Redbooks
ibm.com/redbooks

Business Process Management Design Guide
Using IBM Business Process Manager

Dr. Ali Arsanjani

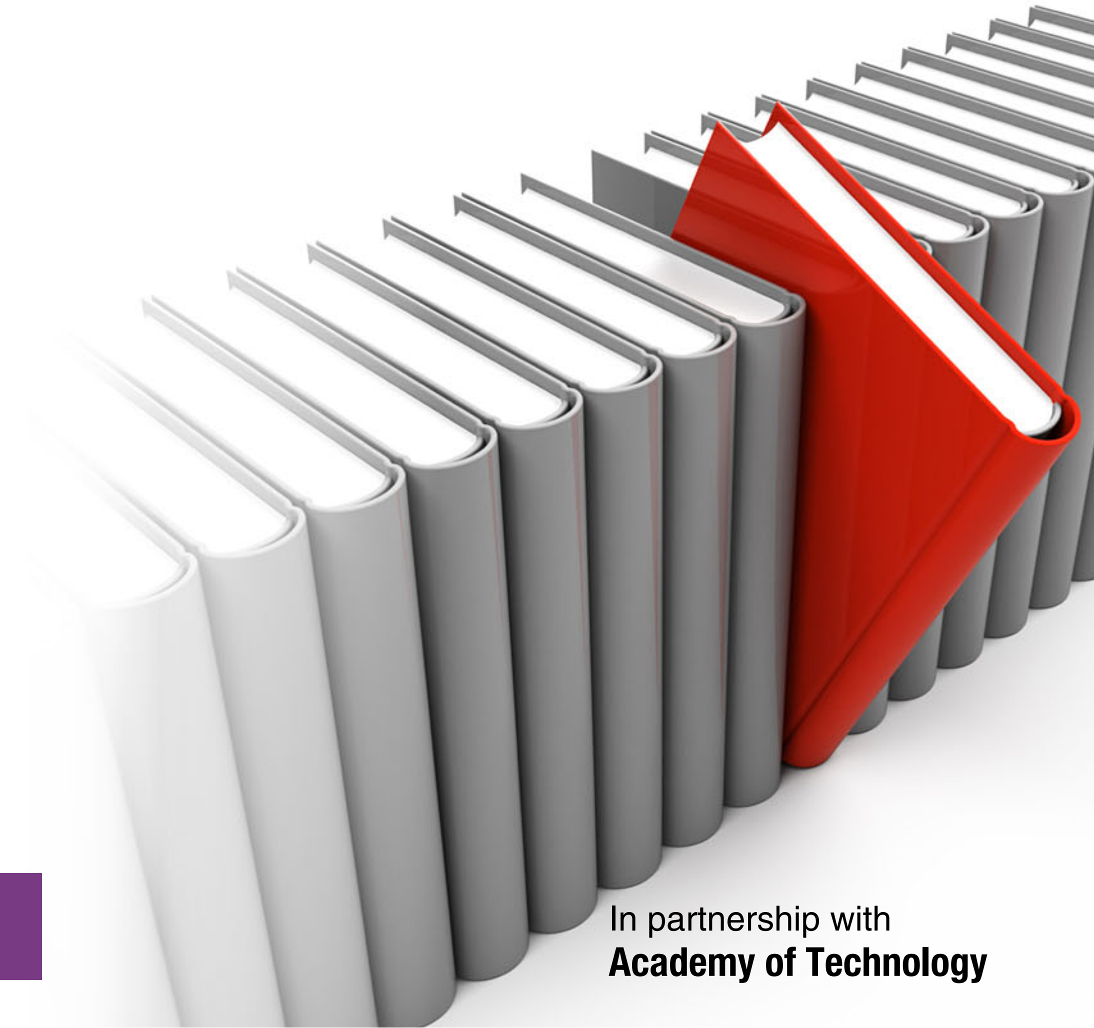
Nakul Bharade

Magnus Borgenstrand

Philipp Schume

J. Keith Wood

Vyacheslav Zheltonogov



WebSphere

In partnership with
Academy of Technology

IBM

Redbooks

International Technical Support Organization

Business Process Management Design Guide: Using IBM Business Process Manager

April 2015

SG24-8282-00

Note: Before using this information and the product it supports, read the information in "Notices" on page ix.

First Edition (April 2015)

This edition applies to IBM Business Process Manager version 8.5.5.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features described in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
Blueworks Live™
Cognos®
Component Business Model™
developerWorks®
Domino®
FileNet®
IBM®
IBM Component Business Model™
Lombardi Teamworks®
OpenPower™
Redbooks®
Redpaper™
Redbooks (logo) ®
System p5®
Teamworks®
WebSphere®
Worklight®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

IBM® Business Process Manager (IBM BPM) is a comprehensive business process management (BPM) suite that provides visibility and management of your business processes. IBM BPM supports the whole BPM lifecycle approach:

- Discover and document
- Plan
- Implement
- Deploy
- Manage
- Optimize

Process owners and business owners can use this solution to engage directly in the improvement of their business processes.

IBM BPM excels in integrating role-based process design, and provides a social BPM experience. It enables asset sharing and creating versions through its Process Center. The Process Center acts as a unified repository, making it possible to manage changes to the business processes with confidence.

IBM BPM supports a wide range of standards for process modeling and exchange. Built-in analytics and search capabilities help to further improve and optimize the business processes.

This IBM Redbooks® publication provides valuable information for project teams and business people that are involved in projects using IBM BPM. It describes the important design decisions that you face as a team. These decisions invariably have an effect on the success of your project.

These decisions range from the more business-centric decisions, such as which should be your first process, to the more technical decisions, such as solution analysis and architectural considerations.

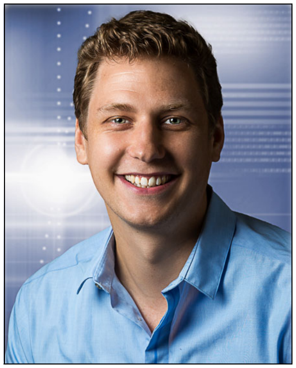
Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Poughkeepsie Center.

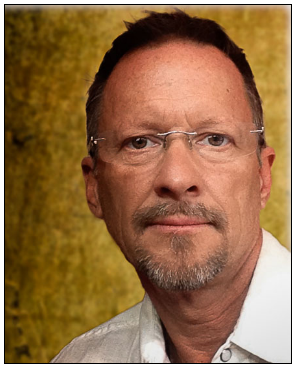
Dr. Ali Arsanjani is an IBM Distinguished Engineer and member of the IBM Academy of Technology. He is the Chief Technology Officer (CTO) of Smarter Process, service-oriented architecture (SOA), and application programming interface (API) Management for Software Services. Ali has over 30 years of experience in software development and architecture. He holds a PhD in Computer Science from DeMontefort University. His areas of expertise include patterns, component-based and service-oriented software architecture and methods. He has written extensively on patterns, SOA, component-based development and integration, business rules and dynamically reconfigurable software architecture, context-awareness, and BPM.

Nakul Bharade is a Senior Managing Consultant for IBM Software Group (SWG). He assists clients with requirements and strategies for developing and deploying BPM solutions. He helps clients align information technology (IT) to support BPM strategies by identifying governance and strategy frameworks. Nakul's current focus is on BPM methodologies, BPM and SOA governance, BPM performance, and tools that enable clients to define, assess, and deploy operational business processes. Nakul has a Master's degree in Computer Science and he is a candidate for Diploma in Strategy and Innovation Program at the University of Oxford.

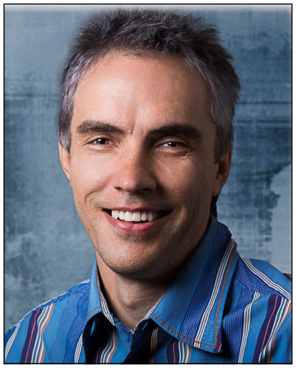
Magnus Borgenstrand is a Senior Technical Engineer with IBM North America. He joined IBM through the Lombardi acquisition in 2010 and quickly became the go-to person for things relating to BPM. He has over 16 years of experience planning and implementing BPM disciplines and solutions with global clients across many industries. His work has taken him and his family to North America, Europe, Asia, and Australia. His particular emphasis has been in bringing together both business and IT as a part of the methodology to ensure project success. His approach is to start with manageable projects and grow into enterprise solutions. Magnus attended Umeå University in Sweden and has a degree in Space Engineering. Magnus also studied Math and Computer Science at Stockholm University.



Philipp Schume is an IBM BPM Solution Architect. He delivered custom application development projects for IBM Germany before he made the transition to the IBM BPM Lab Services group in North America, where he worked in IBM BPM development and consulting. Since its creation in 2013, he has been a core member of the worldwide Smarter Process Center of Competency (CoC). Philipp is an expert in IBM BPM methods and technology, and his main focus is on delivery excellence. His main area of expertise is agile development in collaboration with business stakeholders. Philipp created assets and has written papers about business process modeling, design best practices, and BPM methodology.



J. Keith Wood is a Systems Architect and Security Team Lead in the IBM BPM services organization based in Austin, Texas. After more than 20 years as a software developer and solutions architect, he now consults with BPM clients to promote more secured installations and environments. Before joining IBM, Keith consulted with a wide variety of clients in the financial services, telecommunications, and retail sectors.



Vyacheslav Zheltonogov (Slava) is a Senior Managing consultant with IBM WebSphere® US Lab Services. Slava focuses on designing scalable BPM solutions and promoting BPM methodology and best practices in software architecture and design. As a member of the Worldwide Smarter Process CoC group, Slava helps clients with troubled and challenging implementations as they make the IBM BPM journey. Slava has over 17 years of experience spanning a range of IT operations and solution architecture; extensive background and expertise in SOA, design patterns, BPM, operational decision management, and business intelligence. Before joining IBM, Slava spent several years working as a solution architect in the area of the US Tax domain on the Java and Microsoft .NET platforms. In this area, he was responsible for the architecture and delivery of custom web-based solutions. Slava is an Expert Certified IT Specialist and holds Master's degrees in Mechanical Engineering and Economics from Volgograd Polytechnic Institute in Russia.

Special thanks go to Brian Petrini, Technical Product Manager IBM BPM, and Kim Clark, IBM Certified IT Specialist SOA Foundation, for their valuable input to the overall structure of this Redbooks Publication, and for their course covering "Integration and SOA Design".

Thanks to the following people for their contributions to this project:

Axel Buecker, Project Manager
ITSO, Austin TX

Sam Antoun, Ayyappan Chandran, Mike Collins, Francis Friedlander, Gab Gambassi, Steve Lindauer, Paul Pacholski, Jim Pokoj, Martin Ross, and Jimmy Xu
IBM

Now you can become a published author, too

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time. Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and client satisfaction, as you expand your network of technical contacts and relationships. Residencies run two - six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Learn more about the residency program, browse the residency index, and apply online:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us.

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form:

  ibm.com/redbooks

- Send your comments in an email:

  redbooks@us.ibm.com

- Mail your comments:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Introduction to successful business process management

This chapter provides a high-level introduction to the difference between business process management (BPM) and a BPM suite, such as IBM Business Process Manager (IBM BPM). In addition, it describes the value of a BPM suite, and where it applies.

We also share our experience regarding the design decisions you face, from a business perspective, that affect the outcome of your BPM project. This chapter is targeted at business people and project managers that are new to BPM.

We explain all of these topics in the following sections:

- Introduction to business process management
- Influences on a successful BPM project
- Things to consider before a project starts

1.1 Introduction to business process management

In this section, we cover the *what*, *when*, and *why* for BPM solutions.

1.1.1 What

First, it is important that we emphasize the following statement: *BPM is not a product or a technology*. BPM is a comprehensive management approach to managing and improving the efficiency and effectiveness of business processes across the enterprise.

The goal of a BPM suite (a software solution) is to promote effectiveness and efficiency in your business processes. The BPM suite accomplishes this by using measurable business value to align all of your projects with your corporate strategies, alongside the BPM approach.

BPM relies on an iterative, or agile, delivery methodology that creates process visibility and enables process control in your business. The intention of a business process initiative is to deliver targeted results that directly support the strategic goals of the business. Therefore, a successful BPM initiative requires close collaboration between business operations and technologists. A successful BPM program ties all BPM projects to core business initiatives.

BPM enables you to manage your processes and support corporate initiatives, such as improving product quality, reducing time-to-market, expanding to new markets, raising client satisfaction, increasing profit margins, and so on.

The approach of BPM can be, and many times is, applied without any BPM software. In fact, as you will see later in this chapter, the management approach of looking at and improving your business processes are much older than the software that supports it.

Organizations gain value out of documenting, understanding, and analyzing their processes for efficiency and effectiveness. This is typically done by business analysts, who either talk to the people that are involved in the process or get all stakeholders together in a room for several days to document and analyze the process to see where it can be improved.

This book is mostly about the technical design decisions that affect the outcome of implementing BPM solutions.

As you will see later in this chapter, however, the success of BPM projects is not affected only by the technical decisions that you make. Other aspects, such as the involvement of business, aligning project goals to business outcomes, the methodology chosen, the selection of the first process, cultural change, and so on, also have a big effect on the success. In this first chapter we examine these non-technical aspects.

For a more detailed description of how to adopt BPM across your organization and ensure its success, see the IBM Redbooks publication *Scaling BPM Adoption: From Project to Program with IBM Business Process Manager*, SG24-7973.

Remember that the main objectives with any project using a BPM suite are ultimately to gain business agility, reduce risk, save money by removing steps that do not add value, reduce rework, automate certain steps, and so on.

Before we describe the details, though, we briefly consider the history of BPM.

A brief history of BPM

One of the first people to start looking at process improvement was *Frederick Taylor*[1] (1856-1915), who in the early 20th century sought to improve the efficiency of industrial processes. His work was focused on standardizing processes by studying them from a scientific point of view, first measuring and then improving.

He was also one of the first to push for employees to be trained and developed as a way to improve efficiency. This was focused on training the employee to do one thing only and do it well. Taylor’s work was focused on efficiency, or to put it another way, *do one thing correctly*.

In the mid 1980s, Motorola introduced *Six Sigma*[2], which is a set of techniques and tools for process improvement. Six Sigma is focused on the improvement of process output by removing errors and variations in processes.

Lean[3] was derived from Toyota Production System in the 1990s, and is concerned with the elimination of waste. From a business process perspective, we analyze business processes, classify each step, and remove steps that do not add value. This business process can lead to savings due to reduced processing time. This approach targets the effectiveness of business processes, or to put it another way, *do the correct thing*.

[1] http://www.britannica.com/EBchecked/topic/584820/Frederick-W-Taylor

[2] Michael George, David Rowlands, Bill Kastle, What is Lean Six Sigma? (McGraw-Hill Professional Publishing, October 27, 2003)

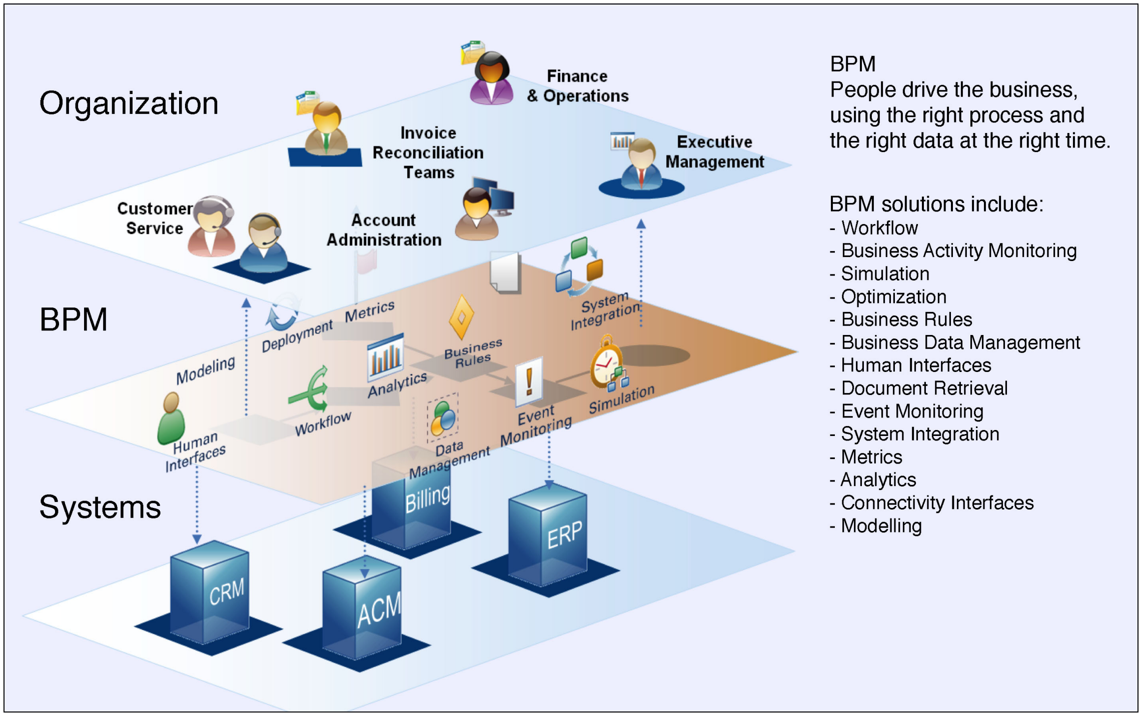
[3] James P. Womack, Daniel T. Jones, and Daniel Roos, The Machine That Changed the World: The Story of Lean Production – Toyota's Secret Weapon in the Global Car Wars That Is Now Revolutionizing World Industry (Free Press, March 13, 2007)

Today, organizations are using BPM together with Six Sigma and Lean to improve efficiency and effectiveness of business processes. For more information about how to best apply Lean, Six Sigma, and BPM, see the IBM Redpaper™ publication *Applying Lean, Six Sigma, BPM, and SOA to Drive Business Results*, REDP-4447.

Enter business process management software

In the 1990s, we saw the emergence of BPM software to help analyze and run processes. Modern BPM suites have advanced considerably since their infancy in the 1990s, when they were based around simple modeling and basic workflow routing. For example, users might be asked to complete a certain task, such as log in to system *x* and complete an order. The main benefit was that the process could be controlled, and that you gained visibility over performance.

Today, BPM suites, such as IBM BPM, are built to support organizations on their entire BPM journey, from the smallest initial projects to enterprise-wide adoption. A modern BPM suite covers many different aspects, including workflow, user interfaces (UIs), reporting, data modeling, integration, and so on. Figure 1-1 shows the different aspects of a modern BPM tool.



BPM
People drive the business, using the right process and the right data at the right time.

BPM solutions include:

- Workflow
- Business Activity Monitoring
- Simulation
- Optimization
- Business Rules
- Business Data Management
- Human Interfaces
- Document Retrieval
- Event Monitoring
- System Integration
- Metrics
- Analytics
- Connectivity Interfaces
- Modelling

Figure 1-1 The different parts of a modern BPM tool

Any modern BPM suite supports the full lifecycle of continuous process improvement. It is able to support a project through process discovery and documentation, modeling, execution, and not least process improvement. A key aspect is also the support of the collaborative nature of BPM, where business people can easily be engaged. BPM supports the following phases:

- Discovery phase

  A key aspect of any project is process discovery, during which the business user is involved in documenting processes. This phase is typically owned by a business analyst who involves employees that are part of the process, or know the process very well.

  This phase of a project is very much focused on documenting and understanding the as-is business process. After that, you look at the to-be process model, which includes potential improvements that can be made to the business process.

  Process discovery has historically been done by having all the stakeholders “locked in a room” for a few days or a week with a stack of post-it notes that is put on the wall or on a roll of brown paper. The post-it notes are moved around until you can agree on the correct business process. At the end of the workshop, you document your process in a modeling tool like Microsoft Visio.

  There are several issues with this approach:

  - To make this a collaborative effort, which is important if you want input from everybody involved, you really need to be physically in the same location, which can be costly for organizations that are spread across several cities or countries.
  - Much manual effort needs to go into the management of all process diagrams and their multiple versions.
  - You also need to manually merge different versions of a process when somebody creates a copy to make some changes locally.

  To make this phase easier and ensure the involvement of business users, it is key that the selected tool is easy to use. It is also important that business users can collaborate at any time, no matter where they are located. You really want business users involved throughout the project, and not just for the first 3 - 4 days. For ease of access and interaction, therefore, it is advised that you use a collaborative process discovery tool, such as a cloud-based tool.

  The tool should also be socially enabled, to support collaboration and help with the organization and management of processes as you build your enterprise process inventory. This is especially important when you take on BPM from a broader organizational perspective, and start having several teams or departments documenting their processes.

- Modelling

  Any BPM tool should not only support the modeling of processes. It is also important that the tool supports the modeling of other aspects of a process that is going to be executed, including but not limited to the following aspects:

  - UIs

    The tool should support modeling both UIs, where the user interacts with the process, and the flow of these UIs, designed to guide the user through the task.

  - Data model

    The tool should also support the modeling of the data model that describes the business data that flows through the business process.

  The BPM suite should also inherently support an iterative approach to development that involves the business. With this, we mean that it should be easy to involve business people as you start to develop the process in more detail, to gain their feedback. This is done by making the development tool graphical and easy to understand, so that business people can follow the process, as opposed to showing them business code.

  Part of the iterative approach is the need for the tool to support ease of demonstrating what you are developing with a single click. This enables you to show what you have done, and to more quickly make changes and play the process back to the user. At IBM, we call this approach a *playback*. After each iteration, you play back the process to stakeholders and business users so that they can provide feedback.

- Execution

  This is a given with any BPM solution. One important aspect of the execution is that there is always a 1:1 relationship between what you are modeling and what you are executing. For human-centric processes, it is preferred that there not be any need for compiling or coding to make the process run.

  The reason for this is that you want to be able to play back what you are modeling quickly to users, as mentioned in the previous section. However, for system-centric processes, such as straight through processing (STP), compiling is preferred due to performance aspects.

- Process improvement

  One key aspect of BPM and the use of a BPM suite is that you want to be able to improve your processes by first modeling them, then executing them and collecting performance data that enables you to analyze the processes for improvement. Because you are working with actual performance data that has been collected during run time, you are no longer estimating or guessing.

  What you are really accomplishing is that you are continuously improving your processes over time. This means that a process is never done and finished. There are always improvements that you can apply to the process because of changing competitive pressures, new regulatory requirements, or other improvements that you find.

  With this in mind, it is important that the BPM suite supports ease of changes to a process. It should also have built-in capability to capture process performance as the process executes, reporting for viewing operational real-time reports and historical reports so that you can look at historical trends.

  The tool should also support analysis of historical data to enable you to find bottlenecks and other historical trends. If you do not have these built-in capabilities, you have to manually capture and analyze all of this data.

1.1.2 When

Now that we know a little more about BPM and how it can be used with a BPM suite, the next question is, when is a BPM suite a good fit for a project? This is a very important question, because in our experience, this can have a big effect on if your projects are successful or not. We have seen projects fail where a BPM suite has been used more as an application development tool in isolation without the iterative approach of BPM.

The issue here is typically that business users are only involved up front when the initial requirements are captured, and do not see anything until the project goes live much later. However, requirements change frequently these days. This does not mean that if you use a BPM suite as a pure application development tool your project is doomed to fail, but in our experience the risk is higher.

Now, examine what typical problems you would find in a process that is executed in a manual and unstructured way compared to what a BPM suite offers.

Figure 1-2 shows the typical issues that you face without a BPM suite, and the value that such a suite brings.

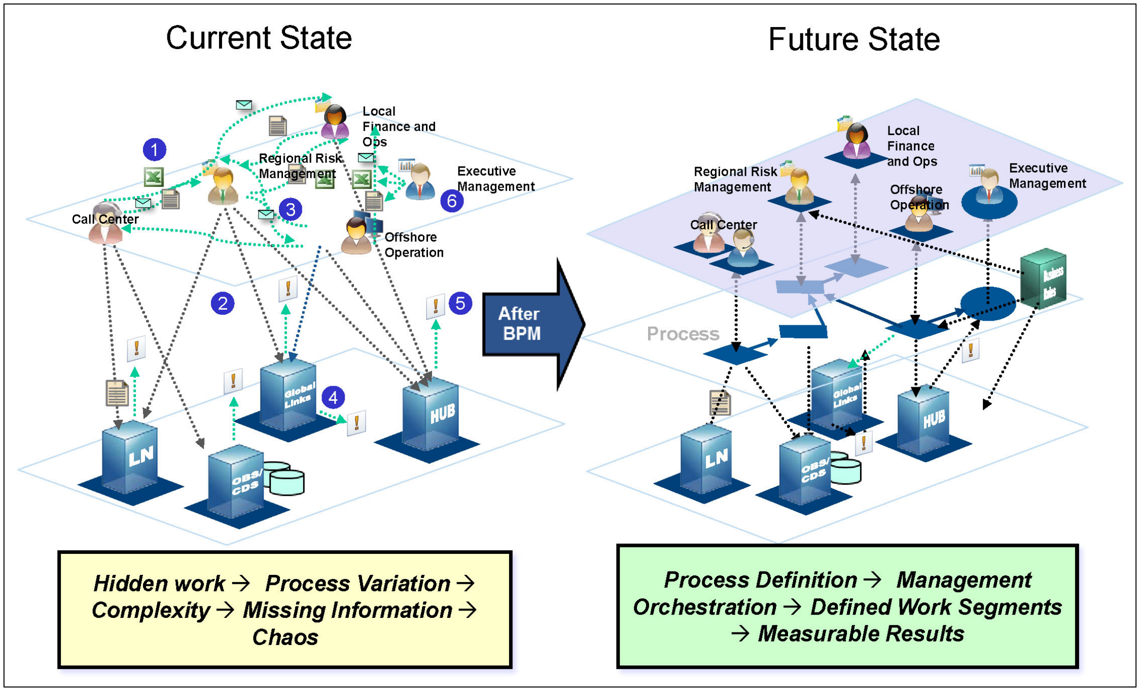


Figure 1-2 The effect of BPM

It is amazing how many business-critical processes today are still managed and run manually. We would even go as far as to say that the most-used tools for running processes are email and spreadsheets.

There are many issues when you are running a process manually:

- Visibility

  Visibility into your processes becomes difficult if not impossible when you either run your process manually, or the process spans multiple enterprise systems or departments. One option is to try to manually capture the data needed, such as processing times and so on.

  However, this is typically difficult and error-prone. From an operational perspective, it becomes challenging for process owners and people working with the process to get an overview of how they are doing.

  For example, from a team manager’s perspective, it becomes difficult to get an overview of what the team is working on. Who is working on what, and how much work do they have? This can have a big effect on client satisfaction.

- Process variation

  With manual processes, you are relying on the user to know what they should do and what the next step in the process is. This invariably leads to many variations in how the process is executed. In turn, variations increase the risk from a compliance perspective, because you cannot guarantee the flow of the process. It also means that it is difficult to measure and improve the process, because you cannot be sure that you are comparing apples to apples.

- Missing information

  One of the biggest issues with manual processes in our experience is rework, because this is both time-consuming and costly. Rework in manual processes is made much worse because users have to manually enter or reenter data into multiple systems. This many times leads to the wrong data being entered or even missing all together. The outcome of this is rework.

- Hidden or missing work

  With manual processes there is always a risk of missing or losing work, especially if you rely on emails, because these can easily be missed or deleted. This in turn can lead to work not being completed on time, which in turn has an effect on client satisfaction.

So, how can a BPM suite help to mitigate these issues? A BPM suite introduces a layer of abstraction between the user and the back-end systems. The process interacts with the back-end system, typically with an enterprise service bus (ESB), and interacts with the users through built-in UIs. Therefore, the user only has one interface to work with, and it presents the information that the user needs to complete the task.

The process is modeled and executed within the BPM tool. The benefit here is that the same process is always executed, minimizing risk. Put in another way, the processes becomes predictable, repetitive, and easy to measure. The process also handles the routing of work to make sure that the correct person always receives the correct task. A BPM suite also gives you end-to-end visibility across the process.

Describing a business process

Previously, we introduced when to use a BPM suite. Next, we want to focus on *what* a business process is. Note that many different types of processes exist, and not all of them are business processes that are suited for a BPM suite. Therefore, the type of process that you are planning to implement has an effect on whether the process is suitable for a BPM system.

The following list describes several key aspects that highlight a *business process* that is suitable for a BPM suite:

- Performing the process *provides value to the business*.
- The process follows a relatively *structured path*.
- The process contains individually *business-relevant steps*.
- The steps within the process are performed by *multiple roles and teams*.
- *Business-relevant data* flows through the process.
- The process *changes over time* as a result of changes in the business.

1.1.3 Why: The value of BPM

In the previous sections we introduced what a BPM suite is and when you should use it to gain maximum business value. In this section, we talk about why you should use it. What is the business value of a BPM solution? We have divided this section into the following five value points:

- Business agility
- Visibility
- Compliance
- Workforce efficiency
- Process governance

Note that this is not intended to be an exhaustive list of value points.

Business agility

The business world we live in today is changing faster than ever. To stay competitive, companies have to change quickly or see themselves be overtaken by competitors that can adapt to new business needs more swiftly. Because of this, businesses are becoming more and more demanding as to how often and how quickly they need changes.

This puts extra pressure on information technology (IT) organizations that are typically already overworked and have limited budgets. It is in this environment where BPM can help, with its iterative approach of delivering small incremental releases and value often. IT can also offload some of the work, such as process discovery, to business. This means they can concentrate on the tasks that add more value.

The historical approach to application development has been more targeted at a waterfall methodology with long development cycles. This involves getting all of the requirements up front and then getting into a longer development cycle of 6 - 12 months or longer. An issue with this is that when the application is eventually delivered, it is not unusual for the requirements to have changed due to altered business needs.

Changes have to be done more rapidly to meet the ever-changing competitive landscape. The effect of this has been that IT struggles to keep up with modified business requirements.

Not only can you develop new processes and change existing processes quickly. BPM also helps to align project goals with the business needs, because business people are involved at all stages during the development cycle.

Figure 1-3 shows the typical phases in the standard playback methodology.

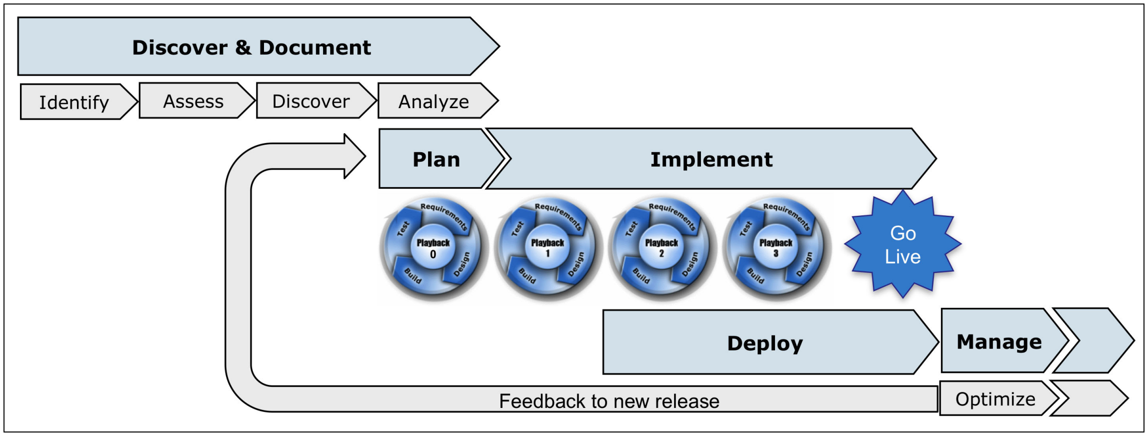


Figure 1-3 Iterative development with BPM

For more information about how to best apply the playback methodology, see the IBM Redbooks publication *Scaling BPM Adoption: From Project to Program with IBM Business Process Manager*, SG24-7973.

Visibility

One reason many companies start looking at BPM is the need for end-to-end process visibility. It is key to have reliable performance metrics if you want to start improving your processes with a degree of certainty. Some clients have processes that span multiple vertical systems. Each of the systems might have workflow and reporting built in, which provides visibility into the system. However, they struggle to get full end-to-end visibility.

This becomes even harder when the process also spans multiple departments. A simple example to illustrate this would be the onboarding process for a new employee. One might think that this process mainly involves human resources (HR). However, it typically touches many other departments and systems, and this makes visibility across the process difficult without a BPM suite.

The following list provides some examples of the departments that the onboarding process typically touches:

- Security. Enable access to building and issue a badge.
- Human resources. Conduct interview, negotiate contract, get contract signed, and add new employee to the HR system.
- IT. Provide a notebook and set up in IT systems.
- Facilities. Give desk to sit at, and so on.

From an operational point of view, BPM enables users to review the performance of processes in real time and take actions. This could be a user reviewing all of the tasks that she is working on to understand what is the most important item of work. This can be classified by priority assigned to work, or from a time perspective, such as what is due today, tomorrow, and so on.

After a business process is automated, a majority of the value that a business garners out of it is in the understanding of the process, and the ability to improve it based on that knowledge. Too often, enterprises attempt to optimize processes in the absence of real performance metrics. A BPM system enables you to understand where there are issues in the process, and how to improve the process.

Compliance

Organizations today must comply with an ever-growing set of government regulations. Not only is the number of regulations growing, but the current set of regulations is often not that clearly defined. They also change often. Regulations are getting increasingly complex, as special cases and exceptions surface.

Moreover, as a consequence of globalization, organizations now have to comply with multi-country regulations, further increasing complexity. All of this brings a new set of requirements that organizations must adhere to.

Compliance is not a competitive area for organizations, because all of the organizations in a specific industry have to adhere to the same set of regulations, and regulations are not a driver for growth. On the contrary, they can have a negative effect on business.

For instance, some financial institutions cannot afford to onboard US clients, because the Fair and Accurate Credit Transactions Act (FATCA) imposes so many constraints. With this in mind, organizations are searching for ways to avoid unnecessary costs related to regulations.

A BPM tool can help organizations to comply with regulations, because it enables easy and quick changes. It also helps with auditability and traceability of processes. From a broader perspective, it helps organizations to keep costs down when you are required to adhere to many different regulations, because a BPM solution is a platform where you can implement any process across the organization to comply with regulations.

We have seen organizations buy a specific application to cover one specific regulation for example FATCA. This quickly becomes unmanageable and expensive, because organizations have to cover all applicable regulations. What clients need is a single platform for addressing all compliance needs, while enabling auditability, agility, and measurable quality of service.

In fact, we have seen an increase over the last couple of years in organizations offering business process outsourcing. These organizations offer compliance processes for many different regulations for example FATCA, The Sarbanes-Oxley Act (SOX), Know Your Customer (KYC), Payment Protection Insurance (PPI) Remediation, and so on. The case is very compelling for both client organizations and service providers.

Client organizations can focus on their core business, and service providers can provide compliance as a service and do repeat business, because regulations are the same for all clients. They can come with a unique value proposition where they provide high-quality service. Again, service providers need to come to the market with a uniform platform and practice, and visibility and agility are strong value points.

The following list includes some of the most important aspects where a BPM solution helps organizations with regards to compliance with regulations:

- Agility

  Due to the fact that a BPM suite supports quick and iterative development, it enables organizations to quickly change their processes and adhere to ever-changing government regulations.

- Visibility

  When implementing processes, organizations gain a new level of visibility with regards to how processes are implemented. This makes it much easier for organizations and auditors to make sure that they adhere to regulations. Chief Compliance Officers also need such visibility, because they are getting more constraints on their agenda. Remember that, with a BPM solution, what you are modeling is also what you are executing.

- Traceability

  There are two main aspects about traceability where a BPM system can help:

  - First, it helps to track who changed a process from a development and change perspective. Not only that, it also helps to track when the change was made and what was changed, and link the change to a change in the regulation. This is key from a traceability perspective.
  - Secondly, it also enables organizations to review historical process data to enable you to trace how the outcome or decision in a process was made. Such capabilities are frequently a requirement expressed in regulations, for instance for enforcing consumer rights.

- Orchestration

  By modeling the process in a BPM system that enforces the rule that what you are modeling is also what you are executing, you can ensure that this is true for the process that you have modeled. Also, because the process is executed in a system and not by a user, the process becomes predictable and repetitive. The same process will always be executed. This means that you are reducing your risk of being in breach of compliance.

- Auditability

  One important aspect of compliance to regulations is the ability to go back in time to understand who did what, where and when in a process. A BPM system captures a complete audit trail that enables organizations to do this.

Workforce efficiency

One of the issues with manual processing is that you very much rely on the individual employee and their knowledge of the process. Every employee makes slightly different decisions, which means the process isn’t exactly repetitive and predictable, which leads to increased risk.

This also means that it becomes more difficult to improve the efficiency of a process. One aspect of this is the need to minimize the amount of time in the process being wasted trying to figure out who should have the work item after you complete it.

This might not be such a big issue for the “heroes” in an organization that have in-depth knowledge and have worked with the process for a long time. This is often referred to as the *hero culture*, where a few very knowledgeable people become the people that do much of the work. These same people help and guide other employees.

Hero culture often makes it hard for organizations to grow, because they rely on a few employees with in-depth knowledge of the processes. In fact, a BPM system can help to encode the knowledge and best practices from these heroes to enable other users to get up to speed faster and become productive.

The following sections provide information about some of the aspects regarding workforce efficiency and the benefits of BPM.

Routing

With manual processes, it is the employee's responsibility to know who to route the work to next. For a simple process, such as applying for time off, it is typically reasonably simple. You complete the form and email it to your manager, who approves (or declines) and then sends it on to HR. However, if you consider more complex and business-critical processes, such as a claims request, it becomes more difficult.

The reason for this is that the routing of a specific work item is many times based on many different attributes, for example the type of claim, client status, severity, location, priority, and so on. It also depends on the skill and availability of the employees handling cases.

This leads to wasted time as the employee looks up instructions or sends it to the person the he thinks should have it. That person in turn passes it on to somebody else that should have it. Even worse, the person you send the work to might be out of the office and so cannot do the work. The consequence of this is that the work sits in an inbox for days or weeks without being actioned.

A BPM suite enables you to encode the routing rules in the process. Two of the benefits are that you reduce the risk, because you always get the same result from the routing rule for the same input. You also speed up the process, because the task is routed to the correct person immediately.

Another benefit with routing rules that are encoded in a BPM suite is that you can also start using more complex rules based on the skills and workload of employees. An example of this could be when a claim comes in to the claims department, the process first executes a routing rule that takes into account the complexity of the case and what skill is needed.

It can also look at who of the people that can actually handle the claim has the least work, and then route it to that person. You could even implement the concept of *follow the sun* if you have employees in different time zones. An example of this is the automatic reassignment of work from a call center in New York to San Francisco at the end of the day in New York.

The fact that you can automate the routing rules is great, but it does not cover all cases. There are always cases when you need to manually assign or reassign work. This is especially important as you evolve your BPM solution. Remember that version 1 is typically not the complete end-to-end solution as you deliver smaller releases faster with BPM.

For example, in version 1 you might have decided to only implement simpler routing rules that cover 80% of the cases. You refine the rules in version 2 when you also have some usage data from version 1. This is why you always need the capability to manually perform overrides.

Because a BPM suite automatically keeps track of all work, it enables the system to make business dashboards available with information about what is going on. This enables a business user, like a team manager, to first get an overview of what the team is working on before deciding how to assign or reassign work.

Parallel work

This one is probably rather obvious, but it is still worth mentioning, because it has an effect on the efficiency of the process. You can gain process efficiency by performing work in parallel. If certain work items or tasks do not rely on each other, you can speed up the end-to-end process by performing them in parallel.

Data entry

Another issue with manual processes is that many times they span multiple back-end systems. Different tasks in a process need access to different systems. In addition, users must interact with multiple back-end systems to get all of the information that they need to complete one single task.

This involves them having to log on to multiple systems to collect all of the data manually. They do this using what we refer to as the *swivel chair approach*, which is looking at one screen and then swiveling over to the other system to type the data in.

This introduces several issues:

- The users waste time logging in and navigating multiple systems. This becomes even worse if the user is new to the system, because they have to try to find their way.
- The windows in the different systems are also typically not designed with the specific task in mind, so much of the data on the window is not needed. This might confuse the users, leading to either them spending extra time figuring out what is needed, or just guessing.

  The outcome of this is that many times the user collects the wrong data, or even misses some of the data altogether. This invariably leads to rework in the process, which is both time-consuming and costly.

A BPM solution solves these issues by presenting one consistent interface that guides the user through the immediate task. This removes the need for the user to interact directly with multiple back-end systems, because this is handled by the BPM system.

The window in the BPM system is also designed with the specific task in mind, and removes any unnecessary data that clutters the window and can confuse the user. This reduces the chance for errors in the process, which in turn leads to less rework and in the end quicker completion of processes.

Decisions

Routing rules are not the only decisions being made in a process. Frequently, you also have process rules that affect the path that the process takes. An example of this could be where a claim needs to have extra approval if the compensation in the claim exceeds a certain value. The benefit here is very similar to the routing rules. It enables you to encode the rules to ensure the same outcome every time they are executed with the same input.

Knowledge sharing

In the last couple of years we have seen an increase in the use of social capabilities within BPM. By socially enabling your processes with capabilities, such as message feeds, screen sharing, and so on, you enable your more experienced employees, the heroes, to collaborate and share their knowledge with other users. The benefit is that new users can complete the work faster and with less error.

An example could be an employee that is new and does not exactly know what to do, or it could be that she just wants to ask for help from an experienced user to make sure that she makes the correct decision. Without a socially enabled BPM system, they can either sit and try to read online instructions, or use the phone to call a more experienced user if they know who that is. In both cases, they would waste time. Also, any communication outside of a tool would not be logged.

Process governance

As we have mentioned before, one of the values of BPM is that you can increase the quality of your processes by modeling them and then executing them in one single system. A BPM system can also help to drive commonality in processes across the enterprise. An example of this could be an approval process that is the same for many higher-level processes.

With a BPM suite, you would implement the approval process once and reuse it many times. This speeds up the development of the process, and makes sure that you handle an approval the same way every time.

A big part of the governance aspects in a BPM program is the establishment of a BPM Center of Excellence. For more information about the value of a Center of Excellence see 1.2.2, “Center of Excellence” on page 19.

1.2 Influences on a successful BPM project

As you would imagine, there are several factors that influence the outcome of a BPM project. Some of these factors are pervasive across IT, and others are more specific to BPM. Our experience is that it is not only technical aspects that have a bearing on a BPM project. In this section, we will spend some time describing the non-technical aspects of successful BPM projects.

1.2.1 Project delivery methodology

One key decision for any successful business process project or program is *what delivery methodology approach to use*. To take full advantage of the promise of BPM such as business agility, it is our strong belief that it is key to use an agile approach to delivery.

For many clients this is a big step, because they are used to a waterfall-based methodology in which you spend a fair amount of time up front gathering and documenting all of the requirements before handing off to IT for the implementation. The issue with this approach is that internal and external business requirements change quickly these days.

We believe that it is much better to deliver incremental value in small steps. This enables you to adjust quickly to new requirements as they come up.

If you are totally new to BPM, there are also some unique challenges that you should be aware of. The following sections describe some of the common issues on a higher level.

For a more detailed description of how to approach BPM and ensure its success, see the IBM Redbooks publication *Scaling BPM Adoption: From Project to Program with IBM Business Process Manager*, SG24-7973.

One approach that we have seen that initially can look promising is to document all of your most business-critical processes and start implementing all of them at the same time. The idea behind this is typically that organizations are trying for the maximum effect. However, we would strongly advise against this approach, because we have seen it fail many times, for several reasons.

If you are new to BPM, there is a learning curve as you adapt to the new software, as with any software. Also, if your organization is not used to an agile and iterative approach to delivery, there is a certain amount of cultural change that has to take place. Furthermore, it can be difficult to get the full value out of reusing assets from other projects, and there can be some amount of rewriting necessary later to cater for reusability.

We advise you to always start small: Do not try to “boil the ocean” in one attempt. The first project should always be a process that is smaller in scope but still delivers business value. It is important that the first project is successful for your continued rollout of BPM. Success breeds success.

Note also that even if the process that you are planning to implement in a BPM tool is a large complex process, you can still approach this by breaking it up into several smaller deliverables or iterations. For example, for the first version you can still get value by implementing a *thin* process that just routes the work and prompts users to complete a task and then report back. From this you would still get visibility and understand where bottlenecks are, and so on.

Or you might opt to implement a part of the process that is currently causing issues and go deeper. In iteration 2 you might start plugging in integrations to automate certain steps.

Remember that it is almost always better to deliver some business value in, for instance, 90 - 120 days than it is to wait for a more complete solution in 9 - 12 months. Another aspect to remember is that in today’s world it is almost certain that business requirements will have changed anyway in the 9 - 12 months.

One of the positive effects of delivering a project in small incremental steps is that the change for the user is kept smaller, which means better user adoption.

More information about delivery methodologies and approaches can be found in Chapter 2, “Approaches and process discovery” on page 25.

1.2.2 Center of Excellence

Although most BPM projects begin as individual, loosely connected (or entirely disconnected) efforts, today’s operational landscape demands scalability and enterprise-wide adoption, which eventually necessitates bringing individual BPM projects together in a consolidated BPM program. To meet the demand of scalability and enterprise-wide adoption of BPM, many organizations today implement a BPM Center of Excellence (CoE).

A CoE must address the following key focus areas of responsibility:

- Defining a higher business goal or vision, driving BPM initiatives, and aligning individual projects with that vision
- Executing a scalable delivery resource model for discovering, implementing, deploying, managing, and supporting BPM initiatives
- Administering a shared infrastructure for hosting and maintaining the solutions that are the outcomes of BPM initiatives

Through experience with both successful and challenging BPM program initiatives, we learned that the following aspects are true:

- A BPM initiative can survive only by achieving business value, and business value must support the strategic objectives of the organization. Business value must be measured objectively with supporting data and be easily visible and communicated to leadership. Without demonstrating business value, the BPM journey will end, or stagnate at best.
- The transformative nature of BPM requires a shared infrastructure, called a BPM system (BPMS) that scales with a growing demand for BPM projects. This shared infrastructure must support the collaborative aspects and governing demands of BPM as a discipline.
- The purpose of a BPM initiative is to create a repeatable delivery model for improving business performance. Long-term success depends entirely on establishing a scalable BPM delivery model as a discipline. Focusing on organizational enablement in BPM methods is essential for an uninterrupted BPM journey. Without BPM method enablement, even the best BPM suite achieves no value.

The previously mentioned points illustrate why strong executive sponsorship and effective working relationships between all departments in an organization are essential. The charter of the BPM CoE is to ensure that these conditions exist.

The CoE has representatives from both business and IT. This is especially important when it comes to project selection, and when deciding the scope of a project or iteration.

For a more detailed description on how to create a BPM Center of Excellence see the IBM Redbooks publication *Creating a BPM Center of Excellence (CoE)*, REDP-4898.

1.2.3 People

One key aspect of any successful BPM project is the active involvement of business users and stakeholders throughout the length of the project.

An often under-appreciated factor in the success of BPM projects is the people that are involved. Frequently, if stakeholders are more used to a waterfall approach, they are rather keen to get all of the requirements into the first release. They expect that the likelihood of it being delivered in a later release is slim, or that it takes a long time for it to be realized. There is therefore always a tendency for business people to try to fit as much as possible into the first release.

The big risk with this is that very quickly the scope starts to swell and the delivery date moves. It falls on the project manager to control scope. It is therefore important that the project manager or the executive sponsor have the power to make the final decision about what is in or out of scope for the current release.

1.3 Things to consider before a project starts

In this final section of the first chapter we want to raise some issues that we have seen in projects where we have been involved. These points might seem trivial but they have a real effect on projects. If you can avoid them, you can save yourself from additional headaches.

The first set of points are about getting the people that are involved in the project access to the correct resources so that they can do their work. The second section describes things that are typically not thought about at the start of a project, but invariably affect the success of the project.

1.3.1 Access to resources

We are keen to highlight these seemingly simple administrative points because we have seen our fair share of projects being delayed by them.

The first thing is to make sure that all developers that are involved in the project have access to all of the correct resources so that they can do their job. This includes but is not limited to the following resources:

- Access to logs on servers for debugging
- Access to servers where your BPM server is installed for eventual server configuration changes, for example, log level

Access problems can typically be avoided by having a good onboarding process in place, especially if you are using external consultants that need access to resources. The onboarding process would also need to make sure that the external consultants have access to the following components:

- Ethernet
- Development machine
- Server access

1.3.2 Often overlooked

In this section we are going to have a quick look at certain aspects of a BPM project that might be overlooked. In the case of reporting, it is often the first thing that gets cut when the project is running out of time.

Performance

Believe it or not, a project aspect that sometimes gets overlooked is performance. This can, for obvious reasons, have a massive effect on the acceptance of a new process that you deploy. Business users are not going to be happy if the system is slow. There are many aspects that you must look at during the project regarding the performance of a BPM system:

- Architecture best practices

  For a more detailed description about best practices for IBM Business Process Manager topologies, see the IBM Redbooks publication *IBM Business Process Manager Version 8.0 Production Topologies*, SG24-8135.

  For more details about architecture best practices, see Chapter 3, “Solution analysis and architecture considerations” on page 73.

- Development best practices

  For a more detailed description about best practices for developing Coaches with IBM BPM, see the IBM Redbooks publication *Leveraging the IBM BPM Coach Framework in Your Organization*, SG24-8210.

  For a more detailed description about best practices for developing mobile user interfaces with IBM BPM, see the IBM Redbooks publication *Extending IBM Business Process Manager to the Mobile Enterprise with IBM Worklight*, SG24-8240.

  For more best practices for IBM BPM, see the BPM section on IBM developerWorks®:

  http://www.ibm.com/developerworks/bpm/

- Performance tuning and configuration

  For a more detailed description about best practices for performance with IBM BPM, see the IBM Redpaper publication *IBM Business Process Manager V8.0 Performance Tuning and Best Practices*, REDP-4935 and the IBM Redbooks publication *IBM Business Process Manager V8.5 Performance Tuning and Best Practices*, SG24-8216.

- Database configuration, tuning, and best practices

  For a more detailed description about best practices for database tuning with IBM BPM, see the IBM Redpaper publication *IBM Business Process Manager V8.0 Performance Tuning and Best Practices*, REDP-4935 and the IBM Redbooks publication *IBM Business Process Manager V8.5 Performance Tuning and Best Practices*, SG24-8216.

Considering these aspects of performance, we strongly suggest that you remember them from the beginning of the project, and plan accordingly as you build your solution.

Reporting

Many organizations overlook the importance of business reporting when they begin their process discovery phase in a new BPM project. It is either this, or reporting is the first thing that gets cut when a project is running out of time.

The lack of business-focused reporting can have a large effect on the success and acceptance of your project. Therefore, it is important to *make reporting a key aspect of your project*. Never implement a BPM project without planning for business reports.

1.4 Conclusion

In this chapter, we have introduced you to the concept of BPM and the benefits of a BPM suite. The success of a BPM project is not only affected by technical decisions that the team makes. Therefore, we have highlighted some of the more business-focused decisions that you as a team must make, because invariably they affect the outcome of your project. These decisions have an even bigger effect if you as an organization have started on a wider enterprise BPM journey.

In the following chapters, we go into more details about the more technical design decisions that you might face.

2

Approaches and process discovery

This chapter provides an overview of approaches for business process management (BPM) with IBM Business Process Manager (IBM BPM), and explores the main aspects of BPM success:

- Why BPM is important
- The BPM journey
- The six pillars of success for BPM
- Business process and Smarter Process Maturity Model: A roadmap
- Estimating IBM BPM projects
- IBM Smarter Process Method
- IBM component business modeling
- Process inventory
- Process Inventory, Variation Analysis, and Roadmap
- Blueprint or Macro Design
- Project
- Program
- Process Factory
- Migration
- Top Down - Bottom Up

2.1 Why BPM is important

Each organization might have a different entry point into its BPM initiative. Each business unit might be at a different point in the journey of adopting BPM, or in its maturity of using BPM after it is adopted.

Our experience indicates certain important considerations for success in the BPM journey. We share these considerations, which we call the *pillars of success* for BPM. Each of these pillars can become an initiative of its own as your organization grows and evolves in their use of BPM.

Many times, we are asked, “What do we do next?” Although the answer can vary depending on the current state of maturity, there are important concepts, initiatives, and considerations that need to be taken into account regardless of how much progress you have made to date. Sometimes, the next best action in your BPM journey implies that there is an order to those steps.

Completing the second step first, or ignoring a step, might make your project falter or fail, because project risks elevate. Because we do not want you to do either, it is important for you to know what the next step is, and when the correct time to implement it is.

Next, we address some common situations to see what the next logical step might be:

- You are trying to identify which part of your organization can benefit most from BPM.

  Next step: Define a heatmap using an IBM Component Business Model™, as explained in 2.7, “IBM component business modeling” on page 52.

- You are considering your first BPM project, but do not know what your processes really are.

  Next step: Conduct a process inventory, as explained in 2.8, “Process inventory” on page 56.

- You know that you have several processes in your business area, but do not know which one you should start with. You would like to extract commonality and reuse subprocesses. You are wondering if there is a way to consolidate and reuse processes. What about the variations in your processes? How do you handle them?

  Next step: Conduct a Process Inventory, Variation Analysis, and Roadmap (PIVAR), as explained in 2.9, “Process Inventory, Variation Analysis, and Roadmap” on page 57.

- You are working on a project and want to get an idea of what it would take to implement the processes for the project.

  Next step: Start blueprinting, as explained in 2.10, “Blueprint or Macro Design” on page 62.

- You are ready to implement your first IBM BPM application.

  Next step: Start your project, as explained in 2.11, “Project” on page 64.

- You have already implemented your first IBM BPM application successfully, and you are thinking of taking the next big step. You have many processes, often in different domains.

  Next step: Start a BPM program, as explained in 2.12, “Program” on page 65.

- You have many processes that you want to make use of commonality and implement them in parallel, or you want to deliver large complex processes more cost-efficiently.

  Next step: Set up an IBM BPM Factory, as explained in 2.13, “Process Factory” on page 67.

- You are using an existing platform for BPM, and want to make the most of the modern capabilities and features provided by IBM BPM.

  Next step: Plan for migration, as explained in 2.14, “Migration” on page 69.

2.2 The BPM journey

BPM, like many other trends that combine business and technology, is a journey. We start from early adoption and lower levels of maturity, and move toward higher levels of maturity. This journey, called *the business process maturity spectrum*, is described in the next section.

This overall BPM journey is depicted in Figure 2-1.

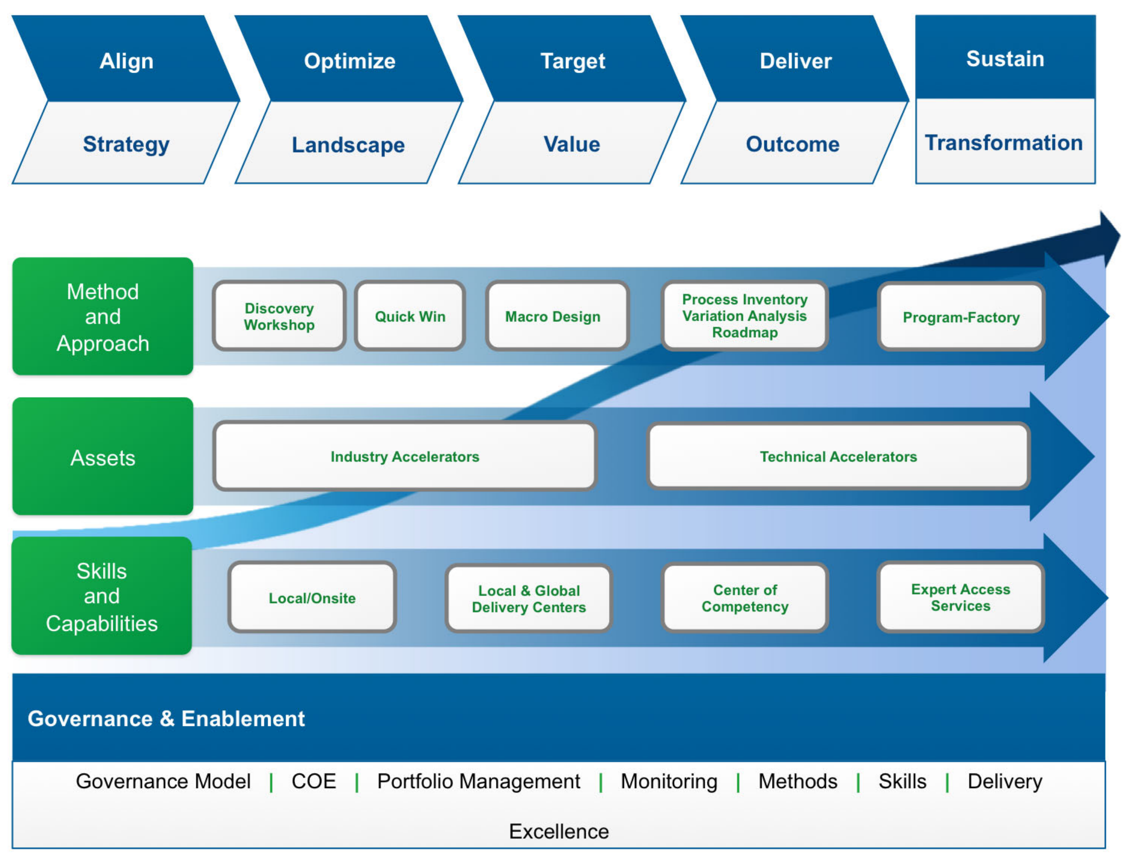


Figure 2-1 The BPM journey

There are three areas, all based on a foundation of governance and enablement:

- Method and approach as a progression
- Assets
- Skills and capabilities

The method and approaches show a progression of discovery workshops, quick wins, macro design, PIVAR, and then moving toward a factory-based approach.

A business case for automation of one or more business processes provides the motivating factor. We would like to be able to break out of tightly constrained, siloed applications, and start building applications from loosely coupled process steps. Loosely coupled steps enable you to rework the process without having to completely rewrite the application.

We want to somewhat mirror the way work is done in the business, but with more automation and efficiency. To accomplish this, we complete the following stages:

1. The first step in the journey typically starts with a *discovery workshop* of a few days that explores the details of single process.
2. Later, we move on to a *quick win* project, which delivers tangible business value in a short time frame.
3. As we gain more confidence, we map out the project using *macro design* or *blueprinting*, identifying the processes to automate. You also identify the integrations that need to take place, and the information architecture that underlies the integrations.
4. If a larger number of processes is involved, we engage in PIVAR so that we can capitalize on commonality and create a roadmap for automating multiple processes.
5. With continued success, we can elevate the effort to that of a program level of sophistication, and involve a Factory model for BPM development that scales to very large programs involving multiple projects.

2.3 The six pillars of success for BPM

Through our projects and across industries, we have accumulated leading practices that significantly increase project success rates and decrease risks. Among them are the *six pillars of success for Smarter Process*. Smarter Process consists of an intelligent (pun intended) combination of process, rules, integration using services, underlying data, and analytics combined to make a business process smarter, more capable of adaptation to business needs.

To increase success rates and decrease project or program risks, we suggest that you consider looking at starting the following tracks or initiatives. Often these streams of activities can be started somewhat independently of one another:

- Assessments and maturity models: Course corrections
- A strategic roadmap and tactical plan
- Agile methods and processes: Driving skills, predictability, and consistency within the organization
- Reference architecture: A blueprint or template to be instantiated by various business lines
- Governance: Institutionalizing consistency through centers of excellence (CoE)
- Tools and software platforms: Automating the process with state-of-the-art capabilities

2.3.1 Assessments and maturity models: Course corrections

A first important step is to agree on the features that characterize your current processes, and then decide which set of capabilities you need to have in a future or *to-be* state. This assessment needs to be predicated on some kind of a maturity model. We provide a high-level view of such a maturity model in the following paragraphs.

Course corrections are then made possible by assessing deviations from the roadmap that you have created. The roadmap consists of a set of initiatives, projects that are implemented to get you to your wanted future state in each of the dimensions defined by the maturity model that you use.

For example, the following dimensions describe the BPM maturity model:

- Knowledge worker processes (less predictable, ad hoc, and case-centric)
- Automated business processes (more predictable, straight through processing (STP), and deterministic)
- Rules and decisions
- Analytics and key performance indicators (KPIs)
- Services and integrations
- API management
- Data or Information Architecture (as it relates to the BPM initiative)
- Infrastructure

2.3.2 A strategic roadmap and tactical plan

You can create a *strategic roadmap* (often as a program consisting of multiple projects and initiatives), and a *tactical plan*. This plan is based on an evaluation of the as-is state, and a determination of the wanted to-be state that you have worked out in your maturity assessment.

A tactical project plan is created for the shorter term, and a longer-term plan is crafted with schedules, initiatives, and objectives to be attained. Resources are trained and allocated to meet these objectives and timelines.

2.3.3 Agile methods and processes: Driving skills, predictability, and consistency within the organization

Paramount to large-scale software development across an organization is consistency and prescriptiveness. This involves large groups of distributed staff working from the same templates so that activities converge and outputs are relevant across the organization. Prescriptiveness is important to help guide technical and business resources to engage in the correct activities to produce the wanted artifacts and deliverables.

IBM provides a Smarter Process Method for use on BPM and Smarter Process projects in general, depicted in the following section. It is advised that in the early stages of the project or program, a Method Adoption Workshop of 1 - 2 days is convened to decide which aspects of the method will be used for this project type. Determine which roles, artifacts, and activities are chosen to be delivered and executed for this project type.

We suggest defining a few project types as the baseline. Map other projects in terms of size, complexity, and domain (for example, private sector or public sector, each with their own specific needs) to these base types. The outcome is a work breakdown structure that can serve as a guide for future projects.

Remember that agile methods are appropriate for smaller projects. As project size and complexity increase, a tailoring of the agile method needs to be conducted. For example, it is tempting to assume that, after a quick win pilot or an initial working prototype, we can continue onward into the full-fledged project using the same approach.

Often we need to pause after the initial pilot or prototype and look holistically at the macro level of the project:

- The integrations
- The architecture as a whole
- Database considerations
- Infrastructure issues
- Configurations for the various development, staging, testing, and production environments

This Blueprinting or macro design phase is advised to alleviate problems and mitigate risks. You see examples and more details in the following sections.

2.3.4 Reference architecture: A blueprint or template to be instantiated by various business lines

Reference architectures, such as service-oriented architecture (SOA), can provide a blueprint for the logical and physical manifestations of the components and architectural building blocks that are agreed-upon standards within an organization. Often, different lines of business instantiate or realize the reference architecture using different products, or choose to alter one or more architectural building blocks to suit their specific needs.

The key point is that you ensure that it is not an ad hoc architecture that gets built every time a new project comes along. A basis is established as a standard in effect, which is adhered to by using governance and management.

Start with an industry-standard reference architecture, and then customize key aspects for your organization or line of business. This practice helps establish uniformity and consistency that can aid in the governance of enterprise initiatives. The SOA Reference Architecture from the Open Group (`http://opengroup.org`) provides the basis for a BPM layer of architectural building blocks that enable refinements needed to support business processes.

2.3.5 Governance: Institutionalizing consistency through centers of excellence

All of the standards and leading practices and methods that are agreed upon by one group of people at a given point in time fade from organizational memory unless they are institutionalized. Usually, a construct such as a CoE or a center of competency (CoC) is established as a steward and focal point for best practices and standards. It also facilitates the enforcement of the standards, or the appeal for exceptions should the need arise.

Governance defines the policies, and the management function enforces those policies in practice.

2.3.6 Tools and software platforms: Automating the process with state-of-the-art capabilities

IBM BPM can provide various configurations based on operational and development needs. Disaster recovery, redundancy, failover, monitoring of performance characteristics, security monitoring, and so on, are *non-functional requirements*. These requirements can be met through the combination of products, platforms, and policies, including monitoring and management of the design time and runtime environments.

2.4 Business process and Smarter Process Maturity Model: A roadmap

Most organizations are on a journey of adopting BPM. If you want to increase your capabilities and BPM area, it is important to understand the current state of the organization or the section of business that is attempting to use IBM BPM. You should also define the capabilities wanted to be achieved in a future state.

Figure 2-2 depicts the anatomy of a Smarter Process. The concept of a *Smarter Process* is one that combines business rules and decisions within the process, and integrates with existing and back-end applications and systems of record.

In addition, a Smarter Process is one that can connect and integrate with case management capabilities, and with structured and unstructured content that provides context to the knowledge worker who is handling a case. Often, data and analytics are gathered or monitored as part of metrics and KPIs within the context of a business process.

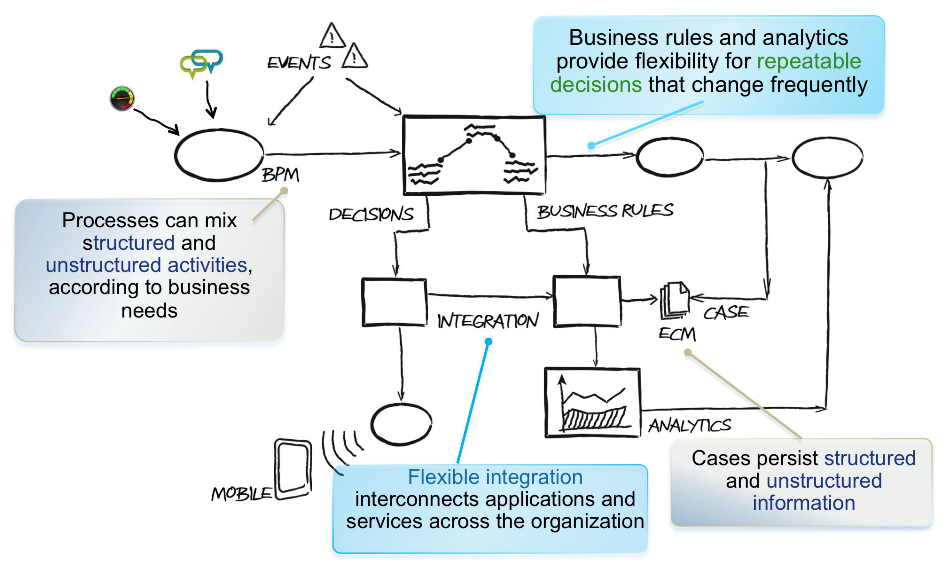
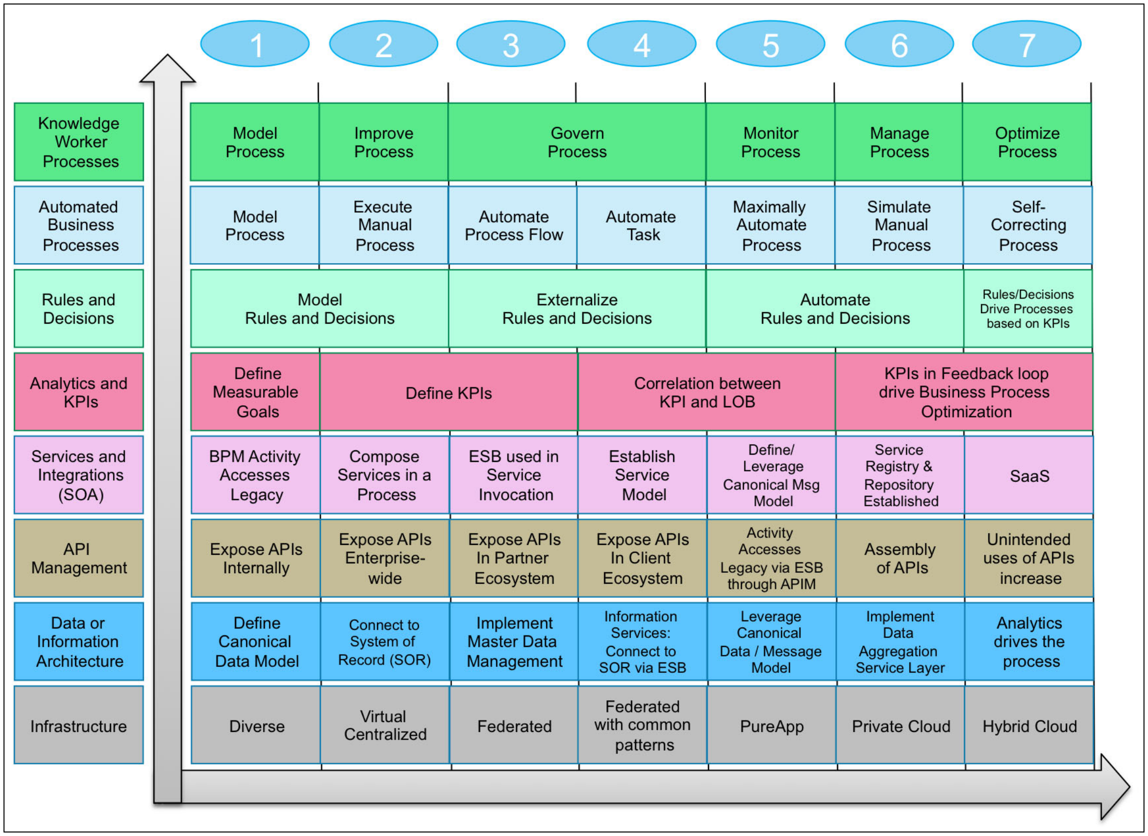

Business rules and analytics provide flexibility for repeatable decisions that change frequently
EVENTS
BPM
Processes can mix structured and unstructured activities, according to business needs
DECISIONS
BUSINESS RULES
INTEGRATION
CASE
ECM
ANALYTICS
MOBILE
Flexible integration interconnects applications and services across the organization
Cases persist structured and unstructured information

Figure 2-2 The anatomy of a Smarter Process

2.4.1 The spectrum of the Smarter Process Maturity Model

Although our primary focus is on IBM BPM, or the business process workstream within IBM BPM projects, there are several other dimensions of maturity that have a direct effect on the success of your BPM projects. Figure 2-3 depicts the overall Smarter Process Maturity Model.



	1	2	3	4	5	6	7
Knowledge Worker Processes	Model Process	Improve Process	Govern Process		Monitor Process	Manage Process	Optimize Process
Automated Business Processes	Model Process	Execute Manual Process	Automate Process Flow	Automate Task	Maximally Automate Process	Simulate Manual Process	Self-Correcting Process
Rules and Decisions	Model Rules and Decisions		Externalize Rules and Decisions		Automate Rules and Decisions		Rules/Decisions Drive Processes based on KPIs
Analytics and KPIs	Define Measurable Goals	Define KPIs		Correlation between KPI and LOB		KPIs in Feedback loop drive Business Process Optimization	
Services and Integrations (SOA)	BPM Activity Accesses Legacy	Compose Services in a Process	ESB used in Service Invocation	Establish Service Model	Define/ Leverage Canonical Msg Model	Service Registry & Repository Established	SaaS
API Management	Expose APIs Internally	Expose APIs Enterprise-wide	Expose APIs In Partner Ecosystem	Expose APIs In Client Ecosystem	Activity Accesses Legacy via ESB through APIM	Assembly of APIs	Unintended uses of APIs increase
Data or Information Architecture	Define Canonical Data Model	Connect to System of Record (SOR)	Implement Master Data Management	Information Services: Connect to SOR via ESB	Leverage Canonical Data / Message Model	Implement Data Aggregation Service Layer	Analytics drives the process
Infrastructure	Diverse	Virtual Centralized	Federated	Federated with common patterns	PureApp	Private Cloud	Hybrid Cloud

Figure 2-3 The Smarter Process Maturity Model

The Smarter Process Maturity Model includes the following dimensions:

- Knowledge worker processes (less predictable ad hoc, and case-centric)
- Automated business processes (more predictable, STP, and deterministic)
- Rules and decisions
- Analytics and KPIs
- Services and integrations
- API management
- Data or information architecture (as it relates to the BPM initiative)
- Infrastructure

Next, we describe the key capabilities that need to be maturing for typical IBM BPM projects. This approach provides a planning framework to assess where you are, and where you want to take the next step and enhance existing organizational capabilities in IBM BPM projects.

The maturity model can also be seen as one of adoption, in which certain capabilities start to appear, or should be expected to appear at a certain point in the growth and evolution of BPM within your organization.

In Figure 2-3 on page 34 we show seven dimensions in the rows, starting from manual processes, to automated business processes, to the rules and decisions needed to run those processes. It also shows the analytics and KPIs that are used to define, measure, and ultimately create a feedback loop for improvement and optimization of the business process.

The next dimension is the all-important SOA, the set of services that will be started when an activity is activated during a business process. Services grow into the Representational State Transfer (RESTful) application programming interface (APIs) that can be made available inside or outside the enterprise. These APIs ultimately rely on data and information in the systems of record.

The maturity of BPM describes more of a journey within BPM. Although there are many entry points and possible paths to take within BPM, the maturity model provides a reference point around which you can plan your journey for BPM. This model defines the underlying dependencies and possibilities that need to be considered to arrive at a higher level of maturity or capability in the adoption and capitalization process of using BPM.

2.4.2 The dimensions of maturity

The concept of aspects or *dimensions* of maturity covers the spectrum from the day-to-day operations of the business down to the details of the information technology (IT) infrastructure required to support it. Dimensions of maturity typically involve an analysis of the following aspects of your business processes:

- STP versus manual or knowledge worker processes
- Automated business processes
- Rules and decisions
- Analytics and KPIs
- Services (of an SOA) used to reflect the portfolio of services required to support a business

This often manifests in the area of service integrations that denote the connectivity to back-end systems, which often supply these services or business capabilities that help run a business.

As we delve into more cloud-based and software as a service (SaaS) capabilities, we enter into the API for RESTful web services, or API management functions and capabilities. Most importantly, we have data or information architecture and analytics that forms the backbone and cornerstone of an organization, and supports its processes, services, rules, and components.

Note that the fields in each of the dimensions running horizontally are not an exhaustive list of the aspects of maturity. Rather, they form categories in which you can perform the following BPM activities:

- Consider more details.
- Conduct assessments to determine the as-is state.
- Designate and agree upon a wanted target or to-be state.
- Based on these source and destination points, plot a set of initiatives that collectively fulfills the requirements and implements the capabilities required by arriving at a to-be state.

The following sections outline a few representative questions that should be asked, and an assessment made of the capabilities that are wanted to be achieved within that domain. Note that a formal assessment has a more detailed set of questions.

2.4.3 Knowledge worker processes

Processes cover a spectrum ranging from an uncertain set of steps that depend on the context of the case that you are dealing with, to a series of well-known steps in an automated business process. To model the problem space accurately so that you can create an appropriate solution, it is important to understand and outline the process in terms of its activities or steps:

- How much of the process is prescriptive? How well-known are the steps?
- Are we fairly certain about the structure and sequence of the steps in the process? Can they be mapped out before the process actually starts?
- How much of the process can be automated deterministically, as in an automated STP? Alternatively, is the process more of a case management scenario requiring more knowledge worker participation, investigation, and decision making that is up to the discretion of a person?
- What are the steps in the process?
- Who is deciding each step?
- Can the decisions be automated? Can the steps be automated? To what extent can they be automated?
- How much is a knowledge worker involved? How much is this STP?

2.4.4 Automated business processes

Map out the process, its activities and steps, the swimlanes, and branching logic as we go from one activity to the next. Examine the degree of certainty that you have in each branch or scenario of the process, and who needs to be involved to make it flow (who decides). Examine the efficiency and effectiveness of the process in trying to accomplish its goals. Examine the roles within the organization that participate in the process.

Business processes consist of activities that often use back-end systems for information, to process the current step and decide what the next step in the process is. Examine the degree of maturity of those integrations. Are they currently “hard wired”, or are they web services calls?

Examine the flow within the business process, the routing that is required between activities. Sometimes we might require more than routing logic. We might need a business analyst (BA) or business subject matter expert (SME) to model the rules separately and the process can use those rules.

We need to evaluate how much data we currently have about a process, to continue optimizing it:

- Certainty of steps

  What is the degree of certainty we have in the sequence and flow of the activities and the steps within the business process?

- Automation

  Of the activities that can be automated, what level of automation is feasible? That is, can we make the decisions at each step, or do we require further information? Do we need an integration with a system of record to pull data? Do we need to request a human to check something?

- Maturity

  Are we currently in the beginning stages of maturity of automation of processes? Do we want to start the automation by initially imitating the manual process, measuring it, and gradually getting rid of the unnecessary knowledge worker interactions that can begin to become more automated?

- Swivel-chair

  Are we in a position to model a *swivel-chair* automation, to help the knowledge worker automate auditing, communication, and work queue-related tasks?

- Process flow logic

  What level of business rules governs each step in the process? Can we introduce routing logic in the process, or do we require a more elaborate set of business rules?

- Process simulation

  We can simulate the flow of the process to find bottlenecks and run what-if scenarios in an automated business process. Do we have the metrics about how long each activity is taking? Alternatively, do we want to run the process a few times to obtain those metrics?

2.4.5 Rules and decisions

Decisions points are often embedded within the process flow. At each of these decision points, business rules need to be applied so that the process can decide what action to take. Start data mining and discovering the rules and decision points by interviewing the knowledge workers and gathering existing spreadsheet information.

This helps you determine how much of the decision can be put in the business process itself, and how much of it is probably better served by externalizing it into a rules engine, such as IBM Operational Decision Manager (ODM).

- Mine the rules

  Where are the rules and decisions today? Are they in spreadsheets? Inside the organizational “tribal” knowledge?

- Externalize business rules

  Are they externalized or embedded within the process?

- Model the rules

  Do they have a series of filters that needs to be applied for multiple rules, or are they less complex rules?

- Differentiate rule types

  Are the rules that you encounter mostly routing rules? Alternatively, are they true business rules that decide how the business operates? Do they contribute to the decision of what action to take next, and what back-end system to pull data from or submit data to, for processing?

2.4.6 Analytics and KPIs

You can record known performance metrics within the process and run simulations. At other times, if information about a process flow is outdated or questionable, you might want to start by simulating the existing process in a *swivel chair* model:

- Current metrics and KPIs

  What, if any, measurements do we currently have in place that can be used as a benchmark? Do we need to make an educated guess based on current trends and feedback, and run the process first to gather this information?

- Measuring the process

  What aspects of the process do we want to measure? How can we help the kanban, kaizen, Six Sigma, lean, or lean Six Sigma efforts through automated measurements and simulation of the process?

- Efficiency

  What aspects of the process do we want to constrain and limit in terms of acceptable execution time of each step, or of the entire process?

2.4.7 Services in an SOA

A business process consists of a set of steps or activities. Each of these activities might request a person (human task through a Coach or user interface) to do something, such as approve a step in the process.

Each activity might also access a back-end system of record. Connection to a back-end system requires integration capabilities that are best suited for an SOA. The SOA consists of a portfolio of business-aligned services that implement the business capabilities required within the context of a business process.

- Identify the portfolio of services that support key business capabilities.

  What are the list of business capabilities required to produce business outcomes? What are the services that need to be in place to support the capabilities required by each line of business?

- Specify the integrations required with the back-end systems.

  What back-end systems, applications, and systems of record are required to integrate with the automated process?

- Differentiate between data integrations and application integration.

  What are the data integrations required with systems of record or analytics data warehouses? What are the integration points with applications such as existing applications, commercial off the shelf (COTS) packages, and more customized packages (such as SAP, PeopleSoft, and so on)?

- Determine the phases that a service goes through.

  What is the lifecycle of the services?

2.4.8 API management

Consumers expect to access data at any time, across multiple devices. Provider companies can reinvent interactions with clients, suppliers, and partners. Explosive growth of potential clients increases opportunity, risk, and innovation. As APIs emerge within multiple industries, the ability to make APIs available to the value chain, and to capitalize on them, becomes a strategic advantage.

This dimension assesses the degree of preparedness or, if in place, the degree of evolution and growth of the API management effort within the organization to support key industry trends, such as mobile, social, and analytics:

- Mobile strategy

  Is there a need for RESTful web services? How are you planning to deliver mobile solutions, if any?

- Factor competitors in technology adoption

  Is there a need for competitive advantage through making APIs available, or using existing APIs, to compose applications that support a cloud-based or SaaS-based model?

- Management of new delivery models

  How are we planning to manage the SaaS aspect of the portfolio?

2.4.9 Data, content, and information architecture

A significant portion of a process is predicated on the data on which it relies, both structured and unstructured. The necessary data is often gathered through analytics, and stored in systems of record that must be accessed using services:

- Data and access

  Is the data structured or unstructured? What parts of the process need access to this content?

- Status of standardized data models for information exchange

  Is a canonical data model in place? Is a canonical message model in place?

- Level of master data management (MDM) adoption

  Is an MDM effort in place to consolidate systems of record?

- Information lifecycle governance

  What is the lifecycle of data and information in connection with the domain under study, and as it relates to the BPM processes? What data is to be used within the context of the process?

- Data for decisions

  Is there data that needs to be used to support rules and decisions? Are they externalized?

2.4.10 Infrastructure and non-functional requirements

The previous considerations are a *snapshot of some of the key questions* that must be asked and answered within the context of a BPM project to assign actions to support the project over its lifetime. Extrapolate these snippets to a larger-scale assessment effort, and you have a picture of what the assessment involves and what the output looks like.

The assessment includes as-is capabilities and wanted to-be capabilities that we need to put in place for the successful execution of BPM projects:

- Environments

  What are the specifications for the environments? Install and configure the development, testing, staging, and production environments.

- Monitoring and management

  What are the required monitoring and management capabilities?

- Support for non-functional requirements

  What kind of redundancies and disaster recovery capabilities are required?

2.5 Estimating IBM BPM projects

As a project manager or a technical staff member, the estimation that you participate in defining significantly contributes to the success of your project. Estimating a project provides the technical team and the business stakeholders with visibility into the viability of the project's success. Estimation is a measure of expectation for planning purposes.

Estimation also provides insight into skills, resources, and timelines for budgets and commitments. One of the initial hurdles to surmount is that of assessing complexity, which helps estimate the level of effort associated with the project. There are various levels of estimation, including rough order of magnitude, detailed estimations, and so on.

2.5.1 Dimensions of complexity

Project complexity has a direct influence on project duration. Next, We look at how the complexity of the BPM artifacts that we intend to build influences the project duration and level of effort. There are various dimensions to complexity, such as the *degree* of complexity (for example, low, medium, and high). Another aspect of complexity runs across a spectrum, which we describe in more detail in the following section.

Next, We start with the concept of the BPM artifact. The following list includes the six main IBM BPM artifacts:

- Business process definitions (BPDs).
- Activities that occur with the business process.
- User interfaces or *Coaches*, as they are called in IBM BPM.
- Services. Activities call services that connect to or integrate with back-end systems of record or applications.
- Business rules.
- Reports.

So, how many of these will you have on your project? And how long will it take to build each one?

First, you need to estimate how many of each of these you need to build. Then, estimate what the level of complexity of each is going to be, within the project context. Finally, estimate how long each takes to build. Then you can integrate these three steps into your estimation for additional project activities beyond development, construction, and testing. Next, we look at this process.

2.5.2 Degrees of complexity

In this section, we describe the aspects of project complexity and provide insight into how these aspects affect your BPM project estimation and scheduling.

Figure 2-4 depicts several key areas of consideration for your projects. They consist of a spectrum running from a value on the left to a value or attribution on the right, which encompasses the range of options or scale associated with that spectrum.

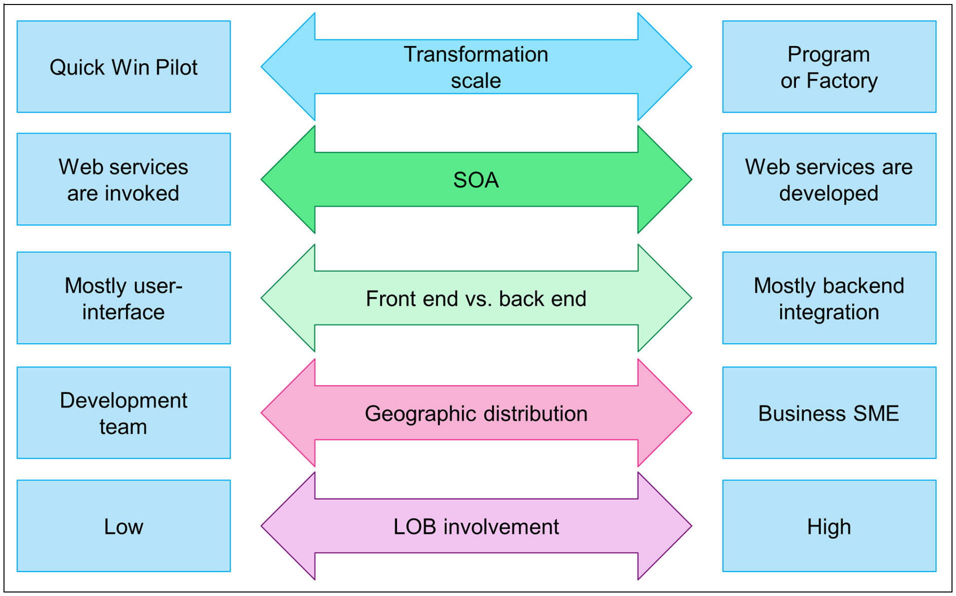


Figure 2-4 Factors and spectrums of scale and complexity

The scale of five: Low to ultrahigh complexity

Before we get into the spectrums of scale and complexity, we calibrate our measurements. It would seem intuitive to map the level of project or artifact complexity to the traditional low, medium, and high levels of complexity.

Statistical analysis of BPM projects shows that *there are, in fact, five tiers of complexity rather than the simplistic three*, as shown in Figure 2-5.

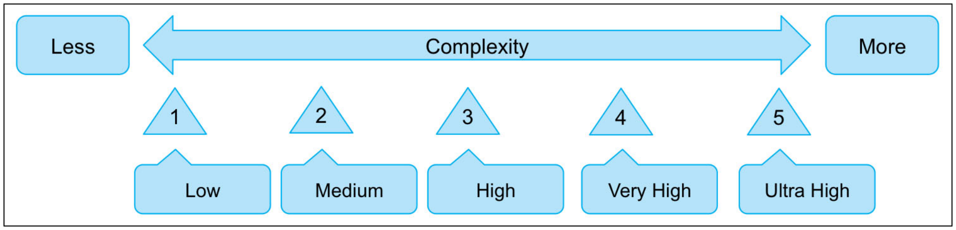


Figure 2-5 The scale of complexity

For a rudimentary order of magnitude, an estimation of low, medium, and high levels of complexity might suffice. But for a more accurate estimation, a scale of 1 - 5 is suggested. The corresponding English language designation might be 1 = low, 2 = medium, 3 = high, 4 = very high, and 5 = ultra high complexity.

You can name them whatever you want. However, note that there is a range that exceeds the intuitive and commonly held belief, a range that goes far beyond traditional levels of complexity.

As we cover the spectrums of complexity, consider where your project falls within each of them. This helps create a more realistic estimate and project plan.

Speed of delivery level or transformation scale

The first spectrum shown on Figure 2-4 on page 43, pertains to the speed at which the business is expecting results to be delivered. A typical way of initiating an IBM BPM project is to start with a “quick win.” The quick win provides the business with a working prototype that provides tangible business outcomes, often focused on showing how fast something of business value can be implemented.

The intent of such a quick win is more a *proof of concept* (POC) than something that goes straight into production. The other end of the speed of delivery spectrum is a program with multiple projects, and possibly the means of industrializing the development process.

Integration using services (SOA)

Many projects start with the assumption that the underlying integrations are available, and that the services only have to be connected and called. Therefore, no further design or construction is required. Often, upon closer scrutiny, this assumption is found to be false: The services are not ready.

Therefore, during estimation, a reasonable amount of time should be allocated to the design and implementation of the web services or other integration mechanisms that are relevant and appropriate for IBM BPM to access existing back-end systems or systems of record.

Front end versus back end

Some projects are focused primarily on the visual user interface (UI) and interaction with users, such as a workflow that supports allocation of human tasks in a work queue. These UIs (front-end) are called *Coaches* in IBM BPM.

At the other end of the spectrum, we find systems whose requirements are heavily focused on the back office (back end). These systems have relatively few human interfaces, and a substantial number of integrations and interactions with existing systems, COTS applications, or systems of record. A consideration of the nature of where your project falls within this spectrum helps create a more realistic estimation and project plan.

Geographic distribution of the development team or SMEs

A key factor in assessing complexity and planning for your BPM project is the selection of the number of iterations, and the duration of each iteration, in your agile software development lifecycle. Geographical location or geographic distribution of collaborating technical teams and business SMEs also plays an essential role in managing the interactions and collaborations that are necessary for making agile software development successful.

An agile software development process is predicated on a close collaboration between the development team and business SMEs. Geographic dispersion creates resource usage that you need to take into account for your project estimation and project plan execution. For example, consider scrum meetings. What does a virtual scrum mean for your organization and your team?

Line of business involvement

Insufficient business involvement in the project is a major risk factor. This forms the basis of an important consideration for your project. Estimation, planning, and the resources necessary to discover, automate, improve, or optimize business processes requires active BA participation.

In fact, we suggest the use of a metaphor of “two in a boat”: The architect and BA rowing together to take the project forward. What this implies is the connected collaboration and joint ownership of artifacts and decisions, with each role contributing to and leading pieces of the project that are related to their domains of expertise. This also helps in instituting governance, or the BPM CoE.

2.5.3 BPM artifact complexity: Some indicative samples

Another important consideration in estimation and planning of BPM projects is the level of effort required to develop the solution. Having historical data becomes increasingly important. Start by making it a discipline to gather data pertaining to how much time each role is spending on tasks.

This historical data is gathered and analyzed across multiple projects, and can facilitate, and provide a reference point for, estimation and planning efforts within the context of your BPM projects. We encourage you to do so, often in the context of the BPM CoE, which consolidates activities around leading practices for BPM within your organization.

In gathering and analyzing this data, note that the velocity of each new team will differ based on skill levels, domain experience, amount of past team experience, and fundamental complexity of the problem domain, as outlined in 2.5.2, “Degrees of complexity” on page 43.

Table 2-1 provides an indicative sample of statistical distribution based on the five tiers of complexity.

Table 2-1 Sample statistical distribution: Indicative only

BPM artifacts	Sample Complexity Spectrum - Indicative Only				
	Low	Medium	High	Very High	Ultrahigh
BPD	58.80%	29.40%	11.70%	0.00%	0.00%
Activity	38.96%	33.77%	16.88%	7.79%	2.60%
Coach	62.96%	11.11%	14.81%	7.41%	3.70%
Integration	17.50%	20.00%	20.00%	30.00%	12.50%

The first column in Table 2-1 represents the BPM artifacts. BPDs are business process definitions, which contain activities. Coaches refer to the UIs. Integrations refer to web services-based integrations, where database integrations pertain to integrations with systems of record.

The complexity spectrum indicates the statistical distribution of each of the BPM artifacts. For example, about 63% of Coaches are low complexity, about 11% are medium complexity, and 15% are high complexity. Interestingly, about 7% of Coaches are very high complexity and about 4% are ultrahigh complexity.

Does that mean that we advocate creating an ultrahigh complexity Coach or integration or activity? Clearly not. What this means is that despite advice and leading practices, there are a statistically significant number of cases where existing projects and business requires you to create or update a UI, integration, or activity that does fall within the very high or ultrahigh scale. If you want realistic estimates, you might want to consider this fact for your estimation.

2.6 IBM Smarter Process Method

In this section, we explore the IBM Smarter Process Method, a BPM method that provides prescriptive guidance of leading practices for consistently successful BPM projects.

Figure 2-6 shows the IBM Smarter Process Method.

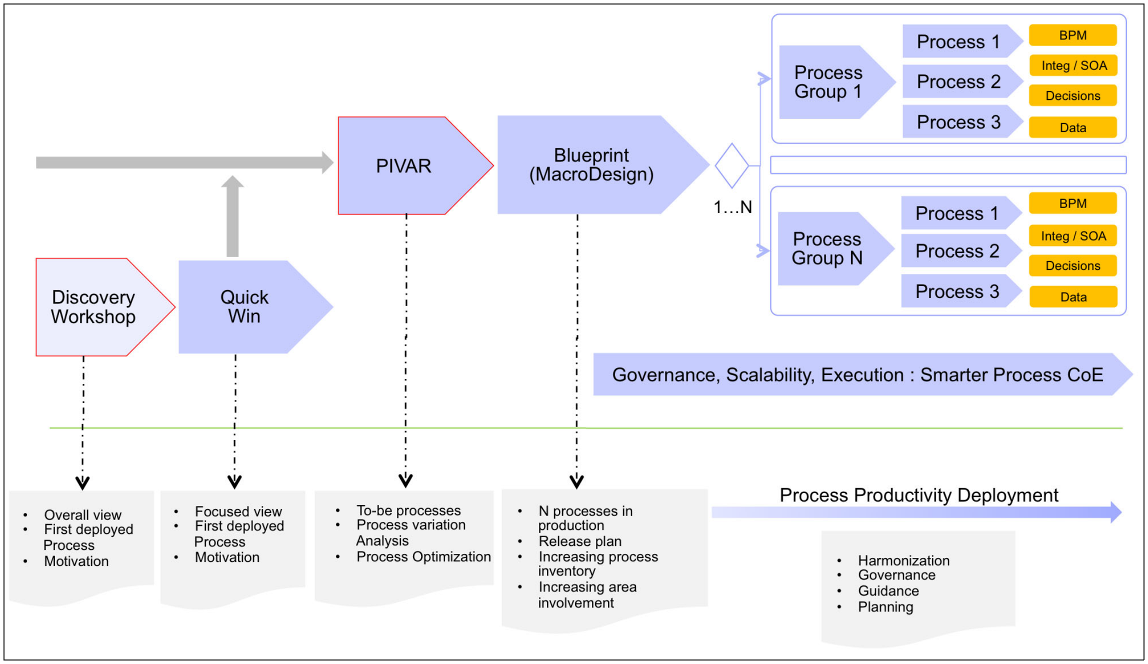


Figure 2-6 IBM Smarter Process Method

The IBM Smarter Process Method provides prescriptive guidance on leading practices for each phase within a BPM project or program. We have several phases, some optional. We often begin with a discovery workshop that delves deeper into one representative process and establishes some initial estimates.

Then we engage in one or more quick win pilots that demonstrate the execution speed of a project, and its technological possibilities, while providing some basic value to the business.

If we implement larger projects, we embark on a blueprinting activity that looks at the global and holistic technical details of the overall project or program. If we have several processes and think there is value in combining efforts and factoring out commonality, we engage in a PIVAR activity. The result of the PIVAR is not only to analyze commonality and variability, and to factor out the commonalities, but also to condense the total number of processes.

Processes are grouped and started in parallel based on their business affinity or technical cohesion. They are implemented together in multiple groups.

2.6.1 Workstreams

There are two main workstreams for us to consider on BPM projects:

- Business Process workstream

  The major workstream in a BPM project is the *Business Process workstream*, which includes the Coaches, the BPDs, and the activities that comprise the process.

- Services and Integration workstream

  An IBM BPM application also needs to access systems of record, and possibly different sorts of established back-end systems or COTS applications. To provide this access, we need to have integrations. The *Services and Integration workstream* uses a set of services within the context of an underlying SOA to connect to these resources.

Figure 2-7 shows the two main workstreams for a typical (low-to-medium complexity) BPM project. If we have a high complexity project with higher levels of complexity of the artifacts, we can add more workstreams to deal with that complexity. But on 80% of projects, the two workstreams, business process and services and integration, are more than adequate.

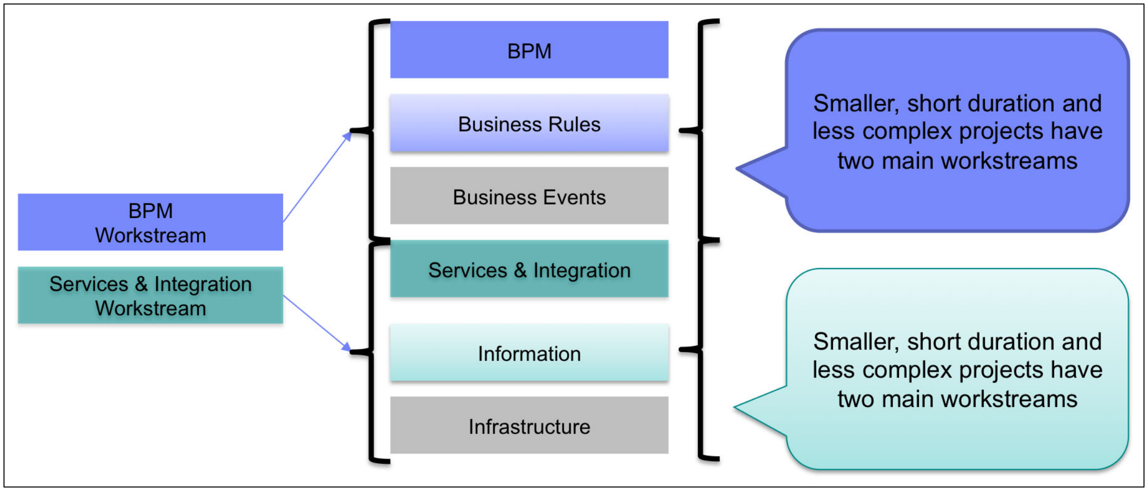

BPM
Business Rules
Business Events
Services & Integration
Information
Infrastructure
BPM Workstream
Services & Integration Workstream
Smaller, short duration and less complex projects have two main workstreams
Smaller, short duration and less complex projects have two main workstreams

Figure 2-7 IBM BPM: Two major workstreams

These two workstreams are the ones to start with on smaller, lower complexity projects, such as those that can be designated as *quick wins*. As the duration, complexity, and number of processes to be implemented increases, we can decompose each of the major two workstreams into a set of subworkstreams.

This enables us to address the issues encountered on larger projects, such as the need to develop new databases, gather analytics, and deploy significant infrastructure. These activities apply to both functional and non-functional (such as disaster recovery) requirements.

Therefore, the BPM workstream can be divided into several workstreams covering business processes, rules, and business events. When required, the integration workstream can correspondingly be subdivided into services and integration, information or data, and infrastructure workstreams.

Coordination among these workstreams is essential. This coordination is especially critical at the end of an iteration, where we typically have a playback for the business.

2.6.2 The IBM Smarter Process Method: Details

The internal structure of the method is depicted in Figure 2-8.

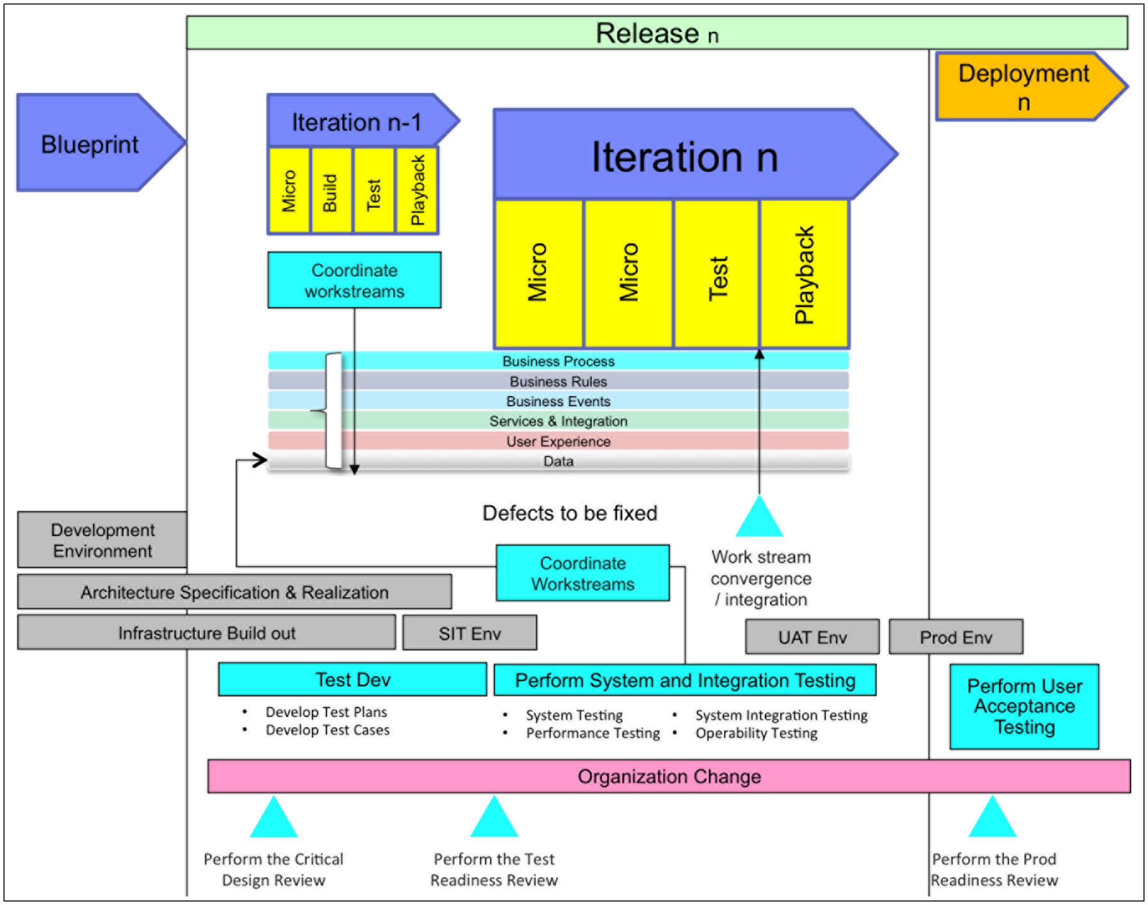


Figure 2-8 Internal structure of the IBM Smarter Process Method

A *blueprint* phase or activity is designed to help provide a holistic picture, set the landscape and pace for the project, and rework the estimates. The results of this phase increase your probability of being on time and on budget.

A *PIVAR*, which is covered in more detail in 2.9, “Process Inventory, Variation Analysis, and Roadmap” on page 57, deals with variation, factoring out commonality and reuse, and decreasing the number of processes to build.

We typically work in at least two *workstreams*, BPM and services and integration. We can possibly expand them into substreams as the complexity and richness of the project increases. This might be true, for example, when requirements exist for more explicit business rules, databases, or infrastructure considerations.

Within these workstreams, we have *iterations*. In an *agile* software development lifecycle, we have the following activities and entities during an iteration, shown in Figure 2-9. An iteration can include one or more *sprints*, and iterations can be grouped under a *release*.

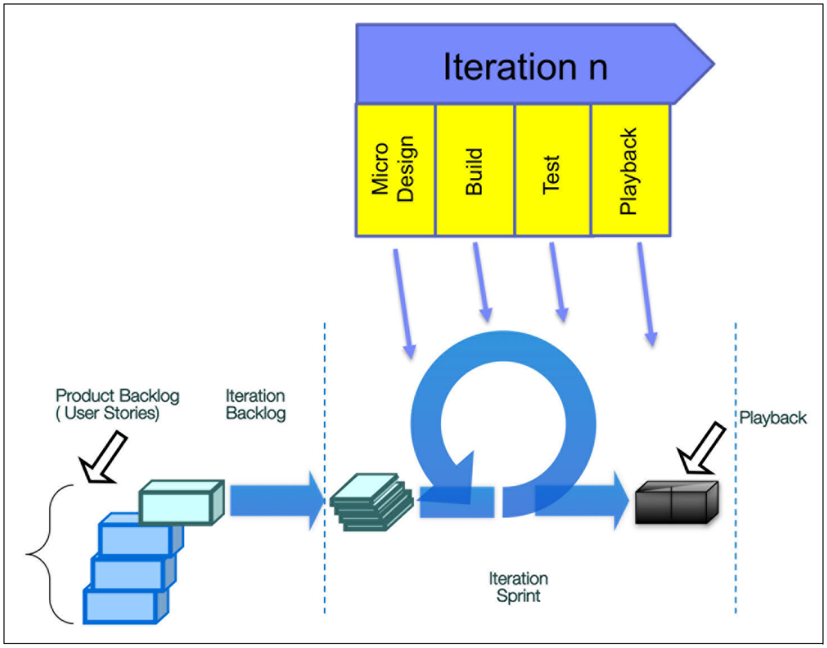


Figure 2-9 Iterations

The agile lifecycle starts with the identification of user-stories, for example, *as a <role>, I want to be able to <action>, in order to <goal>*. The total sum for the entire project is called the *product backlog*. A prioritized selection is allocated for each iteration or sprint.

During that iteration, we conduct a *micro design* activity, where the design is refined for that release. We *build* and *test* the user-stories in scope. Then, if a planned business significant milestone is achieved, as in most iterations, we conduct a *playback*. During the playback, business and IT stakeholders get to review the running program up to this point, provide comments and feedback, and decide on the priorities for the next iteration.

2.6.3 Techniques and phases of the IBM Smarter Process Method

In the subsequent sections, we describe a phase or pattern within the IBM Smarter Process Method that can be used to solve a typical problem or address a specific area within an IBM BPM project or program:

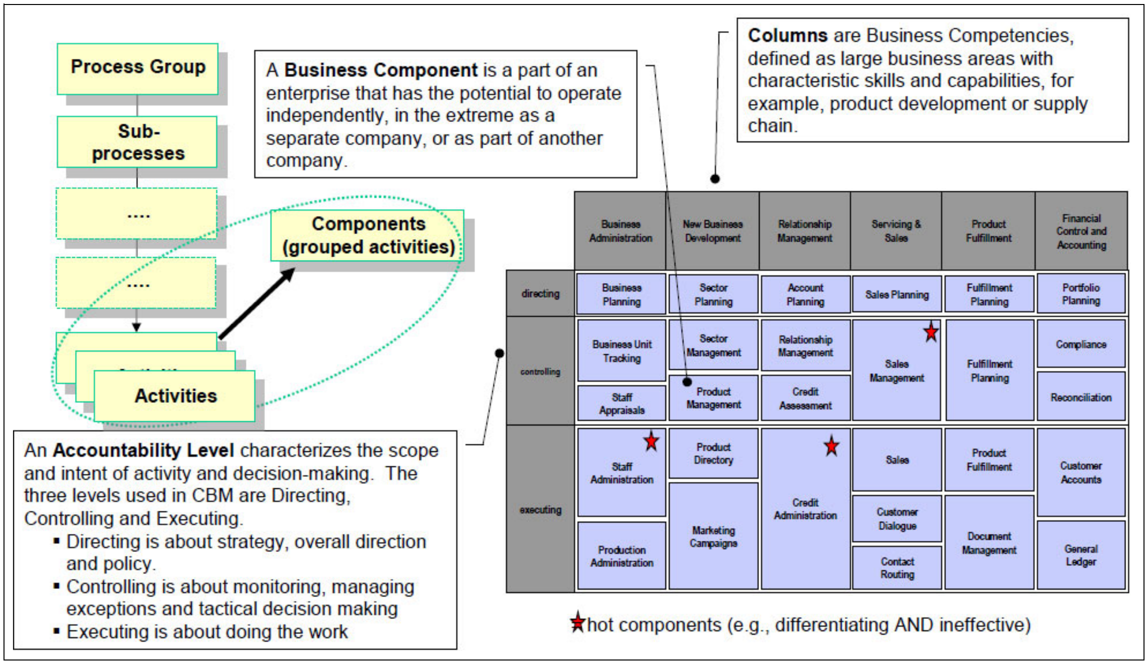
- Quick wins
- Blueprinting and macro design
- PIVAR
- Design
- Construction using IBM BPM

We approach this by describing a technique called *component business modeling* that helps identify areas of greatest potential for transformation.

2.7 IBM component business modeling

Component Business Modeling (CBM) is a partitioning of the business capabilities of an organization into a set of business components. Each business component view focuses on regrouping business activities into components. This is a technique for modeling an enterprise to identify opportunities for innovation and improvement by highlighting them in a heatmap fashion.

As shown in Figure 2-10, a CBM is a matrix visualization of an organization's business components.



A **Business Component** is a part of an enterprise that has the potential to operate independently, in the extreme as a separate company, or as part of another company.

Columns are Business Competencies, defined as large business areas with characteristic skills and capabilities, for example, product development or supply chain.

An **Accountability Level** characterizes the scope and intent of activity and decision-making. The three levels used in CBM are Directing, Controlling and Executing.

- Directing is about strategy, overall direction and policy.
- Controlling is about monitoring, managing exceptions and tactical decision making
- Executing is about doing the work

	Business Administration	New Business Development	Relationship Management	Servicing & Sales	Product Fulfillment	Financial Control and Accounting
directing	Business Planning	Sector Planning	Account Planning	Sales Planning	Fulfillment Planning	Portfolio Planning
controlling	Business Unit Tracking Staff Appraisals	Sector Management Product Management	Relationship Management Credit Assessment	★ Sales Management	Fulfillment Planning	Compliance Reconciliation
executing	★ Staff Administration Production Administration	Product Directory Marketing Campaigns	★ Credit Administration	Sales Customer Dialogue Contact Routing	Product Fulfillment Document Management	Customer Accounts General Ledger

★hot components (e.g., differentiating AND ineffective)

Figure 2-10 CBM overview

The columns represent the *business competencies*, defined as large business areas with characteristic skills and capabilities, for example, product development or supply chain.

The rows represent the *level of accountability*. They characterize the scope and intent of activity and decision making. The three levels used in CBM are *directing*, *controlling*, and *executing*:

- Directing is about strategy, overall direction, and policy.
- Controlling is about monitoring, managing exceptions, and tactical decision making.
- Executing is about performing the work.

A *business component* is a part of an organization that has the potential to operate independently, in the extreme as a separate company, or as part of another company. A component represents a business in a microcosm. It contains activities, resources, applications, and an infrastructure. It uses a governance model and provides goods and services (business services) to other components.

Figure 2-11 depicts the details of a CBM component.

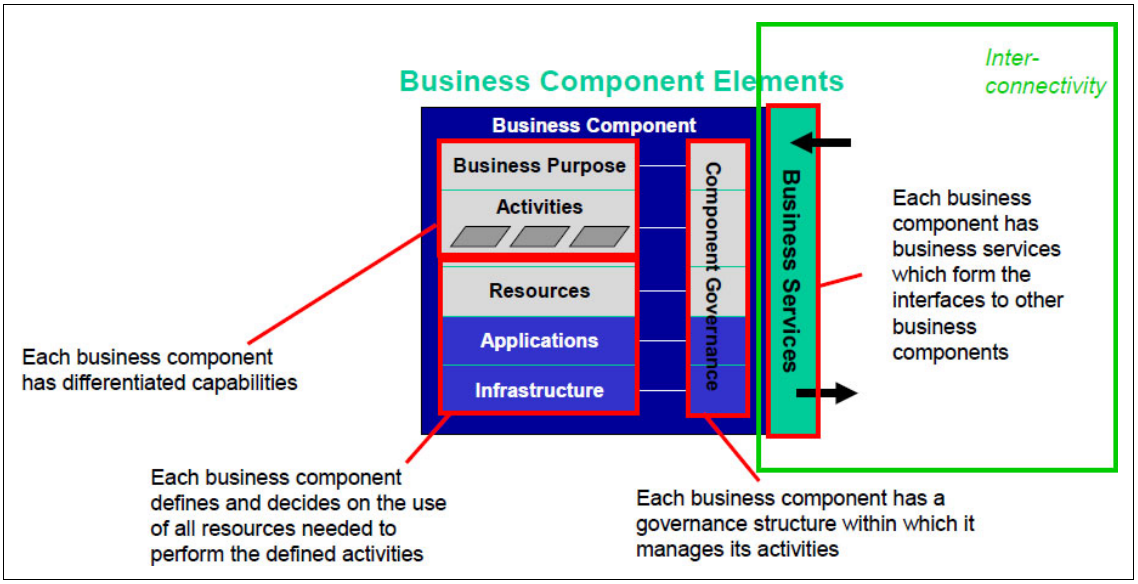

Business Component Elements
Inter-connectivity
Business Component
Business Purpose
Activities
Resources
Applications
Infrastructure
Component Governance
Business Services
Each business component has business services which form the interfaces to other business components
Each business component has differentiated capabilities
Each business component defines and decides on the use of all resources needed to perform the defined activities
Each business component has a governance structure within which it manages its activities

Figure 2-11 CBM component

Every business component has the following attributes:

- It has differentiated *capabilities*.
- It defines and decides on the use of all *resources* needed to perform the defined activities.
- It has a *governance* structure, which is used to manage its activities.
- It has *business services*, which form the interface to other business components.

A business component is the essential, unique, and non-overlapping building block of services, the collection of which represent the enterprise. Its resources create distinct value and are evaluated by specific measures. Each component acts independently, but in harmony with the rest.

CBM in one sentence: CBM enables visualization of the organization, strategy, and performance from a *common perspective*, facilitating business transformation by improving decision-making and decreasing resistance to change.

You can use CBM to create a heatmap to highlight which component is particularly good or bad at something, for example, the following business areas:

- Strategic criticality
- Industry leadership
- Degree of investments

Using CBM, you can answer the following questions (and other similar questions):

- *How well are our investments aligned with our strategy?*

  You might learn that, for example, that a business component gets a high degree of investment (funding), however, it is only of low strategic criticality. Of course, many other kinds of analysis can be made with CBM, too.

  For our business process improvement initiative, we can use CBM to highlight which business component has the highest inefficiencies, most potential for improvement, most value to the corporate strategy, and so on. We can use this information to select this area of the enterprise to start looking more closely into business processes.

  Every component has several business processes. Some processes can even go across several business components. Even though this is the case, the ownership of the process might lie within one particular business component.

- *How do you look closer into the business processes of a particular business component?*

  Well, you start with a process inventory, or a PIVAR activity, to identify the processes that exist, and evaluate which ones to improve or automate to provide the highest value for the business component.

For more information about CBM, see the following resources:

- Executive brief of component business models by the IBM Institute for Business Value:

  `https://www.ibm.com/services/us/imc/pdf/g510-6163-component-business-models.pdf`

- Information about IBM component business modeling services, which can help you gain significant new insights into the strategy, technology, operations, and investment alignment of your organization:

  `http://www.ibm.com/services/us/en/business-services/soa-business-architecture-and-component-business-modeling.html`

Another method: Another method that is similar to CBM is called *business capability analysis*.

Although it is a good idea to use a formal method like CBM or business capability analysis, it is not a strict requirement to start a BPM project.

2.8 Process inventory

Maintaining an inventory of business processes improves the likelihood of meeting business objectives by providing traceability between the process and the business goals. With an inventory, you can plan and trace your process discovery activities to core business objectives. This situation improves your visibility, and helps determine what processes to work on and when. You want to work on the correct processes at the correct time.

A process inventory exercise quickly generates a list of business processes (or process areas) that can be categorized by business unit, size, business effect, risk, and pain. In a brief interview, you gather enough details for each process to map it to the value chain and help prioritize processes for further assessment, discovery, and analysis.

Details include the process owner, a three - five sentence description of the process, a rough estimate of size and complexity, the risk associated with the process, and the level of pain experienced in the as-is process.

The following process details are needed for identification and prioritization for further assessment and discovery:

- Process owner
- Short description (three - five sentences)
- Size and complexity
- Risk
- Pain

For more information about process inventories and process discovery best practices, read the following IBM Redbooks and Redpaper publications:

- *Scaling BPM Adoption: From Project to Program with IBM Business Process Manager*, SG24-7973
- *Process Discovery Best Practices Using IBM Blueworks Live*, REDP-5111

2.9 Process Inventory, Variation Analysis, and Roadmap

PIVAR examines the processes in scope, analyzes their variations, and creates a prioritized roadmap for process implementation.

After or during an initial gathering of an inventory of processes relevant to the scope and focus of the current project, an analysis is conducted called variation-oriented analysis (VOA).

2.9.1 Variation-oriented analysis

Analyzing the business domain for commonalities and gradually separating out and externalizing variations is an important analysis activity that saves time and resources by consolidation and reuse of common elements (processes, subprocesses, rules, information, and so on). This activity is called *variation-oriented analysis* or *variation analysis* for short.

Information: Variations occur in three main versions:

- *Type*, or *data*
- *Process*
- *Rules and decisions*

Variations in type or data can be the information we store about clients when we have different types of clients. Platinum, gold, and silver level clients have some common attributes, but also might have attributes that are unique to each client type.

When you arrive at a rental car facility, for example, and you are designated as a *gold* client versus a *silver* client, the process flow can differ. You do not have to go the counter, submit identification, credit card, pick a car, and so on. With a gold client, a profile is maintained and reused, a certain class of car is specified as a default and rented. The gold client can go straight to the car, versus the client service counter, and pick up the car. They might be eligible for upgrades.

The rules pertaining to a gold client might differ from a silver client. They might get a free weekend extension, or a grace period for returning the car, and so on.

Therefore, processes typically branches at a client type into multiple subprocesses (process variation), and in each branch can have different rules that apply (rule variations).

Externalize Variations: Separate changing from less changing aspects of the business domain. Factor the common aspects into a super class, or a high-level process, rule, or data item. The variations become subclasses, interface implementations, or subprocesses, subordinate rules, or related data.

2.9.2 Steps in the PIVAR

Here is a summary of the different steps involved in PIVAR:

1. Define the scope of the effort (transformation, improvement, or optimization).
2. Construct an inventory of processes, categorized.
3. Analyze and categorize the commonalities in type and data, process, and rules and decisions. This helps decrease or *compress* the number of total processes, decisions, and data by consolidating like and similar ones into a common superclass or in a common process, common data store, and common decision or rule set.
4. Externalize the variations in type and data, process, and rules and decisions into reusable elements such as subprocesses or new rules and decisions.

Next, we take a close look at these steps.

Defining the scope of the effort

First, we define the scope of the transformation. Defining the scope of transformation can range from informal to formal. This can be a semi-informal activity, or a more formal one involving CBM or business capability analysis.

This is done by identifying and documenting which business components are most relevant and important to us in terms of the achievement of the goals of the effort: The *hot* components or business capabilities.

The hot components are business areas or components that are most relevant to the subject of our attribution. They are either exceeding a cost threshold or, for example, exceeding a revenue threshold. They might also be below a certain threshold that is defined by the attribute with which we are assessing that business component or capability.

If this follows a CBM analysis, it is referred to as a *heatmap*. More specifically, the business components' people, processes, and resources, such as IT systems, are analyzed. Also, the business component as a whole is designated to have a certain degree of an attribute, such as cost higher than a certain threshold, or sales below a threshold, and so on.

These components are tagged as *hot* components. If you are using alternative business analysis methods, such as business capability analysis, a similar technique might be used, now applied to the business capability.

The purpose is to find a subset of the business components or capabilities that should be the focus of the business transformation, improvement, or optimization effort.

Constructing a categorized process inventory

Now we inventory the business processes. These processes can either be inside the business component, or involve that business component as a process that cuts across two or more components.

We want to consolidate and reuse as many subprocesses as possible. We also want to opportunistically decrease the number of processes that are changed or automated by factoring out the commonality that exists between similar processes and their underlying subprocesses and activities.

These elements often fall into three broad categories:

- Variations in type or information
- Variations in process or activities
- Variations in rules or decisions

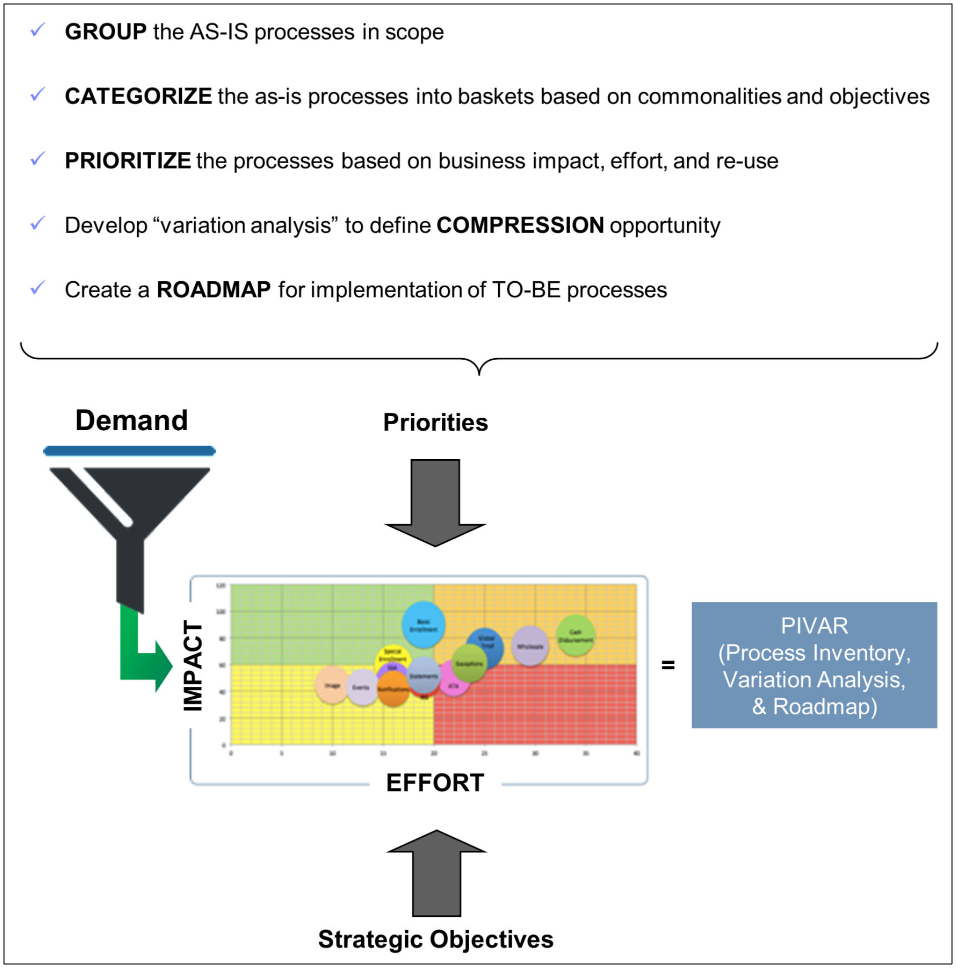
Analyzing commonalities and externalizing variations

From the overall inventory of processes, we analyze the variations and factor out the commonalities. We then create a roadmap consisting of a prioritized order of processes to implement, with a specific emphasis on those relevant for process improvement projects.

Figure 2-12 on page 60 shows the different tools that are part of PIVAR:

- *Group* As-Is processes.
- *Categorize* the As-Is processes into baskets based on commonalities and objectives.
- *Prioritize* the processes considering effect, effort, and reuse (for example, decision points).
- Develop *variation analysis* to define *compression* opportunities.
- Establish a *roadmap* for To-Be processes implementations.

Figure 2-12 depicts the PIVAR tools.



- ✓ **GROUP** the AS-IS processes in scope
- ✓ **CATEGORIZE** the as-is processes into baskets based on commonalities and objectives
- ✓ **PRIORITIZE** the processes based on business impact, effort, and re-use
- ✓ Develop “variation analysis” to define **COMPRESSION** opportunity
- ✓ Create a **ROADMAP** for implementation of TO-BE processes

Demand

Priorities

IMPACT

EFFORT

=

PIVAR
(Process Inventory,
Variation Analysis,
& Roadmap)

Strategic Objectives

Figure 2-12 PIVAR tools

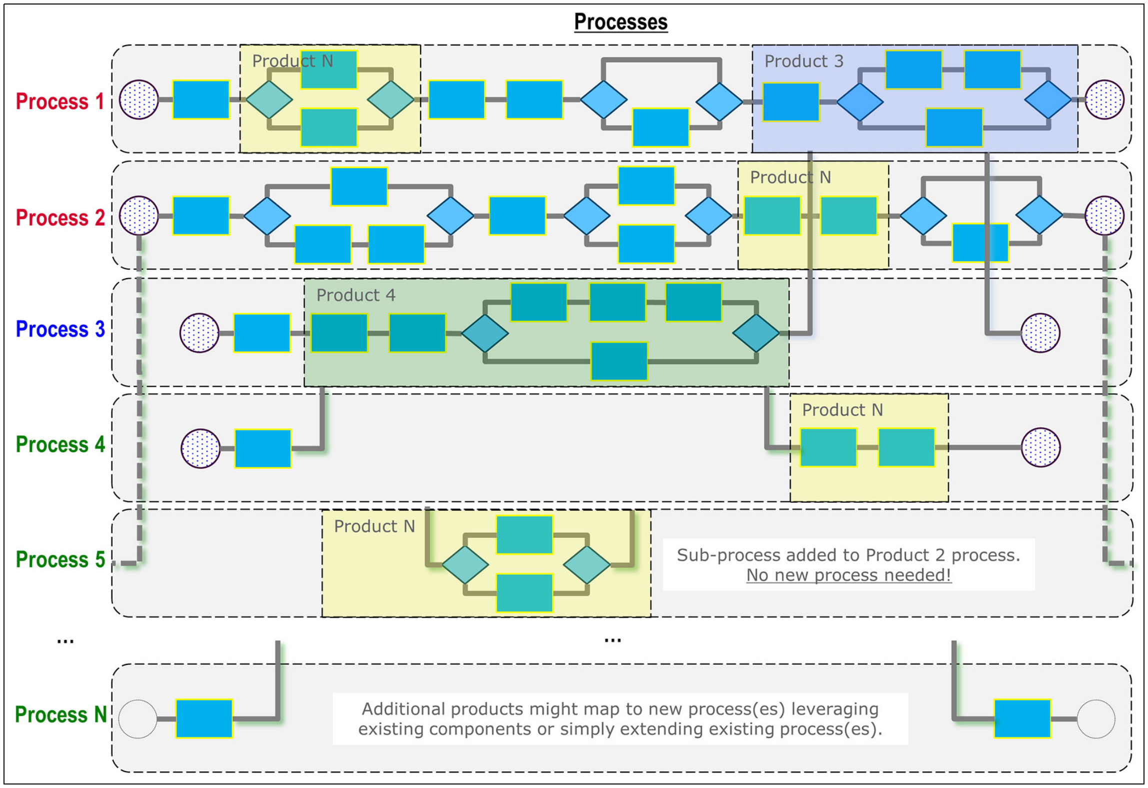
The following list describes the PIVAR tools in more detail:

- Process Inventory

  Also described in 2.8, “Process inventory” on page 56, an inventory creates an overview of all known and unknown processes of the company’s business area. It can be done on a high level, simply by describing the key points (process owner, value, pain points, KPIs, and so on) of it, or on a lower level, by creating process maps (for example in IBM Blueworks Live™. See *Process Discovery Best Practices Using IBM Blueworks Live*, REDP-5111).

- Variation Analysis

  Figure 2-13 depicts the variation analysis of PIVAR. Common components across the portfolio of business processes are identified. With this, potential for reuse can be revealed, an opportunity for reuse identified, and the chance for alignment of operational procedures created.



Sub-process added to Product 2 process.
No new process needed!

Additional products might map to new process(es) leveraging existing components or simply extending existing process(es).

Figure 2-13 Variation Analysis explores the potential for reuse by separating variations

- Roadmap

  To create the roadmap, the outcomes of the previous stages, process inventory, and variation analysis are used to create a sequence of which it is most beneficial for the company to implement the processes. The sequencing takes into account what the *effect* of each process is:

  – Business value
  – Pain points
  – Financial effect

The effect gets assessed against the *complexity* to realize such a process implementation:

  - Implementation complexity in IBM BPM
  - Implementation complexity of systems in the context of IBM BPM (systems that IBM BPM integrates with)

2.10 Blueprint or Macro Design

Before we embark on the development aspect of a project, even though we typically conduct it in an agile fashion, with planned iterations, it is a good idea to spend some time looking at the holistic aspects of the project:

- The overall breadth and depth of requirements
- The overall architecture and underlying infrastructure
- The integrations needed with back-end systems and systems of record
- Exploration of risks
- Exploration of releases

During blueprinting or, as it is also sometimes called, Macro Design, we complete the following activities:

- Explore the broader implications of architectural considerations and decisions.
- Discover integrations with back-end systems and databases:
  - Connectivity to existing systems, packaged applications, and web services
  - Information integration to systems of record
- Identify hidden requirements through more detailed analysis.
- Reassess initial estimations for the project.
- Set expectations of release dates.
- Ensure mutual understanding of shared responsibilities for various testing to production release efforts.

2.10.1 Breadth and depth of requirements

Requirements are often stated vaguely in the earlier stages of the software development lifecycle, and BPM projects are no exception. Explore the breadth and depth of requirements in terms of expected business outcomes by taking business requirements to a greater level of detail. This provides a more accurate representation of requirements, and enables us to validate our estimates of project schedule and cost.

2.10.2 Integrations and services

Integrations with back-end systems and systems of record are often not uncovered until later in the project. This often leads to rework and project timeline extensions, because the effort to integrate surpasses initial, often underestimated expectations. Verify the understanding that the result of integrations indeed meets the business goals. Look at the overall requirements, specifically focusing on integration:

1. Discover integration requirements and dig into the existing integration requirements to acquire better understanding of a state of the existing systems and services that are going to be integrated with IBM BPM.
2. Determine the nature and complexity of integration services:
   - Are existing services truly ready to be used for integration with back-end systems? Do we need to build new services, or are the existing ones adequate?
   - Is the service specification that describes the details of the services available?
   - Identify skill set requirements and resource availability to support the integration effort.
3. Explore the architectural implications of integration with backend systems and with systems of record (databases).
4. Analyze the wanted future state process and to-be solution architecture, and identify what needs to be in place to ready the integration points for implementation with IBM BPM and IBM ODM.
5. Identify dependencies between integration points and processes.
6. Identify the dependencies and cadence of iterations between the IBM BPM and the integration tracks.
7. Identify the business priorities for the integration points and BPDs. This affects the release schedule.

2.10.3 Explore the tracks

BPM is the main driving track or workstream for the project. But the Integration and Services workstream is almost as important. Without the back-end integrations to be called from the BPDs, the system does not provide business value. Therefore, adequate planning (project management and iteration cadence) and technical dependencies must be considered for each of the two main tracks (the BPM and integration tracks).

Coordination between these two tracks is also keenly important.

2.11 Project

Business process implementation is a critical step in the lifecycle of BPM. In this phase, business processes are implemented iteratively, from a business idea to a practical and executable business process application. It is important to keep the *business* part of BPM in focus during the implementation phase.

Often, the development team reverts to more traditional development styles, focusing a disproportionate effort on technical details. This situation is a common failure point for BPM projects, in which a constant vision for business value and ownership is required. Business roles are heavily involved in the discovery and documenting phases. Now is *not* the time to end their involvement.

Many of the benefits associated with successful BPM projects, such as reduced time to market, realized business value, and shortened test cycles, rely on an iterative development lifecycle to achieve them. In preparation for the iteration, we prioritize the remaining user stories in the product backlog (the total features and user stories that need to be implemented for the project) that have been left from the previous iteration.

For the implementation phase to fulfill the iterative methodology, we use a paradigm called *playbacks*. A playback is an event at the end of an iteration where the work done to date is integrated and demonstrated to the business and IT stakeholders for discussion, consensus building, and approval.

Playbacks give stakeholders the opportunities to provide feedback that drives the next iterations of process development. To perform the playback, we must agree that there is enough business-significant content to warrant the meeting.

Also key to the playback is to converge the integration stream with the BPM stream of work.

IBM BPM enables rapid playbacks of the process definition at any point in the development lifecycle. In the Process Designer tool, you develop a model for discussion and visualization, but also an actual model-driven, runtime solution. At any point in development, you can run the process, its subprocesses, and its services to validate your design.

For more information about how to set up or run a project and playbacks, read the IBM Redbooks publication *Scaling BPM Adoption: From Project to Program with IBM Business Process Manager*, SG24-7973.

2.12 Program

You have been successful with implementing one or two quick win process deployments that have a measurable effect on your enterprise value chain. Now it is important to scale this up into a program, so you can automate more processes, capitalize on commonality and reuse, and have a more standardized approach across the line of business or ideally, the enterprise.

IBM BPM is most effective and valuable when deployed at an enterprise level, integrating all processes with the value chain and extending the reach and effect of shared processes and process assets. It should be no surprise that the same principles used to improve a business process can improve a BPM program:

- Measurable business effect
- Incremental delivery
- Transparency and visibility
- Control over your business

Measure the value and effect of BPM by proving business value in short iterative cycles. *Market* that value early and often by providing visibility into that value for stakeholders and management. *Repeat* these procedures and prove that the value of BPM is not unique to a single process, department, or delivery team.

For more information about how to implement a BPM program, refer to the IBM Redbooks publication *Scaling BPM Adoption: From Project to Program with IBM Business Process Manager*, SG24-7973.

Connecting the corporate strategy with KPIs

Although monitoring your process is important on its own, processes do not exist on their own. They are part of a larger set of interconnected processes that make up the core business of a company. If you consider monitoring at a project or process level, you are missing the real power of BPM and process visibility. Your goal should be going from a single process up the value chain to the enterprise level.

Rolling up a process KPI to the corporate value chain is one way to achieve the goal of elevating all the way from project to process to enterprise level. To begin with, you look for common metrics between different processes. Your best candidates are metrics, such as cost and time, because they are almost always measured in any process.

This situation assumes that your first business process is running in production for a few months. You are now starting to work on your next business processes, and trying to extend the visibility across all processes, using common performance indicators.

Figure 2-14 illustrates the three levels of metrics. The way to determine what metrics to capture does not change.

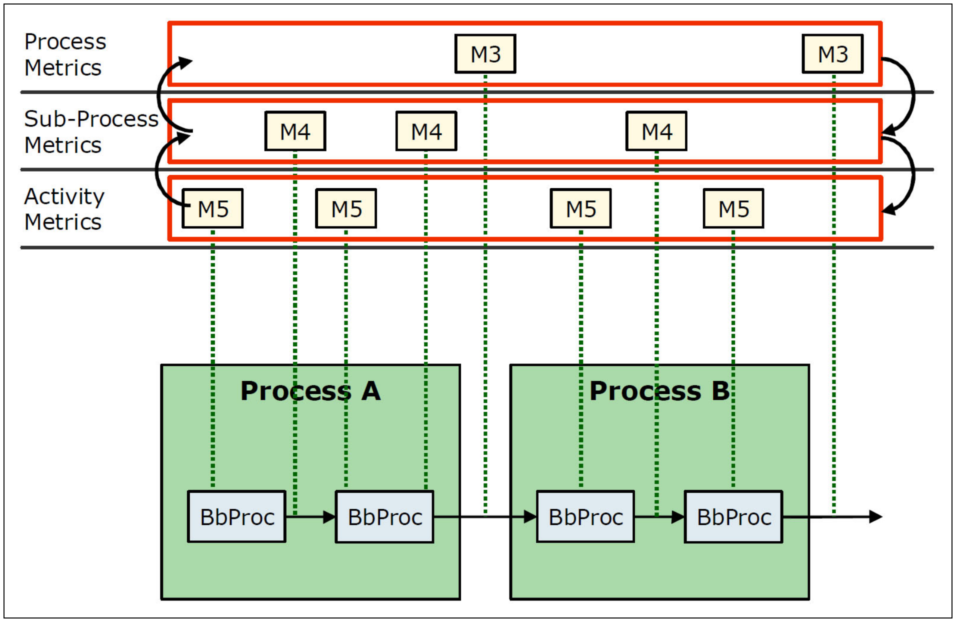


Figure 2-14 Rolling up metrics

You must start by evaluating what is important to measure across multiple processes, and not just the one process or project:

- The process metrics (M3) level represents the process metrics that roll up to metrics that are shared by multiple processes.
- The subprocess metrics (M4) level represents the metrics that you need to capture in each of your subprocesses to roll up to your process metrics. This metric can be represented by the time needed to execute the subprocess selection.
- The same is valid for the activity metrics (M3) level. You need to capture the key metrics at the activity level to be able to roll these metrics up to your subprocess and process metrics.

For more information, see the IBM Redbooks publication *Scaling BPM Adoption: From Project to Program with IBM Business Process Manager*, SG24-7973.

2.13 Process Factory

You have successfully started your IBM BPM adoption, and you successfully delivered one or more processes of different complexities. The maturity of BPM in your company increased, and you're ready to move to the high end of the adoption journey. Then, you want to consider to make use of the approach of an IBM BPM Factory.

The IBM BPM Factory, also sometimes called *Process Industrialization*, can include projects of any complexity. In this approach, you externalize parts of your projects into a factory, which for example can be a remote development team with a focus on particular capabilities:

- Implementation of human-centric processes
- Implementation of system-centric processes
- Implementation of business services
- Implementation of business rules

Because you have these different development streams, you have to make sure to account for them in your project planning activities. It is of course important to keep the interdependencies of the development streams in mind.

Here the standardization of the six pillars of BPM outlined in 2.3, "The six pillars of success for BPM" on page 29 become even more important. Using a standard method such as the Smarter Process Method becomes a critical success factor.

Figure 2-15 shows an overview of the Process Factory model.

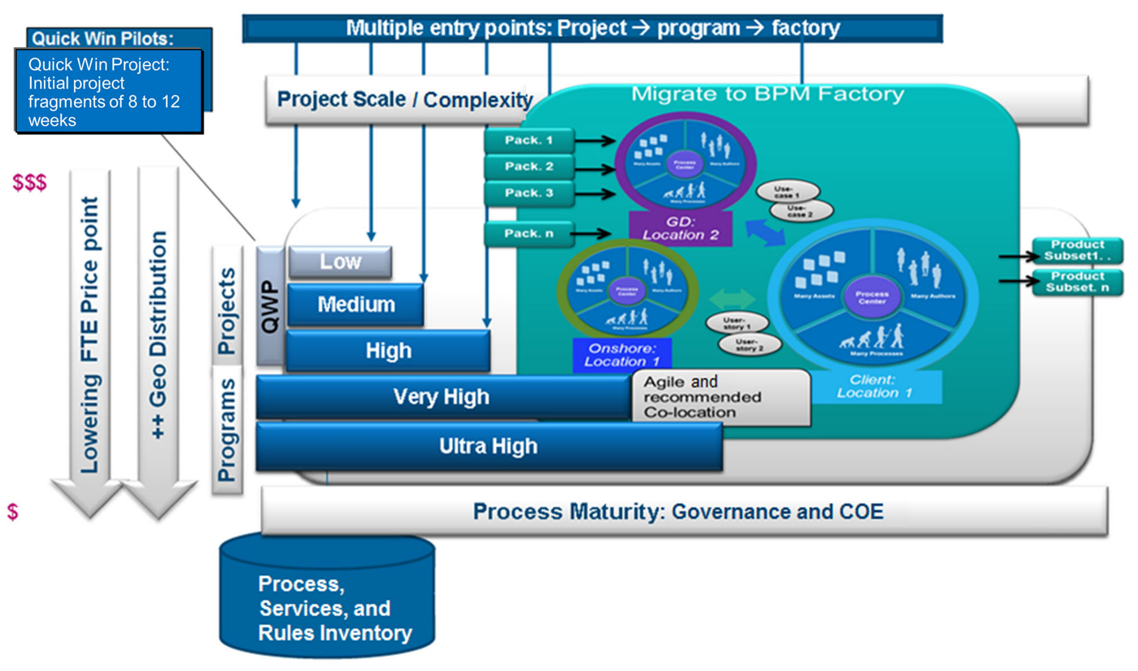


Figure 2-15 Process Factory

With the objective of lowering costs and supporting geographical distribution as we grow from projects to programs, we use standard methods to package the artifacts under a governance model with a CoE in place. The artifacts are sent to the factory for creation or assembly as work packets, and product subsets are sent back to the onshore team.

The intent of the development streams is that you can use lower-cost resources in a distributed fashion, and define concise delivery packages to deliver functionality fast and cost-efficient.

Next, We consider how this might look in a standard human-centric process development project.

Because we want to stay as agile as possible, and because when we’re agile collaboration is key, we would have our process analyst team on site with the process owner and subject matter experts to define the process and its requirements (user stories).

While this happens, the solution architect defines implementation guidelines and patterns, and helps break down the requirements in concise development packages that can be handed off to the off-shore development team. The offshore development team realize the requirements that will be played back to the business by the end of the iteration.

Typically, you keep key resources, such as the analyst and architect (at least at the beginning), closer to the business to get feedback and gain understanding. However, as your project progresses and communication models and refinement cycles got established, an even larger portion of the onsite staff can be shifted toward the factory location.

Process industrialization is an effective approach to lower the amount of high-cost resources and therefore decrease the overall cost of your development team.

2.14 Migration

Concepts of BPM or workflow management have been around for quite some time. Therefore, many organizations have a certain degree of technology support or automation for their processes implemented. But as the business changes, technology changes as well. So it often happens that the current process technology becomes outdated and must be modernized.

To get your outdated processes onto the new platform, you have to perform a migration. The following are typical types of migrations:

- Version-to-version migration

  Migrating from any old IBM BPM version (Teamworks®, Lombardi, WebSphere Lombardi Edition) to IBM BPM.

  For more information, see the IBM Redbooks publications *IBM Business Process Manager V8.5 Performance Tuning and Best Practices*, SG24-8216 and *Business Process Management Deployment Guide Using IBM Business Process Manager V8.5*, SG24-8175.

- Instance migration

  Migrating a running instance from an older snapshot to a new snapshot version.

- Platform migration

  Migrating from a non-IBM BPM platform to IBM Business Process Manager.

  Platform migration is often a migration from either third-party products or internally developed process applications, to IBM BPM.

One thing that is absolutely key to remember is that every platform has different strengths and weaknesses. This doesn’t necessarily mean that one is better than the other one, it all depends on your preferences and business needs. However, because platforms have their differences, it is rarely ever possible to simply take process A from platform P1, put it on platform P2 and have it running in no time.

Nevertheless, the hurdles are usually lower if you migrate between leading industry vendors than if you migrate from niche or custom solutions.

Why is migration so hard?

In most cases, you decide to go with a certain solution because it supports your business needs better than the other solution. In order for one solution to be better than the other, they have to be different. The differences often lie within key aspects of such a platform. Those can be of a technical nature, including but not limited to the following differences:

- The notation, such as Business Process Modeling Notation (BPMN) versus Architecture of Integrated Information Systems (ARIS)
- The interaction channels (for example, web client versus fat client)
- The type of processes (human-centric versus system-centric)

Additionally, solutions also differ in the way they recommend designing things:

- Routing patterns
- Decision logic
- Task automation

Looking at the previously mentioned factors that influence a migration, you can already see that you don’t only migrate a platform, but you also migrate concepts.

How do I approach it the best?

You have to have an overview of the strengths and weaknesses of both platforms, existing and next generation:

- Analyze the differences and evaluate the effect of them.

  You can often realize the same business value in each of the platforms, by using approaches that they support the best.

  It is important at this point to focus on the business value, and not on the technological realization of the business requirement. This is another reason why we like to make use of the concept of user stories. They tell you *who* needs *what* and *why*. It is up to the technology to decide the *how*.

  > **What to consider for migration:** Migrate the *who*, *what*, and *why*, not necessarily the *how*.

- Use the opportunity to perform process optimization.

  BPM is a discipline that is based on *continuous improvement*. Chances are that you haven’t had an improvement cycle in a while. In this case, use this opportunity to revisit the process requirements, pain points, KPIs, and service level agreements (SLAs) with your business. In this case, you can approach the migration almost like a reimplementation.

  That means, you have a discovery phase that leads into (re)implementation or migration. During the migration, you use as much as you can from the old application (for example integration services), but also implement new or changed requirements. Do this by using the information of the old application in combination with the new findings of the discovery phase.

Migrations are often underestimated. Do not make the mistake of thinking, metaphorically, that you’re simply changing the tires of your car. In a migration, you are changing the tires, engine, interior, and so on.

2.15 Top Down - Bottom Up

To conclude the chapter about project approaches with IBM BPM, we want to briefly consider Top Down versus Bottom Up approaches.

In this section, we look at Top Down versus Bottom Up from a high-level or project approach point of view:

- Top Down

  Top Down is the business-driven approach to address BPM. It focuses around process improvement, and often starts without any IT involvement. In this approach, you start to formalize and improve your operational procedures using methodologies, such as Lean or Six Sigma, often supported by process modeling tools, such as IBM Blueworks Live.

  Top down initiatives tend to include automation step-by-step, as described in “Process automation” on page 149, and often gain great value out of simple technology support to enable prioritization, measurement, and visibility.

  One way to combine Lean Six Sigma with IBM BPM is described in the following IBM developerWorks article:

  http://www.ibm.com/developerworks/bpm/bpmjournal/1308_col_schume/1308_schume.html

  Common drawbacks of the Top Down approach are to neglect the power of service integrations and a deeper level of automation. Of course, little things (like prioritization) can have a big effect. However, often with even a little more effort (integration of existing services), you can gain much more.

  Therefore, when you start to think about including a tool, also “think big”. You can always create a user story backlog that fits your wanted project timeline to maximize the value that your process improvement project can achieve.

- Bottom Up

  Bottom Up initiatives are often IT-driven approaches to BPM. This can be to consolidate multiple platforms (bring IBM FileNet®, Microsoft SharePoint, and IBM Domino® processes onto one single environment), migrate from existing applications, and so on.

  In this scenario, services (as defined in SOA governance) exist already. New ways to initiate them (for example revealed human services) or new user interaction patterns (usage of Coaches and modeled page flows, also see “Process flow, page flow, and service orchestration” on page 150) get established.

  For Bottom Up approaches, it is important to keep collaborating with the line of business, and define business value to be realized. *Any* IBM BPM initiative follows the agile principles, which include collaboration and business value.

  A *common drawback* of the Bottom Up approach are that the key aspect of agile development, business value, is neglected. This can result in costly projects that create applications that are not wanted by the users. Overall, do not forget, IT is only there to help (improve) the business.

In 3.2.2, “Top Down versus Bottom Up design considerations” on page 81, we look at this topic from a more technical perspective.

2.16 Conclusion

In this chapter, we have described some of the key concepts and approaches to making BPM a success. We described the pillars of success that are important workstreams that we need to start working on. Among them is a roadmap for BPM supported and informed by a BPM maturity assessment.

Next, we described a consistent method or approach to apply BPM. We also described the various aspects (phases, activities, techniques and so on) within IBM Smarter Process Method. We looked at how we can identify the scope and focus of a business process transformation effort, how to conduct the various phases of a BPM or Smarter Process lifecycle, and suggested practices that help us along the way.

3

Solution analysis and architecture considerations

In this chapter we provide an overview of approaches and practical considerations for several areas associated with high-level solution analysis and architecture of various aspects of a business process management (BPM) solution design.

We outline the most common challenges that BPM solution architects can come across during the initial stage of the project.

We share good practices and considerations that aim to help the IBM Business Process Manager (IBM BPM) practitioners create scalable designs for robust BPM solutions, using the IBM BPM product suite.

This chapter covers the following topics:

- High-level solution analysis and design
- Application architecture considerations
- External system of record considerations
- Integration architecture considerations
- Infrastructure architecture considerations

3.1 High-level solution analysis and design

Before you start to create your solution architecture or solution design, you need to know what this solution is all about. To do this, you must have identified your process and have been through process discovery activities:

- Identify as-is process
- Analyze pain points
- Identify To-Be process
- Capture KPIs and SLAs

The previously mentioned information helps you formulate the *goal* of the business process initiative, and with this, the goals of your architecture. (More information about process analysis can be found in the IBM Redpaper publication *Process Discovery Best Practices Using IBM Blueworks Live*, REDP-5111.

Another result of your process analysis phase is the *user stories*, which are your functional requirements. Every element of the process (the process itself, process steps, decisions, and so on) has a least one, but mostly many user stories. The importance of user stories is that they can be prioritized and estimated, which is key for successful solving and planning of a project.

More information about user stories can be found in *Scaling BPM Adoption: From Project to Program with IBM Business Process Manager*, SG24-7973.

Most often, process goals concern saving time, and saving cost by being more efficient or having more automation. This is exactly where your architecture comes into play.

By analyzing your As-Is process, you can identify process activities that can be executed more efficiently:

- You can increase the level of automation by integrating with the correct system, to gather information, store information, externalize decisions, and so on.
- You are able to learn whether additional services must be created, or whether the current portfolio is sufficient.
- You can improve the level of data aggregation or service orchestration, and make sure that it's being done at the correct level (service versus process).

As you can see from these points, we're starting to touch the broader context of our application (app). Therefore, as you are creating the solution architecture, you need to make sure that you position it on the correct level.

Therefore, learn which other applications are calling IBM BPM (inbound integrations), which ones are getting called (outbound integrations), and which technology is used to do both calls. The commonly known architecture artifact called *System Context Diagram* is your best option to create such an overview.

As in most IBM BPM projects, integrations are on the critical path, so you want to get clarity about them early on.

Also, don't forget that your application itself might require a data store, rule services, or other externalized components to be created. In sum, you have to consider components that are part of your application, and consider components that interface with your application. Based on this information, you can create the *architecture overview*. Mostly it is visualized in a layered architecture design, but the *service provider - service consumer* pattern is also widespread.

As you look closer into the process model, in addition to integration services from all components, you can start to design your *business object model* (BOM). Details about how to do this using either a Top Down or Bottom Up approach can be found in 3.3.3, "IBM BPM business object model design considerations" on page 92.

In sum, because you are in the phase of creating a high-level solution design and architecture, the following five artifacts have been demonstrated to be important:

- Process model
- User stories
- System context
- Architecture overview
- Business object model

All of these artifacts can help to define the scope of your application, and to create reliable estimates and plans to further the progress of your IBM BPM implementation project.

The following sections and chapters help you to make the correct architectural and design decision, so that the goals of the solution can be achieved.

3.2 Application architecture considerations

In this section, we describe the commonly encountered high-level architectural challenges that process architects come across on most projects while architecting IBM BPM solutions. The goal of this section is to depict items that are critical for the initial foundation of the process application solution.

Being high-level in nature, each of the following items addresses one aspect of an overall architecture. Each item can potentially have a significant effect on the solution if overlooked or not properly addressed.

We show examples, where applicable, to demonstrate each individual point as we progress through this section, covering the following topics:

- Using IBM BPM Coaches or building custom user interfaces
- Top Down versus Bottom Up design considerations
- IBM BPM Standard or IBM BPM Advanced
- IBM BPM Advanced or IBM Integration Bus
- IBM BPM rules or IBM Operational Decision Manager

3.2.1 Using IBM BPM Coaches or building custom user interfaces

Coaches are the user interfaces (UIs) for human services. Human services use Coaches to build the web pages that users see. Coaches are composed from UI controls called *Coach Views*. You can create Coach Views in IBM Process Designer. Heritage Coaches, used only in heritage human services, are composed from a fixed set of UI controls.

They are primarily for compatibility with IBM BPM before version 8.0. For processes created with IBM BPM version 8.5 5.0 and later, Coaches are advised.

For more information about Coaches, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/api/content/nl/en/ssfpjs_8.5.5/com.ibm.wbpm.wle.editor.doc/topics/ccoaches.html

For more information about Coach Views, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/api/content/nl/en/SSFPJS_8.5.5/com.ibm.wbpm.wle.editor.doc/topics/cviews.html

Another suggested resource to learn more about Coaches and Coach Views is the IBM Redbooks publication *Leveraging the IBM BPM Coach Framework in Your Organization*, SG24-8210.

In IBM BPM, human services provide the logic and user interface through which users can view and interact with business processes, cases, data, or process instances.

Figure 3-1 provides a high-level overview of the design and runtime Coach View structure.

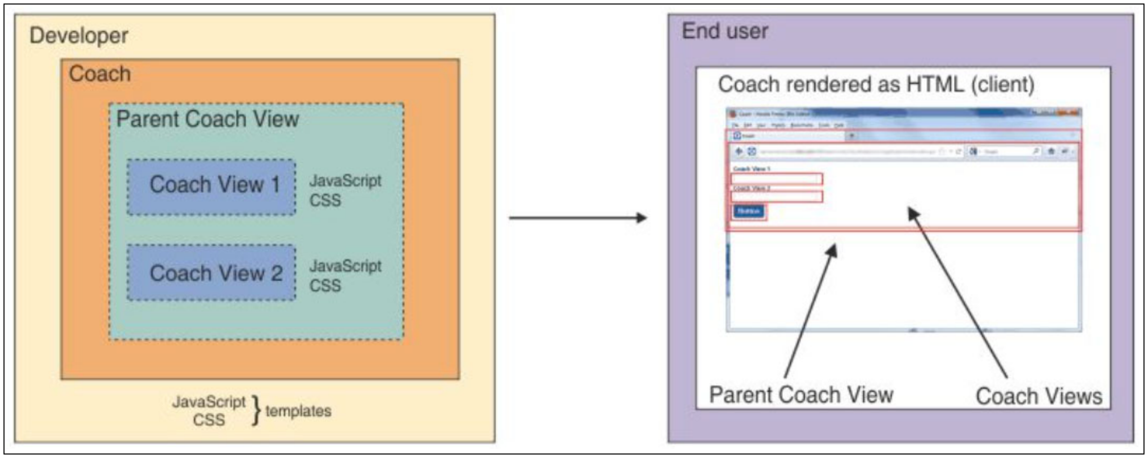


Figure 3-1 Coach Views structure

Some organizations consider not using IBM BPM Coach technology for building UIs for their process applications:

- Organizations with corporate information technology (IT) policies that limit the number of UI technologies permitted to be used
- Organizations that have invested in a single UI framework with employees skilled in the platform's specific technology

Although some organizations decide to stay with their corporate UI platform technologies, they still want to automate their business processes using IBM BPM.

IBM BPM provides a set of application programming interfaces (API) that are implemented based on Representational State Transfer (REST) services. A set of REST resources is available to access IBM BPM artifacts, including business processes and tasks.

Because there are many resources available for development using REST API at the following IBM Knowledge Center, there is no need to cover that content here:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.bpc.doc/topics/cdev_restapis.html?lang=en

Instead, we highlight some key benefits of using Coaches in an IBM BPM process application solution:

- Providing a single uniform platform for process applications, services, and UI assets source code management.

  The platform provides consistent mechanisms for managing development artifacts and a common source code repository.

- Providing a single platform for the entire process development and management lifecycle.

  The following list describes the main benefits of having a single platform:

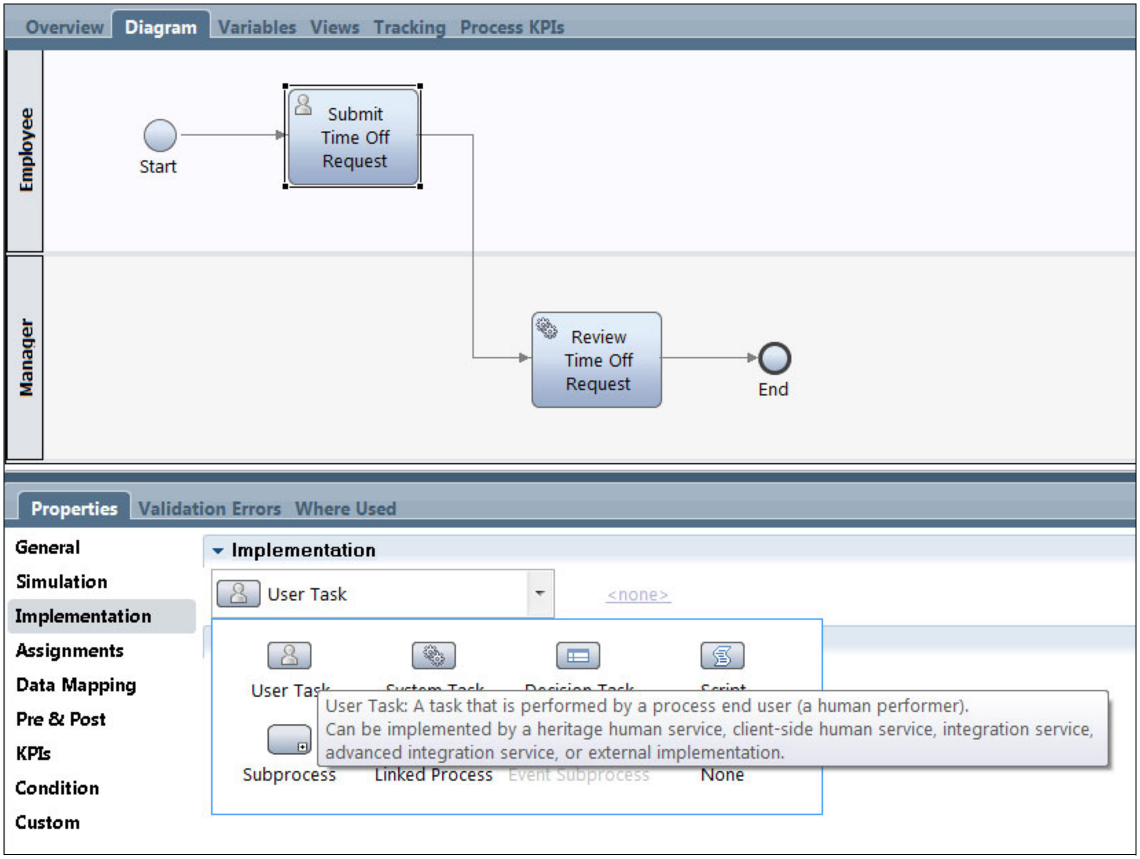
  - Integrated runtime environment for end-to-end development of process applications.
  - Processes and human services provide debugging capabilities.
  - Integrated process optimization.
  - Use of tracking points for tracking business reporting, service level agreements (SLAs), and key performance indicators (KPIs).
  - Built-in functionality for service caching and advanced team assignment capabilities.
  - Uniform and consistent development environment for process applications and human services development.
  - Integrated process application playback capabilities, following the agile IBM BPM methodology.

Considerations for headless processes

A *headless process* refers to a process definition with tasks that is controlled externally by a third-party controller, where regular processes are controlled by the IBM BPM process engine.

You can create external implementations for activities that are handled by applications outside of IBM BPM. For example, you can model an activity that is run by a custom Eclipse rich client platform (RCP) or Microsoft .NET application.

Figure 3-2 shows how external human tasks can be implemented in the IBM Process Designer in IBM BPM 8.5.5. In a similar fashion, the external system tasks can be implemented by using the System Task artifact available from the same Implementation combination box.



User Task: A task that is performed by a process end user (a human performer).
Can be implemented by a heritage human service, client-side human service, integration service, advanced integration service, or external implementation.

Figure 3-2 External tasks implementation in IBM BPM 8.5.5

For more information about using external implementations, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.wle.editor.doc/topics/using_external_activities.html?cp=SSFPJS_8.5.5&lang=en

Speaking of execution mechanics for a process with external implementations, the process instance can be started from the IBM Process Designer or IBM Process Portal at run time. However, developers or users cannot start tasks from either IBM Process Designer or IBM Process Portal. A token from the active task can only be moved to the next task, using an IBM BPM API, for example the REST API.

The controller can be an implementation of a third-party UI technology framework, or even a server-side implementation. When considering the server-side implementation for a controller logic, the implementation can be realized either by a third-party server-side framework or inside IBM BPM Advanced. For example, the controller can be a part of the Business Process Execution Language (BPEL) implementation.

Organizations who decide to go with a third-party UI technology for their IBM BPM process applications have two main approaches to consider:

- Control the headless process externally.

  In this approach, you start the process instance, read active tasks payload, and move process tokens from the external controller or a UI framework. This approach assumes that the external controller drives the entire process from outside of that process.

  Such control separation can add complexity, and can introduce certain ambiguity to an overall design. Specifically, the ambiguity is due to the presence of the process flow, as defined in the IBM Process Designer, and a secondary third-party controller driving the flow from outside of the process.

  In other words, the decision about when to finish the current task and move the token to the next one is made by the external controller. However, the controller might have its own processing to run in addition to the process logic defined in the process in the IBM Process Designer. As you can see, an additional complexity and splitting of control over the process logic is a trade-off for choosing this approach.

- The controlling function of the token movement can remain inside the process in IBM Process Designer.

  The IBM BPM process engine moves the token, but UI functions are outsourced to the external UI framework. Following this approach can be considered a better alignment with the model view controller (MVC) pattern as a good practice to adopt. Because you receive no benefits of a Coach integrated development environment (IDE), this approach can still be more advantageous and can yield more control over the overall process execution.

As an additional design option for the second approach, the IBM BPM Coaches can be built in such a manner that they can be executed both within and outside of the process context. This essentially turns this design into a hybrid approach, where the same Coaches are executed in a regular and a headless fashion in the same process application.

For example, a client wants to be able to start tasks from the portal and, at the same time, to have a UI that provides visibility into business data, including process-related business objects. The client wants to be able to start these tasks at any time, without owning any tasks, and with no respect to the statuses of the currently running processes.

3.2.2 Top Down versus Bottom Up design considerations

In this section we describe the technical design aspects of the Top Down versus Bottom Up approaches, and highlight some common challenges and considerations. When developing business processes that use services to integrate functionality from other systems, there are several different routes that can be taken.

In the following sections, we cover two conceptual approaches for integrating services. We use the IBM BPM Advanced edition as a more complex and distinct example of integrating third-party systems into your human-centric business processes. That said, the Top Down versus Bottom Up concepts are still applicable for integrations built with IBM BPM Standard.

Here we focus on use of the IBM BPM Advanced Integration Services (AIS) artifact that IBM BPM offers for advanced integrations.

Top Down process design

Under a Top Down approach, the process developer defines the interface for the service to be called, and the integration developer then creates the implementation based on the interface provided. After the integration developers complete the implementation, it is published and saved in the IBM BPM Process Center (Process Center) repository.

A Top Down approach is driven by the process developer, and reflects on the nature of process development, using IBM Process Designer. This approach places the focus on the process definition and UI design. More specifically, the design of both business object types and service interfaces starts in IBM Process Designer, addressing the requirements for the business processes and Coaches.

As a result, the same object type definitions are used in the interface definitions that are designed to support integration services on the IBM Process Designer side. In its turn, mapping between object types originated on the IBM Process Designer and IBM Integration Designer sides takes place in the IBM Integration Designer.

More information about BOM design considerations can be found in "BOM Top Down approach" on page 93.

Benefits

Next, we highlight some of the benefits of the Top Down approach:

- Support for data types and interfaces that originate in IBM Process Designer.
- Keeping the mapping effort to a minimum on the IBM Process Designer side.
- Keeping most of the mapping effort on the IBM Integration Designer side, which is equipped with more tooling options that are better suited for mapping heterogeneous data types.

Drawbacks

In most cases, service interfaces built in IBM Process Designer turn out to be different from the interfaces used in the integration layer on the IBM Integration Designer side. These services generally serve the purpose of the facade pattern, primarily supporting needs of the processes and UI. As a result, additional work is required to produce services definitions that exist on the IBM Integration Designer side and interact with the facade services.

This might not be a bad thing because, after all, such differentiation supports loose coupling between the process application and data layer tiers.

Considerations

The key idea behind the Top Down approach is to keep the process layer focused on the processes and Coaches design. Business objects and system and integration services created on the IBM Process Designer side should have the sole purpose of supporting corresponding type and service definitions, which reflect business requirements set for the organization's processes and UI.

From this perspective, Top Down is considered a preferred approach that enables you to define services and data types, which ability supports the process and UI layer, on the IBM Process Designer side. The Top Down approach also ensures that the most suitable and effective tool, which is IBM Integration Designer, is used for data type mapping purposes.

Bottom Up process design

The Bottom Up approach is defined for the scenario where the integration developer creates the service interface and the service implementation, and then makes them available to the process developer in an IBM BPM toolkit.

Benefits

The Bottom Up approach can be considered for practical use in a scenario when the organization’s canonical data model closely resembles data types and services that exist in the process layer on the IBM Process Designer side. The main goal here is to reduce the effort on the IBM Integration Designer side while keeping the mapping effort, which takes place on the IBM Process Designer side, to a minimum.

Drawbacks

One of the major downsides of this approach is shifting the data mapping effort to the IBM Process Designer side. Business objects created on the IBM Integration Designer side need to be mapped to their corresponding counterparts on the IBM Process Designer side.

Because this mapping is implemented in the Server Script assets in IBM Process Designer, it is quite a challenging exercise for more complex type definitions. This is also true for troubleshooting and maintenance efforts down the road, when the project progresses, and changes to the originally built interfaces are introduced.

Other considerations

You should make a final design decision based on your project’s specific requirements and circumstances, using the leading practices and considerations mentioned previously. We advise you to apply the facade pattern when using the IBM BPM AIS artifact.

For more information about the facade pattern implementation, usage, and strategies, see 5.5.3, “Design patterns and performance considerations” on page 190.

Several leading practices regarding use of the AIS artifact have been developed over the last several years. These practices are a result of real life business process engagements with complex system integration implementations. Lessons learned have been collected and thoroughly analyzed for every engagement’s specific scenario.

The following leading practices are some of those that resulted from that effort:

- Use the facade pattern implementation to minimize the effect of changes on the process and integration layers.
- Collaborate before defining an IBM BPM AIS:
  - Have both the process developer and the integration developer engaged during the AIS design.
  - The result from the design can extend beyond this particular process application if the AIS is intended for organization-wide reuse purposes.
  - In addition, the AIS design can have a direct effect on the toolkit design. For more information, see 5.5, “Toolkit design” on page 188.
- Connect to only one Process Center repository from an IBM Integration Designer.
- Import only necessary artifacts to the IBM Integration Designer environment.
- Protect mirrored IBM Integration Designer artifacts in dedicated toolkits.
- Frequently verify the code synchronization between the IBM Process Designer and the IBM Integration Designer development environments.
- Manage artifact dependencies in IBM Integration Designer.
- Make corresponding changes in both IBM Process Designer and the IBM Integration Designer development environments at the same time.

In addition, thoroughly analyze non-functional and integration requirements before making final design decision. Given the specifics of the AIS approach it is worth considering alternatives to AIS integration options.

For example, using web services clients can provide a lesser degree of coupling between the IBM Process Designer and IBM Integration Designer environment due to its different deployment model. Based on your project’s requirements, this can be a fair trade-off option to consider if the requirements enable such flexibility in functionality.

For more information about the IBM BPM Advanced Integration service usage see the IBM developerWorks site:

http://www.ibm.com/developerworks/websphere/bpmjournal/1106_taylor/1106_taylor.html

In this section we covered the two main conceptual integration approaches in IBM BPM. For each approach, we outlined benefits and disadvantages along with advice about leading practices. Additionally, we talked about alternative options outside of IBM BPM AIS, worth considering after fully analyzing integration and non-functional requirements.

3.2.3 IBM BPM Standard or IBM BPM Advanced

This topic highlights the design considerations that typically influence the choice between IBM BPM Standard and IBM BPM Advanced product configurations.

IBM BPM Standard can be considered when the following requirements exist:

- Need for high business involvement in modeling business processes, and rapid deployment of business processes.
- Need to implement human-centric business processes in a highly collaborative business environment.
- Ability to use business rules to drive decisions in business processes.
- Ability to proactively monitor business process performance and team performance.
- Need for basic system integration support.

IBM BPM Advanced can be considered when the following requirements exist in addition to all or some of those that apply to IBM BPM Standard:

- Need for basic case management capabilities in business processes.
- Need for straight through processing (STP) of business processes with no human interaction, and the ability to maintain process state between steps.
- Need for advanced integration support with external systems.

IBM BPM Advanced supports a variety of integration bindings, including Service Component Architecture (SCA), Java Message Service (JMS), HTTP, and web services, which are instrumental in providing complex integration features:

- Need for interaction with popular external applications having complex interfaces and arcane integration styles.

  IBM BPM Advanced provides a set of inbound and outbound adapters to ease the challenges of accessing popular application environments, such as SAP, Siebel, and so on.

- Need for domain-specific industry content packs.

  These packs are designed to integrate seamlessly with IBM BPM, providing a set of prebuilt assets based on industry standards like Society for Worldwide Interbank Financial Telecommunication (SWIFT), International Organization for Standardization (ISO) 20022, Information Framework (IFW), and so on.

  These packs help accelerate delivery of standards-based industry solutions for the banking, telecommunications, and healthcare industries, ensuring consistency and compliance across multiple lines of businesses and geographical areas.

For more information about the differences between the IBM BPM Standard and IBM BPM Advanced configurations, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.main.doc/topics/ibmbmp_overview.html

3.2.4 IBM BPM Advanced or IBM Integration Bus

This topic highlights the design considerations that typically influence the choice between IBM BPM Advanced and IBM Integration Bus.

IBM BPM Advanced should be considered when the following requirements exist:

- STP of business processes with no human interaction involving business results and business outcomes. Such processes typically require maintaining process state.
- Business processes providing services externally, for example, making a web API available.

IBM Integration Bus should be considered when the following requirements exist:

- Complex data mapping for messages.
- Need for high-throughput messaging.
- Support for a wide range of messaging and integration patterns. IBM Integration Bus offers a patterns framework with the ability to use both built-in and user-defined patterns.
- Need to implement common industry standard message formats, such as SWIFT, Electronic Data Interchange For Administration, Commerce and Transport (EDIFACT), Accredited Standards Committee X12 (X12), Financial Information eXchange (FIX) Protocol, Health Level Seven (HL7), and point-of-sale (POS) transaction log (TLOG).
- Need for a service exposure gateway.
- Connectivity between systems using a special type of protocol, such as Common Object Request Broker Architecture (CORBA) or IBM WebSphere MQ Telemetry Transport (MQTT).
- Default message throttling capability.
- Business processes using services offered by external systems.

For more information about features available in IBM Integration Bus, see the IBM Integration Bus Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSMKHH_9.0.0/com.ibm.etools.mft.doc/bm33084_.htm

3.2.5 IBM BPM rules or IBM Operational Decision Manager

In any BPM project, you implement rules to handle decisions, such as application-specific decisions, task routing, process routing, and so on. In most cases, the built-in rules capability in IBM BPM is sufficient for what you need.

However, there are cases when you can benefit from using an external rules engine, such as IBM Operational Decision Manager, to manage some of your rules. Note that you seldom externalize all of the decisions in a process. Simpler process rules tend to remain in IBM BPM. An example of this could be a simple process routing rule, such as *if the value of the order is more than $50,000 you need extra approval*.

It is important that you review and decide if you are going to use an external rules engine from the beginning of your project, because any change later can be both time-consuming and costly.

Ask yourself the following questions:

- How often do the rules change?

  If the change cycle of your rules is faster than your processes, you should consider using IBM Operational Decision Manager. The reason for this is that if you want to change one rule in your process, you have to redeploy the whole process application, and this is not always feasible. IBM Operational Decision Manager enables you to deploy changes to individual rules quickly, and independently of the process.

- Who changes the rules?

  IBM Operational Decision Manager empowers business users to change rules. If in your project the business users need to make changes often, you should look at using IBM Operational Decision Manager.

- Will other applications use the rules?

  If the rules you are building are also going to be used by other applications there are benefits to using IBM Operational Decision Manager. In this case, IBM Operational Decision Manager acts as the central repository for all of your rules. In addition, you can ensure that all applications use the same rule, which is important from a compliance and traceability perspective.

- Are the rules complex?

  The complexity of the rules that you are planning to implement has a bearing on whether an external rule engine can add value. The more complex the rules are, the more value IBM Operational Decision Manager can add.

  There are two aspects to the complexity of rules:

  - The complexity of each rule in itself:
    - How many entities does it involve (the client, the product, the channel, the time, and so on)?
    - How many conditions does it involve?
  - More importantly, the complexity of the rule set:
    - How many rules in total are you planning to create?
    - How many variations (per line of business (LOB), per country, and so on) require overriding?
    - How many steps are in the decision?
    - How complex is the rule flow?

The answers to these questions have an effect on whether you should use an external rules engine, such as IBM Operational Decision Manager.

3.3 External system of record considerations

As one of the key aspects of the IBM BPM solution design, an external system of record (SOR) can play an important role in the overall IBM BPM process application solution environment. In this section, we cover common use cases, and highlight points to consider while designing your IBM BPM process application data persistence layer.

The following topics are covered in this section:

- Introducing an external SOR into the IBM BPM process application solution
- Business entities
- IBM BPM business object model design considerations
- Accessing existing system of records
- Locking mechanism for concurrent access
- Accessing reference data

3.3.1 Introducing an external SOR into the IBM BPM process application solution

Before we get started, we define a few terms that we use as the fundamental terminology in our description of the IBM BPM solution:

- A *system of record* is a data store, hosting information in a structured fashion, enabling retrieval and updates as needed for its purpose. In most cases, an SOR is a database-based repository.
- The term *process data* refers to the body of information pertaining to the basic lifecycle functions of any business process, including but not limited to the following examples:
  - Task start time
  - Wait time
  - KPIs
  - SLAs

  The process data SOR is intrinsic to the process system. In general, this data is of most use to improving the business process when it is viewed in aggregate, rather than the specific details of one instance of the process. The *values* of the process data differ for each solution, although the *types* of data collected for a process remain the same across all process applications.
- The term *business data* refers to any information related to the specific and detailed attributes of the particular business process. These attributes are generic building-blocks of any process system, such as tasks, business process model, KPIs, SLAs, escalations, and so on, that are used to accomplish the process.

  For example, in a Submit Time Off Request process, business data includes attributes, such as Employee ID, Type of Time Off, Number of Time Off Days, and so on. The business data SOR should be distinct and disconnected from the process data SOR, because it needs to support various other consumers and subscribers to its data. This data is generally needed to determine the specifics of a particular instance of a process.
- The *execution context* includes the state of the human service while it is being run, including the current state of the variables and the position in the service flow. Effectively, the execution context, as defined previously, includes among other things the business data that is being carried through the process. For the purpose of our example here, we use the execution context and the business data interchangeably.

  Although some processes might use similar business data, it is much more common that each process has its own set of data collected as part of the interaction with the process.

Next, we continue your consideration with the following question that could be reasonably asked next. Why do we need an external SOR in the first place if IBM BPM includes an internal process database that transparently persists an execution context?

To answer this question, several aspects should be considered while conducting analysis of both functional and non-functional requirements:

- Is process business data going to be shared with other consumer systems?

  If the answer to this question is *yes*, introducing an external SOR into the IBM BPM solution becomes a good implementation option. For example, to make process execution context available for the external system, the process payload needs to be persisted at certain steps that represent specific milestones in the process. The most recent data becomes available for the external systems.

  If the external systems are allowed to make changes to the business entities that are part of the execution context, this scenario adds additional complexity. Now the process context needs to be read at the same steps to ensure that the process operates on the current shared data set.

  As you can see, the execution context is being persisted by the IBM BPM engine, *and* by our custom logic into the external SOR. This is not an uncommon real life scenario. That said, it is always important to analyze the requirements and weigh the benefits of this approach against the cost of an such implementation. Specifically, the following items are examples of things to consider:

  - Performance implications
  - Potential concurrent data access challenges
  - Complexity of the service layer services

  As a good practical approach in this case, you can discuss with the client a scenario in which the shared execution context is being persisted into the external SOR only for a subset of steps. Continuing to work with the client on identifying this subset of steps might surprisingly lead you to a realization about what the client really wants.

  The client wants the data for a particular singular activity to be shared, and not necessarily the data for the entire process duration.

- Are there any requirements to update business entities related to the inflight process instances from the external consumer systems?

  This requirement is not uncommon:

  - Clients build an external UI.
  - Clients use *headless Coaches* (UI outside the process context).
  - Clients use IBM BPM Coaches that run inside and outside of process context, or a combination of both.

For more information about headless processes and Coaches, see "Considerations for headless processes" on page 78.

- Are there requirements to track and report on inflight process instances with a significant amount of business-specific attributes?

  For example, reporting on complex attributes, such as lists or maps, requires either adding extra variables or persisting data in an external SOR:

  - Adding extra variables of simple data types usually involves manipulating the data to populate those variables for reporting purposes.
  - More often, such manipulation is not a feasible option. In those cases persisting data in an external SOR becomes an attractive option.

Separating the business process from its data has been a long-standing topic in BPM architectures. This approach enables strong decoupling between the respective lifecycles, which yields the benefit of managing each side independently without impacting the other side. A model of separate ownership of the process and business data enables a fairly straightforward mechanism.

This section described various scenarios that require introducing an external SOR. BPM practitioners should carefully analyze and consider each scenario, because there are implications associated with accessing the external SOR from the process application. The cost should be carefully estimated and weighed against the business benefits claimed by the business.

Such discussions usually involve IBM BPM solution architects, senior representatives of the client's architecture team, and business owners. You want to include these roles because the decision has the following consequences:

- A direct effect on the process application design
- A potential effect on how the process application interacts with the external systems

3.3.2 Business entities

While considering external SOR for the IBM BPM process application, it is important to look into and analyze key business entities that are involved in the process. More specifically, answering the following questions helps us to better understand the nature of the process, the data that it operates with, and the best way that this data can be retrieved and persisted:

- Does the life of a business entity extend beyond the process lifecycle?

  A business entity that continues after the process completes requires persistence outside of the process's execution context. If reporting is required on process data that includes attributes of the business entity, it is worth considering persisting the business entity in the external transitory SOR.

- What are the different states that the business entity goes through?

  Understanding various states that the business entity could be in is an important aspect of the process itself.

- Where is the business entity originated, and what is its main repository location?

  The origin and home repository of the business entity need to be well-understood. In addition to the business entity lifecycle, business entity origin should clarify the main purpose and duration that the business entity is going to be in the external process application-specific SOR.

- How much of a business entity needs to be stored and carry through the process execution context?

  If the client uses industry-specific canonical models, the main business entities can be quite large and complex. Sometimes it is not necessary and not advised to carry the entire business entity from the integration layer into the process. Analyzing business requirements for the UI, business reporting, and data persistence helps better understanding of whether only a subset of the business entity is sufficient to bring into the process application.

3.3.3 IBM BPM business object model design considerations

A BOM is a logical representation of a hierarchical data structure of an organization's corresponding data model in an IBM BPM process application.

If an IBM BPM Advanced configuration is used, there often can be two separate BOMs maintained in the solution. One BOM supports UI and human-based processes, and another BOM is implemented in the integration layer in IBM BPM Advanced. In addition to these two logical object models, clients usually maintain their own enterprise-level BOM.

There are several reasons for separating and maintaining different BOMs for process, integration, and data layers:

- Each BOM serves its own purpose.

  A BOM supporting UI and process execution context can be quite different from a BOM that supports integration with the client's back-end systems, because separate business and system requirements attributed to each side. Usually, a BOM supporting UIs and processes is a subset of a more complete and complex BOM that is required to interface with the client's external back-end systems.

- Each BOM is intended to support its domain with the most efficiency possible.

  To ensure the most responsive UI from a data exchange prospective, a BOM should only contain data that is necessary to present the content to the users and support processes execution. Anything else in addition to that affects UI performance, and places an unnecessary load on the runtime server.

Next, we consider some challenges that result from BOM separation, and discuss approaches and design options.

Business object model ownership

If a single BOM representation is shared between IBM Process Designer and IBM Integration Designer, it is important to establish ownership and approach:

- Where changes to the BOM originate from
- Where changes to the BOM propagate to the other participating party

We choose to reuse the Top Down and Bottom Up naming convention for the BOM design description. It emphasizes concept similarities. It also helps draw a parallel between the Top Down and Bottom Up approaches described earlier in 3.2.2, "Top Down versus Bottom Up design considerations" on page 81:

- Under the Top Down approach, the data is originated in and flows from the IBM Process Designer to IBM Integration Designer.

- Under the Bottom Up approach the data moves in the opposite direction.

BOM Top Down approach

The BOM Top Down approach suggests that the BOM is originated and maintained on the IBM Process Designer side. Maintaining the BOM during development iterations means that changes occur in the IBM Process Designer side, and are subsequently propagated to the IBM Integration Designer side.

One of the major benefits of the BOM Top Down approach is that it mostly eliminates the complexity associated with the mapping between the IBM Process Designer-based BOM and the IBM Integration Designer-based BOM.

Mapping required: Depending on how changes from the IBM Process Designer side get propagated to the IBM Integration Designer side, it might be necessary to perform some mapping on the IBM Process Designer side.

There are several possible variations on the BOM Top Down approach. Figure 3-3 depicts an example of the BOM Top Down implementation option.

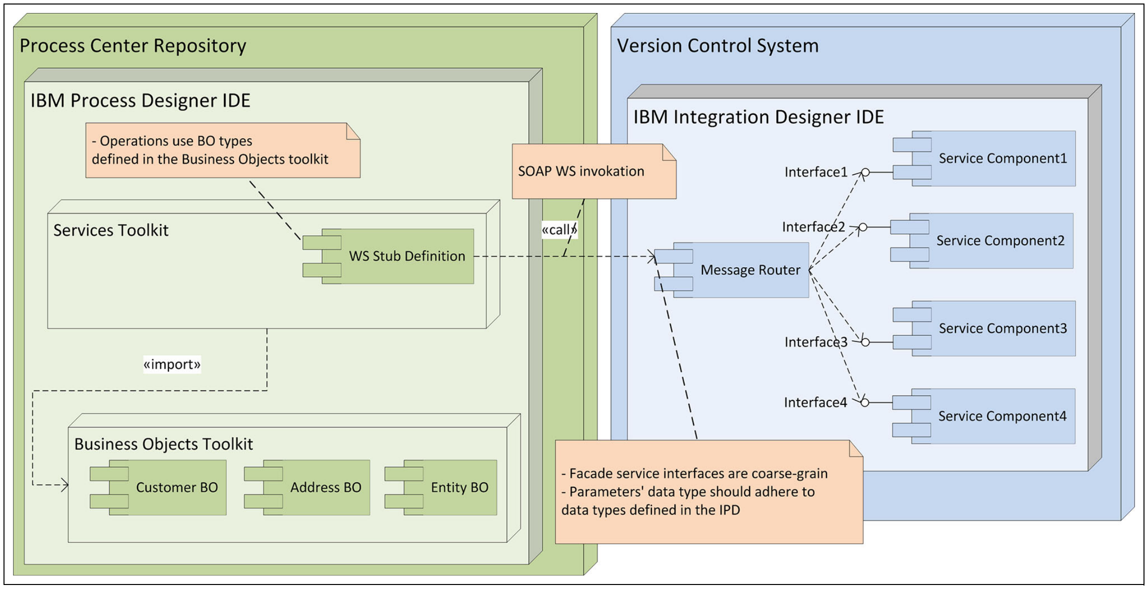


Figure 3-3 BOM Top Down approach implementation option

BOM Bottom Up approach

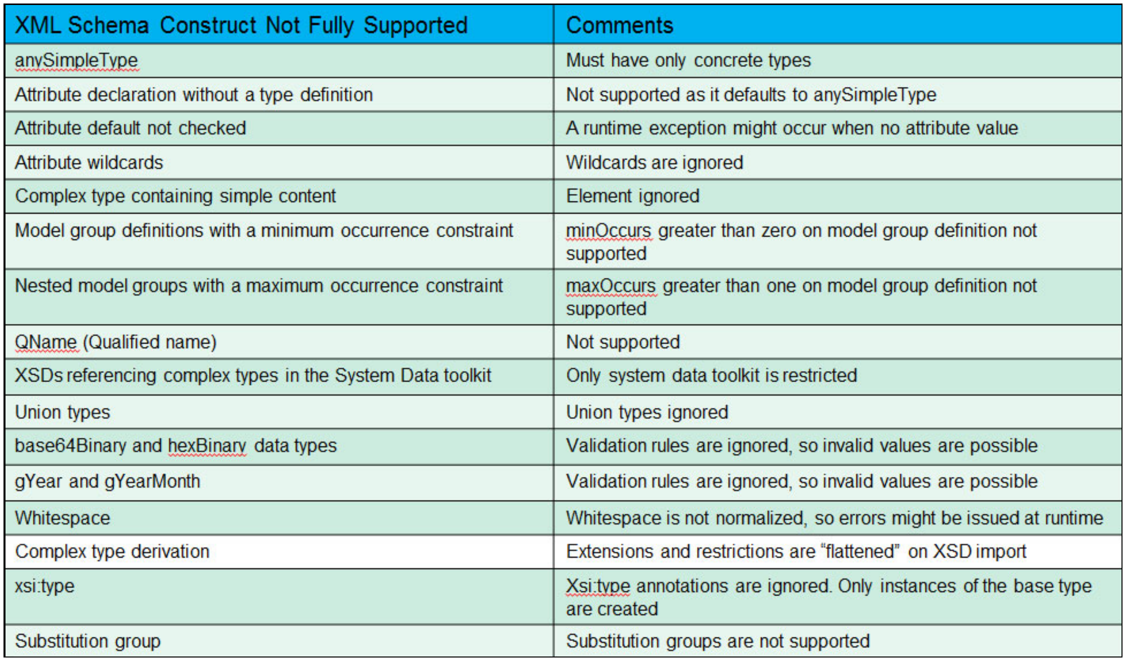
Under the BOM Bottom Up approach, from an IBM Process Designer and IBM Integration Designer relationship prospective, the BOM is originated in, and maintained on, the IBM Integration Designer side. The business objects library is being created and shared across various integration components on the IBM Integration Designer side. The following list describes the main benefits of this approach:

- Integration types, with some exceptions, can be pushed to the IBM Process Designer side with minimum effort.
- The BOM uses many-low complexity entities, which are expected to change frequently during development on the IBM Integration Designer side.

The following list describes the main drawbacks of the BOM Bottom Up approach:

- Data type mapping is necessary on the IBM Process Designer side. Because mapping is performed inside the JavaScript code, it can result in development resource demands. Depending on the nature of the changes, the resource cost can be significant, and it can affect code maintenance and potential further extension.
- Due to limitations on the IBM Process Designer side, certain type definitions cannot be successfully propagated from IBM Integration Designer to IBM Process Designer. The workaround can require using alternative compatible types, which in turn requires an extra mapping effort to populate data into these compatible types.

  Figure 3-4 outlines Extensible Markup Language (XML) schema limitations in IBM Process Designer.



XML Schema Construct Not Fully Supported	Comments
anySimpleType	Must have only concrete types
Attribute declaration without a type definition	Not supported as it defaults to anySimpleType
Attribute default not checked	A runtime exception might occur when no attribute value
Attribute wildcards	Wildcards are ignored
Complex type containing simple content	Element ignored
Model group definitions with a minimum occurrence constraint	minOccurs greater than zero on model group definition not supported
Nested model groups with a maximum occurrence constraint	maxOccurs greater than one on model group definition not supported
QName (Qualified name)	Not supported
XSDs referencing complex types in the System Data toolkit	Only system data toolkit is restricted
Union types	Union types ignored
base64Binary and hexBinary data types	Validation rules are ignored, so invalid values are possible
gYear and gYearMonth	Validation rules are ignored, so invalid values are possible
Whitespace	Whitespace is not normalized, so errors might be issued at runtime
Complex type derivation	Extensions and restrictions are "flattened" on XSD import
xsi:type	Xsi:type annotations are ignored. Only instances of the base type are created
Substitution group	Substitution groups are not supported

Figure 3-4 XML Schema limitations in IBM Process Designer

There are different implementations possible of the BOM Bottom Up approach where changes get propagated from IBM Integration Designer to IBM Process Designer. All these implementations share the same drawbacks as depicted previously.

Data mapping across different business object models

One approach to mitigate the mapping effort is for the IBM Integration Designer to maintain two separate BOMs. One is an *IBM Process Designer-oriented* BOM and another is the organization's *canonical BOM*.

The most difficult mapping between the IBM Process Designer-oriented BOM and canonical BOM is performed on the IBM Integration Designer side. Although there is still a mapping that has to be done in IBM Process Designer, from IBM Process Designer-oriented to actual IBM Process Designer model, the level of effort and complexity can be drastically reduced.

The second approach, in addition to placing mapping complexity on the IBM Integration Designer side adds another mapping to be developed and maintained on the IBM Integration Designer side. Although this workaround still seems like a reasonable option, the new mapping, just added to the IBM Integration Designer side, highly couples IBM Process Designer and IBM Integration Designer sides together.

Considering that the IBM Process Designer represents a UI and business processes side, where the IBM Integration Designer side represents the integration layer, tight coupling between the two layers is not considered a good practice.

3.3.4 Accessing existing system of records

It is common for BPM projects to have requirements to integrate with an existing SOR. The client's existing SOR usually is another integration point in addition to the external SOR that we described in 3.3, "External system of record considerations" on page 88. The reason it is an additional integration point is that, generally, the IBM BPM external SOR serves very specific purposes governed by the project requirements.

As opposed to the project-specific requirements, an existing SOR, as a rule, is built to support a specific business domain or the entire enterprise. If it is the enterprise, the existing SOR is shared within the organization by other systems.

Several important questions need to be answered while considering access to the client's existing SOR. Answers to these questions affect the architecture of the data access layer of the overall IBM BPM solution:

- How often does the IBM BPM application need to read data from SOR?
- Does the IBM BPM application need to write data back to the SOR?

- Do the other systems consumers require the most recent IBM BPM content to be available at all times, or at some periodicity?
- Does the client's SOR require transactionality while performing data persistence?

Each of these items has a direct effect on the process application's integration layer design. In addition, performance implications need to be considered as a potential side effect of the chosen design and implementation approach.

3.3.5 Locking mechanism for concurrent access

When we talk about concurrent access for the IBM BPM process application, we more often mean writing data, whether it is a process or business domain data, into the external SOR. Locking implementation strategies are varied, and selecting the correct one is highly dependent on both clients' requirements and the use case.

As a result, implementation of the specific strategy can also vary in effort and complexity. For this reason, it is important to analyze any related requirements on the subject of concurrent data access early in the project, because the access locking implementation has a direct effect on the data access services design.

In general, the same locking strategies are applicable for IBM BPM solutions as for any other enterprise-level application that requires concurrent data access. For more information and an overview of locking strategies, see the following IBM developerWorks article:

http://www.ibm.com/developerworks/websphere/techjournal/0603_ilechko/0603_ilechko.html

3.3.6 Accessing reference data

Integrating IBM BPM with the client's reference data is another common task. Reference data usually consists of data sets that do not change often, and serve the purpose of a quick lookup of the information from a specific business domain. A simple example of this is a list of states or provinces for a given country. Drop-down and combination list HTML controls are usually used to represent the content of the reference data on the UI.

When designing and implementing reference data integration points, the following implementation aspects are worth considering:

- Who owns the reference data and controls access to it?

  Establishing ownership is always important, because it sets a proper attitude toward responsibilities and dependencies for providing various resources and access to the reference data.

- How often, if ever, is the reference data expected to change?
- Is the mechanism for updating the content of the reference data owned by the client, or is it expected to be built by the project team?
- How is the reference data going to be fetched?

  As a BPM solution architect, you can ask several questions here. When designing a mechanism for retrieving the reference data, it is important to consider (among other things) the performance effect on the process application and user experience. The answer depends on several factors specific to use of the reference data in a given process application.

  For example, if the process application has many Coaches, and each of them presents a combination box or more with the reference data, fetching the reference data using an Ajax service might not be the most efficient approach to take. The following questions help to guide the decision and identify an approach that better suits the current process application:

  - Is the reference data going to be fetched for every Coach, or be passed as a payload to the human service along with the business entities?
  - Is the reference data going to be retrieved inside the human service before the Coach, or dynamically during the Coach load time through an Ajax service?

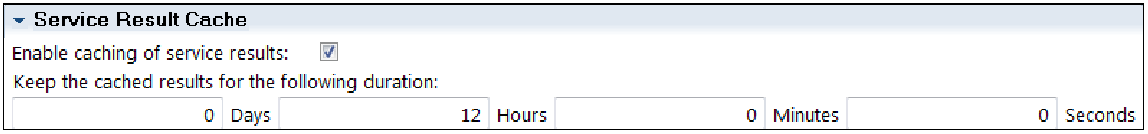
- How critical is it to instantly update the reference data on the UI when it is changed in the SOR?

  For example, if the reference data is of such a nature that it is expected only to have new items added to it, it might not be necessary to have a runtime refresh mechanism. Instead, you can let the cache expire in order to see the additions.

- Given the nature of the reference data, can its contents be cached?

  When accessing reference data from the client's external systems, it is often expensive to retrieve. An additional effort might be necessary to manipulate and compute the data to bring it to the form presentable on the UI. If the reference data does not change over time, caching can be used to maximize the performance and minimize the workload of each application tier, resulting in substantial performance improvements.

The IBM BPM default caching service results option was introduced in IBM BPM 8.0 and later. For the setting to enable caching of service results, see Figure 3-5.



Service Result Cache

Enable caching of service results: ☑

Keep the cached results for the following duration:

0 Days 12 Hours 0 Minutes 0 Seconds

Figure 3-5 Configuring service cache

For more information about enabling caching of service results, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.wle.editor.doc/topics/team_retrieval_service.html?cp=SSFPJS_8.5.5&lang=en

In addition to caching of service results, there are several options that can be considered for reference data caching in the IBM BPM process application:

- IBM DynaCache as a preferred custom option.

  For more information about using IBM BPM with IBM DynaCache see the developerWorks site at:

  http://www.ibm.com/developerworks/bpm/tutorials/1210_sharath/

- jsCache community toolkit.

- Are there existing enterprise services available to access the reference data?

  Building services from scratch requires additional time, in terms of development effort and design time. These required resources need to be properly estimated and accounted for in the project schedule.

- Is the reference data a shared resource in the client’s organization?

  If the answer is *yes*, designing enterprise-level services requires more considerations and potentially more effort, which again needs to be properly reflected in the project schedule.

In addition to these points, a reference data lookup mechanism can be quite sophisticated, depending on the hierarchical structure of the reference data itself. In that case, the data access service needs to be able to accept specific parameters to filter out the data to a certain branch of the entire reference tree.

3.4 Integration architecture considerations

Developing a new business process often involves varying degrees of process integration complexity. Requirements for process integration can range from simple web service-based connectivity with an external system to an extremely complex orchestration component.

This component must start the services in the correct order, and also handle failure, errors, and any other challenges that come along. Further complexity may be added by potentially many services on many different platforms in many different locations. Next, we share some examples of the different types of integration requirements associated with a business process, and the IBM products that can be used to address those requirements:

- *Brokerage and connectivity* requirements can typically be implemented using IBM Integration Bus or IBM BPM Advanced Enterprise Service Bus (ESB). The following list includes brokerage and connectivity requirement examples:
  - Message routing
  - Message transformation
  - Connectivity across platforms and protocols
  - Application-specific connectivity using adapters
  - Content-based message routing
- *Enterprise application integration* requirements can typically be implemented using IBM BPM Advanced BPEL and IBM BPM Advanced ESB. The following list includes examples of enterprise application integration requirements:
  - Event-driven data synchronization
  - Asynchronous data aggregation
- *Service exposition* requirements can include the following items:
  - Business processes providing services externally (acting as a service provider) can be handled using IBM BPM Advanced SCA, BPEL, and IBM BPM Advanced ESB.
  - Business processes acting as service requesters can typically use services implemented using IBM Integration Bus or IBM BPM Advanced ESB.
- *Service orchestration and choreography* requirements can include the following items:
  - STP of business processes with the ability to maintain process state and provide support for transactionality. This functionality can be implemented using IBM BPM Advanced BPEL.
  - STP of stateless services can be implemented using IBM Integration Bus or IBM BPM Advanced ESB.

After the process integration requirements have been identified, we suggest that you use the Service Integration and Maturity Model (SIMM), shown in Figure 3-6. The SIMM was developed by IBM to understand the level of maturity that is wanted for your integration initiative.

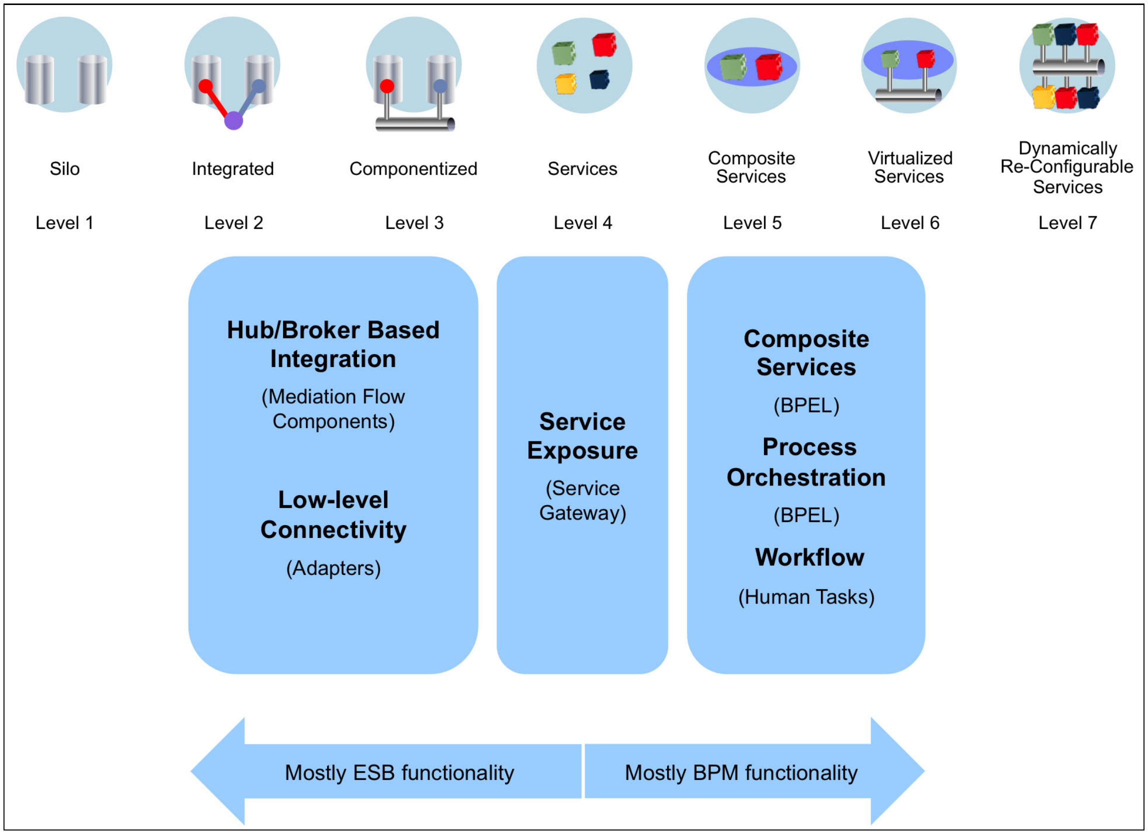


Figure 3-6 Service Integration and Maturity Model

For more information about SIMM, see the following IBM developerWorks article:

`http://www.ibm.com/developerworks/library/ws-OSIMM/`

Understanding the level of maturity that is wanted from your integration initiative is important, because it can affect your design considerations:

- When designing a service so that it meets maturity level 3, your design considerations seek to answer the question, *how do I satisfy the multiple known service requesters?* In this example, the requesters and providers are not assumed to be able to talk in a common protocol. Adapters are used to bridge the protocol or data format gap, and enable communication to the hub. Adding a new requester requires adding a new adapter.

- When designing a service so that it meets maturity level 4, your design considerations seek to answer the question, *how do I satisfy any potential service requester?* In this example, the following service design considerations are typically included:
  - Defining a standardized service exposure for protocol and transport
  - Defining performance concerns, such as expected response time and expected throughput
  - Defining exception handling behavior
  - Defining a versioning strategy
  - Ensuring high availability (HA)
  - Ensuring scalable design
  - Ensuring appropriate capacity
  - Addressing the ownership concerns of the service
  - Having an agreed-upon funding model
  - Defining a service governance strategy
  - Addressing IT-centric visibility concerns
- When designing a service so that it meets maturity level 5, your design considerations seek to answer the question, *how do I satisfy any potential requester using some fully mature and consumable services?* In this example, the following design considerations are typically included:
  - Deciding between implementing a composite service or a business process
  - Addressing transactionality concerns
  - Addressing business-centric visibility and IT-centric visibility concerns
  - Defining performance concerns, such as expected response time, expected throughput, and expected concurrency
  - Deciding when the business process typically ends
  - Defining exception handling behavior for your services and processes
  - Defining business process change management strategy and governance strategy
  - Addressing the ownership concerns of a composite service or a business process
  - Having an agreed-on funding model

From these examples, it is clear that your wanted service maturity level impacts your integration design considerations. Those considerations, in turn, can affect your selection of IBM products. For example, the connectivity use cases can be covered by products implementing the ESB pattern, where services composition and business process orchestration can be covered by the BPM products.

Figure 3-7 shows product usage types over two dimensions:

- Horizontally, the graph shows how the usage can often be correlated with your service maturity level. It becomes immediately clear from which part of the product the capabilities are drawn, with the ESB capabilities toward the left and the BPM capabilities toward the right.
- The vertical axis shows to what extent the Usage Types are message-oriented (throughput based) or synchronous (response time based). Many of the products preceding IBM BPM were one type or the other. IBM BPM supports both types of usage, but each comes with very different architectural, design, and operational considerations. Therefore, it is useful to know on which of the two styles the solution is predominantly focused.

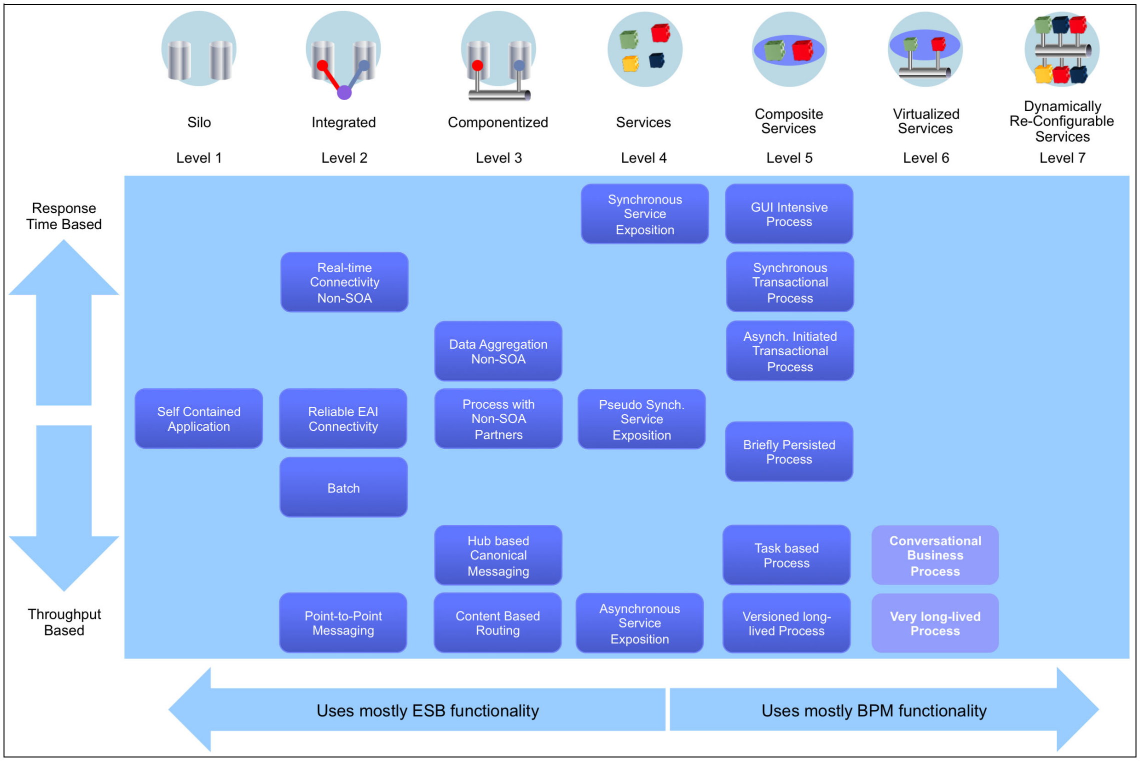

Silo
Level 1
Integrated
Level 2
Componentized
Level 3
Services
Level 4
Composite Services
Level 5
Virtualized Services
Level 6
Dynamically Re-Configurable Services
Level 7
Response Time Based
Throughput Based
Synchronous Service Exposition
GUI Intensive Process
Real-time Connectivity Non-SOA
Synchronous Transactional Process
Data Aggregation Non-SOA
Asynch. Initiated Transactional Process
Self Contained Application
Reliable EAI Connectivity
Process with Non-SOA Partners
Pseudo Synch. Service Exposition
Briefly Persisted Process
Batch
Hub based Canonical Messaging
Task based Process
Conversational Business Process
Point-to-Point Messaging
Content Based Routing
Asynchronous Service Exposition
Versioned long-lived Process
Very long-lived Process
Uses mostly ESB functionality
Uses mostly BPM functionality

Figure 3-7 Product usage types in SIMM

3.4.1 Advanced Integration Services

No matter which kind of integration you have, if you are using IBM BPM Advanced, chances are high that you use AIS components for it. It seems natural to use the IBM BPM internal AIS interface to communicate between the Business Process Modeling Notation (BPMN) model and BPEL integration flow. However, other options are possible, too. As with any good architecture decision, you want to consider all arguments and alternatives to come to your preferred solution:

- *What are the alternatives that we have?*

  The most common alternatives are to use the internal AIS interface, or web services, to communicate between the service consumer and provider.

  Figure 3-8 shows both components (SCA binding as direct IBM BPM Standard to IBM BPM Advanced integration, and web services component). IBM BPM Standard calls IBM BPM Advanced as an external application.

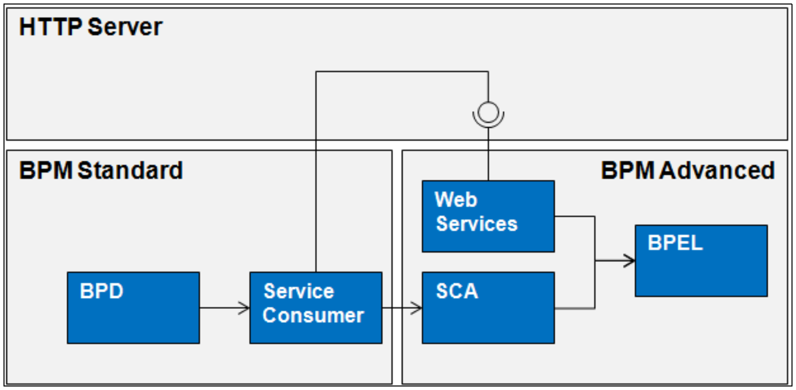


Figure 3-8 Web services and AIS

- *How are they different?*

  Although it might make sense to use SCA at first glance, when we look at it in detail, the preferable approach doesn't appear to be that obvious:

  - Use of AIS with SCA binding

    In this case, IBM BPM communicates internally between its components. You can create the service interface Top Down, so you can define it inside IBM Process Designer and make it available in IBM Integration Designer for the integration team to implement.

    It is important to know that with traditional AIS implementations (using SCA binding) the deployment artifacts of IBM BPM Standard and IBM BPM Advanced have a dependency, so, version compatibility must be ensured.

  - Use of AIS with web service binding

    The integration service is simply made available as a web service. For IBM Process Designer, it doesn’t make a difference at this point whether the web services are provided by an external application or by IBM BPM Advanced.

    Due to the loose coupling through web services, IBM BPM Standard and IBM BPM Advanced are also free from deployment or versioning dependencies. However, extra resource use might be caused by going through the HTTP Server, possibly the load balancer, and serializations.

- *Which option should you chose?*

  In summary, SCA binding creates version dependencies, where web service binding puts an additional load on the web server. It is important to evaluate both alternatives thoroughly, and create a well-defined architecture decision to create a common understanding of why either approach has been chosen.

3.5 Infrastructure architecture considerations

The key architectural considerations in building your IBM BPM infrastructure are HA, sizing, scalability, disaster recovery, security, ease of maintenance and administration, optimal use of system resources, and optimization of licensing fees. It is important that you align your infrastructure requirements with these considerations individually, as described in the following sections:

- The important components on which IBM BPM is built that provide redundancy for failover and high availability include:
  - WebSphere Application Server
  - Databases
  - Messaging engines
  - Web servers
  - Network infrastructure

  Your infrastructure architecture considerations typically include a strategy to address each of these components with your HA requirements in mind. For example, your HA requirements might lead you to pick an *active/passive* configuration for your database.

  We suggest that you review the IBM Redbooks publication *Business Process Management Deployment Guide Using IBM Business Process Manager V8.5*, SG24-8175, for more information about these key components.

- The sizing of your platform can be driven by your requirements, which can include the number of concurrent users at peak load, the number of users logged in to the system during peak hours of the day, contingency requirements, the maximum number of active process instances at any given time, and so on. It is important that the estimates you use for your sizing exercise align closely with the expected production load.

  For sizing your IBM BPM environment, see the IBM Techline Software Sizing service on the following website:

  `http://w3.ibm.com/support/techline/global/swsz.html`

  As future projects are added, we suggest re-evaluating the sizing estimates as part of the capacity planning exercise.

- The scalability considerations for your platform can be driven by your requirement to start with a few users and then expand to support many users over time. To meet this requirement, your scalability design must include the ability to easily add resources, such as hardware, network capacity, memory, and so on.

- Your requirement to recover the environment, even if part or all of the original system is lost, typically drives your disaster recovery considerations. For an example of how your disaster recovery requirement can affect your IBM BPM topology, see the following IBM developerWorks article:

  `http://www.ibm.com/developerworks/bpm/bpmjournal/1308_zhang/1308_zhang.htmL`

- Your security requirements can have a significant effect on your infrastructure architecture design. We suggest that you review Chapter 4, “Security architecture considerations” on page 109.

Often, you can find yourself in a conflicting situation where you are required to keep the licensing cost down, yet provide a larger, scalable topology.

We suggest that you review the IBM Redbooks publication *Business Process Management Deployment Guide Using IBM Business Process Manager V8.5*, SG24-8175, to help you choose the optimal topology that meets your budget for licensing costs.

3.6 Conclusion

In this chapter, we described the importance of high-level solution analysis and design by laying out a sound foundation for your IBM BPM process solution.

We described the architectural considerations, and described how a successful process solution design can be achieved using IBM BPM, by addressing the following aspects:

- Analyzing key business entities that become a foundation of the BPMN BOM. The decision has a direct effect on data pass-through within the process application, and on the success of the integration effort with external systems.
- Choosing between Top Down and Bottom Up approaches while designing a BOM that has a significant effect on the rest of the design, and on the overall development process.
- Covering practical considerations for introducing a transitory SOR into the IBM BPM process application solution that stores process data beyond the life span of the process. This makes the data available to third-party systems, and for complex reporting purposes.
- Making the correct decision in choosing between IBM BPM Standard and IBM BPM Advanced integrations, which results in the most efficient tooling choice for complex integrations.
- Sharing integration architecture considerations that have implications on the development, deployment, and runtime aspects of the final solution.

4

Security architecture considerations

In this chapter, we describe security concerns for an organization that wants to use IBM Business Process Manager (IBM BPM). We talk about common security holes that often occur in this field, and describe techniques for rectifying these holes. We show preferred practices and common security hardening exercises that you can use to achieve a reasonably well-secured IBM BPM installation.

Many of the practices described in this chapter apply equally to generic Java Platform, Enterprise Edition (Java EE) applications (apps), in addition to IBM BPM. However, we focus on aspects that typically do not receive adequate consideration in actual practice. Also, we address IBM BPM Standard and IBM BPM Advanced editions, although there are topics inherent to IBM BPM Advanced that we consider to be out of scope for this book.

This chapter is not meant as an in-depth technical review of any one topic, technology, or philosophy. IBM offers various training and consulting services that can help you understand and evaluate the implications in your organization.

This chapter includes the following sections:

- Why BPM security is important
- Installation concepts and hardening steps
- Authentication
- Authorization

4.1 Why BPM security is important

These days, security is considered generally important in every IT-related project that you embark on. To shine a spotlight on IBM BPM we look at the following aspects:

- IBM BPM is part of your “corporate DNA”
- IBM BPM users have access
- IBM BPM has unique security considerations

4.1.1 IBM BPM is part of your “corporate DNA”

Consider the fundamental need for securing your IBM BPM systems. First of all, IBM BPM is based on Java EE technologies, and is delivered largely through HTTP protocols. It therefore has the same security requirements as any other Java EE enterprise-ready application. Authentication, authorization, and sensitive data protection are all topics that are common to any Java EE application, so many casual observers might stop their inquiry there.

However, in many ways IBM BPM is not just another Java EE application. When you look at an organization’s existing software systems, you typically find applications that are built for a single-purpose.

The successful attack and penetration of these applications can reveal the process data, but it is hard to conceive of how such a breach might reveal the actual business steps, decision points, or overall operational strategy of a department’s business functions.

Figure 4-1 depicts a single-purpose application.

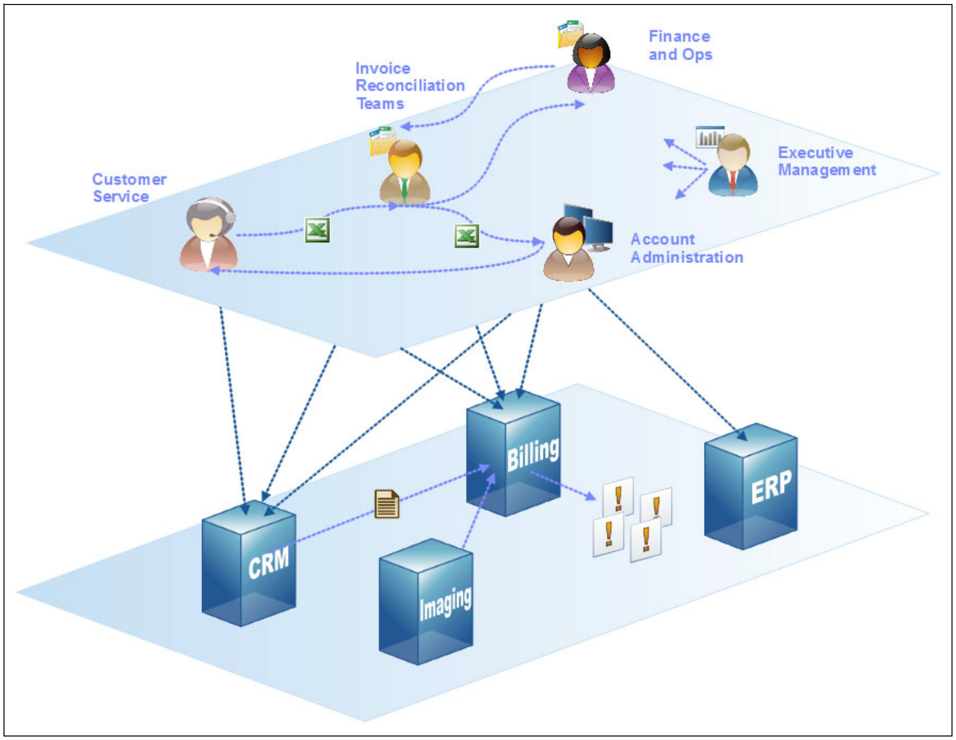


Figure 4-1 Single-purpose applications

This is not true for IBM BPM, which encapsulates more than just process data. IBM BPM process applications capture the very essence and details of a department's way of doing business. IBM BPM provides easy-to-understand flowcharts of process steps. These flowcharts show which employee groups are entitled to execute particular steps, the decision points, and details of how those decisions are evaluated.

The IBM BPM flowchart model is depicted in Figure 4-2.

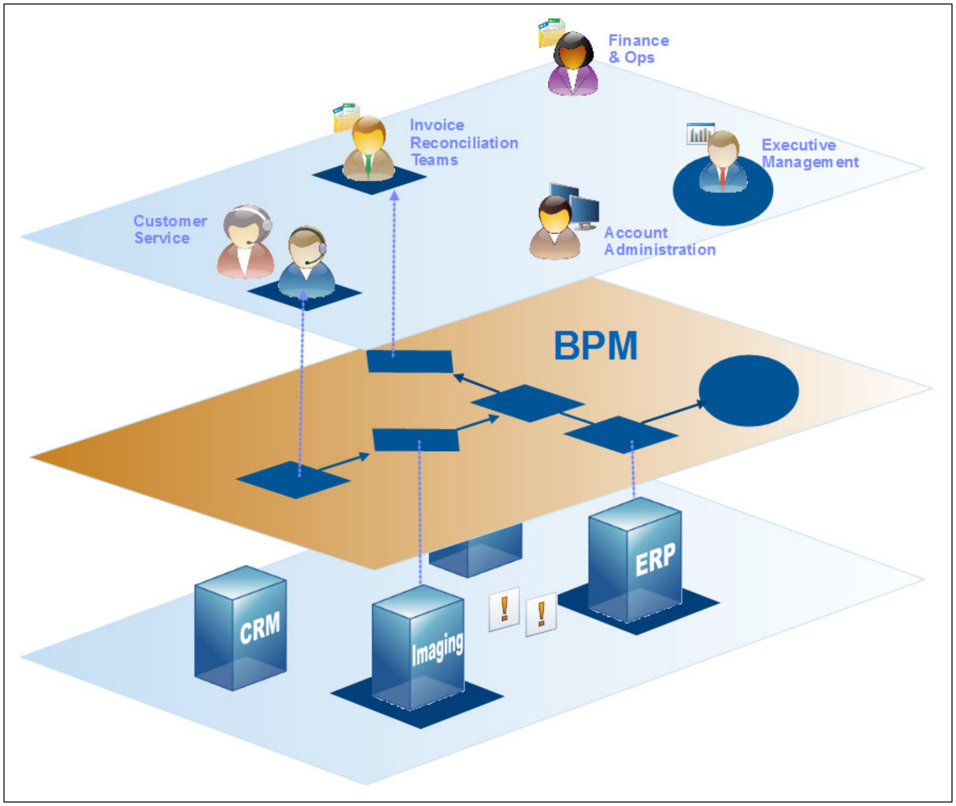


Figure 4-2 Incorporating IBM BPM

4.1.2 IBM BPM users have access

Furthermore, consider the universe of users who have access to IBM BPM. As IBM BPM spreads throughout your organization, thousands of users, possibly tens of thousands of users, might be expected to use some aspect of IBM BPM.

These users have network access and valid credentials.

They know which processes, if compromised, would be most disruptive to your business. They know which data, if delivered outside of the corporate firewalls, would be most valuable.

They might also want to bypass security policies for their personal benefit (for example, when claiming travel expenses).

The most common security hole we see is an overreliance, or faith, in corporate firewalls. It is very common to hear the phrase *our IBM BPM network is internal, so it is secured*. Yet several sources have stated that most security breaches are instigated from within an organization.

Rethink your attitude toward security. Even if you decide to trust 100% of the employees who have access to your internal networks, can you be certain that all actions they take are consistent with your security policies?

What about email or browser exploits, which could serve as an entry point to your corporate network? What about no-charge software, which could include non-trusted code?

What about CDs, DVDs, or Universal Serial Bus (USB) drives, which your employees might insert into their notebook computers? What about external consultants connecting to your corporate network? Can you ensure that 100% of these devices have been thoroughly scrutinized?

Remember that there is no way that anyone can guarantee a 100% secure environment. Notwithstanding, considering the issues that are detailed in this chapter will, at the very least, eliminate the “low-hanging fruit” and drastically reduce the universe of potential attacks against your IBM BPM systems.

4.1.3 IBM BPM has unique security considerations

In addition to these philosophical arguments, there are some practical considerations that need to be addressed concerning IBM BPM security. IBM BPM uses a unique hub and spoke deployment model that is built upon the normal Java EE deployment model, but extends it in ways that are specific to IBM BPM, as shown in Figure 4-3.

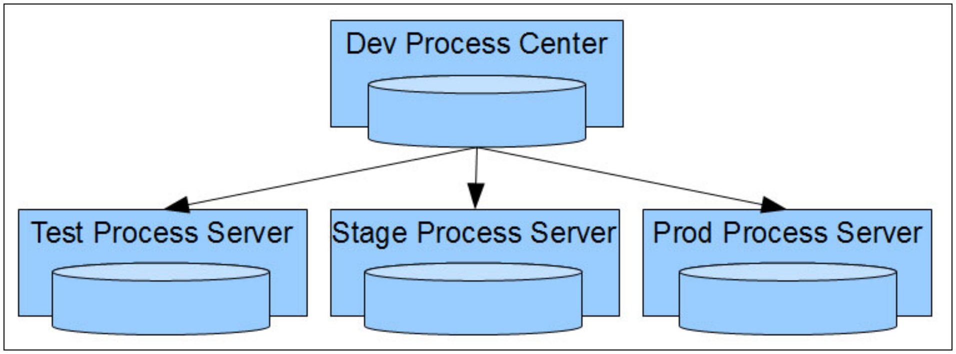


Figure 4-3 IBM BPM deployment model

IBM BPM uses a unique instance-based authorization model that adds significant functionality beyond the normal Java EE authorization techniques. IBM BPM is delivered with IBM WebSphere Application Server, which includes a very sophisticated Virtual Member Manager (VMM). The VMM enables you to access corporate Lightweight Directory Access Protocol (LDAP) and other user and group repositories.

Many applications use this group information to perform a high-level authorization. However, the IBM BPM model extends this in ways that need to be understood in order for you to fully evaluate the security implications.

Finally, the ways in which IBM BPM connects to external systems (web services, message queues, and so on) differ depending on many factors:

- Which IBM BPM product you are running (IBM BPM Standard or IBM BPM Advanced)
- Which version you are running (V7.5, V7.5.1, V8.0, V8.0.1, V8.5, and so on)
- What type of integration you are describing (outbound versus inbound web service calls)

All of these topics are addressed in detail in the IBM Redbooks publication *IBM Business Process Manager Security: Concepts and Guidance*, SG24-8027.

4.2 Installation concepts and hardening steps

Proper security begins before the software gets installed. How your IBM BPM servers fit into your organization requires careful consideration, with an eye toward security implications at every touch point. Next, we look at some common questions:

- How many IBM BPM deployment environments will you have? Are each protected by firewalls?
- Which of your deployment environments will be protected by a firewall configuration (DMZ) for securing local area networks (LANs)? Will they be protected by proxies or dedicated web servers?
- How will you secure your network traffic?
- How many certificate authorities will you trust?
- How will you secure your data at rest?
- How will you secure web service calls?

In addition to these generic questions, you also have to consider IBM BPM-specific questions:

- How many user repositories will the IBM BPM servers access?
- Does each deployment environment support different user repositories?

The installation of your IBM BPM software, including the decisions surrounding that installation, is one of the first steps in security. This is the foundation on which your IBM BPM installation's security relies.

4.2.1 BPM and WebSphere concepts

Portions of the IBM BPM suite originally came from a product acquired by IBM called Lombardi Teamworks®, and others from what had been previously called IBM WebSphere Process Server. Although both products shared a common goal of automating and facilitating the management of business processes, they had different focuses. The similarities have been mostly encapsulated in what we call IBM BPM Standard, and the differences packaged in IBM BPM Advanced.

With each new version of the IBM BPM software, these two disparate origins become more tightly integrated. Portions of the software stack that served Lombardi's need to support a wide variety of Java EE application servers are being replaced with more security-hardened IBM WebSphere counterparts. Similarly, WebSphere Process Server's user interface (UI) has greatly benefited from the joining of the two products.

IBM BPM is built upon, and is included with, WebSphere Application Server. Many security tasks are best thought of from the perspective of WebSphere, but others from the perspective of IBM BPM. Therefore, it is useful to spend a moment to consider the terminology of the two perspectives.

WebSphere concepts

Next, we begin by looking at the basic WebSphere concepts, depicted in Figure 4-4.

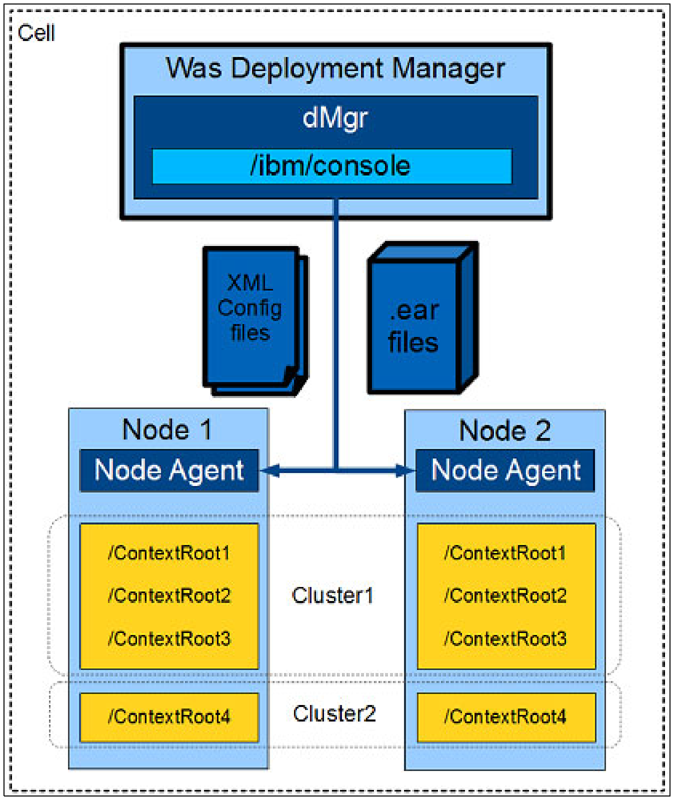


Figure 4-4 Basic WebSphere concepts

The following list explains some of the important concepts in more detail:

Profile A set of Extensible Markup Language (XML) files and application artifacts that describe and facilitate the installation of WebSphere and associated Java EE apps.

Cell The overall security context for centralized management of a WebSphere Application Server.

Node The virtual machine (VM) image (or hardware server) that hosts Java application servers. Each WebSphere instance can have multiple nodes.

App Server Java virtual machines (JVMs) where Java EE enterprise applications execute. These are software constructs.

Cluster A set of application servers, potentially distributed across more than one node, that provide failover and high-availability (HA).

Deployment Manager and Node Agents
Lightweight processes that keep the clusters in sync. There is one Deployment Manager for each WebSphere cell, and one Node Agent for each Node.

IBM BPM concepts

In almost all corporate environments, there are distinct functional (and often physical) differences between servers used for software development, testing (both by the developers and by the users) and their production servers. IBM BPM supports this common pattern with the mechanism of *deployment environments*.

We have already seen that business process applications are developed within the Process Center. The Process Center is also commonly referred to as the IBM BPM *development environment*. When the process application authors are satisfied with their current iteration of the process application, they can take a snapshot of that process application and deploy it to another environment. This next stage is typically called *testing*.

Among the benefits of this redeployment is that you separate the execution of the process application from the dependencies that might have been present in your company’s development environment. Therefore, at a first glance you get the chance to ensure that the process application has been bundled with all of the underlying software and toolkits that it needs to execute.

When all parties are convinced that the process application is free of these dependencies and is ready for wider distribution, it is then typically promoted to another environment. This step is called *staging* or *user acceptance testing* (UAT). The UAT environment is where a wider group of users is given access to the process application. It is their job to ensure that the process application satisfies all business requirements and is bug-free.

When the testing is complete, this *staged* application is eligible for promotion to the *production* environment, with confidence that the application is ready for use.

Mapping WebSphere Application Server to BPM

Each of these IBM BPM environments is their own WebSphere cell. This is not universally true, because there are special occasions where you might require a cell to span BPM environments. However, that topic is an exception described in the IBM Redbooks publication *IBM Business Process Manager Version 8.0 Production Topologies*, SG24-8135.

In Figure 4-5 you can see the dotted boxes as defining a cell's boundary, and you can also see that each cell has a Deployment Manager and one or more nodes.

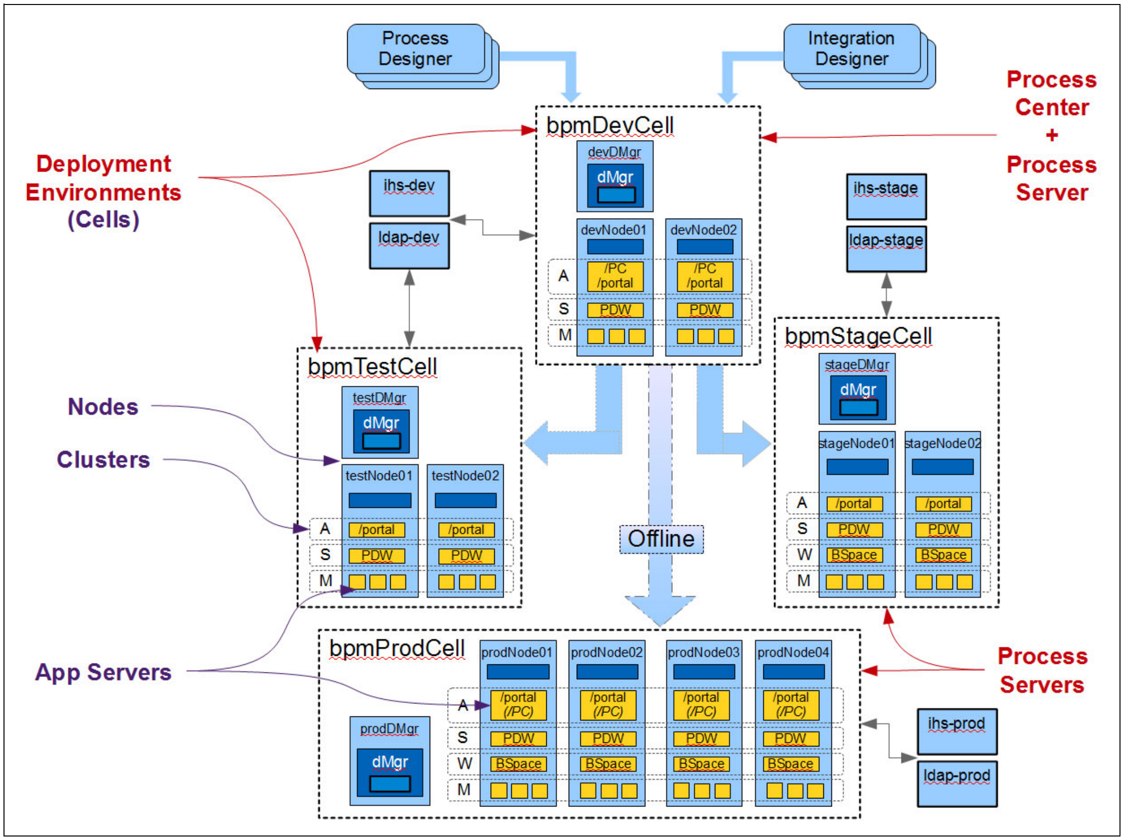


Figure 4-5 Mapping WebSphere to BPM concepts

The Deployment Manager is a WebSphere construct that doesn't map into the IBM BPM lexicon, but the nodes are where the important part of the process takes place. A node contains one or more *servers*, which are JVMs dedicated to a specific set of IBM BPM tasks.

Depending upon the type of IBM BPM topology that you choose, each node might have one - five servers that map directly to the IBM BPM concepts of AppTarget (where the IBM BPM runtime engine executes), Support (which is where the IBM BPM Performance Data Warehouse executes), and others. These servers are then clustered together across node boundaries.

In this diagram, we have named the cells according to their role as deployment environments:

bpmDevCell This cell contains the IBM BPM Process Center and central development repository. This deployment environment is often referred to as *development*, *PC*, or *the dev environment*.

bpmTestCell This cell contains an IBM BPM Process Server runtime deployment environment. This environment is where process application authors can ensure that their process applications are free from any dependencies upon development tools or artifacts. This deployment environment is often referred to as *testing*.

bpmStageCell This cell contains an IBM BPM Process Server runtime deployment environment where process application stakeholders can ensure that the process applications are isolated completely from the development environment. Typically, the staging environments closely mimic the production deployment environment.

The environments are similar because the expectation is that when the staging testing is completed, the process applications can be installed into production environments with no issues. This deployment environment is often referred to as *staging* or *UAT*.

bpmProdCell This cell contains an IBM BPM Process Server runtime deployment environment where process applications are deployed for use in the enterprise. Often, these environments are behind strict firewalls, and have special deployment considerations. This deployment environment is often referred to as *prod* or *production*.

These distinctions between test and staging are largely superficial. There is no difference in functionality between them. Often, organizations have distinct ideas about how test and staging are treated. Some organizations incorporate yet another *stage* environment, called *pre-production*.

By having each deployment environment hosted within a unique WebSphere Application Server cell, the cell's security configuration can be differentiated from that of other deployment environments or cells. For example, you would almost certainly want the staging LDAP server and the production LDAP server completely isolated from those of the development and testing environments.

In addition, in Figure 4-5 on page 118 you can see that the dev and test environments share a common LDAP server (ldap-dev). However, staging and production each have their own LDAP servers (ldap-stage and ldap-prod). Furthermore, both stage and production are four-cluster topologies, where dev and test are only three clusters. The production environment is also the only environment where the deployment manager (dMgr) is broken out onto its own server (physical or virtual).

This is not a strict requirement, but simply an example. IBM BPM and WebSphere enable a great deal of flexibility in how you define your IBM BPM environment.

For more information about this topic, see the IBM Redbooks publication *IBM Business Process Manager Version 8.0 Production Topologies*, SG24-8135.

Complex realities

However, actual environments are rarely as orderly as the example in the diagram in Figure 4-5 on page 118. The complications multiply when you consider the environment into which IBM BPM is being deployed. Each IBM BPM deployment environment typically has the following components:

- Its own database server
- A front-end web server
- Hardware load balancers
- Single sign-on (SSO) solutions (which might also be hardware)
- Servers that host web services
- Any number of corporate enterprise information system (EIS) servers

An example IBM BPM deployment environment is depicted in Figure 4-6.

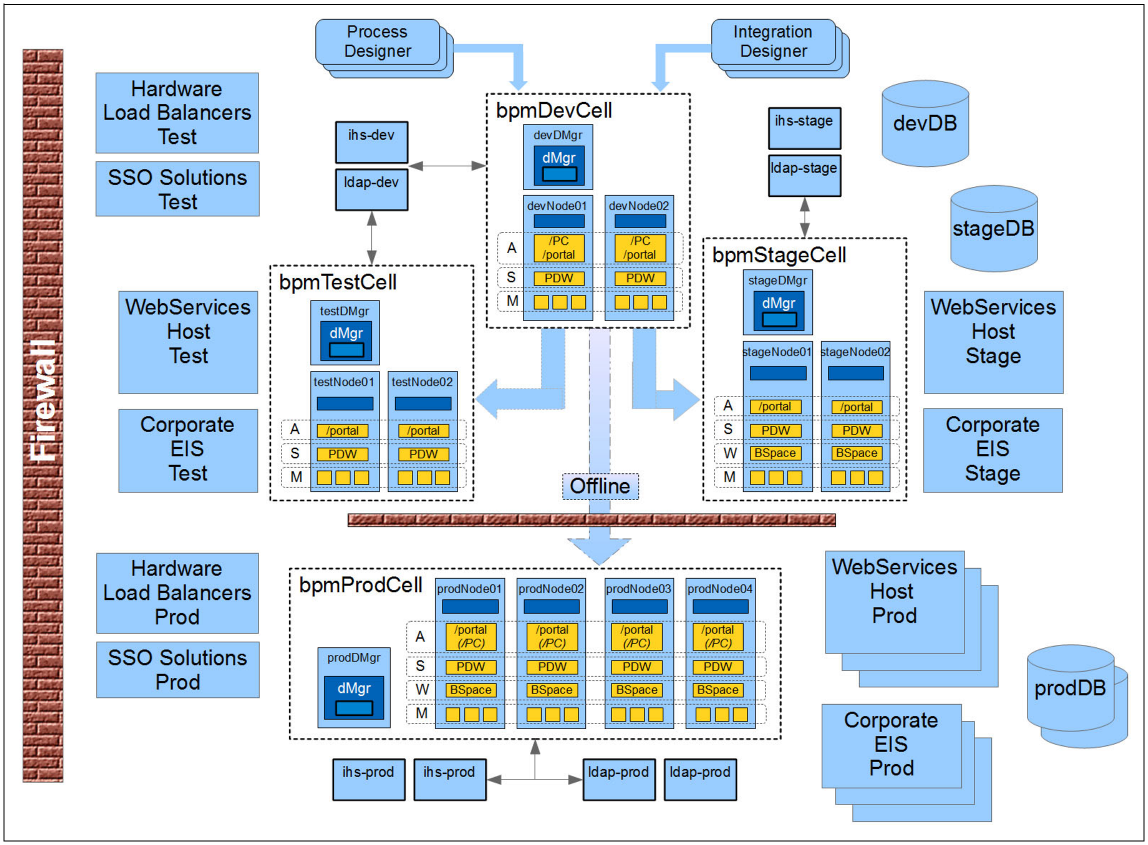


Figure 4-6 Complex realities of an IBM BPM solution

All of these components communicate with each other, and each line of communication introduces another touch point that needs to be scrutinized from a security point of view. Complexity breeds fragility, which in turn breeds opportunity for exploitation.

The antithesis, of course, is simplicity. But how can you achieve simplicity when faced with so many product components, each one speaking to another over different protocols?

Complexity is tamed through decomposition. This is achieved by breaking down the universe of connection points and possible security holes into easily understood components, with an overall framework for management.

Architectural documentation: We strongly advise that your organization invests in architectural documentation that describes the nature of each communication point, together with a discussion of communication options and a rationale for choosing one option over another.

Documentation of this nature provides an excellent repository of knowledge and expertise, which can enable all interested parties to understand the security choices that have been employed at each connection point.

4.2.2 Hardening steps

To get started in hardening your IBM BPM deployment, you need to consider the following steps:

- Secure Sockets Layer everywhere
- Using a DMZ
- Securing your database
- SSL certificate host name verification
- Hardening steps are specific to an organization

Secure Sockets Layer everywhere

Beyond a doubt, the biggest security hole we see is an overly optimistic belief in the security of a corporate, perimeter-wide firewall. Time and time again we hear the phrase *it is the internal network, so it is secure*. This is a very dangerous and precarious posture to take. It is akin to “placing all of your eggs in one basket”.

Can you trust with 100% certainty that your firewall vendors will never release a software update that has a security hole in it? How often is your notebook’s operating system updated with security fixes?

The simple fact is that many studies, from Gartner, Ponemon, the US Federal Bureau of Investigation (FBI), and others, have determined the following facts about security:

- The majority of attacks originate from within.
- The attacker’s identity is rarely known or discovered.
- Security breaches are equally likely to be caused by employees as by external agents.
- An overwhelming majority of attacks are the result of an exploitation of a misconfiguration, or a failure to follow leading practices.

Security breaches do not have to be the result of malice. They could be the result of simple, honest mistakes. But in the end, the intent does not matter. The security breach occurred, and you must deal with the consequences.

Notwithstanding, it is still prudent to consider the following question:

Is it safe to believe that not one of your employees would ever steal data?

The problem we face with hacker activist (*hactivist*) groups like Anonymous is that they are, in fact, anonymous. Until an arrest is made, these people remain nameless. Estimates on the number of arrests (and therefore name disclosures) as a percentage of security attacks are hard to come by.

However, it is reasonable to assume that more attacks occur than arrests. This leaves us with the realization that most security breaches are instigated by individuals who remain anonymous. They could be anyone.

Furthermore, even if you choose to trust that there is not one employee within your ranks who sympathizes with hactivists, can you ensure that 100% of your employees always follow corporate security guidelines? What about email or browser exploits that could serve as an entry point to your corporate network?

What about no-charge software that could include non-trusted code?

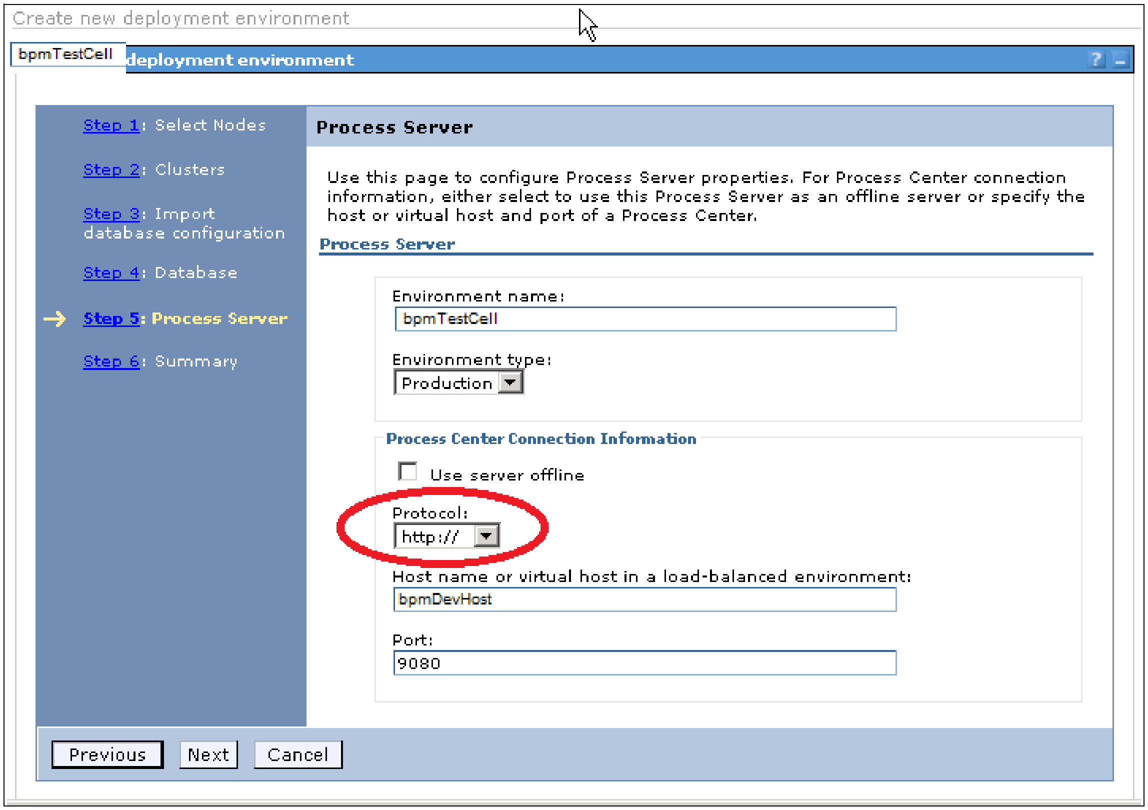
What about CDs, DVDs, or USB drives that your employees might insert into their notebooks? Can you ensure that 100% of these devices have been thoroughly scrutinized?

The bottom line on firewall security is this: It is necessary, it is very helpful, but it is not a stand-alone solution to enterprise security. *Firewalls are necessary, yes, but they are not sufficient.*

Our goal is to secure each touch point, ensuring that the *low hanging fruit* is eliminated, thereby radically reducing the potential number of attackers by simultaneously increasing the skills required to exploit a security hole.

Next, we provide you with an example of an overly optimistic faith in firewalls. During the installation of a Process Server, you (or your IBM consultant) used the WebSphere Application Server Integrated Solutions Console and executed a wizard to create the Deployment Environment. This wizard includes a step where the Process Server specifies the host name of the Process Center that it will be using as its repository.

This step is illustrated in Figure 4-7.



Process Server

Use this page to configure Process Server properties. For Process Center connection information, either select to use this Process Server as an offline server or specify the host or virtual host and port of a Process Center.

Process Server

Environment name: bpmTestCell

Environment type: Production

Process Center Connection Information

☐ Use server offline

Protocol: http://

Host name or virtual host in a load-balanced environment: bpmDevHost

Port: 9080

Figure 4-7 Specifying HTTPS during profile creation

Note that the Protocol defaults to `http://`. During Process Server startup, the runtime environment uses this information to communicate back to the Process Center, notifying the Process Center of the runtime Process Server's availability to receive deployments of process application snapshots. This communication between Process Server and Process Center includes a URL, a user account, and the corresponding password.

This information is all an attacker needs to know to deploy new snapshots of process applications, effectively changing the way you do business. An attacker could also deploy his favorite malware application. This malware monitors the network, carries out denial of service attacks, and spreads other types of malware to other systems in the same network (not only systems that your process applications connect to, but basically any systems that can be reached).

If you do not take the extra step to specify the `https://` protocol in the preceding figure, you will be sending your IBM BPM administrator account name and password in clear text. There is simply no substitution for taking the time to ensure that all IBM BPM product components communicate with each other, and with all external systems, over Secure Sockets Layers (SSL).

Using a DMZ

We advise that you install the web server or reverse proxy *physically* in front of the WebSphere clusters. Specifically, the web server or reverse proxy server should be installed on a physical server that sits between two firewalls, in what is commonly called a *DMZ*.

The outer firewall enables only SSL port traffic (typically 443, but this can be customized). The web server or reverse proxy server then communicates with the IBM BPM servers over a *different* SSL port through the inner firewall.

This accomplishes three things:

- This arrangement adds another layer of obfuscation between the potential attacker on the outside of the firewall, and the ports that are exposed to receive the *actual* IBM BPM traffic.
- It eliminates the need for a node agent on the web server or reverse proxy server, further simplifying what these servers have to do. The simpler the server, the easier to keep it free from security vulnerabilities. Note, therefore, that the installation team needs to use *unmanaged web servers* and ensure that the `plugin.xml` file gets kept up to date on the web server as a manual process.
- It eliminates the need for a JVM on the web server or reverse proxy server. This is a huge consideration for many organizations. The thought behind this is again that it is keeping the web server or reverse proxy server *simpler*, and therefore less likely to create a security vulnerability.

Securing your database

Everyone recognizes that database user accounts should be protected with passwords. What most fail to recognize is how incredibly easy it is to observe database traffic while it is in transit. If attackers can view unencrypted text on its way to the database, they also have the opportunity to store this data in their own systems. They can also gain knowledge of how your Structured Query Language (SQL) statements are formed, possibly leading to SQL injection attacks.

The solution to this is simple: SSL.

The specific steps to ensure SSL between your IBM BPM and database servers will be, of course, unique to the database vendor that you have selected, and to your organization's certificate management strategy. Notwithstanding, the concept is simple enough, and should be familiar to your database analysts and administration team.

Furthermore, IBM BPM is highly database-driven, and much of the user-facing code executes within a web browser as JavaScript. Without disparaging JavaScript as a language, it is safe to say that any meaningful encryption algorithm is almost certainly going to overpower the runtime JavaScript engines found within any browser.

Although it might be possible to overcome some of this by performing the encryption and decryption on the server, and then sending the data to the browser in clear text, the fact remains that this is a moot point. Whether it is because of performance concerns, compatibility constraints, or simple design decisions, IBM BPM does not offer this strategy for encrypting data at rest for any data elements that are intrinsic to the inner workings of the product.

Encryption in the client is very rare. Typically, you want the server to work with the data, so the server needs the data in clear text.

It might be possible to design your business process applications to encrypt and decrypt data using the extension points provided by IBM BPM, but again, this is most likely just a theoretical consideration. The computational resources required to do so within the confines of a runtime JavaScript engine are formidable.

Perhaps even more important is that you cannot know with absolute certainty what percentage of your data might be cached somewhere within the IBM BPM database tables. This caching effectively nullifies your encryption.

Therefore, it is prudent to consider two other strategies for effecting the encryption of your data while at rest:

- Operating system and file system encryption
- Database encryption

A more complete description of these topics can be found in the IBM Redbooks publication *IBM Business Process Manager Security: Concepts and Guidance*, SG24-8027.

SSL certificate host name verification

More recent versions of the IBM BPM applications (including V8.0 and later) include extra hardening steps that were not present in previous versions. For example, previously, it was possible to create the IBM BPM profiles using *localhost* as the name for the BPM server.

There are many scenarios where BPM *talks* to itself (heartbeats, deployment of process apps from Process Center to Process Servers, the WebInspector REST facade, the federated Representational State Transfer (REST) application programming interfaces (APIs), and so on). The product now requires that the host names of each part of these communications match the names in the SSL certificates.

In general, we advise that great care should be taken to ensure that the certificate's common name (CN) matches the actual fully qualified name of the IBM BPM servers.

Hardening steps are specific to an organization

Each new release of the IBM BPM application incorporates more hardening steps by default than previous versions. However, given the fact that most IBM BPM installations are integrated into complex corporate environments over so many different touch points, it is highly advised that the securing of your installations be the topic of a dedicated effort.

It would be impossible to detail every hardening step that anticipates every corporate integration possibility. The IBM services organization has various mechanisms by which they can help you to evaluate the security of your IBM BPM installation.

4.3 Authentication

Authentication is the process of proving the identity of the user who (or even another computer system that) is requesting access to software. The most common type of authentication is, of course, the user ID and password. However, there are other methods of authenticating to a server.

Simply put, there are three categories of authentication:

- What you know: Passwords, session IDs
- What you have: Digital certificates, hardware passcode generators
- What you are: Biometrics, such as fingerprints and retinal scans

For each of these types of authentication, you need to consider the following questions:

- Where is the system of record that holds user accounts?
- How will the authentication information communicate with this system?
- How will you detect a compromise?
- How will you revoke access?

In light of recent research showing the high incidence of security attacks that originate from employees or contractors, the choice of how users authenticate to your corporate systems should be evaluated carefully. Often, the choice of how users authenticate to corporate systems is decades old, and it might well be worth reviewing the policies at your organization.

Regardless of which authentication mechanism (or combination of them) your organization uses, the ensuing steps are the same:

1. The user (or other computer system) presents some authentication information.
2. WebSphere Application Server validates that this information is correct and current.
3. WebSphere Application Server creates a security context (a subject and principal) on behalf of the user that accompanies all future requests during the session.

After focusing on installation and foundational issues in 4.2, “Installation concepts and hardening steps” on page 114, this section should be considered your second step in securing your IBM BPM environments. In this section, we investigate who has access to your IBM BPM applications.

4.3.1 WebSphere user registry

IBM BPM relies entirely on the WebSphere Application Server mechanism for user authentication. When a user attempts to access IBM BPM, the underlying WebSphere security context recognizes that this is a *protected resource*, and the process of identifying the user begins. The user must already be defined using a mechanism that WebSphere can access.

First, a word about repositories and registries. These two terms are often confused. A *repository* is a concrete set of items (in this case, the *list* of user accounts). A *registry* is a reference to something (in this case, the repositories within the user registry). It might be helpful to visualize this as an analogy: A library’s card catalog is a registry, and the library’s stacks of books is a repository.

You can configure WebSphere Application Server's user registry to use any of several authentication mechanisms:

- A flat-file repository (not recommended)
- A corporate LDAP repository (most common)
- A set of corporate LDAP repositories (enables complexity)
- Windows Integrated Authentication, such as Simple and Protected Generic Security Services API Negotiation Mechanism (SPNEGO) or Kerberos
- SSO (very popular), including CA SiteMinder, IBM Access Manager, and Security Assertion Markup Language (SAML)

4.3.2 Flat-file repositories

IBM BPM includes and installs with a single federated registry, which is called *Federated Repositories*. At installation time for Network Deployment environments, it consists of just one single flat-file repository, called the *defaultWIMFileBasedRealm*, depicted in Figure 4-8.

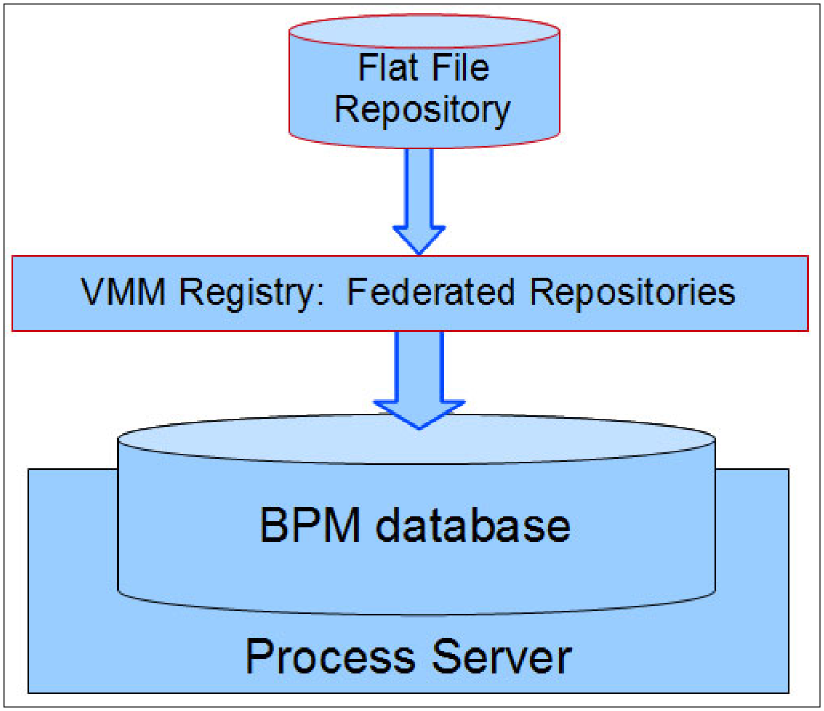


Figure 4-8 Default user repository

The flat-file repository was originally provided as a mechanism to facilitate the immediate securing of a WebSphere Application Server environment at installation time. The repository provides user IDs and passwords for administrative accounts and associated functionality.

Without such a mechanism, there would be a gap in security before the WebSphere Application Server installation could be integrated into an organization's existing security infrastructure.

The flat-file repository is not intended to scale beyond 200-300 users, but it does provide usefulness even in large-scale production environments, where it might be wanted to maintain the BPM administrative accounts locally, so that the BPM environment can be administered in the case of a failure to reach the main repository.

4.3.3 LDAP repositories

By far the most common type of user list in use by corporations today is the LDAP repository. It is also commonly called a directory of user information. It is often described as a database, but be aware that LDAPs are a specialized type of database, with characteristics that set them apart from general-purpose relational databases.

Optimized for read-only access

One of the most important specialized characteristics is that an LDAP repository is read much more often than it is updated. Applications and other users might frequently look up an employee's user ID, name, or phone number, but these values rarely change for this employee. This fact gives rise to the following generalizations:

- LDAP is optimized for read-access.
- LDAP stores static information.
- LDAP does not store application-specific data.

Unlike a general-purpose database, which might be used for any number of high-volume, high-update applications (such as an airline reservation system), an LDAP server can be optimized for read (and search) access. In fact, it is fairly common for the administrators of the LDAP repository to have strict guidelines over which software applications are even allowed to write to the repository.

Often, corporate LDAP systems are considered the system of record for employee data. Because of this, LDAP administrators are typically very careful about what type of data should be stored in the LDAP. This is an important consideration for using an LDAP with IBM BPM. Every IBM BPM user has an inbox that receives business process tasks during a typical work day.

However, even in an organization whose IBM BPM adoption is mature, where potentially every employee listed in the corporate LDAP might share this same requirement, still it is not a good idea to store the inbox or process application task lists in the LDAP. Even though they all share the same requirements, the tasks themselves vary from person to person, and will most likely require updating several times per day. This requirement violates the LDAP concept.

Therefore, we consider LDAP to be a source of user authentication, and only occasionally a source of more generic user data.

LDAP groups and access control lists

Many organizations also have grouping information stored in the LDAP repository. These groups can be along geographic, organizational, or business functional lines. Perhaps the groups reflect other business-meaningful parameters, such as employees who joined as a result of an acquisition.

In each of these cases, the presupposition is that the individual employees are not likely to change their group affiliations frequently. Therefore, this information is appropriate for an employee system of record, such as the LDAP repository.

Furthermore, some LDAP repositories support the concept of an access control list (ACL). However, the use of LDAP groups and ACLs is also relatively constrained by the nature of the LDAP read-only access. Specifically, IBM BPM is a tool that is built on and promotes the idea of rapid iterative development. The tool empowers business subject matter experts (SMEs) with the ability to define groups of individuals who are eligible to perform some business process task.

We strongly encourage that these groups be defined to be as small as possible, so that only relevant tasks are delivered to any given employee. Furthermore, as is the case with all iterative development methodologies, the structure and content of these small groups could change with any given new iteration, as the business process is either refined or improved.

To ask a centralized LDAP administrator to change their LDAP group definition to match the latest iteration of a process application would quickly get onerous for the LDAP administrator. Remember that these corporate LDAPs are supporting dozens, if not hundreds, of other applications.

> **LDAP for authentication:** Consider LDAP to be a source of user authentication, and not of authorization. The way that IBM BPM handles authorization is described in 4.4, “Authorization” on page 138.

Federated repositories

IBM BPM's preferred user registry is called Federated Repositories to point out the fact that it is a collection of independent repositories. It is built upon a sophisticated software layer, the VMM. The VMM enables tremendous flexibility. Multiple repositories, of various types, can be joined together in the federation, and they act as one, as shown in Figure 4-9.

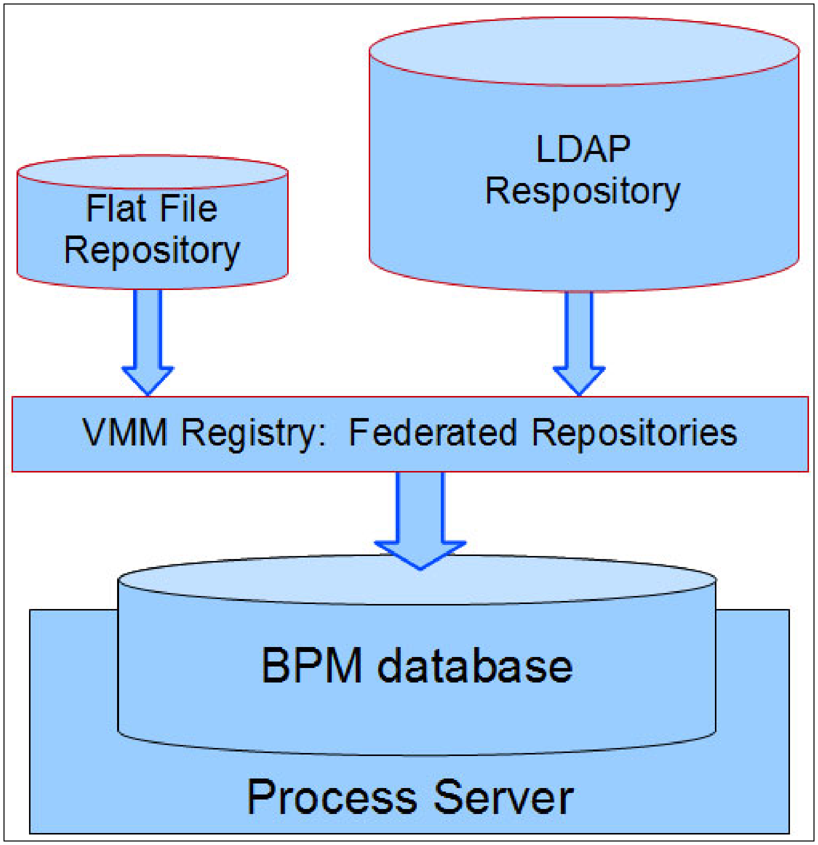


Figure 4-9 A more common federated user registry, including an LDAP

Figure 4-9 depicts a very simple case of one federated registry consisting of one flat-file repository and one LDAP repository. As mentioned before, WebSphere Application Server is compatible with various LDAP vendor's products.

Each LDAP vendor might define a different reserved word to access a particular function, but the IBM BPM VMM can be configured to abstract these differences and therefore work with any LDAP server that is LDAP V3-compliant. WebSphere Application Server also includes several LDAP templates to facilitate installation.

It is equally common for clients to have more than one LDAP server. This is often the case due to acquisitions or geographical dispersion. The acquired companies might well have a host of software systems that rely upon their existing LDAP structures, and there is always a period of integration where old systems are updated or replaced. WebSphere Application Server enables this by federating LDAP repositories together.

Figure 4-10 shows an example where the VMM federated registry is composed of a single flat-file repository, one generic LDAP V3 server, plus a series of Microsoft Active Domain repositories arranged in a tree structure.

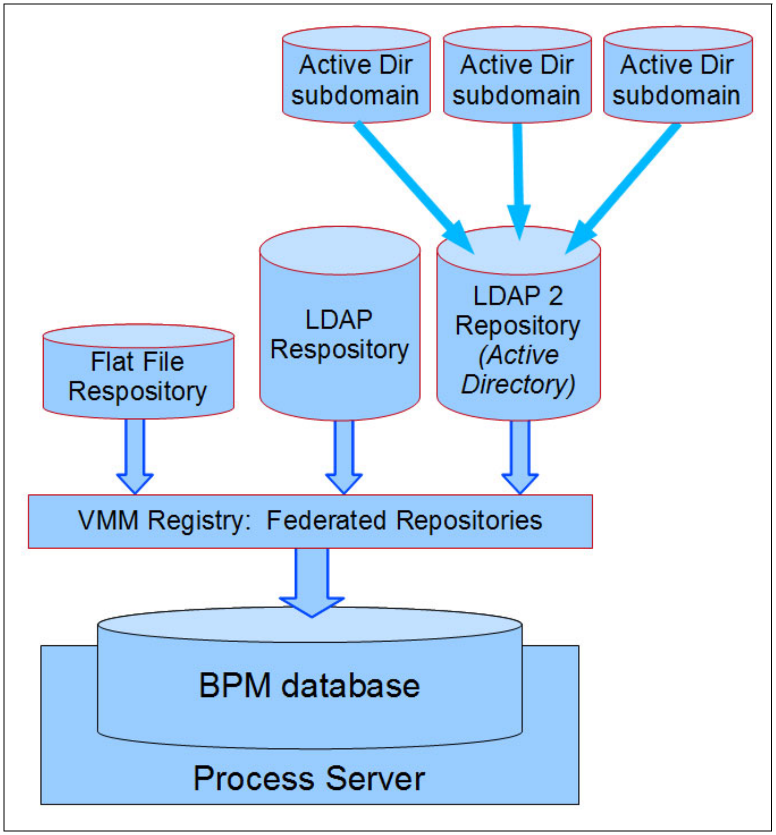


Figure 4-10 A complex user registry, including several LDAPs

Even more complicated arrangements are possible, but a complete treatment of this topic is not relevant to the purpose of this chapter.

For more information, refer to the IBM Redbooks publication *Understanding LDAP - Design and Implementation*, SG24-4986.

Custom software repository

Although the overwhelming majority of clients use an LDAP repository, situations can exist where a corporation's user and group data is in other repositories or custom user registries, such as a database. In such cases, moving this information to either a local operating system registry or a dedicated LDAP registry implementation might not be feasible.

For these situations, WebSphere Application Server security does provide a service provider interface (SPI) that you can implement to interact with your current registry. This custom registry feature can be developed to support virtually any user registry that is not implemented by WebSphere Application Server. However, note that developing a custom software registry is decidedly non-trivial, and developing one that is truly secure is far more complex.

There are several reasons for this, but chief among them is that the WebSphere Application Server security mechanisms are initialized and enabled well before other WebSphere Application Server services, such as data sources and enterprise beans. There is, therefore, a great deal of software that you need to write on your own, effectively "reinventing the wheel".

In addition, consider the following requirements:

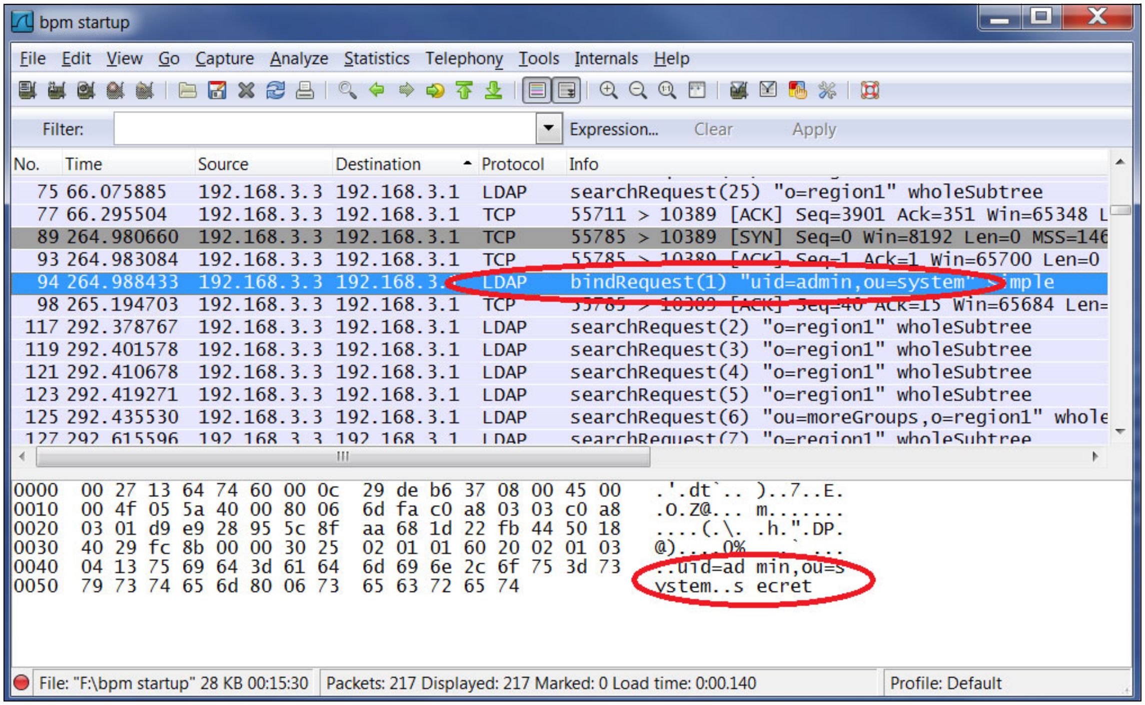
- The SPI must be completely implemented, including error scenarios.
- You need to consider and implement your own mechanisms for availability and failover.
- Portions of IBM BPM require additional security extensions that are above and beyond the WebSphere custom user registry SPI.
- Any custom registry should undergo extensive and rigorous third-party penetration and security vulnerability testing.

The need to secure LDAP

Most people think of LDAP as a server, but remember that LDAP is a *protocol*. Yes, there are LDAP servers, but because it is a protocol, in effect, this is a conversation between two servers.

As another specific example of how an overabundant faith in firewalls can become a security hole, we examine the LDAP connection.

Figure 4-11 depicts the actual network communications as captured by the freeware network protocol analyzer, WireShark, during the normal boot-up process of the IBM BPM Process Center.


bpm startup
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression...
Clear
Apply

```
No.  Time        Source       Destination   Protocol  Info
 75 66.075885   192.168.3.3 192.168.3.1   LDAP      searchRequest(25) "o=region1" wholeSubtree
 77 66.295504   192.168.3.3 192.168.3.1   TCP       55711 > 10389 [ACK] Seq=3901 Ack=351 Win=65348 L
 89 264.980660  192.168.3.3 192.168.3.1   TCP       55785 > 10389 [SYN] Seq=0 Win=8192 Len=0 MSS=146
 93 264.983084  192.168.3.3 192.168.3.1   TCP       55785 > 10389 [ACK] Seq=1 Ack=1 Win=65700 Len=0
 94 264.988433  192.168.3.3 192.168.3.1   LDAP      bindRequest(1) "uid=admin,ou=system" simple
 98 265.194703  192.168.3.3 192.168.3.1   TCP       55785 > 10389 [ACK] Seq=40 Ack=15 Win=65684 Len=
117 292.378767  192.168.3.3 192.168.3.1   LDAP      searchRequest(2) "o=region1" wholeSubtree
119 292.401578  192.168.3.3 192.168.3.1   LDAP      searchRequest(3) "o=region1" wholeSubtree
121 292.410678  192.168.3.3 192.168.3.1   LDAP      searchRequest(4) "o=region1" wholeSubtree
123 292.419271  192.168.3.3 192.168.3.1   LDAP      searchRequest(5) "o=region1" wholeSubtree
125 292.435530  192.168.3.3 192.168.3.1   LDAP      searchRequest(6) "ou=moreGroups,o=region1" whole
127 292.615596  192.168.3.3 192.168.3.1   LDAP      searchRequest(7) "o=region1" wholeSubtree

0000   00 27 13 64 74 60 00 0c  29 de b6 37 08 00 45 00   .'.dt`.. )..7..E.
0010   00 4f 05 5a 40 00 80 06  6d fa c0 a8 03 03 c0 a8   .O.Z@... m.......
0020   03 01 d9 e9 28 95 5c 8f  aa 68 1d 22 fb 44 50 18   ....(.\. .h.".DP.
0030   40 29 fc 8b 00 00 30 25  02 01 01 60 20 02 01 03   @)....0% ...` ...
0040   04 13 75 69 64 3d 61 64  6d 69 6e 2c 6f 75 3d 73   ..uid=ad min,ou=s
0050   79 73 74 65 6d 80 06 73  65 63 72 65 74            ystem..s ecret

File: "F:\bpm startup" 28 KB 00:15:30   Packets: 217 Displayed: 217 Marked: 0 Load time: 0:00.140   Profile: Default
```

Figure 4-11 Using a network analyzer to sniff admin passwords

If you have never used a network packet analyzer before, all you need to know for the purposes of this example is that each line is a summary of the Transmission Control Protocol/Internet Protocol (TCP/IP) traffic. You can watch this list in real time. The lower portion of the panel here is showing the contents of the selected TCP/IP packet.

This highlighted summary clearly shows that this packet is an LDAP `bindRequest` on the account `uid=admin`, `ou=system`. The packet is a request to bind the IBM BPM servers to the LDAP database to make a query. If you know the account name and password used to bind, you can browse the LDAP for any information that it contains. You can see, in the packet's content area, that the password for this account is `secret`.

Yes, it really is that easy.

The bind account name and password are exchanged at each step of the IBM BPM to LDAP conversation. The WebSphere Deployment Manager and Node Agents, the IBM BPM application servers (including `/ProcessAdmin`, `/ProcessCenter`, and `/portal`), and the IBM Process Designer, all communicate with the LDAP server and issue this same bindRequest.

Unless you secure your LDAP server using encryption (SSL), you are leaving your corporate LDAP server open to browsing every time an IBM BPM user logs in to their `/portal` Inbox.

We advise that you:

- Enforce encryption using SSL over the communications channel between the IBM BPM servers and your LDAP servers.
- Be sure to disable non-SSL traffic.
- Create a specific SSL truststore and alias for the LDAP.

4.3.4 Single sign-on

SSO is the ability to share credentials across systems. There are many SSO solutions that can be purchased and integrated into IBM BPM:

- IBM Security Access Manager
- CA SiteMinder
- Microsoft Windows Integrated Authentication
- UNIX and Linux Kerberos-based authentication protocols
- SAML (IdProvider Initiated use case)
- WebSphere custom Trust Association Interceptors

Many SSO technologies rely upon cookies or HTTP headers to carry the user's credentials with each HTTP request. Often, these credentials are encrypted. Unfortunately, the fact that these credentials are encrypted does not matter. An encrypted header can still be sniffed, copied, and injected into an attacker's browser HTTP requests.

The fact that the human attacker cannot read the contents of the encrypted header in no way diminishes the opportunity for attack. She can just paste the contents of this encrypted header into her browser, thereby impersonating the original SSO credentials.

What must occur for any SSO solution to be considered secure is that the destination server must take the extra steps to ensure that the SSO headers originated from a known good, trusted server. Simple strategies like inspecting the IP address of the HTTP request are not sufficient. IP addresses (as in this case) can be spoofed.

The correct, proper, and secure implementation of an SSO solution employs a mechanism for ensuring that the origination of the SSO credentials is valid, thereby eliminating the type of injection attack described previously. This is especially true if you are writing a custom trust association interceptor (TAI) to integrate an in-house developed custom authentication system into the WebSphere Application Server security mechanisms.

The TAI API is shown in Example 4-1.

Example 4-1 Trust Association Interceptor API

```
public class Simple_TAI implements TrustAssociationInterceptor {
public String getType() {}
public String getVersion() {}
public int initialize(Properties props)
throws WebTrustAssociationFailedException {}
public void cleanup() {}
public booleanisTargetInterceptor(HttpServletRequest request)
throws WebTrustAssociationException {}
public TAIResult negotiateValidateandEstablishTrust(
HttpServletRequest request,
HttpServletResponse response)
throws WebTrustAssociationFailedException {}
}
```

As you can see, this is deceptively simple. Six methods, two of which are simple `get()` functions that return (typically) hard-coded strings. Two more are just `initialize()` and `cleanup()` functions. This leaves us with only two methods to do the heavy lifting.

Software developers must overcome many challenges when designing an SSO solution:

- Time constraints
- Other project pressures
- Gaining access to the custom authentication system
- Getting firewall ports opened for their use
- Updating ACLs
- Software development and debugging required to enable the TAI to work

It is very common for us to see software developers celebrate that final change, which ultimately allowed their SSO solution to function, by simply moving on to the next overdue project task. However, it is not enough that your SSO solution works. It must work securely. This is another common security hole that we see: An in-house developed TAI that fails to take the additional step of ensuring that the origination source of the SSO header comes from a trusted source.

Techniques for establishing trust with the origination system include (but are not limited to) the following processes:

- Validating the SSL certificate’s serial number
- Validating the SHA1 digest’s fingerprint

SSO solution: We advise that you bring in an outside security professional to review your SSO solution to ensure that it meets all leading security practices before you put any such code into use.

4.4 Authorization

Authorization is the process of ensuring that a user (or other computer system) has permission to perform a given act.

In general, authorization can be enforced in several ways:

- ACLs
- LDAP groups
- Role-based access control (RBAC), such as LDAP groups in Java EE authorization
- Attribute-based access control (ABAC)

4.4.1 Group-based authorization

By far the most common conceptualization of user authentication has to do with group membership. Managers are authorized to perform some tasks, sales staff are authorized to view certain reports, purchasing agents are allowed to issue purchase orders, and so on. It is easy to conceive of a corporate LDAP with groups for managers, sales, and purchasing, and it is trivial for the business process designers to use groups such as these.

However, in a typical organization, there can be tens of thousands, even hundreds of thousands, of users, and thousands of groups. These numbers can grow quite large, and so it is common to see an LDAP vendor restrict the number of users or groups returned by generic queries.

For example, Microsoft Active Directory returns only the first 4,500 users or groups, truncating the rest from the results. This can obviously cause havoc for a business process application that simply uses LDAP groups for its authorization.

In addition, because LDAP products are optimized for read-only and static data, it is very common that the structure and content of the LDAP repository be under the strict control of an LDAP administration team. Furthermore, these groups might have been defined for existing applications. It is not uncommon for these existing applications to no longer be in use, and yet their LDAP groups remain. This can be contrary to the core value proposition of IBM BPM.

One of the most significant benefits of IBM BPM is its ability to model business processes to gain some perspective and, hopefully, to use this to effect business process improvement. The entire approach to IBM BPM, from process documentation, to process application development, to process improvement, is based on agile principles.

It is quite common for a team of process designers to decide that a process can be improved if the roles are reexamined, possibly creating new roles and combining others. To rely on the LDAP administration team to make these changes to the corporate LDAP is almost always untenable from both perspectives:

- The process designers want immediate turnaround.
- The LDAP administrators need to ensure that changes will not break other enterprise applications that rely on the LDAP structure.

To address this concern, IBM BPM goes much further than simply referring to LDAP groups. IBM BPM provides a product component called the `/ProcessAdmin`, and this component has a section called *Group Management*. Within this section, business process designers can create groups that go beyond what is presented by the corporate LDAP.

Consider the case where an organization might have grown as the result of acquiring a competing company in a different geographical region. Each region would have almost certainly had different LDAP structures.

Perhaps region 1 called their IT department *ldapIT*, and region 2 called theirs *adBPMDevelopers*. IBM BPM enables a unified view of these two LDAP groups under a single group called, for example, *devAuthors*.

These nested groups are depicted in Figure 4-12.

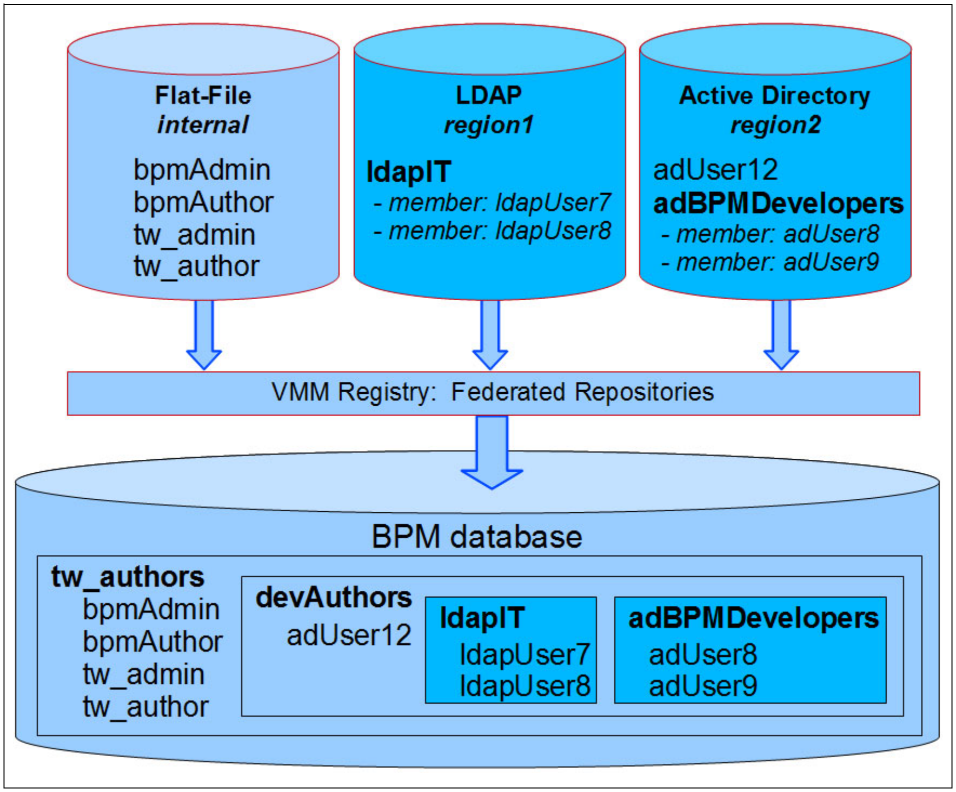


Figure 4-12 Nested LDAP Groups

This is a very powerful feature. LDAP browsers are not typically capable of merging two LDAP sources to provide a unified view. In Figure 4-12, you can see that the two LDAP groups are combined in *devAuthors*. Furthermore, another user has been added to *devAuthors*. When the business process executes, any user who appears in *devAuthors* is eligible to perform this business task.

In addition to combining two groups into one, IBM BPM also enables the inclusion of groups within groups (nested groups). Therefore, the entire management of the groups is handled by the business process designers and SMEs. These professionals are not bound to the LDAP administrator organization, because all of this grouping information is in IBM BPM.

Another important point that is worth considering about your business analysts (BAs) taking control of the BPM groups is that the corporate LDAP groups might have very little, or absolutely nothing, to do with the business processes that you are currently modeling.

In the diagram in Figure 4-12 on page 140, the simple fact that ldapUsers 7 and 8, and adUsers 8 and 9 are all considered *developers* tells you nothing about *which applications* they should be granted access to. Perhaps ldapUser7 is working on a highly confidential application that only he should have access to.

Going to the LDAP administrator organization to request a new group just for ldapUser7 is unwieldy at best. The `/ProcssAdmin` component gives your business process designers and SMEs control of your business group authorizations.

> **Proper group management:** We advise that you ensure that your groups are tightly coupled to the business process functions. Use `/ProcessAdmin` to define groups, and use LDAP groups rarely.

4.4.2 Dynamic authorization: Teams

In the previous section, we described the capability in `/ProcessAdmin` to define groups independently of the corporate LDAP. In earlier versions of IBM BPM, these were called *Participant Groups*. In IBM BPM V8.5 and later, the set of users who can start or end processes, work on tasks, complete tasks, or administer processes and tasks, are now called *Teams*.

Although Participant Groups have been deprecated with IBM BPM V8.5, old definitions are 100% compatible with, and automatically migrated to, BPM V8.5. Teams introduce three new major features:

- Teams can now have *Managers* (identified and modeled within IBM Process Designer).
- Teams can now be specified *dynamically* (with the members being determined by a web service call).
- Teams can now be *refined* dynamically (with non-applicable team members now being filtered away).

You define teams in IBM Process Designer as a part of a process application (or even a toolkit). You can specify that teams be one of two types:

- *Static*. All users are defined using either LDAP groups or `/ProcessAdmin` groups in the same way that Participant Groups had been defined.
- *Dynamic*. All users are determined at run time using a Team Retrieval Service.

Team Retrieval Services can be built using the IBM BPM JavaScript API, or they can be built as standard Integration Services. The JavaScript API provides functionality identical to that found in the `/ProcessAdmin` group management component.

When built as standard Integration Services, they include all of the integration service capabilities. In either case, there is a mandatory input parameter (`Name`) and one output parameter of type `Team`. The Team object can contain both individual user IDs and group names. Whenever a process application uses the Team Retrieval Services, IBM BPM starts the associated service and updates the team membership (and its Manager) with the web service result.

4.4.3 Task-based authorization

Authorization can also be defined for any given application along a continuum of granularity. For example, the following list of theoretical authorizations goes from coarse-grained to fine-grained:

- Everyone is authorized to view this web page.
- A user must be logged in to view this web page.
- A logged in user must also exist in an ACL in order to update this web page.
- A logged in user must be a member of a specific LDAP group in order to have certain sections of this web page made visible.
- A logged in user must be known to the underlying application that generates this web page. In addition, before any data is displayed or any button is drawn on the web page, the application programmatically checks `isUserInRole()` to determine if this user has been granted the appropriate role associations.
- A logged in IBM BPM user can only execute this task if they a) are included in the list of Team members as determined by this specific web service, b) were not the last user to touch this task, c) belong to a level of management that has authority to perform this task, and d) the task has not been unattended for more than four hours.

As you can see, IBM BPM defines a very fine-grained authorization model, and does so in product-specific terms (for example, you are not authorized to run this task if you were also the last user to update this task). This is a strong argument for not simply considering the groups defined in your corporate LDAP as a source of user authorization.

Authorization is being mentioned in this book simply because it is commonly overlooked. Many organizations believe that because they have groups defined within their corporate LDAP, that’s all that really needs to be considered. A description of IBM BPM, together with an agile business process improvement methodology, is outside of the scope of this chapter. Notwithstanding, we encourage you to investigate this topic further.

For a more complete description of these topics, see *IBM Business Process Manager Security: Concepts and Guidance*, SG24-8027.

4.5 Conclusion

In this chapter, we described the importance of security to, and the hardening of, your IBM BPM installation.

We described how IBM BPM and WebSphere Application Server concepts map to each other, and identified specific hardening steps, including the use of SSL, DMZs, and database security.

We described user authentication, how LDAP repositories can be federated together, and how the IBM BPM `/ProcessAdmin` component can be used to provide a true organization-wide view of your employee repositories.

We described user and task authorization, how IBM BPM enables fine-grained groups that identify the set of individuals who should be allowed to execute specific tasks within a business process, and how these teams can be defined dynamically at run time using web services.

5

Design considerations and patterns

In this chapter, we describe the following topics and take a practical look into applying certain design considerations and patterns:

- Product installation considerations
- Business process design
- Service design
- Data flow patterns
- Toolkit design
- Error handling
- Logging
- Asset maintenance and governance considerations

5.1 Product installation considerations

In this section, we describe important points to consider as you install IBM Business Process Manager (IBM BPM):

- Define your IBM BPM topology in advance by completing the product sizing exercise using production load estimates. As future projects are added, we suggest that you reevaluate the sizing estimates as part of the capacity planning exercise.
- Develop a product installation and configuration checklist for each IBM BPM environment before carrying out the installation activity. This checklist can include information, such as server names, deployment environment names, profile names, cluster names, cluster member names, database names, port numbers, external security registry information, installation username on the servers, and so on.

  The checklist can also include the post-installation verification steps. The advantage of having such a checklist is that all of the information that is required to install and configure the product is maintained in one source. This makes it easier to review this information and make changes as appropriate.
- Automate the product installation and configuration steps by using response files and `wsadmin` scripts. Automation reduces the likelihood of errors. For more information about using response files, see the IBM BPM V8.5.5 Knowledge Center:

  `http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.imuc.ebpm.doc/topics/inst_sil_resp_aix.html?cp=SSFPJS_8.5.5&lang=en`
- Start the product installation and configuration process by building a small topology. This gives you the opportunity to address any glitches in the installation and configuration scripts before using them for additional product installations.
- Verify your IBM BPM installation, and back up the installation data and the topology configuration data before deploying your applications in the IBM BPM environment. For instructions about verifying your IBM BPM environment, see the IBM BPM V8.5.5 Knowledge Center:

  `http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.imuc.doc/topics/tins_verifycmd_de_environment.html?lang=en`

5.2 Business process design

In this section, we describe different types of business processes, such as *structured* and *unstructured* ones, and give you an overview of patterns and anti-patterns for process design.

With any business process, we want to consider the following points to determine whether what we have is a viable business process:

- Performing the process provides *value to the business.*
- The process contains individual *business-relevant steps.*
- *Business-relevant data* flows through the process.
- The process follows a *relatively structured path.*
- The steps within the process are performed by *multiple roles or teams.*
- The process *changes* over time as a result of changes in the business.

If the previously mentioned factors are not given, you might want to rethink your decision whether to implement your requirements as a business process, or use a different approach.

5.2.1 Process flow patterns

Processes can be of structured or unstructured nature. Structured processes are flows that follow a common path with some defined exceptions or alternative paths. Unstructured processes are more state-driven processes. You don't define a flow, rather than action-reaction patterns.

The key to structured processes is that they have the following characteristics:

- Measurable
- Comparable
- Repeatable

Unstructured, or case-style processes, alternatively, work more like a state machine.

We address more details about the process types in the following sections:

- Structured processes
- Generic process model
- Unstructured processes

Structured processes

Structured processes can be divided into two categories of *human-centric* and *system-centric* processes:

- Human-centric processes

  Human-centric processes are processes that focus on the coordination of work between human participants. This includes notifying about work items, measuring work, providing information to execute and complete work, and so on. That means the human interacts with the process throughout its complete lifetime.

  In IBM BPM, human-centric processes are implemented in Business Process Modeling Notation (BPMN) language. For more information about BPMN, see the following website:

  `http://www.bpmn.org/`

- System-centric processes

  A system-centric process usually has no, or very limited, human interaction. If it doesn't have *any* human participant, it can be classified as a service. Often however, it has an individual initiating the process or using the outcome of the process. The human interacts only at the beginning or the end of the process.

  In IBM BPM, system-centric processes are usually implemented in the Business Process Execution Language (BPEL). One advantage of BPEL is that it supports advanced integrations, which require transactionality or non-HTTP bindings.

  If the integration logic is more complex (for example, in terms of lines of code), IBM BPM Advanced can be faster, because it provides pre-complied code. Alternatively, IBM BPM Standard operates on JavaScript, which has to be interpreted at run time.

  More information about system-centric process can be found in 5.4.4, "Integration design" on page 183.

In this section, we focus mainly on human-centric processes, and describe the following topics:

- Process automation
- Process flow, page flow, and service orchestration
- Levels of task automation (progressive task optimization)
- Patterns, anti-patterns, and modeling risks
- Divide and conquer business processes

Process automation

Besides the process types, there are different levels of automation for each process, as shown in Figure 5-1.

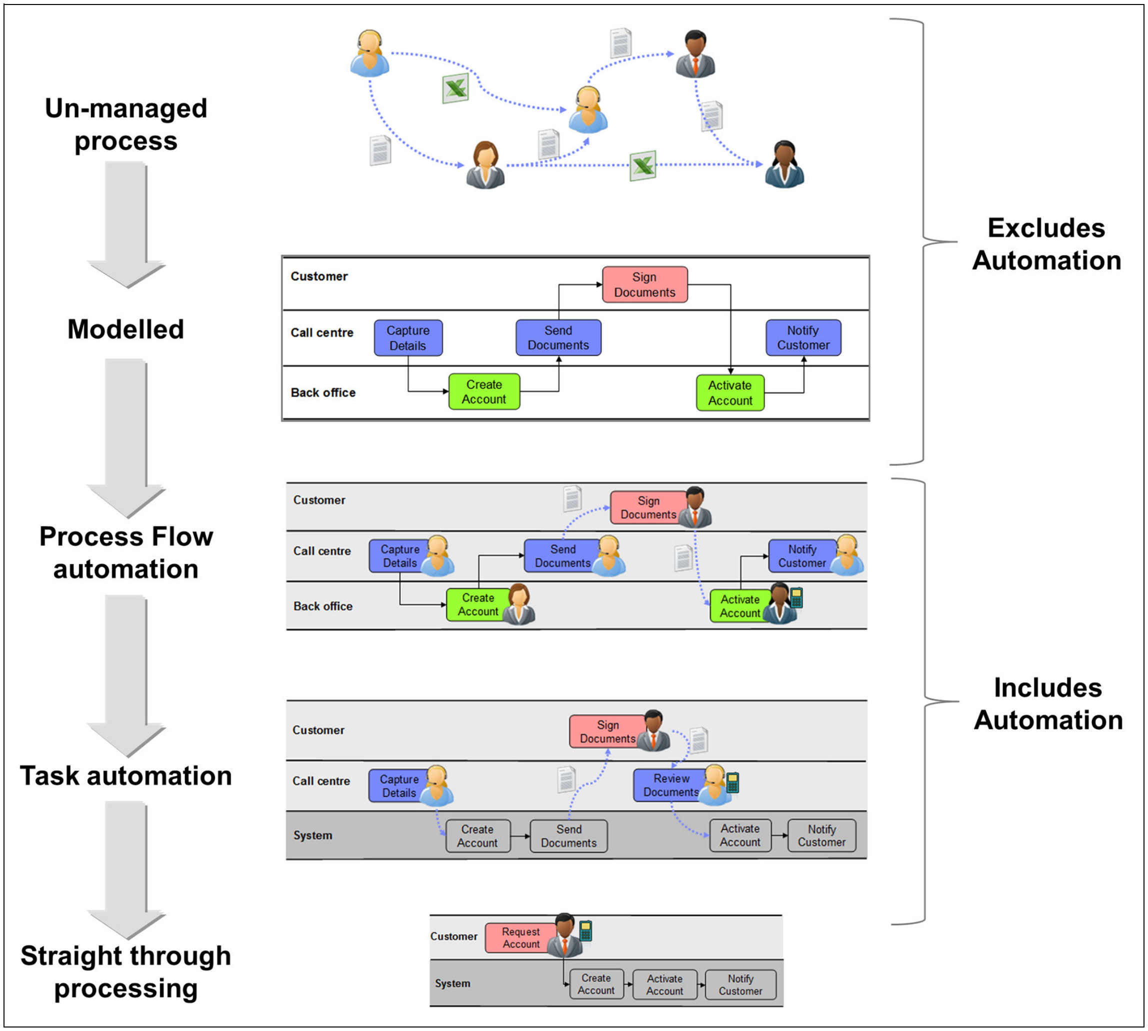


Figure 5-1 Automation hierarchy

The following section describes the different levels of automation:

- *Unmanaged processes* are often the ones that are either not discovered or that have not been formalized yet. Formalization means that the processes have been described either through words or diagrams to define an order of activities and tools.
- *Modeled processes* are the ones that have been identified and formalized. Many different tools can be used for this. One example of a process model tool is IBM Blueworks Live. To learn more about this tool, see the IBM Redbooks publication *Process Discovery Best Practices Using IBM Blueworks Live*, REDP-5111.
- *Process Flow automation* means that the process is running in an IBM BPM runtime environment, and that every task is executed by a human being. Many activities or integrations with other systems are achieved through a *swivel chair* approach, which means that a user switches back and forth between the process application and other systems.
- *Task automation* is the next level of automation after flow automation. Besides the previously mentioned capabilities, the process can execute tasks independently of humans, in an automated way. Tasks include such things as notifications, integrations (to gather or distribute information), and so on. You can find more information about task automation in “Levels of task automation (progressive task optimization)” on page 152.
- *Straight through processing* (STP) describes the full automation of a process. No human interaction is necessary in this stage. A process like this is usually executed in a matter of seconds.

Process flow, page flow, and service orchestration

It is a common mistake to create a process model that contains a mix of multiple levels of process design (for example, as defined by the American Productivity and Quality Center[1] (APQC)) within one single process model. It is very important that you know the answers to the following questions:

- Where to include service orchestration?
- Where to include page navigation?

The answer to both is *not in the process model*, although this is often done incorrectly. Next, we look into why this is the case.

The easiest way to describe this is by looking at the process-level definitions of the APQC or its Process Classification Framework (PCF) at the following link:

http://www.apqc.org/process-classification-framework

[1] More information about the American Productivity and Quality Center can be found at: http://www.apqc.org/

Additional information about this topic can also be found in the IBM Redbooks publication *Process Discovery Best Practices Using IBM Blueworks Live*, REDP-5111.

Figure 5-2 shows how Level 4 processes belong in the business process layer, where Level 5 processes are part of the consumer layer.

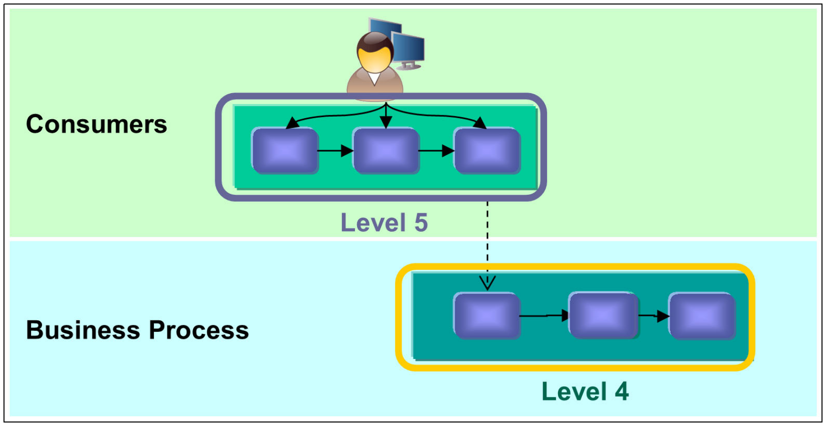


Figure 5-2 Level4 - Level5 Process Diagram

The following reasons explain this distribution:

- *Level 4* includes the definitions of an activity.
- *Level 5* specifies the Task (one activity can have multiple tasks).

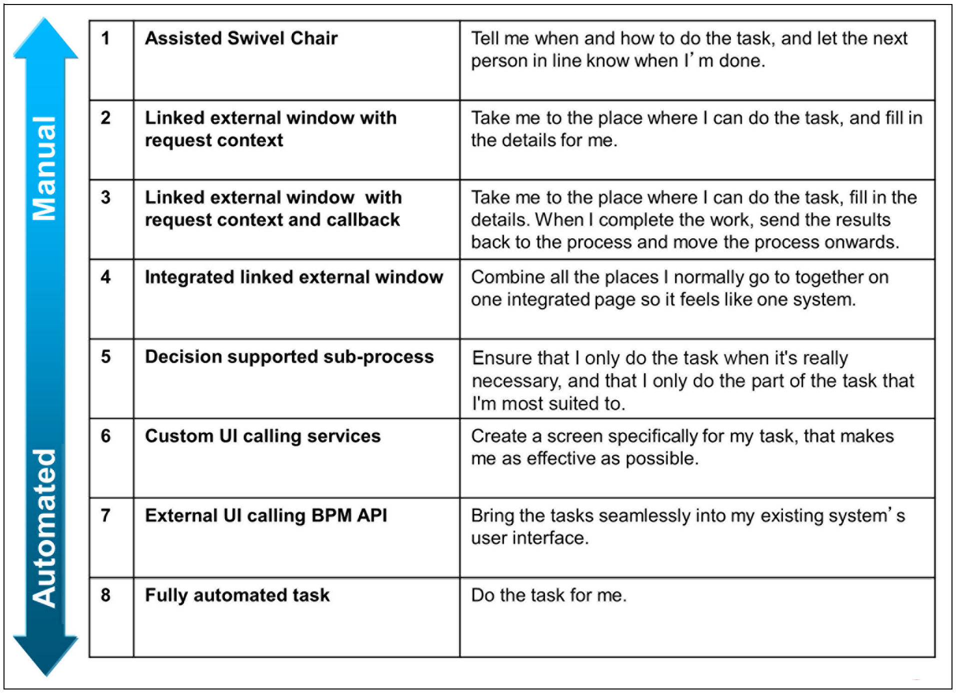
In a process model, we always stay above Level 5. Page flows (or page navigation) and service orchestration are part of the definition of the task (not the activity), and are therefore modeled differently.

Rule: If you have a sequence of more than one activity in a row in one swimlane, chances are that you mixed up different levels of process design.

Multiple steps in the same swimlane of a process diagram imply that they could have been performed by the same entity (system or person) in one activity.

Levels of task automation (progressive task optimization)

Figure 5-3 shows different levels of task automations. From manual (swivel chair) tasks to fully automated ones.


Manual
Automated

1	**Assisted Swivel Chair**	Tell me when and how to do the task, and let the next person in line know when I’ m done.
2	**Linked external window with request context**	Take me to the place where I can do the task, and fill in the details for me.
3	**Linked external window with request context and callback**	Take me to the place where I can do the task, fill in the details. When I complete the work, send the results back to the process and move the process onwards.
4	**Integrated linked external window**	Combine all the places I normally go to together on one integrated page so it feels like one system.
5	**Decision supported sub-process**	Ensure that I only do the task when it's really necessary, and that I only do the part of the task that I'm most suited to.
6	**Custom UI calling services**	Create a screen specifically for my task, that makes me as effective as possible.
7	**External UI calling BPM API**	Bring the tasks seamlessly into my existing system’ s user interface.
8	**Fully automated task**	Do the task for me.

Figure 5-3 Levels of task automation

Next, we examine each level:

1. *Assisted Swivel Chair* means that only the process is controlled by the system. You get a task telling you what you must do. However, you perform the work in a different system. The process model is implemented in the simplest form in a business process management system (BPMS):

   – Work is prioritized and transferred between the participants automatically.
   – Workload distribution is managed automatically.
   – Managers can focus on assistance rather than distribution of work.

   For example, you get an IBM BPM task, telling you to update the client address in the SAP system. The changed address is part of the Coach, but you have to manually open SAP and execute the change address operation.

Why does it make sense?

Using swivel chair integrations or automations, you can track, compare, measure, prioritize, and complete many more tasks using IBM BPM features. Only the integration has to be done manually. Because integrations are usually drivers of implementation effort and time, it helps you to deliver value faster.

2. *Linked external window with request context* is similar to Assisted Swivel Chair, with the addition of a link to the external application where the action has to be performed. The link already contains case-related information, so the fields in the external application are pre-completed as much as possible.

   For example, the task contains a link into the SAP system that opens the page where you need to update the client's information. All address information has been completed automatically, but the user has to log in manually and might have to complete some additional information.

   Why does it make sense?

   Similar to 1) with the difference that the user does not have to know where to perform the action.

3. *Linked external window with request context and callback* is similar to 2), with the addition that the external system can notify IBM BPM about the results.

   For example, the IBM BPM task contains a link to the SAP system to update the client's address. Upon selecting that link, SAP opens, the user has to log in, and the update client information window displays with all of the address information completed automatically. The user simply has to select **Submit**, and SAP notifies IBM BPM through callback if the update was successful. The IBM BPM Coach now contains a status message `update successful`.

   Why does this make sense?

   This is a simple and fast way to create an integration without causing too much duplication of work through swivel chair activities.

4. *Integrated linked external window* is similar to 3) with the difference that the external application does not open in a separate window, but is directly integrated into the IBM BPM application. From a user perspective, it looks like one single application.

   For example, the IBM BPM task includes a user interface (UI) to change the client information. All the user has to do is to submit it in the Coach. The SAP update client information UI is opened there.

   Why does this make sense?

   From a user experience point of view, this feels fully integrated.

5. *Decision-supported subprocess* (task) automation looks at subprocesses or task flows (Level 5) and improves the behavior about who does what and when, and whether it needs human interaction at all. Arguably, this can be considered process flow optimization. However, in this context, the process can be a task.

   For example, if the opportunity value is > USD 1,000,000 the task must be executed by a senior person. If the opportunity value is < USD 1,000,000 and > USD 1000 a junior person can do it. Otherwise (< USD 1000), the opportunity gets automatically approved.

   Why does this make sense?

   Decision-supported subprocess automation can ensure that only tasks that really require human interaction are created for someone's inbox.

6. *Custom UI calling services* means that a window has been created specifically to execute this task. This is unlike the previous points, where an external UI was integrated, or only links or instructions were included.

   For example, the IBM BPM task includes a UI to update the client information.

   Why does this make sense?

   The user does not have to go to various external systems anymore, but can use one UI to execute all necessary actions.

7. *External UI calling IBM BPM API* means that the user does not execute the task in IBM BPM anymore, but from another system in order to seamlessly integrate different systems with each other. The external system then calls IBM BPM using application programming interfaces (APIs) to work with the task (for example, task re-assignments, task completions, and so on).

   For example, the SAP system gets the notification that client information must be updated. The user updates the information directly in SAP. SAP calls IBM BPM to inform the system of the task completion.

   Why does this make sense?

   Users do not have to get familiar with another system. However, task execution, service level agreements (SLAs), key performance indicators (KPIs), and so on, can still be tracked in the IBM BPM application.

8. *Fully automated task* means that the system can execute the task without human interaction.

   For example, the IBM BPM system can automatically update the address information within SAP by calling an exposed web service.

   Why does this make sense?

   Fully automated tasks enable the users to focus on other high-value activities.

Patterns, anti-patterns, and modeling risks

Basic information about process patterns can be found on the following website:

`http://www.workflowpatterns.com`

These patterns are to be considered basic process flow patterns. As you read through the website you discover that in the product evaluation section for each pattern, there is a scoring for BPMN and BPEL whether it is supported in the particular functionality.

Before we go into the details of how to model a process, and which patterns are good and which are not, we have to ask ourselves questions to determine whether this is a process at all.

Determine if this is a process:

- If the flow has only one activity, this is *not* a process.
- If the flow has only one participant, this is *not* a process.
- If the flow is different every time (unstructured) or follows the Anywhere-To-Anywhere pattern, this is *not* a process.

If it is not a process, it is most likely a service that you want to manage using your service-oriented architecture (SOA) governance. In case the outcome is that this service should still be implemented within IBM BPM, the following implementation options must be considered:

- Expose as IBM BPM Standard service

  Services in the IBM BPM Standard Edition can be exposed in different ways. In this case, exposing it as a *Human Service* or *Startable Service* makes the most sense.

  Exposing as a *Human Service* means that it can be started and executed by an individual, either through the process portal or a link. The service is non-persistent. Therefore, if the user abandons it, information is lost, unless that information is manually saved to a system of record (SOR).

  Typical cases, where human services without process context are exposed, are simple maintenance services, such as create, retrieve, update, and delete operations to a database.

  Exposing as a *System Service* means that it can be called by other systems. The other systems can be other process instances or even external applications. System services can be exposed so that they can be called either through web services (SOAP over HTTP) or Representational State Transfer (RESTful) protocol services.

- Expose as IBM BPM Advanced Service

  Services in IBM BPM Advanced Edition can be implemented as either BPEL or mediation flows. For more information about BPEL and mediation flows, see 5.4.4, “Integration design” on page 183.

To further evaluate whether your requirements should be implemented as a business process, consider the following benefits:

- It provides *value* to the *business.*
- It follows a *structured* path.
- It is *repeatable* and *measurable.*
- It contains individually *business-relevant* steps.
- The steps are performed by *multiple roles* or *teams.*
- Business-relevant *data flows* through it.
- The flow *changes with* the change of the *business.*

When we have identified that there is an actual business process, it can be implemented using different patterns. The following sections include descriptions of patterns and anti-patterns:

- Rule of Seven

  The Rule of Seven is a non-binding guideline used in IBM BPM to provide a clear and concise communication tool.

  Every process can be decomposed into a beginning, a middle, and an end milestone, with logical transitions in between each of them. Therefore, a process should have no less than three milestones. A process should also be decomposed into no more than seven milestones to keep the communication clear and concise.

  Within each milestone, there should be between three and seven tasks or subprocesses. If more than seven obvious tasks are present, consider breaking down some activities into sub processes. As the process continues to be broken down, always keep the division between three and seven. By following this guideline, all activities at the same level should be similar in scope.

  This rule enables the creation of a model overview that is digestible as a whole, and explainable in less than 5 minutes.

  In summary, for processes and services, you want to have not more than seven activities in a diagram. If it is more, break it down into subprocesses.

  A description about the rule of seven can be found in the BPM community Wiki on the following website:

  `http://wiki.bpmwiki.com/display/commwiki/Rule+of+Seven`

Consider the following situations as they apply to the Rule of Seven:

  - When to use?

    The approach should always be used.

  - When not to use?

    The approach should always be used.

  For more information, see "Divide and conquer business processes" on page 163.

- Escalations

  Escalations are important to ensure that SLAs are held, or to react or report upon the breaking of them.

  As indicated previously, SLAs do not contain any functionality to ensure that they are being met. In IBM BPM, the purpose of SLA consequences is to either report and react upon their breaking *after* they were not met.

  Consider a task that has an SLA that defines that the execution time must not be longer than one day. However, nobody executes the task for two days, yet still nothing will happen. Only after the task has been completed is the SLA evaluated, and the defined consequences are triggered.

  To create escalations that serve the purpose of ensuring that an SLA is being held, we suggest using timer events. The most common use is to attach a timer and reroute the task to itself while changing the assignment logic to the next higher lever in the management chain.

  Another common pattern is to escalate to a separate task for a manager to make sure that whoever has the task currently assigned either executes it or reassigns the task to another member or group.

Figure 5-4 depicts the implementation of this in IBM BPM.

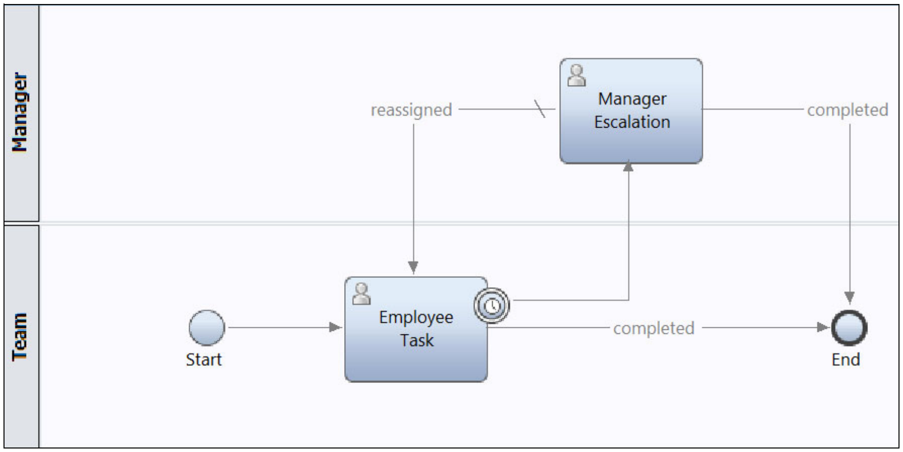


Figure 5-4 Timed Escalation

Consider the following situations as they apply to Escalations:

– When to use?

  Use this approach to enforce that a task is executed in time.

– When not to use?

  Do not use this approach if it is likely that the timer triggers while someone is working on the task. If this happens, the data entered is lost.

- String of Pearls

  A String of Pearls is being created if there are multiple activities in a sequence in the same swimlane, as depicted in Figure 5-5.

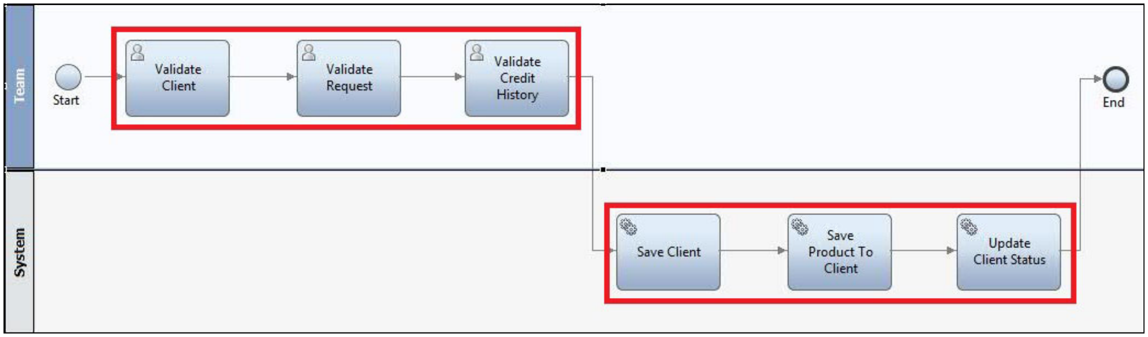


Figure 5-5 String of Pearls

Having a multi-instance loop (MIL) in the system lane is the same as having *n* number of tasks in a sequence. Particularly when talking about the system lane, the string of pearls anti-pattern can have a tremendous effect on the performance of the system. The effect even multiplies if the activities are MIL sub-BPDs or sequential or nested sub-business process definitions (BPDs) that are of system-centric nature.

Avoid: There must *never* be a multi-instance loop in the system lane.

Whether it is a System or a Team Lane, having multiple activities in a row suggests that the level of decomposition is wrong (see “Process automation” on page 149).

For human tasks, their is a disadvantage from a user experience perspective. Because it is modeled as separate activities (as opposed to screen flow or page flow), the user has to make extra clicks to claim and start each of the activities.

For the sequential system lane steps, the sequence affects the performance of the BPM event manager, because the creation of tasks uses more resources than the creation of subservices within a task.

If you need to have multiple activities performed by the system at the same time, put them one level below, into the service level. Consider calling the service logic iteratively (by implementing a loop) rather than in parallel (MIL in BPD).

Consider the following situations as they apply to String of Pearls:

– When to use?

  The approach should never be used.

– When not to use?

  The approach should never be used.

- Process lifetime considerations

Note: A process should be *processing*, and not be stagnant.

If a process spends a significant amount of its time paused on one step, and during this time there are many interactions with the process data, it is likely that we have reached a natural process end state.

In this case, the process should be ended, and the data should instead be stored in an operational system (existing if possible, new if necessary):

- Reasons to break up the process

  Elapsed time of months represents a significant process break. If most instances would be held up for a week or more, this also should be seen as a process break.

  If a process design appears to *expect* that tasks will be in queues for weeks before being addressed, this suggests that there is an issue with the process design, or with the resource distribution. During this time it is likely that the most common activities are to view, report on, and maybe even update the data, which implies that a more traditional operational system would be more appropriate.

- Reasons to retain the process

  If processes are only held up due to system outages, whether for brief maintenance windows, or awaiting overnight batch processing, but are otherwise continuous, this is not necessarily a process break.

  If processes are occasionally held up at human task steps for some days because workload is high and there are no workers available, this is part of what the human task management system is there to assist with. However, the aim of workflow management is to level out these peaks of activity, and enable redistribution of work.

  If tasks are constantly waiting for more than a few days to be worked, this represents a serious backlog. The task distribution or process flow needs to be addressed to reduce these delays.

- Monitoring the high-level status across process breaks

  If we break up a very long-running process into separate pieces, how can we monitor the *whole*?

  It is important to recognize that monitoring of the larger scale end-to-end process does not necessarily require a runtime implementation of the process to be present from end to end. Monitoring events can be emitted by the parts of the broken up process, and gathered into a monitoring model that represents a logical high-level process, even though there is no physical runtime process.

Consider the following situations as they apply to process lifetime:

- When to use?

  The approach should always be used.

- When not to use?

  The approach should always be used.

For more information about process lifetime considerations, see “Divide and conquer business processes” on page 163.

- Event-driven processes

  We use events in a process if we only need to know that it *will be* done, not that it *has been* done.

  Messages are placed in several queues within a single transaction:

  - This action is useful for updates to systems that are not expected to be instantaneously up to date, such as data warehouses and audit logs.
  - This action is useful for notification type requests, such as Short Message Service (SMS), email, and so on.

  Event-driven processes provide the following benefits:

  - They provide a fast answer to the caller.
  - With persistent queues, the messages should never be lost, and will be applied to their destinations eventually.

  Event-driven processes pose the following challenges:

  - Destination systems have not received the data when the response is given.
  - The time taken for destinations to receive the update is not in the control of the composition.
  - Should problems occur while processing the messages, the composition and therefore also the caller are unaware of them. Event-driven processes are reliant on external exception handling.

  Consider the following situations as they apply to event-driven processes:

  - When to use?

    Use this approach if you don’t need to know the result immediately.

  - When not to use?

    Do not use this approach if you need to act (for example, make a decision) based on the response of a call.

- Multiple system swimlanes

  A system swimlane in a business process defines what the IBM BPM system is doing. It *does not* define the systems that it is integrating with. Therefore, a process can always have a maximum of only one system swimlane.

  Note: Every BPD has a maximum of *one* system lane.

Next, we assume that a business process integrated with SAP and Salesforce to retrieve or to store information. In this case, each of the integration steps might be a separate activity, executed at different times in the process. However, each integration is executed by the business process management (BPM) system.

Consider the following situations as they apply to multiple system swimlanes:

- When to use?

  Multiple system swimlanes should never be used.

- When not to use?

  Multiple system swimlanes should never be used.

- Exposed service to initiate process

  In the previous sections, we described how you can expose services. Similarly, you can expose a process to be able to start it (for example, from the IBM BPM Portal). However, if you expose a process, start it, and then abandon it, you will have an orphaned process instance. This is something that happens quite frequently.

  For example, a user starts to complete the form for a credit card application. He starts the process, but gets interrupted by his coworker. He closes the browser. Later, he goes back in and starts another credit card application instance, because he has not completed the form before.

  This example is a scenario that applies mostly if external users are part of the application. However, even internal or trained staff might act in a similar way. To avoid abandoned process instances, you can take the first task of the process and externalize it into a stand-alone human service. The process gets instantiated only after all of the information has been entered and the task has been completed.

  Figure 5-6 depicts the exposing of the process on the left side, and the exposure of the starter human service on the right side.

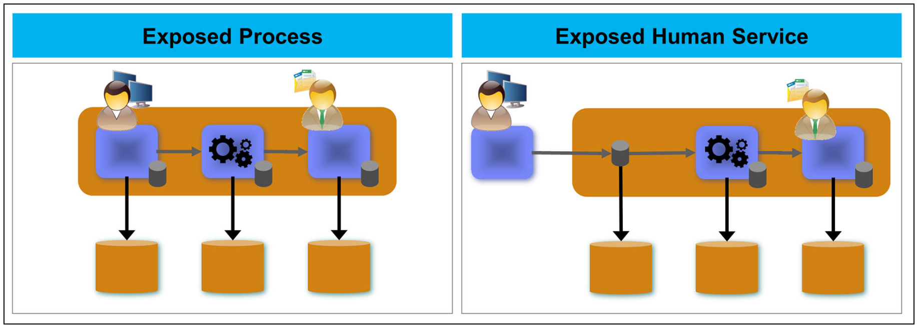


Figure 5-6 Types of process instantiation

Consider the following situations as they apply to exposed services initiating the processes:

- When to use?

  Use this approach when there is a chance that users who initiate the process will end their activity and simply close the browser (abandon the instance).

  Use this approach if the start of the process does not happen through the IBM BPM Process Portal, so that if a process was started and a task created, the user is not able to resume the task (for example, a form that is available on an Internet page).

- When not to use?

  Do not use this approach if the initial task takes a long time, and it is expected that the user can jump in and out of the human service as he completes the task.

Divide and conquer business processes

The “Rule of Seven” on page 156 describes when a business process has to be broken down due to its complexity. The “Process lifetime considerations” on page 159 describe when a process should be broken down due to its extensive run time. In this section, we want to take a close look into the *how* and *what* when it comes to breaking down a process:

- A *complex* process where we apply the Rule of Seven should be broken down into subprocesses or linked processes. In IBM BPM, it is often easier to use linked processes, because they are a separate artifact and can thereby be reused.

  The subprocess artifact belongs to its parent process, and only exists in the context of its parent process. It is mostly used if the subprocess cannot stand alone by itself or cannot be reused, but is there to manage the complexity of the super process.

  Processes that are broken down into linked processes can be put into a Toolkit. With that, they can be reused by multiple process applications. However, when this is the case, it important to know that the lifecycle of the linked process and its consumers might start to diverge. Integrating a Toolkit by adding it to a process application is effectively *tight coupling*, which means, if one changes, the other one has to be changed (redeployed) as well.

  To create a looser coupling, as required by the principles of an SOA, linked processes can also be exposed as a service, and thereby be called through web services or the REST API. To expose and deploy a process, it needs to be part of a process application. Toolkits cannot be deployed in a stand-alone manner.

Toolkit implications: A process application or its deployment artifact (`.twx` file) contain a copy of all the Toolkits that have been associated with it. This means, if one Toolkit is reused across many process applications, there are many copies of this Toolkit. This can result in a versioning problem.

In short, breaking down a process into more atomic pieces can be done through different levels of decoupling. In any case, at the end there remains a master process that controls the complete process instance lifecycle. This is depicted in Figure 5-7.

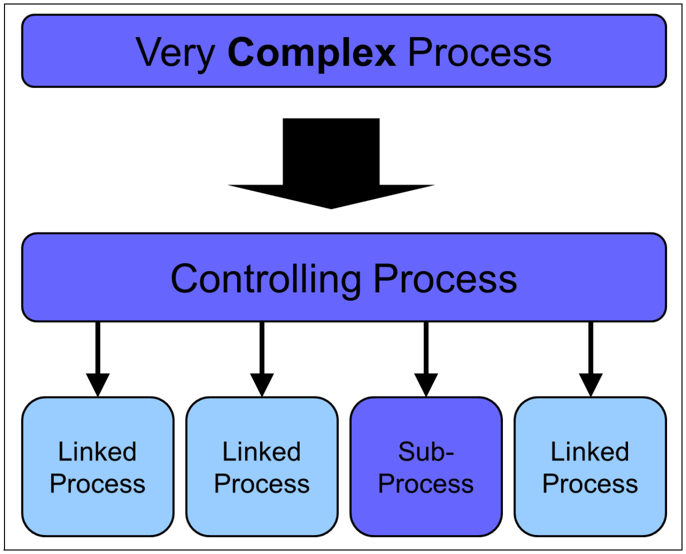


Figure 5-7 Breaking down a complex process

- *Extensively long-running* processes have long periods of time where the process is dormant but the data of the process is still relevant to the business. This can be because there is either a long waiting period, or a particular task that can be executed within an extensively long period of time.

  One example for this can be a mortgage process, where only once a year an action has to be taken. In this case, the business information must be persisted in an SOR, and the process instance is not kept alive for the whole lifetime of the process.

  Rather than keeping each process instance alive, we complete the process when it reaches its waiting point. Using batch process approaches, such as scheduled undercover agents (UCAs), queries can be executed to the SOR to determine whether any of the long-running processes have to become active.

If so, the respective process gets instantiated and executed until it reaches the next long waiting point. This activity is depicted in Figure 5-8.

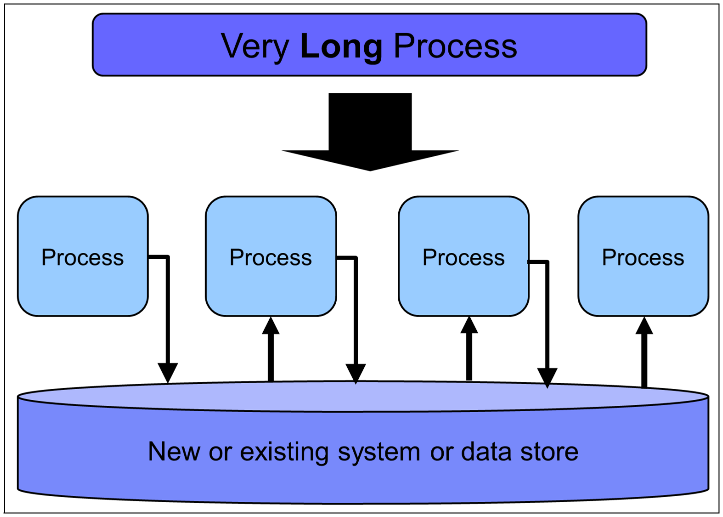


Figure 5-8 Chained process

Generic process model

Since the release of the IBM Redpaper publication *Empowering your Ad Hoc Business with IBM Business Process Manager*, REDP-4995, more and more organizations are interested in stronger support for simple processes to reduce time-to-market or on-demand process modeling to production.

This IBM Redpaper publication can provide more detailed information about how this can be achieved with ad hoc processes.

First, creating a generic process model represents the application of a particular design pattern. As you can see in Figure 5-9, this approach consists of three conceptual pieces.

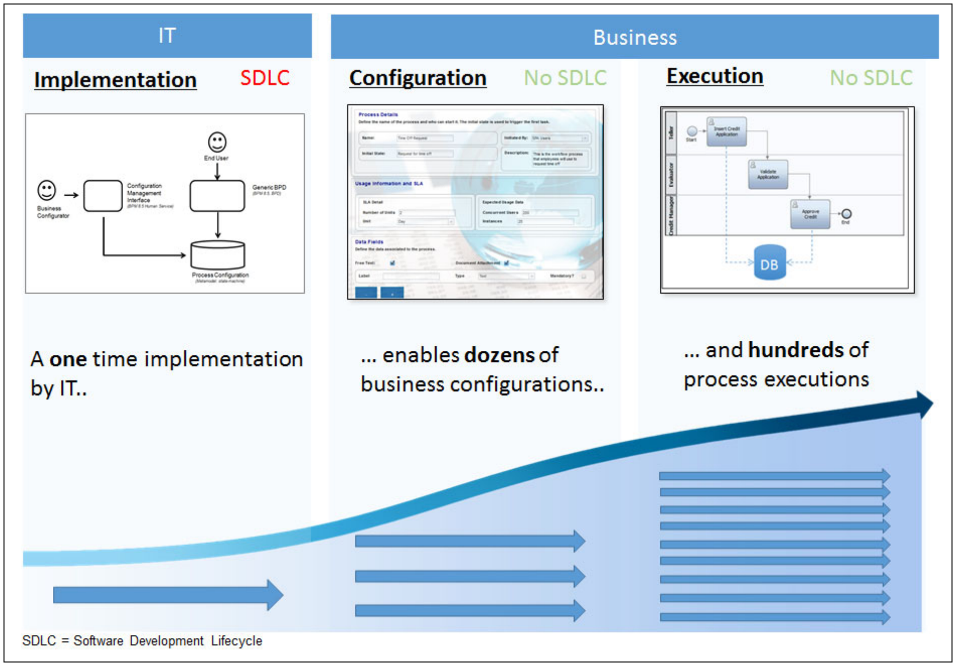

IT
Business
Implementation
SDLC
Configuration
No SDLC
Execution
No SDLC
End User
Business Configurator
Configuration Management Interface
(BPM 8.5 Human Service)
Generic BPD
(BPM 8.5 BPD)
Process Configuration
(Metamodel: state-machine)
Process Details
Usage Information and SLA
Data Fields
Teller
Evaluator
Credit Manager
Start
Insert Credit Application
Validate Application
Approve Credit
End
DB
A one time implementation by IT..
... enables dozens of business configurations..
... and hundreds of process executions
SDLC = Software Development Lifecycle

Figure 5-9 Generic process application design

The following phases take place in a generic process:

1. Implementation

   A one-time implementation of a generic process application that can be customized at run time to execute any kind of process.

2. Configuration

   After the generic process application is deployed, you can create configurations that define your actual business process alongside its elements, such as teams and data.

3. Execution

   After the configuration is created, it can be executed. Physically, a version of the abstract process implementation is running. Virtually, a functional business process gets executed.

The following list include some of the advantages of this approach:

- Instant availability of processes

  No development lifecycle, no change management, and no negotiations with the information technology (IT) department are necessary. The configuration platform is provided to the line of business (LOB), which can model processes as they want. Making it available for execution after it has been configured happens with the push of a button.

- Reporting and visibility

  Reporting and measurement, tracking of KPIs and SLAs are possible in the same way that it works with a standard BPMN process application.

- Increase of return on investment (ROI)

  The initial investment (for example, in license, platform and services) into your IBM BPM initiative is returned much faster, because the generic approach does not deliver only one process, but potentially hundreds of processes.

Although there is much value to the generic process approach, there are also some limitations:

- Integration

  Integration with an SOR, rules, or other services is generally possible. However, integration is only reasonable if a large number of processes that are expected to run on the generic platform require those. To support a generic process model, you need to have well-defined and standardized object models and interface definitions.

- Complexity

  A large variety of process flow patterns can be applied, but not all of them. The intent of a generic process pattern is to support low-complexity processes.

- Automation

  The generic process pattern solely supports human-centric processes with a low level of automation. The intent is to implement process flow automation (see “Process automation” on page 149) as opposed to task automation.

Unstructured processes

A non-structured, or unstructured, process can be defined as a process where each instance of the process differs from another instance by its execution path and its lifetime. Process improvements cannot be easily, or at all, identified for a non-structured process.

A *case* is a good example of a non-structured process. Case management is a separate discipline from the process management because of the fundamental difference between the nature of cases and structured processes.

The IBM BPM case management system can simplify the job of designing and building cases. It also provides a graphical user interface (GUI) for caseworkers to manage cases. With IBM BPM, you design a case management application that is based on closely related cases, and then deploy that solution into a production environment. Caseworkers can then complete work items that are associated with cases.

For example, if you want to design an application for resolving credit card disputes, IBM BPM provides tools to design and create an application that the caseworker can then use to process the cases that are created.

IBM BPM unifies information, processes, and people by providing you with the following functions:

- An active-content infrastructure that manages the persisted case object model and enables content-based events for case activities. For example, the addition of a document or a field change triggers a case activity.
- Enterprise Content Management (ECM) integration that includes complete document lifecycle capabilities. These capabilities include creating a version and security. Some licensing restrictions might apply.
- True content management functions to store document metadata that you can modify and search on.
- A process that manages the logic for running a sequence of activities.
- A flexible and customizable UI that is ready for immediate use. You can use the UI to create a team-based experience that can provide caseworker with the information they must work on.

Case management concepts and typical scenario

Case management applications consist of case types, document types, human services, and other components. A *case management application* is an application that caseworkers use to process cases.

Case types define the activities, and might use *document types* to support the activity. Case types also specify the teams (*human services*) that must complete the activities to solve a business problem. The case type also includes properties that are shown to a caseworker in the IBM BPM Process Portal. Case types make up a case management application. A case is an instance of a case type.

Figure 5-10 provides an overview of the structure of the IBM BPM basic case management feature.

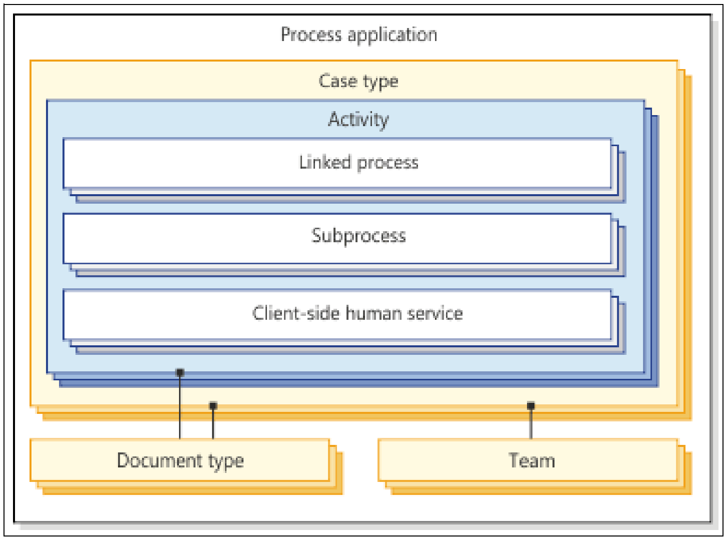


Figure 5-10 IBM BPM basic management feature structure

Business process management (BPM) and case management are complementary ways of solving business problems. The type of business situation that you are addressing determines which approach you use.

Consider the following scenario. The activities are unordered because the sequence of activities is unpredictable. The events determine the order in which the activities in the process are followed. People rather than programs interact to resolve the dispute. External documents are needed for verification. In this scenario, a case would most likely be your implementation choice.

Consider another case scenario that handles credit card billing complaints. The activities are not wired. No predetermined sequence is set at development time. A worker would likely start by describing the complaint, and finish by resolving the dispute. However, whether the worker receives the client information before the vendor information would likely be determined by what data the worker receives in the complaint.

The information that is retrieved about the client determines whether the worker checks with the people who handle the client records. The information that is retrieved about the vendor determines whether the worker checks with the people who handle vendor records. The information that is retrieved about either the client or the vendor might lead to investigating the computer billing system for problems.

The activities are not in a swimlane that determines their type. Although the worker controls the process at run time, the process is dynamic, and is influenced by current events. As in a business process, the required activities must be completed. However, optional activities do not need to be completed.

People who interact with other people, rather than automated programs, determine the activities. To verify the complaint, receipts and invoices (which are external documents independent of the case) must be captured. To implement these activities with human services, subprocesses, services, and teams, you use the same set of tools as you do to implement a business process.

Figure 5-11 outlines the BPM basic case management activities design for a credit card billing complaint system.

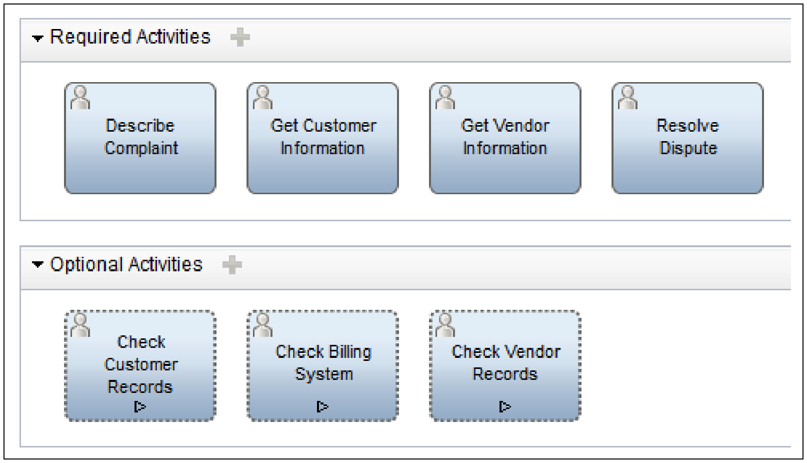


Figure 5-11 Basic case management activities design

Key differences between BPM and case management

Considering the previous scenarios, Table 5-1 summarizes the characteristics of BPM and case management. By examining these characteristics, you can select the type of process that best suits your situation.

Table 5-1 Characteristics of BPM and case management

BPM	Case management
You can define an ordered sequence of activities that can be completed to solve a business challenge.	You can define an unordered set of activities that can be completed to solve a business challenge.
The sequence of activities is stable and seldom changes. The process is predicable and repeatable.	The activities occur in an unpredictable order.
The process determines the events. The first activity determines the first set of events, which then leads to the next activity and the next set of events. The activities are wired to one another, which determines the sequence.	The events determine the process. As events occur, a worker selects the appropriate activity. The resulting process can vary depending on the current event and the subsequent selection by the worker. Activities are not wired to each another.
The activities are often programmatic. A repeatable sequence, such as selecting a set of potential credit card owners from a database, can be automated.	People primarily determine the activities. Handling a client with a billing error is done by a person who uses judgment to determine the best resolution of this particular case.
External documents are not an essential part of the process.	External documents play a key role. For example, receipts provide a record for how the problem that must be resolved began.

5.2.2 Routing patterns

Routing patterns apply for non-structured and structured processes. In either case, you need to define who performs the work that needs to be done. The following list describes the most popular routing patterns:

- Pull
- Push
- Attribute-based routing

One example everyone is familiar with, is supermarket check-out lines. The most commonly used system is that buyers line up at any line they like. This can be compared to the *push* pattern. Work (a cart full of items) gets pushed to a particular system or user (the cashier) to be completed.

The more full a cart is, the more complex is the task (or activity), and the larger the likelihood that this job takes longer than others. Naturally, everyone is looking for the shortest line with the least number of items. However, unanticipated events can cause the process to deviate from expected behavior:

- Consider that you are in a line where the person in front of you has only two items, but takes a long time to count every penny he has in his pocket?
- In addition, suppose you are next in line, but suddenly the cashier's shift ends and you have to wait until the next one is ready to go?

This is exactly where inefficiencies come into play. Many supermarkets have realized that this effect can cause frustration with their clients, and they optimized their check-out system, from a push pattern to a *pull* pattern.

Using the *pull* pattern, there is only one single line for all registers. As soon as a cashier is ready for the next client, he indicates this, the client moves forward and proceeds with the check-out. Although the line appears longer, the overall checkout time has been proven to be shorter and therefore more efficient.

Figure 5-12 illustrates the supermarket example with the push versus pull model.

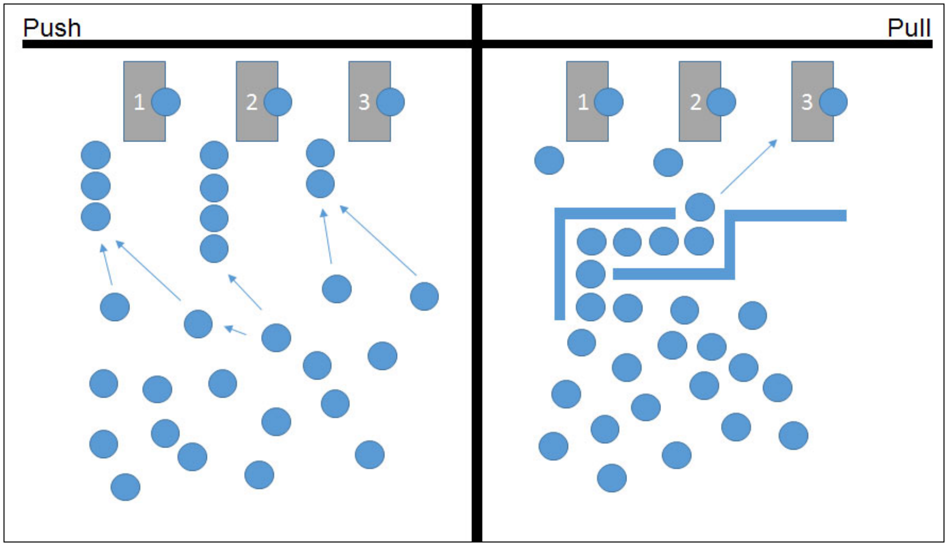


Figure 5-12 Push versus Pull model

We can apply the same pattern to a BPM system.

Pull

Pull correlates to the supermarket example where we have one line for all registers.

In the context of IBM BPM, *pull* means that activities get assigned to a group of people (team) rather than the individual (user). Everyone in the group is allowed to execute the task. As soon as someone starts the task, it gets claimed, which means, that this task is only visible and available for this one person. Nobody else can work on it. If we now make sure that the queue of work items is prioritized, we have a very efficient way to work on tasks with a group of people.

Push

Push correlates to the supermarket example where people line up at the register with the shortest line.

In the context of IBM BPM, push means that activities get directly assigned to an individual (user). The assignment can be based on many different factors, including but not limited to the following options:

- Number of open work items
- Priority
- Round Robin methodology
- Shortest estimated completion time

This is a pattern that we often see in established applications. Sometimes, the process is as inefficient as having one person (for example, a supervisor) assign the work items to individuals. Other times, the process uses more or less sophisticated assignment algorithms.

Attribute-based routing

Attribute-based routing can be applied for push and pull systems. In both cases, the user information includes attributes that are used for filtering. The following list includes some of the more often-used attributes:

- Language
- Skill
- Department

However, other attributes are possible as well. Using these attributes as routing criteria, so-called *dynamic groups* are created inside IBM BPM. Note that dynamic groups are not editable through the process admin console.

In the *pull* system, the user attributes are used to create the *group of users* that receive the task. In the *push* system, the user attributes plus additional routing or distribution information (such as load-balanced, round robin, or others) are used to assign a task directly to a particular user.

5.3 Service design

Service design consists of human services and system services. At this point, we are talking about the implementation artifacts in IBM BPM that are called *services* (human services and system services). These are not necessarily services in the sense of a service-oriented architecture. Integration services are described in 5.4.4, “Integration design” on page 183.

5.3.1 Service complexity and reuse

Don’t be afraid of wrapper services. There is no harm in nesting services or creating wrapper services. Even the other way around, the smaller you cut the complexity of services, the easier they are understood by the development or maintenance team, and the easier it is to reuse them.

The larger services are, the harder they are to understand. This may sound trivial. However, try to apply the following rules (they are not hard rules) to make it easier to reuse your services and change code artifacts:

- Never copy and paste

  If you are about to copy and paste something, do not do it. Put it in a service and reuse it.

- Do not cross lines

  If you cross lines, you most likely reach a complexity that is worth breaking up into smaller pieces. Also, do not forget the Rule of Seven (as described in “Patterns, anti-patterns, and modeling risks” on page 155).

- One Coach per service

  If you have many Coaches within one service, all the navigation, data initialization, and validation can cause much inconvenience for anyone who tries to change the service. Wrap the Coach into a separate human service. You might want to reuse it stand-alone anyways.

With anything that you are implementing, always remember that you potentially want to reuse it. Avoid large JavaScript blocks, because JavaScript (client-side and server-side) is interpreted at run time and therefore slower to process than compiled artifacts, such as Java code. Large client-side JavaScript blocks can also produce very large Document Object Model (DOM) trees, which are expensive for browsers to process and render.

Finally, large JavaScript blocks are often indicative of too much logic being placed in the IBM BPM layer. As a general guideline, limit a JavaScript block to 50 lines. If your implementation exceeds this value, reevaluate the design and refactor the implementation to use smaller JavaScript blocks.

5.3.2 Human service design

The IBM Coach Framework is a key element of the IBM BPM product suite. With the Coach Framework, process authors can create and maintain custom web-based UIs that are embedded in their business process solutions. This ability to create and maintain custom UIs is a key factor in the successful deployment of business process solutions.

There are several design considerations that you should remember when working with Coaches. This section covers the following topics:

- Client-side versus server-side human services
- Coach View design
- Responsive design

For a more detailed description about how to build Coaches, see the IBM Redbooks publication *Leveraging the IBM BPM Coach Framework in Your Organization*, SG24-8210.

Client-side versus server-side human services

IBM BPM was from the start designed to make it as simple as possible to implement changes to processes, Coaches, and so on. The thinking behind this was that speed of change is critical, because business requirements change at an ever-faster rate. IBM BPM was designed with this in mind, and part of this was also to make it easy to deploy newer versions of a process. This means that inflight processes can be upgraded to a newer version.

Historically, a service is executed on the server until a Coach is encountered. The Coach is then generated, and the data is bound to the variable and served up to the browser. When the user selects a button, the control is returned to the server, where the service continues to execute until it encounters the next Coach. This is what we refer to as a *server-side human service*.

IBM BPM 8.5.5 introduced the new *client-side human services*. With the new client-side human service, the service is downloaded to the browser and executed. When a Coach is encountered in this case, it is rendered in the same browser. Client-side coaches provide the following key benefits:

- Enable the use of client-side capabilities, such as client-side validation.
- Help offload the server by executing services in the local browser.
- Reduce the communication between the browser and the server.
- Reduce the number of server requests and size of data on Coach transitions.
- Reduce server-side resource use (more concurrent users with less performance effect).

When you start to think about using either server-side or client-side human services, consider the points listed in Table 5-2.

Table 5-2 Differences between server-side and client-side human services

	Heritage human services	**Client-side human services**
Authoring	Eclipse	Web editor
Runtime	On BPM server	Client-side, in the web browser, with calls for data to the server only when necessary
Case Management Support	Not available	Available
Collaboration support	Available	Not available
Supported Coach types	Available for both new Coaches and heritage Coaches	Available for new Coaches only (heritage Coaches are not supported)
Usage of JavaScript	Server scripts that use server script syntax	Client-side scripts that use standard JavaScript syntax

Coach View design

Coach Views are the reusable UI building blocks that authors use to compose their UIs.

Coach Views can be divided into *atomic* and *composite* views:

- *Atomic Coach views* are authored by programmers who have a deep understanding of HTML, JavaScript, and Ajax services. They often use third-party JavaScript libraries to provide sophisticated behaviors.

  Most atomic Coach Views are general-purpose components that are reused across a wide variety of UIs. An example of an Atomic Coach View is the components that are included with IBM BPM, such as buttons or text fields.

- *Composite Coach views* are authored by business subject matter experts (SMEs). Most Coach Views that present and manipulate specific business object types are Composite Coach Views.

  Composite Coach Views are often created when many of the Coaches of a solution include a common section of UI components. A common header that contains overview information about a process is an example of a section that is often implemented with a Composite Coach View.

By using the Coach Framework in IBM BPM with atomic and composite Coach views, you have many possibilities to create UIs that meet your business needs the best.

As you design your Coaches, you have to consider the implications or trade-offs that the design includes. The two most prominent considerations are *performance* and *business agility*:

- Performance

  Generally speaking, the older the browser you are using and the more complicated the Coach UI, the longer the Coach will take to render. The following specific issues affect performance:

  - Large DOM trees

    Business objects that are bound to Coach views, such as industry-standard schemas including Association for Cooperative Operations Research and Development (ACORD) for insurance, Health Insurance Portability and Accountability Act (HIPAA), and so on, are persisted when boundary events occur.

    This can be a costly operation, especially for large businesses. if the business object used in the process flow is large and complex, you can reduce this cost. To do so, create a separate business object that only contains fields that are relevant to the Coach UI, and bind that business object to the Coach View.

  - Large JavaScript scripts
  - Deeply nested Coach Views

  For better performance, make as few network requests as possible to the process server. This is particularly important if the runtime environment has high network latency, or if it uses a relatively low-bandwidth network.

  The following techniques minimize network requests:

  - Combine multiple JavaScript and Cascading Style Sheets (CSS) files into one (or a few).
  - Design coarse-grained API calls that minimize the number of API calls that flow over the network.
  - Perform `create`, `retrieve`, `update`, and `delete` operations to external systems of record (SORs) before or after the Coach, rather than using Ajax.

- Business agility

  Supporting business agility enables your organization to react to the changes of the business as fast as possible, adjusting to the demands of the market. In IBM BPM, we want to be able to understand, maintain, and change pieces of code (such as Coaches) as easily and quickly as possible. The following Coach design factors impair business agility:

  – Many or deeply nested Coach Views

    Besides affecting the performance of a Coach, having many Coach Views can also affect the ability to change the UI easily. Make sure that the Coach has only the information that the business user needs at this moment, as opposed to all of the information that he can possibly have. You can also consider breaking up a Coach into multiple Coaches.

  – Complex flow of Coach View to Coach View calls

    Using JavaScript, it is possible to create a sequence of Coach View calls inside a Coach. You can even add conditionality to it. For example, a rule might state that if the loan amount is greater than USD 1,000,000, call the `simple loan` Coach view. Otherwise, call the `complex loan` Coach view. This enables you to put much business logic into a Coach View. However, it can take a developer a long time to understand all of the dependencies.

  – Button indirection

    One of the advantages of the way that IBM BPM is built is that it is visual. Business logic can be understood by looking at the flow of processes and services. In a human service, you usually have a 1:1 correlation between the buttons that exist on the Coach and the flows (boundary events) that leave the Coach widget.

    However, the same way other Coach Views can call each other, buttons can call several other Coach Views out of which one implements the boundary event. Looking at a human service, you might have difficulty correlating the button that you pressed on your rendered Coach in the browser with the flow that left the Coach widget on the server.

  – Create, retrieve, update, and delete on the Coach

    It is tempting to perform many Ajax operations on a Coach. After all, this way you can even more easily reuse it in other Coaches. However, besides the performance effect that the network requests can have, at design time it is harder to understand which service depends on which control.

To summarize, the Coach development framework in IBM BPM is a powerful tool to create UIs of all complexities. As with most things, design decisions about Coach Views also have consequences. Make sure that you are aware of the implications, and ensure that the value of the design decision is bigger than the trade-off.

Responsive design

Responsive Coach Views can adapt to different form factors in a flexible way. You can build UIs that are responsive to a user's runtime environment by using the new responsive settings for Coach Views available with IBM BPM V8.5.5. You can configure properties, such as visibility, formatting, or presentation style, for different screen sizes. Therefore, you can design one interface that changes in appearance and behavior based on the user's screen size.

The same code is rendered on IBM Worklight® mobile applications, default IBM BPM Coaches, mobile browsers. The code can even be embedded in other applications, such as IBM Notes and IBM Connections.

Figure 5-13 provides an overview of the environments where a single dynamic Coach View can be displayed.

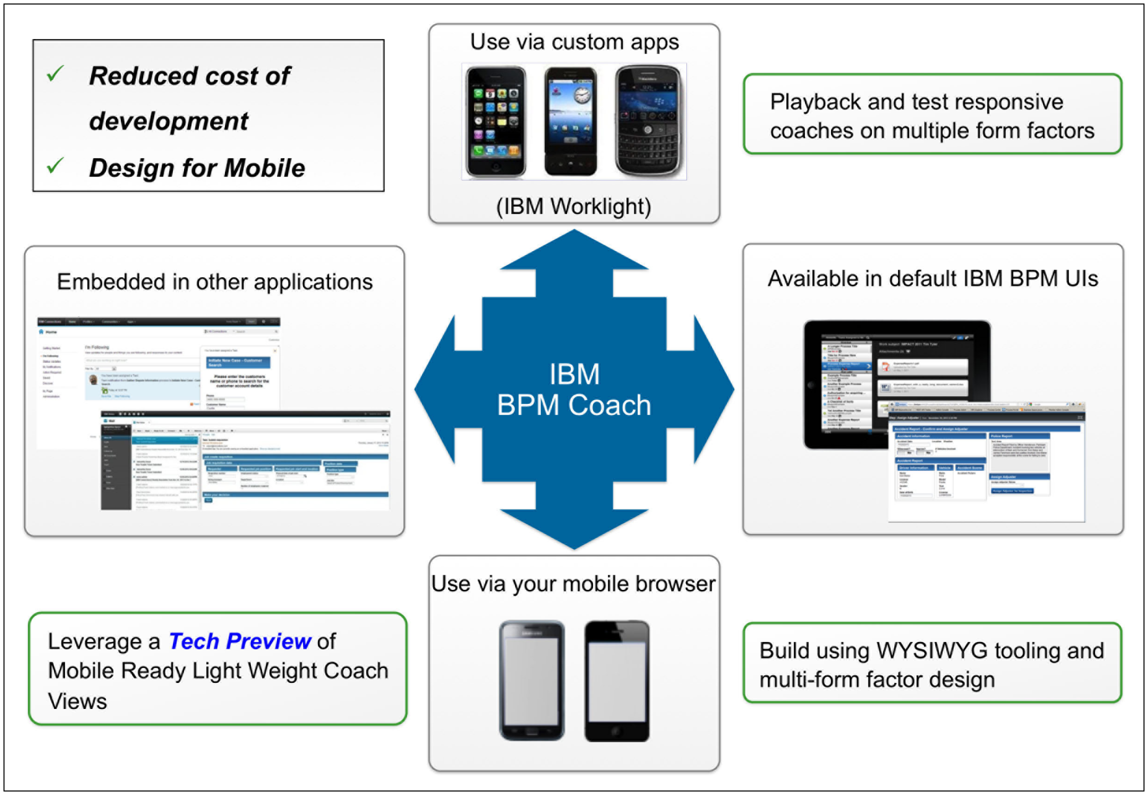

✓ Reduced cost of development
✓ Design for Mobile
Use via custom apps
(IBM Worklight)
Playback and test responsive coaches on multiple form factors
Embedded in other applications
IBM BPM Coach
Available in default IBM BPM UIs
Use via your mobile browser
Leverage a Tech Preview of Mobile Ready Light Weight Coach Views
Build using WYSIWYG tooling and multi-form factor design

Figure 5-13 Responsive Coach design: Single Coach for multiple form factors

The responsive Coach design feature enables you to accomplish the following tasks:

- Create mobile parts of the Coach design.
- Build and maintain several form factor designs at the same time for every Coach View.
- Play back Coaches in the IBM BPM development environment before deployment.
- Use responsive Coaches in any external UI frameworks, including IBM Worklight and custom mobile applications.

5.4 Data flow patterns

A process without data is barely a process. However, you must consider how much data must be in the process. Too much data can affect performance, but too little data decreases the value for the business. In this section, we describe the different data management options to help you make the correct decision about how to handle data in your business process application.

5.4.1 Data flow in processes

Variables are persisted to the database when execution contexts are saved, which happens fairly frequently (for example, when changing from BPD to service execution). These persistence operations use resources. Minimize the persistence cost in the following ways:

- Minimize the number of variables used.
- Minimize the size of each variable.
- Minimize the number and size of variables that are passed to and from each task (human task or system task).

A common misconception is that the complete business object model exists in the process (in this case, the BPD layer). In fact, the process layer must only have the information that is necessary to execute this process.

To manage the data flow of objects that are required inside the process, two major approaches are used. Call by *reference* and call by *value*:

- Call by reference

  With *call by reference*, the object model on the process layer contains only the bare minimum information, which is required for the following items:

  - SOR IDs
  - Naming process and task instances
  - Routing tasks
  - Searching process and task instances
  - Tracking process information

  The SOR IDs are passed into the services (human or system) so that they can retrieve the complete information by themselves from an external data source.

  The advantage of this approach is that it is the most lightweight approach. The disadvantage is that every service has to perform resource-intensive integrations to retrieve and save data.

- Call by value

  With *call by value*, the object model on the process layer contains the sum of all information that is necessary to perform the tasks. The process passes all required information to the task, and the task returns all changed information. This data handling approach *uses the most resources*.

  The advantage of this approach is that it can theoretically exist without an SOR. However, it is advised to create an SOR for auditing, reporting, and other reasons. The disadvantages are that this is the most expensive approach because large data objects are persisted in the runtime environment and database, and therefore this approach uses the largest amount of resources.

5.4.2 Data flow in services

Similarly to the data flow in the process layer, the service layer has a different implementation when *reference* or *value* has been chosen as the wanted data flow pattern:

- If you chose the *reference* data flow pattern for the process, the service has to perform create, retrieve, update, and delete operations to an SOR. These operations are usually performed at the very beginning and the very end of the service implementation.

  To reduce the footprint (consumption of resources) that the service leaves on the server, you want to clear out (set null) all variables, particularly large business objects, when they are not used anymore.

Try to minimize the number of requests to an SOR:

  - Rather than writing multiple Structured Query Language (SQL) statements, create a stored procedure on the database that returns all of the necessary information.
  - Rather than calling multiple web services, have a IBM BPM Advanced or enterprise service bus (ESB) service collect all of the required information for you.

  Make sure that you only request information that you need to perform the task.

- If you chose the *value* data flow pattern for the process, you have to make sure that the following conditions are true:
  - Only the subobjects that are required for the service are passed into it. Any additional object creates unnecessary resource use.
  - Only objects that potentially could have been changed in the service are passed back to the process. Any additional object is considered unnecessary resource use.
  - You do not update the object model with information from an SOR. If information could have been updated by another system, use *call by reference*.
  - If you had to create helper objects (complete drop-down menus, and so on) make sure that you clear them on exit.

5.4.3 Data flow in Coaches

The rules that apply for data flow on the process and service level apply similarly for Coaches.

- The larger the business object, the slower the Coach renders.
- The more network calls (for example, Ajax) that are executed on the Coach, the more the UI performance is affected.

Important information about how to handle data on Coach Views can be found in the following sources:

- “Coach View design” on page 176
- *IBM Business Process Manager V8.5 Performance Tuning and Best Practices*, SG24-8216
- *Leveraging the IBM BPM Coach Framework in Your Organization*, SG24-8210

5.4.4 Integration design

In this section, we describe the following topics:

- Service orchestration and choreography implementation types
- Difference between mediation flow component and short-running BPEL
- When IBM BPM Advanced Integration Service should be used
- Transactionality

Service orchestration and choreography implementation types

The following list describes the common styles of solutions suggested for process integration initiatives associated with Service Integration and Maturity Model (SIMM) maturity level 5 using IBM BPM Advanced:

- GUI-intensive process

  This represents the navigational flow and data aggregation. Typically a single user flow, through collaborative GUIs, might transfer flow. IBM BPM Process Server can be used in the background to provide swiftly responding synchronous services to the UI. GUI-intensive processes are also less linear, enabling the user to hop backwards and forwards.

  Also see “Process flow, page flow, and service orchestration” on page 150.

- Synchronous transactional process

  This is a typical use of a short-running BPEL processes. It is often used for real-time responses to GUIs, or for transactional subprocesses.

- Asynchronously initiated transactional process

  The caller waits only for a one-way request to be accepted. An actual transactional process occurs in the background. This process is used to improve apparent response time where the caller only requires an acknowledgment, as opposed to a completion.

- Briefly persisted process

  This is a special use of a long-running process, where the process completes relatively swiftly. The process lifespan is deliberately short (seconds, maybe minutes), such that issues created by process versions can be avoided. No human tasks or in-process events can be present. The process is commonly used to manage parallel aggregation. The process must be easily quiesced without significant business effect to enable new versions to be deployed.

- Versioned, long-lived process

  A true long-running process that can last a relatively long time (days, weeks, and more). Notice that *hours* are the gray area here, between these and briefly persisted processes. Process instances are always present in the system, so that complex issues of creating process versions must be considered. This process type can contain human tasks and complex error handling, such as compensation. It can receive in-process events.

- Task-based process

  This process type is used to balance multiple tasks between several different users, possibly in different teams or departments. It requires long-lived processes, so you must also consider issues creating process versions.

For more information about these solutions styles, see the following IBM developerWorks article:

http://www.ibm.com/developerworks/websphere/library/techarticles/1004_clark/1004_clark.html

IBM BPM Standard provides integration features, such as non-transactional database interactions, SOAP web services (SOAPWS), REST web services (RESTWS), and Java integration services. For more information about IBM BPM Standard integration services, see the following IBM developerWorks article:

http://www.ibm.com/developerworks/websphere/bpmjournal/1109_cheng2/1109_cheng2.html

Difference between mediation flow component and short-running BPEL

In this section we describe the differences between mediation flow components and short-running BPEL flows.

Mediation flow component

The mediation flow component provides the mediation function on messages between service requesters and service providers. It provides low-level technical implementation for brokerage and connectivity functions, including the following features:

- Message routing, transformation, and enrichment
- Aggregation
- Connectivity across platforms and protocols
- Content or header-based routing of messages

The mediation flows can be used to start the back-end systems in which business context does not change.

Short-running BPEL flow

The short-running BPEL, also called a *microflow*, is a synchronous or stateless service that is used when the corresponding business transaction is fully automated and completes within a short time frame. Microflows run in a single physical transaction, and they are non-interruptible.

Microflows can include the integration logic that entails the invocation of one or more mediation flow components. They provide a complex choreography of sequences of requests to mature exposed services in a single transaction. In IBM BPM, looping constructs, such as `foreach`, `while`, and `repeat until`, are much easier to implement in a microflow than in a mediation flow component.

The choice between the mediation flow component and the microflow is driven by the pattern that you want to implement. For example, consider using the mediation flow component for service virtualization and brokerage patterns. Nevertheless, microflow would be appropriate for implementing composition patterns, such as state-free orchestration. We suggest that you refrain from implementing business logic in a mediation flow component.

When IBM BPM Advanced Integration Service should be used

In IBM BPM, IBM BPM Advanced Integration Service (AIS) is used to implement business services that are to be used by a business process. Business services are steps in a business process that are not apparent to the business users.

These are steps that business users are typically not interested in monitoring. In other words, a business service is a low-level technical implementation that updates the business context but is not of much interest to the business users.

In IBM BPM, an AIS implementation is a collaboration between a process developer working with IBM Process Designer and an integration developer working with IBM Integration Designer. The process developer designs the AIS by defining the input and output interfaces of the AIS. The process developer does not need to know the technical details about the implementation of the business service.

Technical details, such as Service Component Architecture (SCA), Java programming, transactional qualifiers, and so on, are left to the integration developer. In doing so, AIS offers a good separation of business and technical considerations.

In IBM BPM, an AIS interface corresponds to a single operation of a Web Services Description Language (WSDL) definition, and is exposed by an SCA export of an implementation module.

These components can use the full power of SOA through BPEL-based micro and macro flow constructs, ESB-based mediation flows, integration with IBM Java Platform, Enterprise Edition (Java EE) Connector Architecture (JCA)-based connectors, interface and data mapping, relationships, and other advanced process choreography capabilities.

AIS can be a good match in a scenario that involves STP of business steps that are triggered by a business user. In this situation, the human interaction required to start the business steps is implemented by the process developer in IBM Process Designer.

The business steps are implemented as a business service in IBM Integration Designer. In this example, if the SCA module that implements the business service is returning values back to the human service, it is important to ensure that the SCA module is synchronous. This is because the human service is not allowed to suspend. For design considerations regarding using AIS, see 3.2.2, “Top Down versus Bottom Up design considerations” on page 81.

Transactionality

In this section, we describe the following aspects of transactionality:

- Transaction types in IBM BPM
- Synchronous versus asynchronous transactions

Transaction types in IBM BPM

The following transaction types are available in IBM BPM:

- Local transaction is a single phase transaction that cannot be combined in a two-phase commit.
- Global transaction is a transaction that can coordinate two phase transactions to multiple XA-compliant targets.
- Web Services Atomic Transaction (WS-AT) protocol enables transactional requests exposed through a web service to take part in a two-phase commit coordinated by the caller. For more information about WS-AT, see the following developerWorks article:

  http://www.ibm.com/developerworks/websphere/library/techarticles/0703_xu/0703_xu.html
- Activity sessions extend the concept of transactions by grouping multiple related transactions. They don’t provide rollback facility in the way two-phase commit does. Generally, it would be better to handle calls to multiple one-phase resources using a long-running process, which would provide clearer error handling, persistence of partially complete processes, resubmission mechanisms, and more. Activity sessions can provide better performance, but you have to implement the previous transaction yourself.

Synchronous versus asynchronous transactions

Services provided over a controlled local infrastructure can make design assumptions that enable true transactionality, and therefore synchronous interaction patterns can be used in such situations. As a result of the unreliable medium between consumer and provider, enterprise services can only provide true reliability when using an asynchronous pattern of behavior. Truly asynchronous interaction provides the following benefits:

- Requesting thread is not blocked.
- System non-availability problems associated with a provider are naturally handled with the implicit store and forward pattern, therefore removing the need for more coding.

Consider the following points in context of reliability in enterprise services:

- If you are looking for assured delivery, you must include instances where delivery is tried again, and consider how to manage the errors if those tries fail.
- If duplicates matter, your design must account for idempotence, or produce errors on duplicates.

In summary, if you need assured delivery and idempotence, your error handling pattern is typically asynchronous.

In BPM Advanced, we suggest that you avoid designs where a component synchronously calls another component that has an asynchronous implementation (also called *deferred response*) for the following reasons:

- Any synchronous to asynchronous switch, regardless of the technology, introduces the chance of *in-doubt* transactions, and as a result, also the possibility of duplicates. This is caused by the mismatch of single-transaction expectations of the caller, and the multi-transaction nature of asynchronous communication.
- The performance expectations of an asynchronous system are based around throughput. It does not guarantee quick response time on individual transactions. For example, if the response time of the asynchronous transaction is greater than the global transaction timeout of the synchronous call, the asynchronous request completes, but the synchronous request receives a timeout exception.
- Synchronous calls hold a thread open throughout the duration of synchronous calls. Asynchronous calls, conversely, do not hold the threads open, because the focus is on overall throughout. This behavior can result in several blocked threads, and therefore thread pool depletion over time.
- Switching from a synchronous innovation to an asynchronous invocation typically involves a protocol transaction, which is inherently complex.

5.5 Toolkit design

Toolkit is an IBM BPM mechanism for sharing library items across different process applications and other libraries. Although toolkits can contain the same type of artifacts as process applications, toolkits are non-executable assets and cannot be deployed on their own to the runtime server.

Toolkit design has significant effect on the maintainability, serviceability, and scalability of the process application. Toolkit design should be carefully considered during initial process and integration solution architecture.

Toolkits, per design, are meant for reuse purposes. The most common challenges during the initial process application design are *classification* and *grouping* library items into respective toolkits. In addition to those challenges, ongoing maintenance of toolkit artifacts during active development processes can also be a demanding exercise.

Next, we look at some typical challenges accompanying the toolkit design:

- Categorizing library items into toolkits
- Toolkit maintenance
- Design patterns and performance considerations

5.5.1 Categorizing library items into toolkits

The most common practice is to group library items into toolkits by their types or application purposes. For example, reusable business object types are placed into a Business Objects toolkit. Integration services related to a single system can be placed into a separate toolkit if this library is intended to be reused by other process applications, or by another toolkit.

This approach seems simple and straightforward. However, as the number of shared libraries grow along with the toolkits, it can soon become apparent that you need further thorough analysis, and a more extendable toolkit design approach. It is especially difficult to see this upcoming challenge at the beginning of a project when the development team adds only one or two process-related toolkits.

Analyzing the following areas can help to outline a list of potential toolkits:

- Business objects libraries

  It is not uncommon for the business objects toolkit to include integration services that use those objects definitions. This approach can provide a certain degree of encapsulation for integration services and their respective business objects.

- Integration-specific related toolkits

  Bringing together integration services related to persisting data in the existing repository that includes several existing software solutions can be an example of an integration-specific toolkit. It might not make much sense to create individual toolkits for each existing system, but rather to place all existing system-related integrations into a single toolkit.

  Such an approach will keep all existing related integrations in a single library, enabling easy disassociation of the existing integration services from the next generation of services when the former are fully decommissioned.

- System-specific or business purpose-specific

  For example, an IBM FileNet toolkit can include client-owned FileNet related integration services. In addition, business objects supporting the IBM FileNet-specific operations can also be defined in this toolkit. This classification might work better if there are fewer integration systems that the IBM BPM solution needs to interact with. A client solution with a larger number of integrations might, otherwise, have to deal with many dependant toolkits.

- User interface library items

  This is a good way to share reusable UI-related items, such as common libraries of JavaScipt code, CSS style sheets, or Coach Views, across multiple process applications.

- Third-party libraries

  Keeping a toolkit's size to a minimum is a worthy goal. When using third-party toolkits, there are not that many options to achieve such a goal. There is generally no documentation that comes with the toolkit that describes the toolkit's internal dependencies. Refactoring a third-party toolkit's code, in an attempt to break it down into smaller libraries, can prove costly and often yield no guarantee that the final result is as stable as the original toolkit version.

  In cases when you strongly want to reduce the size of the toolkit, one safe option that can be used is to contact the toolkit vendor with a request to refactor the toolkit code. There is a good chance that the toolkit vendor will cooperate and offer an acceptable alternative toolkit implementation. After all, it is in the toolkit vendor's best interest to keep their clients happy.

5.5.2 Toolkit maintenance

We advise you to regularly clean up and keep your list of toolkits to only ones that you actually use. If you no longer need a toolkit, you can delete it from the repository. Deleting a toolkit removes all of the files and other artifacts associated with the toolkit. You can delete any toolkit except the system toolkits.

Before you delete a toolkit, perform the following tasks:

- Archive the toolkit. Archiving deactivates all snapshots. If the toolkit contains artifacts specific to IBM BPM Advanced (for example, AIS), the snapshots on the IBM BPM Process Center (Process Center) server are also stopped and removed.
- Ensure that no process application or toolkit references the toolkit that you want to delete. If any process application or toolkit ever referenced this toolkit, those process applications or toolkits need to be deleted first. This is the only way to remove the references.

For more information about toolkit dependencies and archiving, see the IBM Knowledge Center that describes deleting toolkits:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.admin.doc/topics/tadm_delete_toolkits.html?cp=SSFPJS_8.5.5&lang=en

If the number of library items that are actually being used grows to the point where they affect deployment timing, you can revisit the initial toolkit categorization choice that was originally made at the beginning of the project.

Before proceeding with the refactoring exercise, we advise you to assess and analyze the overall effort to be spent on refactoring and testing against the potential benefits of the refactoring. We also advise you to thoroughly analyze toolkit dependencies and hierarchy at the initial design phase of the project, to avoid potentially troublesome refactoring activities.

5.5.3 Design patterns and performance considerations

Applying software design patterns in IBM BPM process solutions is considered a good practice, and is highly encouraged.

IBM BPM Advanced provides two separate integrated development environments (IDE), namely IBM Process Designer and IBM Integration Designer. They both interact with the Process Center, which provides a shared code artifacts repository and the development runtime environment.

When designing toolkits for integration services using IBM BPM AIS artifacts, we advise you to use a facade pattern. This is a good pattern, because it avoids excessive or accidental breakages by shielding the data types and the interfaces, and isolating the models from changes introduced through other tools.

For information about specifics of the facade pattern implementation in IBM BPM, see the IBM developerWorks site:

http://www.ibm.com/developerworks/websphere/bpmjournal/1112_pacholski/1112_pacholski.html

For more general information about the facade pattern definition, see Martin Fowlers's site:

http://martinfowler.com/eaaCatalog/remoteFacade.html

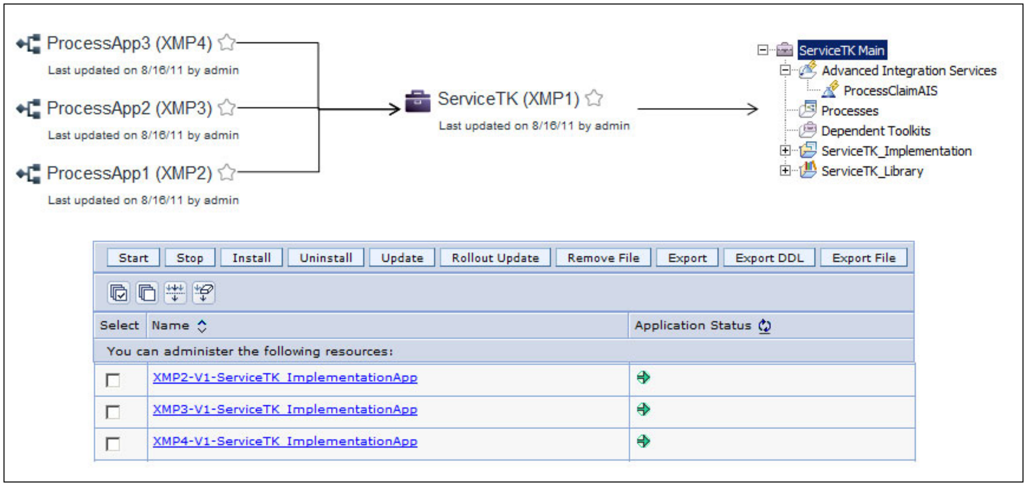
Given the separated development environment landscape, when implementing the facade pattern in IBM BPM Advanced, you should be aware of design choices and the corresponding consequences.

Reuse: At deployment time, toolkits are reused by the process applications *by copy and not by reference*. A new copy of the toolkit is created for every process application that references it.

Toolkits are reused by process applications and other toolkits *by copy*. To take advantage of new changes made in a toolkit, you need to create and deploy a new snapshot of the toolkit where the change was made.

As a result of the snapshot's proliferation, when frequent changes are made during an active development cycle, there is a potential negative effect on the performance of the runtime environment. Such a negative effect can be mitigated by removing unused snapshots from the runtime environment. Keeping the toolkit's size to a minimum when designing the toolkit is also a leading practice.

Figure 5-14 provides an illustration of how toolkits get referenced by different process applications.



Select	Name	Application Status
You can administer the following resources:		
□	XMP2-V1-ServiceTK_ImplementationApp	➜
□	XMP3-V1-ServiceTK_ImplementationApp	➜
□	XMP4-V1-ServiceTK_ImplementationApp	➜

Figure 5-14 Toolkits reuse by copy

In addition to cleaning the runtime environment from the unused snapshots, the potential negative effect can be mitigated by using the Service facade pattern. Next, we look at the main benefits of applying the Service facade pattern during integrations development:

- To limit an excessive number of runtime artifacts deployed onto the Process Center
- To promote isolation of private service interfaces defined in IBM AIS
- To promote and support loose coupling between business objects definitions made in the IBM Process Designer and IBM Integration Designer environments

  For more information about business object model design considerations, see 3.3.3, “IBM BPM business object model design considerations” on page 92.

When applying the Service facade pattern to the IBM AIS implementation, it is important to consider the different types of facade implementations, and recognize their strengths and limitations:

- Managed implementation option

  For the facade pattern implementation with IBM AIS, you can consider placing the AIS implementation in the process application to reduce multiple implementation copies of it. This approach is often referred to as *managed implementation*, because all related facade interfaces and their implementation are managed in the Process Center.

  Note that you still get multiple copies of the facade toolkit every time it is being referenced. On the positive side, it is a relatively thin deployment asset, because the facade contains interface definitions only.

  On the negative side, the code is deployed in three locations:

  - The facade interface definition in the toolkit
  - The process application with the facade implementation
  - The corresponding service code in IBM Integration Designer

  All of these copies need to be in sync. In some cases of frequently changing integration services during development, keeping these three parts in sync can become an additional maintenance challenge.

- Unmanaged implementation option

  In this facade version, services are implemented as SCA assets on the IBM Integration Designer side. The difference between the managed and unmanaged options is that there is no process application asset that is managed by the Process Center. Service implementation code is relatively decoupled from its interface definition, which is part of the facade toolkit.

- Unmanaged implementation with dynamic end-point location

  When end-point location is unknown or inaccessible at design time, it is worth considering a dynamic endpoint selection (DES) pattern inside the facade implementation.

  For more information about the DES pattern, see the IBM Knowledge Center about creating DES patterns:

  http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.auth.stp.doc/services/topics/creatingdespattern.html?lang=en

For a complex routing logic, or when you have many endpoints, you can consider using IBM Operational Decision Management (ODM) as a rule-based DES pattern implementation.

Figure 5-15 represents a typical rule-based endpoint selection scenario.

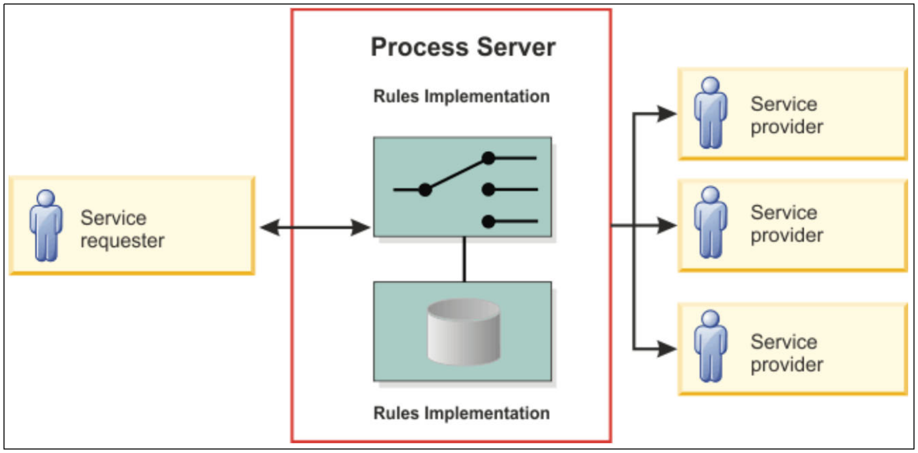


Figure 5-15 Rule-based endpoint selection scenario

For more information about the rule-based DES pattern implementation, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.auth.stp.doc/services/topics/rules_based_des_pattern.html?lang=en

In this section, we looked at the various aspects of toolkit design. We covered toolkit classification and grouping approaches. In addition, we described toolkit maintenance aspects and performance considerations. We outlined strategies related to usage of IBM AIS in advanced integration scenarios.

As the number of scenarios and specific circumstances vary, it is always a good practice to look at the toolkit design as a critical part of the overall process solution design. The toolkit design touches, and has the potential to affect, many important aspects of the solution, such as development, maintenance, deployment, and solution performance.

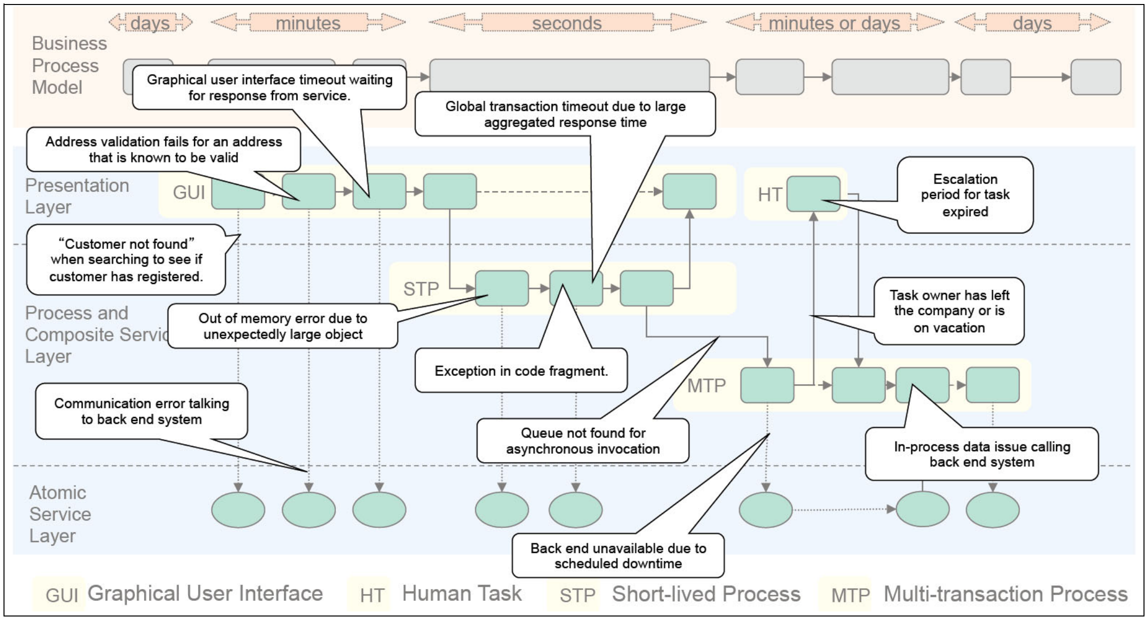
5.6 Error handling

The subject of error handling comes up on every IBM BPM engagement, and should be considered one of the key areas to be carefully designed and implemented. In this section, we describe the main error handling concepts, and cover strategies that can be applied in real life solution implementations:

- Error handling concepts
- Error handling strategies

5.6.1 Error handling concepts

To start, it is important to understand the overall error handling concept. Figure 5-16 provides an example of an abstract process landscape, depicting various application architecture layers. It delivers an idea of what errors can occur where in the process application.



days
minutes
seconds
minutes or days
days
Business Process Model
Presentation Layer
GUI
HT
Process and Composite Servic Layer
STP
MTP
Atomic Service Layer
Graphical user interface timeout waiting for response from service.
Global transaction timeout due to large aggregated response time
Address validation fails for an address that is known to be valid
Escalation period for task expired
"Customer not found" when searching to see if customer has registered.
Task owner has left the company or is on vacation
Out of memory error due to unexpectedly large object
Exception in code fragment.
Communication error talking to back end system
Queue not found for asynchronous invocation
In-process data issue calling back end system
Back end unavailable due to scheduled downtime
GUI Graphical User Interface
HT Human Task
STP Short-lived Process
MTP Multi-transaction Process

Figure 5-16 Abstract process landscape diagram

The following three perspectives matter the most to the design of error handling:

- What kind of errors can occur?

  Rather than look at every possible error, we need to categorize the errors into logical error types, so that we can come up with the smallest possible number of design decisions about how to handle them.

- Where will the errors show up?

  We need to understand how changes in the system state are controlled. More specifically, we need to understand the transactional behavior of the system. The errors end up on the edges of the transactional boundaries.

- How should we manage them?

  The combination of our system, the infrastructure, and the external systems provide us with a myriad of options. We need to define an error handling strategy that is appropriate to the solution.

Next, we take a close look at each of those perspectives individually.

The kind of errors that can occur

Listing errors by their causes doesn't always help to identify patterns in the way that we should handle them. By looking at each cause in detail, however, we begin to see recurring themes in the error type and behavior.

Next, we take a look at examples of different error types, and how they can be categorized.

Application alternative paths

A service call cannot be satisfied in the way that it was requested, for a reason that is understandable at the business logic level. A typical example of such are errors occurring when an entity is not found or the entity already exists. Usually, this error can be summarized as an attempted invalid state transition.

These errors are business meaningful and are known at design time, so there is an option to handle them programmatically, or as part of the surrounding business process. Hoverer, if such errors occurred at run time, they are typically *permanent* in nature.

Application logic bugs

Application logic bugs are errors within the application code that don't cause exceptions, but produce incorrect results. These differ from application exception bugs only in that they are harder to spot, and can only be found through testing or feedback from experienced users. These are typically *permanent* for a specific set of application data. They can only be resolved through a code change and a new release of the application.

Application exception bugs

Exception bugs are errors within application code that result in unplanned exceptions. Poor handing of non-populated data and incorrect creation or parsing of data formats are examples of such errors. These types of errors are typically *permanent* for a specific set of application data. They can only be resolved through a code change and new release of the application.

Application validation errors

Validation errors are returned by the rules engine, such as IBM Operational Decision Manager.

Infrastructure capacity errors

Capacity errors are related to insufficient hard disk space, out of memory errors, exhausting processor capacity, or network bandwidth. Errors are typically *transient*, but can be *permanent*. Such errors are typically only resolvable by upgrading components. Resolution often requires downtime.

Infrastructure downtime errors

Examples of infrastructure downtime errors can be server down, application down, router down, and security service down. These errors include failures of physical components, and therefore also cover disaster recovery. These errors are typically *transient*, but it could be a long time before another try succeeds. Resolution requires logical or physical replacement of a failing component.

Product bugs

These errors occur when the underlying product does not behave as documented. They can only be resolved through a new release or fix pack of product code, or by coding a workaround in application code. Such errors are typically *permanent*, although more rarely they can be *intermittent*.

Analyzing different types of errors helps you classify them into logical categories that further help you to understand what the client's expectations for error handling are.

These same error categories can be presented from a different prospective:

Business errors These occur when an action could not be completed for a reason that is meaningful to the caller. By *meaningful*, we mean that the caller would benefit from the information, and might choose a different course of action. Examples of these errors include *a unique constraint*, *no record found*, and *user not authorized to perform action*.

The error should be at a logical level. It should not refer to specific systems or implementations on the back end. Separate codes should be agreed upon for each meaningful condition. Human readable text should be provided as a part of the error message.

Business rules validation errors can potentially be considered to be a part of this category, because they clearly fall into the definition of meaningful errors that users can be expected to respond to on the UI.

Non-business errors *Transient errors* are defined as a temporary system fault. For example, input/output (I/O) failure, or out of connections or threads. The only sensible action that a caller can take is to possibly try again over a time period appropriate to the caller. The algorithm used to try again should be meaningful to the SLA.

For example, 10 tries with 1-minute intervals would be pointless for a transactional synchronous caller, because the caller would have timed out long before. A single high-level error code should be provided, simply representing that a generic transient error has occurred and the full error logged by the service implementation.

If the back end is known to be unreliable (transient errors are common or the transport is unreliable), it might be necessary to “wrap” the service with attempts to achieve the SLA.

Permanent errors are fundamental errors, occurring in such a manner that the request cannot be processed in its current form. The caller can take no meaningful course of action, and cannot proceed. A typical cause can be a fundamentally incorrect structure or format of the data. A programmatic change is typically required to resolve.

As shown previously, classifying errors by meaningful aspects can provide you with greater insight about how these errors can be handled in a solution, including UI behavior.

Where the errors will show up

Now that we described the types of errors, we look at some of the concepts related to the mechanics of error origins:

- Request-response, blocking transport

  A two-way call with an SLA suited to online users (so the task completes in seconds rather than minutes). All errors are passed back to the caller:

  - Caller can be confident of the end state if a transactional protocol is used.
  - Needs to be transactional for updates.
  - The consumer can initiate a transaction, and the provider can participate.
  - If the communication medium is unreliable, or the consumer is not transactional, there is a possibility for data loss or duplication. In this situation, we lose the benefits of transactionality, because we must implement asynchronous error handling anyway.

- Request-response, non-blocking transport, real time

  A call that involves asynchronous transport, but with an SLA equivalent to synchronous invocation:

  - Most errors are passed back to the caller. If the call succeeds, the caller can be confident of the end state. If the call fails, especially for timeout, the end state can be in doubt. Even brief system outages can result in an in-doubt state.

  - Because this is not a single transaction, there is an opportunity for the caller to be left in an in-doubt state. You must therefore implement an asynchronous error handling pattern to manage the in-doubt state.

- Request-response, non-blocking transport, non-real-time

  A call made over an asynchronous medium whose response is not guaranteed in a time frame appropriate for real-time consumers:

  - Caller can be confident of the delivery of the request. On receipt of a response, success is assured. Any time in-between, the state is unknown. On timeout, the end state is unknown. Transient errors should be rare, because there is time for tries to be made again.

  - The *in-doubt* state should be treated as an expected *in-progress* state, and the surrounding systems should be able to query for, and act upon, that state. All error handling is fundamentally asynchronous.

- Fire and forget, non-blocking transport

  A call made over an asynchronous medium that requires no response:

  - Caller can be confident of delivery of the request.
  - In theory, no error handling is required, but you might need more analysis of requirements and circumstances to make the final judgment about whether a response is not truly necessary.

How to manage errors

Choosing and following error handling strategies is a suggested approach that should be considered for managing error handling in the process solution. The following section provides an overview of the error handling strategies, and some implementation examples specific to the IBM BPM process applications.

5.6.2 Error handling strategies

Keeping to the following agenda is a good way to approach defining an error handling strategy for your IBM BPM project or an overall program:

- Produce a stated preference for the ways that errors are managed if they occur.
- Provide a recommendation to use specific error handling-related design patterns.
- Produce a resultant specific set of coding guidelines that ensure that the strategy is upheld.

A real error handling strategy requires more effort to analyze the client's requirements and overall solution landscape in order to produce the detailed content for the first two items in the previous list. On a project, more than one error handling strategy might be required. For example, it is common to have a different strategy for asynchronous and synchronous interactions, and a specific approach defined for presenting different types of errors on a UI.

The following list describes general guidelines for error handling strategies:

- Where concurrent access to data is probable, all users of the data must be taken into account when considering the locking strategy.
- Solutions should only leave behind failed events that represent a problem to be resolved. Errors that are automatically resolved should be cleaned up. It is also acceptable that they leave an audit trail or log.

Example error handling strategies can be divided into the following categories:

- Strategy for STP

  The strategy assumes no long-lived processes or human tasks on the "happy path" (the ideal version of the process). Rather, consider chained mediations or asynchronously started, briefly persisted processes.

  Error handling would occur offline from the main path in a separate component to the main processing. This might contain long-lived components.

- Strategy for generic offline error handling

  Errors are outsourced to a generic component that enables processes to complete rather than clog up the system. Error handling can be progressively automated, rather than having to be built into all of the processes from the beginning.

  A good implementation example of this strategy in IBM BPM, applicable for both BPMN process and SLA exception handling, is the General Exception Handler toolkit. It is a community-based asset available on the IBM BPM Community Wiki site:

  `http://wiki.bpmwiki.com/display/samples/GEX+Toolkit+%28General+Exception+Handler%29`

- Strategy for failure point resubmission

  This strategy is predominantly relevant to long-running and asynchronous scenarios. All errors are resolved at their failure point, and resubmitted if possible. It works well for ensuring that processes continue to completion wherever possible. All points where errors can occur are provided with a mechanism to store the current error state in a resubmittable form.

- Progressive error handling strategies

  It is not uncommon when a client, due to various reasons, has no input about how their want errors to be handled in the process application. For example, the client can have security-based constraints to provide access to the server-side resources. In such cases, an approach that makes much sense is to follow a progressive error handling strategy.

  You start from the basic approach, which is assuming that all errors are returned to a caller, whether it is a user or a calling system. As the project progresses, more information gets collected, knowledge about the process evolves, and the error handling approach evolves along with it.

  The following list provides a summary of progressive error handling strategies for synchronous transactions:

  Basic All Errors are passed back to the caller.

  Automated Permanent errors are passed back to the caller.

  For transient errors, perform additional tries according to static policy, then behave as for *permanent errors*.

  For business errors, handle according to agreed on static business rules. In case of non-conforming logic, respond as for *permanent errors*.

Autonomic For permanent errors, search for and apply known solutions to data errors. On further failure, behave according to *automated* error handling rules.

For transient errors, perform historically informed additional tries based on similar transactions.

For business errors, make historically informed business decisions where possible. Where no decisions exist, respond as for *permanent errors*.

- Alerting of system errors strategies

  The *alerting of system errors* strategy is a good strategy when it is necessary to proactively alert the system administrator staff about specific errors, or about an abnormal frequency of errors occurring in the logs. When it is used with another strategy (for example, generic offline strategy), it compliments that one with additional capabilities of monitoring log files for the most hazardous error that can potentially occur.

  Usually, this strategy is chosen when clients already own respective software tooling, and want to use them for the IBM BPM logs monitoring. Custom implementation can also be considered as a suitable option if this is a required behavior for an error handling implementation.

As we have shown, the error handling implementation in a business process solution can be quite complex. It is a leading practice to follow these suggestions while approaching the error handling design:

- Error handling discussions in an organization should start with workshops to ensure a common understanding of the key concepts. Error handling discussions should never start with the technology.
- It is important to understand the increased complexity in IBM BPM distributed solutions, integrating with the organization’s established and SOA-based solutions.
- Error handling strategies should be a product of the top-down requirements, further honed by knowledge of the technology, not the other way round.
- Apply leading practices and take advantage of the information, expertise, and existing assets available on the IBM Community wiki site and other IBM online resources.

5.7 Logging

As one of the aspects of runtime system visibility, logging should be considered an important part of every IBM BPM solution. The overall solution visibility aspect is covered in more detail in Chapter 6, “Business-centric visibility” on page 217, and Chapter 7, “Performance and IT-centric visibility” on page 231 of this book.

In this section, we want to describe logging as a stand-alone topic, because often solution architects come from different backgrounds and have various platform skills, and ask the question whether they can use traditional logging tools with IBM BPM.

There are several options that can be used for logging purposes in the IBM BPM process application, including the ones that come with the product.

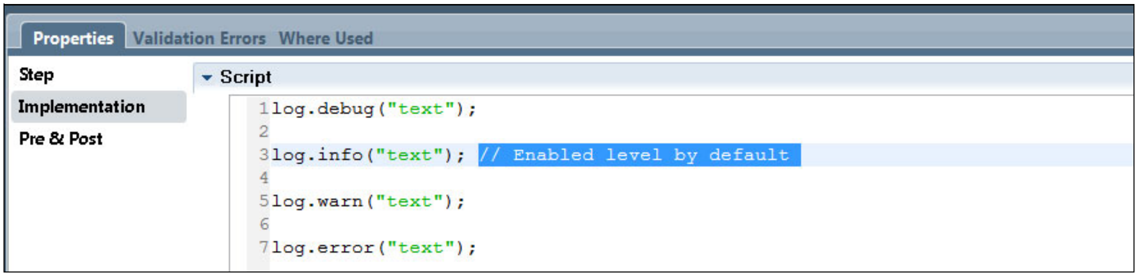
> **Logging and tracing:** It is important to realize the difference between logging and tracing, as described by the following key differentiating points:
>
> - *Logging* can be switched `on` or `off` permanently, and should generally not affect system performance. Logging often operates with `info`, `warn`, and `err` levels, and is implemented as a combination of system input and output configurations.
> - *Tracing* is only switched `on` for detailed diagnosis, showing the exact path through the code. Tracing is used for low-level diagnostics, and typically has a detrimental effect on performance. It is largely systemic (not defined by the implementor).

Before committing to a particular option, it is important to understand from the client requirements perspective what is the specific purpose of logging, and how the logging results are used. More specifically, determine how logging output is going to be accessed, analyzed, and presented. Another aspect that can affect the development scope is whether local logging output is required. In that case, additional design and development effort is required to support local output.

The following sections describe options that you can consider for various types of logging within IBM BPM.

IBM BPM Standard default logging capabilities

Typical use is calling the **`log.error()`** command in the Server script asset in IBM Process Designer, as shown in Figure 5-17.



```
log.debug("text");

log.info("text"); // Enabled level by default

log.warn("text");

log.error("text");
```

Figure 5-17 Calling the log.error() command

The appropriate logging level can be set in the WebSphere Application Server Administrative Console, as described in the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/ttrb_configjavalog.html?lang=en

Using an alternative logging mechanism in IBM BPM

Use of the *Jakarta Common Logging* (JCL) API is covered in the following technote on the IBM developerWorks website:

http://www.ibm.com/developerworks/websphere/techjournal/0901_supauth/0901_supauth.html#icomments

The *Apache Log4j framework* (Log4j) can be configured as an underlying logging component in JCL if specific requirements for logging exist that cannot be met with IBM BPM standard logging tools.

Alternatively, Log4j can be implemented in the IBM BPM Standard configuration, using the Log4j binary files included with IBM BPM, or by importing a specific version of the Log4J binary files. A complete Log4j 1.x implementation example with runtime configuration option is available as a community asset on the IBM BPM Community Wiki:

http://wiki.bpmwiki.com/display/commwiki/Custom+logging+using+Log4j+in+IBM+BPM+7.5.x+and+8

For typical performance considerations and recommendations while using logging in the IBM BPM process applications, see 7.2.1, “Logging” on page 234.

5.8 Asset maintenance and governance considerations

Considering the fact that the vast majority of the lifetime of process assets lies in a post-development phase, asset maintenance becomes an increasingly important aspect of the process lifecycle. Governance around deployment and source code maintenance are other integral parts of the overall solution that require careful consideration and planning.

In this section, we cover common aspects in process asset management and governance:

- Shared assets in IBM BPM Process Center
- Managing IBM Process Designer artifacts
- Managing advanced artifacts
- Deployment governance
- Managing tracks

5.8.1 Shared assets in IBM BPM Process Center

The IBM BPM Process Center (Process Center) includes a repository for all processes, services, and other assets created in IBM Process Designer and IBM Integration Designer. For geographically distributed teams, collaborative development can become challenging as team size, and the distance between the teams, increases. Poor performance can be the result of network latency and disperse topologies induced network failures.

Starting with the version 8.0 release, IBM BPM enables you to federate remote process centers through the Process Center self-registering mechanism. When you register two Process Centers with each other, you can share toolkits with other users, or subscribe to toolkits that other users share.

You can share toolkits with other users, even if those users are working in a different Process Center. You no longer need to first export the toolkits from one Process Center and then import them into another Process Center.

After a toolkit is shared, events are synchronized, and snapshots are released automatically once every 24 hours. The sharing Process Center is notified when a shared toolkit is being used on a registered Process Center. The subscribing Process Center is notified of new released snapshots and of the state of the shared toolkits.

For more information about registering Process Centers and sharing toolkits, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.admin.doc/topics/sharing_pa_toolkit.html?lang=en

Table 5-3 describes the specific rules that you must consider when you share toolkits.

Table 5-3 Rules for sharing a toolkit

Action	Rule
Sharing a toolkit	► There is at least one active registration to a remote Process Center. ► The toolkit has dependent toolkits, and they are all marked for sharing. ► The toolkit has at least one released snapshot. ► If one or more toolkits are dependencies of a parent toolkit, all dependent toolkits must be shared before sharing the parent toolkit. ► If one or more snapshots are dependencies of a parent snapshot, all dependent snapshots must be released before releasing the parent snapshot.
Stop sharing a toolkit	► The toolkit does not participate in a dependency. ► The toolkit does not have a parent toolkit that is marked as Shared.
Sharing a toolkit when the snapshot status is set to Released	► The toolkit does not have child dependencies. ► The toolkit has dependencies, but all snapshots that are used from children are set to Released.
Sharing a toolkit when the snapshot status is changed from Released to another status	► The toolkit is not marked as Shared. ► The toolkit contains at least one other snapshot that is marked as Released. ► The toolkit does not participate in dependencies. ► The toolkit participates in a dependency, but none of the parent snapshots are marked as Released.

For more information about rules for sharing toolkits, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.admin.doc/topics/rrules_tk.html?lang=en

Sharing process artifacts encapsulated in toolkits provides an opportunity for a truly distributed form of development.

Figure 5-18 provides an example of a distributed development environment across multiple geographically dispersed Process Centers.

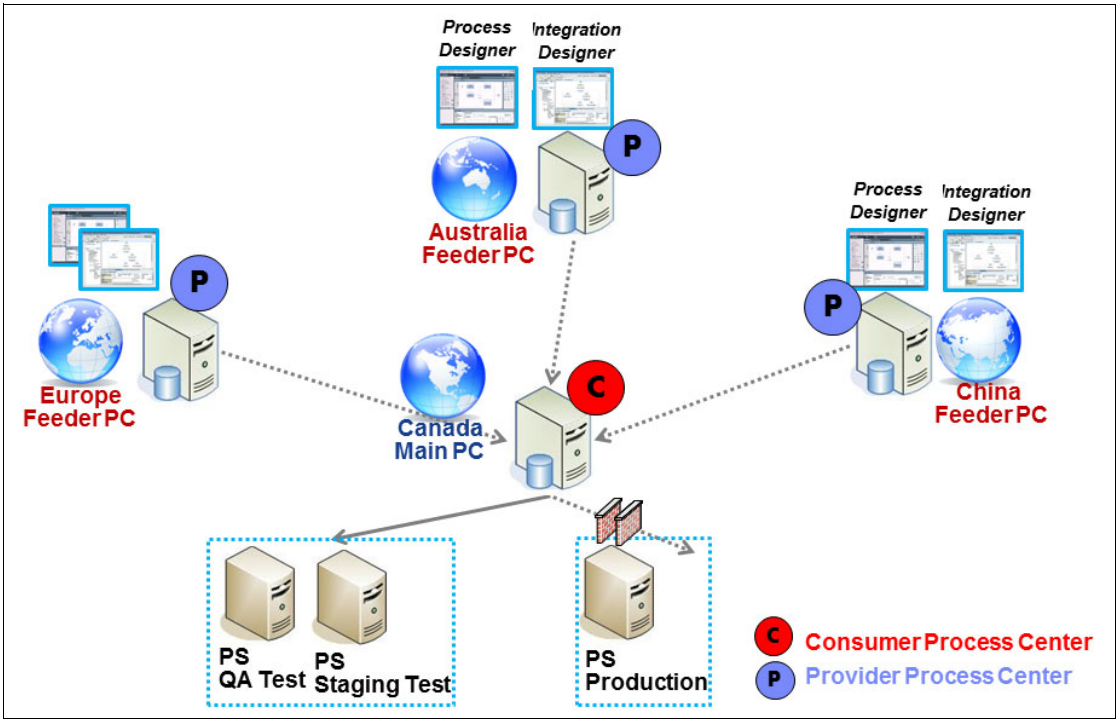


Figure 5-18 Distributed development Process Centers environment

When this form of development environment is selected, you must carefully design your toolkits and plan for a shared distributed development environment. Each *provider* Process Center supplies the *consumer* Process Center with a self-contained piece of functionality that should be fully tested.

The functionality should be ready for integration testing with the rest of the parts of an overall process solution. This approach further encourages component design and code reuse, following leading practices in software design.

For more information about toolkit design considerations, see 5.5, “Toolkit design” on page 188.

When working with process applications or toolkits, you might need to provide a link to related information that is available outside of the IBM BPM environment. For example, you might want to link to a website or a wiki page. You can also reference a change request stored in a change management system, or a test case in a quality management system.

These kinds of links can be used to achieve traceability or provide details about the fixes and features that went into a new asset. IBM BPM uses the Open Services for Lifecycle Collaboration (OSLC) standard, which enables linking across products for the purposes of lifecycle management and traceability.

For more information about using remote assets in IBM Process Designer, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.admin.doc/topics/cusing_remoteassets.html?lang=en

5.8.2 Managing IBM Process Designer artifacts

Users in IBM Process Designer create process models, services, and other assets in process applications and toolkits. IBM Process Designer is an easy-to-use, graphics-oriented tool that enables you to model and implement your BPMN-based business processes. You can use it to easily demonstrate process design and functionality during development efforts.

All development artifacts created in IBM Process Designer get managed by the Process Center, and get implicitly persisted in the Process Center repository. No external source control tool is necessary for assets created in IBM Process Designer. No additional effort for persisting and managing process applications and toolkits is required, because the Process Center does it for you transparently.

Multiple users can simultaneously access and change process applications and library items in IBM Process Designer. When you edit concurrently, you collaborate with other team members to create the library items that you need for your project. For example, you can communicate about your ideas and edits using instant messaging, and see the results in the Designer view as they happen.

When multiple users work on the same library item, such as a human service, each user can see the changes when edits are saved. To ensure that all users are aware of which library items are open and what changes are being made, IBM Process Designer provides the following notifications:

- Notifies you when another user opens a library item, by showing a user icon. You can hover over the icon to see who that user is.
- Notifies you when another user is editing a library item, by displaying the words Read Only next to the library item. When users save their work, the library item is available to edit.

- Notifies you when another user has saved changes while you are editing a library item, by displaying the words Read Only next to the library item. When you click **Save**, a Save conflict message displays. You are prompted either to save your changes and override the other user's changes, or discard the changes and accept the other user's changes.
- Notifies you when multiple users start to edit a library item at the same time, before the Read Only text appears, by displaying a warning icon and message. The message prompts each user to either immediately save their changes or discard them.

You can enable IBM Process Designer users to share library items across process applications. Process applications can share library items from one or more toolkits, and toolkits can share library items from other toolkits.

Users who have access to a toolkit can create a dependency on the toolkit, and use the library items within it for their process development efforts. See 5.8.1, "Shared assets in IBM BPM Process Center" on page 205 for more information about sharing library items across Process Centers.

For more information about an overall Process Center repository management, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.admin.doc/topics/managinglib_introduction.html?lang=en

The IBM Process Designer view offers several tools to ensure that you can access items that you work with regularly. With IBM Process Designer, you can move or copy items between existing or new process applications and toolkits. You can also revert to previous versions of individual library items using snapshots.

We advise you to organize the development artifacts in smart folders, and take advantage of tagging capabilities in IBM Process Designer. As practice shows, using these organizational features of IBM Process Designer increases the productivity of development teams, and improves code maintenance. Improvement in code maintenance is especially visible, and material for the long-term projects with periodic team member rotations.

For more information about managing library items in IBM Process Designer, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.admin.doc/topics/managing_lib_items.html?lang=en

5.8.3 Managing advanced artifacts

IBM Integration Designer provides editors that enable developers to create complex automated business processes and business services. Developers can use available assets, such as medication modules, SCA modules, Extensible Markup Language (XML) Schema Definitions (XSDs), WSDLs, Java archive (JAR) files, and so on. Unlike IBM Process Designer, the IBM Integration Designer does not need to be connected to the Process Center at all times.

Artifacts developed in IBM Integration Designer are typically retained in an Eclipse workspace that is local to the server hosting the IBM Integration Designer. Developers can choose to make the artifacts available to, or refrain from publishing them to, the Process Center:

- Associate these artifacts with the process application for use with the Process Center. In such situations, an initial connection with the Process Center is needed. This connection enables you to retrieve the process application from the Process Center, retrieve updates made by other developers, or publish local workspace updates to the Process Center, as shown in Figure 5-19.

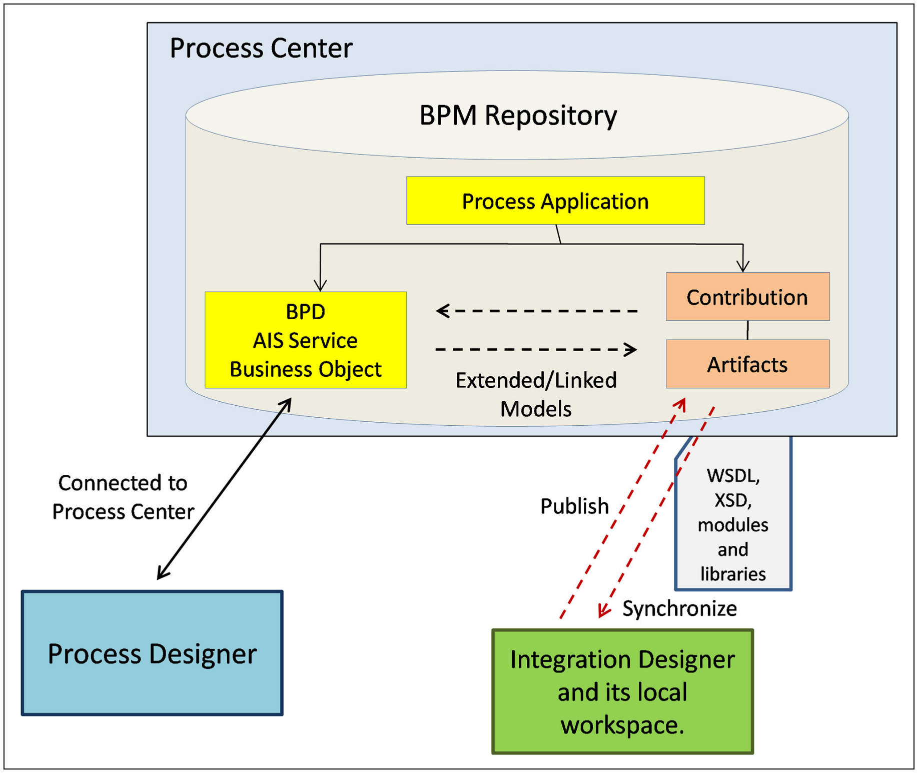


Figure 5-19 Management of the IBM Integration Designer artifacts

These artifacts can be deployed to the IBM Process Server with the process application. An example of such an artifact is an AIS implemented in IBM Integration Designer and published in the Process Center for consumption by activities defined in the business process diagram. For details about how to synchronize IBM Integration Designer with the Process Center, see the following IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFTDH_8.5.5/com.ibm.wbpm.auth.stp.doc/processcenter/topics/cprocesscenter.html

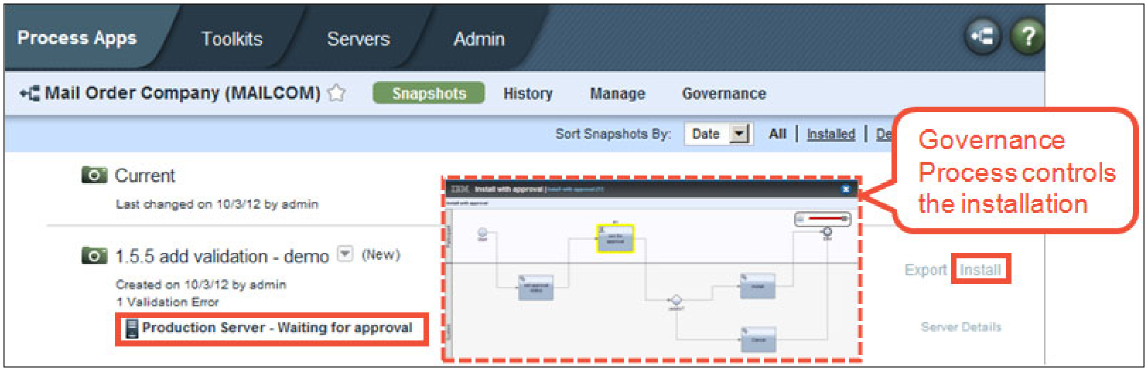
- Refrain from publishing these artifacts to the Process Center. In such situations, the modules and libraries implemented in IBM Integration Designer can be managed using an external source control system, and deployed directly to the IBM Process Server. For more information about deploying assets to the IBM Process Server, see the following IBM developerWorks article:

  http://www.ibm.com/developerworks/bpm/bpmjournal/1212_iyengar/1212_iyengar.html

5.8.4 Deployment governance

In IBM BPM V8.0, a new governance process feature was released that enables you to follow a custom governance process to change the status of a snapshot, or to enable the deployment into any process server environment (online and offline).

Figure 5-20 illustrates the Process Center with a process application that has the status of `Waiting for approval`. When approved, a snapshot is installed.



Governance Process controls the installation

Figure 5-20 Governance process

The process that gets executed implements a template that is provided by the System Governance toolkit, which comes with IBM BPM. The governance process itself can have any kind of flow patterns, human tasks, rules, or any other BPM artifact. when completed, the snapshot gets deployed. Having a governance process in place and formalized in IBM BPM has the great advantage that you can ensure that all necessary preconditions are met.

Consider the following aspects of a governance process:

- Release Notes

  A list of new features and fixed defects that describe the new version.

- Instructions

  Instructions for configurations or installations that must take place before the installation of a snapshot, such as setting up groups, databases, or firewall configurations.

- Test Results (number of open defects, test cases, and so on)

  Test statistics that help an approver to gain an overview of the quality of the new snapshot (which could be a new release).

- Approvals

  Route the tasks to all necessary approvers. These can be, for example, Process Owners, Project Managers, Infrastructure Division, Support and Maintenance, and so on.

- Other deployment artifacts

  The deployment process can make sure that all of the artifacts that are required for a successful installation are created and delivered. Examples of these artifacts can include property files, JAR file dependencies, information to create required database schemas, and so on.

The governance process will be executed whether you deploy online from a connected process center environment or chose to have an offline deployment.

Remember: The governance process works for both *online* and *offline* deployment models.

5.8.5 Managing tracks

A Process Center can create different tracks for each process application. A track is a duplicate of a process application (similar to a snapshot) that can be edited. There is always one main track, which marks the default.

In addition to this, there can be many non-main tracks. All tracks (default and non-default) run under the umbrella of the process application. Therefore, they are not true duplicates.

When a track is the default track, the library items within it run by default when an event or other trigger that applies to items in more than one track is received by the Process Center server. For example, when you are executing a service using a URL, and that service exists in more than one track, the service in the default track is run.

When to use tracks?

Tracks can be used similarly to branching in traditional version management systems. The following use cases are common:

- Tracks for release support

  As you go into production, you might see the need to create branches that can be used in the following ways:

  - To support the production release
  - To continue development on the main code branch

- Tracks for continuous improvement

  Because the purpose of BPM initiatives is to continuously improve the business, tracks are often used so that business analysts (BAs) can focus on improving the process flow based on data gathered during run time.

  This can, for example, happen through the use of the optimizer, but also other process improvement tools and methodologies. By creating a separate track, the BA has all necessary freedom to change the process, and eventually copy it back into the main track. The BA can do this without causing any unwanted side effects of changing it in parallel to the development.

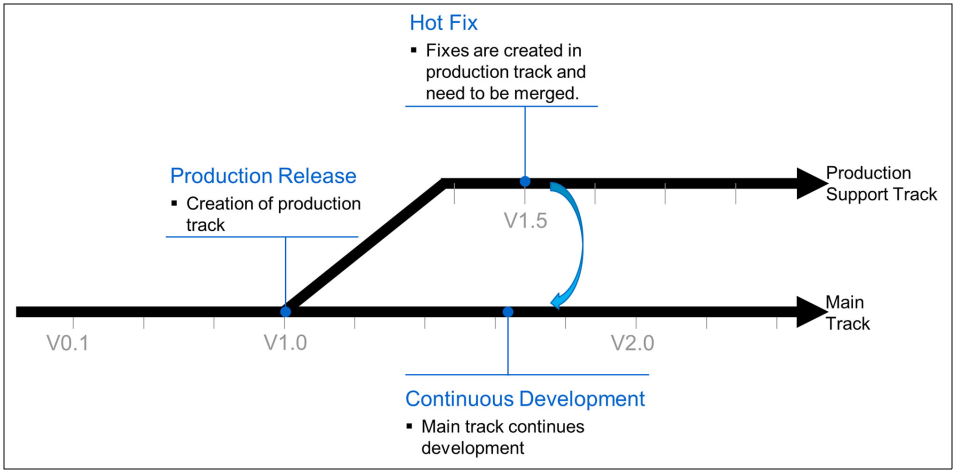
> **Restriction:** IBM BPM has limited functionality to merge contents of different tracks, or to show the differences (changes) between them. Therefore, it requires mature governance procedures to manage artifact changes and control their merge back into the main track.

When not to use tracks?

If you create a snapshot that is being used for testing or demonstrations, you usually don’t want to use tracks to manage this snapshot. The value of tracks can be reduced by the resources required to manage them and their functionality, and to ensure that changes are properly propagated among all necessary tracks and environments. Try to *keep the number of tracks as low as possible*. The most common usage for tracks is to support a production release.

Track-based branching and merging

As mentioned previously, tracks are a good tool to manage changes, such as hot fixes for releases. Figure 5-21 illustrates how a base version can branch into a production support track while continuing to be developed in the main track.



Hot Fix
▪ Fixes are created in production track and need to be merged.
Production Release
▪ Creation of production track
Production Support Track
V1.5
Main Track
V0.1
V1.0
V2.0
Continuous Development
▪ Main track continues development

Figure 5-21 Managing changes with tracks

V1.0 marks the snapshot that will be deployed into production. At this point, it makes sense to create a track for this. In Figure 5-21, this track is called *Production Support Track*.

As the Production Support Track goes into production, on the *Main Track* the development continues as normal, and further versions or releases (*V2.0*) are being created.

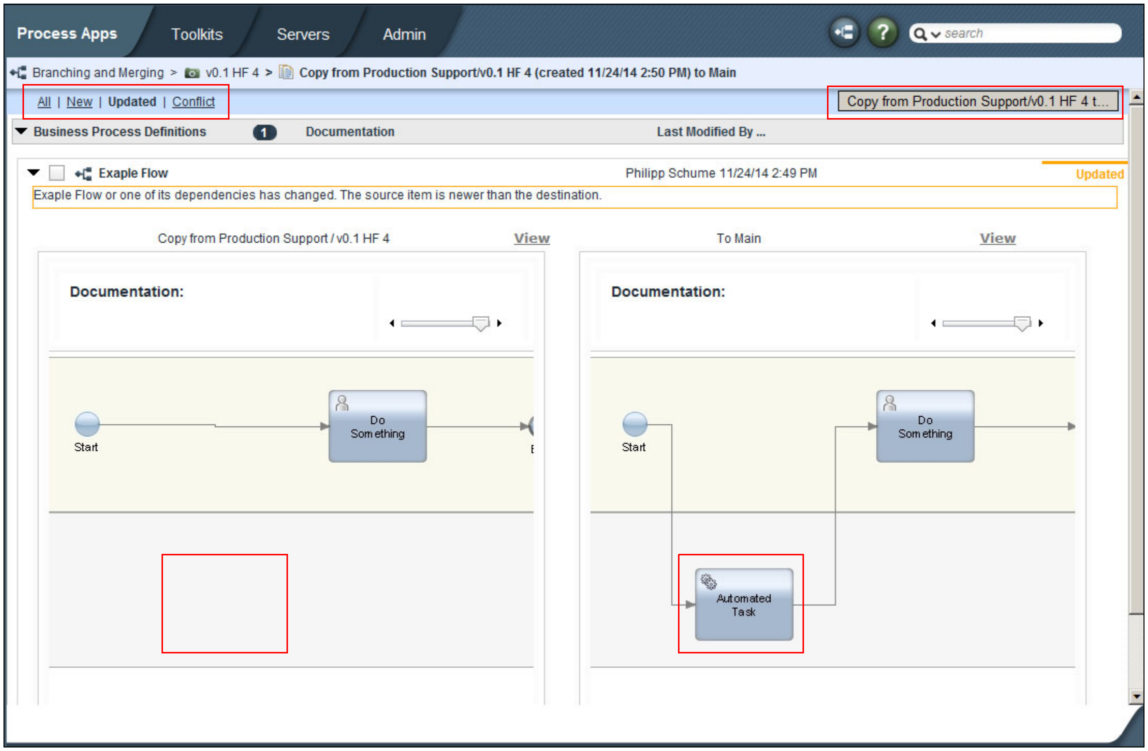
Generally, the release cycles should be short enough that required changes or fixes can be deployed directly from the main track into production (of course, while respecting all required quality assurance phases). However, on occasion, there might be a requirement for an *interim fix* to the production environment.

An interim fix is a high-effect fix that is required to keep the currently deployed version usable. This fix can be created in the Production Support Track, because no other changes, such as the ones that might already be part of the continuous development track, must be deployed yet. After the fix is created, it has to be merged into the main track. IBM BPM provides a set of capabilities to support this, which leaves us with two approaches for merging, *automated* and *manual*:

- Automated

  The IBM BPM process center provides the ability to compare tracks, visualize the changes, and merge changes between track versions.

Figure 5-22 shows the Process Center view that highlights whether there were conflicts or new items.


Process Apps
Toolkits
Servers
Admin
search
Branching and Merging > v0.1 HF 4 > Copy from Production Support/v0.1 HF 4 (created 11/24/14 2:50 PM) to Main
All | New | Updated | Conflict
Copy from Production Support/v0.1 HF 4 t...
Business Process Definitions
1
Documentation
Last Modified By ...
Exaple Flow
Philipp Schume 11/24/14 2:49 PM
Updated
Copy from Production Support / v0.1 HF 4
View
To Main
View
Documentation:
Start
Do Something
Documentation:
Start
Do Something
Automated Task

Exaple Flow or one of its dependencies has changed. The source item is newer than the destination.

Figure 5-22 Process Center copy between tracks

On the upper left corner, Process Center provides you the option to filter by All, New, Updated, and Conflict. The list of items is highlighted as such. In Figure 5-22, you see the orange highlighting of *updates*. If you open the list item, you see a visual representation of the changes. In this case, an activity has been deleted (on the diagram on the left side).

Now you can decide whether you want to copy this change. On the upper right corner, there is a button that you can click to proceed with this operation.

- Manual

  A very common approach is to manage the merge of changes in a rather manual fashion.

  Capture all details for the fix that you created:

  - What was the root cause of the problem?
  - What changes have been applied?
  - Where (which artifacts) have changes been applied?
  - Any recommendation (if different) about how to fix the problem in the main track.

  The ticket with all of the fix-related information is then added to the backlog of requirements for the main track. The development team estimates and schedules the development of the fix then accordingly.

5.9 Conclusion

In this chapter, we described the difference between processes and services, and what to remember when creating either of them. We talked about routing patterns and the difference between pushing and pulling tasks for user execution.

We also introduced four different types of data flow patterns:

- Process
- Service
- Coach
- Integration

We described how to manage the development environment and code base through proper design of toolkits and error handling concepts. We also how to manage them using other features, such as shared Process Center assets, deployment governance, or tracks.

6

Business-centric visibility

This chapter describes the concepts of business-centric visibility, and describes how business-centric visibility is achieved in IBM Business Process Manager (IBM BPM).

This chapter includes the following sections:

- What business-centric visibility is
- Business-centric visibility in IBM BPM

6.1 What business-centric visibility is

Business-centric visibility is one of the most important considerations for companies looking to adopt the business process management (BPM) discipline, because it offers the following benefits in context of a business process:

- Ability to view the business data to make smarter business decisions through proper exposure of business data to business participants and process owners. Business data is specific to the company or companies involved in a business process, and can include the following examples:

  - Type of service requested by a potential client
  - Type of client
  - Client's billing and service address

  Typically, business data is in a system of record (SOR) outside the BPM software. Business data can be particularly useful to process owners and process participants. For example, process owners and process participants can view the business data associated with outstanding tasks, and make business decisions based on the available information.

- Ability to view the process data to get a high-level performance overview of a business process, to make improvements to a business process, and to audit business processes to identify any violations to the stated business policies. The following list includes some examples of process data:

  - Time taken to complete a task or an activity
  - Time spent on a particular activity in the process
  - Average time to complete a process

  Process performance data is also tied to the key performance indicators (KPIs) and service level agreements (SLAs) associated with a business process. Both of these are commonly overlooked dimensions of the process performance data. Process data is captured by the BPM system, and is typically in one or more data sources that are internal to that system.

  This information can be particularly useful to executives, process owners, and process analysts:

  - Executives that are not involved in the day-to-day running of a business process can use this process data to get a high-level performance summary of a business process.

  - Process owners can use the process data to track process task cycle time, high priority tasks, assigned task owners, process participant performance, and so on. Knowledge of this process data can be particularly useful in improving client experience and maximizing staff use.

– Process analysts can use the process data, such as KPIs and SLAs, to make improvements to a business process to meet and exceed the process performance targets. They can use this process performance information to make tangible process improvements to the process. It is important to state that the process analysts must not make any process improvements without first consulting with the process owner.

To achieve any meaningful business-centric visibility, it is important to have access to both the business data and the process data.

6.2 Business-centric visibility in IBM BPM

In IBM BPM, the business data is typically retained in one or more external SORs. It is represented as business objects in a business process definition (BPD) contained in the process application. The process data is captured by configuring tracking groups, KPIs, SLAs, and timing intervals. It includes statistics about the performance of the BPD, such as KPIs, SLAs, and team performance[1].

This process data can be generated from a runtime instance of a business process, and stored in both IBM BPM Process Server and Performance Data Warehouse components. The Process Server retains inflight process data, and the Performance Data Warehouse retains the historical process data. Historical process data is collected by the Process Server and sent to the Performance Data Warehouse. Business-centric visibility can be achieved in IBM BPM:

- Business data and process data reports
- IBM BPM Process Optimizer
- IBM Business Monitor[2].

6.2.1 Business data and process data reports

The business data and the process data reports, called *dashboards* in IBM BPM, provide business-centric visibility through the business data and the process data that is generated, captured, and retrieved in IBM BPM.

For more information about how to collect and use process performance data, see the IBM BPM 8.5.5 Knowledge Center on the following website:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.wle.editor.doc/topics/tpd_designtracking.html

[1] An initial effort must be carried out to identify the process data that must be tracked, and to ensure that the tracked data offers business value.

[2] It is important to note that the use of IBM Business Monitor can require additional licenses.

IBM BPM provides the following capabilities:

- Ready-to-use process data dashboards that provide real-time performance reports in the IBM Process Portal. Examples of such dashboards include My Performance, Team Performance, Process Performance, and My SLA overview for the SLAs that have been defined and made available. For more information about these ready-to-use dashboards in IBM BPM, see the IBM BPM 8.5.5 Knowledge Center on the following website:

  http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.wle.widget.doc/topics/cport_dashboards.html?cp=SSFPJS_8.5.5&lang=en

  The information presented in these reports is retained in the Process Server database.

- Ability to create dynamic ad hoc reports in the IBM Process Portal beyond the standard dashboards. The information required for these reports can be derived from the tracked data that is retained in the Performance Data Warehouse database. For more information about creating ad hoc reports in IBM Process Portal, see the IBM BPM 8.5.5. Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.wle.widget.doc/topics/tport_adhocreports.html?lang=en

- Ability to develop custom reports using Coaches with the help of toolkits such as the Dojo toolkit or the Dashboard toolkit. For more information about the Dashboard toolkit, see the IBM BPM 8.5.5 Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.ref.doc/topics/rdashboardtoolkitcontrolscontainer.html?cp=SSFPJS_8.5.5&lang=en

  These reports enable process participants to make informed decisions about the process tasks. The data for these reports typically comes from sources that are internal *and* external to the IBM BPM.

- Ability to create custom dashboards using historical and inflight process data retained in the IBM BPM. Using these dashboards, developers can build a framework of custom reports and charts that together provide meaningful process values. Developers can also choose to use the Dashboard toolkit to implement such reports. For more information about creating custom dashboards in IBM BPM, see the IBM BPM 8.5.5 Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.admin.doc/topics/t_creatingacustomworkdashboard.html?lang=en

These reports in IBM BPM can be used to get a high-level overview of the business process, including but not limited to the following information:

- Process performance
- Team performance
- Task status

The following subsections describe the steps to create reports from external and internal data sources:

- High-level steps to create custom reports that display data retrieved from *external* data sources in IBM BPM. We also describe important points to consider before implementing these reports in IBM BPM.
- High-level steps to create custom reports that display data retrieved from data sources that are *internal* to IBM BPM. We also describe important points to consider if you have a requirement to retain large volumes of process data in the Performance Data Warehouse database.

Custom reports using external data sources

It is important to note that IBM BPM is *not* a reporting tool for implementing complex reports involving large volumes of application data retrieved from an external data source. We suggest that you implement such reports outside of IBM BPM, typically in a data warehousing environment in your organization.

Consider implementing such reports in IBM BPM when they meet the following criteria:

- They present vital application data information that is critical to making quick and informed decisions in a business process task.
- They require joining of the business data and the inflight process data retained in the Process Server database.

These custom reports involve presenting the application data, retained in an external SOR, to the process owners and the process participants in the IBM Process Portal. Before developing these custom reports, it is important to analyze the external SOR to understand how the information is stored, and how it is retrieved and presented to the process owners and the process participants.

Figure 6-1 depicts the high-level steps to implement these reports in IBM BPM.

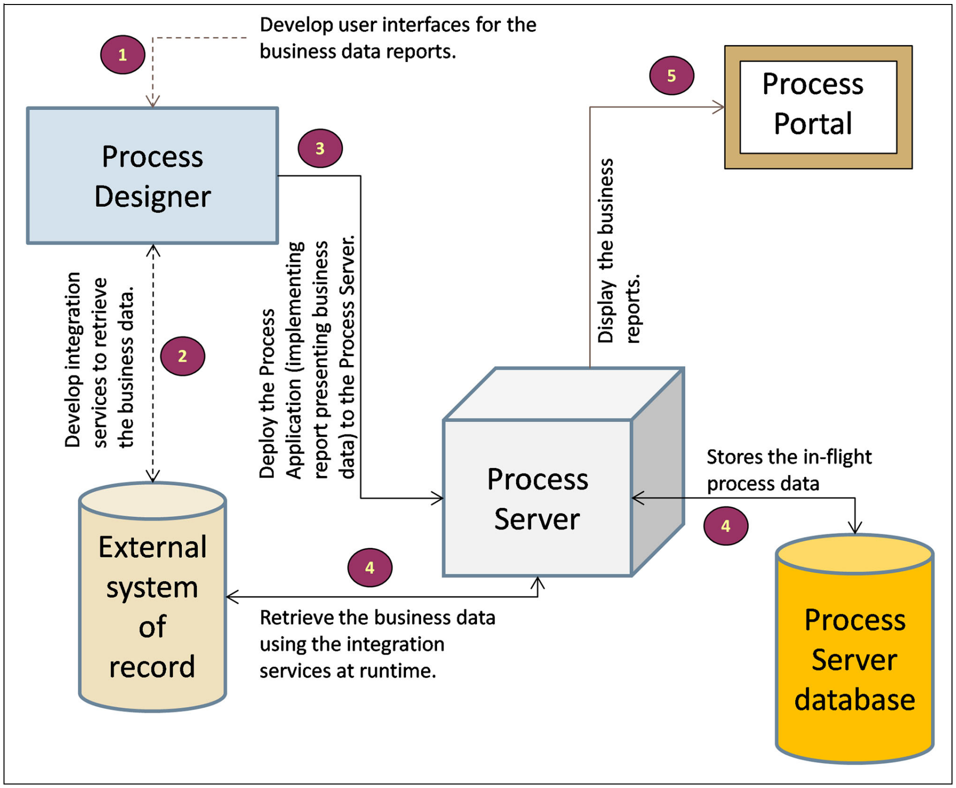

1
Develop user interfaces for the business data reports.
5
Process Portal
Process Designer
3
Develop integration services to retrieve the business data.
2
Deploy the Process Application (implementing report presenting business data) to the Process Server.
Display the business reports.
Process Server
Stores the in-flight process data
4
External system of record
4
Retrieve the business data using the integration services at runtime.
Process Server database

Figure 6-1 Custom reports using external data sources

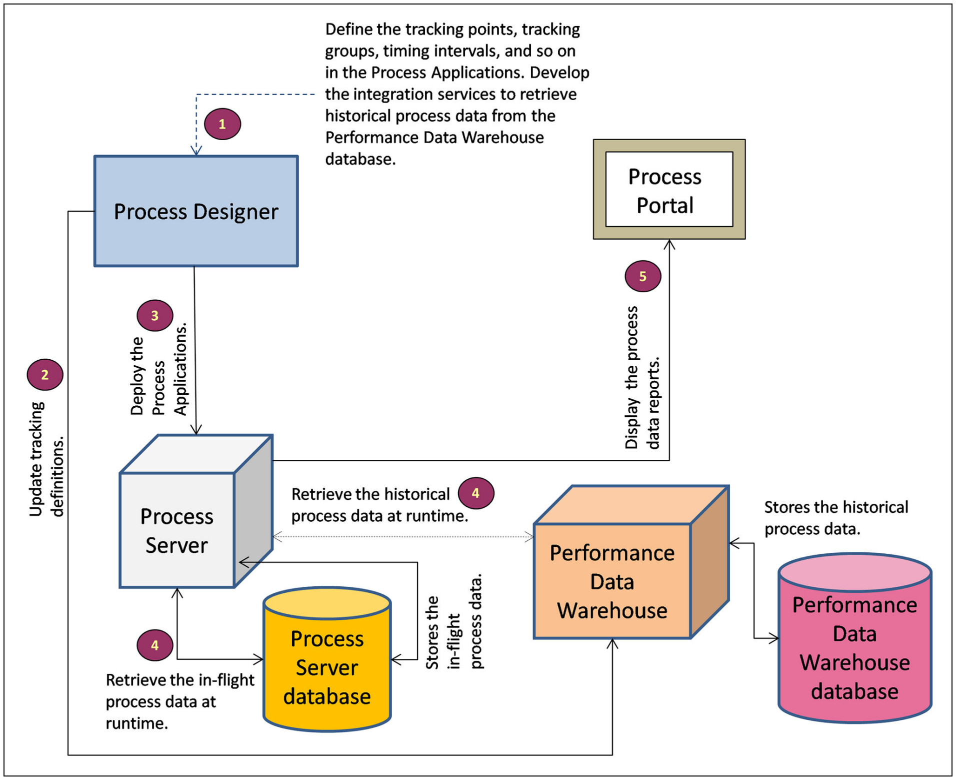
Next, we examine this flow in more detail:

1. Design and develop the user interfaces (UIs) for these reports in the process application using the IBM Process Designer. These UIs are typically implemented using Coach views in Coach designer Coaches.
2. Design and develop the integration services in the process application using the IBM Process Designer, to retrieve the application data from the external SORs. The integration service is typically implemented using Structured Query Language (SQL) queries, stored procedures, SOAP web services (SOAPWS), or Representational State Transfer web services (RESTWS).
3. Deploy the process application containing these reports to the Process Server.

4. At run time, when these reports are started by the business users or the process participants in the IBM Process Portal, the application data required for these reports is retrieved from the external SOR using the integration services. This data can also be combined with the inflight data (retained in the Process Server database) retrieved using application programming interfaces (APIs) defined in IBM BPM.
5. Present the data retrieved in the previous step (step 4) in the UIs displayed in IBM Process Portal.

Custom reports using IBM BPM-generated process data

These reports involve presenting the inflight process data and the historical process data in the IBM Process Portal. Figure 6-2 depicts high-level steps to implement these reports in IBM BPM.



Define the tracking points, tracking groups, timing intervals, and so on in the Process Applications. Develop the integration services to retrieve historical process data from the Performance Data Warehouse database.
1
Process Designer
Process Portal
5
3
Deploy the Process Applications.
2
Update tracking definitions.
Display the process data reports.
Process Server
Retrieve the historical process data at runtime.
4
Performance Data Warehouse
Stores the historical process data.
Performance Data Warehouse database
Process Server database
Stores the in-flight process data.
4
Retrieve the in-flight process data at runtime.

Figure 6-2 Custom reports using IBM BPM-generated process data

To create reports using internal data from IBM BPM, follow these steps:

1. Define the tracking points, tracking groups, timing intervals, and so on in the process application using the IBM Process Designer. Carefully review the potential performance effect outlined in the IBM Redbooks publication, *IBM Business Process Manager V8.5 Performance Tuning and Best Practices*, SG24-8216, before enabling auto-tracking in your business process diagrams.

   Develop the UIs for these reports in the process application using IBM Process Designer, and develop the integration services to retrieve the historical process data from the Performance Data Warehouse database.
2. Update the tracking definitions in IBM Process Designer. Updating the tracking definitions sends the new tracking requirements to the Performance Data Warehouse of the runtime Process Server environment.
3. Deploy the process application containing the tracking points and the tracking groups to the Process Server. At run time, the process data is generated by executed process instances, captured by the Process Server, and retained in the Performance Data Warehouse database for retrieval.
4. At run time, when the executives, the process owners, or the process analysts start these reports in IBM Process Portal, the process data required for these reports is retrieved from the Process Server database (inflight data) and the Performance Data Warehouse database (historical data).
5. Present the data retrieved from the previous step (step 4) in the UIs displayed in IBM Process Portal.

It is important to *carefully consider the amount of process data that is being tracked and persisted in the Performance Data Warehouse database.* We suggest that you carefully consider the business value that you get from the data that you plan on tracking. For example, consider enabling auto-tracking only if you see significant value in doing so. Persisting excessive or unnecessary process data information can lead to performance degradation under significant loads.

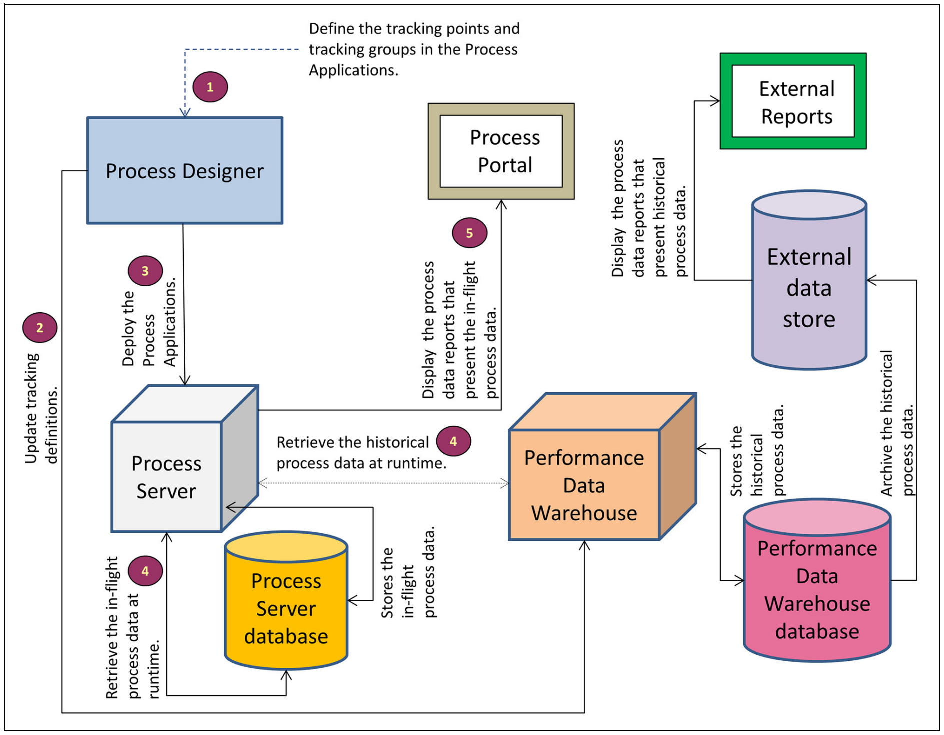
Consider the following points if you need to retain large volumes of process data:

- Manually archive the process data from the Performance Data Warehouse database to an external database[3] regularly. The frequency of the process data archives is typically driven by your business requirements. This approach provides the following advantages:
  - Requests to retrieve the historical process data can be offloaded to an external data store, as shown in Figure 6-3 on page 225.
  - Reports to display the historical process data from an external data source can be hosted outside IBM BPM.

[3] Approach can typically involve additional license, design, implementation, and operational costs.

- Define a purging strategy for the Performance Data Warehouse database. Regularly purging the process data can lead to better response time on process data retrieval requests made to the Performance Data Warehouse database. For information about purging data in Performance Data Warehouse database, see the following IBM developerWorks article:

  http://www.ibm.com/developerworks/bpm/bpmjournal/1312_spriet/1312_spriet.html



Define the tracking points and tracking groups in the Process Applications.
1
Process Designer
Process Portal
External Reports
2
Update tracking definitions.
3
Deploy the Process Applications.
5
Display the process data reports that present the in-flight process data.
Display the process data reports that present historical process data.
External data store
Process Server
Retrieve the historical process data at runtime.
4
Performance Data Warehouse
Stores the historical process data.
Archive the historical process data.
4
Retrieve the in-flight process data at runtime.
Process Server database
Stores the in-flight process data.
Performance Data Warehouse database

Figure 6-3 Custom reports using process data archived to an external data source

6.2.2 IBM BPM Process Optimizer

Often it is assumed that the bottlenecks in a business process can be addressed reactively, by either allocating more time to accomplish the required tasks, or allocating more resources to perform these tasks. In many situations, though, neither solution is a viable option. Knowing how to address bottlenecks, or more formally, optimize your business processes, is a complicated effort.

If you add or remove steps, streamline individual steps, or make organizational changes, every step in the process and every path taken by a process instance can be affected. Business analysts, business process owners, and business process participants need the ability to combine process data in the context of diverse scenarios.

They need the ability to perform path and segment analyses of their processes, and then optimize accordingly. To accomplish this, they need to be able to bring rich performance and business data (beyond just, for example, average task duration) into the context of their business process model.

Bringing this data into context is what IBM BPM Process Optimizer is designed to do. IBM BPM Process Optimizer helps the business analysts to improve processes by showing where and how to change the process model. The IBM BPM Process Optimizer enables the business analysts to perform a wide range of analysis and optimization scenarios.

These scenarios range from simple simulations to validate the overall process modeling strategy, to advanced what-if comparative analyses using historical, inflight, or simulated data (or any combination of the three). IBM BPM Process Optimizer also gives the business analysts the ability to analyze KPIs and SLAs tracked and stored from a process instance at run time.

In the following sections, we describe some examples of how the IBM BPM Process Optimizer can be used to drive valuable process and business insight:

- Simulation of a business process
- Optimization analysis using historical performance data
- Optimization analysis using inflight performance data
- Guided optimization

Simulation of a business process

Process models can be directly simulated using the IBM BPM Process Optimizer. Simulations are driven by user-provided parameters:

- Time distribution
- Resource consumption
- Activity costs
- Branching probabilities
- Resource constraints

The resulting simulation reports provide detailed predictions of process costs, durations, and resource consumption based on simulation results. Different simulation scenarios can be compared side-by-side on a single screen, or exported for offline analysis or sharing.

Optimization analysis using historical performance data

Simulation is a powerful mechanism to identify potential bottlenecks and potential cost reductions, to cost-justify resource allocations based on expected volumes, and to optimize process designs. However, the fundamental flaw in any process simulation of meaningful complexity is that all of the properties that need to be completed are often based on assumptions rather than real-world data.

The outcome is a simulation model and results that are themselves only best guesses at possible performance. IBM BPM provides closed-loop simulation that can automatically populate the property sheets with real-world performance data on the existing process from the Performance Data Warehouse, making the simulation accurate and meaningful.

It is important to ensure that the data collected for such simulations is proper in order to make the simulation accurate and meaningful. For example, you are considering the addition of an approval step that is conditional based on the type of account. How would the addition of that step affect the process based on past history? With IBM BPM, all of the simulation parameters from existing steps can be automatically populated by historical data from any period of time you choose.

How often and in what specific cases is this conditional step taken? To answer this question, you must be able to run historical process data through your proposed process changes. IBM BPM enables you to do exactly this. Furthermore, you can create scenarios that filter the analysis based on specific business data.

Optimization analysis using inflight performance data

IBM BPM Process Optimizer enables you to make changes to the process and affect inflight tasks through task priority management. For example, you might want to adjust a business rule, such as a threshold, or you might want to change the process, as shown previously, by adding new steps.

In the same way that you want realistic parameters for simulation based on historical data, you also want to know how these changes are going to affect your current inflight processes. The IBM BPM Process Optimizer can run the same type of performance analysis using the business and process data of inflight tasks, such as the current numbers of active tasks, overdue tasks, and the metadata associated with each instance.

The IBM BPM Process Optimizer can tell you what is going to happen in the future using current data. In addition, with heat map animation and a time scale as a slider, IBM BPM can tell you where a bottleneck will occur in the future, and when it will occur as well. For the real-time enterprise reacting to business events, such an analysis is essential.

Guided optimization

The IBM BPM Process Optimizer also provides a powerful feature, called *Guided Optimization*, which guides you through a series of steps that help identify problem areas and potential solutions. IBM BPM Process Optimizer performs complex calculations and makes structural change recommendations, including adding Rule Services and Decision Gateways to your process.

Therefore, the IBM BPM Process Optimizer not only tells you what will happen if you make a change, but can also suggest what changes to make based on the same real-world data.

For more information about simulating and optimizing business processes using IBM BPM Process Optimizer, see the IBM BPM 8.5.5. Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.wle.admin.doc/topics/optimizer_introduction.html?lang=en

6.2.3 IBM Business Monitor

IBM Business Monitor provides near real-time visibility into the performance of business activities by processing events from various sources, calculating business metrics, and presenting KPIs through role-based business dashboards. Users with appropriate roles can track current business performance against expectations, and analyze trends over time.

It enables you to understand business performance so that you can compare performance with defined results. You can use the dashboards to see if KPIs are tracking to their wanted targets, and to determine if you have any unforeseen bottlenecks in your business process, such as inactivities involving human tasks. It can also help when something goes wrong, and in situations when expectations are not met.

IBM Business Monitor can also help make an organization aware of potential problems proactively, enabling a directed action to be planned and carried out. Organizations are empowered with the capability of taking the correct action at the correct time, helping to realize more value from IT. IBM Business Monitor is relevant in all areas of the business, across both the functional and process perspectives.

A financial institution, for example, might use it to manage and track loan processes in real-time, combining information about human and automated elements of the process into a single view.

A government could use IBM Business Monitor to gain visibility into operations of the social services agency department.

In health care, IBM Business Monitor can be used to have an overview of all operations within a hospital, including management of insurance claims processing, scheduling of testing equipment needs and staff assignments.

The following key features are offered by IBM Business Monitor:

- A company-wide view of end-to-end business processes, some of which might exist outside the IBM BPM software. For example, the ability to monitor processes implemented with a combination of the IBM BPM application and other existing applications, such as mediation services hosted in the IBM Integration Bus.

  Such a monitoring and reporting ability can give you visibility into multiple business process applications spread across different systems that are part of the same business process. This ability can also help you make informed business decisions.
- Ability to define and view the KPIs and process metrics using the web and mobile interfaces.
- Business user's ability to set and control the conditions under which alerts are sent, without ongoing IT involvement.
- Ability to create custom monitor models in IBM WebSphere Integration Developer using events from business process definition, Business Process Definition Language, or other IBM middleware and third-party systems.
- Ability to include IBM Business Monitor models in the Process Center repository for collaboration, creating versions, and governance.
- Ability to create and view dimensional reports using data collected by IBM Business Monitor. The dimensional model is made available to IBM Cognos® Business Intelligence Server, which is included with IBM Business Monitor.
- Ability to see what is happening as it occurs dynamically using several prebuilt business space widgets. These widgets can be customized by users in their own IBM BPM Business Space.

For more information about IBM Business Monitor, see the IBM Business Monitor 8.5.5 Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SS7NQD_8.5.5/com.ibm.wbpm.mon.doc/kc-home.html

6.3 Conclusion

In this chapter we described the concepts of business-centric visibility, and described how business-centric visibility is achieved in IBM BPM:

- Business and process data reports that can be used to make informed business decisions, and to get measurable insight into process performance. These reports can also be used by compliance team members to spot any violations to the stated business policies.
- IBM BPM Process Optimizer, which gives the ability to simulate business processes, analyze the results of simulation, and make recommendations about optimizing these business processes.
- IBM Business Monitor, which facilitates monitoring of end-to-end business processes, some of which might exist outside the IBM BPM software. End-to-end process monitoring can be key to ensuring corporate and regulatory compliance.

7

Performance and IT-centric visibility

In this chapter, we describe the performance tuning process flow in IBM Business Process Manager (IBM BPM), cover the concepts of information technology (IT)-centric visibility, and describe the performance diagnostic tools that can be used with IBM BPM.

This chapter includes the following sections:

- Performance tuning process flow
- IT-centric visibility
- Performance diagnostic tools
- Sample use cases

7.1 Performance tuning process flow

Because performance tuning is an ongoing activity in IBM BPM, it is important to document the performance tuning process and use it as a reference for ongoing tuning activities. The performance tuning process flow, shown in Figure 7-1, outlines the activities performed for tuning the specific parameters within the environment and the application.

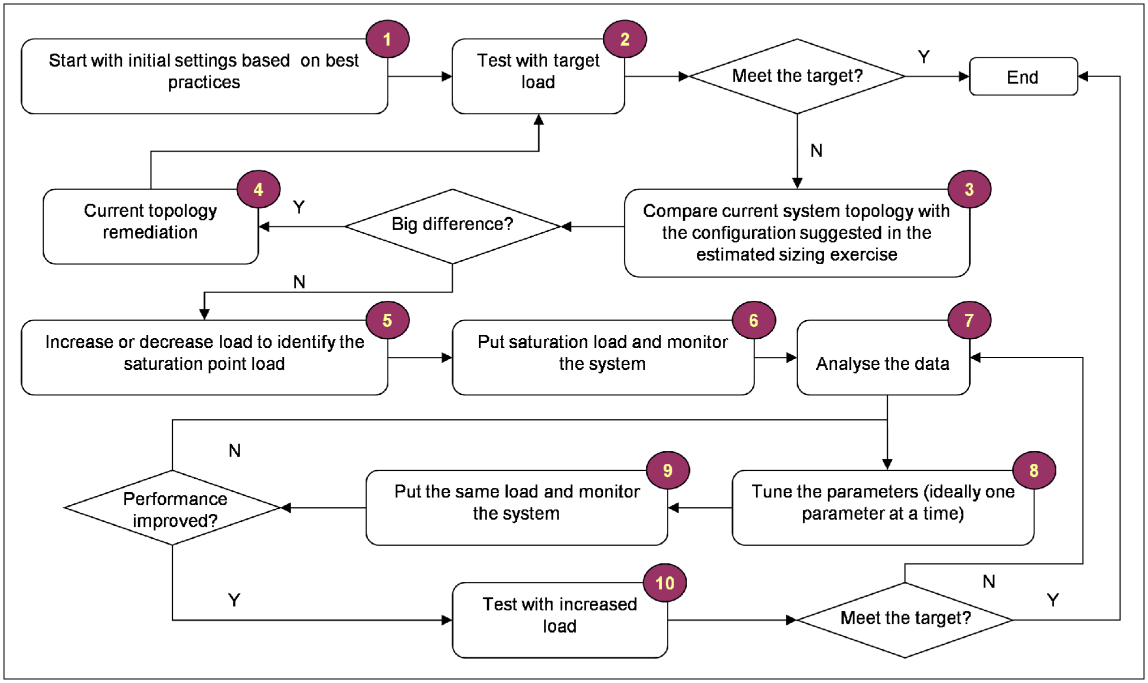

1
Start with initial settings based on best practices
2
Test with target load
Meet the target?
Y
End
N
3
Compare current system topology with the configuration suggested in the estimated sizing exercise
Big difference?
Y
4
Current topology remediation
N
5
Increase or decrease load to identify the saturation point load
6
Put saturation load and monitor the system
7
Analyse the data
8
Tune the parameters (ideally one parameter at a time)
9
Put the same load and monitor the system
Performance improved?
N
Y
10
Test with increased load
Meet the target?
N
Y

Figure 7-1 Performance tuning process flow

The following section provides details about the steps shown in Figure 7-1:

1. Start with initial IBM BPM system settings based on best practices outlined in the IBM Redbooks publication *IBM Business Process Manager V8.5 Performance Tuning and Best Practices*, SG24-8216.
2. Initiate a performance test with the intended target load to determine if the current configuration can meet the performance target, including the throughput and response time target. If the target can be met, no further performance tuning is required.

3. If the IBM BPM system cannot handle the intended target load, verify the current system topology with a suggested configuration based on estimated sizing values[1].
4. If the current environment settings are lower than the configuration settings suggested in the sizing exercise, the current topology should be remediated to increase the system capacity.
5. If the current environment settings match the configuration settings suggested in the sizing exercise, the saturation point test should be conducted to evaluate the current capacity. The load that drives the system to its saturation point can be used as the baseline load (also called the saturation load) for performance tuning.
6. Monitor the system under the baseline load using performance diagnostic tools described in 7.3, “Performance diagnostic tools” on page 238. We suggest that you start monitoring the system using the following tools:

   – Nmon analyzer
   – IBM WebSphere Performance Tuning Toolkit
   – IBM Monitoring and Diagnostic Tools for Java - Health Center

7. Analyze the performance data captured using the monitoring tools described in step 6 previously.
8. Tune the system parameters as necessary. We suggest tuning one parameter at a time before rerunning the performance test to analyze the effect of parameter change on the performance of the system. It is important to note that tuning can involve updating the components that are outside the servers hosting IBM BPM. When tuning such components, we suggest that you follow the appropriate tuning guidelines.
9. Rerun the performance test under baseline load, and monitor the system using performance diagnostic tools described in 7.3, “Performance diagnostic tools” on page 238. If the IBM BPM system does not meet the performance target, repeat steps 7 - 9 until the system performance aligns with the performance target at baseline load.
10. If the system performance meets the performance target at baseline load, increase the load and rerun the performance test again. If the system performance does not meet the performance target at increased load, repeat steps 7-10 until the system performance meets the performance target at expected production load.

[1] We assume that sizing estimates are based on expected production load. As future projects are added, we suggest re-evaluating the sizing estimates as part of capacity planning exercise.

7.2 IT-centric visibility

IT-centric visibility is a capability that gives an overall view of what is happening in a runtime solution. IT-centric visibility includes:

- Ability to audit systems for compliance.
- Ability to ensure that services meet service level agreements.
- Ability to proactively monitor system, subsystem, and application health.
- Ability to diagnose and resolve application issues as soon as possible.

IT-centric visibility encompasses logging, auditing, and monitoring activities. Decisions made on logging, auditing, and monitoring strategies can often have a significant effect on the reliability, availability, and scalability of the system. This section describes the following concepts:

- Logging
- Auditing
- Monitoring

7.2.1 Logging

Logging involves recording the behavior of a running program for debugging, monitoring, and troubleshooting purposes. For example, when troubleshooting problems, logged messages can be used to isolate and make problem determination local. Logged data can also be used by monitoring tools to analyze the health of an application, and generate alerts when predetermined error situations or resource usage limits are encountered.

The granularity of the information to log is often determined by several factors:

- Applications interacting with external systems often require more logging to track the flow of information between systems.
- The complexity of an application adds to the amount of data being logged.
- Certain service types typically have additional logging requirements to expedite the problem determination, analysis, and resolution process. For example, credit card transaction processing services often have extensive logging requirements, as compared to logging requirements for services retrieving address information.

IBM BPM offers the following logging options:

- Built-in logging options offered by the underlying WebSphere Application Server. For more information, see the following IBM developerWorks article:

  http://www.ibm.com/developerworks/websphere/techjournal/0802_supauth/0802_supauth.html

- Apache Log4j[2] (Log4j) logging framework. Organizations typically choose to use Log4j when they are looking for a highly configurable logging utility with the ability to change configuration values with external files at run time. For information about using Log4j to create a logging framework in IBM BPM, see the following IBM developerWorks article:

  http://www.ibm.com/developerworks/bpm/library/techarticles/1410_chandran/1410_chandran.html

  The following list includes typical performance considerations when using Log4j in an IBM BPM application include the following:

  - The amount of data being logged at each Log4j log level. For example, the decision to log an entire message payload or a business object should be made only after much deliberation, because logging excessive data can lead to performance degradation.
  - Enabling logging at the entry point, the exit point, and the exception trace on application services, as a starting point. More logging points are typically required during the development phase, but these should be removed before production release. Therefore, you only take forward those logging points that are necessary.
  - Using Log4j asynchronous loggers, where possible. These asynchronous loggers can offer lower logging latency and higher throughput as compared to synchronous loggers. For more information about asynchronous loggers and considerations for choosing between synchronous and asynchronous loggers, see the following website:

    http://logging.apache.org/log4j/2.0/manual/async.html
  - Guarding Log4j object creation and log message parsing at run time by first checking for the logging level enabled in Log4j configuration settings. Ensure that complex strings are not prepared until the logging level has been checked. This can help reduce the resource use associated with creating objects and building strings that will not be logged.
  - Using common code when implementing common logging logic.
  - Sensible use of Log4j trace levels for logging information. For example, information that can prove useful at DEBUG trace level must not be logged under `INFO`, `WARN`, or `ERROR` trace levels.

For more information about application logging in IBM BPM, see 5.7, “Logging” on page 203.

[2] For more information about Apache Log4j, visit their website at http://logging.apache.org/log4j/2.x/

It is important to consider the amount of data being logged, and its effect on the performance of the application. This is especially true for systems operating under significant load. If there is a requirement for a large amount of application logging that can stress the disk system, consider placing these application logs on a faster disk, potentially separate from other product logging.

In the production environment, we suggest restricting write access to log files, to ensure the integrity of these files. We also suggest that you consider defining a strategy for the maintenance of log files generated by IBM BPM Java virtual machines (JVMs), node agents, deployment manager, and any application-specific log files created using the Log4j logging framework.

7.2.2 Auditing

IT-centric auditing is the practice of recording user actions or a certain type of transactions, performed in context of a software system or service, for purposes of compliance. Reliable audit trails can help answer the following questions in context of user actions performed in an application, or when starting a service:

- What was changed in a system or by a service?
- Who made the change in a system, or who started the service that triggered the change?
- When were the changes made in a system or by a service?
- Was the change performed in the realm of authorization granted to the user?

The amount of data captured for an audit trail and the retention period for an audit trail is typically dictated by the compliance department in an organization. For example, financial institutions typically enforce strict requirements for establishing an audit trail in their software systems and services.

In IBM BPM, an audit trail can be captured for applications and services using the following tools:

- IBM WebSphere Application Server system log files.
- Message logger primitive (in IBM BPM Advanced) to store audit messages in a relational database.
- Log4j logging framework. Performance considerations shared in 7.2.1, “Logging” on page 234 apply when audit information is captured using Log4j.

It is important to carefully consider the amount of data being audited. Auditing excessive or unnecessary information can lead to performance degradation under significant load. Because audit trails can play an important role in a legal situation, it is important to prevent their tampering, unauthorized access, or destruction.

7.2.3 Monitoring

IT-centric monitoring is the real-time scrutiny of systems, applications, or services to proactively identify potential failures, or to detect problems as soon as they manifest. Monitoring is essential to ensure that the systems, applications, or services meet their stated goal within the organization. Different categories of monitoring that must be considered when implementing IBM BPM include:

- System health monitoring

  System health monitoring encompasses monitoring the health of the physical and operational systems. It typically includes monitoring the connection pools, thread pools, queue sizes, memory, processor, network traffic, and disk and file input/output (I/O). System health monitoring can also include monitoring the runtime dependencies of supporting subsystem resources, such as file systems, databases, messaging systems, and a security registry, such as the Lightweight Directory Access Protocol (LDAP).

- Application health monitoring

  Application health monitoring encompasses monitoring the health of functional components that typically include failed events, running components, and adapters.

- Compliance monitoring

  Compliance monitoring encompasses monitoring the audit trail to ensure that the systems, applications, and services are adhering to regulatory and corporate policies.

- Service-level monitoring

  Service-level monitoring encompasses monitoring the service level agreements of exposed services. It typically includes tracking the service response times, number of service requests from a consumer, and service request rate on a per-consumer basis. This information can be used to improve offered services by ensuring that services meet the response times agreed on between service consumers and service providers.

  It can also be used to throttle requests from a consumer during peak load to ensure that all consumers get their share of the agreed service by way of SLA between service consumer and service provider.

- Problem diagnostics monitoring

  This includes drilling down to the details of issues highlighted in each of the previous monitoring categories, such as deadlocks, hung threads, memory leaks, and performance bottlenecks. For more details about problem diagnostic monitoring in IBM BPM, see the IBM Redbooks publication *IBM Business Process Manager V8.5 Performance Tuning and Best Practices*, SG24-8216.

Consider the following important questions before monitoring in each of the previous categories:

- Do you know what you are monitoring? For example, what information does the compliance team in your organization need from your IT-centric monitoring?
- Have you established thresholds to monitor and, more importantly, have you considered how to get alerted when those thresholds are exceeded? This is particularly important with system monitoring and service-level monitoring.
- Can you tie the information captured from one category of monitoring with other categories? For example, high processor use can lead to missed SLAs.

7.3 Performance diagnostic tools

In IBM BPM, some of the key performance metrics, such as processor usage, memory usage, thread pool size, connection pool size, and disk I/O, can be collected and analyzed using different tools. In this section, we list some of the tools that can be used to diagnose performance issues, and we explain how the tools can be used to collect the data and diagnose the issues in IBM BPM:

- Nmon
- WebSphere Application Server Performance Tuning Toolkit
- IBM Monitoring and Diagnostic Tools for Java - Health Center
- IBM Monitoring and Diagnostic Tools for Java - Memory Analyzer
- Pattern Modeling and Analysis Tool for IBM Garbage Collector
- IBM Thread and Monitor Dump Analyzer for Java
- Database monitoring tools

7.3.1 Nmon[3]

The nmon tool can be used to monitor system metrics on IBM AIX® and Linux systems:

- Processor use
- Memory use
- Kernel statistics and run queue information
- Disk I/O rates, transfers, and read/write ratios
- Free space on file systems
- Disk adapters
- Network I/O rates, transfers, and read/write ratios

[3] We suggest that you consider using utilities, such as Task Manager and Perfmon, for monitoring the Windows servers. If you are hosting IBM BPM on virtual servers, contact your operational team for suggestions about monitoring tools that can be used to monitor those servers.

- Paging space and paging rates
- Processor and AIX specification
- Top processors
- IBM HTTP web cache
- User-defined disk groups
- Machine details and resources
- Asynchronous I/O: AIX only
- Workload Manager: AIX only
- Network File System (NFS)
- Dynamic logical partition (LPAR) changes: Only IBM pSeries (IBM System p5®) and IBM OpenPower™ for either AIX or Linux

For more information about nmon, see the following IBM developerWorks article:

http://www.ibm.com/developerworks/aix/library/au-analyze_aix/

7.3.2 WebSphere Application Server Performance Tuning Toolkit

WebSphere Application Server Performance Tuning Toolkit (PTT) is an Eclipse-based tool used to collect real-time performance metrics from the WebSphere Application Server used by IBM BPM.

Using Java Management Extensions (JMX) application programming interface (API), PTT captures data from basic Performance Monitoring Infrastructure (PMI) of the WebSphere Application Server, and presents it in the form of charts and graphs. PTT requires the PMI to be enabled on WebSphere Application Server used by IBM BPM. When enabling PMI tracking points, take care to only activate those that are needed, because there is runtime resource use associated with each tracking point.

For information about enabling PMI in IBM BPM, see the IBM BPM 8.5.5 Knowledge Center at:

http://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.5/com.ibm.wbpm.admin.doc/topics/tmon_prfstartadmin.html

A configured PTT tool captures the following key performance metrics:

- IBM JVM processor use.
- JVM memory (heap size and used memory) use.
- Enterprise JavaBean (EJB) and servlet throughput and response times. The name of the EJB or servlet can normally be used to locate the Service Component Architecture (SCA) modules in the development tool.
- Discarded Java Database Connectivity (JDBC) statement cache.
- Connection pool usage of JDBC data sources.

- Connection pool usage of Java Message Service (JMS) connection factory.
- Rolled back transactions.
- Timed out transactions.
- Used and available thread pool counts for WebSphere Application Server thread pools, such as web container, default, and Object Request Broker (ORB) thread pools.

In addition to capturing the key performance metrics, PTT also shows the topology retrieved after connecting to the IBM BPM cell, and a list of servers configured in the IBM BPM cell. For more information about PTT, see the following IBM developerWorks article:

http://www.ibm.com/developerworks/websphere/downloads/performtuning.html

7.3.3 IBM Monitoring and Diagnostic Tools for Java - Health Center

The IBM Monitoring and Diagnostic Tools for Java - Health Center (IBM Health Center) provides a no-charge, low-resource diagnostic tool and API for monitoring an application running on a JVM. IBM Health Center gives the ability to assess the current status of a running Java application. IBM Health Center continuous monitoring provides the following information to identify and help resolve problems with your application:

- Identify if JVM native memory or JVM heap memory is leaking. When memory leaks are identified, they can be investigated further using IBM Monitoring and Diagnostic Tools for Java - Memory Analyzer to identify the source of the memory leak.
- IBM Health Center can be used with PTT and nmon to identify methods that lead to high processor usage.
- Identify I/O bottlenecks.
- Visualize garbage collection statistics. Concerns highlighted in garbage collection statistics can be investigated further using verbose garbage collection (verbose GC) traces in WebSphere Application Server. We suggest using the pattern modeling and analysis tool (PMAT) for IBM Garbage Collector for analysis of the verbose GC trace files.
- View any lock contentions.
- Analyze unusual WebSphere real-time events.
- Monitor application thread activity.
- Detect deadlock conditions in an application.
- Gather class histogram data.

In an IBM BPM production environment, we suggest configuring the IBM Health Center agent in *headless mode*. In headless mode, the IBM Health Center agent writes data to the local files on the JVM system, and the resulting `.hcd` files can be loaded to the IBM Health Center client on a different system.

This eliminates the need for changes in the firewall rules to establish connectivity between IBM Health Center client and IBM Health Center agent, assuming that the client and agent run on different systems.

For more information about IBM Health Center, see the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SS3KLZ/com.ibm.java.diagnostics.healthcenter.doc/homepage/plugin-homepage-hc.html?lang=en

7.3.4 IBM Monitoring and Diagnostic Tools for Java - Memory Analyzer

IBM Monitoring and Diagnostic Tools for Java - Memory Analyzer (IBM Memory Analyzer) is a feature-rich Java heap analyzer that can help find memory leaks and reduce memory consumption. IBM Memory Analyzer provides both high-level understanding and analysis summaries using several standard reports. IBM Memory Analyzer enables users to perform in-depth analysis through browsing and querying the Java objects present on the Java heap.

The following features are offered by IBM Memory Analyzer:

- Standard report that uses built-in capabilities to look for probable leak suspects.
- In-depth leak identification capabilities that look for collections with a large numbers of entries, single large objects, or groups of objects of the same class.
- Overview report that provides information about the Java heap usage, system property settings, threads present, and a class histogram of the memory usage.
- Top memory consumers report that gives a breakdown of the Java heap usage by the largest objects, and also which class loaders and classes are responsible for using the most memory.
- Reports outlining the top memory consumers, and providing information about potential memory inefficiencies in any selected component.
- Ability to browse the Java heap using a reference tree-based view. This makes it possible to understand the relationships between Java objects, and to develop a greater understanding of the Java object interactions and the memory requirements of the application code.

For more information about IBM Memory Analyzer, see the IBM Support Assistant 5.0 Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SS3KLZ/com.ibm.java.diagnostics.memory.analyzer.doc/homepage/plugin-homepage-ma.html?cp=SSLLVC_5.0.0%2F7&lang=en

7.3.5 Pattern Modeling and Analysis Tool for IBM Garbage Collector

PMAT analyzes verbose GC traces by parsing the traces and building pattern models. PMAT suggests key configurations by executing a diagnostic engine and pattern modeling algorithm. If there are any errors related with Java heap exhaustion or fragmentation in the verbose GC trace, PMAT can diagnose the root cause of failures. PMAT provides rich chart features that graphically display Java heap usage.

The following features are offered by PMAT:

- Garbage collection analysis
- Garbage collection table view
- Allocation failure summary
- Garbage collection usage summary
- Garbage collection duration summary
- Garbage collection graph view
- Garbage collection pattern analysis
- Zoom in or zoom out or selection or center of chart view

For more information about PMAT, see the IBM Support Assistant 5.0 Knowledge Center on the following website:

http://www.ibm.com/support/knowledgecenter/SSLLVC_5.0.0/com.ibm.trove.tool.pmat.doc/docs/readme.html?cp=SSLLVC_5.0.0%2F8&lang=en

7.3.6 IBM Thread and Monitor Dump Analyzer for Java

IBM Thread and Monitor Dump Analyzer for Java (IBM Thread and Monitor Dump Analyzer) enables identification of hangs, deadlocks, resource contention, and bottlenecks in Java threads. IBM Thread and Monitor Dump Analyzer offers the following features:

- Java heap information, such as maximum heap size, initial heap size, garbage collection counter, free heap size, and allocated heap size.
- Ability to report on the number of runnable threads, number of blocked and waiting threads, and total number of threads in the JVM.
- Ability to report on number of locked monitors.

- Ability to identify deadlocks between threads.
- Ability to compare multiple thread dumps (also called *Java dumps* or *Java core files*).

For more information about IBM Thread and Monitor Dump Analyzer, see the IBM Support Assistant 5.0 Knowledge Center at:

http://www.ibm.com/support/knowledgecenter/SSLLVC_5.0.0/com.ibm.esupport.tool.tmda.doc/docs/readme.htm?cp=SSLLVC_5.0.0%2F9&lang=en

7.3.7 Database monitoring tools

We suggest using database monitoring tools, such as IBM Data Studio web console, to view the database health information. This information can indicate the overall health of the databases used by IBM BPM, and provide details about specific database-related health indicators. For more information about IBM Data Studio web console, see the IBM Data Studio Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SS62YD_4.1.1/com.ibm.datatools.db.web.health.doc/topics/health_overview.html

7.4 Sample use cases

In this section, we describe sample use cases, highlighting the usage of performance diagnostic tools in addressing some of the commonly seen performance issues in IBM BPM. For the sake of this description, we assume that IBM BPM is installed on an AIX operating system:

- High processor use:
  - Use the nmon tool to identify the process that causes high processor resource use.
  - If the high processor use is attributed to an IBM BPM JVM process, configure the IBM Health Center tool on that JVM to identify the methods that lead to high processor usage in the JVM.
- High memory use:
  - Use the nmon tool to identify the process that consumes high memory.
  - If the high memory use is attributed to an IBM BPM JVM process, configure the IBM Health Center tool on that JVM to identify if the issue is related to high native memory use or high heap memory use. If the IBM Health Center tool points to high native memory usage, we suggest raising a problem management record with IBM Support.

  - If the issue is related to high heap memory usage, try the following actions:
    - Enable verbose GC on this JVM, and use the PMAT tool to identify any potential issues in the JVM configuration. JVM tuning is typically required to address JVM configuration issues. We suggest that you enable verbose GC in the production environment.

      There is no measurable resource use associated with letting verbose GC run in production. However, consider a file management strategy for output files generated by verbose GC, because these files will typically grow over time if left unmanaged.
    - Use the IBM Memory Analyzer tool to spot any memory leaks, and analyze the application footprint for anomalies in heap memory usage.
- Exhaustion of WebSphere Application Server container resources, such as thread pools, database connection pool, discard statement cache, and so on.

  As an example, one of the common areas to monitor is the web container thread pool. Use the PTT tool to monitor the web container thread pool count. If the used thread pool count continues to rise over time, we suggest taking multiple thread dumps. Analyze and compare the Java core files in the IBM Thread and Monitor Dump Analyzer tool for hung threads, resource contentions, or deadlocks.

  Increase in a web container's used thread pool count over time can also indicate increased workload during peak hours. Multiple thread dumps taken over time can also be used to see if active threads were reaching the maximum thread limit, and whether the thread pool settings needed addressing to enable increased concurrency.

  The PTT tool can be used to monitor other container resources, such as database connection pool, discard statement cache, and so on.
- Delayed response time.

  Delayed response times in IBM BPM can be a symptom of several factors in the underlying system or application:
  - Use the nmon tool to spot high processor or memory use issues, I/O issues, and network issues.
  - Use the PTT tool to monitor the IBM BPM system for alerts associated with thread pool count, used connection pool size, cache statement discard count, servlet and EJB response times, and transaction rollback count.
  - Use the IBM Health Center tool to identify any resource locking issues.
  - Use a database monitoring tool to identify any issues associated with the health of databases used by IBM BPM.

  - In IBM BPM Advanced, use the WebSphere Application Server PMI tool to monitor the service integration bus (SIBus) queue depth. Because SIBus is used by IBM BPM Advanced, it is important to monitor the SIBus exception queues for error messages. A large accumulation of error messages on the exception queue can lead to performance degradation in IBM BPM Advanced.

    For instructions on enabling automatic monitoring of the WebSphere Application Server SIBus queue depth, see the following web document:

    http://www.ibm.com/support/docview.wss?uid=swg21618858

  - When IBM BPM is interfacing with external systems, monitor the network traffic between these systems for potential response time delays. We suggest that you consider using packet tracing tools (approved by your company) for monitoring the network traffic. We also suggest that you consult with the team monitoring the external systems themselves, to see if there are any issues with those systems.

7.5 Conclusion

In this chapter, we described the following topics:

- Performance tuning process flow that can be used as a reference for ongoing performance tuning activity in IBM BPM
- Concepts of IT-centric visibility and important considerations of IT-centric visibility that can affect reliability, availability, and scalability (RAS) of systems or applications
- Performance diagnostic tools that can be used for collecting and analyzing key performance metrics in IBM BPM
- Sample use cases highlighting the use of performance diagnostic tools in addressing some of the commonly reported performance issues in IBM BPM

Related publications

The publications listed in this section are considered particularly suitable for a more detailed description of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topics in this document. Note that some publications referenced in this list might be available in softcopy only:

- *Scaling BPM Adoption: From Project to Program with IBM Business Process Manager*, SG24-7973
- *Applying Lean, Six Sigma, BPM, and SOA to Drive Business Results*, REDP-4447
- *Business Process Management Deployment Guide Using IBM Business Process Manager V8.5*, SG24-8175
- *Creating a BPM Center of Excellence (CoE)*, REDP-4898
- *IBM Business Process Manager V8.0 Performance Tuning and Best Practices*, REDP-4935
- *IBM Business Process Manager V8.5 Performance Tuning and Best Practices*, SG24-8216
- *Process Discovery Best Practices Using IBM Blueworks Live*, REDP-5111
- *Leveraging the IBM BPM Coach Framework in Your Organization*, SG24-8210
- *IBM Business Process Manager Security: Concepts and Guidance*, SG24-8027
- *IBM Business Process Manager Version 8.0 Production Topologies*, SG24-8135
- *Understanding LDAP - Design and Implementation*, SG24-4986
- *Empowering your Ad Hoc Business with IBM Business Process Manager*, REDP-4995

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM

Redbooks

Business Process Management Design Guide: Using IBM Business Process Manager

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

SG24-8282-00

ISBN 0738440590

Printed in U.S.A.