

IAPetus



Quarterly Bulletin For the Institution of Analysts & Programmers

Issue 8 Dec 1993

Council Elections

Nominations for election in 1994

The constitution of the IAP provides for a governing Council of fifteen elected members. Each member serves for three years, five members retiring and five new members being elected every year. The five existing Council Members will serve until 31st May 1996.

Our task now is to select five

more people to join them next year. This is the opportunity to nominate people who you, the members, would like to see running your Institution.

All nominees must be either Members, Fellows or Companions of the Institution, but nominations can be made by anyone, including

Associate Members. Each member can nominate up to five people because there are five vacancies to be filled. You cannot nominate yourself, but anyone who would like to stand for the Council and does not know another member to nominate him may find the Institution office can help.

Nominations should take the form of letters addressed to the Secretary of the Institution, stating clearly who is making the nomination, and the names of the persons being nominated.

Sufficient information should be provided to ensure that nominees cannot be confused with other similarly named members of the

Continued on Page 2

The Institution

Someone said that it would be a good idea to remind you what the Institution can and does do, the services offered and so on. Here goes:-

The Directory – members near you, people with the right talents.

The Office – who may not be able to answer your question but know a man who does.

CIX conference – if you are on CIX look out for it, or email arobertson@cix.compulink.co.uk for details.

A Pocket Diary – hopefully in the same envelope as this lot.

Redundancy Counselling Service – we hope you don't need it, but speak to Mike Ryan if you do.

Professional Indemnity Insurance – offered by Tim Luckitt

Energy Management advice – offered by Derek Worthy (0604 767220)

Academic Dress – you too can wear one. Ask Ede and Ravenscroft about the IAP hoods and gowns.

And the thing we'd like to do...

Branches – it's time that we started to gather. Programming can be very solitary, and an opportunity to meet is often welcome. Even just to drink beer and talk about anything under the sun.

Who would like to start one? Or meets with a group of computer people already and would like to turn it into an IAP Branch?

Here in Crewe every other Monday we meet for what I call "4 Cs" – the Crewe Computer and Curry Club. After sitting in the Imperial pub for a while we go to the Indian down the road. Conversation ranges from computers to zoo animals to the theory of relativity to swopping tall tales of our military exploits....and we often have a few machines along as well.

Call Alex or Megan to ask for the next date, get directions or even if you'd like to do the same and want some ideas.

Inside this issue

Institution News	2
Editorial	3
Conference 1993	4-6
The Beginner	7
Never let the expert write your manual	8

The Director General writes...

Once again I find myself reviewing a year that has past – my third full year with the IAP – and peering forward into the uncertain mists of 1994. What will happen to the Institution and to all of us in the year to come, I ask myself. Old Moore – do you have any off-the-shelf software?

1993 has not been a good year for the IT industry. While there is undoubtedly more optimism about, so far this does not seem to have produced more jobs. Work is picking up, but not yet to the point where firms need to take on more staff or contractors to get the job done.

make a New Year's resolution:
"Sometime during 1994 I will recruit a friend to join the IAP"

Some people have taken the opportunity of a quiet patch to return to college and improve their qualifications. Others have abandoned hope of getting a job in IT, and have left the industry for ever.

All this means is that when the real upturn does come, the shortage of competent professional programmers will be more acute than ever before. Keep your powder dry!

The Institution has continued to grow during this difficult period. We lose less than 10% of our membership each year, implying that the average members stays with the IAP for more than 10 years! But we have been recruiting new members much faster than that. In 1992 we recruited 628, and even this year

more than 500 new people will join us.

Increased membership will do more than anything else to boost the benefits of Institution membership. It will increase public recognition and respect, and enable us further to extend our services to members without increasing subscriptions. It only needs each one of you to make a New Year's resolution: "Sometime during 1994 I will recruit a friend to join the IAP".

A positive trend during 1993 is that I have felt I am getting to know more members. Increasingly my time is spent dealing with individuals, talking to them on the 'phone or meeting them in person. Some have called at the office, where we are always very pleased to receive members, although it is best to ring first. One lady arrived from overseas at 10.30 am on a Sunday morning and insisted on being photographed shaking hands with the DG outside the door of the Institution. She was immaculately dressed; I was in a

dressng gown and unshaven!

One of the reasons I am able to devote more time to individual members is that I have been able to dump a lot of boring routine work onto our faithful administrator Nicole Edwards. Nicole is French, and has no patience with the sloppy British way of doing things, as some members, and more usually their banks, have found out! As a result, we have sharpened up our act.

I have also been assisted to a very considerable extent this year by Alex Robertson, who took a large part of the organisation of the Conference off my shoulders. The support I received from these two people was outstanding, but there were many others, and I thank you all.

My best wishes for a restful Christmas, and a busy and truly prosperous 1994.

Michael C. Ryan
 Director General

The Institution of Analysts and Programmers

Income and Expenditure Account for the year ended 31st May 1993

Income		1992
Subscriptions	£98,266	£62,019
Other Income	<u>1,334</u>	<u>449</u>
Total Income	£99,600	£62,468
Expenses		
Services to Members and Promotional Activities	£42,553	£32,030
Salaries and Staff Costs	36,301	29,468
Office Overhead Expenses	<u>14,825</u>	<u>9,457</u>
Total Expenses	<u>£93,679</u>	<u>£70,955</u>
Surplus/(Deficit) of Income over Expenditure	£5,921	(£8,487)
	=====	=====

Accountants Report to the Members of the Institution of Analysts and Programmers

We have prepared accounts for the Institution of Analysts and Programmers for the year ended 31st May 1993 from the books and records maintained by the Institution.

The summary Income and Expenditure account shown above having been extracted from the full accounts.*

Charles Stuart
 Chartered Accountants
 12th November 1993

* Full accounts are available on request.

Council Elections – Continued from Page 1

Institution. Nominations received by the Secretary after 31st January 1994 will be ignored.

If the total number of people nominated is more than five, it will be necessary to hold an election. Manifestos and voting papers will be circulated with the March issue of IAPetus. Members will be invited to vote for five candidates in order of preference. The result will be determined by a transferable vote system. Results will be announced in IAPetus in June.

Right then. Christmas is coming, the Editor's getting fat....and here's the latest IAPetus.

On page three, by special request, is the information about our last Council Member, Anne Gray. She was left out last time for the simple reason that I couldn't get hold of her in time. She tells me her secret ambition has been to be a Page Three Girl, so here it is realised. I suppose now you are all going to bombard me with your secret desires, go and write to Jim'll Fix It instead.

But do write to me if you have something to say about computing, or life as a programmer or anything

else that would look good in IAPetus. What about a collection of "urban myths". You know, the stories like the customer who is told to copy a disk so sticks it in the Xerox machine.

Or the school computing teacher who made a display. There was a 5.25" disk labelled "A floppy disk" and a 3.5" disk labelled "A hard disk". (That one is true, and boy was her face red when I told her!) Let's hear yours.

In the next issue, we hope to have details of those people who hope to become Council Members in next year's elections. I won't call them manifestos, because unlike politi-

cians I expect you all to keep any promises you are rash enough to make.

A couple of years ago I muttered in the wrong place that a journal was needed, so here I sit. Which of you has thought that you could run the Institution... convince the rest of us and we'll let you have a go.

So, stick a bit of tinsel on your VDU and enjoy your Christmas. If religion isn't your thing, enjoy a rest and the festivities, if you do believe, then blessings.

Megan C. Robertson

Subscriptions for 1994

From 1st January 1994 the following rates will apply:-

Companions	£62.00
Fellows	£55.00
Members	£48.00
Associate Members	£41.00
Students	£34.00

All these rates are reduced by £5.00 for members who elect to pay by direct debit. For members with UK current accounts this is the cheapest and most convenient method of payment.

Once the direct debiting authority is in place, payment of the right amount at the right time is automatic. Members are urged to use this method of payment where possible. Forms may be obtained from the office.

There is no longer a discount for payment by standing order; this method of payment is being phased out.

Members who pay by cheque will receive letters of reminder when their subscriptions are due. To save administration costs the Institution does not normally issue receipts, but will do so in special circumstances if requested.

Overseas members in particular may find it convenient to pay over the telephone by credit card (ACCESS or VISA). This avoids the cost and inconvenience of obtaining sterling drafts.

Last but by no means least...

Some of you commented that of our first five Council members, nothing was written by/about Anne Gray. It's quite simple, I couldn't get hold of her before IAPetus was due to go to press! So here is her bit, to complete the set.

Anne Gray FIAP MBCS

I started working as an analyst programmer in the distant past (late 60s), initially on ICL 1900s, but quickly moving to minis and later micros.

In the early days each machine had its own code, memory was non-volatile, and systems had to be designed and written for 2-3 K of memory. Then came assembler, more memory, and more sophisticated peripherals like disk drives. I have grown with the industry through 3rd generation languages and the arrival of the early PCs to today's hardware and software.

Initially I was employed by a software house in London where I spent time trouble-shooting for a computer manufacturer as well as designing, writing and installing commercial applications, running

on minis. After several years I branched out on my own, moved to Yorkshire, then ran my own company along with my (then) husband for ten years. We supplied hardware and software, training and support, having our own basic ledgers and payroll packages, and developing bespoke software, for a wide range of industries.

I have always been interested in the problems of delivering what the user regards as a reliable and error-free system. In recent years I gained exposure to BS5750 and TickIT. If you don't know what that is, see me later.

My current endeavour is as a Quality Consultant, gearing companies up for BS5750 registration, and I specialise in systems/software houses and TickIT.

Lottery Tickets

In the last edition of IAPetus it was reported that a number of people had been annoyed to receive unsolicited lottery tickets, seemingly posted from the address of one of our members in Malta.

I am pleased to be able to report that the Council has now received an apology and an explanation from the member concerned, both of which they regard as satisfactory. For the record our member was not instrumental in the misuse of the Institution's membership list. He was away from home at the time and therefore unaware that another member of the family, who opened mail in his absence, had used our Directory for his lottery ticket business.

Michael C. Ryan
Director General

So, there we were again, some new faces and some I'd seen before, clustered around the tea urn nibbling Jammy Dodgers in the City University. Eventually we were led away from such temptations and sat listening (perhaps still chewing) to Mike Ryan.

Mike began by saying that the turn-out was about the same as last year, with four people from Malta and plenty who'd been before. The job scene was pretty grim, although there's still work around if you can get it. Then he went on to the changes in the Institution's organisation, its transformation into a proper company instead of a private enterprise operation.

This led into the introduction of the first five Council members – Jim Bates (at whose name every computer virus quails), Harold Ead (an electrical engineer), Alex Robertson (who had just been to America to sort a customer out), Nick Swain (who collects information) and Anne Grey (who is NOT a token person of any nature).

Mike then handed over to the aforementioned Jim Bates, who was to chair the morning's proceedings. He told a cautionary tale about the differences between specialists and experts, preferring himself to be a specialist. Something about Time magazine running a survey to compare the accuracy of professional stock market predictions against those made by a chimp which threw darts at the financial pages. Needless to say, the chimp won hands down, so Time's advice to its readers was that if they wanted to make a killing on the stock market they should eat more bananas!

OOP

The first speaker was Neil Anthony-Pillai of Data Access UK – the company which provides Britain with a 4GL called DataFlex (you are either saying DataWhat? or Fabulous at this point). Back in 1989 poor Neil tried to teach me DataFlex, now he's Product Manager and had come to speak on

Object Oriented Programming and Relational Databases. After introducing his company and its products, he looked at industry trends (old friends like rightsizing, integration and object orientating everything).

The change over to OOP is more of a migration than a straight switch, in many cases individual components of a system are created in a conventional manner than put together OOP-style, a sort of two layer effect – the Programmer, as professional code-hacker, sticking to his old ways whilst the "power user" (who knows nothing of the coding and doesn't need to) being able to manipulate the results.

Neil continued by tracing the development of ideas, beginning with object orientated analysis and design (which he says is useful for large projects that you don't expect to finish!), and moving on to OOP developmental environments, warning that an experienced "conventional" programmer is not going to turn into an accomplished OO one in mere days, and thence to OO database management, which has high overheads in space etc, but is very nice for multimedia applications (despite a tendency to use nuclear explosions to crack walnuts!), and finally object based applications.

The next part of the talk covered the emergence of appropriate languages for the OOP approach – both adaptations of existing languages and the development of wholly new ones – and their evolution to the present state of play. He described what makes a language object orientated, the encapsulation, polymorphism and inheritance characteristics which either make perfect sense or baffle you totally!; then described the three types of 4GL database around today.

These range from a purely traditional forms based procedural beastie through "object based" object orientated procedural database management systems to the full-blown object orientated object administered system. He ended by declaring a need for us all to get back to basics, that we live in a dynamic world and need to keep moving with it.

Software reuse

The next speaker was John Dingle of British Telecom, who gave a presentation on SCC, the Software

Component Catalogue developed by BT for internal use as an aid to promoting one of OOP's key aims, the reuse of software. Reusable it might be, but if you don't know that it is there, it is difficult to utilise it. Moreover, the system enables the deeper ramifications of the whole concept, not just the reuse of code but of other parts of the life cycle as well.

The system had its origins in the commercial requirement to optimise code production. This, if successfully implemented, would have the benefits of speed of develop-

Confe

ment, the creation of a consistent environment and ease of maintenance, giving rise to more projects through the door. Naturally this would be a long-term investment, but the gains would include backwards and cross project consistency.

The mechanism by which this was to be achieved is a central depository/library system, in which any item can be searched for by name (if you happen to know it) or by its properties. Still, it must be remembered that just because it's there it is not always best to reuse it – sometimes you do need to write your own.

On the other hand, most people are used to working that way, the real difficulty is to break through the culture barrier – some people don't like using other people's code, whilst others (especially managers!) equate new projects with new code.

Like any system of this kind, it stands and falls on its usability and on the versatility of the searches possible. Some kind of all-pervasive logic must control the indexing, so to provide users with a common way of describing the code in the system. This was achieved by classifying the types of code fragment into one of six basic groups (then subdividing as necessary). These were man-machine interface, transaction processing environment, database facilities, comms facilities, utilities (libraries, tools etc) and applications.

"some people don't like using other people's code, whilst others... equate new projects with new code"

And making it go? The whole thing was built around FrameMaker, an on-line documentation system, and interfaced with the user via a hypertext web, giving said user maximum freedom and versatility.

There are three main ways of finding what you want – tracing down the hierarchies of classification by function, looking for it by name or hunting through a mesh of links set up between bits that have been used together before. I for one can see plenty of uses for this approach in all kinds of data handling... I wonder

if BT will ever publish?

Conference

Design problems

Next was Igor Shagaev, who treated us – in English far better than my Russian! – to an account of the principles behind the development of software designed to assist fault-tolerant hardware systems, able to run fault-handling routines, control redundancy and so on.

The difficulty of gaining funding even for the initial stages of concept design led Igor on to a discussion of the relative merits of different design methodologies, and their costing.

The one which takes most trouble during the initial concept/design stages (and so costs more at this point) often pays off in the end, with longer product life, less maintenance (for which read bug-hunting) and so on... but it is often difficult to get this across to managers/customers who see lots of money going in and nothing as yet coming out!

What are the 'big players' doing? And even if they tell you something, can you believe them?

Future developments

The next speaker was Jon Honeyball, who opened by announcing that he had a real job, as a consultant! It is believed that this was Jon's response to being called a journalist. Still, the content of his talk, on future developments in microcomputing, is of equal importance to both trades, and Jon does write as well as consult. And he's the only person I know with a bigger home computer rig than the Robertson one!

Jon described the shifts in computing focus brought about by the introduction of the PC – most of all the drastic increase in speed. Two sorts of speed, actually, both the decreasing generation time for new technology/software and the increasing speed of the beasts themselves – illustrated by tales of the "Benchmark" speed-testing programs which used to be timeable with stopwatch but now go so fast they exceed poor Jon's reactions!

Then on to the difficulty of what's actually going on. What are the "big players" doing? And even if they tell you something, can you believe them? This gives Jon problems sometimes, he does quite a lot of beta-testing, and so the muffled noises he sometimes makes owe little to a mouthful of chocolate and much to the dreaded Non-Disclosure Agreement. But current trends include the growing acceptability of using processor power to make life easier, and the increase in networking and interlinking of small units.

Lunch! Enjoyed by all (OK one vegetarian had forgotten to say that she didn't eat fish either, so I almost got two starters!), with a diversity of conversation from the future of IAPetus to jazz drumming with a spate of bad jokes in between.

Quality systems

And back, to Professor Darrell Ince of the Open University. Who needs Quality? he asked, and proceeded to explain that we all do. Quality isn't just clipboards and procedures, it's about learning from the cockups of the past, from holding post-mortems of projects that didn't work as well as reusing ideas from the ones which did.

In his early days, Ince confessed to being marked down as a deviant because he not only liked writing code he liked testing it too. Or was this more that he didn't like retesting

it later! Foreshadowing a later talk, he described C as a "programming language that is the work of the devil".

"quality is a state of mind. Not conformity with protocols but a way of thinking that should permeate through everything that you do"

A project is a dynamic entity. Change is a fact of life. A quality system must address change... if students knew what software development is like they'd take up something tractable like crocodile wrestling instead. Still, project reviews tend to be successful because if you put four software engineers in the same room it is likely that at least one of them is competent.

Like many things, quality is a state of mind. Not conformity with protocols but a way of thinking that should permeate through everything that you do. But I still haven't discovered why any company with a fish tank in the reception room doesn't get through an Ince BS5750 audit!

Designing for safety

And so on to Paul Ganney, talking about designing with an eye to safety. Good idea, particularly if, like Paul, you happen to program computers for use in the NHS. Most of his comments are, however, just as useful whatever purpose your code is used for, even if its failure will lead to no more than your own red face!

From his particular standpoint, Ganney defines safety as a measure of the program's ability to harm people if it goes wrong. Nothing about its functionality, the concern is not does it do what it should, but how does it do it, and what happens if it fails? From this definition comes a need for a system specification that highlights the safety critical areas, and use every means possible to avoid places where errors might creep in.

Some of his suggestions include the use of meaningful names for routines, flags etc. (I still remember

Continued on Page 6

Continued from Page 5

one row with my husband over my using "funny_money" as a flag for a currency conversion!) and the very sensible idea of externalising configuration as much as possible – let the user change the colours without the need to recompile the whole program, for example.

Use and reuse proven code, with a "black box" approach – skeleton routines which you know work, just change as little as you have to. Apply rigorous standards, so that everybody can follow what's there – it makes maintenance so much easier if you don't have to decode a certain programmer's quirks to follow what he was doing.

And it's not just the code. Think about the user interface. Make it as consistent as possible over a wide range of systems, give them on screen help and make it as friendly and tailorable as you can.

Expect that things will go wrong – users press the wrong buttons, the operating system or the hardware might play up, or the modem link... and there's always a mug of coffee lurking around. Prevent any such errors by anticipating them, have an "undo" feature and make sure that you always have a fail-safe situation.

Test procedures are important – have two lots, the development tests you use whilst writing it and the acceptance tests used once its finished, with different test data (and preferably different testers!). And don't tamper – once code has been validated leave it alone (or retest it completely if you really must change something).

And the dreaded documentation. User guide, programmers guide,

maintenance guide (a full journal of everything done), embedded comments. Don't rely on memory. It fades, even if the head containing it is still there, not in Ulan Bator or under a bus! (I take this to heart – I have an eidetic memory and still take notes!)

Do you C?

Onwards, via tea cups and more biscuits, to Ian Godman and "Why C should be killed". Always one for an assassination, I listened eagerly as Godman said that his ambition was to become an ex-C programmer.

The job of any computer language, he said, was model your solution to whatever problem caused you to write a program in the first place, then the computer turns the model into actual physical programs. C can be used to model absolutely anything (surely that's good?)...even that which you do not wish to model (Aha!).

1993 Conference

An error may be defined as an unacceptable action performed by the finished (you thought!) program. C is good at hiding things, even from its own compiler. So you think you have a working program because it compiled without mishap, then it goes off blithely doing its own thing. That's when the fun starts, because it isn't at all obvious what the actual structure is when you look at the source.

The very terseness of the language makes it unintelligible. In short, using C it is very easy to shoot yourself in the foot. (And with C++ it isn't so easy but when you do you are liable to blow your leg off!)

The result of all this is that C costs money and time (which costs money) in getting rid of all the bugs... sorry "undocumented features"...that sneak in, which if you wrote in some other language would not get there, the coding protocols eliminate them or the compiler slings them out. So pick a language appropriate to the problem, don't just write in C because you can do anything in it!

Exit Godman stage left in hail of

"pick a language appropriate to the problem, don't just write in C because you can do anything in it"

bullets. No, I think most people agreed. For some reason C is popular, and there is a lot of work available. But its flaws are becoming widely recognised and people are being more cautious or using something else.

Customer expectations

Nick Swain then arrived and painted a rosy picture of a project sign-off meeting with everybody singing his praises and saying "Thank you for my lovely software". His talk focussed on the gap between what the customer thought that he was getting and what the programmer thought that he was supposed to be providing.

It's all about unrealistic expectations: the customer knows his business, not computing, and thinks he's told the programmer everything that he needs to know.

So the trick is management of customer expectations. Thrash the whole thing out in writing before you start, but that is only the beginning. Use prototyping to check that what you thought they wanted is what they thought they wanted.

It also makes them feel involved in the project all the way through, and minimises the chances of your turning up with the finished code to have it thrown out and you with it, followed by cries of "If we see you again we'll set the dogs on you!"

However, it will be the other way around at next year's conference – if we DON'T see you we'll set the dogs on you! This one was good, maybe next year it will be even better.

If you'd like to read the papers, the Institution office has copies. Next time come and hear them for yourself.

IAPetus is the Quarterly Bulletin of the **Institution of Analysts and Programmers**. The Editor is Megan C. Robertson. All views expressed herein are those of the authors, and do not necessarily reflect the Institution's or *IAPetus'* opinions or position. All material is © Copyright The Institution of Analysts and Programmers 1992. Produced by Breeze Ltd, 061-792 4442.

Correspondence about *IAPetus*, contributions etc. should be sent to the Editor at 12 Bude Close, Crewe, Cheshire CW1 3XG (Tel: 0270 500565). Correspondence about the Institution should be sent to Charles House, 36 Culmington Road, London W13 9NH (Tel: 081 567 2118, Fax: 081 567 4379).

The Beginner

R.A. Pachner

I used to own and use a Beeb. Mostly for word processing (View) which resided inside the machine on a ROM. Nice and simple.

My information was all stored on big black floppy disks and I could send it all to a printer whenever I needed to. Then I sold it, hoping to do better than keep everything as documents that were only 6 to 9 pages long.

Thanks to help from unexpected quarters I now have a 386 PC sitting on my desk. I also have a small library of manuals which are supposed to aid me in finding my way around the multitude of applications that reside on the machine's hard disk.

"the manuals seem to speak a totally different version of the Queen's English to the one I was taught"

Unfortunately, the manuals seem to speak a totally different version of the Queen's English to the one I was taught at school, shortly after Neanderthal Man left his home cave. After spending several days reading DOS guides, User Guides and Reference Guides, I was as bewildered as ever.

Try something, I thought.

1. Connect everything to the largest box, then connect that to the mains...

Whirring noises, flickers of life on a screen and information about bits being hunted down, tested and passed as OK, Not Found or Not Supported.

2. Horrible sounds of ball race or bearing dying of terminal lack of lubrication or forty thousand years of overuse...

It was the cooling fan in the monitor. A small touch of engineer's "thump" and the problem was cured.

3. A Menu appeared... "Press Number and enter"...

Enter what? None of the keys on the (comparatively) enormous keyboard had that word written on them. No doors were visible that would accommodate an overweight wrinkly. Find a book...

"Enter, also called Return". From where? To what?

"The RETURN or ENTER key is the one with the crooked arrow"

AHHHHHHHHH!

I chose "Word Processor", number 2, stood up to make myself a cup of coffee while the program loaded and found that it already had... quick that.

I looked for the User Guide, there wasn't one...?????

With the help of my friendly grey rodent, I positioned the cursor at the top left of the screen, where it said "File" and pressed the button... nothing happened, it was the wrong button. Pressing the correct one gave me another menu, this one had 6 choices and I clicked on "NEW".

The screen changed, and I was ready to begin typing.

I finished typing the required text into the Word-Processor and, using the 'Page Up' key, returned to the beginning in order to check for spelling mistakes, punctuation etc. then I thought... hadn't the man mentioned a 'Spell-checker' or was that something only used by Sorcerers?.

Across the top of my TV page (sorry VDU) were a number of small words, each containing a highlighted letter...

File (F), Edit (E), View (V), Insert (I), Style (S), Layout (L) and Utilities (U). Using my friendly grey rodent I positioned a small red blob on each in turn (apart from File, as I had already used that one) and pressed the button... and there it was, under UTILITIES, 'Spell Checker, Global'.

Did I really want to check the spelling of every word in the world, I asked myself?, then realised that it probably meant only those words that were in the textual 'World' of my article.

Pressing the FGR button when the red blob was in the correct position produced a whirring from the disc and a large grey rectangle at the top of the screen, a highlighted word

and a number of alternative spelling suggestions, also the choice of changing the word, ignoring the word or adding it to my personal dictionary...

How can I tell it that I don't own one?, a Dictionary that is.

I did not realise how bad my spelling was, or how many alternative words would be presented for my selection. Then I thought 'but I'm sure that the word 'COLOUR' has got a U in it'.

I went next door to see if I could borrow their Dictionary, djst two cheque.

SUCCESS!!!

I have 'Word Processed' and printed out my first document, nicely laid out, spell-checked, proof read and generally messed about with. I took it around to a (different) neighbours house to show him what goodies my new toy could produce, only to find that he had just bought a computer.

I have now become an EXPERT. (At least in HIS eyes).

Having finished my article, I decided to try some drawing. This was a mistake as I now can't find my article. I know I left it somewhere in the computer (at least I think I did) so I must be able to retrieve it mustn't I?...

Go to the DOS prompt, type DIR and press Return (it said in the book). A lot of words and numbers flashed before my eyes and I thought 'It's Broke'.

Suddenly the cateract of words ceased and I was presented with a list of about 6 items, dates and times...

None of them look remotely like my article.

I know... WINDOWS, that should help.

First find the book...

NO BOOK.

(*%\$£\$%^&")

The author, after a varied career with the RAF and as a store man, has been inspired by his early attempts to make his PC go, and started on a BSc in Computer Science in October. How did you lot start on your career in computing?
— The Editor

Never let the expert write your manual

by Brian Halley FIAP

I recently purchased a British Telecom answerphone. The manual, though well laid out, was so grossly inadequate that I wrote to BT's technical publishing department pointing out 21 deficiencies.

A polite letter of response agreed with 18 of them. All of these resulted from lack of planning and forethought by the writer, not the technical incompetence of the equipment. If a professional technical writer can get it so wrong, what will a non-professional be capable of?

Journalist or programmer?

Would you commission a journalist to write your programs? Then why expect a programmer to write manuals? No doubt many will answer "Because he understands the system" and that is the very heart of the problem. He understands the system too well. No matter how fastidious, there is a danger that in producing manuals for the systems they've written, the expert will gloss over points, leaving end users floundering for lack of information.

"There is strong argument in favour of producing a first draft of the manual before a single line of code has been written"

If the programmer has to write the manual it is often written in a hurry. Scenario: The program is complete. The Marketing Director wants it out there with the distributors. The programmer has a week to present it to the printer. It may look good, thanks to DTP, but reviewers slate it, users hate it and the firm's reputation is at stake – no matter how good the software. Quite apart from all that, most good programmers get paid more than writers, which makes for an expensive manual.

There is strong argument in favour of producing a first draft of the manual before a single line of code has been written. Next to analysis down to third normal form, this would be much appreciated by programmers.

It means bringing the technical writer in at the design stage, but this should result in suggestions that could improve the system's usability

and therefore the firm's reputation.

Standards

Standards for technical manuals were established in 1974 under BS4884, although you wouldn't think so from the standards that generally pertained in the computer industry until fairly recently. The publication in 1988 of the Department of Industry's Instructions for Consumer Products marked a turning point. More writers began to think about conventions of text as well as pretty layout. Some even produced spiral bound books which would stay open at the right page (oh the wonders of modern technology!).

There are usually clear headings and sensible paragraphs numbering systems and better layout with fewer words to the page, but even so not all products come up to this standard. I still see works that look as if they've been photocopied (even from Borland!) and there are some pretty obvious "bludners". Like numbering disks 0 to 5 and the paragraphs that instruct the user what to do with them 1 to 6.

Few of the manuals I use contain proper indices. Most of them refer to the order of occurrence of a term, leaving the user to wade through the references to find the one that gives the first practical information about using it. There are many holes in some indices, missing references or a solitary reference with no context.

The index problem is quite likely the result of writers using index software, which they assume will do an adequate job. But just as word processors don't write letters by themselves indexing software is only an aid to those who know about indexing.

Clear terminology

The worst sin of all is the use of loose terminology. It should be established at the outset whether the term (for instance) VDU, console or screen is going to be used and if more than one term is necessary, in what context each should be used. Are users going to be asked to "type", "press" or (thanks to Uncle Sam) "hit" keys. How do you explain to an inexperienced how to key the Ctrl W combination. The instruction "Ctrl + W" can result in someone pressing three

keys. "Press Ctrl W" might be interpreted as "press the Ctrl key then press the W key". I've also seen users holding down both keys waiting for an instruction to let them go again whilst the screen fills up with garbage.

The correct instruction may be "Hold down the Ctrl key with the forefinger of your left hand and press the W key once with the forefinger of your right hand". This may sound pedantic, but some people need that level of help (though maybe not all through the book) and manual writers have to write somewhere near the lowest common denominator.

The English language is one of the most powerful in the world. With over half a million words to play with the English writer has more material than the French (150,000) or the Russians (130,000) and is therefore capable of a greater degree of obfuscation (if you understand my meaning).

Communicating with humans

Programmers pride themselves on writing succinct code that works accurately and fast. But communicating with a machine is quite different to communicating with humans. When wrong syntax is used in conversation the recipients eyes don't flash an error message, though the brain might be thrashing around for a meaning. At least the person can ask you for an explanation. Once your words have congealed into print the only way to get them explained is to phone the supplier, and every phone call costs that supplier money.

A professional writer, who knows nothing about the program, will have to pass through the same learning curve as the end users. A good technical writer will make suggestions that may improve the user interface and will explain the use so clearly that reviewers and users will hold it up as an example to all. So, don't skimp on the manual – use the right sort of expertise.

The writer is a technical writer and journalist, and a member of the Society of Authors as well as a Fellow of this Institution.