## Kimball Design Tip #33: Using CRM Measures As Behavior Tags

By Ralph Kimball

Let's describe a simple classic example of assigning behavior tags to complex patterns of micro transactions arising from our customer facing processes like call centers, web site visits, delivery systems, and payment reconciliation systems. We'll use our standard data warehouse reporting techniques to summarize three customer behavior metrics: recency, frequency, and intensity. Recency is a measure of how recently have we interacted with the customer. Let's take a broad perspective and count any transaction from all the customer facing processes we mentioned above. The actual metric of recency is the number of days elapsed since the last contact with the customer.

Similarly, frequency is a measure of how often we have interacted with the customer, again taking the broad perspective of all the customer facing processes. And finally, intensity is a numeric measure of how productive the interactions have been. The most obvious measure of intensity is the total amount of purchases, but perhaps we think the total number of web pages visited is a good measure of intensity, too.

All the of the RFI (recency, frequency, intensity) measures can be subdivided into separate measures for each customer facing process, but we'll keep this example simple.

Now for every customer, we compute their RFI metrics for a rolling time period, such as the latest month. The result is three numbers. We imagine plotting the RFI results in a three dimensional cube with axes Recency, Frequency, and Intensity.

Now we call in our data mining colleagues and ask them to identify the natural clusters of customers in this cube. We really don't want all the numeric results: what we want are behavioral clusters that are meaningful for our marketing department. After running the cluster identifier data mining step, we find, for example, eight natural clusters of customers. After studying where the centroids of the clusters are located in our RFI cube, we are able to assign behavior descriptions to the eight behavior clusters:

A: High volume repeat customer, good credit, few product returns
B: High volume repeat customer, good credit, but many product returns
C: Recent new customer, no established credit pattern
D: Occasional customer, good credit
E: Occasional customer, poor credit
F: Former good customer, not seen recently
G: Frequent window shopper, mostly unproductive
H: Other

We can view the tags A through H as text facts summarizing a customer's behavior. There aren't a lot of text facts in data warehousing but these behavior tags seem to be a pretty good example. We can imagine developing a time series of behavior tag measurements for a customer over time with a data point each month:

John Doe: C C C D D A A A B B

What do you think of this time series? We successfully converted John Doe from a new customer, to an occasional customer, and then to a very desirable high volume repeat customer. But in recent months we have seen a propensity for John to start returning products. Not only is this recent behavior costly, but we worry that John will become disenchanted with our products and eventually end up in the F behavior category!

This little time series is pretty revealing. How can we structure our data warehouse to pump out these kinds of reports? And how can we pose interesting constraints on customers to see only those who have gone from cluster A to cluster B in the most recent time period?

We can model this time series of textual behavior tags in several different ways. Each approach has identical information content but they differ significantly in ease of use. Let's assume we generate a new behavior tag for each customer each month. Here are three approaches:

1) Fact table record for each customer for each month, with the behavior tag as a textual fact.
2) Slowly changing customer dimension record (Type 2) with the behavior tag as a single attribute (field). A new customer record is created for each customer each month. Same number of new records each month as choice #1.
3) Single customer dimension record with a 24 month time series of behavior tags as 24 attributes.

Choices 1 and 2 both have the problem that each successive behavior tag for a given customer is in a different record. Although simple counts will work well with these first two schemes, comparisons and constraints are difficult. For instance, finding the customers who had crossed from cluster A to cluster B in the last time period would be awkward in a relational database because there is no simple way to perform a "straddle constraint" across two records.

In this example we are influenced very much by the predictable periodicity of the data. Every customer is profiled each month. So, even though the behavior tag is a kind of text fact, design choice number 3 looms as very effective. Placing the time series of behavior tags in each customer record has three big advantages. First, the number of records generated is greatly reduced, since a new behavior tag measurement does not by itself generate a new record. Second, complex straddle constraints are easy because the relevant fields are in the same record. And third, we can easily associate the complex straddle constraints with our complete portfolio of customer facing fact tables by means of a simple join to the customer dimension.

Of course, modeling the time series as a specific set of positional fields in the customer dimension has the disadvantage that once you exhaust the 24 fields, you probably need to alter the customer dimension to add more fields. But, in today's fast changing environment, perhaps that will give you an excuse to add to the design in other ways at the same time! At least this change is "graceful" because the change does not impact any existing applications.

We have succeeded in boiling down terabytes of transactional behavior data into a simple set of tags, with help from our data mining colleagues. We then have packaged the tags into a very compact and useful format that supports our high level ease-of-use and ease-of-application-development objectives. We are now ready to pump out all sorts of interesting behavior analyses for our marketing end users.