This article describes how BPM can be used for enterprise modernization. It introduces techniques and best practices to ensure BPM success, including business process discovery, business process decomposition, business process ownership, service identification, and code modularization. Compared to other legacy modernization approaches, BPM allows for effective scoping; instead of boiling the ocean, you look at a specific business process and what is required for that process to execute, and only modernize the IT functions that are needed by the process. As you implement more and more processes, you can reuse the services you developed for previous processes and gradually modernize all that is needed from the legacy application, making it relevant and essential for the future of the enterprise. The BPM approach provides quick time to value and engages business users.

Introduction

Enterprises run and maintain significant amounts of legacy code such as COBOL on the mainframe or Report Program Generator (RPG) on System i (iSeries). The business is asking for an easier-to-use environment, more visibility into performance, reduction of manual work, and a better way to handle exceptions. At the same time, IT is faced with tough decisions about what to do with these legacy applications, which usually have been around for many years and enable critical core business functions. Other approaches such as packaged applications or even build-your-own force companies to move off these legacy platforms. Business process management (BPM) is a strategy companies can use in the context of enterprise modernization that enables them to leverage legacy applications and make them relevant for the future. This article describes how BPM can be used in the context of enterprise
modernization. It describes techniques that you can use to make legacy investments relevant to your long-term enterprise architecture and strategy.

Why BPM?

This section describes why BPM is a viable approach to enterprise modernization.

Legacy application assessment

Modernization is not easy! We have worked with customers who, at the request of their CEO, CFO, CIO, or other senior executive, have asked an outside consulting firm to perform an assessment of one or more of their core business applications. Typically performed in the context of an application portfolio assessment, with the goal of rationalizing the application portfolio, companies do legacy application assessment for a number of reasons, including but not limited to:

- Growth into new geographies.
- Expanding business network to new business partners.
- Dissatisfied customers.
- Lack of operational productivity.
- Lack of skilled workforce for the technology used to implement the core application.

This detailed assessment can be the consequence of a more global application portfolio management approach that highlighted the selected applications as having a high business value but as low on functional or operational performance. Examples of core applications under assessment might be a home-grown order management system, a heavily customized Customer Relationship Management (CRM) system, or a Members Management system. When engaging in such a study, companies may be looking to answer some of the following questions:

- Is the core application going to meet the needs of business initiatives?
- Should we keep building on top of the core application?
- Are we going to find skilled developers and IT operations staff for the technology used to implement the core application?

Such application assessment studies typically include interviews with key stakeholders (both business and IT), technical reviews, assessment against specific criteria, documentation of results and recommendations, outlined key activities, and roadmaps and high-level estimates for implementing the recommendations. The output of the assessment is a recommendation and high-level plan of action aligned with the company's core business objectives.

Based on the results of the assessment, companies then make decisions on what to do with the core business application (for example, modernize, retire, or expand) taking into account factors such as industry trends, avoiding vendor lock-in, performance and flexibility of the solution.
BPM as an alternative to build vs. buy

When acting on the application assessment recommendations, companies are faced with making applications management decisions referred to as build vs. buy. Such decisions are driven by IT principles typically set by the CIO. For example, a company's application management principle may be to buy first, and build only if you cannot buy. Build offers the most flexibility and allows for the application to be highly customized and unique to the specific needs of the enterprise. Build, however, requires a significant investment of time and money. Buy, on the other hand, typically provides better time to value and requires a lower initial cost, but it doesn't meet all the unique requirements of the business. BPM alleviates these challenges, providing a flexible and efficient solution. BPM leverages existing investments and adds the required layer of visibility and flexibility, providing a solution with quick time to value and better total cost of ownership.

Business and IT collaboration

We have met with a number of companies that have both a strong Process Improvement team (sometimes called Business Process Re-engineering or BPR) and an IT team focused on producing high-value enterprise services. The process improvement team is on the business side and looks at operational effectiveness for core business processes, typically without a focus on IT implementation. The IT departments typically have a Service-Oriented Architecture (SOA) strategy aimed at identifying, implementing, and maintaining reusable enterprise services. It is unfortunately quite common that the two groups do not collaborate and interact enough. This results in business policies that cannot be enforced and expensive IT functions that don't get leveraged. BPM can alleviate these challenges. BPM facilitates the collaboration between business and IT, allows for the implementation and enforcement of business policies as executable business processes or business rules, and for the development of high-value enterprise services in support of these business processes. It is essential to understand that BPM is not a replacement for the existing business logic in the application, but rather provides a way to augment and extend the application reach.

The BPM layer

Figure 1 illustrates the process layer, positioned between business users and back-end systems. One of the back-end systems at the bottom is the core application that was the subject of the assessment (for example, CRM or Billing). The BPM solution makes use of these core business applications and legacy systems, and provides an extra value-add layer between these systems and the business users.
Adding the process (BPM) layer alleviates a significant number of the challenges described at the beginning of this article by:

- Automating business policies, workflows, and decision making.
- Reducing error and improving consistency.
- Standardizing procedures across geographies or markets.
- Providing visibility and control to team leads, managers, and workers.
- Alerting for critical events and initiating actions.

**Business process ownership**

With all the promises related to BPM, we still see too many BPM projects fail today. Studies show that a lot of the critical issues are not technical in nature but are more related to organizational change, governance, and skills. Specifically, business process ownership is one of the most critical success factors for BPM initiatives. Multiple business owners or organizations competing for rules or logic in a shared business process, or the lack of contractual aspects between collaborating parties, can lead to BPM failure. For this reason, before we dive into the specifics of BPM for enterprise modernization, we would like to take a step back and look at a specific technique that alleviates business process ownership issues.
Business process classification: implicit and explicit business processes

In order to prevent issues related to process ownership, it's essential to map the functions of the application under consideration to a classification framework. The MIT Process Handbook is a good source for such process classification frameworks (such as APQC, SCOR, LEAN, and EFQM). IBM's Component Business Modeling (CBM) is another example of such classification framework. As described on the APQC's web site, APQC's Process Classification Framework "is organized into 12 distinct categories, including five categories of operating areas and seven of support areas. Each category contains groups of processes and activities that, when considered as a whole, represent the operations of an organization. The frameworks are available in cross-industry and select industry-specific formats."

The Process Classification Framework allows for the decomposition of a business into level 1: category, level 2: process group, level 3, process, level 4: activity. Figure 2 is the APQC's Process Classification Framework decomposition for an end-to-end sale. For example:

- 8 (category): Manage financial resources
- 8.2 (process group): Perform revenue accounting
- 8.2.2 (process): Invoice customer

Figure 2. APQC categories, group and processes involved in an end-to-end sale
At the architecture level, there can be confusion about the *implicit* end-to-end business processes that the business wants to manage and monitor (e.g., Order to Cash) and the *explicit* finer-grained modular business processes (such as Service Provisioning and Billing). Implicit end-to-end business processes are the basis for the business process monitoring model. They provide the view that is relevant to the consumer of the business process – for example, as a customer you want to know about the status of your order. However, because they span multiple business domains, it is practically impossible to assign a business owner to these implicit processes. The explicit business processes, however, typically fit into a single business domain, which naturally becomes their business owner (such as Billing). Based on field experience, the authors recommend an approach where only the explicit business processes become the target for process automation and execution. The authors map implicit processes to BPMN's public processes and explicit business processes to BPMN's private processes. A business owner needs to be identified for each private process. Note that in the above process classification frameworks, private (explicit) processes are at the third level of decomposition (level 3), for example, Invoice Customer.

Flexible enterprises delegate the operations of their processes or process groups to the business owners managing these processes. This is why business agility requires the differentiation of the domain controlled by a business owner and the overall enterprise flows that are service contracts between different owners. Figure 3 depicts the APQC processes at third level involved in a customer sales end-to-end public process. Depending on the organizational decision there could be a single owner for level 2 decomposition 3.5 Develop and Manage Sales Plans or different owners for 3.5.2 Manage Customers and Accounts and 3.5.3 Manage Customer Sales. In the first case, formal BPM automation driven by a BPM engine can occur between all processes in that group, while the second case implies that the level 3 processes can be automated but interact through services or events.

As illustrated in Figure 3, this best practice results in the process layer being split into two:

- The *explicit process layer* is an extension of the existing legacy applications.
- Above the explicit process layer is the *implicit process layer*, which is used for monitoring purposes only.
While the explicit process layer will automatically be monitored from its automation model, the implicit process layer requires a monitoring model and the definition of probes and events that will reflect a high-level view of the chain of explicit processes and applications.

**BPM implementation**

This section describes best practices for BPM implementations.

**Business process discovery**

Compared to other legacy modernization approaches, BPM allows for effective scoping; instead of boiling the ocean, you look at one business process at a time and what is required for that process to execute. Business process discovery is about understanding how the target business (the selected core application) operates today (the "as-is" state) and, more importantly, how the business should operate tomorrow (the "to-be" state). Considerations include:
• How much value each task adds (or doesn’t add) to the process.
• How long it takes to complete each task.
• How much it costs.
• What information flows back and forth (including paper documents, faxes, and so on).
• Who participates in the business process (roles).
• What information channels participants use (for example, email, phone call, web portal, and so on).
• What back-end systems (applications) support the business process. **Note:** Depending on the business process, it may be necessary to access more than the targeted core business application. For example, it may be necessary to access the billing system and the customer complaint management system.

In order to understand the as-is process, the authors like to ask business users (for example, call center team lead, customer service rep) to describe how they perform their daily tasks (day in the life), including how they interact with the core application being modernized. Modeling is used to understand and document business processes (in both as-is and to-be states). The modeling tools should be user-friendly offerings (small footprint, minimal configuration requirements, and designed for non-IT business users) and at the same time provide the required modeling capabilities. IBM Blueworks Live, shown in Figure 4, or IBM Business Process Manager's Process Designer are full-fledged modeling tools that provide industry-standard constructs such as Business Process Modeling Notation (BPMN). This capability is critical to support a proven model-driven development (MDD) approach. These tools enable both business and technical users to design and build process models, which can be analyzed, optimized, and converted directly into implementation code.
A business process management approach to enterprise process modernization

Figure 4. A business process model in IBM Blueworks Live

A cloud delivery model for BPM, such as Blueworks Live, is ideal. Leveraging the public cloud eliminates dependency on the evaluating organization's IT department for the procurement or deployment of the software; the only required end-user software is a web browser which is typically already installed on the users' personal computers or laptops.

Implementing the explicit and implicit business processes

Techniques used for implementing the explicit business process for execution on a BPM system, such as IBM Business Process Manager, are outside the scope of this article. Today's best practices include quick time to value, business user involvement, and iterative development. In this article, our goal is to relate these BPM techniques to enterprise modernization. For more information on the BPM method, refer to Resources. Another success factor for BPM in the context of enterprise modernization is data modeling. The data model used by BPM (and business rules) is referred to as the business object model or BOM. It's important to understand the relationship between the BOM and the other data models. However, that is a topic for another article. In a future article, we'll describe a technique to monitor the implicit business processes from the explicit business processes execution.
Probing existing legacy for events

You can also link data events with both business processes and data synchronization solutions. Classic event publishing provides efficient change-data publishing environments for the following scenarios:

- Providing application-to-application integration, which makes it possible to push operational customer data to a packaged Customer Relationship Management (CRM) application.
- Initiating business processes, for example the addition of a customer record initiates a welcome e-mail, a credit verification, and updates to the CRM system.
- Monitoring critical data events, such as low inventory levels driving a business process for restocking a product.

For example, IBM InfoSphere Classic Data Event Publisher for z/OS captures data changes in non-relational databases, packages the changes into a consistent relational format, and publishes the changes as self-describing messages to WebSphere MQ message queues or to flat files, including event publishing from VSAM and CICS® VSAM files, IMS, ADABAS or CA-IDMS. Also, WebSphere Operational Decision Management provides a complete business events processing (BEP) platform, which is well integrated with IBM's Business Process Manager.

Consuming applications - in our case, the BPM system - can then use the data events to drive subsequent processing and integration with the process layer either with explicit processes or implicit monitoring. This loosely-coupled integration helps to ensure that each application can be changed independently of other applications.

Service identification

Business process discovery (BPD) and business process classifications are top-down approaches that start with understanding the business process at a high level and gradually drill down into the business process details. Explicit (automated) business processes typically live at the third level of business process decomposition. As described in Service Oriented Modeling and Architecture (SOMA), it is at the next level (level 4) that we typically identify reusable business functions, that is, service operations, with the right granularity. Example services operations include: `getPastDueOrders`, `validateAddress`, or `updateCustomerInfo`. Note that having the right granularity for these services and their operations is very important for reuse. A coarse-grained service provides high business value but less opportunity for reuse, whereas a fine-grained service provides lower business value but more opportunities for reuse. Note that because these services were identified from the business process decomposition, they are automatically business-aligned (they provide business value). Once these services are identified, you need to look at how the core application under modernization can provide the required functionality or implementation of the service. This is the bottom-up aspect.
Code modularization

Back to the RPG or COBOL code, you now need to make sure the legacy code can be leveraged in the modern BPM solution. The most common challenge with this is the lack of modularization, or the "spaghetti code" effect, where the same piece of code (such as a sub-routine) includes business, process, data, and presentation (UI) logic. In the modern solution, we want the process logic to reside in the business process layer and be managed by the BPM system, which is outside of the legacy application. Also, it's quite common that the process logic includes human interaction steps where users have to fill in forms in a specific order. In that case the process layer can include the screen flow logic (both the sequencing of screens and the actual screens). What we really need is a piece of code that does just what the process needs, or the business logic, for example, `validateAddress`.

To get there we need to modularize the legacy code (COBOL or RPG). This is typically done in the tools legacy developers are used to (like Rational® Developer for z, IBM U2 for Universe). It requires thorough regression testing to make sure that modularizing the code didn't break the application. This is an investment of time and resources, but once the code is modular, it is positioned for success for the future. The same piece of legacy code can then be used by BPM and other enterprise initiatives, such as SOA and Master Data Management. Also, we've found that the identified services can be quickly used by other processes, which are already targeted for implementation. Once you have modular code, you have a number of options to make it part of the modern BPM solution. Three of the most common options are web service enablement, the use of ESB and adapters, or business rules extraction.

Web service enablement

Web service enablement consists of taking the legacy COBOL or RPG code business logic function (such as `validateAddress`), and wrapping it as a web service. This is done using a wizard. Under the cover, the wizard creates a Java™ wrapper around the COBOL or RPG, and then a web service wrapper around the Java wrapper. For example, using Rational Developer for i, you can use a wizard to create web services from ILE RPG or COBOL source or from Program Call Markup Language (PCML) files. With this method, the business logic code still runs on the legacy platform, and the business process logic, as well as some user interface logic (screen flows), now run on the BPM system.

Enterprise service bus and adapters

A common architectural pattern used to connect the BPM layer to the back-end legacy system is to use an enterprise service bus (ESB), possibly with a platform specific adapter. For example, the WebSphere Adapter for IBM i is based on the Java Connector Architecture (JCA) and allows for inbound and outbound integration between the modern application and the legacy system. The adapter is invoked using Java, and can discover objects on the IBM i server. This pattern makes sense when
the data transformation or routing requirements are prominent. It provides a solid foundation for a connectivity strategy. It does require, however, new skills to write mediations in the ESB tool. For example, when using WebSphere ESB, you would use WebSphere Integration Developer to develop the mediations.

**Business rules extraction**

Business rules extraction consists of mining the legacy code for business logic (rules and policies), extracting those from the legacy COBOL or RPG, authoring them and managing them in a Business Rules Management System (BRMS). Common advantages to doing this are:

- The business rules can be understood by business users. *(Note: This statement is true for IBM WebSphere Operational Decision Management (the new ILOG jRules), but it may be more difficult for business users to understand rules on non-IBM platforms.) Rules are no longer coded in RPG or COBOL, but instead in a pseudo-English (or French!) language or as decision tables.*
- It's quicker and easier for business users to change some of the parameters of the business rules (such as the age of the driver) outside traditional software development lifecycle (SDLC) cycles.
- It is easier to reuse the business rules in the context of other applications.

A common architecture pattern with business rules in the context of legacy modernization consists of web service enabling the business rules, hence making business rules enterprise decision services. With this method, the business logic no longer runs on the legacy platform. Instead, it runs on and is managed by the BRMS, such as WebSphere Operational Decision Management, IBM's BRMS.

**Integration between the legacy application and the business process layer**

Now that you have the business logic from the legacy application available as a service (Web service-enabled COBOL code, or decision service), and now that you have the business process or the business rules implemented, the last thing to look at is how the core application can be integrated with the BPM system. *(This article doesn't cover the techniques used to implement the business process or rules. Refer to the Resources section for good links on this.) In other words, how is the legacy application going to communicate with the explicit, or process, layer? This is required for the core application to fire a business rule or kick off a process instance (such as kick off a fraud review). Two common techniques for the legacy application to interact with the process are web service invocation (synchronous) or message queues (asynchronous). For example, using libraries or toolkits, it is possible to generate web service client stubs for use in the RPG or COBOL application, thus allowing the legacy application to invoke business rules or kick-off process instances. *(IBM Business Process Manager provides a web service to kick off process instances). The other technique is for the legacy application to post messages onto queues. The BPM system then processes these messages, kicking off process instances. *(IBM Business Process Manager does this via JMS or WebSphere MQ).*
Note that it is straightforward for the business process layer to interact with the core application by invoking web services (for the web services-enabled functions or the decision services). For example, as part of the Order Management process, the process invokes the verifyAddress service.

**Conclusion**

In this article, the authors described how BPM can be the answer to a legacy application assessment and enterprise modernization. BPM allows IT departments to address business requirements such as an easier-to-use environment for the end users, visibility to procedures and performance, more control over business outcomes, reduction of manual work, or quicker response to exceptions. The authors talked about techniques and best practices to ensure BPM success, including business process discovery, business process decomposition, business process ownership, service identification, or code modularization. Compared to other legacy modernization approaches, BPM allows for the effective scoping; instead of boiling the ocean, you look at a specific business process and what is required for that process to execute, and only modernize the IT functions that are needed by the process. As you implement more and more processes, you can reuse the services you developed for previous processes and gradually modernize all that is needed from the legacy application, making it relevant and essential for the future of the enterprise. The BPM approach provides quick time to value and engages business users. This is a welcomed refresher from IT-centric legacy modernization efforts!

**Acknowledgements**

The authors would like to thank their IBM customers and colleagues around the world for the shared experience with BPM over the last few years. In particular, we would like to thank Jenny Ang, IBM CTO for Enterprise Modernization, and two of the finest BPM architects on the planet: Scott Simmons and Stuart Jones!
Resources

- IBM Redbook: Scaling BPM Adoption: From Project to Program with IBM Business Process Manager
- IBM Redbook: Patterns: Integrating WebSphere ILOG JRules with IBM Software
- Rational Developer for i
- Service Oriented Modeling and Architecture (SOMA)
- Rational Developer for i
- WebSphere Adapter for IBM i
- Comment lines: Scott Simmons: Modernizing banking core systems
- Comment lines by Scott Simmons: Evolving approaches for connectivity and core banking systems
- developerWorks BPM zone: Get the latest technical resources on IBM BPM solutions, including downloads, demos, articles, tutorials, events, webcasts, and more.
- IBM BPM Journal: Get the latest articles and columns on BPM solutions in this quarterly journal, also available in both Kindle and PDF versions.
About the authors

Bertrand Portier

Bertrand Portier is an IBM Executive IT Architect. Over the last twelve years, Bertrand held various positions at IBM in development, consulting, or technical sales. Until March 2012, Bertrand was a solution architect in the BPM Tiger Team, working with customers on BPM discovery, architecture, governance, and roadmaps.

Marc Fiammante

Marc Fiammante is an IBM Distinguished Engineer Software Solutions Chied Architect for Europe in the office of the Vice President and CTO of the IBM Software Group for Europe, where he helps large organizations with their SOA and BPM transformations. He is a member of the International Standards Organization (ISO) Standards Committee (SC) 38, which consolidates and creates standards on web services, SOA and cloud. In this organization, Marc is one of the five global co-editors of ISO SC38 "General Technical Principles of Service Oriented Architecture." He is an IBM Master Inventor, a recognized thought leader and innovator, and is a member of the IBM Academy of Technology.

© Copyright IBM Corporation 2012
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)