This article is unique because it is a bit embarrassing, but in a respectable way. When our book introduced The Decision Model (TDM) to the public, client organizations already had several years of experience with it. However, mistakenly we never intended it for Data Quality incentives. In fact, we advised strongly against using TDM for Data Quality (DQ) as stated below:

“Business logic best targeted for a decision model is of a purely business nature …..Logic of a purely business nature does not include logic for determining data quality…..Most often data validation rules are not the optimum target for a decision model. One reason is that they do not arrive at a business-oriented conclusion. Rather, data validation rules ensure that data meets certain requirements so that it is of good quality……however, the data validation rules themselves are typically not part of the decision model, but external to it.”

**TDM and Data Quality Come Together**

It is no surprise that data quality is critical to business-oriented decision models. After all, good decision models operating with bad data are no better than bad decision models operating with good data. The surprise is: not only are decision models a preferred way for managing true business logic but they are remarkably suitable for managing data quality logic!

Despite the book’s quote above, some clients created decision models for managing data quality logic. These initiatives have been extremely successful. The success is due to two factors: decision models represent DQ logic in readable form and they enable active business governance over it. Further, the governance life cycle extends beyond the creation of initial DQ decision models. It continues to the detection of logic errors and execution against test data. This means that data stewards deliver DQ decision models for deployment that are fully validated against decision model principles and tested.

---

The Unanticipated Role of The Decision Model in Data Quality

The Intensifying Need for Data Quality

These successes are noteworthy because data quality remains a growing priority among organizations worldwide. For example, the EDM Council, created by leading financial industry participants has initiated a DQ Working Group (http://www.edmcouncil.org/sec_data_quality.asp). The MIT Chief Data Officer (CDO) and Information Quality Symposium July 17-20 had record attendance with registrants from 14 countries (http://mitiq.mit.edu/CDOIQ/2012). A presentation on TDM introduced the role of DQ decision models for Data Quality at the symposium. To hear a recording of TDM and Data Quality go to http://www.screencast.com/t/5DMaJXEzuXd.

This article explains why and how DQ decision models provide Chief Data Officers, business data stewards, and data professionals with a new weapon in the ongoing battle for higher data quality. It assumes the reader is familiar with TDM. Readers new to it will find an introduction specifically for data professionals at “The Decision Model: Bottom Up” (http://www.tdan.com/view-featured-columns/11671)

TDM’s Vital Role in Data Quality

It is first important to distinguish between true business logic that inspired the creation of The Decision Model and data quality logic (DQ logic).

DQ Logic versus Business Logic

DQ logic is the logic applied to data elements to come to a conclusion about data validity. Data Quality frameworks divide data validity into various data quality dimensions, such as data completeness, reasonableness, accuracy and so on. So DQ logic for these dimensions ensures that data is of acceptable quality. For example, it is DQ logic that determines whether or not a particular instance of an address is a valid address.

Business logic, on the other hand, is the logic that comes to a conclusion about business-oriented judgments. We can divide business-oriented judgments into various kinds, such as eligibility judgments, compliance judgments, and so on. So, business logic for these kinds of judgments ensures that decisions using data (of acceptable quality) result in conclusions of acceptable quality. For example, it is business logic that determines whether or not a particular valid address is in a neighborhood eligible for a particular marketing campaign.

Because both DQ logic and business logic are logic, each reduces to an atomic state of conditions leading to a conclusion. This means each fits naturally into the format of a decision model even though each serves a different purpose.

Improved Process Models

Historically, The Decision Model solves the shortcomings of business rules approaches. That is, with the introduction of The Decision Model, decision-aware process models contain tasks driven by whole decision models rather than by lists or groups of individual business rules.

However, with the introduction of DQ decision models, an improved approach to decision-aware process models now allows for tasks driven by whole DQ decision models rather than DQ logic scattered in various places. This improvement not only separates process and logic, equally important to data professionals, separates one kind of logic from another.
The Unanticipated Role of The Decision Model in Data Quality

Business Processes with DQ Decisions

Figure 1 illustrates two ways to include decision models for various DQ dimensions within a business process.

Example 1...

![Figure 1: Including DQ Decision Models in a Process Model](image)

All task names in Figure 1 contain the letters “DQ” indicating that the tasks determine the validity of a Data Quality dimension. The first task contains the word “complete” because its decision model comes to a conclusion about the DQ dimension of completeness – specifically, the decision model determines whether or not all required data is populated. If it is populated, this process continues with DQ decision models for conclusions about data accuracy, consistency and reasonableness. It executes these in parallel.

In Example 2, like Example 1, the first task determines whether or not all required data is populated. However, in this case, the process then continues with DQ decision models for other dimensions in a specific sequence.

In both examples, the circles at the end of the process confirm that all DQ decision models execute prior to the subsequent parts of the process model that actually use that data.

What Do DQ Decision Models Look Like?

Figure 2 contains a DQ decision model. According to the name in its decision icon, this decision determines whether the data for an Insurance Claim satisfies the DQ dimension of completeness. That is, are all required data fields in an Insurance Claim populated?
The Decision Rule Family has four conditions between the solid and dotted lines and each condition name ends with the word “complete.” So, each of these conditions is a completeness conclusion in a supporting Rule Family.

The supporting Rule Family on the far left comes to a conclusion for “Insurance Claim Type DQ Complete” which determines whether or not all data required for Insurance Claim Type is populated. It has input only the Insurance Claim Type. The second supporting Rule Family does the same for “Auto License Plate Number DQ Complete.” It has as its input a value for Insurance Claim Type and one for Auto License Plate number.

To understand the logic behind these Rule Family structures, Figure 3 contains Rule Family tables. The top Rule Family table comes to a conclusion about whether the data in an Insurance Claim is complete or not.
The Unanticipated Role of The Decision Model in Data Quality

In Figure 3, the first row concludes that the Insurance Claim data is complete if Insurance Claim Type is complete, Auto License Plate Number is complete, Insurance Claim Amount is complete, (and Fact Type n is complete). The remaining rows simply conclude that the Insurance Claim data is not complete if any of those data elements does not meet its completeness requirement.

The individual completeness tests for each data element are found in the supporting Rule Family for each one of them. Examining the supporting Rule Family on the left, it comes to a conclusion that the data element “Insurance Claim Type” is complete if it is populated. If not populated, the conclusion is that it is not complete, that it does not meet its completeness requirement.

Examining the supporting Rule Family on the right, Auto License Plate Number must be populated if the Insurance Claim Type is “Auto,” but otherwise must not be populated. So, this Rule Family table concludes in the first row that the Auto License Plate number is complete if it is Insurance Claim Type of “Auto” and License Plate Number is populated. It also concludes in the fourth row that if Insurance Claim Type is not “Auto” and Auto License Plate Number is not populated, Auto License Plate Number is actually complete, as it is irrelevant. The second and third rows conclude that Auto License Plate Number does not meet the completeness requirement if it is populated and Insurance Claim Type is not for “Auto” or if it is not populated and the Insurance Claim Type is, in fact, “Auto.”

What about another data quality dimension, consistency? The decision model for consistency in Figure 4 determines if the four data elements meet their respective consistency requirements. Therefore, in the Decision Rule Family, the condition names below the solid line and above the dotted line end with the word “consistent.” Each of the four supporting Rule Families, checks the relationship of one of the data elements against whatever that data element needs to relate to if it is to pass its consistency requirements.
The Unanticipated Role of The Decision Model in Data Quality

The left-most Rule Family in the bottom level has as input values the Insurance Claim Date and Incident Date. The Rule Family content for this Rule Family therefore contains logic that tests the relationship between these dates. If the relationship is acceptable, the Rule Family concludes that the Insurance Claim Date passes its DQ dimension of consistency.

The top Rule Family will evaluate to true only if all of these supporting Rule Families conclude that each of these data elements passes its consistency requirements.

The Ultimate DQ Decision Model

Figure 5 illustrates another way to build the process around DQ decision models. Like Example 1 and Example 2 in Figure 1, this process model contains one task to check for DQ completeness. However, unlike Example 1 and Example 2, the process model has only one more DQ task, the one circled, referencing one decision model. This one decision model checks every single DQ dimension of every single data element. If all is good, the process continues successfully onto the tasks that use these data elements. In decision-aware process models, such tasks are usually those that use these data elements as input to decision models.
The Unanticipated Role of The Decision Model in Data Quality

Figure 5: Another Way to Include DQ Decision Models in a Process Model

Determine Insurance Claim DQ Values is a super Decision that contains the conclusions for Consistency, Reasonableness, and Accuracy

Figure 6: DQ Decision Model for Process Model in Figure 5
The Unanticipated Role of The Decision Model in Data Quality

The Decision Rule Family at the top checks three DQ dimensions as illustrated by the conditions between the solid and dotted line. One condition name ends with the word “consistent” another with “reasonable” and the other with “accurate”.

There is one branch for each of these DQ dimensions. The left most branch checks consistency, the middle one reasonableness and the last one accuracy.

The left most branch has a Rule Family for each data element that must pass consistency logic. The second branch does the same for reasonableness logic and the third for accuracy logic.

Imagine if Figure 6 were to address DQ logic for many data elements! It would become quite large in terms of the quantity of Rule Families. But keep in mind that not all DQ logic must be part of one super decision model. Dividing them into individual decision models is appropriate depending on related business processes and reusability of DQ logic. Also, while DQ decision models tend to have a lot of Rule Families, the logic is usually fairly simple compared to business logic for compliance or eligibility business decisions.

Here is the good news. Practice proves that business people who function as data stewards can understand DQ decision models. These data stewards validate that the logic meets decision model integrity principles, change it as needed, and also test the decision models against real data.

The Holy Grail in DQ: Governance

For decades, business rule advocates have promised business rule authoring and management capabilities for business stewards. Yet, business rules approaches and technology fall short. They lack two critical aspects to close the gap between business-friendly specification of logic and its automation to target platforms in a seamless fashion.

The first critical aspect is that the representation must be simple and rigorous yet intuitive and self-correcting to minimize human error. It is not by accident that business people can easily interpret and change decision models. The Decision Model was developed from its inception, for business, not technical, audiences. Despite its rigor, it deliberately lacks technical aspects unrelated to business governance of logic. (Technical aspects become relevant for technical professionals only after business stewards achieve final approval for their logic). This is unlike other approaches which started first with target technology aspects and grew slowly into semi-successful business friendly interfaces. Some of these still require technical artifacts (e.g., data, object, or fact model) prior to expressing or modifying logic. This unnecessarily slows down the whole stewardship process often to the point of failure.

The second critical aspect is support for automated but customizable governance workflows. These workflows link business stewards and technology professionals in a continuum that works comfortably for both. With a Business Decision Management System (BDMS)², decision models become a natural asset for governance by business people and for deployment by technical experts. This is extremely important for Data Quality. For example, there can be one governance workflow for a particular business initiative and its business-oriented decision models. And there can be another governance workflow for the related (possibly reusable) DQ decision models. This separates business decision stewardship from business data stewardship in a precise and orderly manner.

²Sapiens DECISION (http://www.sapiensdecision.com/) is the first product we recognize as a BDMS. A BDMS supports automated support for all principles of The Decision Model as well as advancements, such as Decision Model Views, Decision Model testing, and Governance Workflows.
Lessons Learned

Today, there are as many DQ decision models as there are business-oriented decision models. Some organizations create and manage both. There are three lessons learned from current adopters of The Decision Model for DQ logic.

Lesson 1: Do not mix DQ logic and business logic in the same decision model.

Mixing DQ logic and business logic in the same decision model creates unnecessary complexity in the decision model. It also becomes impossible to deliver shareable DQ decision models and shareable business decision models. Mixing DQ and business logic in the same decision model also makes governance difficult, if not unattainable. That's because the people who are the stewards over DQ logic are not necessarily the same people who are stewards over the decisions that use that data.

Lesson 2: Be sure that DQ decision models execute before decision models of business logic that use that data.

When creating process models, make sure that a decision model of business logic never executes until all of the data it needs is available and of acceptable quality.

Lesson 3: Ignore anything anyone says about not using decision models for Data Quality endeavors!
Larry Goldberg has over thirty years of experience in building technology based companies on three continents, and in which the focus was rules-based technologies and applications. Commercial applications in which he played a primary architectural role include such diverse domains as healthcare, supply chain, and property & casualty insurance.


Contact us
For more information please visit us:
www.sapiensdecision.com
info@sapiensdecision.com
+ 1-800-858-9473 (US)
+ 44 1895 464 000 (UK)

Published in: www.tdan.com on Sept 1, 2012