



BI and Data Quality: challenges of secondary business processes

Tom Breur
February 2009

Introduction

Why is it that Business Intelligence professionals always seem to be whining about poor data quality? Business Intelligence (BI) projects are fraught with data quality issues. There are some structural reasons for this, which have a lot to do with the role BI plays in organizations. On top of that, in many organizations, a BI project or data warehouse may well be the place where data streams from independent (legacy) source systems are being confronted for the first time. In many cases the “surprising” data quality issues that arise, “belong” to the project team, despite the fact that these problems have existed long before the BI project was ever envisioned.

I find it ironic that data quality largely remains a “motherhood and apple pie” issue. Everybody agrees that it’s a problem, and everybody agrees that something should be done about it. However, when it comes the time to find volunteers who want to actually *do* something about it, everybody seems busy...

To clarify the context of data quality issues, I find it useful to distinguish between primary and secondary processes in an organization. What I refer to as primary processes, are business processes that directly create value for the customer. Secondary processes support primary ones. Examples of primary processes are Customer Service, Sales, Manufacturing, etc. Examples of secondary processes are Controlling, Human Resources, Corporate Strategy, and of course also Business Intelligence. On average, you can typically expect (significantly) lower data quality in secondary processes. Why might that be?

Primary versus secondary processes

There is a fundamental reason why secondary processes are fraught with data quality issues, and primary processes don’t. This has to do with the way in which these data are *used*, and the time lag between

producing data and confronting them with reality. In the case of data from primary processes there is an “automatic” correction mechanism, which is embedded in business workflows.

Let’s take an example. My hairdresser recently asked me if they had registered my details in their system, yet. What’s a hairdresser without a CRM system these days? “Yes”, I replied, and spelled my name: “B-r-e-u-r.” A few minutes later she was back. “Are you sure?” When we were at the checkout, she suggested checking the system by looking up my address details, which duly led to a hit: “Ah! I see your name was spelled as B-r-e-*i*-r.” “No”, I said, “that’s the wealthy branch of our family; *my* name is spelled as B-r-e-*U*-r.” This is an example how the primary process of registering loyalty immediately leads to the discovery of an error.

Let’s take another example. When I make a reservation with an airline company, and when I check in there turns out to be no seat available for me (let’s dismiss structural overbooking, for the sake of this example), we face an *immediate* problem. And this problem will need to be resolved then and there, tapping into resources of the airline company. It’s up to them to find a solution for this.

If and when a pattern occurs in such errors, this will be discovered rather quickly. And management will be notified as well, because resolving these kinds of operational issues costs time and money. This is at the heart of the explanation why primary processes typically are characterized by *higher data quality*. When errors occur this will be noticed very quickly, *and* resolving the issues occurs at the expense of the service provider. So they are quite “naturally” motivated to prevent reoccurrence.

Secondary processes, on the other hand, usually have (much) lower data quality. This is due mainly because the self-correcting mechanism we just described is largely missing. If our hairdresser wants to use the database for direct marketing and would fail to deduplicate records from her system that belong to Mr Breir en Mr Breur (who are *really* the same person), chances are low this will be spotted in the first place. If the error is found, this will be much later, and possibly remote not only in time but also in space. And even when they do discover the error, there is much less of an incentive to innovate work processes to prevent such errors from happening again in the future. So errors tend to linger on in secondary processes like running a direct marketing campaign.

But apart from the timeliness of detecting errors, there is another consideration to take into account: problem ownership.

Business alignment: problem holder versus problem owner
We'll define a problem holder as the person who experiences "pain" when a problem occurs. The problem owner is the person who controls resources required to resolve the problem. Business alignment is created when problem holder and problem owner are one and the same person, and/or, they are closely related in time and (organizational) place. Business alignment implies that all efforts within an organization will all work in unison. Internal friction is minimized. As a result of this, energy and investments will effectively contribute to the corporate bottom-line.

Part of the complexity in dealing with thorny data quality issues is that the nature of most organizations allows conflicting goals within an organization. When the problem holder is the problem owner, he or she can himself decide whether he wants to invest in resolving the problem, or whether the issue really isn't important enough to be bothered about. A BI professional who is capable of elucidating such innate internal conflicts as the root cause of lingering data quality issues, will bring the essential conflicts that senior management needs to reconsider to the surface.

If our hairdresser wanted to schedule a direct marketing campaign, a duplicate mailing to Mr Breir and Mr Breur would constitute a loss. Two mailpacks are sent to the same address, of which only one person can respond. So the marketer who pays for the postage is now the problem holder, the problem owner is the hairdresser who can determine whether there are really two clients with similar names living at the same address.

We will see similar conflicts when we think about issues with data quality that arise when disparate data streams need to be brought under conforming dimensions in the data warehouse.

Consolidating data streams in the data warehouse

In many cases when a data warehouse is built, or when data streams need to be consolidated in a data mart, we are faced with fundamentally irreconcilable differences. What is the data warehouse team to do? In many cases, the data warehouse team is the first to ever attempt to combine these data streams, so although the

inconsistencies may well have been present long before, the business as a whole was oblivious to them.

Two or more disparate source systems supply data that need to be combined in order to create value for the business in a BI project. However, for the purpose of primary business processes, these source systems always functioned well. So they themselves do not “feel” any “pain” as a result of imperfect data quality. Let’s take an example.

A car dealership handles prospects, people who take a car out for a test drive, and they also gather data on eventual buyers. The repair shop of this same garage (where often important revenue streams reside) maintains data on their clients, which are largely a subset of previous buyers, and may also consist of owners who purchased their car elsewhere. Both the sales staff, and the repair team have been quite pleased with their respective systems, and never suffered any serious problems with data quality.

When the data warehouse team now attempts to combine these data streams, they may well run into inconsistencies neither of them were ever aware of. Sometimes, for example, the (legal) person purchasing the car may not be the one driving it. So the same vehicle may appear to have different “owners” to the sales staff and the repair team.

Now the problem holder has become the data warehouse team, attempting to resolve these differences, but they are really not in a position (don’t hold the resources) to change the practices that led to these inconsistencies. Maybe, the data model needs to be changed to allow for a difference in owner and driver of a car. How much effort that would require is hard for them to determine, and they might also not be the best to assess whether that appears a commercially sensible investment or not.

The data warehouse team might be able to “fix” the existing data, but their fix will not attack the root cause, so any new data entering the system is still prone to these same inconsistencies.

Conclusion

The difference between primary and secondary business processes elucidates that each require a different approach if we want to improve data quality. Business Intelligence, as a secondary process, by its very nature is much more susceptible to issues with data quality.

There is a fundamental distinction between primary and secondary business processes, and one of them is that the distance between problem holder and problem owner is typically much larger for secondary processes. The further these are apart, the greater opportunity exists for conflicting data to persist.

If you need to deal with data quality issues in primary processes, the solution typically resides in a redesign of the process. For secondary data streams, the fundamental conflict first needs to be made apparent, where there might well have been hidden costs to the business previously unknown. Often only the BI team is in a position to surface such costs, and making those transparent might well provide the lever to instill fundamental changes.

Tom Breur

Tom Breur runs XLNT Consulting, www.xlntconsulting.com, committed to helping companies make more money with their data