A Missing Profession: Information Design

In business intelligence it seems to me like there is a missing profession: information design.

Business Intelligence (BI) solutions ultimately aren't about the data that an organization has: they are about the information that the data carries. This information has to be uncovered, it has to be validated, and it has to be refined in a way that is usable.

What Information Design Isn't

Information is different from data. For instance, imagine a bank; we've got checking balances, transactions, and monthly trends. That's data. What does this data mean about the chance that the customer will leave the bank? What new accounts and services they might want? The chance that there are fraudulent activities associated with the account? That's information. Data is something that is clear and unambiguous; information needs to be inferred from the data available. Information ultimately is meaning and that makes it both messy and rewarding.

Information design relies on database design but isn't database design. Paradoxically data can be wrong, or noisy, or incomplete, and still carry a lot of information. For instance I was working on customer purchasing behavior and I found a segmentation code that carried a lot of data about purchase patterns. I asked about this segmentation and found it was done over a decade ago and it was considered obsolete because it was done so long ago – even though when I investigated it was very useful. Whoever had done the segmentation in the first place had clearly done an damn good job.

Information design isn't software design. Computer programs like a web browser function by presenting data in a certain form, regardless of the content. If a web page properly follows the HTML protocols then a browser can show the page, regardless if the page is the IBM main page or a blog for a cat. This means that there are clearly right and wrong software solutions. Either the pages display or they don't, and if some pages don't display that's a bug that needs to be fixed. Information design doesn't have clear right and wrong but it does have better and worse answers. An information design answer can work – produce a number where a number needs to go – but not be very good.

An Example of Information Design

Let's say we're designing an attrition[1] system. When a customer call customer care, we give the representative a recommendation. We have a lot of options:

1) We can't ever know exactly who is going to leave so let's not address the problem.
2) Have an overall policy that treats all customers exactly the same.
3) Present an attrition score to the customer representative.
4) Present an attrition threat flag to the customer representative.
5) Give a graded response with reasons and some specific recommendations to the representative.

Depending on the company any one of the solutions may be appropriate.

If a company is going through a period of low attrition, ignoring attrition may be the best response. There can easily be more important problems for an organization to worry about. I have seen this happen in companies where attrition has been a critical focus of the company: the incremental effect of a new attrition-focused system is small. However, if attrition is a problem in the company (1) can be a foolish approach. It is usually impossible to tell who exactly is going to leave but good analytic design can tell you how to make bets and get a good return on your efforts.

Solution (2) is what companies usually do, and if the policy is well thought out this can be sufficient.

Solutions (3) and (4), while apparently more sophisticated that solutions (1) and (2) are asking for trouble, How are service representatives supposed to interpret the data they are given? If we give customer service representatives a raw score without guidance, then good representatives will worry about their interpretations and the attrition score will become a source of stress. If we give an “Attrition Threat: Yes / No” flag, then we've lost the ability to distinguish between a slight risk and a substantial risk. we'll be giving the representatives clear guidance but that guidance probably not be appropriate and the company will be worse off than if they had no policy.

What we want to do is solution (5): break the base down into segments with guidance and insight in each

1 Attrition is when a customer stops doing business with a company.

segment, making sure that our intervention is appropriate and effective in every case.

still, there are right ways and wrong ways to build this solution. One wrong way would be to interview the marketing and customer service managers about what they think drives attrition and value and build a segmentation that reflects their combined opinions. Unless the managers are surprisingly sophisticated their opinions are most likely going to be far removed from the reality of customer behavior. Another wrong way is to grab data from a handy database and build statistical and financial models without benefit of the organizational understanding of the company. We would want to make sure we combine the best data we can get with the deepest insight we can find. Optimization plus insight beats either alone.

When we build this system we'll want to make sure we can track who called in to customer care, was the recommended offer made or not, and be able to link that back to actual customer behavior so we can track the effectiveness of the program and continuously improve our design.

If we can't commit to this level of sophistication and effort we are better off with a simpler solution that we can execute effectively.

There is a substantial and subtle design trap in solution (5). The solution will require database deign and interface design, and it is very easy for those design issues to overwhelm the project and for information design to get slighted. Without solid information design, without being able to say who is likely to leave, why, what their future value is, and being able to track and learn from the results the best database and interface design will be useless.

Real-Life Examples

That's a hypothetical example of information design; let's talk about some examples where I don't think that design was done so well, and how I think it could have been made better.

The Rate Plan Wrong-Sizer

I was working for a telecommunications company when my group was introduced to the Rate Plan Optimizer Project. IT had just spent one million dollars in development budget and they needed a group to take over the product.

The goal of the Rate Plan Optimizer was to help customer service reps suggest rate plan improvements to customers. The product did this by

1) Assume every customer had exactly the same usage patterns with the only difference being their minutes of use and then
2) Look at a series of rate plans and suggest to the customer the plan that would be most profitable to the company.

The product had a number of parameters that could be managed, and IT wanted our group to do the managing.

I can't tell you that much about the parameters because my group got as far away from the project as quickly as we could. The project was broken enough that no amount of parameter tweaking could fix it and we didn't want to take the blame for generating bad customer experiences.

What's wrong with the Rate Plan Optimization Project and how should it have been designed?

Let's start with the customer usage profile. To start out with a project that's intended to give individual recommendations to customers and start that project by assuming that all customers act the same is amazingly dense. The Rate Plan Optimizer project manager explained that they had a study done several years ago saying that most customers were fit pretty well by their profile.

First off, a study done a few years ago doesn't mean that much in a constantly changing world, not when data can be updated easily. Second, even if most customers are pretty well fit by the profile that means that some customers are badly fit by the profile and will be negatively impacted by the system's recommendations.

The reason that the IT department went with using a one-size-fits all usage pattern was that the customer data warehouse did not actually have customer usage data in it, only how the customer was billed. The IT department should have taken this project as an excuse to get the usage data into the data warehouse. The customer recommendations could have been been done at an actual customer level.

The next major problem with the Rate Plan Optimization project was choosing the rate plan that was most profitable to the company and then suggesting the customer adopt that plan. In other words, the Rate Plan Optimizer had the goal of making the customer's bills as large as possible and making sure the customer got the worst possible plan from the customer's standpoint.

In order to fix this problem the company has to do some hard thinking about what kind of company they want to be and what kind of customers they want to have. Other things being equal companies want the customers to pay more for goods and services and the customers want to pay less; on the other hand companies want to attract customers and customers are willing to pay for goods and services they want. This means that in order to maximize the total return there is a real tension between maximizing the price (to get as much as possible from each customer) and minimizing the price (to attract customers and make sure they stay). How to resolve that tension is by no means trivial. One option is to assume that “our customers are stupid people and won't care that their bill just went up” but I don't think that's a good long-term strategy.

Ideally we want to find services that are cheap for the company but that customers like a lot. Standard customer surveys will just give us average tendencies when what we care about the preferences of each individual customer. Fortunately we have an excellent source of that customer's preferences: the rate plan they are on. Let's assume that the customers are in fact decently smart and are using roughly the best rate plan for them, but they might need some help fine tuning their plan.

Take the customer rate plans and divide them up into families. When a customer calls up, look at their actual usage and calculate their monthly bill in the different rate plans in their families. If a customer can save money by switching rate plans, move them but keeping them in their rate plan family. This method makes sure the customer is getting a good deal and sticking within their known preferences, and the company is still maintaining a profitable relationship with the customer.

Premiums from Credit Data

A company I was with was building a modeling system to look at individual credit history, compare it with insurance premiums and losses, and identify customers where the insurance premium was either too high or too low. I was only peripherally involved with the project and only brought in at the end. What we were asked to predict was the overpayment or underpayment ratio so the insurance companies could adjust their premiums.

The project started by receiving large files from the client and starting the model building process. The team decided to start out with a simpler problem by predicting if there was a claim or not, and once that problem was solved using the understanding gained to move on to the larger problem.

Things didn't work out so well.

The modeling effort ran into trouble. The models were drastically underperforming from what was anticipated. The team tried every modeling approach they could think of, with little success. Eventually the whole project budget was used up in this first unsuccessful phase with little to show for it. I was brought in at the end but couldn't help much.

There's a long list of things that went wrong.

The team forgot the project they were on. They were using approaches appropriate to marketing response models and they were working in a different world. Doing 40% better than random doesn't work well for marketing response models but here it meant we could improve the insurance company rate models by 40% which is fairly impressive. Before the project started the team needed to put serious thought into what success would look like.

The team let an initial step in the project take over the project. At the least, that initial step should have been ruthlessly time-boxed. Since that initial step wasn't directly on the path towards the outcome it should not have been in the project.

The team didn't do any data exploration. When I was brought onto the project near the end, one of the first things that I did was to look closely at the data. What I found was that over 10% of the file had under $10 in six-month premiums, and many other records had extremely low six-month premiums. In other words, a large chunk of the data we were working with wasn't what we think of as insurance policies.

This goes to an earlier point, that often DBAs know the structure of their data very well but often have very little idea of the distribution and informational content of their data. Averages, minimums, maximums, most of what we can get easily through SQL don't tell the story. One has to look closely at all the values and usually this means using specialized software packages to analyze data.

A new team, including myself, was brought in to take a second pass at the project. What we did was to 1) look at the data to make sure we had a valid data set, validated with the client 2) make sure we had standards to meet that were appropriate to the project and 3) started with a simple solution and then built more complex solutions. What approach 3) meant was that very quickly we had some solution in hand.

Predicting Daily Customer Attrition

Attrition is when a customer leaves a company. I was charged with producing daily attrition forecasts that had to be within 5% of the actual values over a month. The forecast vs. actual numbers would be feed up o upper management to understand the attrition issues of the company and the effect new company programs were having on attrition.

Because my group had been working at the company for a few years we were able to break the attrition down by line of business, into voluntary and involuntary (when customers don't pay their bills), we were able to build day-of-week factors (more people call to leave the company on a Monday) and system processing factors (delays from the time a person calls to have their service canceled and when the service is actually canceled). Our forecasts performed within 3% of actual attrition. Often we were asked to explain individual day's deviations from predictions which we were always able to do – invariably major deviations were the result of processing issues, such as the person that processed a certain type of attrition taking a vacation and doubling up their processing the next week.

We were able to break down the problem like this because we knew the structure of the information that the company data contained and we were able to build a system that respected that information.

The story eventually had a less than desirable ending. After producing accurate daily forecasts for months our work was replaced by another group's work, with the predictions that were much higher than ours. It turned out that having attrition sometimes be higher than predictions and sometimes lower was very stressful to upper management and what they really wanted to be told wasn't an accurate prediction of attrition but that they were beating the forecast.

Ultimately the problem was a large difference between what management wanted and what they said they wanted. What management said they wanted was an attrition forecast at a daily level that was very accurate. to this end my group was constantly refining and testing models using the most recent data we could get. What this meant was that all the most recent attrition programs were already baked into the forecasts.

What management really wanted to be told was the effect of their attrition programs, and by the design of the forecasts there was no way they could see any effect. It must have been very disheartening to look at the attrition forecasts month after month and being told your programs were having no effect.

What my group should have done is to go back roughly a year, before all of the new attrition programs started, and to build our forecasts using older data. Then we could make the comparison between actual and forecasts and hopefully see an effect of programs.

What Does It Take to Be Successful?

All of the examples that I have given are of poor information design. Some of them have had more or less success, but they all had substantial flaws. There's a reason I'm saying that information design is a missing profession. Why is it so hard?

First off, true information design projects are fairly rare. BI is usually about straightforwards reporting and ad-hoc analysis. People don't get much of a chance to practice the discipline.

Information design requires a lot of other disciplines. It takes statistics but isn't limited to statistics. Data mining can help but can easily bog down a project in complicated solutions. It requires being able to think about information in very sophisticated ways and then turn around and think about information very naively.

It requires knowing the nuances of an organization. Who are the clients? The users? What is the organizational culture? What does the organization know about itself? What does the organization strongly believe that just isn't so? It's not impossible for an outside consultant to come in and do information design, but it is impossible for a company to come it with a one-size-fits-all solution. When it comes to information design, one size fits one.

Because the profession of information design hasn't been developed yet, it isn't included in project plans and proposals. For two of the projects above information design wasn't even thought of and for the third it wasn't done well because the clients true needs weren't uncovered.