Alternative Technologies

Understanding and Selecting Integration Approaches

David McGoveran
Alternative Technologies
6221A Graham Hill Road, Suite 8001
Felton, California, 95018
Website: www.AlternativeTech.com
Email: mcgoveran@AlternativeTech.com
Telephone: 831/338-4621 Facsimile: 831/338-3113

Report Number 20050405

Disclaimer and Notice

This report is produced and published by Alternative Technologies, an independent consulting and analyst firm in Boulder Creek and Felton, CA. The information and opinions presented in this report are exclusively those of Alternative Technologies, except where explicitly quoted and referenced. Although reasonable attempts have been made to insure the accuracy of the report, no guarantees or warrantees of correctness are made, either express or implied. Readers are encouraged to verify the opinions stated herein through their own efforts.

This independent report was supported in part by Ascential Software without influence on the opinions herein. Client (Ascential Software) is granted a non-exclusive license for unlimited distribution of this report in its unabridged and unaltered form for internal use. Neither this report, nor any abridgement or derivation of this report, may be otherwise reproduced in any form or media without the explicit written permission of AT (Alternative Technologies). This restriction in rights is not intended in any way to impede the mutually intended use of this report by Client and AT; rather, it is intended to preserve both the integrity of the report while maintaining both AT's rights of reuse of AT's intellectual property and any confidential information of either Client or AT contained herein. Portions of this report which directly discuss facts pertaining to Client, Client's products and services, or other aspects of Client's business are assumed to be Client Confidential (if this report is designated "Client and AT Confidential") and otherwise are published by permission. This report is, in general, the sole property of AT and protected under International Copyright laws.

For information about this or other reports, or other products and services (including consulting and educational seminars), contact Alternative Technologies directly by telephone, mail, or via our Web site:

Alternative Technologies
6221A Graham Hill Road, Suite #8001
Felton, California 95018
Telephone: 831/338-4621 FAX: 831/338-3113
Email: research@AlternativeTech.com
Website: www.AlternativeTech.com

TABLE OF CONTENTS

1. Introduction

Integration is a key success factor for improving business efficiency. There are a plethora of integration approaches, some that focus on processes and some that focus on data and information. Each of them comes with specialized tools, like EAI, ETL, and EII, tailored to their purpose. Most companies employ some combination of these approaches and technologies to achieve different IT initiatives. Recently, the traditional lines between IT initiatives have begun to blur, and the capabilities of these technologies to overlap and converge.

So how do you know which integration approach to use when? The easiest way to select an appropriate technology solution is to identify integration patterns, using concepts derived from pattern matching. An integration pattern associated with a technology solution is a requirements scenario common to successful multiple projects implemented using that approach. An ideal pattern comprises those requirements that are uniquely associated with exactly one technical solution; requirements common to all solutions can be set aside for the purposes of technology selection. By looking at the requirements of common patterns and matching these up against the specific capabilities of each of these technologies, the appropriate technology choice becomes more obvious.

This white paper explores the different integration problems and the technologies which are typically applied to each of them. It provides insight into the distinct capabilities of each type of technology. In addition, it identifies some general patterns which are strong indicators of specific approaches, and includes patterns that indicate the use of multiple combined approaches.

2. Integration: Key Factor for Improved Business Efficiency

Information technology (IT) has long been associated with improved business efficiency, in part because a monolithic application provides a degree of cohesive operation that cannot be achieved manually. As information technologies evolved, specialized and diverged, islands and barriers developed between them. Integration enables the combining of diverse technologies so as to achieve a high degree of operational cohesion, while enabling flexibility. Today, most IT organizations treat integration as a core competency. Appropriately done, integrating data, applications, and systems is a key factor in improving business efficiency.

Many businesses tried to achieve business efficiency by creating ever larger enterprise applications. By the mid 1990s, it became clear that trying to reengineer business processes for efficiency and then use commercial enterprise applications (such as ERP systems) to automate them was a costly effort that entailed many difficulties. It tended to over-emphasize so-called “best practices” in standard business processes to the exclusion of competitive differentiation, regional market requirements, resource availability, and so on. By 2000, traditional EAI approaches were proving costly, having their own form of rigidity. The result has been a renewed interest in highly flexible enterprise applications, whether commercial or proprietary, and the rapid adoption of process driven, service composition approaches.

Highly custom and proprietary solutions need to be integrated with commercial, standardized solutions, all the while recognizing both their strategic and operational importance. At the same time, corporate management has come under increasing pressure to react more rapidly and even proactively, have a so-called 360° business view, improve accounting timeliness and accuracy, report material decisions openly, regulatory compliance, and so on. These drivers have emphasized the integration of business activities so that they can be managed more effectively and in near real-time. Thus, a subtle shift in emphasis has taken place from application integration for operational and technical efficiency to *business integration for effective strategic and operational management.*

3. A Plethora of Integration Approaches

Application integration is the most common approach to integration. It focuses on stringing together application programming interfaces across systems to enable transactions and associated data to flow automatically from system to system. In most cases, this approach involves a middleware layer that enables "brokering" the transaction data between systems, translating it into the required format of each individual system and ensuring the completion of the unit of work. This middleware layer approach is often referred to as enterprise application integration (EAI), and typically includes a messaging backbone to ensure asynchronous delivery of the data across systems, and an integration broker, which provides the translation and routing logic.

Application integration approaches provide insight into the transaction, including the status of the transaction at any point in time and what the complete data within the transaction looks like (e.g. all of the information that makes up a complete purchase order). However, the scope of understanding is always limited to a single transaction, or at best a summary level understanding of a time series of transactions. From this information, it becomes impossible to derive, for example, a complete view of a customer, since any individual transaction only has a limited view of that data. The business process context is unknown, making it difficult to know relative importance and to selectively optimize operational efficiency. Any information outside of the context of any given transaction is simply outside of the EAI charter.

Application integration practice involves the modeling of data that is exchanged between applications (i.e., messages), rather than between databases. The number of applications and messages is very large compared to the number of databases and the total collection is constantly changing. These data models or schemas (as they are usually called) tend to implement a structural representation of only those logical relationships essential for the application-to-application exchange.

Application integration specialists have developed methods and tools for capturing and representing data along with its schema in XML. Unfortunately, the lack of a common reference model with which these constantly evolving schemas can be integrated leads to redundancy, discrepancies, and errors in of data definitions. These, in turn can lead to increased costs of application integration projects and maintenance, operational errors when data is used improperly from one application to the next, and even failures.

Data integration focuses on pulling together information from multiple data sources (usually databases), typically to create a new unified representation of that data in a distinct database. This unified representation leads to an improved understanding of the business and better decisions. The most common association of this approach is with data warehousing, but it is also often used in the implementation and consolidation of enterprise applications. Data integration has developed sophisticated techniques for bulk movement of large quantities of data in batch, and is often thought of being completely delivered by extract, transform, and load (ETL) tools. However, data integration is actually a data centric approach that extends beyond traditional ETL to include data profiling, data quality, and metadata management, and may involve data change capture, replication, and synchronization as well .

The underlying driver of the data integration approach lies in understanding the complete scope and context of data across multiple systems and using this understanding to allow specific information to be assembled and merged together in a new place where it can be used for business purposes. This requires the ability to profile the data in source systems, to get a true understanding of usage, content and structure. This level of understanding is impossible to obtain by simply looking at a structural schema.

While data integration is perfectly suited to obtaining a holistic view of information across an enterprise, it is not particularly concerned with the structure of individual transactions. Data integration deals with data as it relates to a broader information model and, though it may ensure data integrity, is typically unaware of structures specific to a transaction.

Also unlike application integration, data integration is not particularly well-suited to maintaining the context of transaction data across a multi-step process. This is partially because messaging is not an inherent feature of most data integration technologies. It is also attributable to the fact that data integration processes are not usually orchestrated, but are highly optimized data flows that would be treated by application integration technologies as a single complex application function or service.

Information integration also focuses on pulling together information from multiple systems, but takes a different approach. The information integration approach leaves data in source systems, even if they use disparate technologies, rather than moving it to a unified database. It then enables the data to be assembled on demand using a federated query. Fairly recently, the category of technologies designed for this purpose has come to be called enterprise information integration (EII).

The information integration philosophy is to leave data where it lives, so that duplicate copies do not need to be created, thus eliminating potential points of error. The idea is to allow the data to be queried as if it were all in one location, even though it is actually spread across systems that use different technologies (e.g., SQL vs. XML).

Like data integration, information integration is heavily reliant on a strong metadata foundation to make this happen. However, most information integration technologies rely on schema definitions of data sources, rather than undertaking detailed data profiling. The information integration approach involves mapping together these schemas to arrive at a unified metadata model. Once defined, specific information can be queried from this metadata model, and the EII technology will assemble the data from source systems on the fly.

Unlike data integration, information integration does not predefine a data flow. Instead it relies on the query definition to assemble and transform the data as needed from across the systems, moving only the result to the calling system. This assembled data is not persisted like it is in most data integration scenarios, unless the calling system chooses to persist it. In addition, queries can be completely *ad hoc*, rather than being predefined operations as they typically are in data and application integration.

Most companies employ some combination of these approaches to address different requirements within their businesses. For example, most large companies have both EAI and ETL technologies in house. In most cases, if a company wants to automate transactions across multiple applications, EAI is the top choice. If a company wants to merge and load a large volume of data from multiple sources to create a new analytical database, ETL is the first consideration.

However, many of the vendors of these technologies are beginning to merge, and are consolidating the technologies, blurring the lines between them. At the same time, business requirements are not as cut and dried as they used to be, and companies are accelerating the use of analytics in their operational environments for process optimization and business performance management.

So in this time of converging technologies and converging business requirements, how do you know when to use which of these? The answer to this question requires a deeper understanding of what these technologies are actually doing to determine which design patterns are most applicable

4. How These Technologies Work

Application Integration

Application integration, as the name implies, involves connecting applications together, usually for the purposes of automating transaction processing. At the core of most EAI technologies is a messaging bus. This bus carries messages from application to application throughout the transaction lifecycle, ensuring that messages get to the proper destination and are not lost along the way.

EAI products usually include a simple event layer, which initiates the movement of information onto this bus. Events range from application events like an inventory decrement, to database events like a row being inserted, to time-based events like a specified elapsed time, to file events like the arrival of a file within a specified directory. Some EAI products even support combinations of these events. The event layer responds to these events by triggering an EAI sequence that is waiting for just such an event. At this point the sequence will complete its designed task according to routing rules defined in the EAI product.

EAI products typically interact with various systems through their application programming interfaces (API), or through packaged adapters that abstract the programming requirement. Through these interfaces, the data required for a transaction is extracted, in the format native to the source application, and written to a queue. In most cases, the data is mapped into a neutral format (usually XML-based), that is specific to the type of transaction being processed.

The EAI product manages the movement of the transaction data to each application involved in its processing, mapping the data into the format native to each application along the way until the transaction is complete and all systems are updated.

EAI products are tuned to handle large volumes of discrete transactions, where each unit of work is relatively small. The transformation functionality is typically focused on mapping to and from the native data formats specific to each of the various involved systems. In most cases, there is no way to view data across systems or transactions, other than using monitoring and logging tools that are sometimes included.

Data Integration

Data integration involves the capabilities of data profiling, data quality, metadata management, and traditional ETL. Each of these capabilities is required to achieve the complete understanding of data across systems, and hence logically pull them together into an integrated whole.

The data integration approach typically begins with data profiling. By profiling multiple source systems and using the resulting information to build a metadata representation of those systems, data integration builds a metadata map of the data across those systems, allowing a single model to represent multiple systems. The metadata model should include a logical cross-system model and its relationship to the physical models within each system. This information becomes the basis for how data will be assembled from across those systems.

Using the metadata map as a guide, data quality capabilities can then be applied to uncover the actual relationships between rows of data within and across systems and improve quality in the consolidated data. Data quality products allow this to be done without relying on an accurate key structure relationship, by looking at row data and determining which rows have a high probability of being matches. These rows can then be combined together to create a single record that “survives” the best data from each source system, based upon pre-defined rules. The net result is a single record of the best information from across multiple systems.

ETL comes in at the tail end of this, typically moving data from source systems, processing it according to the quality rules and any required transformation rules, and then loading it into a new destination as a sequence of predefined steps. ETL comes with a set of predefined transformation rules which are particularly good at bulk data transformations involving sorting, row merging and splitting, and data derivation. The definitions of the data flow and transformation logic can be accelerated, and sometime even automated, using the metadata information derived from the profiling and quality processes.

ETL has traditionally been relegated to batch processing, since the burden of moving entire databases of information from source to target tends to be very time and processor intensive. Because of this heritage, ETL products are tuned to handle massive volumes of data. The dataset sizes which most ETL products process would be impossible for most EAI products to handle, since EAI products are typically tuned to deal with large numbers of relatively small datasets. Some ETL technologies even have specialized capabilities that allow for the parallel processing of large datasets. This allows them to split the data processing apart and run specific tasks in parallel, often without writing to disk between process steps.

Despite this lineage, ETL is now commonly being implemented as a service, where these rules can be applied per transaction. This is typically accomplished in conjunction with some EAI middleware, which simply executes the ETL process as it would any other business rule or process step. This approach allows data-centric rules to be defined using the unique capabilities of data integration technology, but applied to application integration tasks.

Information Integration

Information integration technology allows multiple data sources to be treated and queried as single data source. This is traditionally referred to as federated query. The technology does a few things to make this viable. First of all, it allows the query to be written as if it were to be used with a single local schema. This allows developers to avoid problems of remote access and allows them to simply join tables together across sources.

Second of all, information integration technology optimizes the query, similar to the way a relational database would. It does this by decomposing the queries down to the individual source databases, and then assembling a query plan across them. In essence, it translates the metamodel-based query into an appropriate query at each source. Some information integration technologies also work against other types of data sources, like unstructured documents and content repositories, which neither application nor data integration technologies have traditionally been strong at handling.

The third thing that some information integration technologies provide is caching, allowing frequently accessed data to be kept in memory so that subsequent requests do not overwhelm the sources. This also speeds up query response dramatically, versus repetitive querying of remote databases.

Many EII technologies allow the data to be queried using standard query languages like SQL and XQuery, providing a natural interface to many of the products that companies already use for reporting and data analysis. Standard search interfaces are beginning to be introduced on some EII products, as well, allowing simple search strings to return federated results.

Most EII technologies also include some level of support for transforming the resulting data. This is often done in the query itself, using XQuery or SQL, so the transformation capabilities tend to be limited as compared to the transformation support available in data and application integration approaches.

5. Using Patterns to Help Make the Decision

The easiest way to understand which integration approach to use is to apply integration patterns. A pattern associated with an integration approach is a requirements scenario common to successful multiple projects implemented using that approach. The integration approach applied to each pattern takes into consideration technology capabilities available in the market, available skills, and implementation process experience.

By looking at the requirements of common patterns and matching these up against the specific capabilities of each of these technologies, the appropriate integration approach for each pattern becomes more obvious. By using a consistent integration approach each time a pattern is identified, companies can optimize reuse, reduce redundancy, and capitalize on best practices.

The problem of selecting an appropriate integration approach is reduced to one of pattern matching. Pattern matching is the process of looking for these pre-identified patterns among the requirements of each new IT initiative. When a pattern is spotted, the appropriate technologies, processes, and best practices for that pattern are applied to that initiative. This simplifies the selection process and allows the technology team to focus on the business requirements, rather than worrying about technology evaluation. It also allows each project to benefit from the experience of its predecessors.

Patterns Indicating EAI

Patterns indicating the use of EAI technology involve scenarios where data is flowing from application-to-application, either in high volumes, or as part of a complex multi-step process. These types of interactions may be long-running, and are likely largely asynchronous. In these patterns, a holistic understanding of the context of data is not nearly as important as the validity of the transaction data and the timely completion of its processing.

Pattern	Examples
High volume application-to-application communications	• Straight-through processing • Item/Inventory synchronization • Master customer data synchronization
Coordination of a multi-step distributed process	• Order fulfillment • Supply chain synchronization • Inventory allocations • Payment processing

Patterns Indicating Data Integration

Patterns indicating the use of data integration technology involve scenarios where data is being physically reshaped and moved to a new location. Bulk data movements requiring transformation to a new format are automatic indicators of data integration, but data integration can also be applied to individual transactions, where complex integration logic is being performed like cleansing or reshaping, or where bulk data processing rules need to be reused in a transactional setting.

Pattern	**Examples**
Analytical data consolidation	• Bulk data mart population • Bulk data warehouse population
Data rehousing	• SAP instance consolidation • Migration away from a legacy system • M&A application consolidation
Master data rationalization	• Customer master database creation • Customer record matching • Marketing database cleansing • Product cross-reference database creation • RFID product information consolidation

Patterns Indicating EII

Patterns indicating the use of EII technology involve scenarios where data is being pulled together from multiple sources, but a new copy of the data is not desired. While some data integration products are also capable of doing this, they tend to do it only using static, predefined queries that are exposed as Web services or Java objects. EII technologies allow the queries to be dynamic, and provide caching of frequently accessed data to insulate source systems from redundant queries and improve performance. EII technologies also allow the queries to be expressed using SQL, XQuery, or even search interfaces, making them well-suited for plugging into portals, and for analysis and reporting tools. In addition, some EII products can span relational and non-relational sources.

Pattern	**Examples**
Ad hoc access to distributed data	• Portal-based query tools • Business Activity Monitoring • Business Performance Management
Cross-application query & reporting	• Consolidation of data for display in a portal • Reporting across ERP & CRM • Analytics blending a data warehouse with real-time operational data
Applications that require a mixture of content and relational data	• Enterprise search • Warranty claims mining • Call center record queries • Customer feedback analysis

Patterns Indicating Combinations

There are many patterns where these technologies can be used in combination to solve problems. In general, EAI is a good supplement to both data integration and EII when real-time processing is required, when processing is part of a larger, distributed process, or when complex event triggers are required for processing.

Data integration is a good supplement to EAI when data is required that is outside of the scope of any individual transaction, when data quality is required as part of the transaction processing, or when data transformation logic needs to be reused across both batch and real-time scenarios. Data integration is almost always a good supplement to EII, since it provides a richer understanding of source data, and better metadata management, along with strong transformation and quality capabilities that can be applied within a query.

EII is a good supplement to EAI whenever data is required that spans multiple systems, when content-oriented data sources need to be accessed, or when high-volume transactions access the same data repetitively and the caching capabilities of EII can help reduce access overhead. EII is a good supplement to data integration when data caching can be used to reduce system load, when a simple SQL, XQuery, or search interface to data is desired, or when content-oriented data sources need to be accessed.

Pattern	**Examples**
Quality validation within transactions	• Customer lookup without a key • Supplier product code validation
Reuse of transformation logic across batch and real-time	• Data quality rules applied in batch ETL and directly from data entry in a self-service portal
Closed-loop processing	• Inventory optimization using historical analysis • Inventory allocations using detailed analysis of product distribution • Point of sale up-sell recommendations based on customer history and trend analysis
Federated queries requiring quality or transformation logic	• Customer data queries across systems where keys are not known • Queries across packaged and legacy applications with inconsistent data formatting
Repetitive lookups within transaction processing	• Lookup tables that are frequently accessed by high-volume inbound transactions
Real-time analytics	• Active data warehousing

6. Considerations for Combining Approaches

For patterns which indicate a cross-technology approach, several selection criteria should be considered before investing in technologies. These criteria are not always as important when the technologies are used in a standalone way, but when combined together to address these patterns, they become vital.

High Availability and Scalability

The nature of application integration processing is such that high availability (HA) and scalability must be core features of the solution. In most cases, if application integration processes stop, the business stops, so these technologies have always been designed to deal with unpredictable peak loads and unexpected system failures.

Data and information integration technologies have not always had this same level of requirement. Therefore, it is important to realize that, as these technologies are employed to augment application integration, the requirement for HA and scalability becomes vital to them, as well. When selecting data and information integration products, insist on architectures that are capable of withstanding the environment in which your application integration processing operates.

Standards, SOA and ESB

One of the keys to being able to combine these different technologies together is standard interface mechanisms that are open and flexible. A service-oriented architecture (SOA) provides an excellent integration mechanism, since it both standardizes the interfaces and allows them to behave in a loosely-coupled manner. One important and popular implementation of an SOA is an enterprise service bus (ESB). An ESB uses standard interfaces to provide dynamic binding of logical services to physical resources and orchestration of services, including the integration services described above.

When choosing any of these technologies, it is important to insist on support for publishing and consuming standard services. The Web Services – Interoperability Group's (WS-I) standards provide one such set of standards. Compliance with these standards helps ensure that services from different vendors will interoperate seamlessly.

Metadata Sharing

Data integration practice has long emphasized the importance of metadata, common metadata repositories, metadata exchange standards, and data semantics. Data lineage is commonly tracked, which can greatly aid data analysis. By contrast, application integration has traditionally ignored data semantics, generally assuming that correct data field formats suffice. Semantic correctness has generally depended on the domain knowledge of the developer, making it easy to err. Information integration has also been driven by metadata, although the models and depth of the metadata is generally different from that used in data integration.

Ideally, the metadata across these different technology approaches will converge, to allow them to more easily leverage one another during the development process. Currently, there isn't a solution on the market that combines metadata approaches across these technology disciplines. As the markets and technologies continue to converge, however, a solution will become a necessity and can be expected.

7. Conclusions

As businesses strive for operational efficiency, regulatory compliance, and competitive advantage, the need for appropriate integration technologies will increase. Market pressures rarely permits the time needed for extensive experimentation and approach evaluation. As a result, understanding the available integration approaches and recognizing which are best used in a given requirements scenario quickly is essential. It is even more important when multiple integration approaches might be used in combination. Unfortunately, there aren't many experts available with broad experience in all these integration approaches and with all the scenarios a business is likely to encounter.

In this paper, we've given an overview of several integration approaches and introduced a method for rapidly selecting an appropriate integration approach and technology. The use of integration patterns can save time and reduce costs while improving quality. Developing integration patterns should a standard practice within IT organizations. The integration patterns described here can provide a good start.

###

The author gratefully acknowledges the helpful contributions and suggestions of Ascential Software's Michael Curry in the preparation of this report.

About the Author and Alternative Technologies

David McGoveran is the President and Founder of Alternative Technologies, an independent analyst and consulting firm founded in 1976. Mr. McGoveran has been a pioneer in the definition, technical architecture, and uses of relational databases, distributed applications, integration, and Business Process Management Systems. He has helped define marketing strategies, products, and technical direction for companies such as Hewlett-Packard, IBM, Candle Corporation, Microsoft, and many others. He has been lecturing and writing publicly on the topics of EAI and BPMS since 1997. He is Senior Technical Editor and co-founder of the Business Integration Journal – formerly the eAI Journal, in which appears his monthly column *Enterprise Integrity*. Mr. McGoveran provides consulting and teaches seminars on Business Process Management, as well as other topics.

For pricing and availability of reports or other products and services (including consulting and educational seminars), please contact Alternative Technologies directly at (831) 338-4621 or via email addressed to mcgoveran@AlternativeTech.com.

Alternative Technologies
6221-A Graham Hill Rd., Ste #8001
Felton, California 95018
Telephone: 831/338-4621 FAX: 831/338-3113
Email: mcgoveran@AlternativeTech.com
Website: www.AlternativeTech.com