INFORMATION ARCHITECTURE FOR STEEL

by

Madhurak Rajagopal*, Gabriel Vladut**

SYNOPSIS:

Information Technology (IT) has come a long way since the first transistor was invented. Across all industries, IT has been used to varying degrees and has delivered mixed results.

Within the Steel industry, IT has a long history, from the time the first IBM Mainframes were used, to mini-computers such as DIGITAL PDP11 machines and finally to the client/server and web-based systems of today. These systems are from different vendors, run on different operating systems and support an array of business functions; some of the applications have been built in-house, some bought from vendors; some are deployed locally, some centrally.

This disparate array of systems, solutions and deployment alternatives has many ramifications to the effective running of the companies:

- Many different computers, operating systems, databases adding cost and complexity in operations.
- Complex communications infrastructure needed to enable even the simplest link between applications and users.
- Data redundancy as the same item of data is stored in many different formats in many different places.
- Lack of consolidated data on which to base a company's decision-making process.

This paper discusses these issues and proposes an "Information Architecture for Steel". The Oracle Information Architecture (OIA) is in part a vision, a direction, and a blue print that is based on many years experience of bringing comprehensive information based systems to the world's most complex companies. OIA is a plan that recognizes the huge advances that have been made in technology over the last five years, the urgency behind the desire of companies to get a better understanding of their business, whilst also recognizing that companies are tending to become more complex.

This paper describes OIA as a ten-stage plan. The order of the stages is a logical order of progression but is by no means rigorous. Key benefits from such architecture are also described.

[*] *Solutions Director, Oracle Corporation, Singapore ; **Director IPA Craiova, Romania

INTRODUCTION

It is clear that human history and the development of iron and steel are intricately linked. Just as civilization reaped benefits as humanity gained an understanding of using fire and tools, use of iron by humanity led to even more prosperity of civilization and culture.

In a similar line, computing has an intricate link with human history. Whether it is the use of stones, the Abacus or the latest high technology gadgets and computers that we use today, computing has also been a major contributor to the prosperity of civilization and culture. Today’s computers, as we know them, have come a long way from the early Abacus to the Vaccum Tubes, Transistors, Integrated Circuits, Personal Computers and Super Computers. Combined with the applications and other infrastructure, this collection came to be known as Information Technology - IT.

Very early, businesses grasped the importance of IT in automating and speeding up tasks – from the mundane to the most complex. With the evolution of computers, they were deployed in waves, from mainframe to mini-computers to the personal computers. Keeping pace with this evolution, a plethora of components were also deployed - architecture, operating systems, applications and tools, databases, etc.

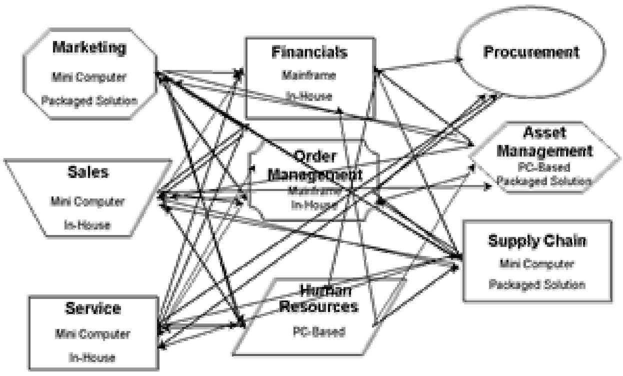

Figure 1 - IT Infrastructure Complexity

As a result, if we look at any medium/large company’s IT infrastructure today, we will find a variety of computers, operating systems, networks, tools, databases, applications and user interfaces. Some have been deployed locally while some have been deployed across the company. Some have been developed in-house, while others have been bought from vendors. We also see that this IT infrastructure has had an evolutionary growth over time.

Given this reality, it is not easy for companies to move to new technologies as they appear or as the technology vendors introduce them. Many factors are considered before embarking on an IT project – ROI, business benefit, cultural implications, etc. The Oracle Information Architecture (OIA), as described earlier, is a vision, a direction, and a blue print that is based on many years experience of bringing comprehensive IT systems to the world’s most complex companies.

Before undertaking to move towards the OIA, however, a few harsh realities must be taken into consideration:

- Systems integration through data is preferable to integration through messaging
- Holding data more than once does not improve its quality
- With more off-the-shelf applications available to address “commodity” business tasks, the need to hand-craft such applications diminishes
- A wider mix of IT infrastructure (platforms, applications, databases, etc.) cause complexity, leading to wasted resources and higher costs

The Information Architecture for Steel is built on the 10-stage plan of the OIA.

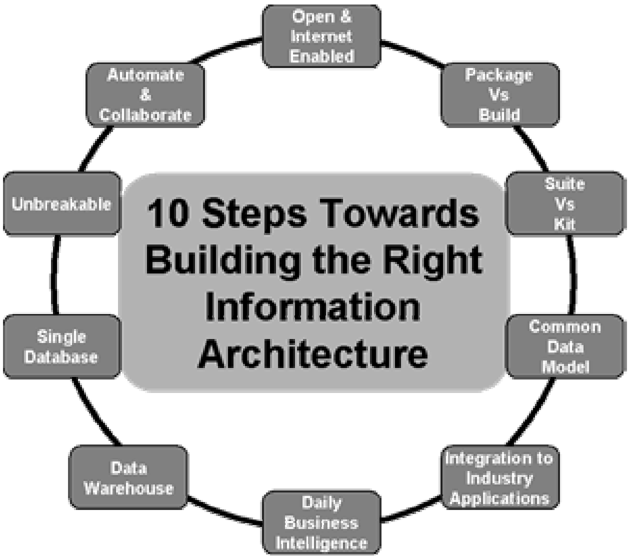

Figure 2 - The Right Information Architecture

It takes into consideration the nature of steel industry, the evolution of IT infrastructure within Steel companies and the IT requirements of these companies. The stages are divided into three major principles:

1. Application architecture principles
2. Data architecture principles
3. Technology architecture principles

APPLICATION ARCHITECTURE

Step One – Open and Internet Enabled

Enforce the use of standards in all respects of IT deployment to ensure effective communications within and outside the Enterprise at all times by exploiting the web.

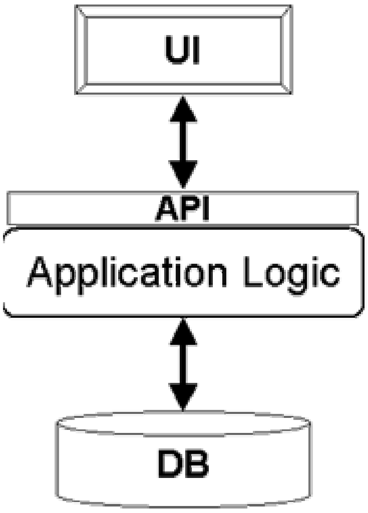

Figure 3 - Web Enabled Architecture

'Internet Enabled' does not necessarily mean that systems have to be connected to the open World Wide Web. In order to take advantage of technologies in a quick and intelligent manner and also to ensure that communication with the outside world is effective, IT systems must use 'standards' in every aspect of deployment. These standards are used to define communication protocols, computer languages, interfaces, etc.

However, there is one 'standard' that is not official and that is to adhere to a web enabled architecture – the 3-tier architecture of thin, web-based client, the middle application server and the third database layer. This architecture is very critical to an effective IT rationalization. The Web is changing the world; it should be exploited ruthlessly to derive full benefits.

Step Two – Package vs Build

Increase the use of non-industry specific *Commodity applications* instead of building them in-house.

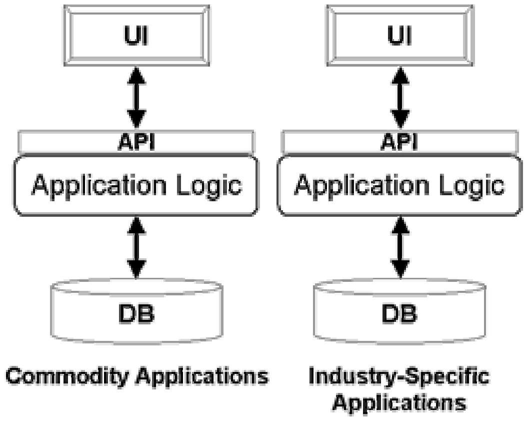

Figure 4 - Industry & Commodity Applications

The wide range of applications implemented in any company's IT infrastructure can be classified into two broad categories – industry-specific and commodity.

Industry-specific applications, as their name suggests, are specific to a particular industry, tend to be transactional in nature, are mission-critical, often real-time and where most of the data is created. Some examples include Airlines Reservation systems, Telephone Mediation systems, Retail Point of Sale systems, Bank ATM Networks, etc. Specific to the steel industry, these could be Level 3 Production Planning systems, Level 2 Control systems, etc.

Such applications are provided by niche and specialist suppliers and take a fair amount of effort and time to install and run. In almost all cases, these applications have to be customized to a specific customer/plant/location. There is also a fair amount of integration with other systems.

Commodity applications, on the other hand, do not share the characteristics of 24X7 and real-time requirements (although there is increasing pressure for 24X7 as companies go global and become web-enabled). These applications are process-driven and provide similar functionality across a wide range of industries. Some examples include Marketing, Finance & Accounting, Human Resources, Purchasing, Project Management, etc.

Commodity applications are provided by a wider range of "packaged" solution providers and require little industrialization (configuration, not customization) and can be provided out-of-the-box with rich functionality demanded even by the biggest companies.

Step two therefore recommends that companies move towards the acceptance of pre-built software for the second category of commodity applications. In most cases, it is often cheaper and faster to change business processes rather than to build these applications in-house to suit the companies' "commodity" processes.

There are two inherent advantages with this approach. The first is that, although more money may be spent in buying pre-packaged applications, avoiding application-to-application integrations and dedicating a team of software engineers save a significant amount of money. The second advantage is that the pre-packaged applications deliver a richer and broader set of functionality than those developed in-house because these applications are built by companies dedicated to software engineering, taking inputs from a wider industry set.

Step Three – Suite vs Kit

Ensure that the packaged applications are all based on a single, integrated set of relational tables – suites rather than kits, integrated by data, not by a bus.

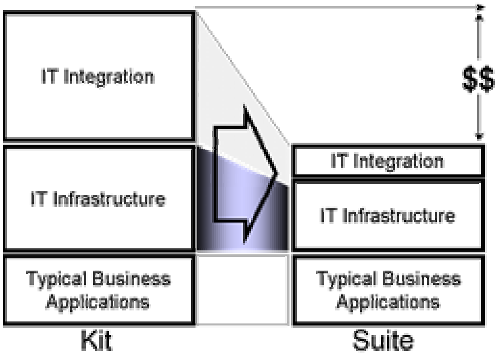

Figure 5 - Potential Savings

As Step two outlines, companies will have to use applications from different vendors. However, when these applications need to work together, many issues come to the forefront. The major issue in terms of Information Architecture is that these packages will tend to use different database storage systems and structures. This means that key data such as Customer information, Item definition, etc., is stored in more than one application, and, most certainly, in disparate structures, making integration at data level very risky and complex.

Step three therefore recommends that as many applications as possible must run off the same, single, structured and integrated set of relational tables. This ensures that:

a) data is stored in a single ‘location’
b) data is not replicated
c) data is shared by many applications
d) data is current and accurate across applications

Technically speaking, this single set of relational tables supporting the applications is referred to as an **Operational Data Store (ODS)**. The data is:

- volatile
- updated in real-time
- shared by many applications
- operational in nature

Step Four – Common Data Model

Ensure that all applications share the single data model (ODS) by publishing the model and all relevant Access information

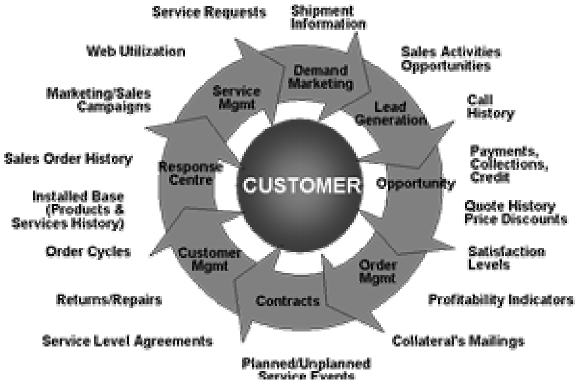

Figure 6 - 360-degree View of a Customer

One of the end goals of this Architecture is to ensure that information is held only once, in a centralized data model/database. If this has to be achieved, three things must happen – a) this data model has to be made available to all relevant applications, both, internal and external to the enterprise, b) all applications must communicate through a set of Application Programming Interfaces (APIs) that maintain data integrity across