

## ***The Data Constraint to Successful Enterprise Integration***

Without Data Integration, Enterprise Integration is extremely costly and overly complex. Many businesses are convinced that Business Intelligence, the Internet, and Customer Relationship Management are critical, strategic areas of investment, however most are struggling to deploy the enabling technologies. What are the constraints that prevent businesses from achieving these important goals? There are many obstacles to overcome, from proper organizational change management to maturing underlying technologies.

The teams behind the implementation of these solutions frequently are able to harness the new technologies, and are capable of gaining organizational support. However, there is another critical constraint that must be overcome but all too frequently is not: the overwhelming task of integrating a set of otherwise disparate systems.

The struggle is trying to share and use data, from one application to another, or from one business or business unit to another, in ways that were not foreseen when the original applications were deployed. The data, frequently assumed to be of the appropriate quality for the new application, is not so. The result? Many labor hours are spent in unplanned and unbudgeted tasks trying to find, understand, and reconcile the data to meet the new need. This issue is also manifested when changes to legacy or newly developed applications must be made in response to changing business needs. Many times the cost of making “very small changes” to an application is unbelievably large. This is very difficult to explain to business sponsors and executives, especially when the existing applications work properly.

What causes this problem? The answer lies in the data architecture of the installed base. An enterprise has tens or even hundreds of applications with hundreds or thousands of point-to-point connections (interfaces) between them. Furthermore, each team deploying a new solution creates new interfaces or makes corrections or adjustments to existing ones, to achieve implementation. In these “on the fly” adjustments, little is documented of the interface or the modification. This continuous development and maintenance of point-to-point connections make an already large and complex problem larger and more complex for the next application or modification. Why is this done? This is how system interface development has been done since the beginning of times, to this day this is how “application integration” is taught to systems personnel, and this data sharing approach based on a point-to-point is pervasive in the industry.

This approach to building interfaces or executing conversions, on a point-to-point approach creates the IT applications “Fur-Ball”. The term Fur-Ball was selected because it best describes the behavior of the overall architecture. If an application or interface is modified, the impact is near impossible to predict due to the extremely large number of interdependencies. Most of the time, a number of them are not identified, much less analyzed for impact.

This “Fur-Ball” is the actual data architecture of the installed base. It is an architecture no one intended or consciously selected as the state they wanted to achieve (no one said “we want a Fur-Ball of data interfaces and conversions between our

applications”). Nevertheless, it is the real underlining of the enterprise. It is an architecture that imposes constraints on how applications are designed, integrated to one another, deployed and operated.

The Fur-Ball creates two undesirable conditions in the enterprise data systems. First, it creates a mechanical problem. In this architecture the applications are tightly coupled. An application is tightly coupled to another application if the first (the target) is exposed to the internals of the latter (the source) beyond the data that is shared. In this architecture, if one application changes, it impacts many other applications, that in turn impact others, creating a domino effect. Many businesses and application areas believe that their applications are loosely coupled (data coupled) because they only have a “file interface”; the applications “are exposed only to the data they need from the other application”. However, the data supplied by the “source” application is not only facts (it is more than a reflection of the facts represented), but it is loaded with application processing information such as sequencing, timing or other processing content. These data contain facts but also contain “imbedded” rules that must be known and handled by the recipient application to properly use the data. When the source changes its rules, even though the target does not need to change per se, the target must now change to align with the new rules, and it must do it in lock-step!

Second, it creates a semantic problem. In this architecture, the applications contain the same facts, but they are represented in different and many times inconsistent ways. A typical symptom is when the same report is produced with the same information from two different applications or business areas, and the information is inconsistent (e.g., one report indicates 12,354 claims opened last month, the other indicates that 13,405 claims opened last month). In many instances, the same fact, say the “collision limit” is represented by different applications based on how they process the data (how they sort, sequence, establish hierarchy between coverage limits, etc.) not the fact that it is intended to represent. Other examples of imbedded rules include the combination or aggregation of facts to meet one application processing constraints (if is not in the continuous 48 states, “roll it into foreign”) but then causing other applications to misreport or misrepresent the information if unaware of the imbedded rule.

In summary, this environment is characterized by incoherent information, hundreds of scheduled and ad-hoc conflicting reports, “departmentalized” data files, “data specialists,” ad-hoc Tiger Teams launched on “data quests,” etc. What is even scarier for businesses is the fact that many times the rules used to make these transformations are only known by the code in the interfacing application –the programmer and the business matter experts involved are no longer with the business areas or the company. When a business with this condition tries to deploy a new application whether CRM, BI, EAI, IAI, an Internet solution, or a traditional solution, the company is faced with this dilemma, the well known “data problem”.

How do you know if your enterprise suffers from the “Fur-Ball” syndrome or the “data problem”? It is simple; if changing an application must be coordinated in lock-step with other applications unrelated to the change, you have mechanical problems. If the business areas indicate that they get “too much data, but not enough information”, inconsistent or conflicting reports or information, or erroneous data that “they know is good in the source system”, you have semantic problems. When the cost of not managing

data, including its replication, exceeds the “pain threshold” of your company, you know you have a severe Fur-Ball problem!

How did the enterprise get to this state? First, seldom do businesses recognize that they must manage their data as an asset, and they don't. Second, seldom do they recognize the fact that data is being replicated, and even less often they see the need to manage replication of their data, and again, they don't.

For several years now, I have work with organizations trying to understand this problem, defining approaches to solve it, identifying and applying policies, processes, procedures, technologies, etc. In my work I have encountered several critical findings; those thing that frequently become show-stoppers. The first finding indicated that not having an agreed upon, company wide operational definition of each piece of data was diminishing its value. Another critical finding indicated that the use of EAI and Data Warehousing technologies in isolation was less effective than using them in concert. Anther critical finding indicate that Metadata practices are lacking, forcing business and IT areas to the “shoe box” approach for Metadata management. However, the most important finding was that not having top level executives understand the significance of the data problem and lacking their active participation in implementing solutions was the largest impediment to a successful resolution of the “data problem” or, as I like to call it, “the untangling the Fur-Ball”.

In subsequent articles, I will cover some of the actions taken, describe how issues can be framed, how to assess the alternatives, select a solution, and implement it. What do you think? I would like to hear from you.

**About the author.** Andres Perez is a Senior Information Management Consultant with IRM Consulting, Ltd. Co. He has over 25 years in the data processing field, with over 23 years of data management and architecture experience. He is an avid practitioner of data management and data architecture. You can reach him at [andres-perez@msn.com](mailto:andres-perez@msn.com) or at (210) 413-1481.