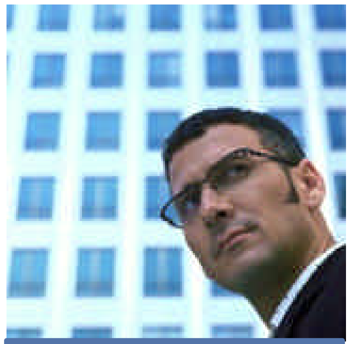
Data Centric Integration and the Modern Enterprise



A Descriptive White Paper

by:

Jonathan Arneault

Senior Technology Advisor

August 2003

Integration: The Problem and Approaches

The modern enterprise relies on information technology as the life blood of its operations. The ability for a company to have deep understanding of its customers, operations and suppliers is key to success in the age where customers and suppliers have a high degree of access and diversity of information on all aspects of business. In the modern enterprise, he who knows the most, and can act the fastest, wins.

The difficulty enterprises face in light of this is that their IT investments (their software, operating systems, databases and specialized systems) do not easily communicate with each other, if at all. Worse, even where there is "communication", rarely do those systems understand the information that is available from other systems. The resulting mismatch of data types, content, and data profiles causes companies to spend billions of dollars and millions of hours every year trying to piece together a patchwork of mechanisms to provide minimal interoperation between the components which make up their IT infrastructure.

"In the modern enterprise, he who knows the most and can act the fastest, wins."

This is not ideal. Nor is it the envisioned outcome of the business leaders who sought IT solutions to complex business issues. IT investments have always been sought as enablers to the business cycle, not means to their own. The failure of companies to capitalize on the exponential value potential that is locked up in their IT investments is a direct result of those systems not acting in a cohesive manner; and this is due to the inability of those systems to communicate vital business information to each other in real time, and to do so in a way that each system natively can utilize to its maximum potential.

To create a modern Enterprise IT Ecosystem, it is crucial that companies grapple with the need to provide integration across their enterprise IT environment. The problem will simply not go away.

Companies that approach integration as a core service that is leveraged across their IT infrastructure can achieve exponential, rather than incremental value from their IT investments.

Integration Approaches

"Companies that approach integration as a core service that is leveraged across their IT infrastructure can achieve exponential, rather than incremental value from their IT investments."

There are three main approaches to integration. They are:

? Programmatic Integration

? Canonical Format Integration

? Data Centric Integration

Within the scope of these three approaches, a variety of terms and variations are found within the IT industry. However, most solutions fall into one or more of these descriptions, with varying degrees of faithfulness to their core approach.

Programmatic Integration

Programmatic Integration is where applications provide information and execution commands to each other via programmatic interfaces (often referred to as API's). This approach is common, and has the distinction that it provides the programmer with an ability to make real time (or near real time) interaction with the applications that need to communicate. The major downfalls of this approach are that it requires a high degree and cost of maintenance, and that programmatic integration allows only two applications at a time to be integrated. For large numbers of systems, a hierarchy of application code sets must be written, and then maintained. Worst of all, with this approach, many systems which need to interoperate, never can be integrated, since there is no communication mechanism across their different environments (such as differing operating systems or underlying programming languages). In spite of all of this, Programmatic Integration remains a much used approach, as programmers understand the means to do it, and it typically requires little outside investment (although it has a high internal personnel investment.)

Canonical Format Integration

Canonical Format Integration has gained much favor as of late, but it is not new. Companies have long used this means to "integrate" certain aspects of their operations, such as a Customer Master File which is accessible across applications, and standard formats such as BIA (Bank Institute of America), EDI

(Electronic Data Interchange) and other "standardized" or "canonical" data formats. Recently, tagged data, in the form of XML (eXtensible Markup Language) has gained preeminence in the minds of companies who are coming to realize the high cost of "operational isolation" within their IT environments. The promise is that if you turn all of your data into XML, then everything will be able to communicate with everything else. This is a noble idea, but it bears out very little in reality. There are key values in providing canonical integration between enterprises, most notably the exchange of business execution information (such as in EDI, orders and payment information, or in BIA ,with reconciliation and account update information). The difficulty in this approach is that it requires that applications limit their capabilities to the constraints of the "canon" of the data, and it also requires that all applications understand that data format. Further, canonical data formats rarely provide "IT ecosystem" information as part of their makeup (and this is good), so the company using them is not able to use their IT infrastructure in a flexible way which will benefit their business. Hence, they spend a great deal of time and money re-tooling their IT investments.

"… if you turn all of your data into XML, then everything will be able to communicate with everything else. This is a noble idea, but it bears out very little in reality."

Data Centric Integration

Data Centric Integration is typically viewed by savvy IT executives as a 'holy grail' of IT. However, the rationale IT exec knows that the hundreds of different data types that exist within their environment and the thousands of different instructional interfaces and integration points that are needed are not simply addressed by some mystical solution. As a result, they look to formulate a "integration architecture" which uses Data Integration as the means to allow all systems to interoperate. By means of this approach, a highly flexible mechanism for systems to communicate and interoperate is achieved.

"Data Centric Integration is typically viewed by savvy IT executives as a 'holy grail' of Information Technology."

Key to the Data Centric Integration approach is the need to transform data between native formats, and the need to understand API's as data interface points, rather than programmatic requirements

Data Centric Integration is a long established means of transforming data from its source format into its target format, and providing some level of logic in the transformation process to make it useful and coherent to the target environment. Data Integration approaches have included "helper applications" and "Transformation Tools", and over the past five years or so, data transformation architectures which can be leveraged across the enterprise IT environment.

"Key to the Data Integration approach is the need to transform data between native formats, and the need to understand API's as data interface points, rather than programmatic requirements."

The key value of this approach is that an extreme degree of flexibility can be achieved in the IT environment, giving this value to the business that possesses it. Further, a tremendous reduction in cost for interoperability compared to the alternative approaches is inherent to this approach. Lastly, the data integration approach has the least amount of moving parts, and hence the least amount of complexity in implementing and maintaining, and hence is the most likely approach to succeed for the complex enterprise environment.

Data Transformation: descriptive and example

The concept of transforming data is relatively simple, but making it happen is much more complex.

Information systems store and communicate data in a variety of means and formats. Within those formats, there are further complexities such as data relationships, constraints and data definitions. As the number of systems and data stores increases, so does the variety and incompatibilities of data types and data formats.

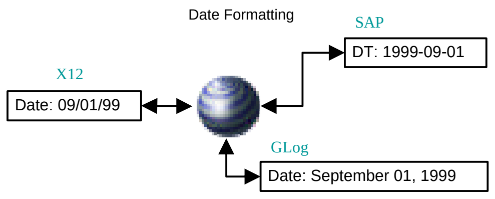
When these diverse systems need to exchange information, it must be modified from its source formats into its target formats, as well as needing to meet the "data rules" for the target systems.

Transformation is more than taking extracting data from one form, and placing it into another form. Data may need complex logic or value compilations performed prior to sending to the target. Further, the originating data may require the combination of information from many backend systems. Above this, the target system and format may need to be determined, depending on the content of the data itself.

An excellent example of data transformation is the receipt of a purchase order in EDI (ANSI X12) format, and sending this information into the needful backend systems in their native format. In this case, we'll presume the backend Enterprise Resource Planning (ERP) system is SAP, and the Manufacturing control system is GLog (two popular systems for manufacturing logistics). The originating document (an X12 850 document) contains only the required customer information, and abbreviated value fields, while SAP requires the internal customer number, and GLog requires the customer ship-to-location ID, as well as the detailed component information for the "stuff ordered".

What we see in this example is the need for more than one originating data source (the EDI file, a customer master file, and a parts inventory database), and two disparate target systems, each of which have distinct data format and data rules requirements. We also can imply the need to change the contents of the target data based on lookup tables for part information and for the full values for the abbreviated data elements.

Example of Data Transformation



A simple transformation that might be included in this example is DATE forming. If we presume that the EDI 850 forms the date field as MM/DD/YY, how can we make each of the target systems receive their required date value (for SAP, YYYY-MM-DD and for GLog, MONTH NAME DATE, YYYY)? Through processing this using a data centric approach, we can use the internal logic of the data transformation system to construct the required values, as opposed to programmatic interrogation and code based logic.

Data Transformation can be much more complex than this. Often, it requires inquiring and aggregating data from multiple sources, and having logic applied to the combination and manipulation of the source data. Even so, the purpose of data transformation remains true: perform a transaction on source data which results in the required target data being generated, without the need for programmatic or canonical data intervention. Quite simply, the data is "molded" into the form of the needed output in as few steps as possible.

As one can imagine, to make often rigid and highly complex data pliable enough to be crafted into multiple output types requires a high degree of sophistication on the part of the Transformation Engine. Thankfully, such sophistication shields a great degree of complexity from the Analyst, who is able to manage integration from the "data centric viewpoint". Indeed, such engines allow for near code free integration, since they utilize a data centric integration approach.

Data Routing

A transaction is more than simply shuffling bits around and following inflexible steps. Often, a decision of "the next step" needs to be determined based upon the content of the data being dealt with, while the "target formats" of the data will not be not known until some inquiry of the source information is made. The ability to be deterministic about the destinations of those data transactions, and the form the data transformation will take is known as Data Routing.

"Data Routing is the determining of the target systems and data formats based upon values contained within the source data itself. The Integration System must be able to interrogate the content of the source data, and through loosely-coupled rules, designate the outcome of the transaction set and then process it to the desired end points."

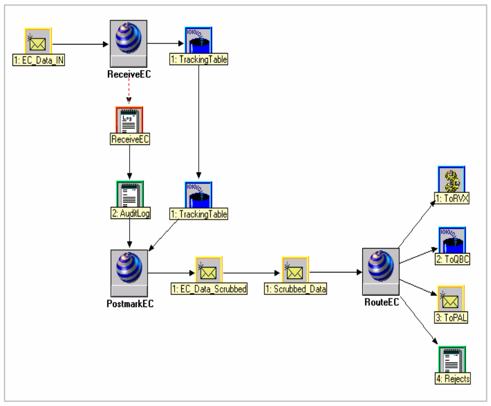
Many people have misperceptions about data routing. Often, the viewpoint is skewed by thinking that there must be explicit instructions or destination information in the source data in order to provide "routing of the output". This thinking comes from a common connectivity mechanism, known as "Message Oriented Middleware" (MOM), which in itself doesn't provide integration, but does in many cases provide much needed network connections and cross operating system communication capabilities. In such an environment, the "destination" of the data message must be explicitly determined. Data routing, however, is quite different from this lower level service, and provides a higher level of business functionality.

Data Routing is the determining of the target systems and data formats based upon values contained within the source data itself. The Integration System must be able to interrogate the content of the source data, and through loosely-coupled rules, designate the outcome of the transaction set and then process it to the desired end points.

Data routing is near impossible in the "canonical format" model, since all content must adhere to pre-determined data set types.

It is very do-able within the programmatic model, but such a mechanism requires a large degree of code based logic to be constructed, and is very difficult to maintain since any change in process or addition of options requires a change to the source code and all of its components. Further, this also forces the developer to understand all of the different types of systems being integrated, and all of the differing data types, constructs, rules and elements, and then duplicate these within his or her code.

An Example of Data Routing



Within the Data Centric Integration model, Data routing, while still a high value function, is made much easier. Since the incoming data is already interrogated by the system so that it can be directly transformed, the same process can have logic added to it that can be deterministic based on values or combination of values that are contained in the source data sets. Further, since Data Centric Integration treats all systems equally (as producers and consumers of data, with source and target rules), being flexible with the end points of the process is much simpler.

Data Routing is different than Business Process Integration (BPI). Data Routing addresses transactional level functions, where BPI orchestrates transactions (such as data integration transactions). Business Processes may also provide deterministic behavior, but they do so at a transaction coordination level, not “inside the transaction” control. Data routing provides control inside the transaction itself.

Data Integration in an “API required interface”

Many custom systems, and indeed many commercial applications have no “data centric” input or output. Instead, they rely on Application Programmable Interfaces (APIs). This mechanism of invoking an application and passing “parameters” to it is the most common mechanism for moving data around and performing business logic.

“In the Data Centric world, an API is a ‘Data Interface’, not a place where code must be written.”

Common to this are systems that are written in C++ and Cobol, as well as Java (although Java seems to be somewhat more flexible in this regard). Realistically, it is really dependent on the way that the original developer wrote the system. The problem in the eyes of most developers is, if there's an API, they have to write code to talk to it.

In the Data Centric world, an API is a 'Data Interface', not a place where code must be written.

The ability to interface with a system that requires an API call without having to know the original language, or to write code is a huge boon the enterprise, as it significantly decreases the complexity and cost of interfacing with such systems, and opens the world to other systems who do not have the means of communicating using that particular API means. Further, the Data Centric approach of integration in an API world means that when one or more sytems need to communicate with an API, yet don't meet all the data needs, format requirements, or logical edits, the Data Centric integration system is able to "get the missing information" from other systems, or to accommodate the missing information through data construct logic, without the need to code or to make external system exits to accomplish it.

An API, in the data centric world, simply becomes another data point (both in and out). The Data Centric Integration Engine speaks to the API via its interfaces (often called adapters), and presents data to the adapter, even when that data may be a function call, executable parameter, or data string. These elements are treated as mission critical data by the data centric integration system.

Mercator's Data Integration Solutions

Mercator is the world leader in data centric integration. Through over eighteen years of research and development, Mercator has served thousands and thousands of companies across the globe with integration solutions spanning legacy systems, web environments and complex B2B and B2C worlds.

Mercator Inside Integrator is the Enterprise Integration Architecture for thousands of companies, and is the backbone of ecommerce for over half of the Fortune 500.

In addition, Mercator has solutions for data centric integration for a variety of Enterprise Applications, including SAP, PeopleSoft, Oracle, and many, many more.

Mercator also provides out of the box data integration for over 35 common systems, including native integration with messaging bus systems, databases, Java App Servers, XML, Web Services, Web Integration, and Cobol based applications. Our experience spans the breadth of the largest enterprises in the world.

Further, Mercator has specialized, industry focused solutions for Financial Services, Health Care, and Manufacturing, Retail and Delivery (MRD) environments.

Mercator Inside HIPAA serves over half of the BlueCross / BlueSheild companies, as well as hospitals, insurance companies, large provider groups, and regulatory users.

Mercator also is the engine for transactions at the London Stock Exchange, the Chicago Mercantile Exchange, and for many of the Financial Service firms back end global execution systems. Our InsideOMGEO and InsideGSS solutions are industry leaders.

In Manufacturing, Retail and Delivery, we provide integration solutions such as Global Order Management, Supply Chain Visibility, and others. Our experience and understanding in MRD has helped us save hundreds of companies millions and millions of dollars every year in their operating and production costs, and speeding their time to market for their products.

Many case studies, as well as detailed information about the Mercator product line can be found on our website at www.mercator.com.

HEADQUARTERS
45 DANBURY ROAD
WILTON, CT, USA 06897
VOICE: 203.761.8600
FAX: 203.762.9677

EMEA OPERATIONS
CITY TOWER, 40 BASINGHALL STREET
LONDON, ENGLAND
EC2V 5DE
VOICE: 020.7314.9600
FAX: 020.7314.9601

APAC OPERATIONS
LEVEL 3, 110 PACIFIC HIGHWAY
ST LEONARDS NSW 2065
AUSTRALIA
VOICE: +61 (0)2.9478.3100
FAX: +61 (0)2.9478.3110

WORLDWIDE CUSTOMER CARE
TOLL FREE: 1.800.215.9633
DIRECT: 1.203.563.1211

WWW.MERCATOR.COM

ABOUT MERCATOR SOFTWARE, INC.

Mercator ® delivers large global organizations The Advantage Inside Integration™, providing Industry-Ready Integration Solutions™ that solve critical business problems in real-time, while leveraging current technology investments and maximizing ROI.

Mercator's core integration technology, **Mercator** Inside Integrator™ 6.7, features a Solutions-Oriented Architecture™ which easily and seamlessly automates high-volume, complex transactions.

Over 1,100 global businesses leverage the power, speed and flexibility of Mercator's proven integration technology and industry expertise to build better business value and faster ROI. To hear why our customers and partners believe Mercator is the advantage inside integration, visit our Web site at www.mercator.com.