

Paper 115-2009

Optimizing Product Formulation with JMP® and SAS® Enterprise Miner™

John Leveille, d-Wise Technologies, Raleigh, NC

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

ABSTRACT

There are many products across a range of industries that rely on formulation and testing in a laboratory setting. In some cases, the formulation is guided by well-understood scientific principles. In other cases, historical process and empirical data show the way to effective product recipes and formulation techniques. This paper illustrates how JMP® and SAS® Enterprise Miner™ software can be used to create a black box model for empirical lab data and help to reduce the cost of trial-and-error product formulation. This paper is aimed at individuals who work in a product research or laboratory setting and are interested in applying data modeling software to identify the best formulations and improve laboratory or manufacturing efficiency.

INTRODUCTION

There is an interesting show on television called, "How it's Made." This program gives viewers an up-close look at the manufacturing process for all kinds of products including rubber gloves, candy coated chocolates, shopping carts, plastic kayaks, and more. The show does a wonderful job of detailing the process that accompanies many of these products. However, it never discusses the research it took to arrive at that formulation. How did they figure out that they need to freeze the product before coating it with chemical X? Are there alternative formulations which do not require chemical Y but achieve a similar result? Often, the product formulation team knows the answer to these questions. This is because they have spent large amounts of time and money on trial and error product formulation. Over time, this trial and error represents a significant cost burden to the company and fills up the testing lab with many wasteful trials that will, in the end, result in failures.

SAS Enterprise Miner is a powerful tool for modeling product formulation and, if applied correctly, allows the user to optimize research and laboratory functions. JMP is an excellent desktop tool for data visualization and ad hoc analysis. In addition, it has the capability to display custom user interfaces that leverage the capabilities of both JMP and the SAS server. This paper illustrates a system that was created using JMP to extend the power of SAS data models to the desktop for subject-matter experts within the R&D organization of a chemical company.

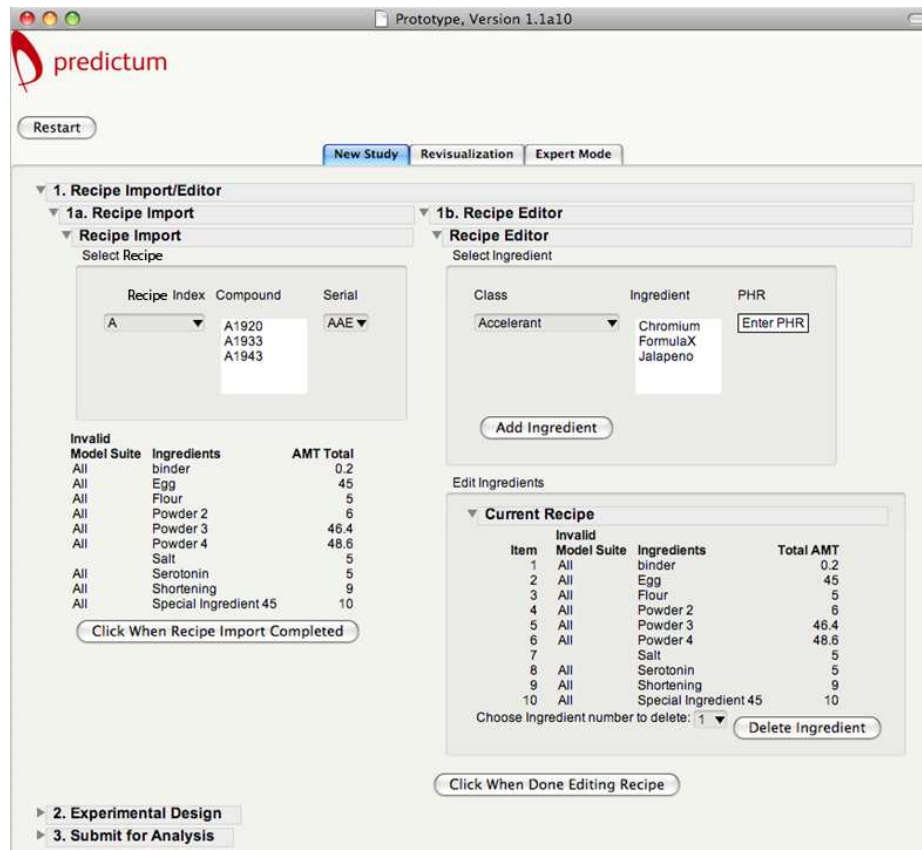
Due to confidentiality constraints, the examples and details that follow have been contrived using the application of building cake recipes instead of the proprietary business application in use by the client. The illustration will describe the roles of users: Data modelers were given the task of utilizing SAS to create a collection of models that could be leveraged within the system. Other researchers had the role of using JMP to formulate new cake recipes by executing these models. Finally, the paper shares a limited view of the system functionality, again, to maintain client confidentiality.

A JMP INTERFACE FOR RECIPE CREATION

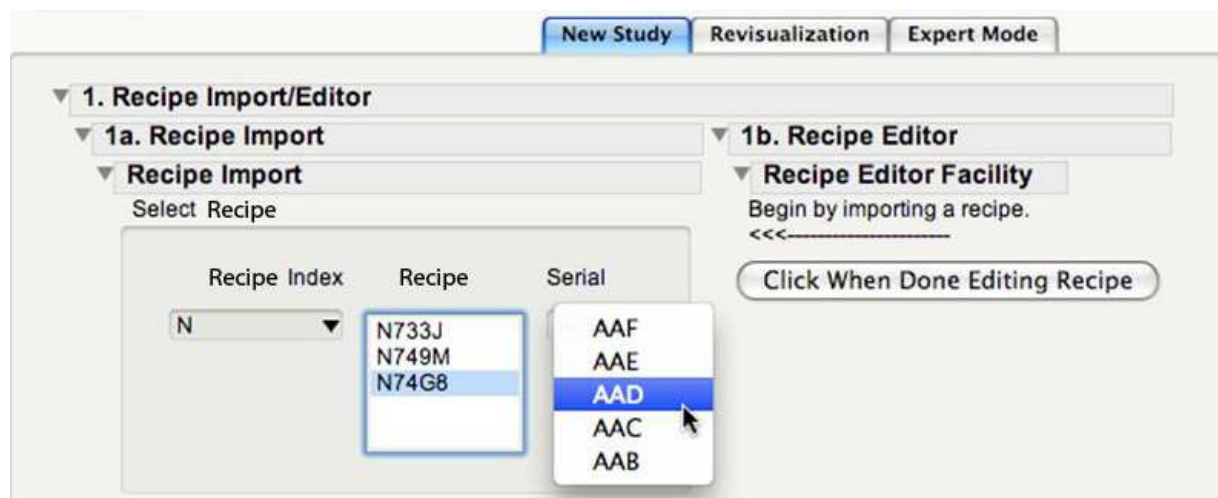
Management at the fictitious company *Cake Bakers Inc.* has identified a problem. The company is spending too much time and money experimenting with cake recipes. They have determined that they can cut down on the expense of their kitchen laboratory by creating computer models of the cake baking process. On one hand, there are only a few people within the company who have the expertise in building computer models for cake recipes. On the other hand, there are many individuals who are responsible for developing new cake recipes. The solution is to develop a user interface in JMP that is customized for cake recipe formulation. This application interface will allow the recipe developers to invoke SAS-based models developed by the cake recipe modelers.

JMP resides on the user's workstation and assists in designing the experiment. The choices made by the recipe developer are used by the JMP application to generate SAS code. JMP creating SAS code? Yes, via JMP Scripting Language (JSL). JMP was initially designed as a point-and-click environment but as of version 4 in 2000, the ability to program JMP was added. JSL is not SAS or even SAS-like. It is a powerful object-oriented language with infinite abstraction abilities. You can write a script that writes a script that writes a script to as many levels as you wish before it's evaluated. In addition to JSL's ability to write JSL, it can also generate SAS code. JSL scripts can be very flexible, allowing them to respond to their current situation, adding functionality and reducing maintenance requirements at the same time.

In this example JMP queries a collection of database tables and uses the results to populate the user interface. As new recipes and ingredients are added to the database, the JMP interface automatically adapts and displays updated selection lists and parameters.



In addition, JSL and database queries are combined to create cascading pick lists. For example, when Recipe N74G8 is chosen the Serial drop down is immediately updated, as well as the recipe below it. From here, the recipe developer can inspect the recipe and import the full detail into the JMP application.



Next, the interface provides a set of controls to edit the chosen recipe.

1b. Recipe Editor

Recipe Editor
Select Ingredient

Class	Ingredient	PHR
Accelerant	Chromium FormulaX Jalapeno	<input type="text" value="Enter PHR"/>

Edit Ingredients

Current Recipe

Item	Invalid	Model Suite	Ingredients	Total AMT
1	All	binder		0.2

Once the recipe developer is done editing the recipe they can move on to the Experimental Design. Here we can only show you the Informal approach where specific recipes are constructed. The production version takes advantage of JMP's Custom Design Platform. The screen shown here allows the developer to choose a model for cake mixing and baking. In addition, they can specify some of the parameters such as time to age the mixture, baking temperature, etc.

Prototype, Version 1.1a10

2. Experimental Design

Design Controls

Design Editor

Model Suite

Tick at least one check box

AAA
 BBB
 CCC

Test Conditions

Aging	Baking
150min@30C	5days@100C
18min@150C	6min@170C
MOD2 Temp	55min@165C

Informal Design Edit Panel

Ingredients	PHR	RHC
binder	<input type="text" value="0.2"/>	0
Egg	<input type="text" value="40"/>	0
Flour	<input type="text" value="6"/>	0
Powder 2	<input type="text" value="11"/>	0
Powder 3	<input type="text" value="34"/>	0

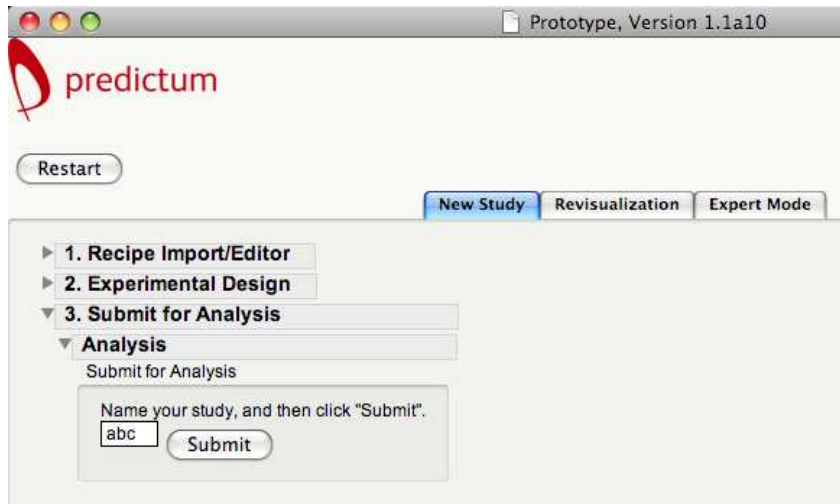
Saved Recipes

Factors	a	b	c
binder	0.2	0.2	0.2
Egg	45	45	40
Flour	5	5	6
Powder 2	6	12	11
Powder 3	46.4	34	34
Powder 4	48.6	44	44
Salt	5	2	2
Serotonin	5	5	5
Shortening	9	9	9
Special Ingredient 45	10	10	10
Jalapeno	12	12	12
AMT Total	192.2	178.2	173.2
MAIN_AGING_TIME	.	.	.
MAIN_AGING_TEMP	500	500	505
MAIN_BAKING_TIME	0	5	5
MAIN_BAKING_TEMP	23	100	100
MMM_TEMP	150	150	150
MOD1_TEMP	95	95	95
MOD2_TEMP	0	0	100

Delete Recipe

Select recipe to be deleted: a

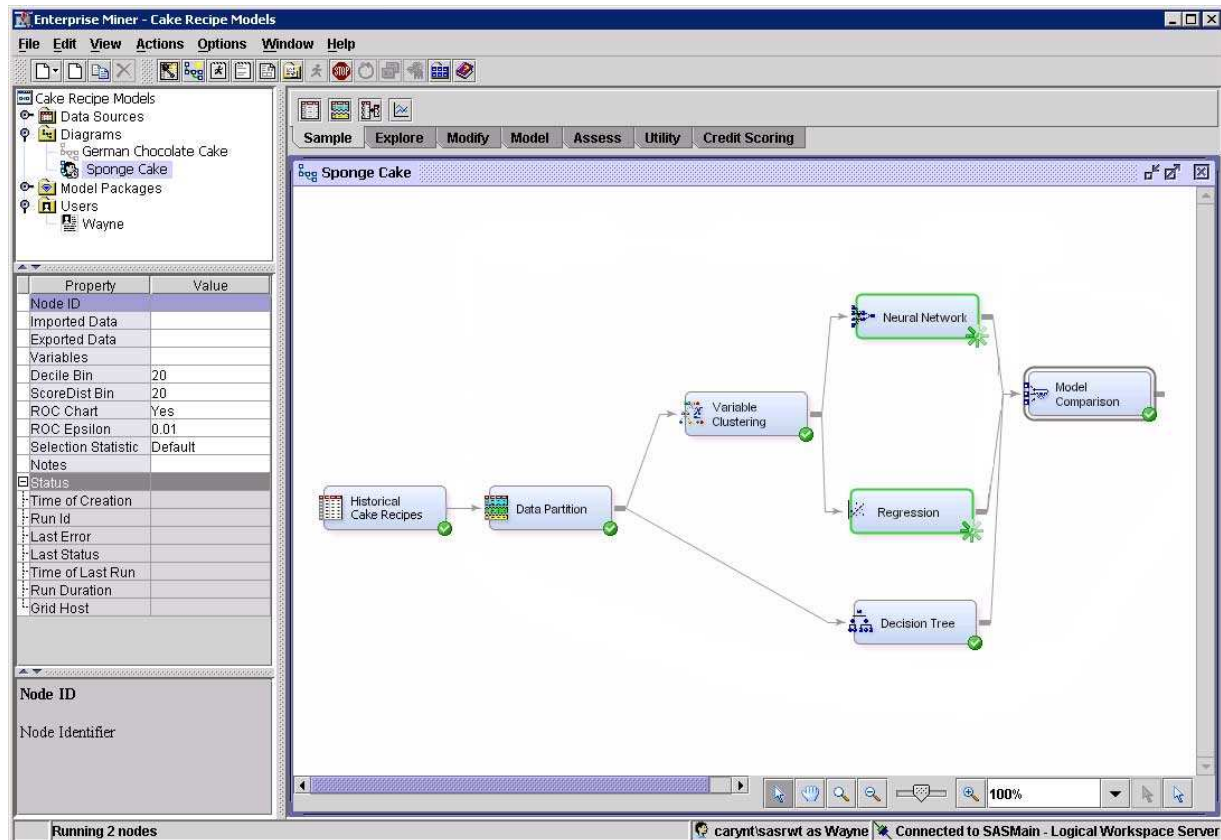
Once the user has completed the recipe customization, JMP facilitates the design of an experiment using its Optimal design capabilities and generates SAS code, submitting the whole package to the SAS server for model execution.



SAS Data Modeling: The Black Box on the Server

Each recipe experiment initiated by a user in JMP executes a model created in SAS. The recipe developer leverages the work product of the model builder. In addition, the developer is able to treat this complex software component like a black box. He/she doesn't need to understand the inner workings of the model. The developer can simply concentrate on the input and outputs of the experiment.

Before the recipe developer begins his/her work, the model builder must construct and train the models. This involves collecting a cleaning a sufficient amount of historical data, as well as carefully choosing an appropriate model for the data. SAS Enterprise Miner is the model builder's workbench. It supports a variety of models including regression, neural networks, decision trees, and more.



Suppose that the model builder selects a neural network to model the formulation of sponge cake. Using a new SAS Enterprise Miner project diagram, the model builder drops in a neural network node. Next, he/she sets the properties

for the neural network model and attaches other nodes to guide the flow of data as input to the model. Once all of the settings are made, the model builder trains the model by running the process in SAS Enterprise Miner. The resulting model is a series of algebraic equations implemented as a verbose SAS data step program. Here is just a portion of the resulting code

```

*** *****;
*** Writing the Node H1 ;
*** *****;
IF _DM_BAD EQ 0 THEN DO;
  H101 = 0.06548431518846 * S_BINDER + 0.01689895447745 *
        S_BAKING_TEMP + 0.26554413258402 * S_AGING_TIME
        + -0.01650255447580 * S_EGG + -0.15468434771002 *

```

EXERCISE THE MODEL & VISUALIZE THE MODEL

The model builders at Cake Bakers Inc defined a convention to save the SAS program resulting from model building activities to a known network folder location. The JMP application knows how to locate this folder and include the model code. Extending the analysis with new models or revising existing models simply involves adding or changing the SAS files in this folder. Each time the script is run, it's using the latest models available.

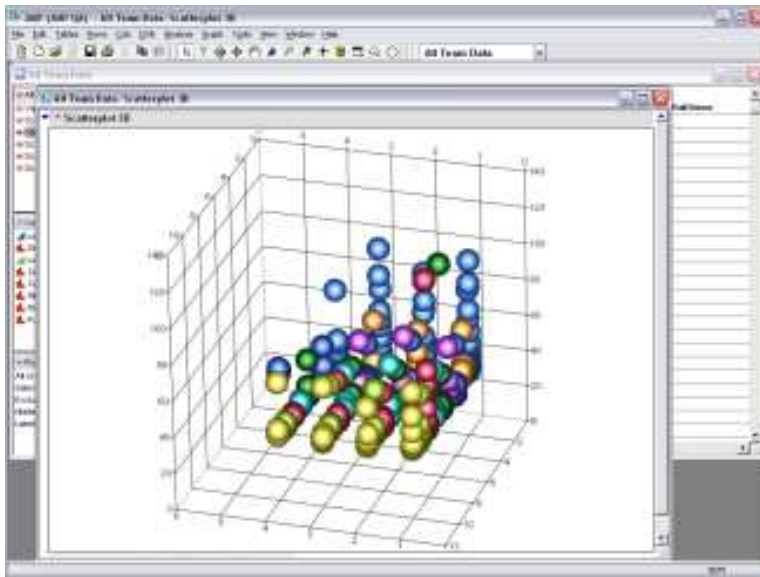
When the user of the JMP application presses the submit button, the JSL script connects to the SAS server and runs the model code. After the model code finishes running the JMP application imports the data results from the SAS work library. This is done using the following JSL code.

```

sasconn = SAS Connect( "SASMain" );
sasconn << Submit( String_SASCode, NoOutputWindow);
sasconn << import data( "work", "ModelResults" );
dt_SASresults = Current Data Table();

```

The first line of code makes the connection. The second line sends the SAS code to the server via the connection. The third line retrieves the results from SAS as a data table and the fourth line gives this returned data table an object reference name so that JSL can proceed. From this point, JMP can take any number of actions to help the user visualize the results of the cake recipe experiment. Often times, a 3-D scatter plot is used to visualize a result space and determine how close the recipe is to the desired outcome.

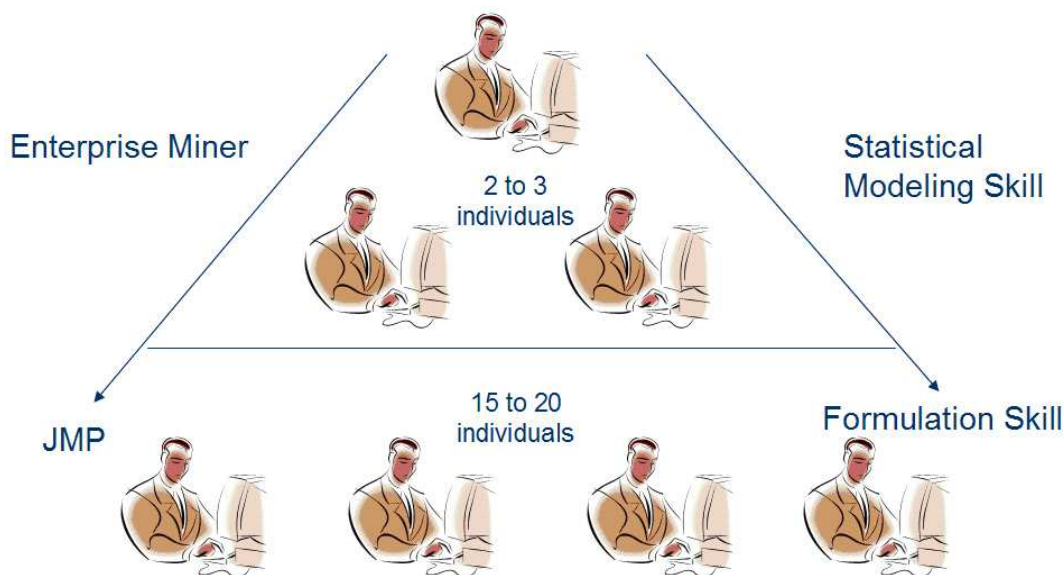


The example graph displayed here is only one of a very many visualization and analysis tools available to the user in JMP.

MODELING PLATFORM: DEPLOY TO THE ENTERPRISE

In many organizations there will be only a select few individuals with the expertise to build and train models, but there may be many more interested in exploiting the models. JMP can answer this need by providing users with a point and click analytical tool that can run the models and visualize the results. JMP can request the SAS server to perform a series of computations and return a package of results. This is often an optimal architecture because the

SAS server is probably close to the data and those data sets may be very large. The result is an ideal combination of analytical power on the desktop and optimized analytical power on the server, close to centralized data.



This solution is not only powerful and efficient from a computer processing perspective; it is also efficient from an organizational perspective. Organizations that deploy a modeling platform will see immediate benefit as personnel are directed toward their optimal skill set and high-value work tasks. In addition, the organization will reap benefits over time by centralizing and standardizing core work tasks and work product. These benefits are in addition to the obvious time and cost savings of modeling formulations without physical laboratory experimentation.

CONCLUSION

JMP and SAS are a powerful combination; two products that play to each other's strengths. JMP provides visualization and point-and-click usability while SAS provides heavy-duty analytical capabilities and data manipulation that well exceed JMP's limits. These tools can be paired to create an intuitive modeling platform for product formulation – a "virtual laboratory". Deploying a virtual laboratory will result in a substantial cost savings and organizational improvements for manufacturing R&D organizations across all industries.

REFERENCES

SAS Institute, Inc. SAS Product Documentation. "SAS Enterprise Miner."
<http://support.sas.com/documentation/onlinedoc/miner/index.html>

SAS Institute, Inc. JMP 8.0 Product Documentation. "JMP Scripting Guide."
http://support.sas.com/documentation/onlinedoc/jmp/scriptguide_11146.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

John Leveille
 d-Wise Technologies, Inc.
 3115 Belspring Ln
 Raleigh, NC 27612
 Tel: 888-563-0931 x101
jleveille@d-wise.com
<http://www.d-wise.com>

Wayne J. Levin
 Predictum, Inc.
 #506 - 592 Sheppard Avenue West
 Toronto, ON M3H 6A7 Canada
 Tel: 416 398 8900
levin@predictum.com
<http://www.predictum.com>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.