Simulation frameworks: The key to dashboard success

White Paper

Simulation frameworks: The key to dashboard success

White Paper

May, 2011
www.mu-sigma.com

Simulation frameworks:
The key to dashboard success

Regardless of the organization that you work for, chances are that you use dashboards to display and deploy metrics. The technology for building dashboards has continuously evolved, so much so that it is now possible for a non-technical person to "build" a dashboard. Despite their ubiquity, whether dashboards have been able to achieve their utmost potential is subject to debate.

Most dashboards typically start life in a business function (e.g. a spreadsheet tracking report). With increasing use, more data integration is required and the number of users burgeons, spawning the need for a full-fledged dashboarding solution. Departments (or governance bodies, in some instances) typically determine key metrics that must be part of the dashboarding solution, and IT is brought in to gather requirements and select the technology for a successful implementation.

Independent of the hierarchy of implementation, each such exercise must attempt to answer two key questions:

- What metrics must be chosen to maximize impact on business?
- What is the relationship between metrics, and is there an overarching framework into which these KPIs slot in?

Often the latter of the two — the focus on the big picture — is lost during the development of dashboards.

Dashboards to "Cockpits" — A System Dynamics Approach

Imagine that you're piloting a space shuttle. Would you prefer a conventional dashboard displaying certain choice metrics and trends or would you prefer a control panel, a "cockpit," with "actionable insights" to negotiate the vagaries of inter-stellar travel? Piloting an organization is often not very different from helming a space shuttle, and the future of dashboards depends on the extent to which they can emulate "cockpits," "flight simulators" and "auto-pilot mode," a notion first explored by Rob Walker [1].

The secret to developing dashboards of such astounding efficacy and power could lie in the disciplines of simulation and system dynamics. A commonly studied concept in simulation is the "Stock Flow" where a "stock" is simply an accumulation of an entity over time, and the status of stock varies depending on the "flow" variable. The mathematical equivalents of stock and flow are the "integral" and "partial derivative," respectively. This metaphor is appealing given its simplicity of explanation and intuitive appeal; stocks can be thought of as a bathtub and a flow will fill or drain the stock. Using these building blocks, one can then visually build a system complete with graphics and metrics that derive from the model. Someone non-technical could intuitively verify the model assumptions.

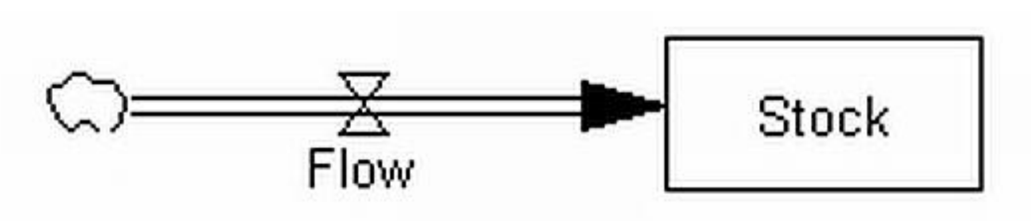

Figure 1: The stock flow concept

Customer Loyalty Program — The Dashboard Framework

Let's say you're responsible for creating a tool to monitor the health of a loyalty program. Following the system dynamics approach would first entail the creation of a stock-flow map (see figure 2). Performing this exercise early on in the life of a dashboard ensures that the subsequent steps are grounded in theory and are sufficiently representative of reality. For a loyalty program, the key actors in the map are the customers ("stock") and their inflow/outflow represents the "flow" variable.

Prospects flow into the enrollee customer state, and enrollees either activate into customers or they never conduct business with the loyalty program. Customers that flow from the active state to inactive are considered the loyalty program's churn flow.

Now that we have "The Big Picture" in place, designing the dashboard is a straightforward process. We have an accurate view of the interrelationships governing the key metrics. The map display as a navigation device is a useful addition to any dashboard. Metric trends may be animated on the map. When one clicks on a stock or a flow, all the key metrics describing that state are displayed.

For example, if one clicks the "Actives" stock, one could then see the number of active customers, customer segment distributions, "recency" and frequency tables, revenue and OLAP-style drill-downs displayed in a dedicated dashboard view. A benefit of this approach is it segments metrics immediately into two groups: Stock or Flow.

Upon practice, one can utilize a similar template for stock variables and another for flow variables. The variable segmentation promotes re-use of the designed templates, thus enabling simpler implementations from a technology standpoint.

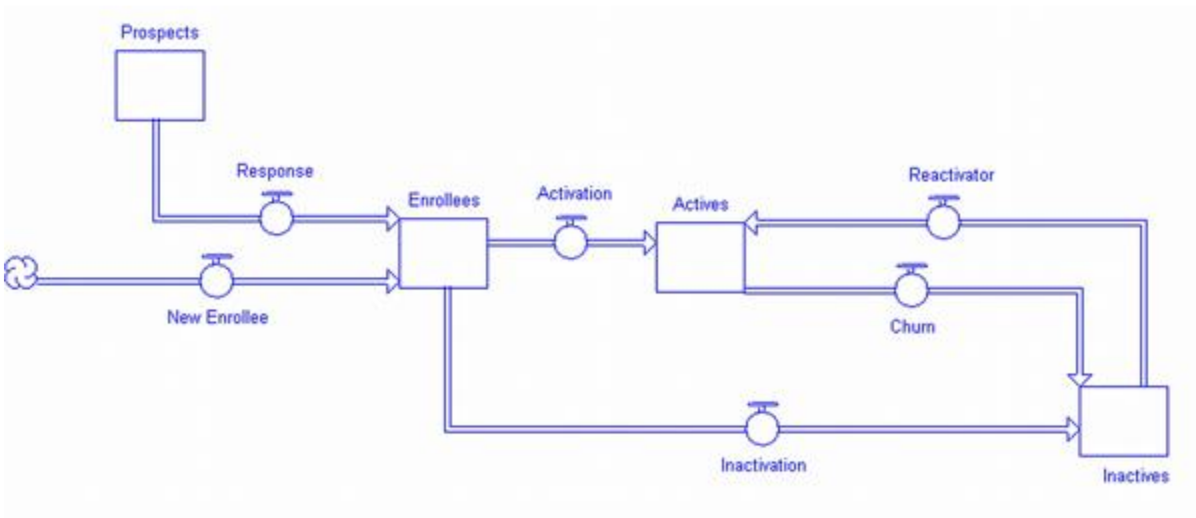

Figure 2: A stock flow map for a typical loyalty program.

Customer Loyalty Program - Onward to Simulation and Optimization

With the stock flow map in hand, one can then form the basis for constructing a mathematical model of the system. The mathematical model opens the door for robust simulation and optimization as one matures beyond the dashboard reporting view. In its simplest form, the evolution of the system over time is constructed using the stocks and flows in the published map. For example, an analyst observes the active customer base in the firm's customer loyalty program has begun to stagnate. The number of active customers is not increasing over time as expected.

You need to intervene and try to boost active customers, but what do you do? Viewing Figure 2, let's increase spending in the prospecting area of the map and boost the flow of spending dollars into the prospect stock. What would be the outcome with respect to active customers of this action? For example, increased prospect spending would likely cause an increase in the number of prospects, given an estimate of the response activation rate. You then can calculate the new stock of enrollees. Increased enrollees translate into a boost of active customers through the new enrollee activation rate.

Improving response activation performance, attempting to reduce churn or some combination of these strategies present other scenarios to focus on. All these example scenarios are estimable from the map. In order to further enhance simulation accuracy one could introduce hierarchy. For each stock of customers, utilize customer segmentation to form sub groups. A customer segment is treated like a sub-stock to the parent stock, and one can track the inflow and outflow of each customer segment. This will reconcile in the parent stock, and one would gain considerable improvement in tactical ability.

As the process of scenario analysis matures, most likely users will begin asking for the dashboarding system to recommend optimal scenarios given constraints. Optimization naturally extends the simulation apparatus; one can link the optimization engine with the automated output of the simulator, iterate and search for an optimal condition or control rule. Stochastic optimization as well as probabilistic meta heuristic approaches such as simulated annealing work fine in these applications.

Last Word

In sum, incorporating the "stock flow" mapping technique empowers the developer and end-user by giving them an extensible framework for understanding dashboards. Furthermore, this approach paves the way for successful implementation and is a natural step in the progression toward flight simulator and auto-pilot dashboards.