



FileMaker® Pro

Thinking About Solution Design

Paper prepared by R J Cologon PhD

October 2002

Introduction

A question which is often asked is 'how can I approach designing a new database - how should I think about it'. The question is not to be treated lightly, and although the 'safe' and 'conventional' answers are in wide circulation (and data modelling and prototyping techniques certainly have their place) it sometimes happens that much of the real 'work' takes place in a less tangible way.

Devising the core of the design of a new solution is an act of creation. FileMaker Pro™, more so than many other applications and programming environments, provides a minimal set of constraints and a vast range of flexible options - almost anything that can be achieved can be approached on more than one way - and some things in literally dozens of ways.

Though this flexibility can be seen as a strength, and is precisely what attracts some to FileMaker, it also presents some challenges. Chief among these is the lack of a universal road map for how to approach solution design. It really is open ended.

This being the case, here are some notes - for what they are worth - about some of the things you may wish to think about before getting down to the business of designing a new database solution with FileMaker Pro. Before beginning to make concrete plans and structures for a new solution, I try to go through two stages of 'thinking about the structure' and two stages of 'thinking about the interface design'.

Loosely speaking, it goes something like this:

Stage 1 - Static Structure:

What are the natural 'data centres' around which information, action and change revolve?

At this point, what matters is identifying the major driving forces of the solution. There are many angles to come at it from. Who is the client (no the real client - not always the one paying the bill - not infrequently the answer is that there are several key client groups). Who benefits and how if the system is successful. Why have other systems (albeit similar in intent) fallen short or failed in some respect(s).

When this all boils down, there is generally a rather short list of the things which really matter in the solution - around which everything else will revolve in some way. Once this begins to emerge it is time to set aside this line of thinking (before becoming locked into any one perspective of the job ahead) and move on to a challenge of a different order.

Stage 2 - Dynamic Structure:

How will the rest of the database 'look' from the perspective of each of these centres (taken separately one at a time as a new vantage point) What will be the critical paths?

The answers to questions posed at this point sometimes cast the ideas which may have emerged from the previous round of thoughts into some doubt. It is at this point that one gets the opportunity to harness imagination and try to 'experience' the flow of events and information from the vantage points of different participants with different (sometimes competing) agendas.

What do they want - why do they want it - how will it get to them and where will it come from?

Stage 3 - Interface Paradigm:

Assuming that this system need not necessarily resemble any other system previously designed, what might it look like and how might users interact with it?

A key question at this point is, given all that one knows about what the system will be like and what it will do, what would be a really appropriate 'look and feel' with not only the right 'mood', but to communicate the right 'mind-set' about what the thing 'IS'?

So many solutions roll out the ubiquitous tabbed interface for yet another encore. All well and fine - and not infrequently highly appropriate. But if we are to go that way, lets have a tabbed interface because it really works and it is so 'right' for this solution, rather than because that's what we always do (or, perhaps, what everyone else always seems to do).

Stage 4 - Interface Interaction

If I were a user of such a system, how would I ideally like it to work for me? - What would I be really thrilled if it did (or if it did on a particular way)?

A system is quite likely to mirror aspects of the real world, and of the lives people live and the environments they inhabit. Some of the most successful (arguably) interfaces are that because they are so strikingly familiar - because they succeed in building and sustaining a compelling metaphor.

The answers at stage two may have given some user perspectives, but what are the metaphors that will be vehicles for translating users experience into interface actions within the solution?

More Questions than Answers

Each of the above thinking stages presents an open-ended set of challenges. It is likely to result in more and more questions occurring to the hapless thinker - creating the impression that any 'progress' has been in a backwards direction. Oddly enough, this is the point. The solution itself must provide the answers - and it is unlikely to do so unless the 'real' questions have begun to emerge before the designer has settled on a plan.

The idea, then, is not to get final answers at any of these formative stages, but to throw a number of lines - large and small - out into the 'subliminal ocean of possibilities' and see which fish are nibbling!

As a participant in the development processes for a number of systems, large and small, and as a user of systems designed by others, I can attest that all too often the thinking about each of the four areas referred to above takes place (if at all) after the solution design has been brought into existence - and not infrequently after the coding has begun.

In many cases this is quite understandable - since the tools which will be used to do the job are dictated by circumstances - and the method by which the job will be done is largely dictated by the tools.

Not so with FileMaker Pro.

Creative Solutions

Design is a creative process, solution design no less than any other.

Like any creative discipline, development in FileMaker Pro operates within some broad constraints and known limits. Without contradiction, however, the extent of flexibility and extensibility available is a sufficient basis for the delivery of a vision. But only if you have one before you start - or are able to acquire one in the course of your formative thinking.

While each developer will settle on a style and approach - and this paper is not intended to detract from that - the strategies suggested may provide a starting point for breaking out of the mindset of template-based development.

Prepared by:
R J Cologon, PhD
NightWing Enterprises
Melbourne, Australia
CobaltSky@nightwing.com.au
<http://www.nightwing.com.au/FileMaker>

© 2002 NightWing Enterprises

* FileMaker Pro is a Trademark of FileMaker Inc.
