

JMP® and SAS® Complement One Another to Produce a Virtual Laboratory

Wayne J. Levin, Predictum Inc., Toronto, ON

Brian McFarlane, Predictum Inc., Glasgow, UK

Abstract

This paper describes a system that combines the Scripting and DOE functionality of JMP with the powerful modeling capabilities of SAS to create a virtual lab. This new approach replaces the previous practice of one-factor-at-time physical experiments with a rigorous system of custom designed experiments that estimate responses using models built on years of empirical observations. The complementary use of JMP and SAS produces a productive system which has the potential for significant improvements in costs and reduced time to market for new products.

Introduction

We have had the pleasure of developing a major, mission-critical application that can best be described as a virtual lab. Our client, a well known, multi-national, chemical manufacturer has conducted tens of thousands of formulation experiments keeping the results in a large database. The goal of this application is to exploit the experience recorded in these databases.

Rather than send proposed formulations directly to the lab for construction and testing our client's Formulators are now able to construct and test them first virtually. There are several benefits in doing so. First Formulators are making use of models generated from the results of thousands of previous experiments. These models have proven to be quite worthy surrogates to real-world experience. Second, Formulators no longer waste Lab resources unknowingly making and re-making similar or near-similar formulations. Third, the virtual lab deepens the analysis associated with building new formulations and it brings new product to market in substantially less time.

With the virtual lab, we have constructed an Analytical Workflow™ to guide the user along the process of developing their study, executing it and assisting with interpreting the results. The virtual lab is being used by dozens of Formulators at Research Centers in the United States and Europe.

Please note that we are purposely not discussing all aspects of this application for two reasons. First, this is a mission critical application for our client and, as a result, they have requested we not reveal certain aspects of the application, including modeling details. Second, the focus of this paper is on JMP & SAS Integration: the whys and hows. So we will discuss only sufficient detail to explore and illustrate these capabilities.

Physical Development Versus the Virtual Lab

The former approach to Formulation development is depicted in Figure 1 on the right. Back then, it took at least 3 or 4 iterations to develop a formulation with. Each iteration requiring about 4 weeks to conduct and get all the results back from the Lab.

With the virtual lab, on the right in Figure 1, new studies require fewer iterations because they are conducted using JMP's extensive Design of Experiments (DOE) capabilities and, because the experiment is being conducted virtually, it takes only a few minutes to complete. Proposed formulations are sent to the Lab for construction and evaluation only after they have been extensively studied virtually.

Studies that usually took 12 weeks with several iterations through the physical Lab can now be completed in as little as 4 weeks with the virtual lab and with only one physical Lab exercise. The virtual approach cuts costs and accelerates development while ensuring that proposed formulations are extensively researched using all of the insights available from thousands of previous experiments.

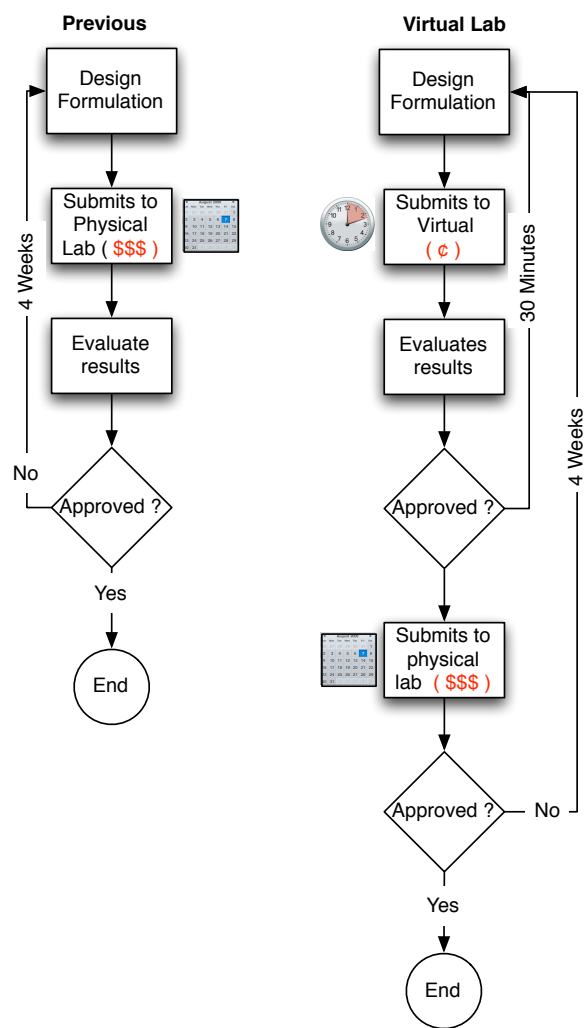


Figure 1: Previous Process compared to virtual lab

JMP & SAS: Complement and Extend Each Others' Capabilities

The virtual lab blends the best of both JMP and SAS. JMP alone could not be used because the modeling required heavy-lifting beyond JMP's capabilities. At the same time, Formulators, require the point-and-click ease of JMP as well as JMP's excellent Design of Experiments and Visualization capabilities.

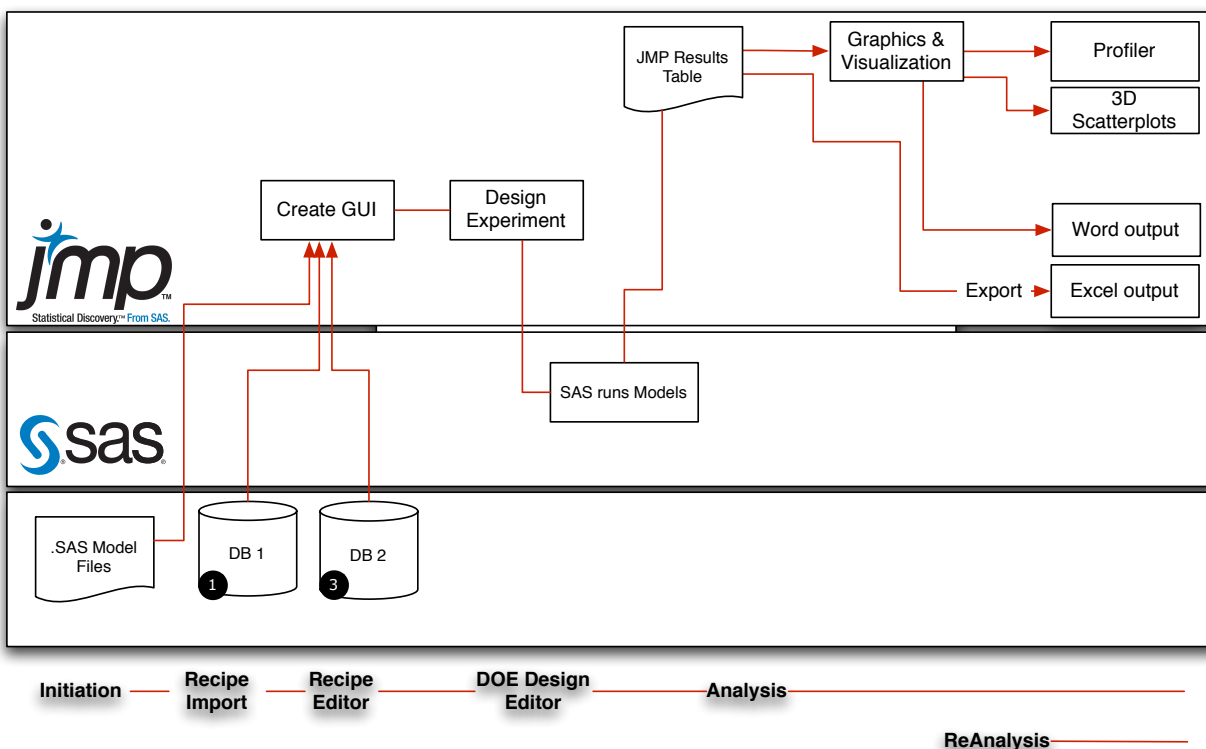


Figure 2: Virtual Lab Deployment Flowchart

The virtual lab is illustrated in Figure 2: During the Initiation Phase it pulls the latest information on formulations from various databases along with a list of the latest Models stored in a network folder as .sas files. This information is used to populate various elements of the Graphical User Interface (GUI) that is constructed in JMP. Once completed, Formulators find themselves in the Recipe Import phase.

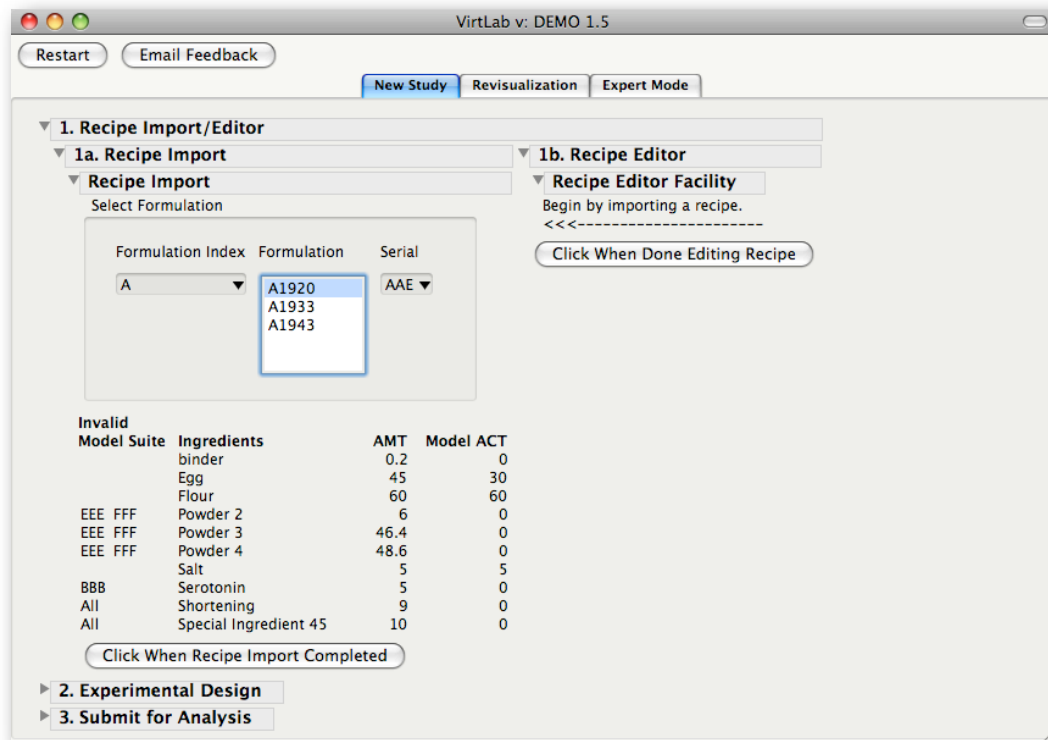


Figure 3: Virtual Initial Dialogue Box

Figure 3 shows the initial dialogue box in JMP where Formulators can import a recipe similar to one that they would like to experiment with. Formulators can navigate among the thousands of previous formulations via the GUI and observe which ingredients are not modeled in particular model suites (Invalid Model Suite). Formulators then proceed to the Recipe Editing phase by clicking on "Click When Recipe Import Completed."

The table in the Recipe Import panel shows two columns called AMT and Model ACT. AMT refers to the total amount of the ingredient used in the formulation while Model ACT refers to the quantity of active ingredient provided by this amount of the ingredient.

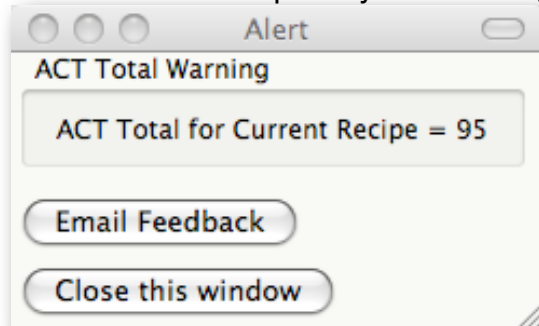


Figure 4: Warning and Email Feedback

The virtual lab evaluates the imported recipe and provides cautions where necessary at various points in the workflow, including the transition to Recipe Editing, as shown in Figure 4. Note that there is also a button that allows Formulators to send questions and feed-

back to the virtual lab developers. These buttons can be found at various points around the virtual lab's workflow.

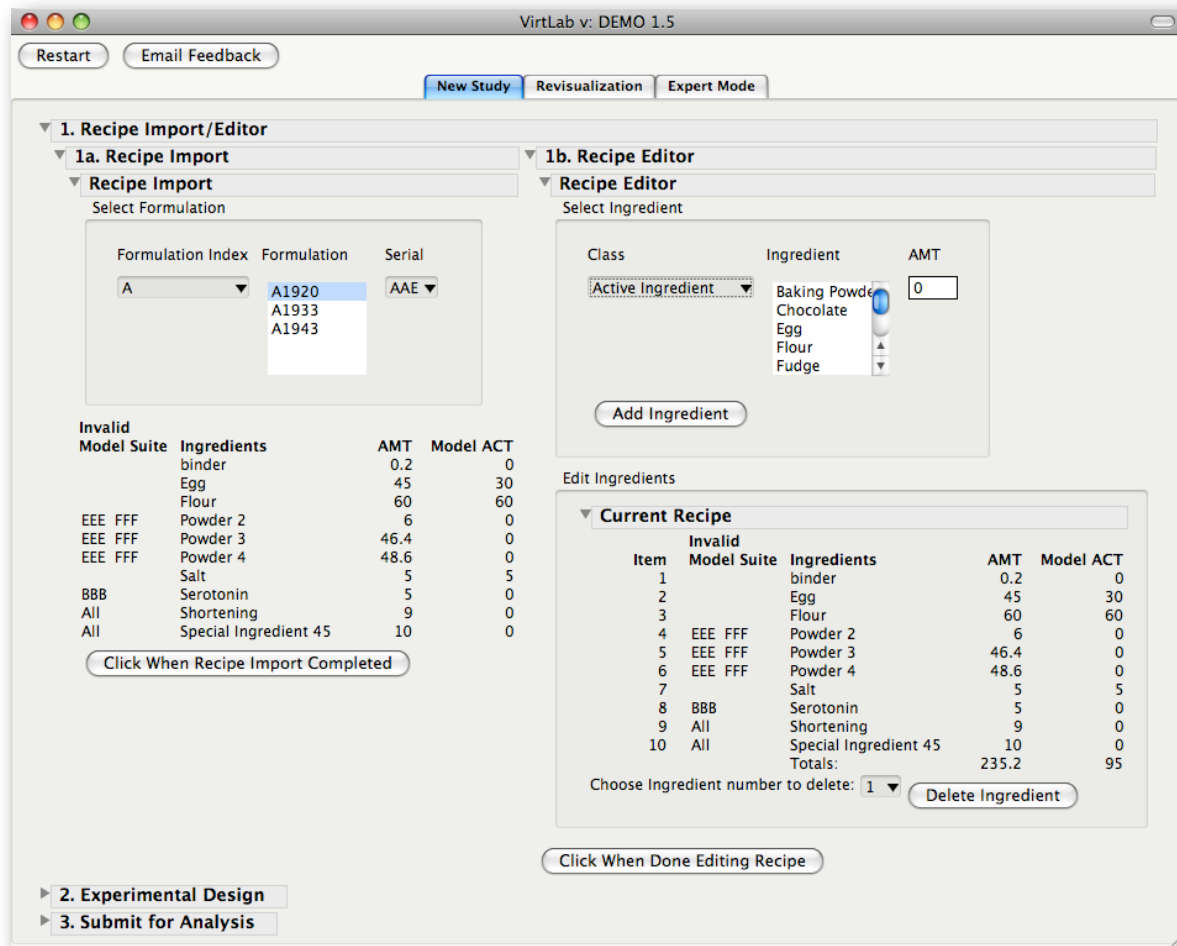


Figure 5: Recipe Editing

At this point Formulators add or delete ingredients in the desired quantities. Once Formulators finish editing their recipe, they click "Click When Done Editing Recipe" and move into the DOE Design phase.

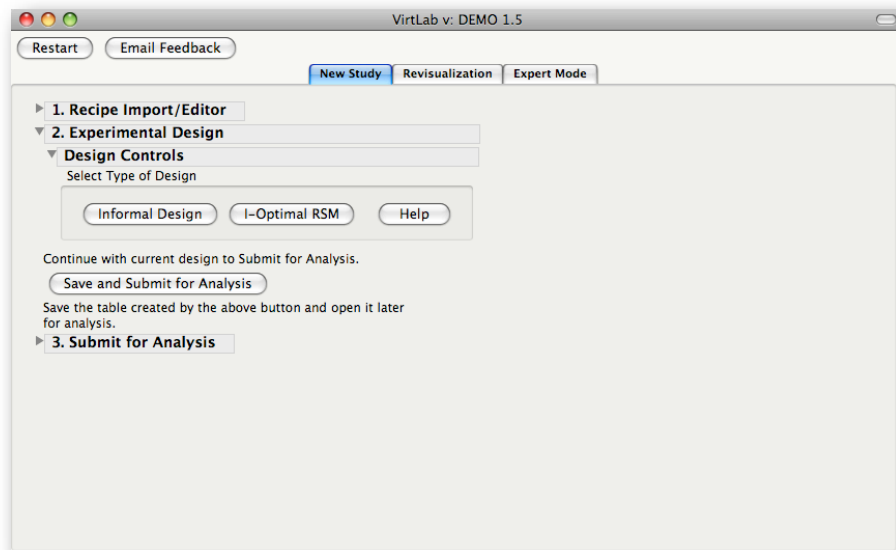


Figure 5: Choose Design Method

Formulators have a choice of approach to the design of the experiment. They can take the “informal” route, which is something like their current method by simply creating a set of specific recipes to test. Their employer, however, wants to move them into the world of modern DOE methods building I-optimal, response surface designs, which is what we'll illustrate here.

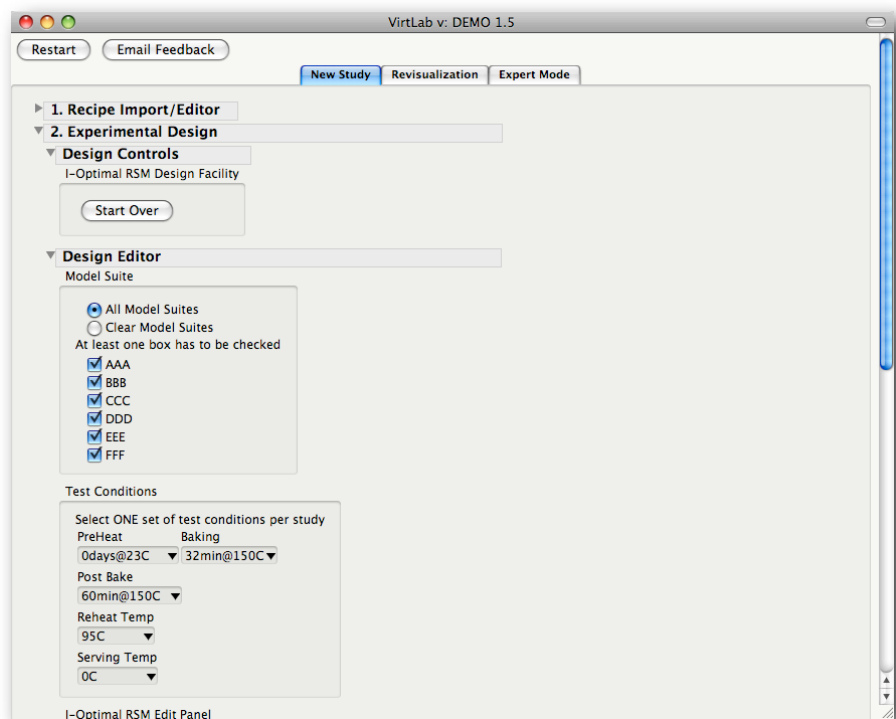


Figure 6: Model Suites and Test Conditions.

First, Formulators can choose from among the suite of Models available and set their Test Conditions to be included in the models.

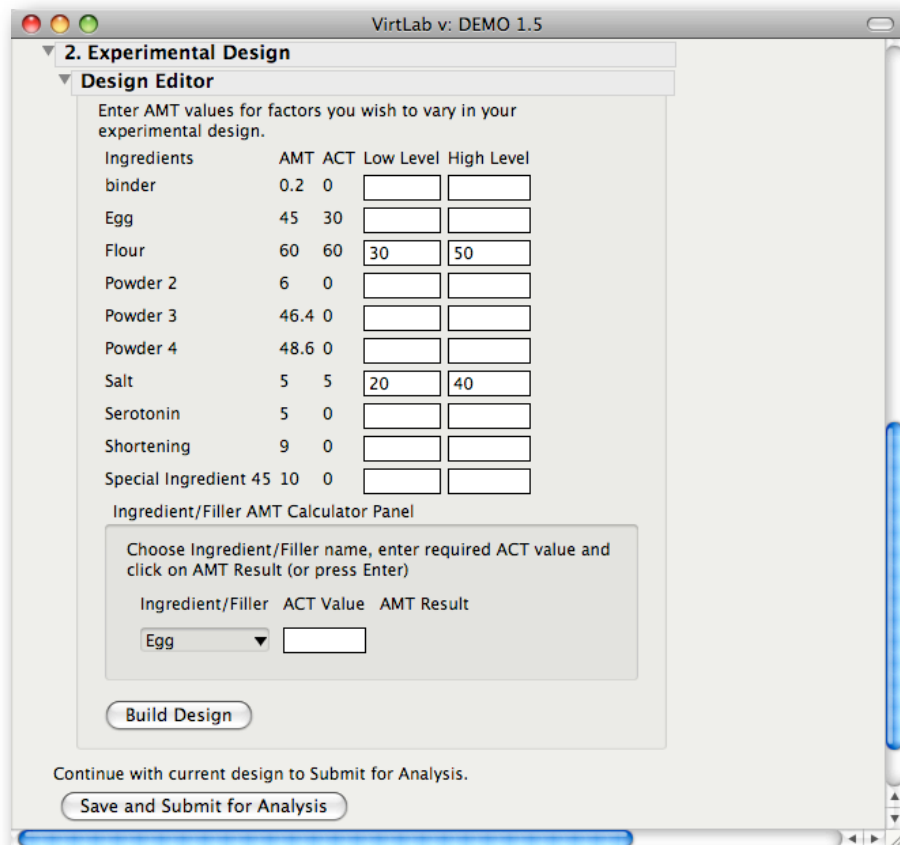


Figure 7: Setting up the DOE

Then, Formulators can establish their low and high points.

The Design Editor can be set up for an experiment using two active ingredients and as many inactive ingredients that are of interest. It is important that the levels of the active ingredients are chosen to provide a starting point that is appropriate to meet the constraints for the custom design.

In the example shown Egg is left at the original level of 30 active units leaving 70 units for experimentation with the remaining active ingredients Flour and Salt.

For the experimental factors of Flour and Salt the Low Level of one ingredient plus the High Level of the other (and vice versa) have to add up to 70 to allow JMP Custom Designer to find a valid starting point for design generation.

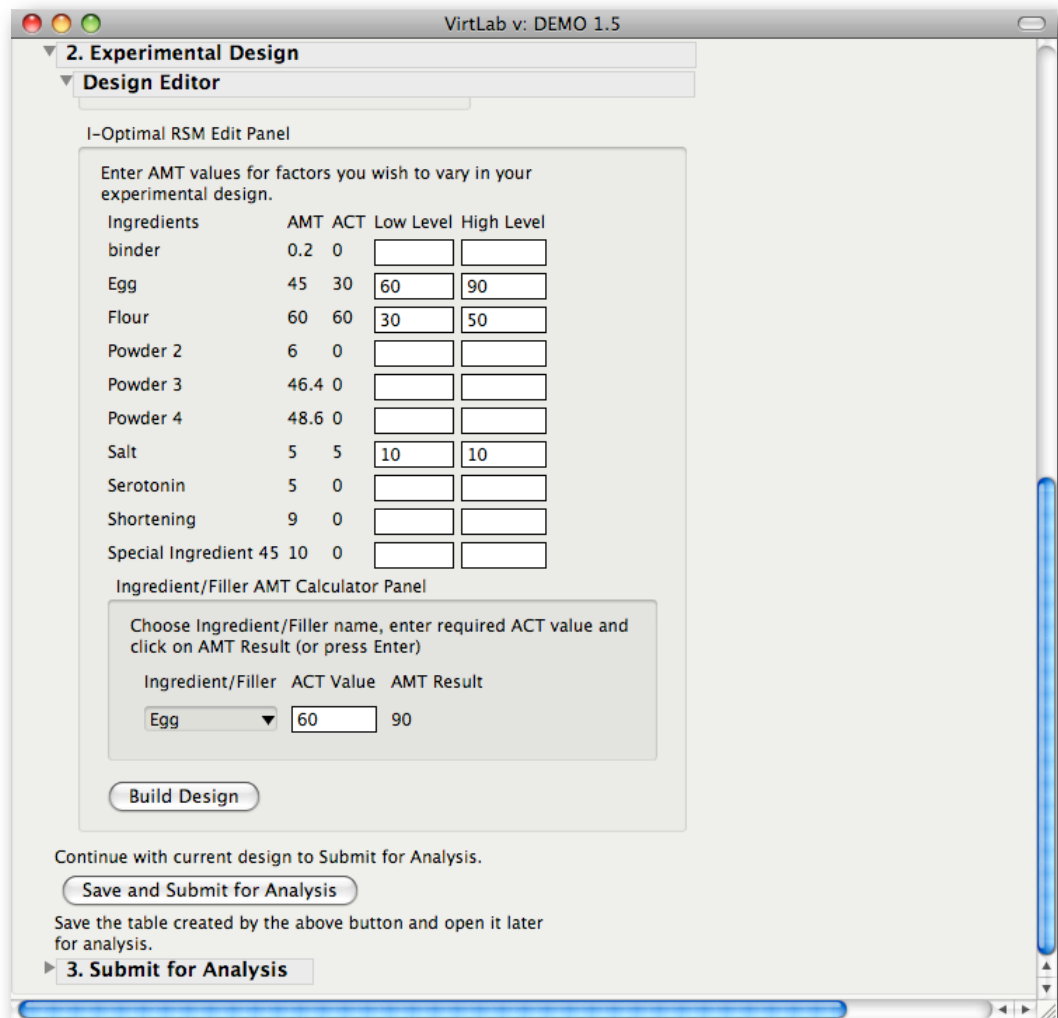


Figure 8: Facilitating Level Setting for Active Ingredients

When Egg is used as an experimental factor choosing a starting design is more complex since Egg is not 100% active ingredient, it is a mixture of active ingredient plus filler.

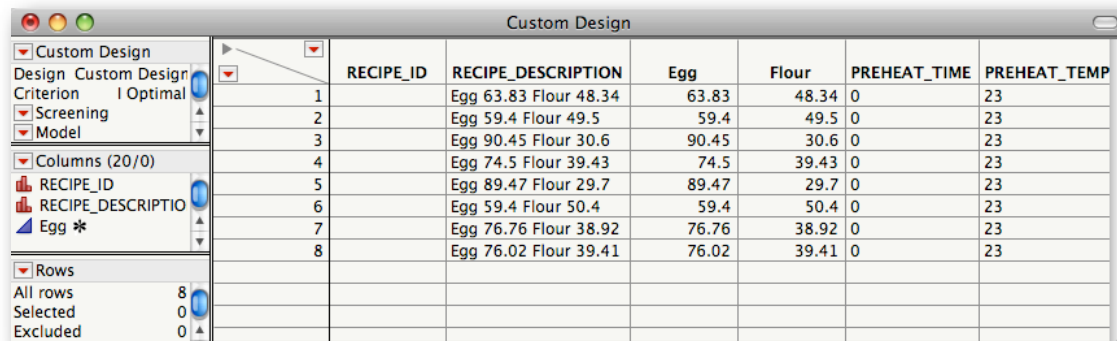
In this example, Salt is set at a constant level of 10 units it leaves 90 units for experimentation with Flour and Egg.

For the experimental factors of Egg and Flour the Active Low Level of one ingredient plus the Active High Level of the other (and vice versa) add up to 90 to allow JMP Custom Designer to find a valid starting point for design generation.

Egg levels have now to be increased to provide the appropriate amount of Active ingredient. This amount is calculated using Ingredient/Filler AMT Calculator Panel.

Finally, the fun begins when they click Build Design. At this point JMP will collect and forward the information in the GUI to JMP's Custom Designer, creating a design and re-

ording it in a data table. Some additional information is added to the data table to identify the study and combinations being run. In a future version of the virtual lab these studies will be recorded as well to, among other things, evaluate formulation experimentation activity.



	RECIPE_ID	RECIPE_DESCRIPTION	Egg	Flour	PREHEAT_TIME	PREHEAT_TEMP
1		Egg 63.83 Flour 48.34	63.83	48.34	0	23
2		Egg 59.4 Flour 49.5	59.4	49.5	0	23
3		Egg 90.45 Flour 30.6	90.45	30.6	0	23
4		Egg 74.5 Flour 39.43	74.5	39.43	0	23
5		Egg 89.47 Flour 29.7	89.47	29.7	0	23
6		Egg 59.4 Flour 50.4	59.4	50.4	0	23
7		Egg 76.76 Flour 38.92	76.76	38.92	0	23
8		Egg 76.02 Flour 39.41	76.02	39.41	0	23

Figure 9: The DOE as created by JMP's Custom Designer

The virtual lab is now ready to take the DOE, as shown in Figure 9, incorporate it into a pre-defined SAS model and submit it to SAS for evaluation.

Integration

At this point, the virtual lab has pulled the results that have been collected from decades of physical lab work and facilitated the design of a new experiment while informing the Formulator where models will ignore various recipe components. With the design now ready, it's time to execute the models. At this point, the JMP side of the virtual lab takes the Design Table, in JMP format, and turns it into a series of major SAS jobs (each could easily be thousands of lines long). It then submits the SAS code over the network to a SAS server for evaluation. The results are sent back from SAS to JMP in the form of a JMP data table containing response information. This table is joined, in JMP, with the original design table for JMP to profile graphically.

The next few lines, taken from various parts of the script, are where JMP & SAS communicate.

```

Connection_SAS = SAS Connect( "SASMain", prompt( ifNeeded
), show dialog( 1 ) );
Connection_SAS << Submit( String_SASCode, NoOutputWindow );
Connection_SAS << import data( "work", "ModelResults" );
    
```

The first line establishes a connection and names the resulting object "Connection_SAS". The SASCode, put together on the JMP-side of the virtual lab, is in the form of a very long string that is stored in a global variable called "String_SASCode." The second line Submits this long string to SAS. The third line instructs SAS to send down the re-sults in the form of a JMP table.

Visualizing the Results

Following SAS’s evaluations, the virtual lab returns a GUI allowing Formulators to analyze the responses visually using JMP’s Custom Profiler and Surface Profiler.

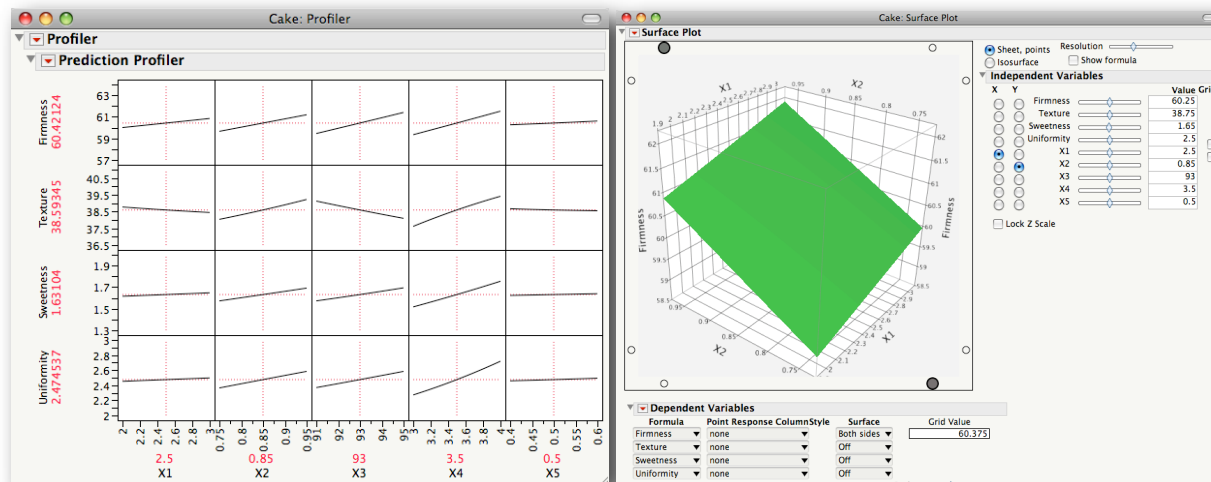


Figure 10: Visualizing results with JMP’s Profilers

The profilers are standard JMP fare allowing users to explore their experiments interactively. This capability is one of the primary reasons our client chose a JMP/SAS combination for the virtual lab.

Challenges

As with any major project, particularly involving new software releases, there were some challenges in creating the virtual lab. One of them involved JMP 8’s new Mixture Sum capabilities which are important because the formulations are mixtures. In JMP 8 Mixture sum is varied interactively as part of the advanced DOE options. It turned out that it was not scriptable in 8.0, though it was intended to be. So we had to work around the issue to hold the virtual lab over until the next maintenance release of JMP 8. The work-around involved staying anchored with a mixture sum of 1 by normalizing the levels in JSL and a corresponding fix to the constraint in the input data table for SAS.

This is one of several cases where JSL provided the flexibility and capability required to work way around just about any road-block, regardless of its source.

Advantages of JMP/SAS Combination

The JMP and SAS combination for the virtual lab offered a number of benefits

Cost/Benefit

The virtual lab’s construction cost a fraction of other solutions with a relatively short development time-line. After preliminary investigations and proof of concepts were completed, The virtual lab took about 10 months to complete. We could have delivered a lesser application

earlier but the client decided that they would rather add more functionality in the early releases and delay introduction to the Formulators.

This client was extending their investment in JMP and SAS since they were already using JMP and already had SAS available on their server (though they added new components to SAS). The virtual lab allowed our client to retire other software concentrating their analytics on JMP and SAS. Repeatedly our client has told us that we are well ahead of where they expected to be given their experience with competing products.

Low Maintenance

The virtual lab exploits JMP's powerful abstraction capabilities which substantially reduces the need to edit much of the code (JSL Scripts) that make up the virtual lab. For example, they expect to be adding new models for evaluation as well as swapping old models out for new, enhanced models. This activity can take place without changing a line in any of the JSL code. Also, as new formulations are added to their data base, the GUI automatically includes them.

Because of JSL's abstraction capabilities, carefully crafted scripts can respond to the information and resources provided. In areas that change regularly, such as models and database information, we have ensured that the virtual lab is self-maintaining.

Scalable

JMP scripts can be scaled and extended. The virtual lab will soon have a second, parallel function that models less demanding situations entirely in JMP. It will not be too difficult to make the necessary changes for the same code to work in both areas. Additionally, different research centers have different sets of model suites. The virtual lab, as is, can work with as many different sets as required - all from the same script.

Bottom Line

The physical lab is as busy as it ever was but the work they are receiving now has already undergone several revisions and tests, virtually, in the virtual lab. This increases the value of the work done in the Lab.

Time To Market and Tacit to Explicit Knowledge

As important as costs are, the bigger payoff actually comes from getting new and better products to market faster and the ability to harness insights stored in their database from physical Lab work over the years. The virtual lab has made tacit knowledge accessible and exploitable. New Formulators are able to get up to speed faster, incorporating the know-how that has been gleaned from thousands of studies over decades.

Future Developments

Development so far has been directed at putting together a workable application. We deliberately held back on refining the workflow for several reasons. First, it was apparent fairly early on that there would be challenges associated with JMP 8 that we would have to contend with. Second, we had to figure out along the way how best to distribute

and integrate JMP, SAS, the databases and several other key files. Third, no matter how well you design a GUI on paper, once it's implemented you can be sure that changes will be requested. So we thought it best to leave sharpening the GUI and refine the workflow until after a critical number of Formulators spent a good number of hours working with the virtual lab across a good number of projects.

Beyond the GUI there are a number of areas for enhancement including optimizing the 4000 lines of JSL code that make up the virtual lab to make it denser (more functionality with less script) and easier to document and support. We're also looking into capturing model formulas directly from SAS and implementing them in JMP instead of the current practice of fitting RSM models to the the predicted responses provided by SAS.

Contact Information

- Brian McFarlane: mcfarlane@predictum.com +44 (0) 141 339 5539
- Wayne Levin: levin@predictum.com 416.398.8900

Acknowledgments

We're very grateful to JMP Technical Support, Malcolm Moore and Ian Cox of SAS UK for useful assistance over the course of the project so far. Also, we are grateful to our client and their excellent team who have pursued the virtual lab's development with enthusiasm.

Erin Vang of Predictum is a major contributor to the virtual lab looking after several aspects including the integration of JMP with SAS, abstracting alert windows and optimizing the code.

Biographies

Wayne J. Levin

Wayne Levin is President of Predictum Inc. a company specializing in SAS® and JMP® application development in a variety of industries including semiconductor, consumer products, pharmaceuticals and automotive. Wayne was certified as an instructor with SAS Institute for JMP® in 1997 conducting dozens of courses in statistical methods every year since. He graduated from the University of Toronto in Industrial Engineering with a Master's degree in Engineering from the University of Waterloo.

Brian McFarlane

Brian McFarlane was educated at Glasgow University, Scotland where he received an Honors Degree in Physics and a Diploma in Education. He has more than 20 years experience in the Semiconductor Industry where he has worked in areas from Design through to Final Test. He has four US Patents to his credit for innovation in logic chip processing.

Brian joined Predictum Inc. as a Senior Developer in 2007 where he has worked on JSL scripting projects designed for both Academic and Manufacturing applications.