

UServ Product Derby With DMN

Using DecisionsFirst Modeler and OpenRules to build an executable DMN model for UServ.

In response to the UServ Product Derby a Decision Requirements Model was developed in DecisionsFirst Modeler and linked to Decision Logic modeled in OpenRules.

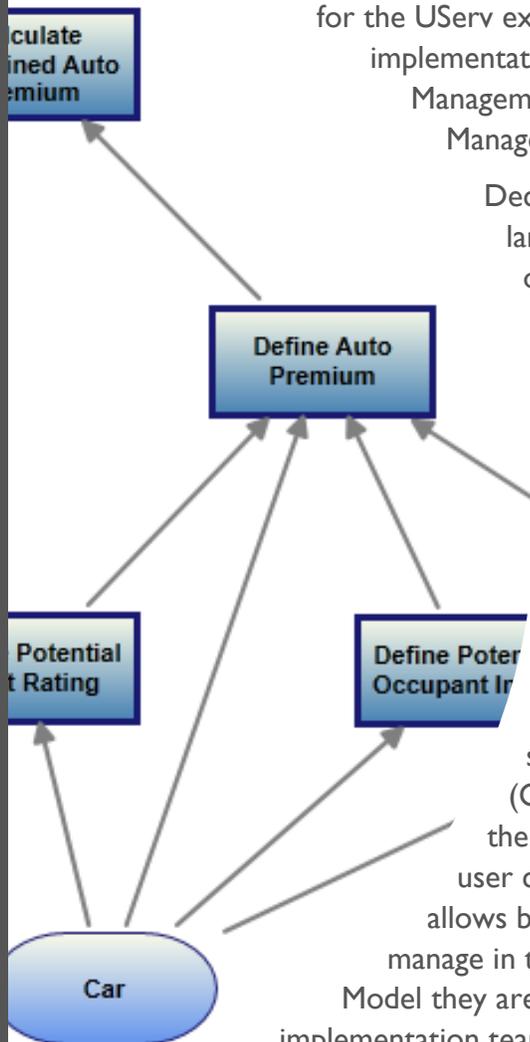
This paper outlines the Decision Requirements Model and associated Decision Logic created using DecisionsFirst Modeler and OpenRules to implement the UServ Product Derby example. It does not contain everything describes the iterative steps that develop and complete an effective Decision Requirements Model using industry standard notation (DMN). Defining decision requirements as part of your overall requirements process offers many benefits;

A Decision Requirements Model provides such as the one created for the UServ example provides needed structure for the implementation of decision logic in a Business Rules Management Systems (BRMS) or Business Decision Management System such as OpenRules.

Decision Requirements Models are a common language across business, IT and analytic organizations improving collaboration, increasing reuse, and easing implementation.

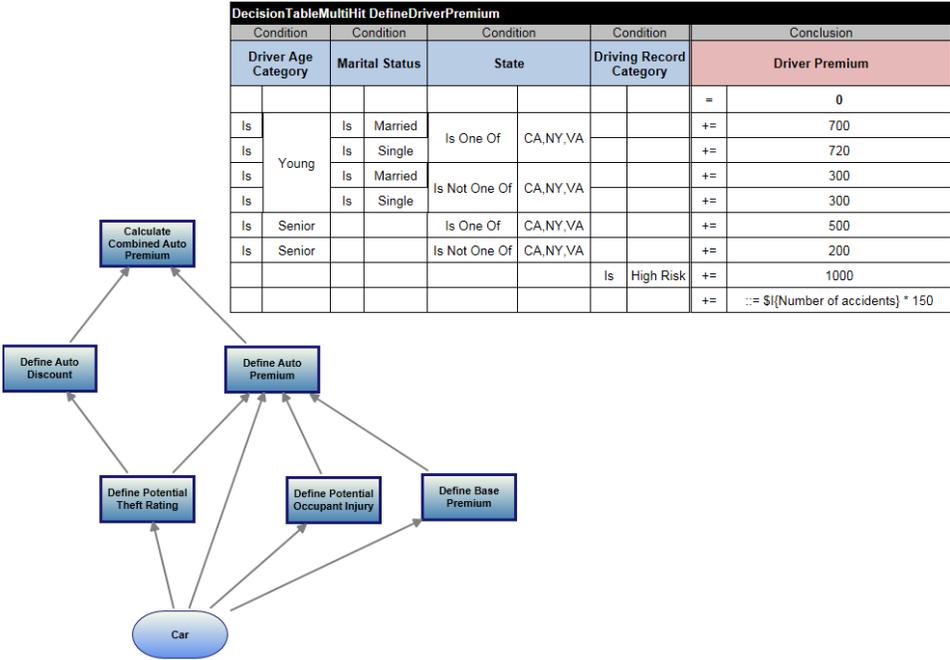
Extending these models with Decision Logic Models built in a tool such as OpenRules allows them to be executable. Some tools allow this logic to be added directly to the model but it is also possible to link the requirements model to decision logic components developed in an implementation environment.

In this approach the decision logic for a specific decision is specified in the BRMS (OpenRules) and then linked to the a decision in the model (in DecisionsFirst Modeler) so that a user can easily navigate from one to the other. This allows business users to find the rules they need to manage in their BRMS using the Decision Requirements Model they are already familiar with. It also allows the implementation team to create other artifacts in the BRMS without exposing them to the business users. Figure 1 shows an example



of linking the decision Define Auto Premium in a Decision Requirements Model to a decision table implemented in OpenRules.

Figure 1: Linking Decisions to BRMS implementations



Source: DecisionsFirst Modeler

This approach allows Decision Requirements Models to be linked to business rules or decision logic, to predictive analytic models developed in data mining and other advanced analytic tools, indeed to any implementation. In the UServ example all the decisions are represented as decision logic.

Decision Requirements Models are also linked to business processes through the decisions involved in the model. Any decision can be identified as required by a business process and/or as executed by a task within that process. This clearly identifies the process context for the decisions that are modeled in the Decision Requirements Model. It is the decisions, however, not the models that are linked.

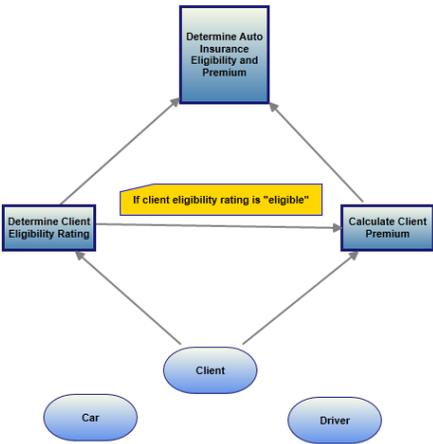
This approach of mapping decision requirements to implementations and using the Decision Requirements Model as a map has a number of benefits:

- ▶ It ensures that business users can find and edit the implementation components that matter to them directly from the Decision Requirements Model they are familiar with – the one that describes their problem.
- ▶ For decision logic, it ensures only one version of the rules – the rules executed by the BRMS – are ever used, preventing duplication and confusion.
- ▶ It allows the reuse of existing assets in Decision Requirements Models.

This is the approach used in the submission. While the full model cannot be illustrated in a short document, the basic process is illustrated below:

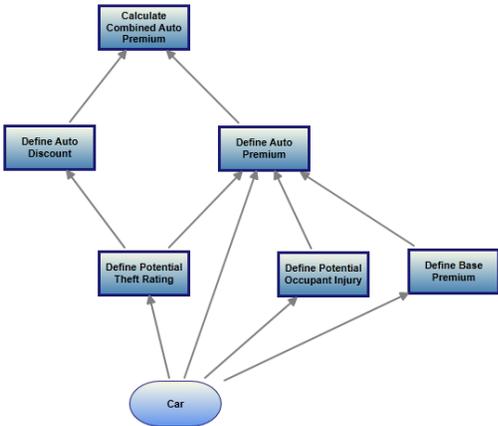
First high level models were developed in DecisionsFirst Modeler to show manage the high level specification of the problem. Core Decisions and Input Data were identified and linked. DecisionsFirst Modeler has an underlying shared repository so these objects can be on multiple diagrams without duplication – they are reused between the diagrams. Each area can be focused on individually, creating a model for that one area and taking advantage of the repository to manage reuse. Figure 2 and Figure 3 show the high level and a focused detail model.

Figure 2: High Level Overview



Source: DecisionsFirst Modeler

Figure 3: Focused Detail Model

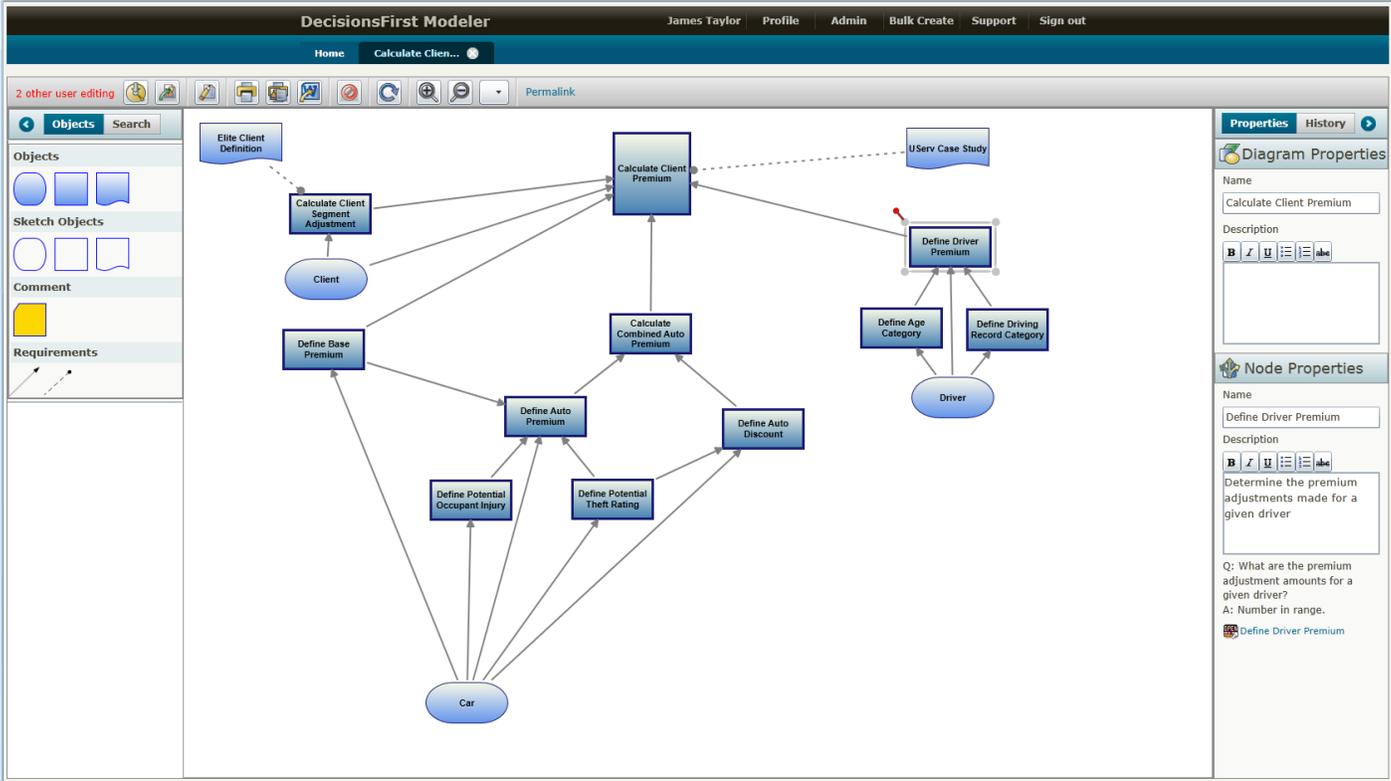


Source: DecisionsFirst Modeler

The complete model was developed iteratively in this way, creating the model shown in Figure 4. Decisions in this model were described, especially specifying questions and allowed answers for each, and the main decisions were linked to processes, business objectives and the other business-focused objects defined in DMN. An example of the complete information managed in DecisionsFirst Modeler is shown in Figure 5.

The UServ product derby lacks much of the typical business context but a reasonable amount can be inferred. In addition few if any Knowledge Sources were specified. Generally Decision Requirements Models should include Knowledge Sources to show where the decision logic can be found or what can be used to refine it.

Figure 4: Complete Model



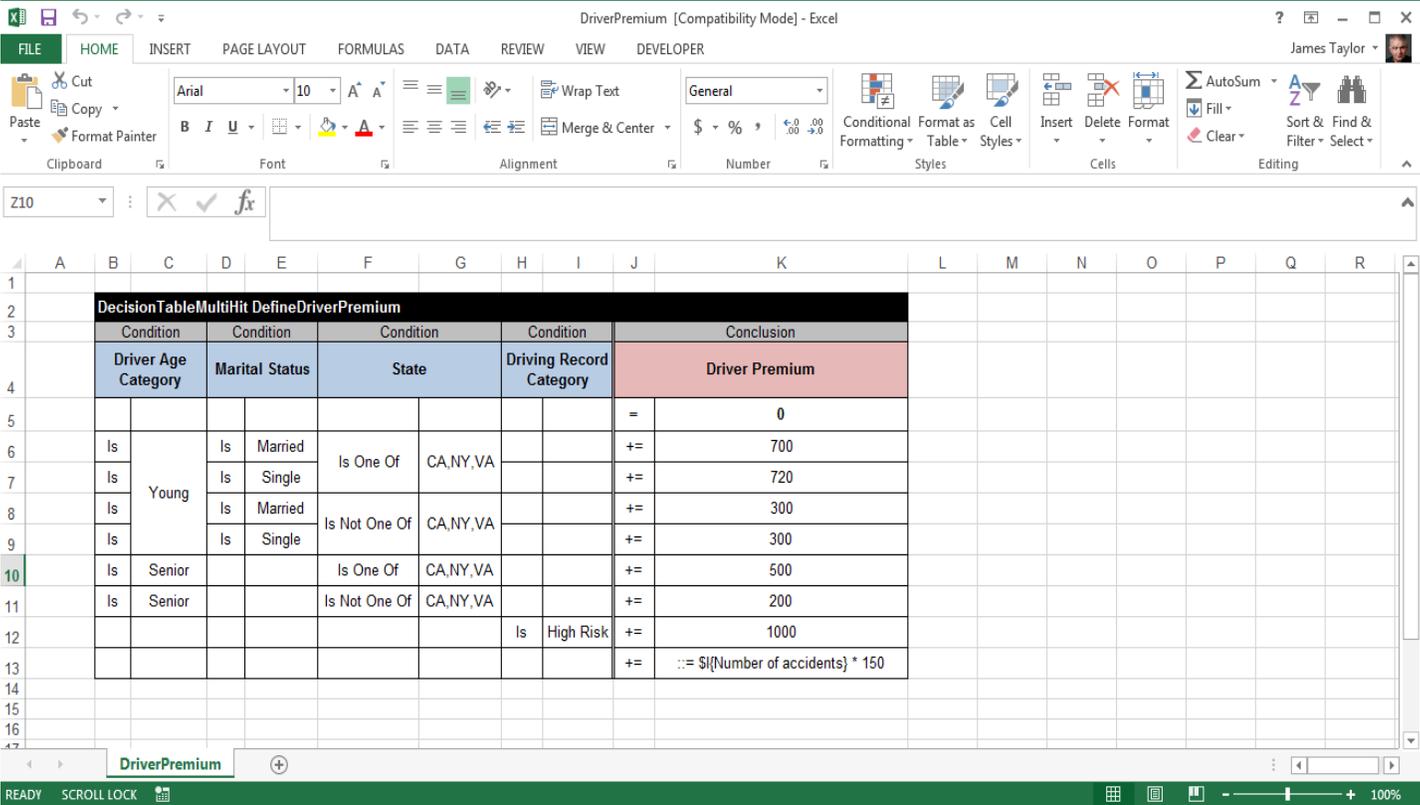
Source: DecisionsFirst Modeler

Figure 5: Decision Details

Source: DecisionsFirst Modeler

As the model is completed it can be linked to the Decision Logic required for each decision in the model. These are written and tested in OpenRules using Excel. An example for a specific decision, Define Auto Premium, is shown in Figure 6.

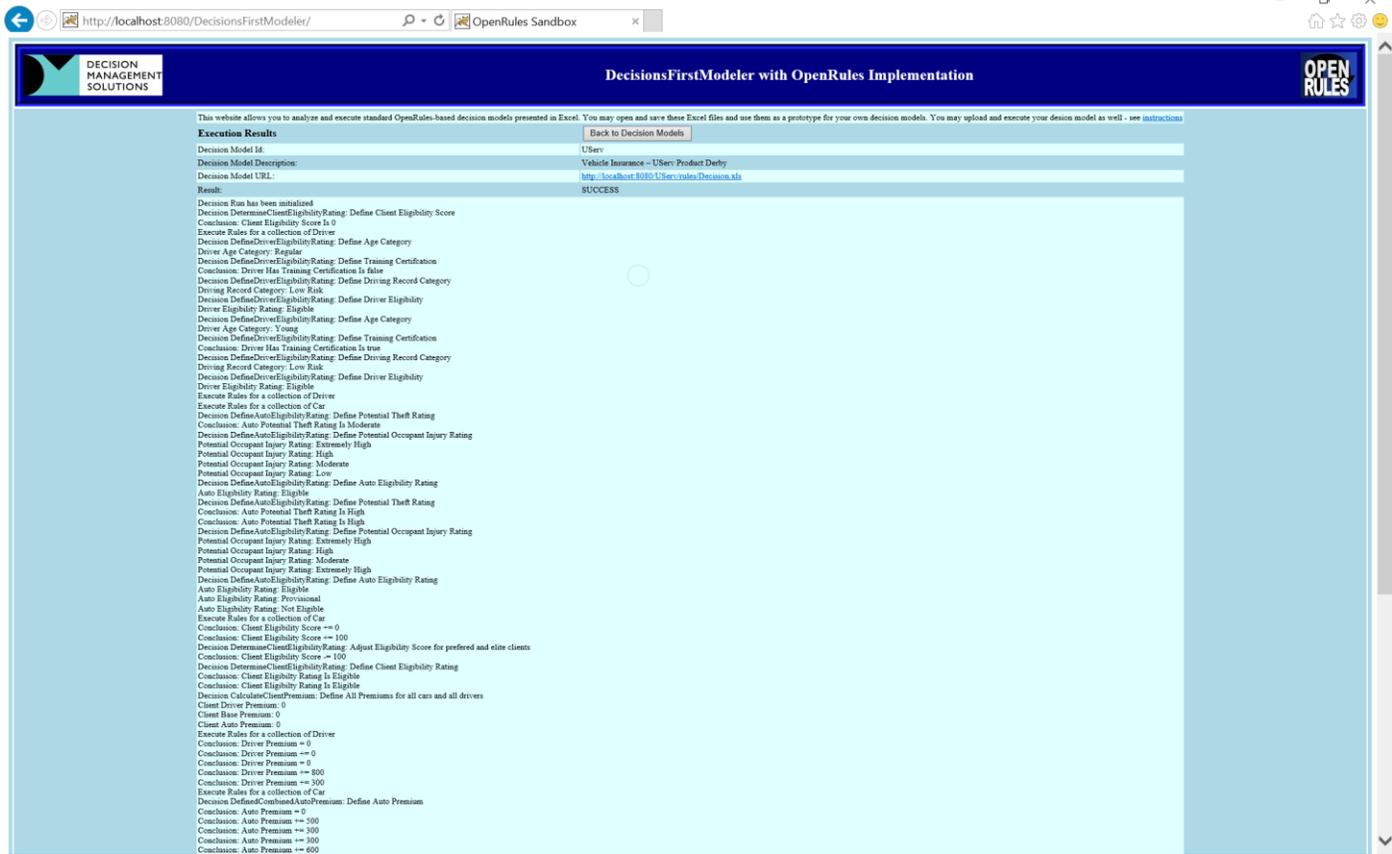
Figure 6: Linking Decisions to BRMS implementations



Source: OpenRules

The model developed in this way is executable as all the decision logic has been implemented in OpenRules decision tables. Figure 7 shows the results of a test execution.

Figure 7: Linking Decisions to BRMS implementations



Source: OpenRules

This document only has a subset of the model and does not show the integration to its full extent. A video demonstration is available showing the integration at <https://www.youtube.com/watch?v=XtC5MU-pci4>.

More Information

[DecisionsFirst Modeler](#) is available from Decision Management Solutions. DecisionsFirst Modeler is available as a free Basic Version (SaaS) and a paid Enterprise Edition (SaaS or on-premise). DecisionsFirst Modeler Enterprise Edition provides a full API. This allows read-only access to the full Decision Requirements Model for easy integration. In addition the API is used to integrate DecisionsFirst Modeler with implementation environments.

Decision Management Solutions provides consulting and training in Decision Requirements Modeling as well as the use of DecisionsFirst Modeler.

Information on [OpenRules](#) is also available and a free version can be downloaded.

References

- ▶ Debevoise, Tom and Taylor, James (2014). *The MicroGuide to Process and Decision Modeling in BPMN/DMN: Building More Effective Processes by Integrating Process Modeling with Decision Modeling*.
- ▶ Object Management Group. *Decision Model and Notation (DMN) Specification 1.0* Current version at <http://www.omg.org/spec/DMN/Current>
- ▶ Taylor, James (2011). *Decision Management Systems – A Practical Guide to Using Business Rules and Predictive Analytics*. IBM Press.

Contact Us

If you have any questions about Decision Management Solutions or would like to discuss engaging us we would love to hear from you. Email works best but feel free to use any of the methods below.

Email : info@decisionmanagementsolutions.com

Phone : +1 650 400-3029

Web : www.decisionmanagementsolutions.com

