Decision Model and Notation (DMN) standard deliver a powerful ROI by improving processes, effectively managing business rules projects, framing predictive analytics efforts, and ensuring decision support systems and dashboards are action-oriented.

This paper describes the iterative steps that develop and complete an effective Decision Requirements Model using the industry standard DMN notation. Defining decision requirements as part of your overall requirements process offers many benefits;

- Modeling decisions makes business processes less complex, more robust in the face of change and easier to manage.

A Decision Requirements Model provides the needed structure for the implementation of a Business Rules Management Systems (BRMS), supporting iteration and agile development.

- Framing data mining and predictive analytics projects with a Decision Requirements Model links analytics to business results and helps ensure successful deployment.

- Understanding the decisions relevant to a dashboard or decision support environment structures knowledge and puts a premium on taking action.

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

This paper describes why modeling decisions is effective and necessary and walks through the steps involved in identifying, describing, modeling and refining Decision Requirements Models. Appendices include a discussion of the various kinds of decisions suitable for modeling and some techniques to help discover them.
I Why Model Decisions

There is an emerging consensus that a Decision Requirements Model is the best way to specify decision-making. Decision Requirements Models can and should be developed in an industry standard way using the Object Management Group’s Decision Model and Notation (DMN) standard. Adopting this industry standard gives users access to a broad community and a vehicle for sharing expertise more widely.

Decision modeling is a new technique in the International Institute of Business Analysts (IIBA) Business Analyst Body of Knowledge or BABOK® 3.0 released in 2015.

Today business analysts use a variety of techniques to accurately describe the requirements for an information system. However, current requirements approaches don’t tackle the decision-making that is increasingly important in information systems.

1.1 Over-complex and Fragile Workflow

Most systems involve some workflow and this is increasingly described by business analysts in terms of business process models. Experience shows that when process modeling techniques are applied to describe decision-making, the resulting process models are over complex. Decision-making modeled as business process is messy and hard to maintain. In addition, local exceptions about decision-making can quickly overwhelm process models. Yet without some kind of model for decision making the risk is that contradictory or short descriptions will lack the necessary detail and resolving this during development creates delays and additional costs.

Identifying and modeling decisions separately from the process ensures these local exceptions no longer clutter up the process. This makes business processes simpler and makes it easier to make changes. By modeling the decisions, a clear and concise definition of decision-making requirements can be developed. A separate yet linked model allows for clarity in context.

Most process models today are developed using the Business Process Model and Notation (BPMN) standard published by the Object Management Group. The Decision Model and Notation (DMN) standard has been designed to work alongside BPMN, providing a mechanism for modeling the decision-making represented in a Task within a process model. DMN need not be used with BPMN but it is highly compatible with BPMN.
1.2 Unstructured, Waterfall Business Rules Development

Many implementations of Business Rule Management Systems (BRMS) create what can best be called a “Big Bucket O’ Rules.” The team interviews Subject Matter Experts (SMEs), reads policy manuals and reverse engineers code into business rules. Because there’s no organizational structure to the rules they end up being grouped by source or by the person doing the rule identification. The result is a large number of low-level rules in one, big, bucket. Such a bucket of rules is not usable as it’s difficult to manage the rules properly. As policies or regulations change and as new business needs are identified, teams often struggle to update the right business rules.

Successful business rules projects begin by focusing on the decision-making involved. For business rules projects, clarity about decision requirements scopes and directs business rules analysis. It is essential to first define the decision-making required and only then focus on details like the specific business rules or predictive analytic models involved. Specifying a Decision Requirements Model provides a repeatable, scalable approach to scoping and managing decision-making requirements for business rules efforts.

1.3 Poorly Framed Advanced Analytics Projects

Established analytic approaches like CRISP-DM stress the importance of understanding the project objectives and requirements from a business perspective, but to date there are no formal approaches to capturing this understanding in a repeatable, understandable format. Decision Requirements Modeling closes this gap.

Decision Requirements Modeling is a successful technique that develops a richer, more complete business understanding earlier. Decision Requirements Modeling results in a clear business target, an understanding of how the results will be used and deployed, and by whom. Using Decision Requirements Modeling to guide and shape analytics projects reduces reliance on constrained specialist resources by improving requirements gathering, helps teams ask the key questions and enables teams to collaborate effectively across the organization—bringing analytics, IT and business professionals together.

Using Decision Requirements Modeling to document analytic project requirements enables organizations to:

- Compare multiple projects for prioritization, including allowing new analytic development to be compared with updating or refining existing analytics.
- Act on a specific plan to guide analytic development that is accessible to business, IT and analytic teams alike.
- Reuse knowledge from project to project by creating an increasingly detailed and accurate view of decision-making and the role of analytics.
- Value information sources and analytics in terms of business impact.
1.4 Passive and Confused Dashboard Designs

Dashboards are decision support systems but their design does not usually consider decisions. Most designs are driven by user interface and information visualization requirements and by the underlying data models. A dashboard design based on decision modeling provides a logical structure for going beyond information presentation to include predictions and recommendations so users can progress from ‘what happened?’ to ‘what can happen?’, and on to ‘what action can be taken?’

Decision Requirements Modeling is an ideal technique for capturing business requirements for a dashboard, and then for driving the design, implementation and maintenance processes. Decisions are described in the context of goals, metrics, business processes, trigger events, systems and stakeholders. This ensures a comprehensive, future-proof dashboard design—and as a side-effect enables formal Performance Management, governance, compliance, escalations and training.

Decision Requirements Models enable dashboard implementation by tying data requirements and knowledge requirements to presentation elements. These requirements are defined in the context of stakeholders and systems so that governance and control requirements are simultaneously identified. The graphical nature of Decision Requirements Models allows visual partitioning to drive phase-wise implementation, as well as a robust and logical work-breakdown-structure for formal project management.

1.5 Decisions as a Shared Framework for Implementation

Decision modeling provides a framework that teams across an organization can use and that works for business analysts, business professionals, IT professionals and analytic teams. Decisions are more easily tied to performance measures and the business goals of a project. This makes it easier to focus project teams where they will have the highest impact and to measure results.

Many business analysts have known all along that decisions, and decision-making, should be a “first class” part of the requirements for a system. Systems that assume the user will do all the decision-making fail to deliver real-time responses (because humans struggle to respond in real-time), fail to deliver self-service or support automated channels (because there is no human available in those scenarios) and fail front-line staff because instead of empowering them with suitable actions to take it will require them to escalate to supervisors. What business analysts have lacked until now is a standard, established way to define these requirements.

*Decision modeling is a powerful emerging technique for business analysis. Using the standard DMN notation to specify Decision Requirements Diagrams and so specify a Decision Requirements Model allows the accurate specification of decision requirements.*
2 Building Decision Requirements Models

Decision modeling has four steps that are performed iteratively:

1. Identify Decisions.
   Identify the decisions that are the focus of the project.

2. Describe Decisions.
   Describe the decisions and document how improving these decisions will impact the business objectives and metrics of the business.

3. Specify Decision Requirements.
   Move beyond simple descriptions of decisions to begin to specify detailed decision requirements. Specify the information and knowledge required to make the decisions and combine into a Decision Requirements Diagram.

4. Decompose and Refine the Model.
   Refine the requirements for these decisions using the precise yet easy to understand graphical notation of Decision Requirements Diagrams. Identify additional decisions that need to be described and specified.

A Note on Terminology

In the Decision Model and Notation Standard V1.0, Decision Requirements Diagrams and decision logic together create a decision model. We use Decision Requirements Model to refer to the models and diagrams and decision modeling when referring to the overall approach.

This process repeats until the decisions are completely specified and everyone has a clear sense of how the decisions will be made. At this point you can generate a requirements document, packaging up the decision-making requirements you have identified. This can act as the specification for business rules implementation work or for the development of predictive analytics. Alternatively, you can extend the model with decision logic, such as decision tables, to create an executable specification of your decision-making.

Figure 1: Iterative Decision Modeling Cycle
This paper will describe each step in this process in turn. Remember though that this is an iterative not a linear process.

All of the examples in this document have been produced using DecisionsFirst Modeler. DecisionsFirst Modeler is a collaborative platform for Decision Requirements Modeling available from Decision Management Solutions. For more information, see Appendix C: DecisionsFirst Modeler.

2.1 Identify Decisions

When getting started with decision requirements, decisions can be readily identified as part of the overall requirements gathering process. As discussed in the introduction, decision-making appears in business processes, in use cases or as a specific requirement.

Additional techniques, especially useful in finding decisions that could be automated, managed or improved using business rules and predictive analytics are:

- Examine Business Processes for hidden decision-making
- Examine Performance Measures for the decisions that make a different
- Examine BI reports to see the decisions they are designed to influence
- Examine legacy systems to find previously automated decisions
- Brainstorm with subject matter experts to identify the decisions they care about

For background on the kinds of business decisions these techniques might identify, see Appendix A: About Decisions. Decision identification techniques are discussed in detail in Appendix B: Decision Identification Techniques.

As you iterate you will also identify decisions, generally as part of specifying the information requirements of a decision you have already identified (see Decompose Decisions below).
2.2 Describe Decisions

Decisions are the core of DMN and are shown using a simple rectangle that contains the name of the Decision:

![Select Marketing Offer]

A Decision Requirements Diagram can be created that contains one or more such Decisions and it is common to initially create one such diagram for each area of focus on the project.

The properties defined in this section are those specified in the Decision Model and Notation standard except where noted. These properties would be recorded for each Decision in a tool such as DecisionsFirst Modeler to begin creating a repository of decision requirements.

We now have a set of business decisions. Typically these will be repeatable and non-trivial decisions where it is worth understanding how the decision will be made in advance of making it. Having identified these decisions, we must next describe them at a high level to ensure everyone means the same thing by them.

The list will generally include a name and a short description for each Decision. This is important but additional data should be captured for each Decision at this point:

- Question and Allowed Answers.
- Business context.
- Organizational context.
- Application context.

2.2.1 Question and Answers

Questions are a powerful tool for explaining a decision. Look at each decision and determine the question that must be answered to make the decision. Make sure it is clear about the subject of the decision, about timing and about the scope or limits to the question. Make it as precise as possible and avoid questions that start “How” or use I/We.

Table 1 shows some examples with both good and bad, or at least less good, questions. Each question implies a set of allowed answers. The allowed answers, the options from which the decision is selecting, should be documented for each question. Allowed answers come in a variety of formats as shown in Table 2.
Table 1: Example Questions

<table>
<thead>
<tr>
<th>Decision</th>
<th>Good question</th>
<th>Bad question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Retention</td>
<td>What retention offer should be made to this customer when they call to cancel their service?</td>
<td>How can we retain this customer?</td>
</tr>
<tr>
<td>Supplier Selection</td>
<td>Which of the currently approved suppliers should be selected for this specific parts order?</td>
<td>What supplier should we use?</td>
</tr>
<tr>
<td>Preventative Action</td>
<td>What is the prioritized list of preventative actions for this quality team on this line today?</td>
<td>What preventative action should the quality team take?</td>
</tr>
</tbody>
</table>

In addition, any answer may come with supporting information such as messages or explanatory text.

Make sure the allowed answers follow from the question, that all answers are reasonable given the question. Use the answers as a way to confirm and refine the question. Do not skip this step as questions and allowed answers are powerful tools for clarifying a shared understanding of decisions.

Table 2: Allowed Answer Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes/No</td>
<td>Yes or No</td>
<td>Or True/False, 1/0 etc</td>
</tr>
<tr>
<td>Number</td>
<td>A numeric value</td>
<td>Often constrained to a value in a specific range</td>
</tr>
<tr>
<td>Specific Value</td>
<td>One of the values specified in a list</td>
<td>For example Accept / Reject / Refer</td>
</tr>
<tr>
<td>Database value</td>
<td>One of the values stored in a database</td>
<td>The specifics of how to get the list of options is specified. Could be a product, piece of content etc.</td>
</tr>
<tr>
<td>Other</td>
<td>Generally a string or block of text</td>
<td>Such as a custom script or personalized email body</td>
</tr>
<tr>
<td>Structure</td>
<td>A set of values each of which is of one the allowed types</td>
<td>Some decisions involve the rolled-up output of their component decisions.</td>
</tr>
</tbody>
</table>

DMN defines both Question and Allowed Answers as text properties of a Decision. For Allowed Answers in particular it is worth thinking about the structure of that text as noted.
In addition to the core question and allowed answers properties, DMN puts decisions in context by linking them to various objects—both new ones and some defined in existing standards. A simplified model of these relationships is shown in Figure 2.

These relationships are not shown on Decision Requirements Models but are nevertheless important to capture to fully understand the decisions being modeled. They can be divided into business, organizational, and application context relationships. These relationships should be captured in a repository so they too can be kept up to date.

2.2.2 Business Context

Each decision should be put into a business context. This involves assessing the KPIs and objectives in the business area and determining which KPIs or objectives are impacted by the decision. A decision impacts a KPI or objective if changing how the decision is made could be reasonably expected to have an impact on the value recorded for the KPI or objective.

It can sometimes seem that every decision impacts every KPI or metric, or at least that it has the potential to do so. Try not to leave out links that are material but also be realistic and focus on linking decisions to the KPIs or metrics on which they have a significant impact.

This linkage tells you two things about your decisions.

- First it tells you how to tell good decisions from bad ones. A good decision will generally have a positive impact on the KPIs or objectives to which it is linked and a bad one will not. Of course, this can be complicated when a decision is linked to several KPIs as the impact of a single change may be positive on one and negative on another.

- Second it helps scope what the organization has to worry about if the decision-making changes. It allows a proposed change to decision-making to be explicitly linked to the possible impacts.
2.2.3 Organizational Context

Decision-making is central to organizations and the various roles played in decision-making by an organization matters. Three particular roles can be played by organizations, organizational units or roles within those organizational units. For repeatable decisions these relationships are not generally to an individual but to a team or department or a role within them. It should also be noted that multiple organizations or roles can own, make or care about a specific decision.

- Who owns the decision?
  Who decides how the decision should be made? Who is the approver of the approach taken? Who decides that we shall decide this way and not that way? Who will be maintaining the business rules for this decision?

- Who makes the decision?
  Often one part or level of the organization owns a decision but a different part or level makes the decision day to day. Sales management might own the pricing decision, for instance, but pricing decisions are made by the sales teams themselves. If decisions are automated or partially automated, then the organization that passes on the decision to a customer or supplier could be considered the maker too.

- Who else cares about the decision?
  While many decisions have only owners and makers, sometimes there are other parts of the organization that might take an interest and expect their opinion to be taken into account.

2.2.4 Application Context

Decisions also have an application context. There are places and times where they are applied. In particular decisions show up in Business Processes, Systems and Events. Some of this context information is often specified in advance for a project—the business process that contains the decision for instance—but it is worth checking that all the uses of each decision are identified.

- Business Processes.
  Very few business processes can execute all the way through without decisions being made. These decisions might be automated or manual but they are...
essential to the process. Generally, these show up as a task or a task list with multiple paths following the decision. Understanding which processes and tasks require a decision helps you see where in the business it will be used.

DMN only defines a link from Decision to a BPMN Process—there is no equivalent to System or Events at present. It does also allow a Decision to be linked to a specific Task within the Process.

- **Systems.**
  Existing systems are often fixed points in a company’s ecosystem, hard to change and essential to day to day operations. Understanding that a decision must be delivered into or supported by a decision helps identify implementation constraints on the technology and approach used for decision-making.

- **Events.**
  As organizations move to a more event-based environment it can be very indicative to show which business events might trigger a decision, typically to determine what to do in response to the event.

This context shows where and how any future decision-making will be used.

Often these context relationships need only be captured for “top level” decisions. These are those understood by the business to be “atomic” and generally those that are identified in a business process context. The organizational links in particular, however, may be usefully captured for decisions throughout a model.

### 2.3 Specify Decision Requirements

Identifying and documenting the decisions involved scopes the project. To effectively automate or improve decision-making, however, we need to move beyond describing decisions to specifying detailed decision requirements. This involves identifying a number of elements:

- The data a decision requires.
- Knowledge about how the decision is to be made.
- Related or component decisions.

All of these elements can be displayed on a Decision Requirements Diagram where they can be associated to form a Decision Requirements Model.

Each of these elements can be developed sequentially but you are more likely to iterate through these items to develop a rich Decision Requirements Model. You may identify some related decisions first and then use this understanding to identify additional Input Data requirements or begin with a Knowledge Source. The order in which these elements are created is not material.
2.3.1 Input Data Requirements

Input Data is shown in DMN using an oval containing a name that represents the information—the business entity—in question:

![Image](Marketing Offers)

As the information that is required is identified, the Input Data entities that represent it are added to one or more Decision Requirements Diagrams so that they can be linked as discussed in A Decision Requirements Diagram.

Decisions often require Input Data, information from outside the decision-making context about customers, parts, orders etc. This information can describe the subject of the decision or it could be reference data of one sort or another.

Input Data is generally described at the level of business concepts or business objects. These are coherent sets of information, things the business cares about. These could be commonly used internal data structures such as a customer record or a claim, or external data sources such as a credit report.

Try not to focus on databases or content management systems but on the kind of information stored in them. This is a requirements model rather than a technical architecture.

It is generally sufficient to describe the Input Data only at the concept or business entity level but you can also specify the data structure within an Input Data if you like. You will need to do this before the model can be extended to include decision logic (see Extending Decision Requirements to Decision Logic below).

The Input Data required for a Decision is linked to the decision with Information Requirement links. These are shown using a solid line with an arrow head pointing from the Input Data to the Decision that has the requirement.

There is no distinction in DMN between Input Data that is always required and Input Data that is only sometimes required so information requirements may be optional when the decision is made for a specific transaction.

For important Input Data it can be useful to record:
- Is the information internal or external to the organization?
- Is it structured, unstructured or semi-structured data?
- How complex is the data and where can more information about it be found?
2.3.2 Knowledge About The Decision

Knowledge Sources are shown in DMN using a document shape containing a name that represents the Knowledge Source in question:

As the Knowledge Sources that act as authorities for decisions are identified they are added to one or more Decision Requirements Diagrams so that they can be linked as discussed in A Decision Requirements Diagram.

Decisions are driven by policies, regulations, best practices, human expertise and analytic insight. Each such Knowledge Source should be documented.

A policy document or regulatory framework can generally be modeled as a single source—there is no need to model each policy item or clause separately. Similarly, multiple regulations issued by various bodies for the same purpose, differing State regulations for example, may often be modeled as a single source. It is sometimes less clear what the right division should be for expertise and best practices but it is generally a good idea to be fairly general at this stage rather than getting too specific too soon.

If the knowledge in a Knowledge Source explains how a decision should or must be made, for example a policy or regulation, or how it might be made more accurately or effectively, for example a best practice or analytic, then it is an authority for the decision. The Knowledge Sources that impact a Decision are linked to that decision using an Authority Requirement Link—a dashed line with a circle as the arrow head. The Knowledge Source at the blunt end describes, enables or constrains the decision at the circle end.

The intent is to make it clear where the knowledge is coming from, no more.

To find a robust list of Knowledge Sources, consider

- What tells me what I must do?
- What tells me what I should do?
- What tells me what I can do?
- What tells me what I will probably do?
- What would help me do it better?

“If only we knew xyz we could make a more profitable decision”

This last category in particular will tend to identify potential analytic knowledge, analytic insight, that might be helpful. Analytic Knowledge Sources should focus on
the category or type of model or analytic such as "customer churn models" or "root cause analysis" rather than on a specific analytic model or visualization at this point. It is often the case that a cluster or group of analytic models will be developed with a common purpose and this should be modeled as a single analytic Knowledge Source rather than many.

For all knowledge Sources, it is worth documenting:

- What kind of knowledge is involved—Policy, Regulation, Expertise, Best Practice or Analytics?
- How complex is the knowledge?
- Where can additional details be found?

If a policy was originally written based on data mining or some other kind of data analysis, it should be documented as an analytic Knowledge Source.

### 2.3.3 A Decision Requirements Diagram

All of these elements should be combined into a Decision Requirements Diagram. Such a diagram contains Decisions, Input Data and Knowledge Sources along with requirements links between these objects. A diagram can be for a single top-level decision or a couple of closely related decisions.

In Figure 3 below a high level diagram shows a single Decision, Select Marketing Offer, along with two Input Data and three Knowledge Sources. The central Decision has Information Requirements linking it to the two Input Data as making it requires that information. It also has Authority Requirements to the three Knowledge Sources as these act as authorities for the decision-making, describing how it should be performed.

Another way to understand this diagram might be to read it as a sentence: We select a marketing offer by applying marketing know-how, customer churn risk and a customer’s propensity to accept offers to customer data to select from available marketing offers.
The final stage in Decision Requirements modeling is to decompose and refine the model, working until enough detail is included that the project can be successful. Obviously the level of detail required is somewhat subjective but the initial model is unlikely to be sufficient for any except the most straightforward projects. The process for this step involves:

- Decompose Decisions.
- Refine Input Data.
- Refine Authorities.
- Define Input Data for Analytics.
- Iterate.

### 2.4.1 Decompose Decisions

Besides Input Data, decisions often require information that comes from other decisions. Identifying additional decisions that produce information needed for the decision being modeled is a critical step in further specifying the decision.

You can think of these decisions as sub-decisions or pre-cursor decisions. These decisions must be made before the top-level decision(s) because the results of these decisions must be available before the top level decision can be made.
Refining the decisions required to make a decision is an iterative process. These more granular decisions are often less action-oriented than the top-level decision and more about answering questions and supplying information.

If a large number of initial decisions were defined early in the discovery process it is common for these to include decisions that are part of some other decision. When decomposing a decision, it is important to check for decisions that have already been identified on which it depends. The requirements of Decisions are a network rather than a simple hierarchy so reuse is both common and often extensive.

Figure 4 shows the same diagram as Figure 3 but with an initial set of related Decisions. In this case the project team has identified that the decision-making for selecting marketing offers relies on making sure that it is an appropriate time to make an offer, determining which product the offer should relate to, and determining how value the offer should be. The information produced by these three Decisions is also required by the main decision so it has Information Requirements to them as shown.

This analysis can and should be repeated for each decision in the diagram. This will identify additional pre-cursor decisions or parts of the decision-making. Decomposing the decisions in this way breaking them down into more granular decision-making pieces clarifies exactly how the decisions should be made.

**Figure 4: Related Decisions in a Decision Requirements Diagram**

*Source: DecisionsFirst Modeler*
Each such decision should be linked to the decision being decomposed as additional information requirements for the higher level decision. This uses the same link as before—a solid arrow—and the decision at the arrow end is said to require the information produced by making the decision at the blunt end.

These information requirements show which smaller, more granular decisions must be made before a higher level decision can be. They show dependency—the higher level decision is dependent on the results of the lower-level ones. As such the lower level decisions will be made before the higher level decision. The diagram implies nothing about the sequence of these lower level decisions: In principle they could all be made at once or in any order. If there is an order that is truly required, rather than simply an order that seems logical or efficient, then this should be reflected in the information requirements, with the decision(s) that must come first being information requirements of the decision that comes later.

It is often helpful to show the decisions at the same level in a decomposition in a particular order, especially when human decision-makers are used to doing one thing before another. Just remember that any required sequence should be reflected in the information requirements being modeled and that the spatial layout is just illustrative not reflective of an actual requirement.

For each new decision consider refining the organizational, business and application context of the decision-making. Often all the sub-decisions will share the same links in this regard as their parent and nothing need be specified. If, however, they do not then this should be documented.

### 2.4.2 Refine Input Data

Any new Decision added may require an existing or new Input Data. It is common for existing Input Data to be required by some of these new Decisions as many Input Data are required by multiple Decisions in a typical model. Add any additional Input Data and Information Requirements to your diagram.

It is also common to find that adding additional detail to the decision-making reveals that an Input Data is not required by a Decision it is currently linked to but by one or more of the Decisions it requires. Add or change Information Requirements as necessary to correctly show where each Input Data is required.

Information needed by a decision might appear to come from an Input Data source but may really come from another Decision. Some external data sources actually supply the results of decisions (is this location prone to flooding?) or analytic decisions (a credit score) rather than data. Replace Input Data with Decisions (and perhaps analytic Knowledge Sources also) as appropriate.
2.4.3 Refine Authorities

Just as new Decisions may require new Input Data, so they may also require new authorities. It may become clear that an additional Knowledge Source should be added to describe how to perform one of the new Decisions. Similarly, existing Knowledge Sources may describe how to make a new Decision. Add additional Knowledge Sources and Authority Requirements as necessary.

As more detail is added to the decision-making in the Decision Requirements Model it will become clear that the Knowledge Sources modeled actually relate to one or more of the Decisions added to the diagram. It is common that an authority originally linked to a decision is actually an authority for one or more of the sub-decisions of that decision rather than for the decision itself. Change the Authority Requirements shown on the diagram to reflect this.

It may also be discovered that some Authority Requirements actually involve a new Decision. The Authority Requirement should be replaced with an Information Requirement to a new Decision that has an Authority Requirement to the original Knowledge Source.

Additional Authority Requirements may be found by reconsidering Information Requirements. Sometimes it is not the information that is being required by a Decision but the result of an analytic that uses that information. Be prepared to insert an analytic Knowledge Source between an Input Data and a Decision, replacing the original Information Requirement with an Authority Requirement.
In Figure 5 above a more complete diagram shows that an additional layer of decomposition has been performed, identifying several additional Decisions. Some additional Knowledge Sources and Input Data have also been identified and the Customer Loyalty analytic has moved from being an authority for the top-level decision to being more specifically associated with the decision about customer loyalty used as part of determining the offer value.

### 2.4.4 Add Input Data For Analytics

In the same way that a Knowledge Source can be an authority for a Decision, so Input Data can be an authority for an analytic Knowledge Source. After all, for any analytic Knowledge Source there must be an ability to determine how to derive the analytic. Input Data are authorities for analytic knowledge as they are analyzed to produce it. Add any additional Input Data that the analytic is likely to require and link it to all the Input Data expected to be part of producing the analytic using Authority Requirements links.

It is also possible, though less common, for Decisions to be authorities for Knowledge Sources. For instance, you may have to decide if an existing analytic is still sufficiently predictive or requires updating or have to decide whether to re-evaluate a risk policy. Such tactical decisions can be modeled also and can act as an authority for a Knowledge Source such as a risk score or risk policy.

### 2.4.5 Repeat As Necessary

This iterative refinement can be repeated as often as necessary to flesh out a complete, coherent and useful Decision Requirements Model. A couple of things to note during this process:

- If the diagram being developed becomes cluttered or overly complex, consider developing multiple diagrams as views on the same overall model. Each diagram can show some of the requirements involved. In general, a high-level diagram with just the top layer of decisions and several diagrams showing how that layer of decisions is decomposed works well. Similarly diagrams that only show information requirements can be helpful as they remove a great deal of complexity if there are many Knowledge Sources.
- Don’t feel that every part of the model has to be equally detailed. It is common for some parts of the model to have more layers of decisions than others either because the decision-making is just more complex or because there is more value in understanding it in detail.
- Annotate diagrams thoroughly as they are built. Not everything discovered while building the model can necessarily be shown on the diagram formally so capture this as notes.
Keep showing the diagrams to those who understand the business, who makes or owns the decisions, to ensure the view of the decision-making remains accurate as more detail is added.

When to stop iterating and refining is always an interesting question. Good Decision Requirements Models are unambiguous and make it clear how decisions will be made, breaking down the decision-making into atomic elements that are easy to describe and understand. In practice this remains subjective.

Decision Requirements Diagrams can be used as requirements for a wide range of projects from business rules to predictive analytics, and from complex automated decisioning to dashboards and BI. Using a Decision Requirements Diagram in a requirements approach is described in the next section.
3 Using Decision Requirements Models

Once a Decision Requirements Model is developed the requirements for a decision will be accurately specified. The best way to use such a model is as a shared specification of decision making in a collaborative environment such as DecisionsFirst Modeler. Decision Requirements Models can also be used to generate requirements documents as part of a formal requirements process. DMN allows for them to be extended to include decision logic for an executable model. Finally, it is also possible to use these models as a map to implementation environments.

3.1 Generate a Requirements Document

Once a Decision Requirements Model that is a complete and accurate representation of the decision-making involved has been developed it can form the basis for a requirements document. This document should include the following information derived from the Decision Requirements Model:

- **Objectives To Be Impacted**
  The decisions involved in a project are those identified explicitly as being involved and those that they require, directly or indirectly. From this it is possible to determine which business objectives will be impacted by the project—all the objectives impacted by any of these decisions. For each such objective the project should describe the expected or desired impact on that objective. How will the project impact that objective going forward and how will this be measured and managed?

- **Decisions Impacted**
  The decisions involved should be identified and their descriptive information provided. Each such decision could have its own set of requirements, showing what other knowledge might be involved for instance. Each may also be required by other, higher-level decisions and require other, lower-level decisions. It is often useful to present the information on decisions impacted in two parts.

  - In terms of the decisions planned for implementation in business rules and their information and authority requirements. The view "down" the hierarchy.
  
  - In terms of the context of those decisions, how those decisions fit in the broader decision-making. This is the view up the decision hierarchy.

Instead of packaging up a model as a specification of the decision-making you need DMN allows you to extend the model with decision logic, in the form of Decision Tables and Business Knowledge Models for instance, to create an executable model of the decision. See Extending Decision Requirements to Decision Logic for details.
Sources of Business Rules
The Knowledge Sources that must be analyzed and harvested to find the required business rules should be described. Challenges of access, for example to busy SMEs, or of complexity, say legal documents, should be documented. The relationships of critical Knowledge Sources to the decisions they influence should be clear.

Analytics Required
The analytic or analytics that are the focus of the project should be described along with the information being analyzed to produce them, their authority requirements. Additional detail will be required here on likely techniques, risks and issues common to the kind of analytic in question etc.

Application and Organization Context
The organizations that own the decisions impacted, directly or because they own a parent decision, will likely have to approve and manage any business rules developed and should be listed along with their ownership role. Other organizations may care about the decisions being impacted and should be kept informed. Those organizations that make the decisions involved, directly or because they make a parent decision, will need to understand the rules and the explanations the rules give for decisions. The document should include a plan for ensuring this. The business processes and information systems that will have to contain the decisions should be listed. Any business rules developed will have to be deployed in a way that allows them to be used in this application context and that may constrain deployment approaches.

All projects will have additional details such as phases and scope, sponsorship and organizational change requirements. These are easier to specify precisely with reference to a Decision Requirements Model. While these will need to be added to the document this should be done using the Decision Requirements Model as the touchstone of the project’s intent.

3.2 Extending Decision Requirements to Decision Logic
To implement a Decision Requirements Model, the decision logic or business rules behind it must be specified. This is a very large topic in its own right, however some discussion is appropriate both in terms of its coverage in DMN and because decision logic modeling can be useful even when specifying requirements.

Business rules are managed assets or groups using a wide variety of constructs or metaphors including:

- Decision Flow or Rule Flow
- Decision Tables or Rule sheets
- Decision Trees
- Rulesets
DMN focuses on Decision Tables as a way to represent decision logic or business rules, defining a standard notation for them. It should be noted that the standard also acknowledges that there are other representations for business rules or decision logic and that some elements of a Decision Requirements Model might be better represented by, for instance, a predictive analytic model rather than business rules.

DMN allows a Decision to be represented directly by a decision table or linked to Business Knowledge Models (see below) to support reusable decision logic.

Decision Tables have value even before the transition to decision logic modeling. When specifying requirements, it is possible to model the decision logic for more stable and more central decisions to provide detailed requirements. In addition, more changeable decisions can be more fully described if they are represented by exemplar rules and decision table structure. Decision Tables need not be focused only on automation either as they can be a valuable tool to define how to make decisions made manually.

### 3.2.1 Decision Tables

The decision logic for each decision can be specified using a decision table. This involves:

- Defining the Information Model for involved in each Information Requirement. Describing the structure of the information represented by Input Data and specifying the structure of information returned by a Decision.
- Identifying information used in rules. Not all the properties identified or Input Data will be used in a Decision.
- Writing decision logic, business rules, to generate each of the allowed answers. The table cannot produce an action, an answer, that’s not in the list of allowed answers. Similarly, conditions that use the data from other decisions should only check for allowed answers in those dependent decisions.
- Validating and cross-checking logic if it is intended to be complete. As noted, Decision Tables allow initial and detailed decision logic to be written and if a particular decision table is intended to be complete it must be validated.

Figure 6 shows the basic elements of a decision table. Tables are named and a series of input properties identified from the specified Information Requirements. Rules are written that involve checking expressions against some or all of these properties. If all the expressions in a rule evaluate to true, then the relevant action is selected as the result of the decision.

Decision tables come in several styles as described below. Rules can be numbered when shown in rows or columns for reference. The allowed values can be specified for input and output expressions,
based on the allowed answers of decisions or allowed values specified as part of the information model. Cells can be blank, where blank inputs mean “don’t care” and blank outputs mean “do nothing”. Multiple outputs are supported, to allow explanations to be added to answers or structured answers to be populated. The double lines matter (separating labels and structure from rule content) but the colors don’t. Cells can be merged like in Excel.

Tabular decision tables represent a set of rules in a tabular layout. These are typically but not always exhaustive and exclusive. DMN defines two formats and various hit policies for these tables. Decision tables can have rules as rows with columns representing properties and actions (Figure 7) or rules as columns with rows representing properties and actions (Figure 8).

Decision Tables can be single hit, returning only the result of a single rule, or multi-hit. To specify how the table is processed a variety of hit policies are allowed. Single hit tables can be Unique (defined such that only a single rule can be matched to any given set of data) or Any (defined such that if multiple rules match a given set of data they all have the same output so it does not matter which rule fires).

Multiple hit tables can either just Collect the results of all the rules that fire, sum them (C+) if the output is numeric or count them (C#). A decision table that handles every possible combination of Input Data is considered exclusive also.

Technically DMN also allows single hit tables to select a rule based on the first one that hits in table that matches the data or using a priority setting. Multi-hit tables can have rule or output ordering in the results. None of these approaches are recommended as the result is a decision table where the output might change if the table was reorganized even though the rules themselves were not changed. This makes business user editing of the tables impractical and so should be avoided.
Decision tables can also be shown as cross-tabs or look up style tables such as Figure 8. All cross-tab Decision Tables are unique and exclusive so no hit policy need be specified. Multiple attributes can be shown on each axis if desired.

### 3.2.2 Decision Table Syntax

The syntax available for decision logic in DMN is quite extensive but decision tables have a relatively simple subset defined:

- **Data Types**
  - Number, String, Boolean, Day Time duration, Year Month duration, Time or Date.

- **Arithmetic operators** +, -, *, / and **(exponent)** are supported.

- **Comparators** for numbers include =, Not, <, >, <= and >= with = and Not supported for strings.

- **Ranges** can be specified and are used to check to see if the property’s value is in the range specified. Square brackets - [ ] – should be used as this shows that the values specified are included in the range. E.g. both 2 and 6 are included in [2..6]. () and [ ] can be used to define ranges with the specified values excluded from the range but this should never be done as it totally confuses non-technical reviewers.

---

Decision Logic—business rules—is found in Knowledge Sources and specified in Decision Tables. In this regard Knowledge Sources are documentation while Decision Tables are (potentially) executable. The content of a Knowledge Source has to be analyzed to determine the decision logic for decision tables.
3.2.3 Information Model

As part of defining decision logic it will be necessary to define an information model to support the Decision Requirements Model. Each Input Data’s information model must be defined and the structure of the information represented by each Decision’s Allowed Answers must likewise be defined. These information models can be defined in class diagrams, entity diagrams or fact models as long as it specifies the fields involved, their structure and data types along with allowed and default values.

3.2.4 Business Knowledge Models

DMN defines a fourth shape and a third link type to manage reusable decision logic. Sometimes (though rarely) in a model there are multiple decisions that need the same logic because they have different (but compatible) Input Data. For instance, the decisions Validate Invoice Address and Validate PO Address are different because their information requirements are different but their decision logic should be the same.

Figure 10 shows the shape of a Business Knowledge Model and a Knowledge Requirement to show which Decisions are supported by which Business Knowledge Model.

To use Business Knowledge Models, you can specify the parameters for the Business Knowledge Model and then define Decision Tables for each that use these parameters as input properties. You can link Business Knowledge Models to Decisions using Knowledge Requirements links. If only one Business Knowledge Model is linked to a Decision, then you need only specify how the Information Requirements of the Decision map to the Business Knowledge Model’s parameters. If you are using several required Business Knowledge Models for a single Decision, or using one Business Knowledge Model to support another, then the specific logic for doing so must be specified. For clarity of documentation, Knowledge Sources can be authorities for Business Knowledge Models and are linked to them using an Authority Requirement.

When decision logic is reusable the Decisions themselves are almost always reusable also. It is far preferable to reuse the Decision than to create two Decisions and link them to a shared Business Knowledge Model.
3.3 Using Decision Requirements Models as an Implementation Map

An alternative approach exists for implementing a Decision Requirements Model. Instead of specifying the decision logic in the model, the model can be linked to components developed in your implementation environment.

For instance, where the decision requirements are best implemented as decision logic or business rules, the model can be linked to Business Rules Management Systems. These contain decision logic in well-defined artifacts, not only decision tables but also decision trees and more. The decision logic for a specific decision can be specified in the BRMS and then linked to the model 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 11 below shows an example, linking a decision in a Decision Requirements Model to a decision table implemented in the open source rule engine JBoss Drools. This is the approach taken by DecisionsFirst Modeler Enterprise Edition and more information can be found in Appendix C: DecisionsFirst Modeler.

Figure 11: Linking Decisions to BRMS implementations

Source: DecisionsFirst Modeler

Decision Requirements Models can also be linked to the predictive analytic models developed in data mining and other advanced analytic tools. The model shows the
information from which the analytic was derived as well as how it is used in decision making. Linking the Decision Requirements Model directly to the predictive analytic model allows business users to see where analytic models are used and ensures these analytic models are kept in a business context.

Decision Requirements Models can also be linked to other analytic implementations such as visualizations, reports or dashboard components. Where the decision making being modeled is manual, this ties the BI and analytic components being developed more explicitly to best practice decision-making approaches. It allows those recording and managing these best practices to see what analytic components are available to support them.

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 managed and edited, preventing duplication and confusion.
- It takes advantage of the increasingly sophisticated editors available in a modern platform to validate, verify, test and simulate decision-making.
- It allows the reuse of existing assets in Decision Requirements Models.
- It allows a single Decision Requirements Model to be linked to multiple implementation tools. Elements of a single Decision Requirements Model might be implemented in several commercial BRMSs while also leveraging a predictive analytic model built in an analytic tool and visualizations to assist in the manual elements of the decision.
4 Conclusions

Defining decision requirements as part of your overall requirements process offers many benefits:

- Modeling decisions makes business processes less complex, more robust in the face of change, and easier to manage.
- A Decision Requirements Model provides the needed structure for the implementation of a Business Rules Management Systems (BRMS), supporting iteration and agile development.
- Framing data mining and predictive analytics projects with a Decision Requirements Model links analytics to business results and helps ensure successful deployment.
- Understanding the decisions relevant to a dashboard or decision support environment structures knowledge and puts a premium on taking action.

Decision Requirements Models are a common language across business, IT and analytic organizations improving collaboration, increasing reuse, and easing implementation. Decision modeling has four steps that are performed iteratively:

1. Identify Decisions.
   Identify the decisions that are the focus of the project.

2. Describe Decisions.
   Describe the decisions and document how improving these decisions will impact the business objectives and metrics of the business.

3. Specify Decision Requirements.
   Move beyond simple descriptions of decisions to begin to specify detailed decision requirements. Specify the information and knowledge required to make the decisions and combine into a Decision Requirements Diagram.

4. Decompose and Refine the Model.
   Refine the requirements for these decisions using the precise yet easy to understand graphical notation of Decision Requirements Diagrams. Identify additional decisions that need to be described and specified.

A Decision Requirements Model can be used to generate requirements document as part of a formal requirements process, extended to include decision logic for an executable model or used as a map to implementations in analytic and business rules technologies.

For more information on decision modeling, check out http://decisionmanagementsolutions.com/decision-modeling.

The appendices discuss the kinds of decisions for which decision modeling is appropriate, the different techniques available for discovering decisions and DecisionsFirst Modeler.
Appendices

Appendix A: About Decisions

Organizations make decisions of various types as shown in Figure 12.

They make infrequent but large impact strategic decisions. These one-off decisions typically involve large numbers of people and large investments in time and money. Much analysis is done before the decision is made and the implications for a business can be dramatic.

Regular tactical decisions involving management and control are also made. These are less impactful and there is generally still time and energy for significant analysis but there is time pressure too, a need for consistency and the opportunity to learn what works.

Finally, every organization makes large numbers of operational decisions about individual transactions or customers. Time pressure is often extreme and these decisions must generally be embedded into operational systems and processes.

One aside. Some executives and managers are confident in their own "gut" decisions and resist the use of analytics to improve these decisions or business rules to automate them. Such resistance need not derail projects however as these same executives are often very willing to consider analytics and business rules that will improve and control decisions made by junior staff. In other words, there may be far more support for business rules and analytics in a department’s operational decisions than in the tactical decisions made by those who run it.

Suitable Decisions

Decision Requirements Modeling can be used for any decision. Because Decision Requirements Models take some time and energy to build most organizations will only do so when a number of things are true of the decision:

Action oriented

As noted above, the essence of decision-making is to select from an array of possible actions, pick one and then take it. Describing a suitable decision as "action-
oriented" may therefore seem redundant but some decisions are more about getting an answer than about taking action. The top-level or target decision in any project, however, will generally be action-oriented.

**Value in defining**

There needs to be some value in defining how the decision will be made in advance. This is clearly true for decisions that are made many, many times as knowing this improves consistency and makes it easier to learn what works, share best practices and so on. Any decision being considered for business rules is likely to meet these criteria. However, it can also be true for one-off decisions if they are complex and if the decision-making approach is a point of contention or needs to be transparent. Some combination might drive value also—the combination of being moderately complex and repeating occasionally could make it worth building a Decision Requirements Model.

**Non-trivial**

If a decision is truly trivial then there is no value to modeling decision requirements. If many policies or regulations apply, if there is a wide range of options to select from or lots of data to consider then the decision is likely to be non-trivial and so worth modeling. If the way a decision is made must change often, if it is very dynamic, or if there is a mix of drivers combined with a more modest pace of change then that will also make the decision non-trivial. Almost any decision that involves an assessment of fraud, risk, customer opportunity or similar through the analysis of historical data is non-trivial.

**Measurable**

The value of the decision must be measurable and should be definable in advance. We should be able to identify the KPIs and metrics that will be improved by a better decision or weakened by a poor one. Defining this in advance allows us to baseline our current decision performance and so shows the value our investment in improving the decision. It can also be critical in understanding what a good decision even looks like.

**Types of Top-level Decisions**

Most organizations have many candidate decisions and the use of decision words such as determine, validate, calculate, assess, choose, select and, of course, decide gives us a clue. For instance:

- Determine if a customer is eligible for a benefit.
- Validate the completeness of an invoice.
- Calculate the discount for an order.
Assess which supplier is lowest risk.
Select the terms for a loan.
Choose which claims to Fast Track.

We can categorize top-level decisions into various types, though some decisions include characteristics of several types. For instance:

- **Eligibility or Approval**—Is this customer/prospect/citizen eligible for this product/service?
  These are made over and over again and should be made consistently every time. The use of a business rules-based system to determine eligibility or to ensure that a transaction is being handled in a compliant way is increasingly common. These decisions are policy and regulation-heavy and the use of a Business Rules Management System to handle all the business rules is very effective. While eligibility and compliance decisions can seem fairly static, changes are often outside of the control of an organization and can be imposed at short notice.

- **Validation**—Is this claim on invoice valid for processing?
  Validation decisions are almost always operational, they are overwhelmingly rules-based, and the rules are generally fixed and repeatable. Validation is often associated with forms and online versions of these forms are of little use without validation. The move to mobile apps makes validation even more important.

- **Calculation**—What is the correct price/rate for this product/service?
  Calculations are usually operational and they are overwhelmingly rules-based. The rules are generally fixed and repeatable but making them visible and manageable using business rules pays off when changes are required or when explanations must be given. Sadly calculations are often embedded in code.

- **Risk**—How risky is this supplier’s promised delivery date and what discount should we insist on?
  Making a decision that involves a risk assessment, whether delivery risk or credit risk, requires balancing policies, regulation and some formal risk analysis. The use of business analytics to make risk assessments has largely replaced “gut checks” and predictive analytic models allow such risk assessments to be embedded in systems.

- **Fraud**—How likely is this claim to be fraudulent and how should we process it?
  Fraud detection generally involves a running battle with fraudsters, putting a premium on rapid response and an ability to keep up with new kinds of frauds. Managing the expertise and best practices required to detect fraud using business rules gives this agility while predictive analytics can help with the kind of outlier detection and pattern matching that increases the effectiveness of these systems.

- **Opportunity**—What represents the best opportunity to maximize revenue?
  Especially when dealing with customers, organizations want to make sure they are making the most of every interaction. To do so they must make a whole series of opportunity decisions such as what to cross-sell or when to upsell.
These decisions involve identifying the best opportunity, the one with the greatest propensity to be accepted, as well as when to promote it and where. A combination of expertise, best practices and propensity analysis is required.

- **Maximizing**—How can these resources be used for maximum impact?
  Many business decisions are made with a view to maximizing the value of constrained resources. Whether it is deciding how best to allocate credit to a card portfolio or how best to use a set of machines in a production line, deciding how to maximize the value of resources involves constraints, rules and optimization.

- **Assignment**—Who should see this transaction next?
  Lots of business processes involve routing or assignment. In addition, when a complex decision is automated it is common for some percentage to be left for manual review or audit. The rules that determine who best to route these transactions to and how to handle delays or queuing problems can be numerous and complex, ideal for managing in a Decision Management System.

- **Targeting**—What exactly should we say to this person?
  In many situations there is an opportunity to personalize or target someone very specifically. Combining everything known about someone with analytics predicting likely trends in their behavior and best practices, and constraining this to be compliment with privacy and other regulations, individuals can feel like the system is interacting only with them.
Appendix B: Decision Identification Techniques

Examine Business Processes

Most, if not all, business processes require decisions to be made: claims must be approved or rejected before a claims process can complete, cross-sell offers must be selected and product discounts must be calculated before an order to cash process can complete and so on.

Explicitly modeling the decisions that happen in your business process ensures that the process model is closer to reality and identifies opportunities for business rules and analytics in the business process. For most organizations this is the critical approach, the one most likely to identify suitable decisions.

There are two main clues when looking for decisions in business processes:

- When a business process must handle multiple scenarios, modeling the decision-making in that process using only gateways and branches can become very complex. Nests of gateways with few intervening tasks often represent decision-making modeled in a business process. Decision modeling replaces the nest of gateways with a single, explicit decision point—a Decision Task\(^1\). This clarifies the behavior of the process, makes it easier to see if the process or the decision must change, and allows for changes in the decision-making approach to be independent from process change.

- Process instances often wait while items are put on task lists or in queues. Unless data is being added manually and even sometimes when it is, the reason for putting a process instance into a holding pattern in this way is generally because a decision must be made. Identifying this as a decision and automating it keeps transactions moving so only exceptions end up on task lists or in an inbox.

Even if you decide to keep the decision manual and retain the work queue there is real value in modeling and understanding the decision itself to assist in training, ensure consistency and identify areas for improvement.

Examine performance measures

Another approach is to look at the Key Performance Indicators (KPIs) and metrics that the business has in the area(s) under consideration. Any KPI or metric is valuable only if it helps motivate suitable behavior, implying that someone’s actions can change the value of that KPI or metric. By investigating KPIs and metrics, and finding out when and where people make choices that move KPIs/metrics up or

\(^1\) The BPMN standard refers to these as Rule Tasks. We prefer Decision Task as the task may involve analytics, for instance, not just business rules.
down, a project team can identify decisions. Each opportunity for choice-making, for selecting an action from a possible set of actions, is a decision.

To begin the team can simply ask what decisions make a difference to a KPI or metric but they may find they have to ask business experts to walk through their day or week keeping the KPI in mind to find all of them as people may not think of the choices they make as explicit decisions.

Examine business intelligence outputs

Most business organizations use Business Intelligence (BI) tools to generate a lot of reports, build dashboards, provide queries and dump data into Excel for analysis. While some of this can be justified by a need to monitor what is going on, most of it is intended to improve decision-making. As a result these BI outputs can be used to find a list of decisions being made.

- When someone says they get a particular report or look at a dashboard the team can ask "then what?" to see what action might be taken as a result.
- Users of each report or dashboard can be asked when they refer to it, what they are doing when they use it.

By listing the possible actions that can be taken when the report is read they can scope out decisions being made as a result of the reports content. By considering those actions taken or discussed with the report in hand, where the report might be consulted, they can find others. Especially when users can describe ways to identify sets or types of transactions where they "always do X," decisions that might be suitable for automation are involved.

Examine Legacy Systems

When legacy systems are being modernized, or enterprise applications extended, decisions suitable for automation with business rules can be found in several ways.

- Those modules or components that are frequently changing and where change requests are made regularly, or that have a long backlog, are often decision-making components. This is because decision-making is highly agile, changing regularly in response to external stimuli like new policies, regulations, competitors’ pricing changes and much more.
Modules that generate lots of exceptions or referrals, especially those where far more are being generated now than in the past, are also often decision-making modules. Exceptions get referred to people so they can make a decision the system cannot. This naturally implies the module in question is making decisions the rest of the time.

Any place in an existing system where users have lots of yellow sticky notes on their screens or other ways to keep notes about navigating and using the system is worth checking. Pricing and discount notes, specific instructions for a customer and much more are often recorded outside the system because the decision-making coded in it is too hard to change.

**Brainstorm**

The decisions involved can often be listed in a straightforward brainstorming session with the business owners. This is generally a top-down approach that begins with the more senior staff in a business area.

The first decisions identified may be those strategic or tactical decisions made by executives or management. These are not generally a good target for decision modeling as they don’t repeat often enough. As the conversation continues, however, it is often possible to get to operational decisions by asking managers and executives what decisions must be made at the front line.

Techniques such as asking what decision-making training and authority is given to front line staff, determining which decisions have published guidelines, and asking managers what decisions sometimes get referred to them, implying front line staff make most of them, help identify operational decisions.

> This can be a weak approach, only useful when supporting materials required by other approaches are not available. Nevertheless, it has political value, helping to build buy-in, and so should be included in most projects for this reason if no other.

Not all decisions are considered in the current state. Often there are places where everyone or everything is treated the same way. Parts are moved from the warehouse based on the same logic no matter what the part, all customers see the same information on the website, or all orders get the same discount. The organization is doing the same thing every time without considering alternatives.

The team should document these as **Micro Decision** opportunities. A Micro Decision is one where each transaction, each opportunity for a decision, results in a specific decision made about that transaction. When micro decisions are being made, the organization focuses on each opportunity to make a decision as a unique opportunity to improve results. Generally, it is analytics that provide the insight needed for differentiation and business rules that make it happen. In each case the team can ask why something is always done the same way and in what circumstances might it be done differently?
Appendix C: DecisionsFirst Modeler

All of the examples in this document have been produced using DecisionsFirst Modeler. DecisionsFirst Modeler is a social, collaborative platform for Decision Requirements Models available from Decision Management Solutions.

Figure 13: DecisionsFirst Modeler

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 such as RedHat BRMS and IBM Operational Decision Manager.

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

For more information, go to http://decisionsfirst.com.

Figure 14: Example Integration

Source: DecisionsFirst Modeler Enterprise Edition
References


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