

# Re: Books / Articles on Embedded SW Architecture

---

*Source:* <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2006-06/msg01030.html>

---

- *From:* "Paul E. Bennett" <[peb@xxxxxxxxxxxxxxxxxxxxxx](mailto:peb@xxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Sun, 25 Jun 2006 12:12:47 +0100
- 

Steve at fivetrees wrote:

Unfortunately, these options are not always available and some quick and dirty patches must be done to already fixed architecture.

Hmmm.

I know the point you're making, and I sympathise, but – let me take the harsh view and say that if an addition requires a rethink of the decomposition, then there was a problem with the decomposition. Nowadays I take the view that this is a Good Thing – these kinds of Wrong Models tend to come back and bite sooner or later. Sooner == better, and most certainly cheaper.

The thing about decomposition to reveal the structure of the problem and develop a structure for the solution is that such structures really are just a decent framework from which the rest of the edifice of the application can be supported. Think of it in architectural terms. You wouldn't go changing the steel framework of a building to something that was so radically different halfway through a construction project. It is, though, always possible to accommodate changes to the building within the supporting framework or by additions and extensions to the framework. Knowing the type of application you are dealing with is important to knowing how you should structure the framework of your system so that it is identifiable where likely client requests for changes can be accommodated with relative ease. The nearest text I have seen for component oriented development (the style I tend to employ) is those related to .NET. If there are others that are more related to embedded systems I would be interested to hear of them.

Having said that, let me add that it's taken me \*years\* to get to a point when I'm (mostly) happy with my own work in this regard. It's been quite a long time now since any last-minute change, or later enhancement, has caused me to substantially restructure a design. (One of those projects

## Re: Books / Articles on Embedded SW Architecture

has had something like 40 major additions in the last 17 years, and is still going strong. I \*did\* do a major restructure, mostly to decouple [1] the human interface further, about 10 years back, and it's held me in good stead since.) It's down to what CBFalconer said: one learns from one's mistakes [2].

Having worked for industries where last minute changes are highly discouraged (due to the cost and complexity of re-certification efforts) I have only had one occasion in 38 years where a last minute change was demanded and implemented. This involved significant effort to put in a change that ended up affecting just 10 lines of code. When I say last minute, the request came in 6 weeks before intended ship date and the client accepted the slippage of 2 weeks in delivery before we began the change process. The change did improve the system useability though.

[1] Now there's a word: "decouple". I put a lot of effort into decoupling things nowadays. Again it's part of the "avoiding unwanted interactions" thing. Vital, in my opinion.

That was always the mantra way back. "High coherence minimum coupling". It still holds very well as a guiding principle.

[2] As I get older (I'm 50 next birthday), I get more desperate to pass my hard-won skills on. Hence the book idea. Occasionally I get to do some mentoring, and I absolutely love it. At the last company I worked for, I wound up being the elder sage that the young 'uns would come to if they were stuck. I was honoured. And very pleased to find that I could almost always help – not by sermonising (as I do here ;)), but by asking awkward questions... and getting them to think differently. More, please.

Between CB-Falconer, Lewin, you and I, I think we have almost written a book on the subject just by contributing to this type of discussion in these newsgroups. At approaching 50 you are still a young whippersnapper with plenty of time ahead of you for the book writing. I have been considering it myself and have already written a few parts of several chapters. Must find some more spare time to continue with it.

and Darin Johnson wrote:–

That's true. But I also think that the majority of programmers spend the majority of their time working on code that someone else wrote. They don't get to do the design or the decomposition. So good programmers have to know how to deal with it and make the best of a bad situation.

Re: Books / Articles on Embedded SW Architecture

To implement a change in any design (hardware or software) the starting point should be a full and thorough technical review of the existing system. Without this you are just prodding the parts that stick out without full knowledge of the knock-on effects for the rest of the system.

So back to the original topic perhaps – it would be great if there were more books that dealt with this aspect of programming. The Mythical Man Month is one at least. Others?

The best pair of books on general analysis and design topics I used way back are:–

"Introducing Systems Analysis" and "Introducing Systems Design" which are both by Steve Skidmore and Brenda Wroe of the NCC. The first one's ISBN is 0–85012–630–4 but as someone borrowed the other and hasn't yet returned it I do not know the ISBN of the second one.

Forth related but a good one generally is "Thinking Forth" by Leo Brodie. This is now, fortunately, on–line at:–

<<http://thinking-forth.sourceforge.net/>>

Also good on general problem solving technique is "How to Solve it" by George Polya. The seven I have thus far suggested in this thread have stood me in good stead and all bear re–reading once in a while. Some of them may have been old publications but I see no problem in that. The message remains the same. Doing a decent job of systems development requires a certain attitude to doing that job and can be immensely enjoyable and rewarding seeing your creations doing what they were designed to do with little fuss or on–going attention required.

—  
\*\*\*\*\*  
Paul E. Bennett .....<email://peb@xxxxxxxxxxxxxxxxxxxx>  
Forth based HIDECS Consultancy .....<<http://www.amleth.demon.co.uk/>>  
Mob: +44 (0)7811–639972  
Tel: +44 (0)1235–811095  
Going Forth Safely ..... EBA. [www.electric-boat-association.org.uk](http://www.electric-boat-association.org.uk)..  
\*\*\*\*\*