

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

The Obsolescence of Programming - GENASYS/3000.

by: Ian Farquharson

President - Info-Boutique Ltd.

Tuesday D-6 - 01

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

Part 1. The Obsolescence of Programming.

*In order to commence this discussion, we first address two questions
viz:*

i) IS IT POSSIBLE? that we can live without programming.

ii) WHAT IS PROGRAMMING?

*i) Firstly, we must say that if one does not think it is possible for
programming to become obsolete AND one is not prepared to listen to
persuasion thenHALT.*

Otherwise , let us continue.....

*ii) After some research into defining "programming", I have come to
the conclusion that the world outside of data processing is pitifully
ignorant still of basics. For example, in one noted dictionary the
following definition is given.*

PROGRAMMING: DATA FOR A COMPUTER.

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

As a result, I have attempted to define programming, in a way in which we can relate.

Is it informing the computer a) WHAT you want it to do.

b) HOW to do what you want.

c) Both of the above.

I believe it is c). ie. PROGRAMMING IS INFORMING THE COMPUTER BOTH WHAT YOU WANT IT TO DO AND HOW TO DO IT, IN A WAY IN WHICH IT CAN UNDERSTAND.

This is acheived through the use of various 'languages' which, as we know, ultimately translates down varying levels to the binary notation of the computer (COBOL, FORTRAN, RPG etc).

Within computer programs we find both the WHAT and HOW of our definition above.

The WHAT invariably becomes actions such as PUT, GET, DISPLAY, PRINT, UPDATE etc. The HOW is the logic that pulls together these actions and other programs to acheive the desired results. IF-THEN-ELSE-(GO TO?) etc.

Most of us concerned with this discussion will at one time or another,

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

have inherited programs which resemble 'spaghetti with meatballs'; programs where the actions (meatballs) are interconnected by logic so messy (spaghetti) that understanding it is 90% of the work and modifying it only 10%.

I would also like you to give thought to the following, related, questions:

- a) where is most programming time spent?*
- b) where are the worst "bugs" found?*
- c) which "bugs" create the worst problems?*

answer any of the following.....

Answer 1). Incorrect or invalid actions.

Answer 2). Syntax errors.

Answer 3). Logic flaws.

My answer is 3).

With both invalid actions and syntax errors, the results are usually obvious and fast to resolve especially with an on-line system. In fact with syntax errors, the solution is often just to re-compile.

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

*With flaws in the programs logic, the symptom is often intermittent.
With spaghetti programs (& many are!) the time involved is
significant.*

*Also true of logic flaws, is that the larger the program, the more
difficult it is to fix, and the fact that the logic involved is that
of another person and each person thinks in a different way!*

*We must conclude from the above points that the current status is not
very acceptable and we should look for alternatives.*

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

Alternatives.

- 1) back to the quill pen.
- 2) there is no solution and we must live with it.
- 3) structured programming.
- 4) program generators.
- 5) application generators.
- 6) robotics.

For myself, I believe that ultimately we will see the day when we can talk to robots and tell them what we want. However, I do not believe that the technology is yet ready although it is progressing rapidly. I therefore select 5) - application generators, as my solution and goal.

Before discussing my reasoning, however, let me first comment on the other alternatives:

1) & 2). I expect no supporters of these alternatives.

3). Structured programming. If one must program computers, then let them be structured. Unfortunately, too many of us are of an 'old school' where we have left behind us years of programming to become

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

Senior Analysts, DP Managers or Consultants and as a result have carried forward too many bad habits. We think we understand structured programming but many of us would have problems knowing where to start. Fortunately, the Universities and Colleges are now producing people who do not know any other way to program other than structured.

Structured programming definitely REDUCES logic problems but does not eliminate them. It is still susceptible to costly conversions with new machines and languages.

4). Program generators. examples are SL1, GENASYS Inc. NB. this product is not to be confused with Info-Boutiques product called GENASYS/3000 which is an application generator.

These products allow us to tell the computer WHAT we want with out detailing the logic. The computer reacts by producing COBOL programs for you which are nicely structured and commented.

Magic you say; wonderful! OR IS IT.

The advantages are obvious, of course, but let us analyse the disadvantages:-

a) COST..... typically \$100,000 to \$200,000!

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

b) COBOL..... is usually the result. You may hate COBOL.

COBOL standards are becoming less and less
compatible with previous versions, and it is
too inflexible for fine actions like string handling.

c) EFFICIENCY COBOL is not necessarily the most efficient language.

d) DATABASE.. there are no database standards in COBOL.

e) PEOPLE.... you still need people trained to program.

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

Application Generators.

Examples of this are our own GENASYS/3000 (Generator of Application Systems) and APL.

APL is mentionned because it is an attempt to develop systems by entering specifications and avoiding detailed logic, although most people think of it as a programming language. Its biggest disadvantage is that it uses special symbols which bear no resemblance to English.

The CONCEPT is: Tell the computer WHAT you want, NOT HOW to do it.

In other words, avoid time-wasting with logic.

In as simple a form as possible, we should be able to enter specifications into the computer relating to any new application that we wish to build (eg. with an on-line EDITOR). END OF STORY.

The computer should then be able to interpret these specifications and do what we require without any further action on our part.

It is possible and if we look at the following broad spectrum of

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

programming areas we can assess each individually:

```

*****
* COMPUTER      *
* DATABASE      ***** Documentation
*****
*              *
*              *
*              *
*              *
Terminal      Batch
oriented      reports
  %/
  %/
Speech
oriented

```

The computer can do all of the above without having to write programs.

(except for speech, which is in the near future)

I believe in letting the computer do the work.

It is the last necessary step before we enter the age of robotics. If we can not just tell the computer WHAT we want, how can we tell a robot?

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

PART 2. GENASYS/3000

GENASYS/3000 is an application generator which was designed and produced by Info-Boutique Ltd.

Of the three main program categories listed earlier, it currently satisfies the terminal oriented functions (and would be easily adaptable to speech) and the documentation. A report-writer is planned soon, however it will interact easily with QUERY, QUIZ etc. It is a database oriented product (ie IMAGE/3000) although much can also be accomplished with KSAM/30000.

At this point it would be useful to reiterate point 5) from PART 1.

Tell the computer WHAT you want. ie enter your specifications with an on-line EDITOR.

Thats exactly what we do. Referred to as a SPECIFICATIONS file (SPECS file) we use the standard HP EDITOR and enter our system specifications in a relatively free-format but using key-words.

NB. We can have any number of SPECS files. eg. 1 containing all

EXPERTS EN ORDINATEURS · COMPUTER SPECIALISTS

applications or many containing different applications.

The concept is really the same as QUERY but the idea is carried much farther. eg. we can create an XEQ file to produce a report with just a few lines. The traditional approach would have been to write a program which would have been a few pages of code. We would have depended on each programmer to use correct logic for such repetitive functions as numbering pages, printing the date, totalling etc.

GENASYS

QUERY

.....

.....

.....

.....

..... SPECS

..... XEQ

..... file

..... file

.....

.....

.....

.....

:FILE MENU=specsfilename

:RUN QUERY.PUB.SYS

:RUN GENASYS.GS.INFOSYS

>XEQ filename

.....

.....

*order entry,**outstanding orders report.**accounts receivable,**file maintenance,*

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

*accounts payable,
inventory control,
etc, etc.*

NB there are no programs to be written in either of the above.

I am sure that most people will agree that QUERY is extremely limited but for some strange reason HP have never improved it. I believe that the concept is right and the proof is the wealth of comparable products on the marketplace. eg ASK, WIZARD, QUIZ, REX, etc. All of these products use a similar concept to QUERY but succeed in all areas where QUERY does not. (mainly multiple data-sets.)

There are some points that must be clarified now with respect to GENASYS/3000:

1) It should NOT be confused with V/3000.

V/3000 maintains forms. You still have to write programs.

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

2) It should NOT be confused with data-entry programs.

GENASYS/3000 can enter data but it has two way communication with your database. It does most of the functions that your programs would have done.

3) You do NOT need either block mode or even HP CRT's.

4) Yes there must be some limitations but very few.

See the following list of features.

Main features:

- 1) Consistent & compatible with IMAGE, KSAM etc.
- 2) Terminal independant.
- 3) Multilingual.
- 4) Powerful in-built help feature.
- 5) Ability to branch out to standard subroutines.
- 6) Ability to run other programs (interface with packages).
- 7) User control over batch STREAMS.
- 8) Multiple data-sets per program function.
- 9) Multiple screen forms per program function.

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

- 10)multiple screen forms per data-set.
- 11)Verification option (double-entry) for keypunch.
- 12)Self-test, self-demo options.
- 13)STACKED input.
- 14)MENU control.
- 15)MENU-level security.
- 16)SCREEN level security.
- 17)FIELD level security.
- 18)Subroutine library included.

After assessing the above features the obvious question that will be raised is WHAT ABOUT EFFICIENCY?

The fact that the HP3000 automatically makes programs re-entrant. means that GENASYS/3000 automatically becomes re-entrant. eg. if all your applications were done with GENASYS only, and let us say for example you had 50 terminals running 30 different program functions, there would still be only 1 copy of 1 program in memory - GENASYS!

Bearing this in mind, it is important that GENASYS itself is not cumbersome. FACTS... if GENASYS is completely locked in memory the total requirements are 35k bytes plus user data-segments. Alternativley, if it is not to be locked in memory then segments of

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

*function code are swapped in as required. The largest segment is 4k
bytes!*

GENASYS/3000 is written in SPL.

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

Documentation

Documentation warrants a special section in its own right.

If all your system specifications are in the computer and field statistics are in the IMAGE root file AND the MODUS OPERANDI is consistent, THEN it is feasible that the computer can write your complete system documentation for you!

We have achieved this with an optional GENASYS product called the DOCUMENTOR.

You don't have to write a word. It will present you with sizeable well written, consistent documentation in a matter of minutes which is....

- 1) Word processed.*
- 2) Table of Contents.*
- 3) Standard Operating procedures.*
- 4) All MENU's used.*
- 5) All SCREEN layouts used.*
- 6) All FIELD specifications.*
- 7) All EDITING rules used.*
- 8) Explanation on operating each function.*

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

9) *IMAGE schema and SPEC's file.*

10) *Complete cross-index of words.*

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

The Bottom Line

Where does this discussion lead us finally?

I think that perhaps a look at one companies experience with an application generator such as GENASYS is worthy of consideration.

According to the M.I.S. Manager of B.A.S.F. (Canada), Hank Van Leuwen, the following was true:

B.A.S.F. installed an HP3000 series III in 1979. The mandate given to DP was an overwhelming amount of new on-line systems and requests which were growing faster than results could be obtained.

The initial action taken was to hire people and contract much of the work to outside individuals/companies.

The standard approach was used ie. COBOL with V/3000 and either KSAM or IMAGE., until the discovery of the GENASYS prototype. After purchase of GENASYS and a report-writer from Info-Boutique, B.A.S.F. progressed at an outstanding rate and they concluded the following after a detailed analysis:-

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

- * 2,500% improvement in development over COBOL & V/3000 (ie 25 times faster)*
- * greater m/c efficiency (purchase of extra memory was delayed after seeing GENASYS performance).*
- * immediate cost savings by not contracting out.*
- * high moral of users, getting action fast from DP.*
- * high moral of programmers, can now concentrate on interesting problems and progress to systems design.*
- * less cost in terminals by switching to HP2621 in many areas.*
- * documentation & systems standardized and always up to date.*
- * luxury features available to users at no cost eg HELP (?)*
- * their projects, which were estimated to take 14 man-years with COBOL & V/3000, took just 4 man-years with applications generation.*

EXPERTS EN ORDINATEURS - COMPUTER SPECIALISTS

CONCLUSION.

The obsolescence of computer programming is not a dream about the future. Neither is it a complex system of COBOL program generation, only available for hundreds of thousands of dollars on IBM 370's.

It is a reality, here and now, on your own HP3000 computer.

It's biggest enemy is the closed mind.