

MVC-EAV In FileMaker Pro

Brad Stanford
Tranquility Basics
Dublin, TX

August 24, 2012

This is a short description of a FileMaker-specific data model that seeks to solve multiple problems simultaneously. This data model uses a model-view-controller environment to create records both in XML and in an entity-attribute-value setup in standard FileMaker fields.

The nature of FileMaker to be record-centric and relationship-dependent causes design issues that can reduce speed, robustness, and crash recovery time. This is my attempt to address these issues in a way that is replicable for most solutions.

The main results achieved (or to be achieved) are:

1. Simplified graph
2. Constant but simple use of ExecuteSQL
3. Use of XML to describe entire records
4. Truly separated data and UI
5. Portable controller-model code

Like most other data models, complexity has simply moved from one place to another. The fact that the complexity is handled with new tools not yet embraced by the FileMaker community as a whole might gain this method the reputation of being more difficult to implement. In reality, it is no more difficult than the typical script processes written by FileMaker developers on a daily basis.

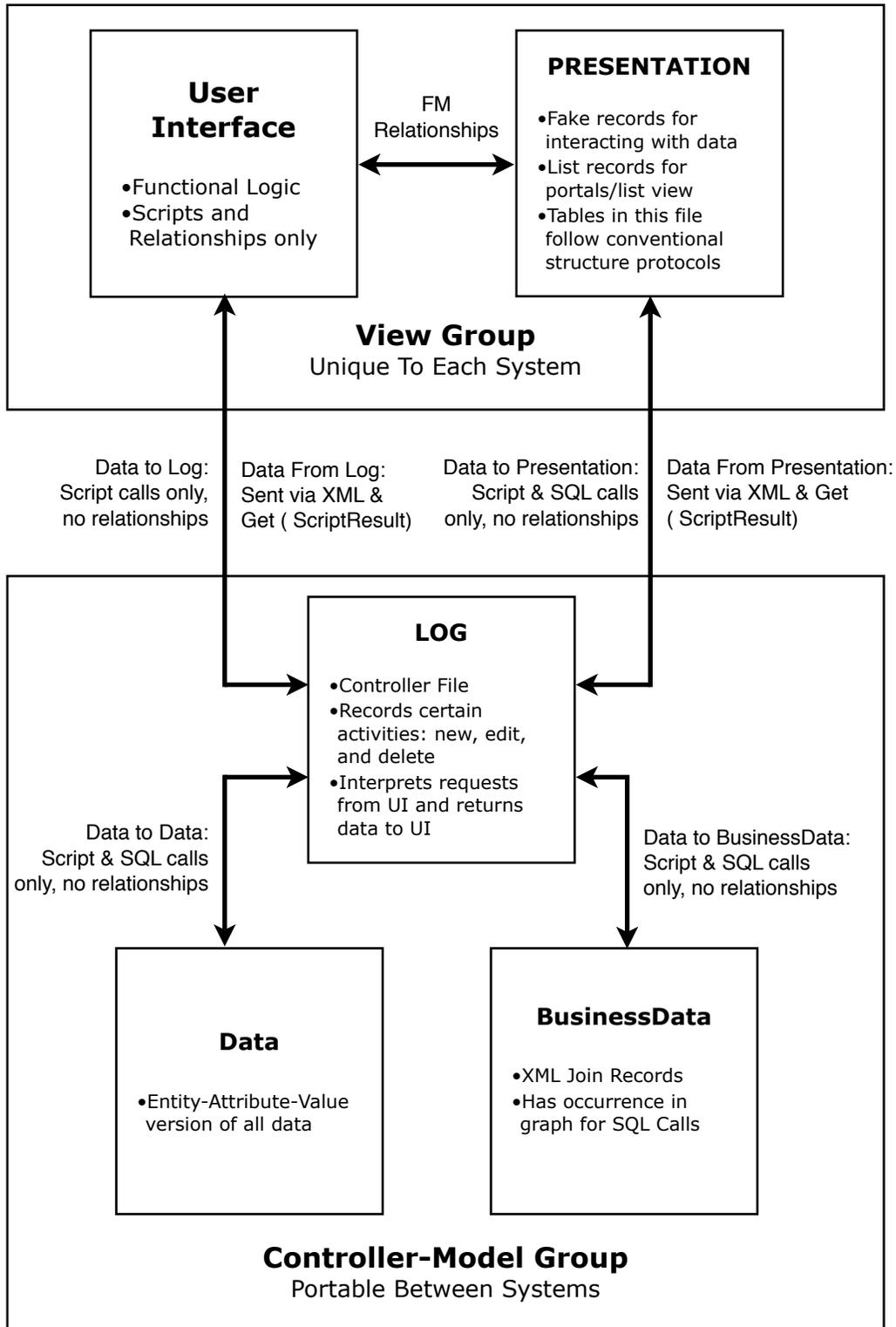
The advantages of this method are straight forward:

1. Fast finds across large data sets
2. Built-in audit trail
3. Mobile-ready
4. Best combination of network speed and FileMaker ease of UI building.
5. Once built the controller-model structure never has to be rebuilt.

Following are the basic operational and structural concepts for understanding this model.

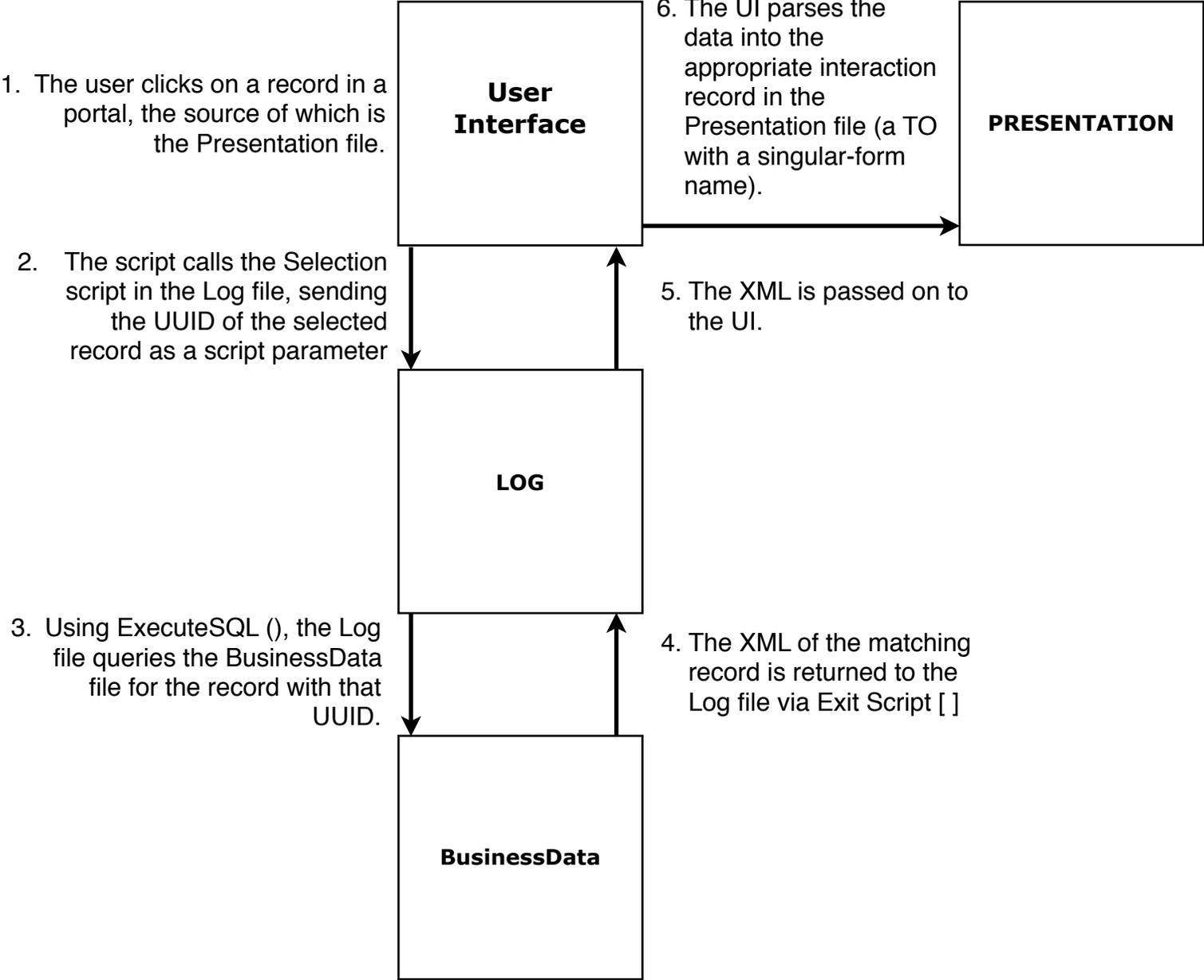
DATA MODEL

Overall System Design



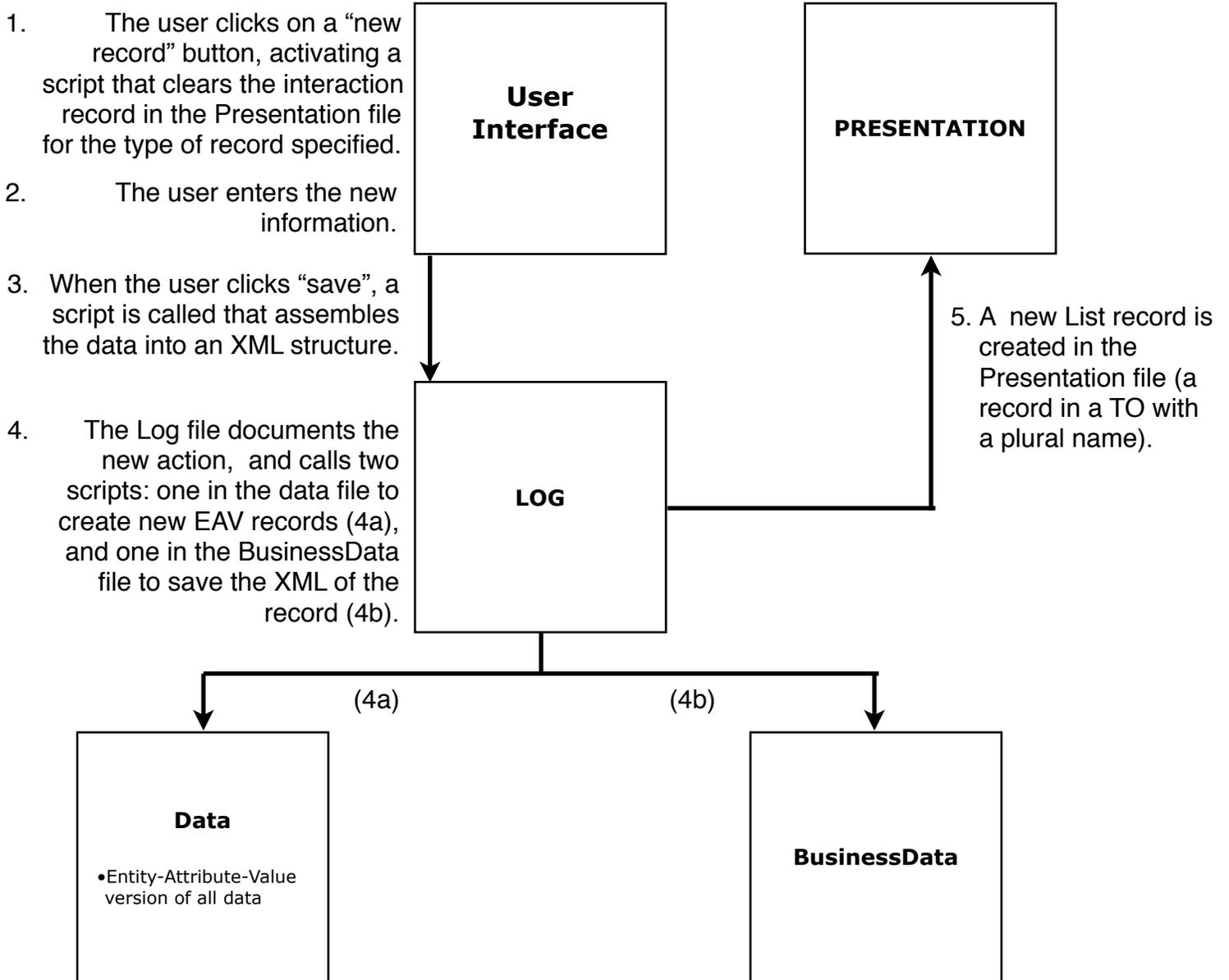
SELECT RECORD

Typical Record-Selection Process



NEW RECORD

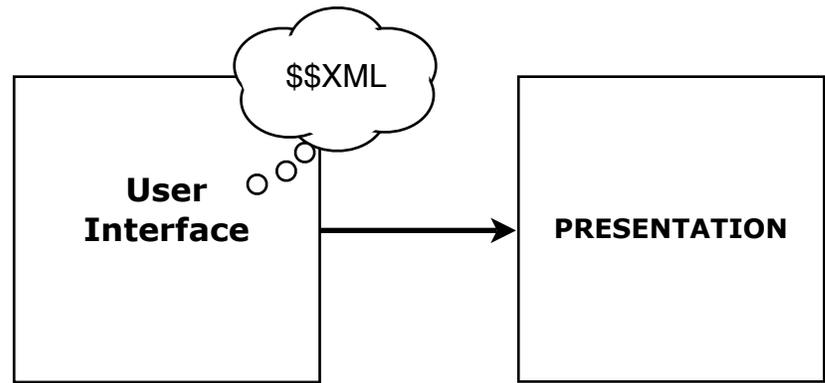
Typical Process



RECORD EDITING

Expected Editing Process

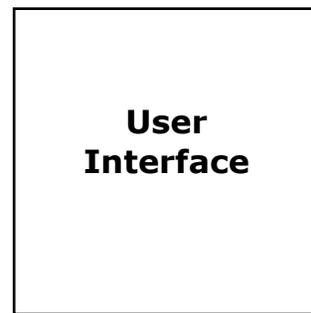
User selects or creates a record, which loads it into the presentation editing record, and into a variable.



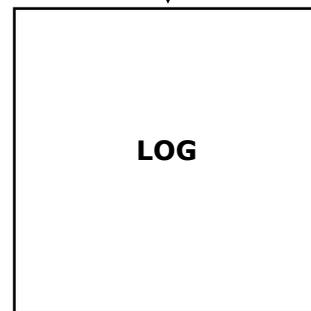
When a user exits a field, the value of that field is compared to the value that is in the XML variable.

-OR-

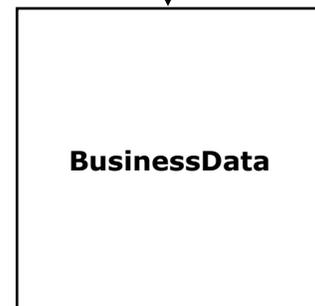
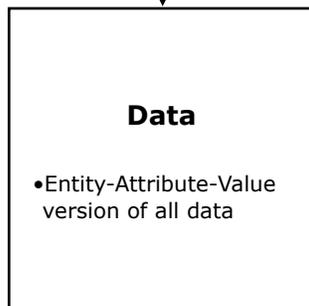
An entire record is edited and then the user initiates a Save action.



If a change was made, the change is sent to the Log file, and the XML variable updated.



The Log file updates the related records in the Data and BusinessData files.

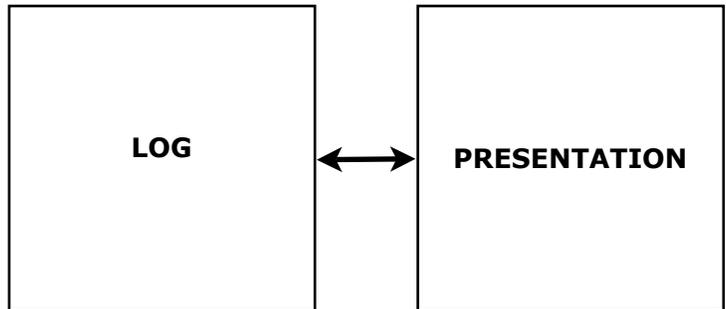


ADVANTAGES

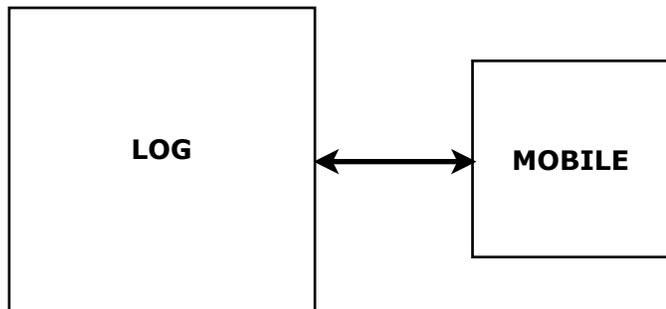
Main Selling Points

1. FAST FINDS

SQL Finds on tables with small field counts is as fast as finds can get for FileMaker



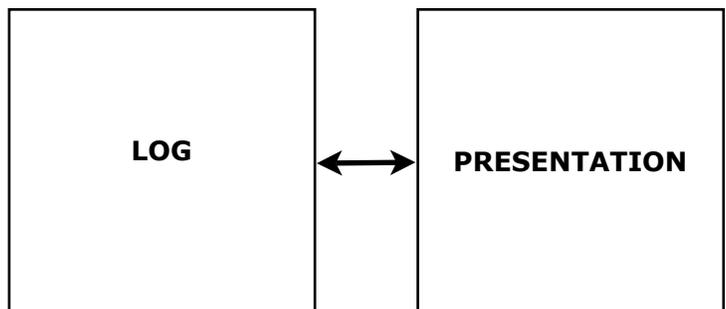
2. The Log file servers as an audit trail and a synching mechanism for mobile data.



3. The BusinessData file stores entire records in a single field. A record can be sent anywhere via script parameters or SQL calls. FileMaker is no longer pushing records over the network.



4. The Presentation file allows the network speed of the XML data, but reserves the native list & edit uses of FileMaker fields.



Closing Thoughts

I'm not suggesting in any way that this modeling system be adapted by the typical FileMaker developer. After all, the main market for FileMaker is the knowledge worker who needs something that is easy to use.

For the advanced developer, however, this model might very well have him or her working much faster than before. Once built, the controller-model unit is two-thirds of a project already finished. Not only is a two-thirds of a project, but it can be two-thirds of a very complex project.

Images and documents have yet to be incorporated into this system, but will be within the next few months.

The EAV table is not necessary, really. Its inclusion is simply to show that it can be part of the system if needed.

Projects and business models are different, but data storage is the same. This data model is set to take advantage of that.