



From Data to Decision: How SAS® Decision Manager Automates Operational Decisions

Contents

Introduction to Enterprise Decision Management....	1
What Are Decisions?	1
Benefits of a Decision Management Approach.....	2
Decision Management: An Example	3
SAS® Decision Manager: Feature Overview	3
Operational Data Preparation	3
Data Summaries and Exploration	4
Managing Models	5
Predictive Models in Decision Making	5
Building Predictive Models.....	6
Managing Models	6
Model Testing	7
Model Publishing.....	8
Model Monitoring	8
Models and Decisions	8
Managing Business Rules	8
Defining and Testing the Business Rules.....	8
The Business Vocabulary.....	9
Rules, Rule Sets, and Rule Flows.....	10
Building Business Rule Sets.....	10
Creating Business Rule Flows	12
Analytically Driven, Discovered Rules	12
Testing Rules for Validity.....	12
Rule-Fired Analysis	13
Publishing Rule Flows	13
Integrating the Decision Components.....	14
Deploying Automated Decisions	14
Deployment of Decision Components.....	15
Conclusion.....	15

Organizations make decisions every day that affect different parts of their operations. From dispatching a service team to acquiring a new customer, these operational decisions – while tactical in nature – outnumber, and often overshadow, strategic decisions. And if these operational decisions are incorrect (or inconsistent with other decisions), the organization can be exposed to risk, compliance violations or inhibited growth.

Today, organizations can modernize their operational decision environment, automating these repeated decisions while ensuring consistency, accuracy and rapid execution of desired results. When decisions are automated, the elements needed to make a decision become part of a decision flow. The best decision outcome for a specific scenario is based on operational data and tested by a business analyst, who deploys the decision model into production.

Once the model is in production, you can monitor decision results to validate the desired outcomes. These decision models then run automatically, for any area of the business. This includes marketing offers, sales transaction processing, payment processing, call center interactions, industrial maintenance, transportation scheduling, and thousands of other applications – all of which have a significant impact on the business bottom line.

Advanced analytic models help you assess current scenarios and examine the likelihood of future events. Such models provide valuable insight into customers, products and markets. Yet, many decision systems cannot maximize the power of analytics in the operational business processes at the point where decisions are made. And those that do are often outside of a single decision scenario definition – often requiring special handling, coding and manual integration.

For organizations that rely on analytical models in the decision process, SAS® Decision Manager provides a single point of control to integrate sophisticated analytical models with your business rules. It also helps simplify deployment of both models and rules into the operational systems where these decisions are made. In a web-based environment, analysts define, test,

and deploy models and rules, then monitor, assess and improve them over time. This removes the risk associated with inconsistent decisions. Now, operational decisions are governed, traceable and fully documented.

This paper describes how SAS Decision Manager works, explaining how the benefits of automated decisions can help organizations by modernizing their operational decision environments.

Introduction to Enterprise Decision Management

What Are Decisions?

Most companies start out with the idea that they want to implement a business strategy driven by a market need, such as reducing the number of delinquent loans or improving the quality of customers approved for credit cards. To do that, you need to bring together various elements that typically constitute a decision. These elements include:

- **Operational data.** Manage and prepare operational inputs to better understand what has happened and use this information to formulate new strategies and implement better decision making.
- **Analytical modeling.** Prepare and monitor analytical models to better understand why events and certain behaviors occur.
- **Business rules.** Define and discover rules that determine what action needs to be taken based on likely future behaviors, allowing you to react quickly to situations detected in the marketplace.
- **Optimization.** Use the information gained from past decisions to improve how decisions are made in the future.
- **Process integration.** Deliver the decisions directly into the business process where decisions are made, reducing the lag between event detection and taking action.

Benefits of a Decision Management Approach

There are many challenges to delivering a comprehensive and integrated decision making environment – and often these challenges prevent an uninterrupted flow of information across the different phases and roles within the decision.

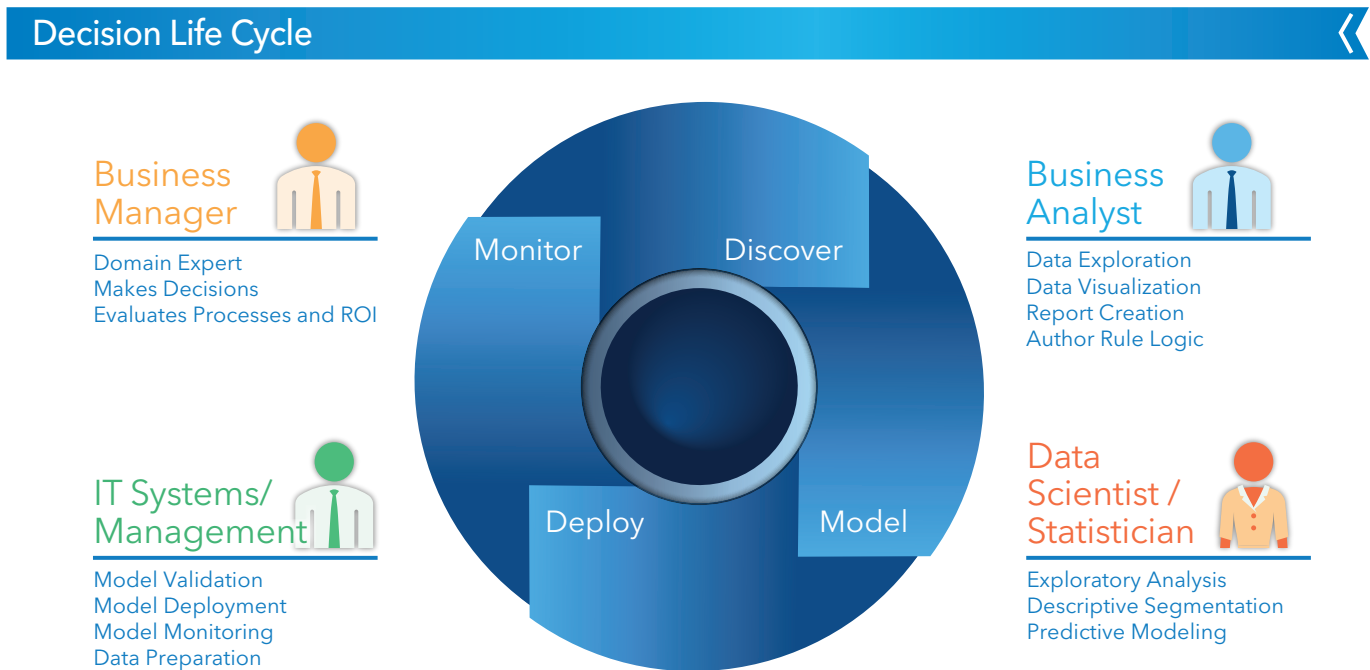


Figure 1: Decision life cycle.

Roles and activities within modern businesses are not always implemented as smoothly as Figure 1 suggests. A typical decision life cycle contains numerous handoffs by overwhelmed and understaffed teams in an environment of dynamic regulations, policies and changing market conditions. All of this can lead to a less than optimal decision-making process.

However, a more deliberate approach can yield a variety of benefits – especially for companies in more dynamic and competitive markets – including:

- **Fewer silos separating IT and business.** Allow business users to create or access models needed to support decision-making strategies in a collaborative environment with the statisticians and data scientists who are developing the models.
- **Faster decision deployment times.** Eliminate the need for custom coding to generate business logic and simplify integration of analytical model output with business processes.
- **Treat decision components as corporate assets.** Provide governance and version control for rules and models as well as automate deployment steps for greater auditability and repeatability.
- **Move analytics closer to where decisions are made.** Take action based on model output within business processes while avoiding deployments that are slow and disconnected.
- **Create business agility.** Allow business users to design and change business rules and react quickly to changing regulations and policies. Business analysts know the business processes best, and because no programming is required, all code is generated by the system from interactive rule discovery and definition.

Decision Management: An Example

This paper presents a scenario for using SAS Decision Manager to obtain the benefits from an integrated decision life cycle. For this example, here are some basic facts about the company and its decision management problems.

Company	Large, national reseller of used vehicles.
Data being analyzed	Past sales and return-related data.
Problem	An increase in returned cars negatively affected profits.
Challenges	<ul style="list-style-type: none"> • Identify relationships between purchased cars. • Identify attributes associated with vehicles that have been returned.
Solution	Add decision management aspects to existing systems to help those purchasing cars make better decisions about which cars to purchase – and when.

This company decided to investigate how it was making the decision to purchase vehicles at the start of the process to find ways to improve its decisions about which vehicles to purchase. The company anticipated that it could purchase fewer used vehicles and decrease the return rate. This would lead to higher profits through reduced inventory costs, lower repair costs, lower reimbursement costs, and happier (and hopefully repeat) customers.

The process followed the following steps:

- Review the data to understand the attributes of the used vehicles, including the types of vehicles returned.
- Develop predictive models that could help determine the probability a vehicle will be returned.
- Augment the model outcomes with business rules driven by corporate policies and guidelines.
- Deploy these vehicle purchasing decisions to business processes to help buyers make better decisions about purchases.

SAS® Decision Manager: Feature Overview

SAS Decision Manager supports collaboration between business analysts, statisticians, IT staff and business managers by unifying key elements of the decision life cycle. The software centralizes control of decision components, documenting the decision process in a single user interface. That interface includes a variety of activities, such as:

- **Operational data.** Include and manage the data table inputs in SAS libraries, defined to SAS metadata.
- **Business rules.** Author, test and publish business rules.
- **My tasks.** Automated workflow for both business rules and model projects to automate authorization activities, deliver notifications and provide tracking and auditability for increased governance.
- **Models.** Import new models using shared SAS metadata and local files, retrain models, monitor model performance, and publish models to integrate with business processes.
- **Workflows.** Workflow definition support of activities such as reviews and approvals to support repeatability and auditability.

Operational Data Preparation

Like most business decisions, the auto reseller's process starts with gaining a strong understanding of the data. To determine what decisions need improvement, the company started by understanding what attributes were available from past sales and vehicles returns. The operational data preparation capabilities in SAS Decision Manager include:

- **Data access.** IT built the tables for this process. Analysts using SAS Decision Manager can select these trusted tables from SAS libraries (listed in SAS metadata), which minimizes custom data pulls and the overall strain on IT resources. Analysts can directly review the auto data sales in the software.
- **Data summarization.** Sales data summaries show the different counts for vehicle type, color, size, transmission types and sales price.
- **Base table creation.** Using query building, column selection, and SAS Visual Analytics integration, analysts can create tables that combine vehicle sales data along with geographic data about where these vehicles were purchased.
- **Data table and library management.** SAS Decision Manager allows business analysts to locate tables and create libraries for their specific projects.

The data in this scenario contains information about the vehicles and their prices, as well as some statistical data. See Figure 2 for a summary of the table metadata.

Figure 2: Vehicle table metadata.

Data Summaries and Exploration

Data summaries, seen in Figure 3, provide additional statistical information that can provide greater insights into the data. Numeric and character summarizations provide information about maximum, minimum, mean, average, percentiles, skew, kurtosis, and missing values within the data.

Because data is always changing, business analysts can create data summaries that change as the data changes, allowing them to visualize these changes over time and identify patterns in the data that might be useful in the decision definitions.

After analyzing vehicle data and preparing the base tables from the operational data, the business analysts would now like to understand how they can predict which vehicles have a high probability of being returned. By collaborating with the statisticians/modelers/data scientists, business analysts can use this data to build analytical models that can improve their ability to predict what vehicles have a higher probability of being returned.

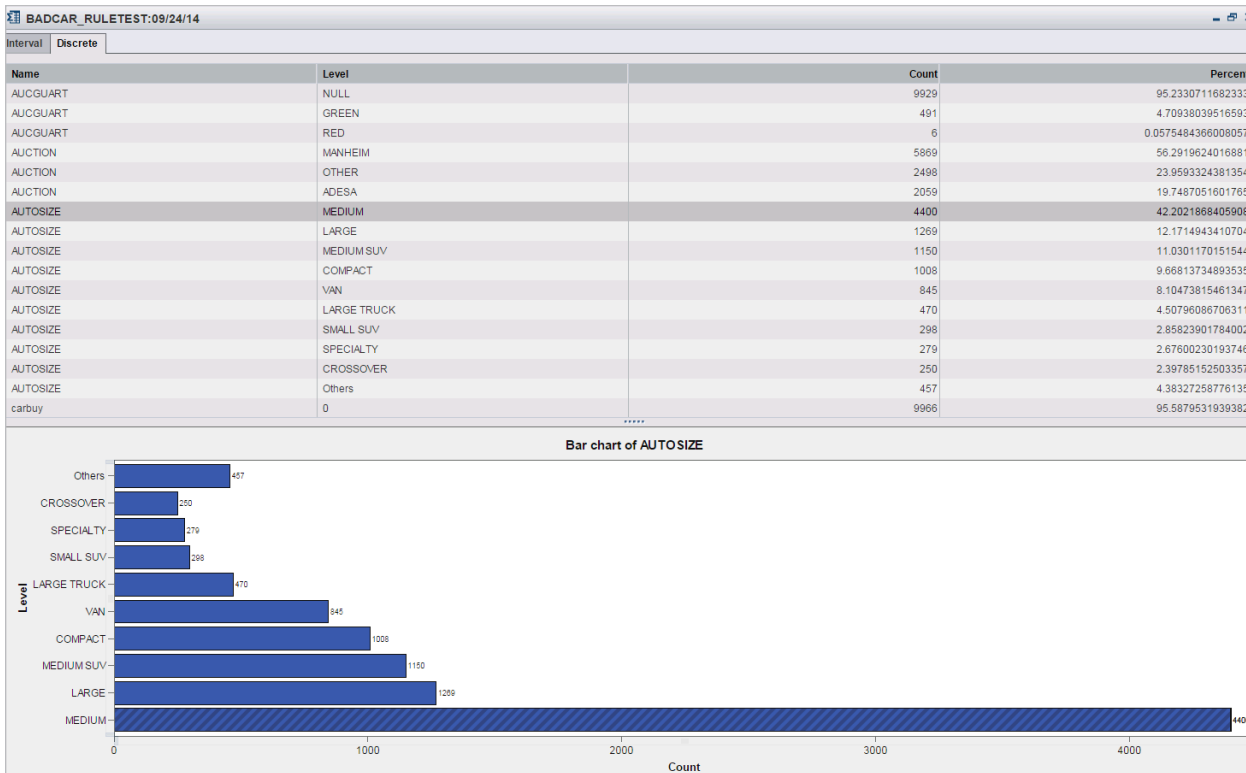


Figure 3: Data summarization example.

Managing Models

Predictive Models in Decision Making

Predictive modeling is the most essential analytical step of the automated decision-making process. Predictive models are often more accurate than expert systems¹ and can detect patterns that human recognition will miss. Models are trained on historical data that contains both input variables and target variables. Inputs are conditions that can be used to make predictions; targets are the measurements that need to be predicted.

In this case, the input variables represent the information that was available to the purchasing dealership at the time of the used car auction. The data label is a new variable named *isBadBuy* that indicates that the car was later returned by the customer to the dealership as a lemon – a bad car.

The company is seeing that about 12 percent of the time, vehicles are being returned. That is a high percentage of returned vehicles that the auto dealerships want to avoid, so it's important to build a model that can predict which cars will be returned to the dealership as bad buys. The company will use that prediction along with other information about strategy and policies in the decision flow process.

There are two actions needed for deploying a predictive model into a decision-making process.

- 1) Model training.** This runs a statistical or data mining procedure that will build a model to predict *isBadBuy* based on a set of input variables. Most users employ either SAS/STAT® or SAS® Enterprise Miner™ for this purpose. The result is a model scoring expression such as *model_score_code.sas*, a program represented in SAS code. For each row of data, the model will output a probability that *isBadBuy* is true.
- 2) Model scoring.** This runs the model scoring code with new data to produce new scores. Analysts at the auto company trained the model using historical data labeled with *isBadBuy*. The model scoring process adds a prediction of *isBadBuy* to each record of data that is scored. For SAS programmers, you can conceptually represent this as a simple DATA step.

```
data newscores; set newdata; %include
"model_score_code.sas"; run ;
```

¹ Expert systems are computing systems that emulate the human decision-making process. Please refer to http://en.wikipedia.org/wiki/Expert_system for more information.

Building Predictive Models

SAS provides several tools for building predictive models. SAS/STAT and SAS Enterprise Miner can construct and compare accurate and robust statistical and machine learning models. “Accurate” indicates models that successfully detect patterns in the training data that produce the minimum error rate in their predictions. “Robust” means that the model’s predictions will be reliable on data that is similar to the training data. Data scientists or statisticians can create several models with different attributes as candidates for a new decision making project.

In this scenario, the company created two candidate models using SAS Enterprise Miner and saved each model as a model package file. This file includes all possible details about training and scoring this model. SAS Decision Manager imports these models for use in a decision flow.

Managing Models

Since predictive models are an important asset to the business, it’s important to manage them carefully. Models should be used only in situations that match the scenario in which the training data was collected. There are frequently several candidate models that have different performance attributes. Some models perform better or work for different customer segments, time periods or economic conditions. Selecting the champion and challenger models from the set of candidate models is often a process governed by internal and external regulatory concerns, and it can have a large impact on business profit.

SAS Decision Manager includes SAS Model Manager, the preferred software for managing collections of statistical and data mining models. It can import models from SAS/STAT, SAS Enterprise Miner, and the PMML (Predictive Model Markup Language) standard, as well as models created in the R language. SAS Model Manager provides functions for:

- Testing model scoring functions.
- Computing performance metrics that can be used to select models.
- Monitoring model performance in production systems.

All of this is managed within SAS Decision Manager, the framework that provides automation, workflow planning and tracking, and model governance using versions, comments and attachments.

Back to the scenario. The company imported both of the *isBadBuy* models created with SAS Enterprise Miner into a model project named *AuctionPurchaseBadCar*. Comparing the models, you can now see that the Decision Tree-based model (on the right) uses many fewer inputs than the Logistic-based model (on the left) – shown in Figure 4. Since the project will be deployed in a real-time system, the company prefers the model with fewer input variables because that will make the real-time operational data payload input smaller and simplify the data needs for deployment. The analyst marks the Decision Tree model as the champion model to be used in the decision flow.

Auto: badcar_regression					
	Name	Type	Measurement	Length	Description
Input					
Output					
Output Mapping					
Properties					
General					
Specific					
System					
User-Defined					
Scoring					
Status					
Advanced					
Score Code					
Model Files					
History					
Log					
Published					

Auto: badcar_tree					
	Name	Type	Measurement	Length	Description
Input					
Output					
Output Mapping					
Properties					
General					
Specific					
System					
User-Defined					
Scoring					
Status					
Advanced					
Score Code					
Model Files					
History					
Log					
Published					

Figure 4: The model project showing both the decision tree and logistic models.

Model Testing

After selecting the model, the next phase is to test the scoring function using new data to validate how well the model performs outside the development environment. The model project provides score testing functions. You can set up multiple tests with different data sets representing conditions that may be found in the production environment, using operational data inputs.

Auto						
Properties Versions Models Variables Scoring Performance Retrain Reports History Attachments Comments						
<div> <div>+</div> <div>+</div> <div>+</div> <div>+</div> <div>+</div> <div>+</div> </div> <div>Version:</div>						
Tests	Results	Schedule	Job History			
Status	Name	Model Name	Date Created	Date Completed	Created By	
✓	scoreAuto	badcar_tree - Champion	Aug 13, 2014 04:56 PM	Aug 26, 2014 12:23 PM	stspar	
✓	Q2ScoringVersion	badcar_regression - Challenger	Sep 24, 2014 05:56 PM	Sep 24, 2014 05:56 PM	stspar	

Figure 5: Multiple test cases can be run interactively or scheduled.

Once the tests have run in SAS Decision Manager, analysts examined the scores as depicted in Figure 6, using the data summary function found in the data table category. For this project, the company tested both the logistic and the tree models on the current operational data and found them both acceptable, so they can continue to use the Decision Tree model as the champion.

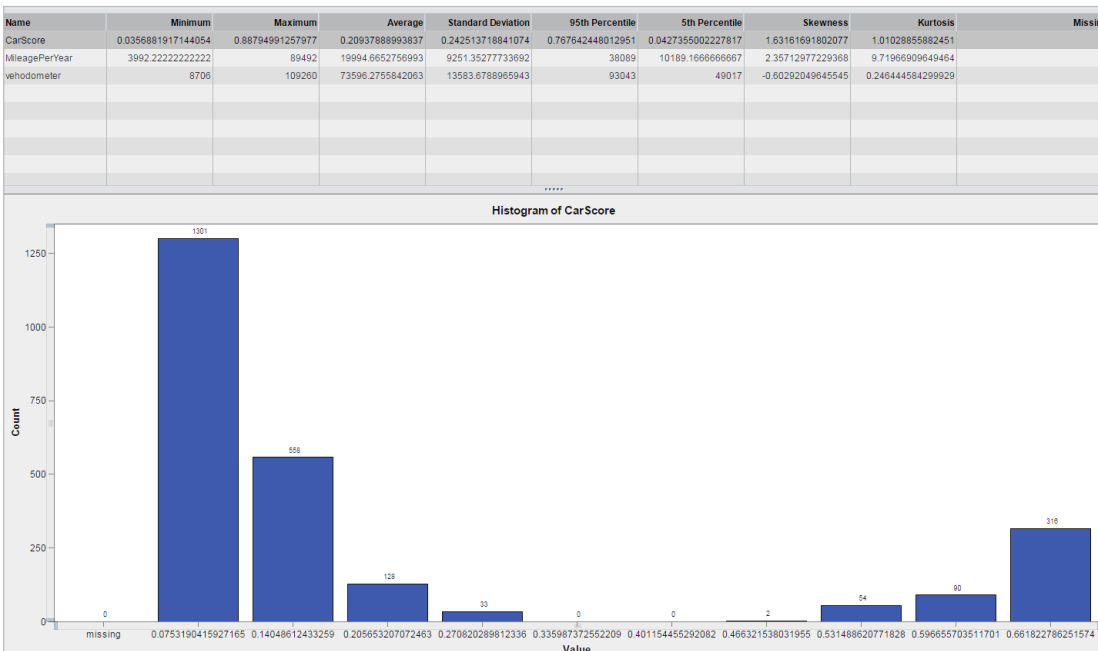


Figure 6: Distributions of model scores.

Model Publishing

Once the analysts choose the Decision Tree model, they publish the model so it can be used in the decision flow. Once published, the models (along with the associated business rules) are deployed as part of a SAS Decision Manager executable decision flow.

Before that, it's important to address analytical models – and how these are continuously managed.

Model Monitoring

Once the model is in a production decision flow, the accumulated results are collected in the ongoing operational data. These results must be monitored to make sure the model continues to produce desired predictions. There is typically a time lag between model deployment (putting it into production) and model monitoring (post-production assessment). In this scenario, the auto decision process deploys in January with cars purchased and resold in January, February and March. During that time, analysts can monitor the scoring processes for an “algorithmic drift” – the change in the distribution of either the operational source data inputs or the scores generated by the model.

For example, analysts find that a new set of cars has different characteristics than the cars originally modeled. For instance, if one line of cars proves to be less reliable than previously measured, then the model the analyst selected may not perform well. This can lead to a change in the distribution of scores and will ultimately affect the decisions made about purchasing cars. Earlier, the original data contained 12 percent of records considered to be “bad buys,” but the current data predicts that 50 percent of cars will be bad buys. This variance (between the initial data used to develop the model and the results being observed under current conditions) will lead the company to consider rebuilding the models to adapt to the new conditions.

SAS Decision Manager includes processes that will monitor the health of models during their production processes. The software can alert, or automatically switch to a challenger model, when model performance is below desired preset thresholds.

Models and Decisions

At this point, the auto company has a model that will predict the probability that a car will be returned. However, the statistical model is only one part of the decision making process.

The model initially used historical data representing conditions that were present when data was collected. Conditions have likely changed since that time. In addition, changes in business policy may lead purchasers to prefer one type of car over another due to evolving customer preferences and available inventory. At this point, it's time to build business rules to produce a valid decision process.

Managing Business Rules

Once the business analysts have the model defined by the data scientists/statisticians, they can apply business policies to the model output to further fine-tune the predictive nature of the models. Business analysts view business rules as a way to extend the decision-making power of a model and as a means to ensure compliance with internal policies or regulatory requirements. It's important to do this without retraining the models or having another group adjust the models. The business rules can be modified and adjusted easily with SAS Decision Manager, adapting to changing business requirements in a fully traceable and explicit centralized system.

Defining and Testing the Business Rules

Business rules represent the business logic that drives decisions in companies. These business rules are often found in government policies, regulations, tax laws, financial guidelines, and federal and local laws. These rules change frequently (for example, marketing and customer relationship applications), and rules must be modified, tested and deployed quickly. Each of these updates should be tracked for reporting and auditing purposes.

In this scenario, the vehicle dealer application requires that business rules override the output of the models developed by the data scientists/statisticians. Further, these business rules determine the action taken when they are evaluated to be true.

Based on the model output, the business must evaluate the probability of the vehicle being a bad purchase in conjunction with other business policies. These could be things that senior management would like to implement related to the vehicle's attributes, including:

- **Vehicle cost.** Avoid purchasing vehicles above a certain price.
- **Color.** Avoid vehicles of a specific color.
- **Make.** Avoid some makes due to a lack of available parts.
- **Wheel type.** Avoid vehicles that have a specific type of wheel.

- **Vehicle VIN state.** Account for the state where the vehicle was purchased.
- **Probability.** Apply the probability of vehicle being a bad purchase decision. This is available as output from the model.

These attributes are a means to override the probability that a vehicle is a bad vehicle. So, even if the model says that the vehicle is a bad purchase, the business would still want to override the model results and reset the bad car classification to a good car classification (or vice versa).

The Business Vocabulary

Business vocabularies allow business analysts to define the terms (or variables) used when creating business rule logic. The vocabularies consist of two primary components: entities and terms. Entities are higher-order components, and each entity has one or more terms associated with it. Collectively, a set of entities and terms represents a vocabulary.

Vocabularies can contain many terms, so it's important to be efficient in creating the methods to create and manage them. SAS Decision Manager analysts have the following approaches that support the creation of a vocabulary:

- User-entered. Manual method to define vocabularies, terms and entities.
- Import and export macros. Macros that support importing and exporting vocabularies, terms and entities from an existing table registered in SAS metadata.
- Analytically discovered rules. Driven by analytical algorithms applied to the operational data, which calculate or discover the terms and entities that form the vocabulary.

Based on the operational input data on the vehicles, the business analysts develop a vocabulary of automobile terms for use in business rule definitions. They also catalog conditions that govern the decision process and the corresponding actions taken after automating the entire decision process.

Rather than manually entering these terms and entities, the discovery approach can expedite the process. Business analysts can take advantage of the data elements contained within the same data set used for testing their rules. Figure 7 is an example of vehicle data terms imported into a vocabulary that can be used to author rules.

The screenshot displays the SAS Decision Manager interface. The top window, titled 'BADCAR_RULETEST', shows a data table with columns: REFID, PURCHASEDATE, AUCTION, VEHYEAR, VEHICLE, MAKE, MODEL, TRIM, SUBMODEL, COLOR, TRANSMISSION, WHEELTYPE, and WH. The table contains 10 rows of vehicle data. Below this, a 'Business Rules: Vocabularies' window is open, showing a list of terms and entities. A red arrow points from the 'WHEELTYPE' column in the data table to the 'WHEELTYPE' term in the vocabulary list.

REFID	PURCHASEDATE	AUCTION	VEHYEAR	VEHICLE	MAKE	MODEL	TRIM	SUBMODEL	COLOR	TRANSMISSION	WHEELTYPE	WH
137	01MAR2010:00..	ADESA	2003	FORD	TAURUS	SE	4D SEDAN SE	BLUE	AUTO	1	Alloy	
144	01MAR2010:00..	ADESA	2001	FORD	EXPEDITION 2.	Edd	4D SUV 5.4L E.	RED	AUTO	1	Alloy	
151	08MAR2010:00..	ADESA	2004	CHRYSLER	SEBRING 4C	LX	4D SEDAN	GREEN	AUTO	1	Alloy	
158	15MAR2010:00..	ADESA	2005	CHRYSLER	300	Tou	4D SEDAN	SILVER	AUTO	1	Alloy	
165	15MAR2010:00..	ADESA	2002	FORD	ESCAPE 4WD VE	XLT	4D SUV 3.0L XLT	BLUE	AUTO	1	Alloy	
172	22MAR2010:00..	ADESA	2004	DOODGE	STRATUS V6	SE	4D SEDAN SE	SILVER	AUTO	2	Covers	
179	22MAR2010:00..	ADESA	2003	SATURN	ION	3	4D SEDAN LEV.	MAROON	AUTO	1	Alloy	
186	29MAR2010:00..	ADESA	2006	MTSUBISHI	ENDEAVOR F..	LS	4D SPORT UTI..	WHITE	AUTO	1	Alloy	
193	29MAR2010:00..	ADESA	2006	CHRYSLER	PACIFICA FWD	Tou	4D SPORT TO..	SILVER	AUTO	1	Alloy	

Name	Type	Description	Date Modified	Modified By	Data Type
Auto	Entity	Folder contains objects related to Auto business logic	Aug 13, 2014 08:20 AM	Steve Sparano	
AutoVocabulary	Vocabulary	Vocabulary of terms related to auto purchase business	Aug 13, 2014 08:20 AM	Steve Sparano	
AutoActions	Entity	Entity for all business rule action terms	Aug 13, 2014 08:35 AM	Steve Sparano	
AutoTopEntity	Entity	Auto top level entity	Aug 13, 2014 08:33 AM	Steve Sparano	
aucguart	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Character
aucuon	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Character
byrno	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Integer
color	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Character
is online sale	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Boolean
meanvehicaldelta	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Decimal
mmracquisitionau...	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Integer
mmracquisitionau...	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Integer
mmracquisitionret...	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Integer
mmracquisitionreta...	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Integer
mmrcurrentaucuon...	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Integer
mmrcurrentaucuon...	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Integer
mmrcurrentretailav...	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Integer
mmrcurrentretaild...	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Integer
model	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Character
nationality	Term		Aug 13, 2014 08:33 AM	Steve Sparano	Character

Figure 7: Importing vocabularies from data sets.

Rules, Rule Sets, and Rule Flows

Business rules are organized into rule sets that allow the business analyst to express more complex logic, including ELSE and OR logic, and to define the interdependencies of business rules. Each rule has a condition, and when that condition is met – and evaluates to “true” – then that rule’s results can be used in downstream rules and in other rule sets. This feature allows business users to chain together business rules within rule sets.

With rule sets as a way to logically group rules, the business analyst can create multiple rule sets, each with its own objective. The rules are defined using the business rule category of functions. This provides a table-like format to select the entities and terms and a way to define both conditions and actions.

In this scenario, the business analyst needs to create rule sets for rules that address the following scenarios:

- General vehicle qualification rules, such as rules that examine the appraised cost of the vehicle, the probability that it will be returned, vehicle age and odometer readings.
- Evaluation of the vehicle’s VIN state (for example, some states suffer from more severe weather conditions, so cars from these locations are, on average, in poorer condition).
- Evaluation of the vehicle’s manufacturer, to align to end-consumer preferences.

Each of the rule sets contain individual business rules that assign a value to a specific, targeted business term. In this case, it’s an indicator of whether the vehicle is a good buy or a bad buy. To

do that, the business analyst must define the explicit conditions for each rule set.

For example, a business policy states that if the model predicts the probability of a vehicle being returned is greater than 70 percent, then the vehicle is considered a bad buy that should be avoided at auction. Conversely, if the vehicle has a calculated probability of less than 70 percent, then the vehicle is a good car and represents a risk that management has deemed acceptable.

However, the organization wants to use business rules to define special cases where the probability of return, e.g., a bad buy is greater than 70 percent, but the vehicle is still a good car to purchase. This determination is based on changes in the market that the predictive model doesn’t account for – and is therefore a good candidate to represent as a business rule(s).

Building Business Rule Sets

The following rules are created for the general vehicle qualification rules that will override a probability of “bad buy” being greater than 70 percent:

- **IF** make is Chevrolet AND odometer is less than 65,000 miles AND cost is less than \$4,000 AND vehicle is less than or equal to six years old **THEN** good car.
- **IF** make is Nissan or Toyota AND vehicle is less than or equal to eight years old **THEN** good car. These rules are shown in the decision table in Figure 8.

The screenshot displays a business rule configuration tool. The top section is a table with two main columns: 'Condition Term' and 'Action Term'. The 'Condition Term' column contains four rows of conditions, and the 'Action Term' column contains a single row with the action 'BadCar'.

#	Condition Term	Action Term
1	If CarScore > .7	1
2	Else CarScore <= .7	0
3	If CarScore > .7 AND make = 'CHEVROLET' AND vehicleage <= 6 AND vehbcost < 4000 AND vehodometer < 65000	0
4	If CarScore > .7 AND make = 'NISSAN' OR make = 'TOYOTA' AND vehicleage <= 8 AND vehbcost < 4000 AND vehodometer < 65000	0

Below the table is a 'Rule Details' section for the 'Nissan Exception' rule. It includes fields for Rule name, Record rule-fired data (checked), Name, Rule folder name, Rule set identifier, Rule order, and Description. The Rule expression is shown as: Condition: (CarScore > .7) AND (make = 'NISSAN' or make = 'TOYOTA') AND (vehicleage <= 6) AND (vehbcost < 4000) AND (vehodometer < 65000). The Action is: BadCar = 1.

Figure 8: Vehicle qualification rules.

In addition to the rule set for general qualification, the business also wants to implement policies using business rules based on the state where the vehicle VIN is registered. These rules include the following:

- **IF** VIN state is one of the states that management has deemed as high-risk **THEN** bad car.

These rules use the LIKE function to verify input values against a list of valid values and return a true or false result. Figure 9 shows a reference list of invalid states that management doesn't want to purchase vehicles from, and if the state matches the value in the list, then it will be noted as a bad car.

The screenshot shows a business rule configuration window. The top section is divided into two panes: 'Condition Term' and 'Action Term'. The 'Condition Term' pane shows a rule with the condition 'LIKE ('VA', 'CT', 'NJ', 'MD', 'GA', 'FL')' under the variable 'vinstate'. The 'Action Term' pane shows the action 'BadStateIndicator' with a value of 'True'. Below these panes is a 'Rule Details' section with the following information:

- Rule name: Invalid Vin States
- Record rule-fired data: ☒
- Name: AutoStateCheck
- Rule folder name: Auto
- Rule set identifier: 10001
- Rule order: 2
- Description: Identify vehicles from states from which we do not want to purchase

Figure 9: Invalid states defined using the LIKE function.

Lastly, due to the high cost of spare parts, the management team wants to avoid vehicles with parts that are no longer manufactured or simply too expensive to fix. Analysts create additional rules to address this scenario, including:

- **IF** make is from (list of makes no longer manufactured), **THEN** bad car.

These rules, as shown in Figure 10, enforce the company policy that vehicles from certain manufacturers are deemed as bad buys.

The screenshot shows a business rule configuration window. The top section is divided into two panes: 'Condition Term' and 'Action Term'. The 'Condition Term' pane shows a rule with the condition 'IN ('OLDSMOBILE', 'PONTIAC', 'SATURN', 'HUMMER', 'PLYMOUTH', 'TESLA', 'BENTLEY')' under the variable 'make'. The 'Action Term' pane shows the action 'BadCar' with a value of 'True'. Below these panes is a 'Rule Details' section with the following information:

- Rule name: Init
- Record rule-fired data: ☐
- Name: BadMakes
- Rule folder name: Auto
- Rule set identifier: 10047
- Rule order: 1
- Description:

Figure 10: Manufacturer rules.

Creating Business Rule Flows

Business rule sets are typically created and organized into a sequential flow as defined by their objective. The execution of the rule sets is determined by the order in which the rule sets are listed. Rules must be executed in a set order to get the desired results. The rule flow in Figure 11 illustrates the rule execution sequence based on the following requirements:

- Eliminate vehicles from bad states (Line 1).
- Eliminate invalid vehicle makes (Line 2).
- Apply rules based on model predictive values (Line 3).

The screenshot shows the AutoRuleFlow application window. The main table lists rule sets, and a detailed view of the 'VehicleQualification' rule set is shown below.

#	Run	Name	Version	Description	Date Modified	Modified By
1	<input checked="" type="checkbox"/>	AutoStateCheck	1.2	Rules for determining if vehicle's VIN	Aug 13, 2014 02:12 PM	Steve Sparano
2	<input checked="" type="checkbox"/>	BadMakes	1.1	Identify Vehicles with bad makes	Oct 27, 2014 06:04 PM	SAS Demo User
3	<input checked="" type="checkbox"/>	VehicleQualification	1.1	Rules for Vehicle Qualification	Oct 27, 2014 05:59 PM	SAS Demo User

#	Rule Name	Condition	Action
1	Default Rule Name 2	If (CarScore > .7)	then BadCar = True
2	Default Rule Name 3	Else if (CarScore <= .7)	then BadCar = False
3	Default Rule Name 1	If (CarScore > .7) AND (make = 'CHEVROLET') AND (vehicleage <= 6	then BadCar = False
4	Nissan Exception	If (CarScore > .7) AND (make = 'NISSAN' or make = 'TOYOTA') AND (...)	then BadCar = False

Figure 11: Ordering rule sets in rule flows.

Analytically Driven, Discovered Rules

Data-driven rule discovery uses analytical routines to derive business rules. Rather than manually defining the rules, or importing rules from an existing file, this data-driven technique allows a user to automatically generate business rules from a representative data set of historical data. The technique defines one or more business rule sets and business rule flows based on predefined algorithms. Using a point-and-click wizard, the analyst can discover rule sets and rule flows, which can then be inspected and refined.

SAS Decision Manager provides four different options to discover business rules, including decision trees, market basket analysis, scorecarding and RFM (recency, frequency, monetary) statistical analysis. An important feature of discovering rules is that automated clustering techniques can group similar rules together into rule sets. This rule clustering essentially collects similar rules into rule sets that can be managed, modified, ordered and deployed as rule flows to meet the decision needs of the organization.

The decision table allows analysts to view the discovered rules in a familiar format for editing the rule sets and the rule logic, eliminating the need to program and supporting the same familiar format as business rules that are defined by the analyst to the system.

Testing Rules for Validity

Testing helps evaluate that the documented business rules are producing the desired output. This is an important step to take before publishing them into a decision process.

Testing also allows a business analyst to evaluate the business logic in the rule flow by using operational data inputs and running those tests to assess the output. After conducting rule flow tests, users can verify results in the corresponding output tables.

Rule-Fired Analysis

Understanding what rules were triggered (and the number of rules triggered), or fired, helps the analyst evaluate the results – and to assess whether the correct rules have been specified.

Business analysts can make this determination from the rule-fired analysis reports, displaying the rules that lead to the result as depicted in Figure 12.

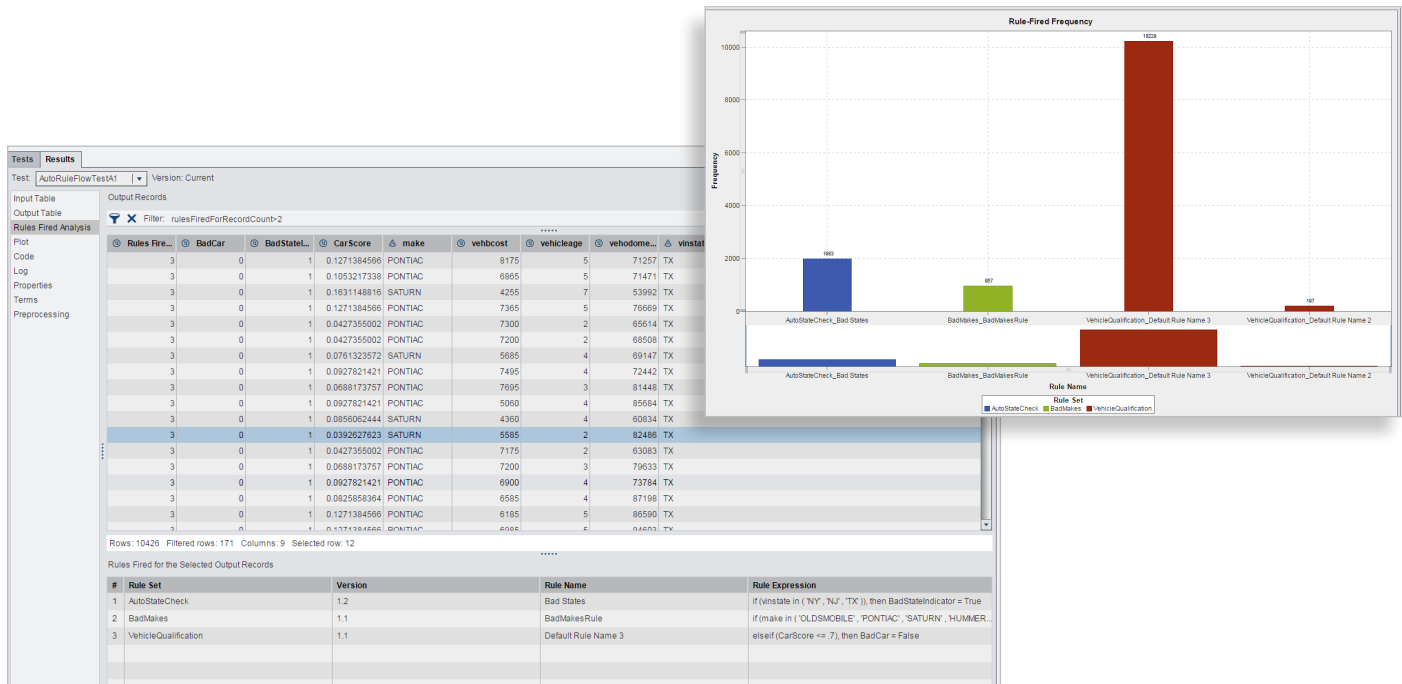


Figure 12: Rule-fired analysis.

Publishing Rule Flows

After testing and verifying results, a business analyst can publish the rule flow for consumption by various deployment engines. To publish, the versions of rule sets used in the rule flow should be locked down to prevent any further changes to the logic once deployed. By locking the rule set version, the business can make sure that the published versions are unchanged until a new version of the rule flow is available.

SAS Decision Manager supports detailed authorizations for this type of activity. Figure 13 shows the selected versions of

rule sets that have been locked in the auto company's scenario. These are now included in rule flows to be sent for publishing.

The next step in the process is to put deployed rules, along with the corresponding predictive models, to work in business operations where these decisions are made. The decision to purchase a used car is made by vehicle purchasing agents who access web applications. These applications use web services behind the scenes to call the decision engine to render decisions – or as part of a nightly batch process that renders decisions on incoming transaction files.

#	Run	Name	Version	Description	Date Modified	Modified By
1	<input checked="" type="checkbox"/>	AutoStateCheck	1.2	Rules for determining if vehicle's VIN ...	Aug 13, 2014 02:12 PM	Steve Sparano
2	<input checked="" type="checkbox"/>	BadMakes	1.1	Identify Vehicles with bad makes	Oct 27, 2014 06:04 PM	SAS Demo User
3	<input checked="" type="checkbox"/>	VehicleQualification	1.1	Rules for Vehicle Qualification	Oct 27, 2014 05:59 PM	SAS Demo User

Figure 13: Rule flow with locked rule set versions.

Integrating the Decision Components

With SAS Decision Manager, analytical models and business rules can be deployed individually or in combination as part of a decision flow. The matrix in Figure 15 summarizes deployment targets supported out-of-the-box.

Deployment Target	Rule Flow	Model	Decision Flow (Rules and Models)
Web Services	✓		✓
In-Database	✓	✓	✓
SAS Stored Process	✓		
SAS Batch Processing	✓	✓	✓

Figure 14: Matrix of out-of-the-box deployment targets.

Deploying Automated Decisions

Once the auto dealer created and deployed rules, the next phase is to automate these rules within a decision process. The logical flow of a decision is illustrated in Figure 15. Analysts can relate operational data, predictive models and business rules to each other by simply dropping objects onto the palette and setting their individual properties. Additional objects such as branching, filtering, etc. are all predefined to enhance decision logic complexity. Users can test the entire flow from the interface. All objects seen in Figure 16 are created using definitions that are accessible within a decision flow.

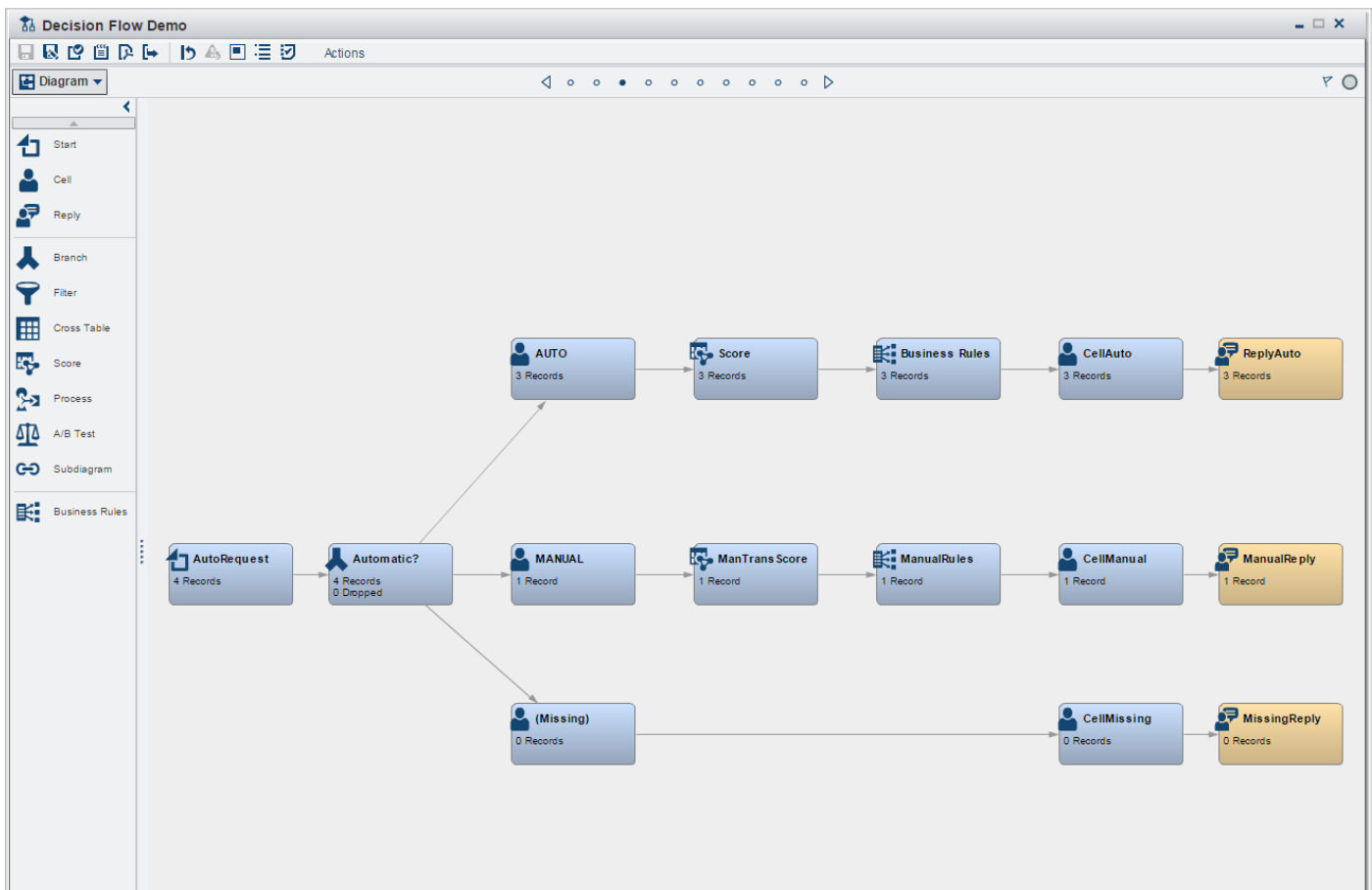


Figure 15: Operational data, models and business rules defining the BadCar decision process.

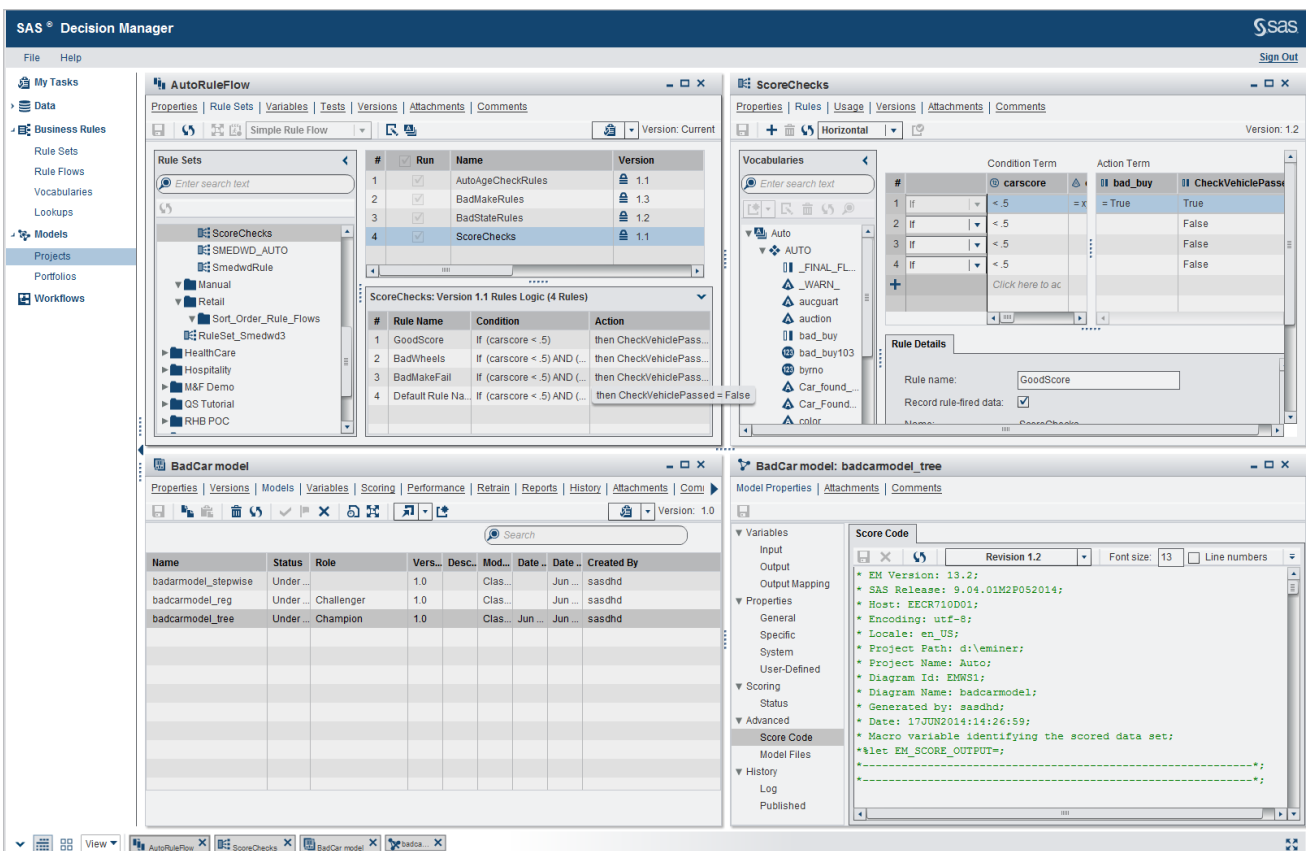


Figure 16: The left side shows categories of object definitions accessible within the Flow Builder.

Deployment of Decision Components

To deploy the rules flows and other elements that support decision-making efforts, SAS Decision Manager supports batch, real-time and in-database deployments.² SAS Decision Manager provides a scalable tiered architecture for design time and run time. Decision flows can be deployed as web services (using SOAP or REST) for real-time execution.

To minimize data movement, in-database deployments are supported for both rules and models.³

Conclusion

Many forward-thinking organizations are finding new ways to be more efficient and drive better – and more automated – decisions that fuel business operations. And these same organizations know that business analysts and data scientists need to collaborate to get the best results.

SAS Decision Manager provides the features that organizations need for faster and easier ways to deploy powerful analytical models to their operations. SAS Decision Manager has the capabilities required to combine powerful analytical models with the flexibility of business rules that can adapt to changing business policies and guidelines. By offering an integrated suite for modelers, business analysts and IT, SAS Decision Manager streamlines operational data into a decision process to deliver better operational actions faster and more efficiently.⁴ This reduces overall costs and minimizes the lag between the time the organization can react to changing environments and customer demands.

² For batch deployments, SAS Data Integration Studio is included with SAS Decision Manager so that you can save the defined code and then schedule execution with any scheduler. Any process flow built within SAS Data Integration Studio can be a candidate for scheduling.

³ For in-database deployment, SAS Code Accelerator and/or SAS Scoring Accelerator would be added for implementation (for business rules and models, respectively).

⁴ One company reduced predictive model deployment by 75 percent and saw other benefits with SAS Decision Manager. Read more in this IDC paper: [SAS Enterprise Decision Management at a Global Financial Services Firm: Enabling More Rapid Implementation of Decision Models into Production](#).

To contact your local SAS office, please visit: sas.com/offices

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. Copyright © 2014, SAS Institute Inc. All rights reserved.
107361_S128208.1114

