forwarded the message to Doug Snell (of Macharsoft, the UniComal distributor), who contacted me and a small snowball got rolling. Oh, and by the way, resulting from my ereshmeat posting, I received a small source code patch within hours of release that contained the modifications necessary to compile OpenComal on BSD/OS 4.3 (an operating system of the Unix family).

OpenComal (part IV)

And so now it's out there: the OpenComal free interpreter; free as in "gratis" (i.e. costs nothing), but also free as in "freedom": use it, abuse it, modify it, port it, sell it, redistribute it, whatever (but, in order to inflate my already not small ego even more, let me know!).

My restless mind is already contemplating various extensions and activities: packages (the UniComal USE statement), native code integration (e.g. modules written in C), porting it to Win32 (should be really easy but I don't like to use Windows), compiling it for MsDos (I already got the code for that, but no MsDos system :-)) and in my most careless moments I am even thinking about porting it to my Lego Mindstorms "brick" (so as to let OpenComal programs control the robots you can build with MindStorms).

As with any free software project, your input is invaluable! So why not go to the [OpenComal web site](http://www.josvisser.nl/opencomal) (<http://www.josvisser.nl/opencomal>) and get involved!

by *Jos Visser* <josv@osp.nl>
Wed Sep 11 08:46:15 CEST 2002

Other articles by Jos Visser:

- [Kernel based random number generation in HP-UX 11.00](#)
- [Unix failover clusters - If you've seen one, you've seen 'm all](#)
- [Interessante ontwikkelingen in Linux](#)
- [Rock Linux: Not for wozies!](#)

E-mail: Jos Visser

[[Index](#)]



For more information write e-zine@e-zine.nluug.nl

