

COSA is spot on . . . September 15, 2009

I have finished reading your book "Breaking the Time Barrier." I found your analysis of the problem and proposed solution spot on. Your logic is impeccable. The biggest problem today with software (besides its ever increasing complexity) is incomplete logic. That incomplete logic is directly attributable to the ITE style of coding. For example, consider the following pseudo code:

```
If (1 <= buffer_size <= 1000)
Then ( ++buffer_size)
<do something>
```

The flaw in this logic is that there is no error handling when the buffer size is greater than 1000. The buffer size counter won't be incremented but the "do something" code will still be executed. A simplistic example, yes. But similar coding mistakes have led to buffer overflow and other security exploits in commercially available software.

You sum it up well on page 8: "Because most COSA tables represent binary behavior, errors in logic and operation are covered by default. This is not true in the traditional ITE approach, which requires the programmer to define a contrary action." This is the key "selling point" of this methodology.

I particularly like the embedded trace capability of COSA/BNF table. With typical programming techniques, the trace code has to be added and then compiled in/out. This actually creates different execution paths and timing with the trace on or off. Debugging time critical software becomes much more difficult in that case.

Another compelling argument for your proposed methodology is the elegance and simplicity of both the state and call diagrams. I think the examples you used had a 3-4x reduction compared to ITE counterparts. I have seen too many examples in actual coding where it is almost impossible to determine what is going on in the software because these diagrams are too complicated to understand.

Thanks again for the book. I enjoyed reading it.

John Terzakis, Staff Engineer/Project Manager, Intel

COSA provides the answer..., August 13, 2009

In 20 years of software development and testing, I have sought a methodical approach to the task. With an ingrained hatred of spaghetti code, I struggled on the fringe to bring together something similar to COSA. Morrison's *Breaking the Time Barrier* outlines a cohesive and disciplined approach that has been lacking in the software industry for decades. I am pleased to see it presented in a text, which is easy to read and provides ample examples. With a strong PASCAL and Delphi experience the book resonates with me and I hope the reader unfamiliar with PASCAL and Delphi takes the time to appreciate the approach. The benefit of built in debugging will likely save time and money on many projects. I am looking forward for an opportunity to experiment with the technique and recommend the read to anyone involved in the software industry.

T. Wickard, Director of Engineering, Souriau

Gordon Morrison has used expert knowledge of compiler design techniques and data structures to produce a beautifully elegant method to write computer programs. His book is intriguing and hard to put down once started. He shows how software is simplified when control and data are fully separated. He demonstrates the clarity of logic when conditions of data are removed from the implementation. This forces the programmer to think about the logic of the solution instead of special exceptions. And he proves that programs so written are more efficient than modern programming alternatives and are, in fact, specifications with 100% traceability and observability. I found myself thinking how straightforward implementing the control solution in programmable logic would be while the data half would easily be handled in external RAM. The concepts Gordon Morrison explains will surely become commonplace.

In case you need credentials, I am a senior member of the IEEE, hold multiple engineering degrees, and hold several patents. My current title is Director of Global Technical Marketing for Premier Farnell. I was previously a Senior Engineering Manager at Harris Corporation.

Randall Restle, Director, Global Technical Marketing at Premier Farnell

Reviews of *Breaking the Time Barrier: The Temporal Engineering of Software*

The thing that keeps me motivated on your approach is that I believe technology reaches perfection as it emulates the divine Creator's works. Since COSA works more like the human brain than the Touring model or John Van Neumann's architecture, it is no doubt a better approach.

Gregory M. Pope

Software Quality Engineering Group Leader/Science and Technology Center
SQE Project Lead/Advanced Simulation Computing V&V
Lawrence Livermore National Laboratory

Breaking the Time Barrier is short, to the point, and loaded with examples and illustrations of all kinds, perfect for the advanced engineers to which it is intended. For the non engineer, such as I, the book offers a unique and original philosophical landscape which can lead to new and necessary solutions for overworked and under productive people in today's hyper-dynamic and accelerating world. I need that.

CJ Coolridge

Author: The Squaredime Letters

The concept of streamlining and simplifying computer code seemed to be pretty self-evident. Just because something has always been done in a certain way doesn't necessarily mean that is the only or best way today. As one who remembers writing the old "if, then, else" statements in line after line in dbase programming, this seems to be a vast improvement. I am sure that if some capable software engineers start to work in this manner it will produce some revolutionary breakthroughs.

Mark Rossetti

Kemah, TX

I bought your book, thank you. I hope you will be able to promote your new paradigm. Keep me informed, I believe in it.

Uri Levy,

Business Development and Marketing
Technology Consultant
