We Are Sorry to Inform You . . .

Simone Santini, University of California, San Diego

Once upon a time there was a little-known patent clerk in Bern who received a disappointing annual performance review in '05 (www.norvig.com/performance-review.html) ....

How much damage could be caused by a peer reviewer having a bad day?

E.W. DIJKSTRA

"Goto Statement Considered Harmful." This paper tries to convince us that the well-known goto statement should be eliminated from our programming languages or, at least (since I don't think that it will ever be eliminated), that programmers should not use it. It is not clear what should replace it. The paper doesn't explain to us what would be the use of the "if" statement without a "goto" to redirect the flow of execution: Should all our postconditions consist of a single statement, or should we only use the arithmetic "if," which doesn't contain the offensive "goto"?

And how will one deal with the case in which, having reached the end of an alternative, the program needs to continue the execution somewhere else?

The author is a proponent of the so-called "structured programming" style, in which, if I get it right, gotos are replaced by indentation. Structured programming is a nice academic exercise, which works well for small examples, but I doubt that any real-world program will ever be written in such a style. More than 10 years of industrial experience with Fortran have proved conclusively to everybody concerned that, in the real world, the goto is useful and necessary: its presence might cause some inconveniences in debugging, but it is a de facto standard and we must live with it. It will take more than the academic elucubrations of a purist to remove it from our languages.

Publishing this would waste valuable paper: Should it be published, I am as sure it will go uncited and unnoticed as I am confident that, 30 years from now, the goto will still be alive and well and used as widely as it is today.

Confidential comments to the editor: The author should withdraw the paper and submit it someplace where it will not be peer reviewed. A letter to the editor would be a perfect choice: Nobody will notice it there!

E.F. CODD

"A Relational Model of Data for Large Shared Data Banks." This paper proposes that all data in a database be represented in the form of relations—sets of tuples—and that all the operations relative to data access be made on this model. Some of the ideas presented in the paper are interesting and may be of some use, but, in general, this very preliminary work fails to make a convincing point as to their implementation, performance, and practical usefulness. The paper's general point is that the tabular form presented should be suitable for general data access, but I see two problems with this statement: expressivity and efficiency.

The paper contains no real-world example to convince us that any model of practical interest can be cast in it. Quite the contrary, at first sight I doubt that anything complex enough to be of practical interest can be modeled using relations. The simplicity of the model prevents one from, for instance, representing hierarchies directly and forces their replacement with complicated systems of "foreign keys." In this situation, any realistic model might end up requiring dozens of interconnected tables—hardly a practical solution given that, probably, we can represent the same model using two or three properly formatted files.

Even worse, the paper contains no efficiency evaluation: There are *no* experiments with real or synthetic data to show how the proposed approach compares with traditional ones on real-world problems. The main reason for using specialized file formats is efficiency: Data can be laid out in such a way that the common access patterns are efficient. This paper proposes a model in which, to extract any significant answer from any real database, the user will end up with the very inefficient solution of doing a large number of *joins*. Yet we are given no experimental result or indication of how this solution might scale up.

The formalism is needlessly complex and mathematical, using concepts and notation with which the average data

Continued on page 126

Continued from page 128

bank practitioner is unfamiliar. The paper doesn't tell us how to translate its arcane operations into executable block access.

Adding together the lack of any real-world example, performance experiment, and implementation indication or detail, we are left with an obscure exercise using unfamiliar mathematics and of little or no practical consequence. It can be safely rejected.

A. TURING

"On Computable Numbers, with an Application to the Entscheidungsproblem." This is a bizarre paper. It begins by defining a computing device absolutely unlike anything I have seen, then proceeds to show—I haven't quite followed the needlessly complicated formalism—that there are numbers that it can't compute. As I see it, there are two alternatives that apply to any machine that will ever be built: Either these numbers are too big to be represented in the machine, in which case the conclusion is obvious, or they are not; in that case, a machine that can't compute them is simply broken!

Any tabulating machine worth its rent can compute all the values in the range it represents, and any number computable by a function—that is, by applying the four operations a number of times—can be computed by any modern tabulating machine since these machines—unlike the one proposed here with its bizarre mechanism—have the four operations hardwired. It seems that the "improvement" proposed by Turing is not an improvement over current technology at all, and I strongly suspect the machine is too simple to be of any use.

If the article is accepted, Turing should remember that the language of this journal is English and change the title accordingly.

C.E. SHANNON

"A Mathematical Theory of Communication." This paper is poorly motivated and excessively abstract. It is unclear for what practical problem it might be relevant. The author claims that "semantic aspects of communication are irrelevant to the engineering problems," which seems to indicate that his theory is suitable mostly for transmitting gibberish. Alas, people will not pay to have gibberish transmitted anywhere.

"IBM has decided to stay out of the electronic computing business, and this journal should probably do the same!"

I don't understand the relevance of discrete sources: No matter what one does, in the end, the signal will have to be modulated using good old-fashioned vacuum tubes, so the signal on the "channel" will always be analogical.

A running example would have helped make the presentation clearer and less theoretical, but none is provided. Also, the author presents no implementation details or experiments taken from a practical application.

Confidential comments to the editor: The only thing absolutely wrong with this paper is that it doesn't quite "resonate" with what the research community finds exciting. At any point, there are sexy topics and unsexy ones: these days, television is sexy and color television is even sexier. Discrete channels with a finite number of symbols are good for telegraphy, but telegraphy is 100 years old, hardly a good research topic.

The author mentions computing machines, such as the recent ENIAC. Well, I guess one could connect such machines, but a recent IBM memo stated that a dozen or so such machines will be sufficient for all the computing that we'll ever need in the foreseeable future, so there won't be a whole lot of connecting going on with only a dozen ENIACs!

IBM has decided to stay out of the electronic computing business, and this journal should probably do the same!

C.A.R. HOARE

"An Axiomatic Basis for Computer Programming." I am not sure I understand this article. It claims to be about programming, but it doesn't contain a single line of code.

The paper introduces the idea that certain inference rules can be associated to statements in a program and used to show that the program does indeed compute what it is supposed to. I have some reservations that the program's *purpose* can be defined in the terms the author claims—we all know how fuzzily defined the features of real programs are—but the idea, if suitably justified, might have some merit. However, in its current state, the work is far too preliminary to be considered for a journal. It may well be insufficient for any kind of publication, so I would advise the author to try a workshop at which these kinds of preliminary ideas will be more likely to find a home.

Before the author attempts journal publication, he should complete this work in several respects. The method assumes that the function of a program can be specified as the final value of certain variables. This is an unrealistic view for interactive programs: The author should show how his method fits with the industry's standard way of specifying requirements. He should also extend the method to be applicable to a standard programming language such as COBOL or PL/I and provide the details of his implementation, possibly with a few graphics to show how the system works in practice.

Until this is done, I fear the work is too tentative and preliminary for publication.

R.L. RIVEST, A. SHAMIR, AND L. ADELMAN

"A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." According to the (very short) introduction, this paper purports to present a *practical implementation* of Diffie and Hellman's public-key cryptosystem for applications in the electronic mail realm. If this is indeed the premise, the paper should be rejected both for a failure to live up to it and for its irrelevance.

I doubt that a system such as this one will ever be *practical*. The paper does a poor job of convincing the reader that practicality is attainable. For one thing, there is the issue of the number n used to factor the message.

The scheme's security relies on the factorization of n in prime factors taking so long as to be impractical. The authors also stress that the encryption algorithm must be fast and—if their application, electronic mail, is to make sense—the algorithm should run on all sorts of machines. Let us be generous and assume that every computer user has access to a latest-generation minicomputer such as the VAX. This 32-bit machine's speed considerations limit the choice of n to $n < 2^{32} = 4,294,967,296$. Granted, this is a large number, but by the very results of the paper's Table 1, it can be factored in a couple of hours. Scarcely a time margin that will grant security!

Further, as the authors acknowledge, a data encryption standard already exists, supported by both the US National Bureau of Standards and IBM, currently the largest computer manufacturer. It is unlikely that any method that runs counter to this standard will be adopted in any significant degree. True, the IBM method presents the problem of distributing the encryption key, but their method is a standard and we must live with it. Instead of creating nonstandard methods that will soon be dead for lack of users, the authors should try to extend the standard and devise ways to distribute the encryption keys securely.

Finally, there is the question of the application. Electronic mail on the Arpanet is indeed a nice gizmo, but it is unlikely it will ever be diffused outside academic circles and public laboratories—environments in which the need to maintain confidentiality is scarcely pressing. Laboratories with military contracts will never communicate through the Arpanet! Either normal people or small companies will be able to afford a VAX each, or the market for electronic mail will remain tiny. Granted, we are seeing the appearance of so-called *microcomputers*, such as the recently announced Apple II, but their limitations are so great that neither they nor their descendants will have the power necessary to communicate through a network.

The introduction is only two paragraphs long, the relevant literature is not presented or cited, and there is virtually no comparison with the relevant work in the area. In summary, it looks as if this paper is a mathematical exercise with little originality (the authors claim that most of their ideas come from other papers), too far from practical applicability, running against the established standards, and with a declared application area of dubious feasibility. Not the kind of material our readers like to see in the journal. Reject.

… and the rest is history. ■

Simone Santini is a project researcher at the University of California, San Diego. Contact him at ssantini@sdsc.edu.

Editor: Neville Holmes, School of Computing, University of Tasmania; neville.holmes@utas.edu.au. Links to further material are at www.comp.utas.edu.au/users/nholmes/prfsn.