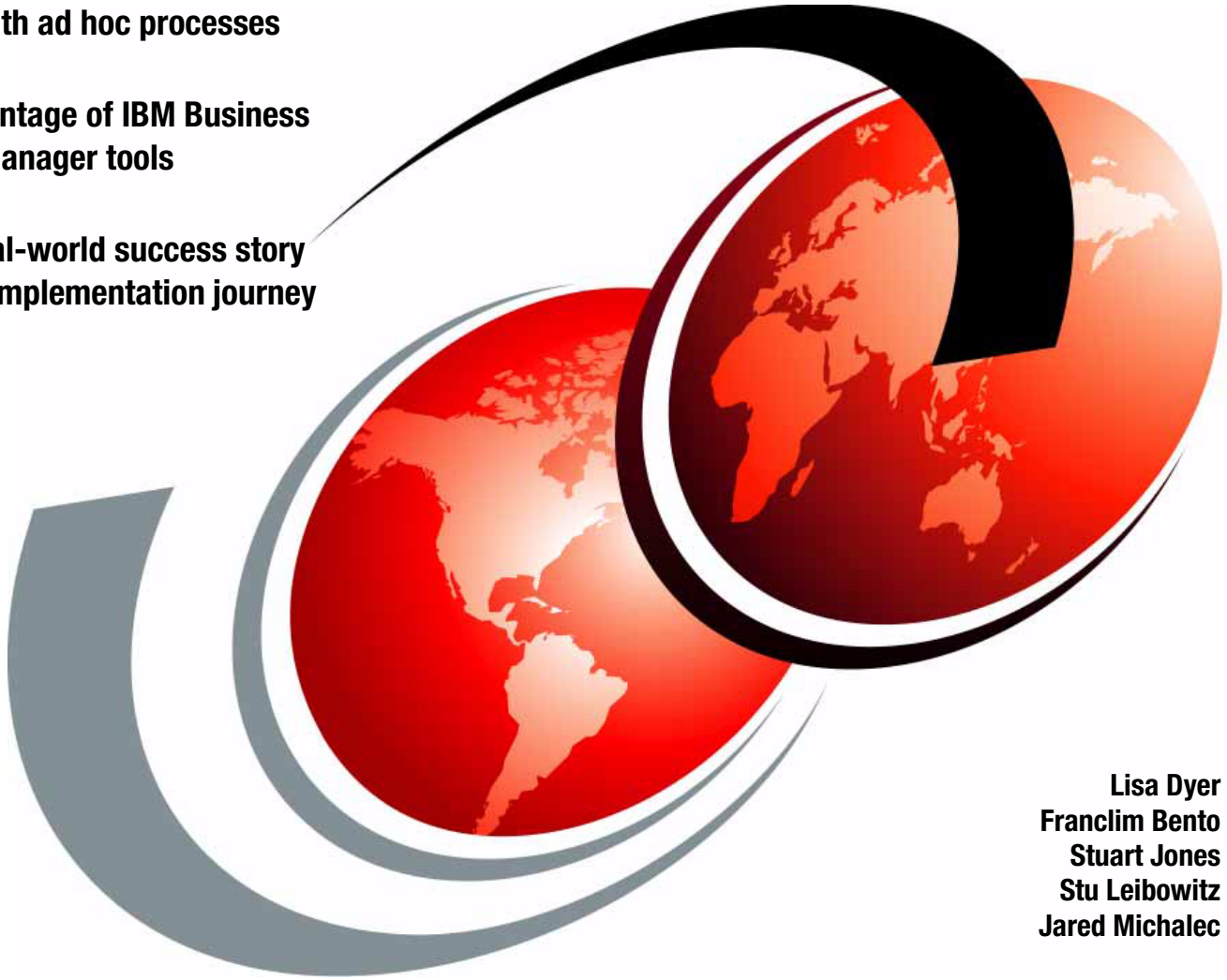
IBM

Empowering your Ad Hoc Business with IBM Business Process Manager

- Realize the benefits of BPM when dealing with ad hoc processes
- Take advantage of IBM Business Process Manager tools
- Read a real-world success story of a BPM implementation journey



Lisa Dyer
Franclim Bento
Stuart Jones
Stu Leibowitz
Jared Michalec

Redpaper

ibm.com/redbooks

International Technical Support Organization

Empowering your Ad Hoc Business with IBM Business Process Manager

June 2013

REDP-4995-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (June 2013)

This edition applies to Version 8 of IBM Business Process Manager.

.

Contents

Notices vii
Trademarks viii

Preface ix
Authors ix
Now you can become a published author, too! x
Comments welcome. xi
Stay connected to IBM Redbooks xi

Chapter 1. Introduction. 1
1.1 The ad hoc debate 2
1.1.1 Ad hoc imperatives 2
1.1.2 A classic case of process woes 3
1.2 Democracy, adhocracy, and business agility 4
1.2.1 Operating between adhocracy and bureaucracy 4
1.2.2 Why democratization of business improvement is important 4
1.3 Creating a framework for your ad hoc business 5
1.3.1 Establish a common vocabulary 5
1.3.2 Choose what parts of your business need more (or less) agility. 6
1.3.3 Choose where and how to apply management concepts and techniques 6
1.3.4 Implement enough structure and instrumentation. 6
1.3.5 Interpret and respond 7
1.4 Ad hoc capabilities with IBM Business Process Manager. 7
1.5 How is this book organized 8

Chapter 2. The ad hoc spectrum 9
2.1 Defining the term ad hoc 10
2.2 Ad hoc characteristics 10
2.2.1 Predictability 10
2.2.2 Manageability 11
2.2.3 Visibility 11
2.2.4 Flexibility 12
2.2.5 Sequencing 12
2.2.6 Granularity of ad hoc characteristics. 12
2.3 Ad hoc styles. 14
2.3.1 Structured and predictable 14
2.3.2 Ad hoc by design 14
2.3.3 Dynamic and generic processes. 15
2.3.4 Event-based task flows 16
2.4 Ad hoc capabilities 17
2.5 To be or not to be ad hoc 18

Chapter 3. Applying the ad hoc spectrum to your business. 21
3.1 Airline 22
3.1.1 The problem 22
3.1.2 Applying the ad hoc spectrum. 23
3.1.3 An example solution 24
3.2 Healthcare. 24
3.2.1 The problem 24

3.2.2 Applying the ad hoc spectrum . . . 25
3.2.3 Example solutions . . . 26
3.3 Banking . . . 27
3.3.1 The problem . . . 27
3.3.2 Applying the ad hoc spectrum . . . 28
3.3.3 An example solution . . . 29
3.4 Leasing . . . 29
3.4.1 The problem . . . 29
3.4.2 Applying the ad hoc spectrum . . . 30
3.5 Corporate tax . . . 31
3.5.1 The problem . . . 31
3.5.2 Applying the ad hoc spectrum . . . 32

Chapter 4. Implementing solutions for your ad hoc business . . . 33
4.1 Start with a more structured BPM approach . . . 34
4.1.1 Credit approval initial implementation . . . 36
4.1.2 Improving ad hoc aspects over the initial implementation . . . 37
4.1.3 Implementing task reassignment . . . 38
4.1.4 Handling ad hoc events on separated processes . . . 39
4.1.5 Extending the task transfer process . . . 39
4.1.6 Conclusion . . . 41
4.2 Using events for a more ad hoc approach to process implementation . . . 42
4.2.1 A simple example . . . 42
4.2.2 Examples in business . . . 45
4.2.3 Instrument, and then manage . . . 46
4.2.4 Instrument, and then discover . . . 46
4.2.5 Conclusion . . . 46
4.3 Extending BPM support to all processes in the business . . . 46
4.3.1 Main benefits of this kind of approach . . . 47
4.3.2 Implementing a solution for generalized support of processes . . . 48
4.3.3 Two distinct approaches for generic processes . . . 50
4.3.4 Advantages and disadvantages of each approach . . . 51
4.3.5 Conclusion . . . 51

Chapter 5. Ad hoc process examples . . . 53
5.1 Well-defined process with ad hoc actions . . . 54
5.2 Completely user-defined process . . . 58

Chapter 6. Case study: How Banco Espírito Santo approached ad hoc processes . . . 59
6.1 Overview . . . 60
6.2 The BPM path followed by BES until 2008 . . . 61
6.2.1 BES Digital and the BPM revolution . . . 62
6.2.2 Good, but not good enough . . . 63
6.3 The turning point: Vision and strategy . . . 64
6.3.1 Establish a Center of Excellence . . . 64
6.3.2 Define the frameworks . . . 65
6.3.3 Development and management of the frameworks . . . 66
6.4 How Generics environment got its start . . . 68
6.4.1 Generic Process Engine . . . 68
6.4.2 GPE Supporting Database . . . 73
6.4.3 GPE Configuration Utility . . . 75
6.5 Generics rapidly increased . . . 82
6.6 The benefits of Generics to BES . . . 83
6.7 Examples importance and usefulness of generic processes . . . 84

6.7.1 Leasing company merger 84
6.7.2 Credit processes servicing 85
6.8 Managing generic processes 85
6.8.1 Business team activities 85
6.8.2 IT team activities 92
6.9 Conclusion 95

Chapter 7. Next steps 97
7.1 Using ad hoc capabilities built into Business Process Manager 98
7.2 Non-traditional BPM approaches 100
7.2.1 Generic BPM processes 100
7.2.2 Event-driven processes 101
7.3 Resources to help you get started 102

Abbreviations and acronyms 103

Related publications 105
IBM Redbooks 105
Online resources 105
Help from IBM 105

Notices

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

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on 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.

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:

Blueworks Live™
CICS®
FileNet®
IBM®
IMS™
Lombardi Teamworks®
Lotus®
Redbooks®
Redpaper™
Redbooks (logo)  ®
Teamworks®
WebSphere®

The following terms are trademarks of other companies:

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

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

Preface

In the context of daily business, *ad hoc processes* are those activities and events that occur within an organization's operations that typically are undocumented or unmonitored. At times, these ad hoc processes can seem chaotic and unpredictable. In many cases, these "off the platform" processes represent an opportunity for you to realize visibility into your organization operations. By taking advantage of the benefits of business process management (BPM) and IBM® Business Process Manager solutions, you can bring order and stability to these business processes and improve the organization's agility in order to stay adaptive and competitive.

This IBM Redpaper™ publication presents examples and a case study that illustrate how having a choice of where on the *ad hoc spectrum* you operate your business is both necessary and vital to producing better outcomes and achieving agility. You need agility to stay relevant and to survive. The intent of the prescriptive framework in this paper is to give you the confidence and motivation to choose how much business agility you want and to begin achieving it. This paper is intended for Executive Sponsors, Team Leaders, Lead Architects, and anyone interested in adding business agility and ad hoc processes to their enterprise.

Authors

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Lisa Dyer leads the business of community for the IBM BPM line of business. As a Community Strategist, she works closely with practitioners, often using an adhocratic management style to foster the kind of cultural transformation that leads to the democratization of business improvement. Lisa holds a degree in Design from Finland, and has been in the software business for over 15 years. She has written extensively about business process implementation and improvement.

Franclim Bento is the Chief Architect for BPM and ECM at Banco Espírito Santo (BES) in Portugal. He has 22 years of experience delivering IT solutions to solve business problems in various areas such as government and telecommunication companies, with 12 years working in financial services. In 2006, Franclim started his BPM journey; since 2007 he has conceived, designed, and built one of the most successful IBM Business Process Manager references in the world, the BES BPM initiative. He holds a degree in Computer Science and Software Engineering from Instituto Superior Técnico in Lisbon. His areas of expertise include BPM, SOA, ECM, WebSphere Application Server, Agile, Java, and Java Platform Enterprise Edition from high-level solution design to development.

Stuart Jones is a Business Process Management Solution Architect in the United States. He has 10 years of experience in the Business Process Management field and has worked for IBM for 29 years. Stuart holds a degree in Mathematics and Computing from the University of Bath in England. His areas of expertise include enterprise service bus, business process management and systems architecture.

Stu Leibowitz is a member of the New York Area Technical Sales team. Stu joined IBM through the Lombardi acquisition. As one of the “veteran Lombardi” technical resources, Stu has over eight years of experience helping customers embrace business process management across multiple industries and geographies. Stu is known for his ability to traverse both the business and deep technical aspects of business process management. Although his main focus has been executing customer proof of concepts, Stu is also expert in building and presenting custom demonstrations that help “capture the imagination.” Stu also enjoys evangelizing BPM through product demonstrations and briefings and also conducting BPM workshops. Stu earned his B.S. in Computer Science from The University of Vermont and his M.S. in Computer Science from Boston University.

Jared Michalec is an IBM Worldwide Technical Sales Leader for BPM and Decision Management. He joined IBM in 2008 as part of the acquisition of AptSoft Corporation, which provided the foundation for IBM WebSphere® Business Events. Jared has worked with many key IBM customers to help them identify and implement strategic decision management solutions. He continues his technical sales leadership by being at the forefront of IBM Operational Decision Management, which was released in 2011 and represents the convergence of IBM industry-leading business events and business rules capabilities. Prior to IBM, Jared managed presales, services, and support at AptSoft Corporation. He started his career at Andersen Consulting (now Accenture) as a Senior Consultant implementing enterprise applications for many large organizations. Jared graduated from Cornell University with a degree in Operations Research and Industrial Engineering.

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

- Jonas Zahn, IBM BPM Solution Architect
- Phil Gilbert, Vice President, IBM Business Process and Decision Management
- Martin Keen, IBM Redbooks® Project Leader
- Debbie Willmschen, IBM Redbooks Technical Writer
- Jorge Soares, Banco Espirito Santo Senior Manager
- Pedro Inácio, Banco Espírito Santo Executive Manager

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 customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  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

It is basic human nature to seek structure and predictability in everything we do. We are uncomfortable with chaos because in it lies risk and insecurity. This disposition is deeply encoded in us for reasons of survival. In life, as in business, we often find ourselves balancing between seemingly polar opposites: questioning how to remain flexible yet rigid, structured while unstructured, predictable but unknown, more managed and less managed. And to determine where we want to be on the spectrum between these characteristics, we make the necessary tradeoffs.

In the context of business processes, chaos stems from the myriad unscheduled, uncharted, undocumented, and invisible activities and events that occur within the daily operations of a business. For many enterprises, an overarching motivation in embracing business process management (BPM) as a discipline is to bring order to chaos. A promise of BPM is that you can structure the unknown into something you can better control or, at least, that you can begin to gain visibility into unknown aspects.

As you gain more visibility into your business, you can begin to optimize it for better business outcomes. Often, the first optimizations involve the automation of some manual work (for example, routing manual activities to straight-through processing based on business rules or choreographing and instrumenting “swivel-chair” activities).

As these initial improvements lead to more optimization, your visibility into more areas of the business increases, and you begin to gain better control over your destiny, as illustrated in Figure 1-1.

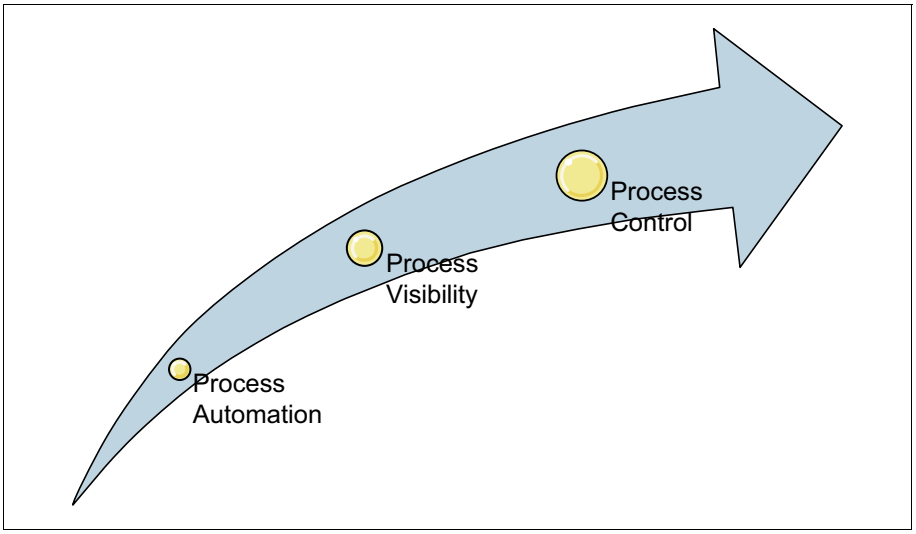


Figure 1-1 Three elements of the compounding value of a managed process

1.1 The ad hoc debate

A narrow and still surprisingly common view of BPM is that you must have a structured view of your business to even begin to practice BPM, or even sustain it and scale it throughout your organization. Yet you cannot expect any structure to serve you well if it does not reflect and support what actually happens in your business. Any model that represents a state in time does not have much longevity to remain relevant in an environment that is surrounded by rapid and constant change. (And, who is not surrounded by rapid and constant change?)

1.1.1 Ad hoc imperatives

Reflect upon your work just this past week. Chances are that you participated in a process where situations took an unplanned turn or where you struggled to get the outcome you wanted fast enough. Instinctively, we find ways around the predetermined process when it does not adequately support us. For example, you might have sought people with more expertise than your own to help with an issue and probably an email or chat was involved. Those “off the platform” activities represent the current reality of the process, and that reality imposes a significant cost[1]:

- Close to 75% of business is conducted in email and other invisible silos.
- Nearly USD $488 billion is lost in process inefficiencies.
- As little as 60% or less of work time is actually productive.

Thus, with a large portion of your business processes occurring off the platform, knowing what processes to optimize and when to optimize those processes can be a challenge. You cannot predict and plan with confidence because you cannot easily see some patterns and problems. You do not have the business agility you need to stay adaptive and competitive.

[1] Source: Tech CEO Council Report October 2010: “BPM improvements can be expected to yield savings of up to 5% of sales”; CNN Money U.S. Fortune 500 Revenues, May 2010

You might think that a certain amount of loss of control because of the ad hoc nature of your processes is just the cost of doing business. In business, as in life, an element of *ad hocery* is always present, and taming it to your will can seem like wishful thinking at times.

Ad hocery: Defined as the “reliance on temporary solutions rather than on consistent, long-term plans[a]”.

a. Source: http://dictionary.reference.com/browse/ad+hocery

But with better insight into your business, you can make more informed and confident decisions about where and how to optimize. If that statement resonates with you, just imagine how empowering everyone in your organization with better visibility can truly make business improvement “go viral.”

But how can more visibility help you if the process needs to run differently every time, and how do you instrument something so dynamic and unknown? And, how do you use that visibility to drive change that is transformational and lasting?

1.1.2 A classic case of process woes

Consider the case of a process for an exception process for making network changes (a true story). Seizing on an opportunity, a team is rebranding a website and needs an IP address change in time for the biggest customer event of the year. This change should be easy enough with only a few stages, right?

1. The customer provides all the information needed to approve the request.
2. The approver receives the request, and then rejects (end of process) or approves the request.
3. When approved, the work to get the change done is orchestrated and completed.
4. Completion of the change is validated with the customer (end the process).

But the process stumbled from the beginning when its design failed to fit its wide-ranging purpose. It called for assessing risk factors that were not relevant to the case, while failing to provide a means to identify the key factors that made the case a clear candidate for a quick implementation. At this stage, the process went *off platform*, and a series of phone, email, and chat conversations ensued in an attempt to reach a conclusion. Meanwhile, the window for making the change closed (the quarterly change freeze took effect). So, the process had to be handled by an arduous exception process. More expert hunting, more hours of research, another handful of conversations and dozens of emails and obscure forms later, the request was finally approved unanimously: only one day before the business deadline.

The approval hearing was over in five minutes, after all the participants agreed that the change was isolated and presented no real business risk, based on data that was evident from the beginning. The remainder of the time was spent discussing how painful and expensive the process was and how it could be fixed.

The type of conversation the experience sparked (a conversation that is surely happening in every organization every day) emphasizes how grassroots involvement in business improvement is powerful. As a result of that conversation, the participants got the process change they advocated for.

1.2 Democracy, adhocracy, and business agility

Continuously, we participate in processes without fully knowing what started them, what the end results were, what their true impact on the business was, and what needed to happen for a successful outcome. Perhaps the process design is delegated to serve what the tools can do, or perhaps the design did not involve the correct input. Providing a framework that enables the people who know most about the true process, that is, the people who participate in the process, define where and how much ad hocery they need or can tolerate, is essential to optimizing your business.

In the last decade of BPM, we have learned to design visibility and flexibility into the business. At the same time, the impact of the continuous streams of unstructured information being injected in real time is fundamentally changing our day-to-day business. Successful businesses are agile and intelligent enough to quickly capture opportunities, solve problems, and get results. You must be able to adapt and respond to unique circumstances in the moment with accuracy and effectiveness. Equally important, you must be able to change your mind, considering that the world around you is always changing. The need for informational flexibility can lead to adhocracy.

1.2.1 Operating between adhocracy and bureaucracy

The spectrum of management styles includes the following extremes:

- Adhocracy

  Adhocracy as a management style provides practically unbounded flexibility and agility but presents challenges to democracy and legality because of its low-key profile. Adhocracy is not concerned with repeatability. Connecting the BPM methodology and the philosophy of adhocracy is the common goal of empowering people to cross organizational lines to solve problems in an agile and effective way.

- Bureaucracy

  Juxtaposed with adhocracy is *bureaucracy*, which gives governance and economies of scale but which impedes agility. Connecting the BPM methodology and the philosophy of bureaucracy is the common goal of governing to give the work legitimacy, currency, accountability, and economies of scale.

Where you are on this spectrum is a reflection of the culture within your organization. Changing your position on the spectrum requires a cultural transformation over time. The results of your first successful BPM projects will help you drive this transformation. (For a prescriptive guide to running successful BPM projects, see *Scaling BPM Adoption from Project to Program with IBM Business Process Manager*, SG24-7973.)

1.2.2 Why democratization of business improvement is important

When you involve knowledge workers, partners, and customers in designing solutions that you deliver to them, you are intentionally putting some of the decision making that was previously done by a few people into the hands of the many who are highly motivated to succeed. Democratization provides the following benefits:

- Enables the right people, in the right place, at the right time to become active participants in the management of your business processes, rules, and events.
- Engenders grassroots innovation.
- Enables users to see and correlate events and rules to process (to make better decisions).
- Drives cultural and lasting transformation.

And, as posited previously, giving the people who know the most about the true process—the people who participate in the process itself—the ability to define the process is essential to optimizing your business.

Fact: To deliver better products and services, 69% of business leaders are collaborating with customers.[a]

a. IBM CEO Study, 2012

Obviously, your business includes aspects that are not subject to change by just anyone, such as IT policies and regulatory guidelines. Governance serves the purpose of giving your organization confidence in the process and trust in the change. You can identify the aspects of your business that you can open to grassroots-led improvement and support those aspects with the governance capabilities.

Consider the story of Banco Espírito Santo (BES), one of the largest banking institutions in Portugal with operations in 23 countries on four continents. In 1998, BES was an early adopter of BPM, setting out to drive process innovation throughout the organization. In the beginning, BES chose high flexibility, a minimal structure, and a process that was instrumented for visibility. The results led to methodical process improvements at scale. For some processes, more flexibility was added, but for processes where repeating patterns were discovered and implemented, more structure was added.

If your technology strategies are not aligned to equip your organization to participate in business improvement, today's market conditions demand that you begin considering it.

1.3 Creating a framework for your ad hoc business

Business agility is about making choices. You must choose where and how to apply management concepts and techniques. You also must choose what parts of your business need more (or less) visibility, flexibility, and predictability.

This section offers a prescriptive framework of concepts, tools, and methods to identify where and how to harness the ad hocery in your operations to achieve better agility and adaptability and to grow your business. To create a framework, consider the following actions:

1. Establish a common vocabulary.
2. Choose what parts of your business need more (or less) agility.
3. Choose where and how to apply management concepts and techniques.
4. Implement enough structure and instrumentation.
5. Interpret and respond.

1.3.1 Establish a common vocabulary

What does the term *ad hoc* mean to you when applied to your particular business context? Using Chapter 2, "The ad hoc spectrum" on page 9 as a conversation starter, you can engage your organization in a simple discovery exercise in defining a common vocabulary.

How you complete this exercise does not matter. The key is enabling it at every level of your business. You can use IBM Blueworks Live™ or another online collaboration tool that is ubiquitously available to your organization. The result should be a living glossary that can help ensure that everyone is "on the same page" when defining where you need to be on the ad hoc spectrum operationally.

You can access IBM Blueworks Live at the following website:

https://blueworkslive.com

1.3.2 Choose what parts of your business need more (or less) agility

Before you change your position on the ad hoc spectrum, establish a roadmap for business agility. The roadmap describes a strategic change that you need to make and why you need to make it. Table 1-1 offers some generalized examples of areas to consider, but your roadmap will depend on your business context.

Tip: For help with this exercise, see 1.3.1, “Establish a common vocabulary” on page 5.

Table 1-1 Roadmap for business agility

Attribute	More	Less
Visibility	► Business owners are relying on subjective and anecdotal evidence of progress (work happening in silos). ► Your workforce cannot easily show how business goals are being met (lack of instrumentation or access to measurements). ► Events and rules that cause change in the real-time process.	► Internal decisions and tasks that do not impact a business goal
Flexibility	► Process improvement and innovation are driven by knowledge workers. ► Decision-making where the impact and risks are best understood.	► Regulations and policies (such as SOX compliance or HR)
Predictability	► Change in policies and regulations.	► External and internal events driven by real-time information

1.3.3 Choose where and how to apply management concepts and techniques

Consider the tenets and mandates of the two opposing management styles (that is, adhocracy and bureaucracy). Where do you need to move on that spectrum to achieve more agility and better business outcomes? For example, the business must be able to change an inflight process to respond to an emerging trend or market event. However, IT must also have confidence that the change will not break anything.

1.3.4 Implement enough structure and instrumentation

You can use executable toolkits to get started with a pilot project. Your goal is to show rapid value and to get more projects funded. See Chapter 4, “Implementing solutions for your ad hoc business” on page 33 as inspiration. Choose a process that has a high probability for success, configure a toolkit, and track all data (turn on auto-tracking) so that you can make well-informed interpretations.

See the IBM Sample Exchange for Process Solutions for toolkits.

http://bpmwiki.blueworkslive.com/display/samples/SAMPLE+EXCHANGE+HOME

1.3.5 Interpret and respond

After you have collected enough data from your running process instances, you are in a position to make informed decisions about where you need to be on the ad hoc spectrum. The data can help you determine where you might need more structure or less structure.

See Chapter 6, “Case study: How Banco Espírito Santo approached ad hoc processes” on page 59, which goes deeper into this subject with concrete examples.

1.4 Ad hoc capabilities with IBM Business Process Manager

IBM Business Process Manager is a comprehensive BPM platform that can provide the visibility and insight that you need to manage business processes. It allows IT to enable business users and managers to track their entire business operation on a single dashboard, to receive alerts, and to drill down to the lowest level of an instance detail.

Figure 1-2 depicts the major components of IBM Business Process Manager.

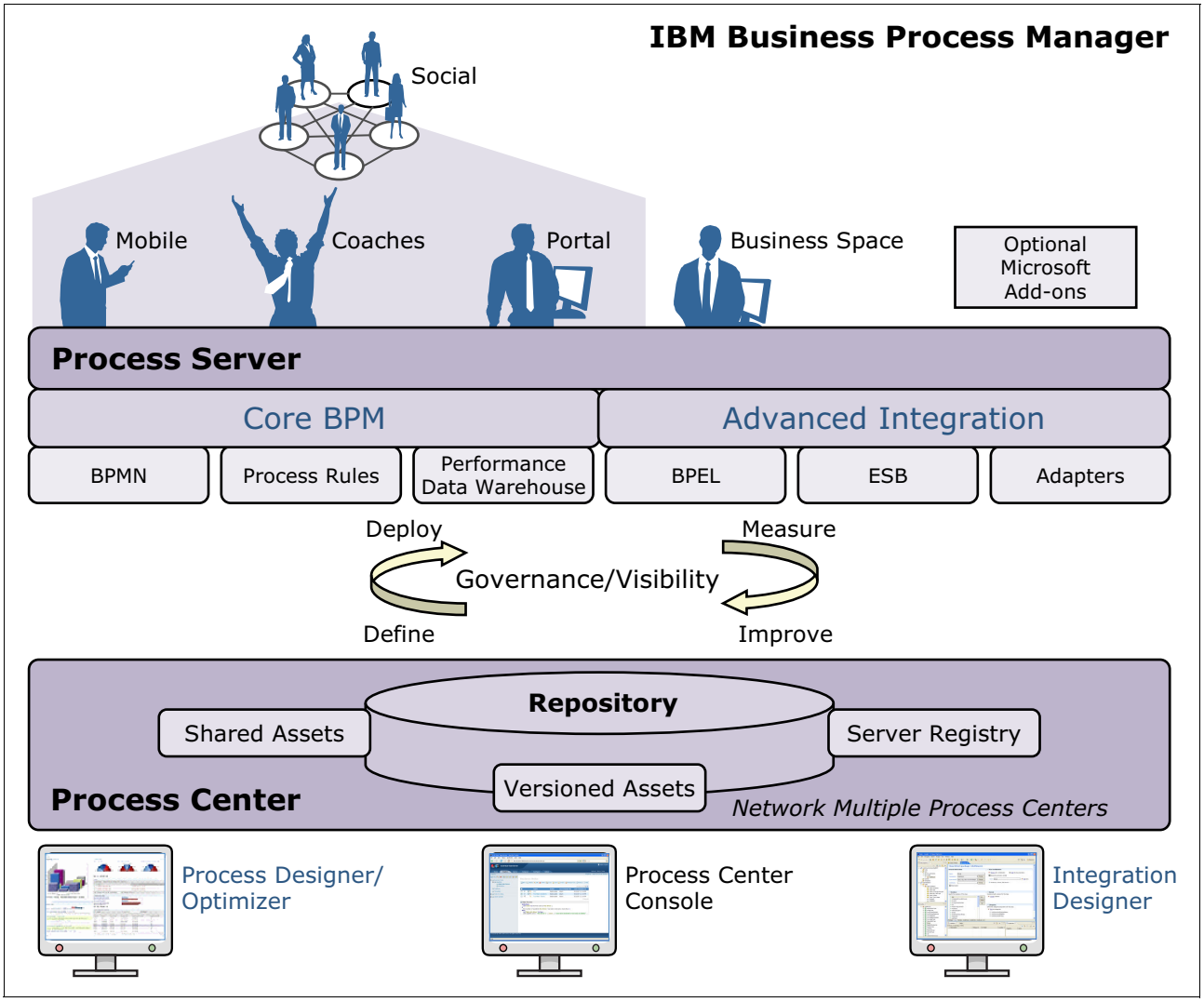


Figure 1-2 Major components of IBM Business Process Manager

IBM Business Process Manager includes the following major components:

- Process Server provides a single BPM runtime environment that can support a range of business processes, service orchestration, and integration capabilities. Because it is integrated with Process Center, you can run your processes as you build them directly in the authoring environments.
- Process Center is a runtime environment that includes a repository for all process models, services, and other assets created using IBM Process Designer or IBM Integration Designer, the IBM Business Process Manager authoring environments. It has a console with the tooling that you need to maintain this asset repository.

IBM Business Process Manager Advanced offers the following authoring environments:

- IBM Process Designer is used to model business processes that involve human tasks.
- The Optimizer is used to show the impact of the change before it is deployed, and the built-in governance capabilities enable the business to deploy changes.
- IBM Integration Designer is used to build services that are self-contained or that invoke other existing services such as web services, enterprise resource applications, or applications running in IBM CICS® and IBM IMS™.

IBM Business Process Manager also includes a set of administration tools to help you accomplish tasks ranging from installing and managing snapshots to administering processes and working with the resources in your IT environment.

The Process Center console provides a convenient location for users to create and maintain high-level library items such as process applications and toolkits. It helps provide a framework in which BPM analysts and developers can build their processes and underlying implementations. In addition, the Process Center console provides tools for maintaining the repository, including setting up the appropriate authorization for users and groups.

1.5 How is this book organized

The remaining chapters of this book can equip you on your journey toward better business agility and growth. You begin with the exercise of discovering and expressing what *ad hoc* means to your business. Then, you choose the correct style and implementation approach for your business from the various approaches that are described by example scenarios. The BES case study provides the perspective and context of one corporation's successful ad hoc strategy in which many of the described ad hoc styles are applied.

This book includes the following chapters:

- Chapter 2, "The ad hoc spectrum" on page 9
- Chapter 3, "Applying the ad hoc spectrum to your business" on page 21
- Chapter 4, "Implementing solutions for your ad hoc business" on page 33
- Chapter 5, "Ad hoc process examples" on page 53
- Chapter 6, "Case study: How Banco Espírito Santo approached ad hoc processes" on page 59
- Chapter 7, "Next steps" on page 97

2

The ad hoc spectrum

This chapter describes what it means for a process scenario to be *ad hoc*. It describes the fundamental ad hoc definitions and the basic ad hoc characteristics. These definitions and characteristics lead to a set of ad hoc styles, which in turn are enabled by a set of ad hoc capabilities.

2.1 Defining the term ad hoc

The formal definition of *ad hoc*, from Latin, means “for the particular end or case at hand without consideration of wider application.” A more colloquial definition is “one off” or “out of the ordinary.”

Now, this formal definition is applied to business process scenarios; traditional business process scenarios have historically tended to be quite structured, with each instance of the scenario providing a predictable, consistent, ordered sequence of activities. This has been a factor of the available technology and the desire on the part of many businesses to have structured, predictable operations. If ad hoc characteristics are introduced to a business process, then we are introducing capabilities for that process to have *one off*, *out of the ordinary* characteristics; that might be different for each instance of that process. Consequently, this process becomes less structured, less predictable, and less ordered. One might think that this is something that the business does not want, but, as you see next, ad hoc capabilities enable business processes to accommodate a greater number of real world considerations and thus are more applicable to the way that business actually works.

For more information about the relative appeal of structured processing versus ad hoc processing, see 2.5, “To be or not to be ad hoc” on page 18.

2.2 Ad hoc characteristics

Now that a high level view of what ad hoc might mean to a business is established, what characteristics can be applied to the notion of an ad hoc process scenario (in no particular priority)? We have attempted to define just a few characteristics that are for the most part mutually exclusive. Perhaps others might have to be added at some point but these defined next are a start. There is no formal definition of the set of characteristics to be considered when discussing *adhockness* and so this section suggests a reasonable starting point for discussion.

2.2.1 Predictability

When a process scenario is *ad hoc*, each process instance might operate in a different manner. Thus, the execution and outcome of each instance, including the tasks that are executed, the way and order in which they are executed, and the events that are accommodated, are less predictable. The set of process instances for the scenario overall has more variability. In contrast, a *structured* process executes in a predictable way.

Although a business might prefer to have predictable operations, this type of process does not necessarily reflect the real world, where unexpected things happen and must be accommodated. Consider the following examples of how businesses processes can be unpredictable:

- If a business process depends on its input data for the paths that are taken, each instance can take different paths. Furthermore, the greater number of paths and input data, the more variety (that is, unpredictability) you will see in the way that the process runs. Similarly, the business process might depend on the decisions and opinions of business users. Humans are inherently unpredictable, and the process instances are likely to reflect this unpredictability.
- If a business process depends on external events, its level of predictability depends on the arrival and order of these external events.

2.2.2 Manageability

Many business processes need to exercise significant control and governance (that is, management) over the process template and for each process instance. Processes that are a core component of the business usually need to be well controlled and well managed. Other business processes (those not part of the core business) often do not need the same levels of oversight and allow users a larger degree of control over how the processes run and are altered. Consider the following examples:

- A review process needs to be set up for a “one-off” project that will last just three months. This process is confined to a single department within an organization. After, the process will be removed. This process does not need to be managed by any central IT or business body.
- An account-opening process is defined for a financial institution with many account holder types. The process will be a core business capability, and it must be consistent across the various account types. The process must implement certain regulatory aspects for the multiple countries where the institution operates. This process must be well defined and documented, and it must be regularly reviewed to account for changing regulations in the various countries. This process needs full management.

The term *unmanaged process* might have negative connotations, but that is not the case here. Having an unmanaged process has some advantages in terms of delegating control to a wider population of users. Furthermore, you might substitute the terms *control* or *governance* for *manageability* if either of these is more palatable.

2.2.3 Visibility

One of the key reasons for implementing business process management (BPM) is to gain visibility into business process execution. This follows the notion that you cannot optimize cannot optimize or understand something until you can see and measure it. In general, an ad hoc process is somewhat less visible than a structured and ordered process. This statement is especially true of highly ad hoc processes where there is no well-defined structure or order. In many ways, the more ad hoc and unpredictable a process might be, the more you want to have visibility into it to track what actually happens for each instance. In addition, you might discover common patterns in ad hoc, unpredictable environments that warrants attention.

The term *business activity monitoring* (BAM) is also applied to visibility. BAM is an optional feature of BPM that is not necessarily applied to all types of processes. The following examples of visibility (or lack of it) are of consequence in a claims process that is implemented at an insurance company:

- Having visibility into the process instances provides a measure of the number of claims that are processed in any period. You can drill down to determine which types of claims are processed the fastest and which departments (or even business users) are the best at which types of claims. You can see the value (totals, averages, and so forth) of claims that settled in a particular period.
- A customer calls the insurance company to find the status of a claim that has not yet been settled. Without visibility into the process, you cannot see the claim status, that it is currently being processed by which claims adjuster, and that its status means that processing is delayed.
- Visibility provides information that a particular set of claims adjusters are severely overloaded. Perhaps a significant weather event has increased claims for particular group. Visibility allows you to see the situation in real time and to allocate more adjusters on a temporary basis.

2.2.4 Flexibility

An ad hoc process is generally different for each instance that runs. It can execute differently and produce a different result than a seemingly similar process instance. Ad hoc capabilities enable a process instance to vary its execution, react to external events, and be more flexible than a structured process as shown in the following example:

- An order process can react to an external event that cancels part or all of the order.
- The same process can accept changes to the order up to a certain point in the process.

  Before this point, the process is flexible, but after this point, the process is reasonably inflexible.

Order processing is one type of processing that is becoming more flexible to accommodate real-world requirements. Just a few years ago, you could not change an online order after submitting it. Now, however, an order is often flexible up to the point of delivery, and in some cases it can be changed even when it is on the truck for delivery.

Flexibility has similarities with predictability, discussed previously. You can proactively build a process to be flexible, which introduces a greater degree of unpredictability.

2.2.5 Sequencing

Ad hoc processes include a variability in the sequence of tasks within the process. Although a structured process might have alternate paths, those paths are generally well defined and predefined. In an ad hoc process, the order of task execution can vary based on knowledge worker input, rules, external events, and so on. You can extend this idea, for highly ad hoc processes, where there might not be any formal, sequenced process defined at all. You might have a scenario that is composed only of disconnected process fragments that are invoked by specific events or user actions as in the following examples:

- Prescription handling is a process where most, if not all, of the steps must be done in a particular order. Often these steps are mandated by a governing body. A prescription cannot be filled before the customer and prescriber are verified and before drug interactions are investigated. In addition, the pharmacist must review each order before a technician can fill the prescription, and then the pharmacist must verify the order's correctness.
- Certain aspects of tax return processing are independent and can be carried out in any order, or in parallel, although all of the necessary steps must be completed before processing can move beyond some checkpoint.

In some ways, this variable sequencing of tasks is closely associated with the aforementioned flexibility characteristic, because variable sequencing leads to flexible processes.

2.2.6 Granularity of ad hoc characteristics

When considering these ad hoc characteristics, it is not appropriate to take a simple binary view, where a particular process is either flexible or inflexible. Rather, you need to consider the varying levels, or *spectra*, of each characteristic. You might represent this spectra as a spider chart that illustrates the degree to which processes exhibit these ad hoc characteristics, as shown in Figure 2-1 on page 13.

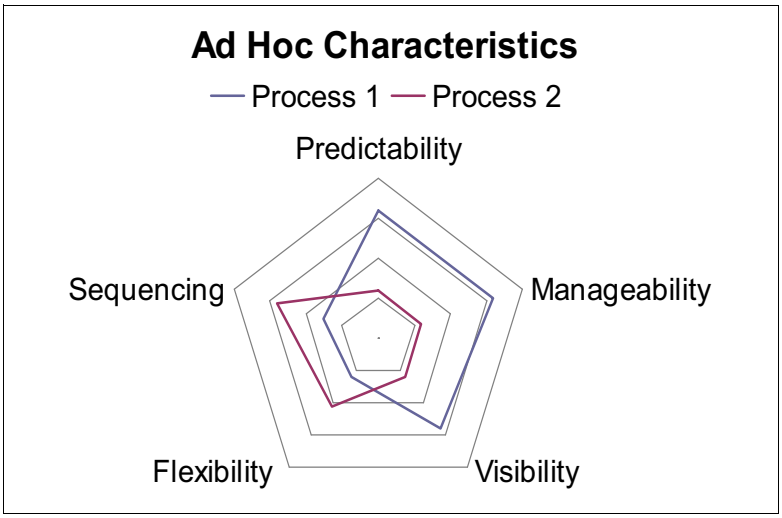


Figure 2-1 Degrees of ad hoc characteristics

In a similar way, when you think about any ad hoc aspect, think in terms of a spectrum. It is not really appropriate to think of any particular scenario as being either ad hoc or not ad hoc. Rather, there are differing degrees to which a scenario exhibits, requires, or implements ad hoc characteristics. So instead of a scenario either being ad hoc or not, take the view that it will exhibit differing degrees of the various ad hoc characteristics. This thought process leads to a spectrum with strictly ordered and structured processes at one end of the spectrum and entirely unordered, unpredictable, and dynamic ad hoc scenarios at the other end of the spectrum, as shown in Figure 2-2.

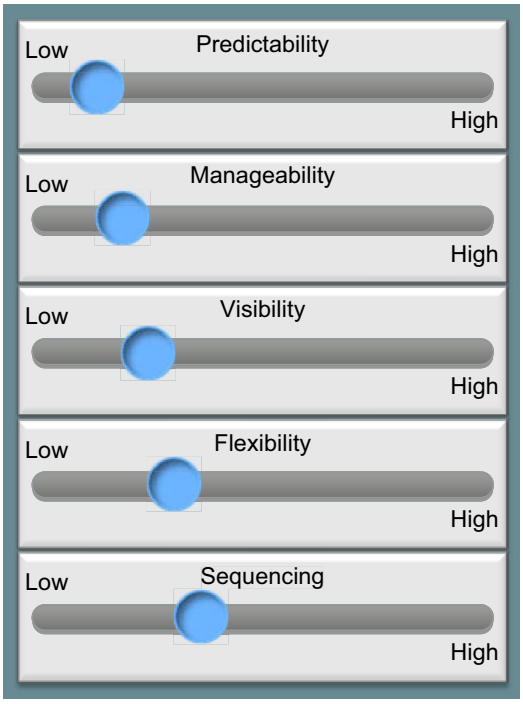


Figure 2-2 Ad hoc spectrum

2.3 Ad hoc styles

The previously described characteristics lead to a set of ad hoc styles, as detailed in the following sections.

2.3.1 Structured and predictable

Processes that have no ad hoc nature at all are simply a set of structured and ordered steps that provide a predictable sequence of activities.

2.3.2 Ad hoc by design

At first, it might seem strange to design a process specifically to be ad hoc because you are designing for (that is, predicting for) unpredictability. However, for many business situations, you can have a reasonable idea of where something might happen, or at least where you might want to check whether something has happened.

In addition, you can use BPM capabilities to execute a part of a process on receipt of some event. You can use these BPM capabilities as part of the mainline process or as an entirely separate “event handler.” What you cannot do is react to something completely unexpected.

You can design the following types of ad hoc scenarios into the process:

- Event-driven activities

  Ad hoc scenarios include the following styles of events:

  - Events that are placed at some fixed point in a process model

    One aspect of this type of event is situations where we might require some external activity before the process can move forward. Examples of this event are where the process needs some document before proceeding or some message to be received. The second aspect of this type of event is where an event is significant at some particular point in the process. The most common example is an escalation event where a task exceeds its allotted time span and you escalate that task in some specific way.

  - Events that might occur at any point in a process

    The most common example of this type of event is a cancellation event where you need to be able to support a cancellation at any point in the process, cleaning up all of the activities, backing out any updates, and terminating all threads.

- Dynamic workflow

  You can design a workflow to be dynamic and flexible using several methods:

  - The most common example is to present the knowledge worker with a choice of what will happen next. The user can select from a menu, enter a choice, and so forth. This choice enables the user to control what is executed and the order of execution based on whatever is deemed important to this particular process instance. Often, there is some level of exclusive or inclusive grouping that is associated with the choices that the user makes. For example, if you do task 1, then you have to do task 3; if you do task 2, then you cannot do tasks 4 and 5.
  - A fairly straightforward effort is to build a point in a process where there are, for example, fewer than 10 paths to take. However, when you get to a larger number of available paths, a process model quickly becomes unwieldy and an alternate style is preferable.

– You must also consider the way in which a single knowledge worker makes decisions about what to do next. There are the events and documents, but there is also a total set of knowledge workers and others with whom the knowledge worker might interact. People do not work on activities in isolation. Instead, they interact with colleagues and established experts on the task. Also, they look at the history of this task and other similar, related tasks to make better decisions. The prevailing term for this style of activity today is *social BPM*.

– External components can be used to implement dynamic workflow by providing variable information to the workflow. The common examples here are environment variables and business rules. This mechanism provides a set of components (the environment variables or the rules) that have a lifecycle outside of the process. Thus, you can change the external components without a need to redeploy the business process. Using these external components, you can alter the way that a process runs (sometimes radically) by altering these components, and sometimes even by altering inflight process instances.

2.3.3 Dynamic and generic processes

A growing trend is to build a generic business process that does not serve any specific business function. The *generic* process provides a framework that you can then customize through additional information (for example, a database table, a flat file, an Excel spreadsheet, and so forth) that configures the steps to be executed and the parameters of each step, such as the target user or users, task layout, tile, instructions, and any required attachments. The general model is to have a central, generic task that reads the additional information where each section of the additional information dictate the task's activities this time around. When a defined step is completed, you loop back and read the next section for the next task instance. This type of generic process is shown in Figure 2-3.

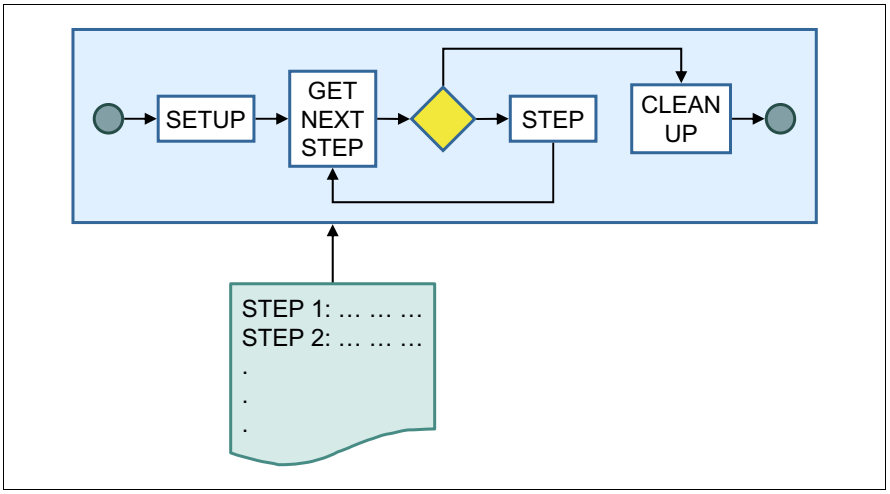


Figure 2-3 Generic process

The generic process provides a constrained set of possible activities that the process implementer wants to support for this style of unmanaged activity. If the capabilities that are required by a business user fit into this constrained set, this type of generic process provides an alternate way to provide a process definition.

However, this method also has the following primary consequences:

- This method provides a rapid way of specifying a set of process steps without IT involvement. Rapid because the business person will often have a better understanding of the file format being used than the BPM modeling environment.
- This method has the disadvantage of enabling only a constrained set of process activities, specifically those that the dynamic process implementer chooses to support.

Thus, using a generic process has the effect of enabling a process to be created, within the bounds of the generic process, without touching any of the traditional BPM design tools. All one has to do is to create the document that describes the desired process. In principle, having IT involvement is not necessary either, although having some level of interaction at some point, no matter how small, clearly makes sense. The effect is that anyone who can build the required document can build a generic process and start running it, if their requirements are supported by the generic process. With this sort of capability, you hope to widen the participation in BPM by a broader section of the business teams and begin the move towards the wider democratization of BPM.

The benefit of this approach to the business is that many simple processes that IT has no knowledge of can now be monitored and tracked, providing visibility into activities that currently are not as evident. This approach gives a much improved view of what is really going on in the business. Further, the highly used generic processes are obvious candidates for full BPM modeling and development into a non-generic process implementation.

Clarification: There is no suggestion here that avoiding the BPM modeling tools is an advantageous thing to do. Specifying a process through a file or spreadsheet is simply an alternate, and sometimes more familiar, mechanism for a business person.

2.3.4 Event-based task flows

Event-based task flows are a class of processes that have no process model. These flows are composed simply of a set activities that are linked by a set of events that occur in some order. Typically, the following scenarios use event-based task flows:

- There is no desire to use any formal process modeling or runtime environment that controls the scenario.
- The scenario is so ad hoc that considering a formal model of how the scenario operates is simply not possible (or worthwhile).

  One view of this sort of scenario is what is known as a *case* or *case management*. In this scenario, the best that you have, in terms of a process view, is what are referred to as process *fragments* that are loosely coupled together. The people involved with each case (referred to as *knowledge workers*) are the connections between these fragments.
- The process is not yet understood.

  In this scenario, the best that you can do is to begin with an entirely ad hoc view and gradually learn enough about the activities to determine whether some process model can be gleaned from it or whether the scenario is a truly ad hoc scenario, as described previously.

One challenge that often arises for this type of scenario is how you track what is occurring. With a process model, there is a place (the model) from which you can define and generate monitoring points within the scenario. For a mostly event-based scenario, you need a similar mechanism that enables you to see a view of the overall scenario.

2.4 Ad hoc capabilities

Ad hoc *characteristics* lead to a set of ad hoc *styles*. This section describes the set of ad hoc capabilities that are provided for business processes. Depending upon the particular process engine being used, you get differing combinations of the following capabilities:

- Asynchronous events

  Although it is convenient to think of a process as being an ordered set of activities, this situation rarely reflects the real world and the real way that the process runs. There are, almost always, certain things that happen:

  - In some unpredictable way, that is, they can happen at any time during the life of the process.
  - At some unpredictable time. When you get to that point in the process, the event might have happened already, or you might need to wait for something before proceeding.

  Asynchronous events provide a way to support these types of requirements. You can set up the process environment to accept events generated either inside or outside of the process instance that are passed to the process and can cause specific activities to take place.

  Asynchronous events can include the following examples:

  - Document events: When a document in a content management system changes state (added, deleted, replaced, and so forth)
  - Cancel events: When the process instance needs to be cancelled and all data and state cleaned up
  - Escalation events: When a task in a process instance is overdue or the entire process instance is overdue, you can take specific action, such as changing the priority, starting tasks with expert or senior experts, and so forth

- Collaborative scenarios

  The knowledge of how to best complete a particular step in a scenario does not necessarily lie with the person to whom that task gets assigned for any number of reasons. There are several collaboration capabilities that enable knowledge workers to contribute their expertise to some step in the scenario that is not formally assigned to them:

  - Adding comments to a process instance to help others in the execution of their tasks. Why is this ad hoc? The addition of such comments can affect the way that a particular step is executed by adding information that the task owner uses to complete the task. The outcome might have been different without the comment.
  - Instant messaging between knowledge workers can affect the outcome of the scenario by adding more information to the tasks at hand and providing instruction to the task owner on how or why to complete a task.
  - It is possible to share the UI of a task and to then collaborate on completing the task. Both parties get to see the task in their local language. When completed, control of the task returns to a single individual.
  - A history of which tasks have been executed for this process instance so far, who executed them and what data values were at that point is the Process Activity Stream. It also includes comments that other users have added to this process instance. A similar facility might be available for all process instances, though the volume of information might make this less useful.

- Analytics

  It is often useful to know what has occurred with other instances of the process or with other processes that have something in common with this process instance (for example, the same client, same product, or same company). The ability to search through a set of processes enables this sort of determination.

- Dynamic processes

  Most of the capabilities described in this section contribute to the dynamism of the process.

- Event subprocesses

  When events are received into a process, they can be handled in one of the following ways:

  – There might be inline steps in the process that are executed, which is for cases where the process is designed to wait for a particular event at some point in the process.

  – There might be an entirely separate set of steps, not part of the mainline process. This type of process is referred to as an *event subprocess,* which is a subproccess that is driven by some event arriving in the system. Although an event subprocess is separate from the main part of the process, it still has access to the process data variables.

  One key point about subprocesses of this type is how much you need to keep track of them and any dependencies between them. For some subprocess types, such as Cancel, it is obvious that the subprocess has run (as the process gets cancelled). For others, it might be important to keep track of which subprocess have run and to be aware of any inclusive or exclusive dependencies between them. In some implementations, this subprocess is an exercise for the student, and it must be manually built. For other implementations, here is significant capability for expressing this sort of grouping.

- Rules based processing

  Rules, either internal to the process or some external rules engines can be used to influence the execution of a process instance. Especially when the rule implementation is external to the process, the rule can be changed and re-deployed without change to the other components, and this can significantly alter the behavior of the otherwise unchanged scenario.

2.5 To be or not to be ad hoc

Ad hoc processes can provide significant capability for a scenario. Does this necessarily mean that implementing some set of these capabilities is better than a more structured and ordered approach that works more or less the same way every time? This topic is worthy of discussion because there are likely to be various points of view.

- Structured and ordered environments provide predictability and consistency to an implementation. Most businesses find great value in having an ordered, predictable, and consistent set of processes. These types of processes enable them to set expectations, do resource and capacity planning, and ensure that their business operations work in the same manner a majority of the time. The use of ad hoc capabilities must be kept to a minimum and the implementation of some of these capabilities must be discouraged.
- Real-world considerations might not always allow the kind of structured environment that a business might prefer. If you want to accommodate those considerations, some ad hoc capability is warranted.

- Ad hoc capabilities enable a business to accommodate the unpredictable, unordered, and flexible nature of real-world scenarios. You can have a much more flexible environment if you enable as much ad hoc capability as possible.
- Generic processes, enabled by ad hoc capabilities, enable a level of process development and innovation by a broad population within an organization. This method enables a much wider participation in process innovation than can be achieved through a primarily IT-centric approach.

Is one style better than the other? Each has compelling aspects and distinct issues. Structured and ordered processes often do not reflect the real world, but ad hoc processes, for some, represent a chaotic environment that they prefer to avoid. It is a matter of balance and applicability, not one of better or worse.

You must enable the correct amount of ad hoc processing to accommodate real-world requirements, and not just implement some capabilities just because you can. At the same time, you want to have as much structure as possible to enable repeatable and manageable operations from a business viewpoint.

In conclusion, ad hoc capabilities provide a useful set of functions, but designers and implementers need to be careful to use these capabilities where necessary and not just where the capabilities look interesting and fun to use. If you decide that too much ad hoc capability is more than the average user can cope with, you might consider making these capabilities available only to particular users (or roles) within the organization, for example supervisor or manager functions. With this pattern in mind, it might be necessary to call on this role for functions such as cancelling a process or overriding certain activities.

3

Applying the ad hoc spectrum to your business

The scenarios in this chapter apply ideas about the ad hoc spectrum to a set of industry examples. These examples illustrate how ad hoc capabilities might be applied and provide value. The objective is to show that there are examples in almost all industries where ad hoc capabilities can be usefully applied. Although the examples are from real implementations, they are described in non-specific terms.

3.1 Airline

This section describes how to apply the ah hoc spectrum to an airline industry example.

3.1.1 The problem

If you have traveled on an airplane before, you likely have experienced some sort of travel disruption. This disruption might have been a delay, a cancellation, or a lost bag (or all of these scenarios). In most cases, these disruptions originate from unpredictable events that are outside of the control of the airline, such as weather, airport traffic, or mechanical issues. In addition, in the airline industry, one small event early in the day can cause a complicated down-stream impact.

For example, a flight is scheduled to travel from Chicago to Denver in the morning. Because of thunderstorms in the area, the flight is delayed one hour. This seemingly small delay can effect many other parts of the business:

- First, consider the impact on the passengers. Those passengers traveling on the airline who have a tight connection in Denver might now miss their next flight and will need to be re-accommodated.
- Second, because the delay is in the morning, the aircraft used on that delayed flight is likely scheduled for another flight that is departing Denver later in the day. So, now that second flight might also be late.
- Third, the crew might be affected. The pilot, co-pilot, and flight attendants on the delayed flight might all have subsequent flights that will be impacted if they are late. In addition, regulatory considerations require that the crew cannot work more than a certain number of consecutive hours. So, if this delayed flight pushes them over that time, they are not allowed to be the crew on their next flight.

Although this is just one example of one disruption on one short flight, an airline might have hundreds or even thousands of flights per day, all of which are subject to unpredictable events that can impact their schedules. Traditional approaches can certainly help with these difficulties, although well after the fact. Perhaps you can think about the problem in a slightly different way by using the dimensions of ad hoc.

3.1.2 Applying the ad hoc spectrum

Using the ad hoc characteristics described earlier, Figure 3-1 illustrates the ad hoc spectrum for this airline industry process.

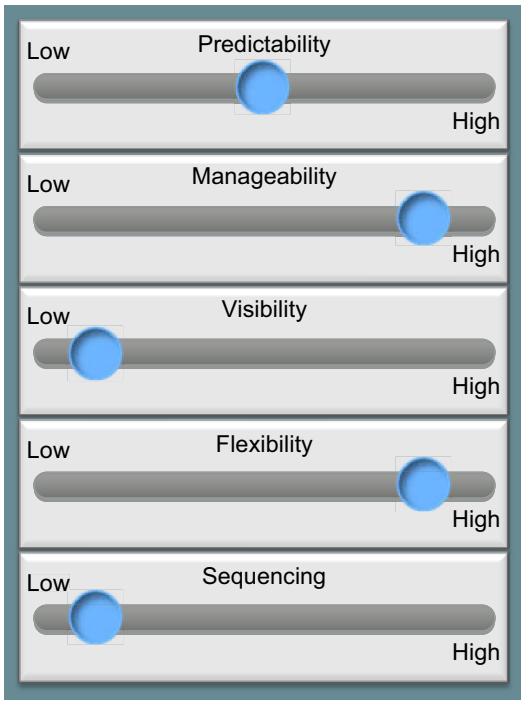


Figure 3-1 Applying the ad hoc spectrum to an airline scenario

The airline scenario can apply the ad hoc spectrum for the following ad hoc characteristics to determine the correct approach and solution for addressing its needs:

- Predictability

  Daily operations are predictable so that the “regular” operations are mostly the same or similar with little variation. However, events (such as weather) can arise at unpredictable times that cause “irregular” operations to occur.

- Manageability

  Although the airline needs to be flexible, critical factors are at play that require the process to have specific management capabilities. One area is the regulatory requirement of maximum flight hours for the crew. Another is the need to attend to passenger schedule disruption, especially for the most loyal fliers. A management requirement like this does not necessarily mean to have a strict business process in place, but rather to enforce policies at the correct time, in the correct context, and for the correct reason.

- Visibility

  The current state of this example is quite difficult to visualize given the possible disruptions and variability. However, if the correct solution were deployed in the correct way, one of the benefits is a higher degree of visibility.

- Flexibility

  To respond to the unpredictable (and inevitable) events, the process requires a high degree of flexibility.

- Sequencing

  Some pieces of this example lend themselves to a structured sequence. For example, during normal operations the flight crew going from their arrival gate to the next departure gate can be mapped in a certain number of steps that usually happen in the same order. However, sequencing the overall operations is difficult, because the first flight that is late can disrupt the entire downstream process. Instead, you might consider this as an unsequenced set of activities, each of which can consist of a small set of sequenced steps.

3.1.3 An example solution

A large North American airline was interested in addressing a similar issue. Although it was not directly referred to as *ad hoc*, it was clear that approaching their business and technical requirements required a different sort of application than typical BPM. Specifically, the airline wanted to dynamically notify its customers when unpredictable events occurred and to do so in a personalized and contextual way. In addition, the airline wanted to offer discounts and other benefits to its customers based on purchase history and frequent-flier status.

3.2 Healthcare

This section describes how to apply the ad hoc spectrum to a healthcare industry example.

3.2.1 The problem

In the context of this discussion, the healthcare environment might be characterized as follows:

- Chaotic

  There is much activity associated with a large number of disconnected subjects (the patients). Also, multiple activities might be associated with the same patient. The activities associated with patients are unstructured, unordered, (almost) unbounded, and subject to cancellation or rescheduling.

- Multi-user

  The healthcare environment includes actors from many areas, such as the attending physician, other physicians from the attending physician's discipline, other physicians from other specialties, nurses, nursing assistant, medical testing areas (X-ray, scans, lab tests, and so forth), physical therapists, administrators, and so on. In addition, most of the actors are multi-tasking among the various patients under their care.

- Not particularly IT-centric

  The various actors are caring for patients and completing hand-written charts rather than electronic charts. Most external observers do not discern any processes going on as a part of medical care.

Although everything might seem random, a clear set of activities exists for each patient. Because of the nature of the business, the healthcare industry faces the following challenges:

- Patients in hospitals are sick and unexpected things happen to them.
- Many patients in hospitals have multiple issues and each issue is handled separately. For example, broken bones are one set of problems, lacerated organs are another, and so on.
- As treatment of a patient proceeds, more information is discovered, new activities are associated with that information, and new tasks are carried out.
- Although activities (such as tests, surgery, and examinations) are scheduled in some way, a great amount of dynamism exists, because higher priority (emergent) patients continually cause the schedule of activities to change.
- All of the actors need to collaborate and communicate.

3.2.2 Applying the ad hoc spectrum

Clearly, the healthcare industry includes a great deal of ad hoc activity. Figure 3-2 illustrates the ad hoc characteristics on the ad hoc spectrum.

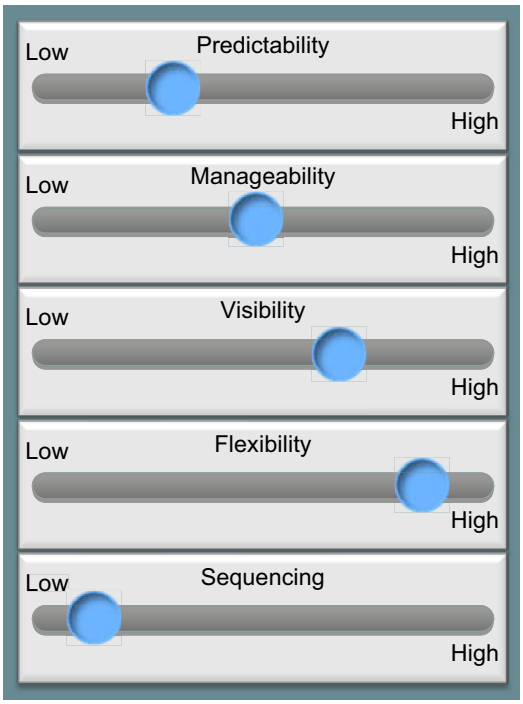


Figure 3-2 Applying the ad hoc spectrum to a healthcare scenario

If you accept that processes underlie these activities, these processes have the following ad hoc characteristics:

- Predictability

  Although you might think that the established procedures (processes) are never followed, most procedures follow a path. That said, the processes must operate in a highly volatile environment. Thus, the extreme is one of low predictability.

- Manageability

  The sequencing and predictability aspects makes manageability low for complex care situations, that is for patients with multiple, chronic problems. However, not all patients

arrive at a care facility under these circumstances. A fair degree of predictability, therefore manageability, exists for births, for many kinds of surgery, and so on. Finally, a high degree of manageability also exists for certain care functions that are often rote in nature, such as requesting a consultation, ordering blood work, and providing dietary instructions.

- Visibility

  Each activity that is done is visible and is often tracked by the organization. The challenge is to capture the events in real time and to report on the status of each process instance.

- Flexibility

  To respond to the unpredictable (and inevitable) events, the process requires a high degree of flexibility.

- Sequencing

  Because of the volatile nature of healthcare activities, tasks are added and deleted on a frequent basis. Scheduled activities are delayed, cancelled altogether, or need to be rescheduled.

3.2.3 Example solutions

The following set of examples illustrates how ad hoc capabilities are important to healthcare processes:

- A patient is admitted to general medicine with a particular ailment. The patient suffers a cardiac event and is transferred to the intensive care unit (ICU). The patient never returns to the general medicine department.
- As a patient is being discharged, the patient slips, falls, and breaks a leg. Discharge is halted, and the patient is readmitted.
- A patient is scheduled for an X-ray. While waiting, a set of higher priority patients are introduced. The X-ray scheduled for the patient never happens.
- A patient is admitted to the emergency department for one ailment. While there, diagnostic activities are added for the patient. During diagnostics, other ailments are discovered that need to be handled in parallel with the original ailment.
- A doctor needs to consult with specialists and with nurses. Real-time collaboration is needed, rather than asynchronous paging.

3.3 Banking

This section describes how to apply the ah hoc spectrum to a banking industry example.

3.3.1 The problem

Credit is an important subject for any bank. Credit approval typically happens after several steps, which are normally hidden from the customer. In most cases, the approval process uses steps similar to the following steps:

- Customer scoring from internal and external systems and entities
- Evaluation and approval at several different levels inside the organization (branch employees, branch managers, regional directors, and board members)
- Update of internal back-end applications during the credit approval circuit such as auditing, fraud-control, and compliance
- Data retrieval from several different back-end systems to help all the actors to have the most recent updated data so that they can make the correct decisions

Well-defined rules describe what might happen at each step. Based on each actor's decision, the system decides what should happen next. Those rules are also well known inside these organizations. The processes are continuously monitored and audited internally and externally to assure that everything runs as expected.

Typically, this method leads to highly structured processes, and generally people want these processes to be the least ad hoc in nature as possible. However, even in this well-structured process, non-planned events might happen. And if they do, there must be a way to manage them while maintaining coherence on all the systems and recording everything as needed for future proof.

As an example, a costumer can always cancel a credit request by calling a bank branch or using online banking. Then, a predefined set of steps can happen.

Another example is where the customer does not have all the necessary documents when the customer initiates the request. But the missing document being needed does not prevent the customer from initiating the request. The document can be attached later in the decision process. So, after a few days, the customer comes back to the branch and submits the document to a bank employee. Now the employee, who is not tasked with this process (according to the bank's organization) needs to attach that document and some data to the correct instance.

Keep in mind that these examples probably require a set of applications to be notified of the events and a set of tasks to happen. These are examples of ad hoc aspects of structured processes that must be managed.

3.3.2 Applying the ad hoc spectrum

Figure 3-3 illustrates the ad hoc spectrum in this situation.

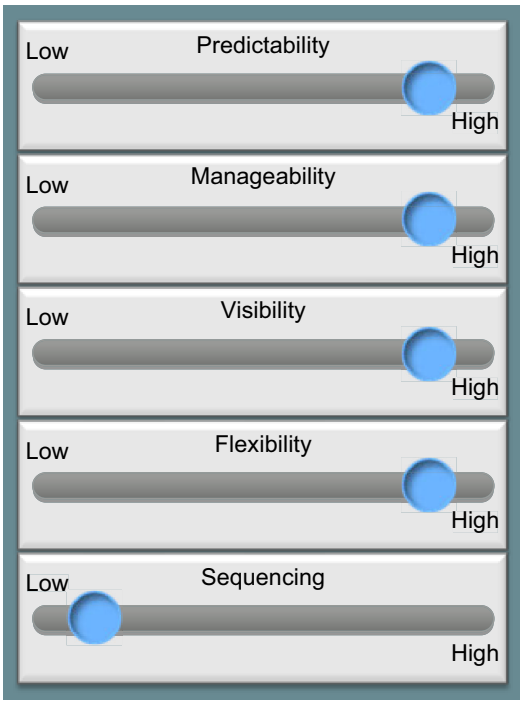


Figure 3-3 Applying the ad hoc spectrum to a banking scenario

Within this spectrum, a banking scenario uses the following ad hoc characteristics:

- Predictability

  In this example, everything other than the non-predictable cancellation is predictable. The way that a process flows inside the organization between the actors is well-documented and known.

- Manageability

  Normally, these processes run automatically between actors. Banks want a lower degree of freedom around what will happen next. Everything is well defined and the system knows exactly what to do next. So, there is a high degree of control and governance.

- Visibility

  Everything about the process is visible within the bank and follows a set of authorization criteria and rules. For each of the actors involved at the various approval levels, it is key to have visibility for all aspects, including the credit request, the customer's bank profile, the comments and decisions from the previous actors on that process, any other requests from the same customer that might be running in parallel, and so forth.

- Flexibility

  It is expected that similar processes follow the same paths and have equivalent decisions along their flow. Thus, if two customers ask for comparable amounts with similar conditions, the bank's decision should be the same.

- Sequencing

  The paths that the process flows follow are well defined and there is no degree of freedom to go outside of that process flow. Even the ad hoc example has a well-defined sequence of steps that it follows.

3.3.3 An example solution

A Portuguese bank had different solutions to cover the ad hoc aspects around structured processes. From artifacts that might be added to those processes to new parallel processes triggered by those ad hoc events, a considerable set of solutions could be used. However, after considering the need for a different approach to cover more ad hoc processes in the organization, and the decision to use a different strategy for the overall BPM initiative, a revolution occurred. A new solution was built to quickly support the different areas for both ad hoc events and for those events that could be handled temporarily in an ad hoc manner.

3.4 Leasing

This section describes how to apply the ah hoc spectrum to a leasing example.

3.4.1 The problem

Consider a situation where the vast majority of the work is manual and paper-oriented. Understanding the flow and having visibility during each iteration can be difficult. A good example of this type of process is a commercial lease application process.

In this scenario, many individuals and groups are required to create, review, and update the lease application before approving or rejecting it. These actors can include the applicant, the leasing manager, budgeting, legal, executives, and many others. In many situations, this process is driven primarily by email, phone calls, and in-person conversations, all of which can be difficult to manage or document at a high level.

Approaching this situation with traditional process discovery or enforcement of an arbitrary process might not be appropriate. Perhaps there is no repeatable process or perhaps the handling of each lease needs to be entirely unique to the situation. It could also be that the effort to discover the process up front through consultants or other means will outweigh the benefits gained through a managed process. This process might require a different type of business process capability.

3.4.2 Applying the ad hoc spectrum

Figure 3-4 illustrates the ad hoc spectrum in this situation.

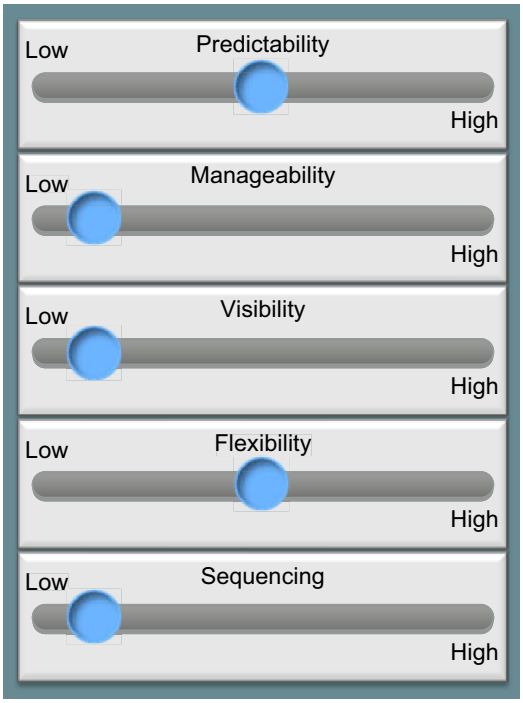


Figure 3-4 Applying the ad hoc spectrum to a leasing scenario

Within this spectrum, a leasing scenario uses the following ad hoc characteristics:

- Predictability

  Although the order and the frequency of each step can be unpredictable, the steps themselves are fairly well defined. You know who might be involved in the process and the tasks to complete.

- Manageability

  Because of the lack of visibility and the lack of understanding of the process, it is difficult to manage the current operation.

- Visibility

  Because of the level of manual input and method of communication, there is little visibility into the process.

- Flexibility

  Having some flexibility in the process is critical in this example and is likely one of the contributing factors to it still being primarily a manual effort. Much of the process flow resides in the heads of the knowledge workers. So, an important aspect is to allow for those individuals to have flexibility over the application.

- Sequencing

  Although some degree of flexibility must exist to handle the variations between types of applications, it is probable that the order in which these variations happens cannot be changed. For example, the leasing manager reviews and approves the lease application before it goes to legal or budgeting.

3.5 Corporate tax

This section describes how to apply the ah hoc spectrum to a corporate tax example.

3.5.1 The problem

Corporate tax processing is similar to personal tax processing, only bigger, much more complex, and with more serious consequences if you get it wrong. A scenario for completing a corporate tax return might look as follows:

- One tax return per taxable entity exists within a company. A large corporation might have several contained entities, each of which requires its own individual tax return.
- The team that completes the return will be dynamically constructed from a pool of consultants, depending upon the requirements of the particular corporation. This team includes the types of tax expertise that is required for the company, its subsidiaries, and the tax period in question.
- For each of the tax returns, a separate variable set of tasks must be completed and in a variable sequence that depends on details such as the following items:
  - The specific tax return details and the information available at any particular point in time
  - The progress and availability of each of the team members in completing assigned tasks where some team members might be on multiple teams at the same time, especially if they have a particular, in-demand expertise
- There will be significant ad hoc interaction between the team members and others outside of the core team to resolve issues, record status, and so forth. Generally, these interactions need to be logged so that an audit of the tax preparation activities can be kept for regulatory purposes.

So, if you build a business process to govern the completion of tax returns, this process will require ad hoc capabilities to ensure that it can be completed in a reasonable way. Each company has significant variance in the process that governs completion of the return (or returns). At the same time, a process such as corporate tax return completion must be closely monitored and audited to ensure that there is no foul play and that each step can be revisited to see how it was processed and by whom.

3.5.2 Applying the ad hoc spectrum

Figure 3-5 illustrates the ad hoc spectrum in this situation.

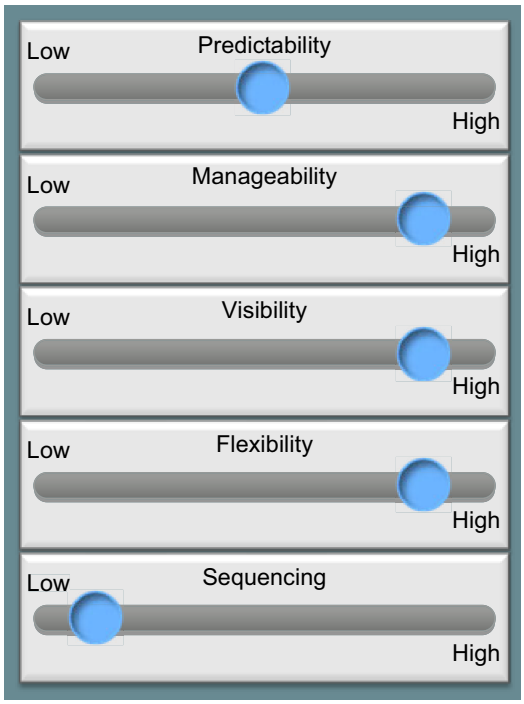


Figure 3-5 Applying the ad hoc spectrum to a corporate tax scenario

Within this spectrum, a corporate tax scenario uses the following ad hoc characteristics:

- Predictability

  You have a good feel for the set of tasks that make up the completion of a corporate tax return. The requirements are well defined. You are confident that by the end of the scenario you will have a completed tax return for the client.

- Manageability

  The team-oriented nature of tax preparation means that the process overall must be highly managed.

- Visibility

  The following factors drive the requirement for high visibility:

  – Tax preparation has a time-based factor. So, knowing the status of each team member for each member's part of the process is important.

  – The tax preparation process must be auditable.

- Flexibility

  The set of tasks that get initiated for each corporation are different depending on the tax structure of that corporation. Team members will work at their own pace and within the bounds of the information available to them at any point in time.

- Sequencing

  The highest level set of activities is reasonably well sequenced, but as you get to the more granular tasks you need the ability to initiate tasks in any sequence.

4

Implementing solutions for your ad hoc business

This chapter describes various implementation scenarios and approaches, using various styles, paths, and options, and explores the examples presented in the previous chapters. Although implementing solutions includes several factors, the essential elements are the starting point (the current position on the ad hoc spectrum) of each organization and the objective (the endpoint that they want to achieve in that spectrum).

When implementing business process management (BPM) solutions for your ad hoc business, consider the following important factors:

- An organization just starting its BPM initiatives wants to keep its processes as structured as possible and minimize any ad hoc aspects. Examples included in this chapter show how to do achieve this goal without losing sight of the ad hoc aspects.
- An organization can have an entirely event-driven solution without any process design or modeling. In these cases, the goal is to gather data from these events to identify any underlying processes and to enable a structured design of those processes. The starting point is something completely ad hoc. This chapter provides hints about extracting process information from the solution.
- A wide range of organizations are not at either extreme:
  - Some organizations might have mechanisms to support ad hoc processes or handle some of the ad hoc aspects that they need to support.
  - Some organizations will seek to improve and enhance their ad hoc implementations.
  - Some organizations will notice that parts of their ad hoc processes actually operate in the same manner every time and will want to formalize this method by modeling parts of their processes.
- The most important consideration is the attitude of the organization to ad hoc characteristics within their existing processes. For example, the organization must consider whether ad hoc capabilities are to be encouraged or kept to an absolute minimum, how ad hoc capabilities are to be handled, and what strategy can be followed to support them. Only after this attitude is established can the organization decide which adjustments to make to existing solutions or how to develop new ones.

4.1 Start with a more structured BPM approach

A more structured approach attempts to minimize ad hoc capabilities that are implemented in a solution. However, the world and organizations in particular have a lot of ad hoc needs and have to deal with them in some way. Picking this kind of approach might mean that you need to determine how to minimize the ad hoc aspects of a process. Being completely structured, with no flexibility for ad hoc aspects, might mean that you are missing the benefits of BPM.

Consider the following factors that justify beginning with a structured BPM approach:

- To avoid the challenges of implementing ad hoc capabilities, you might first (or only) choose to model and implement the most structured or ordered processes. These processes are often the most mature and well-understood processes within the organization.
- Ad hoc capabilities tend to accommodate the unexpected. By anticipating and modeling any and all possible scenarios you might get to a point where there is no need for ad hoc capabilities. (You might debate the practicality of this approach, but it is an approach that some have tried successfully.)
- In some industries certain core processes have been standardized and provide well-defined structured processes. The telecommunications industry is a good example.
- In some industries a subset of processes are structured because they adhere to some well-defined legal or regulatory sequencing.

With a structured process as a baseline, the following sections describe how to augment processes to accommodate ad hoc needs. These sections use the example of a simple bank that has implemented a credit approval process. Figure 4-1 illustrates the bank's levels of approval for this scenario.

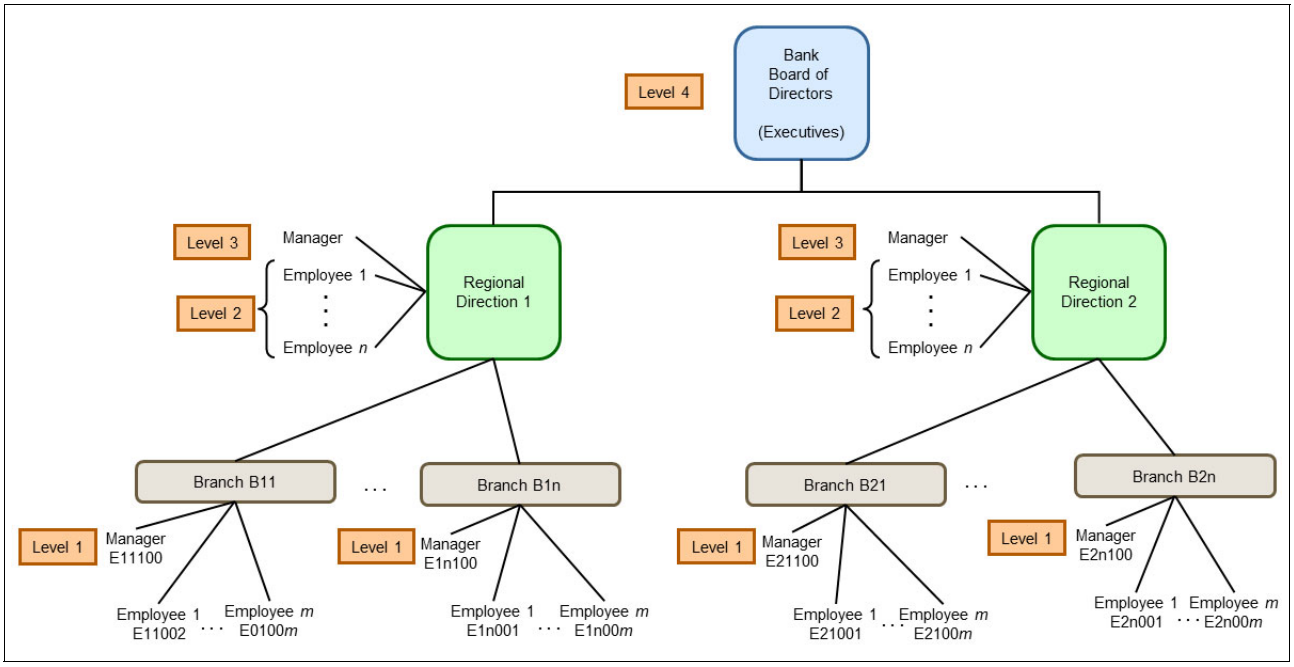


Figure 4-1 Example bank's organization

The decision-making process has the following structure:

1. The manager at each branch is at Level 1 of the decision process.
2. The employees reporting to the manager (also called regional director) are at Level 2 of the decision process.
3. The regional director is at Level 3 of the decision process.
4. The board of directors is at Level 4 of the decision process.

When a customer visits a branch, the customer submits a credit application and is assisted by a branch employee. The data is collected, and the process flows automatically through the bank's approval hierarchy following a set of well-defined rules. These rules dictate the next actors based on previous steps and the available options for each actor.

Examples of these rules are as follows:

- Requests under USD $100,000 can be decided by Level 1 (the branch manager where the request was made).
- If not decided by Level 1, the employees reporting to the regional director at Level 2 of the process can make the decision. They also have the option to consult with the regional director at the Level 3 of the process, which implies that the process can be returned from Level 3 to Level 2 for a decision.
- For requests over USD $100,000, the branch manager at Level 1 can give a non-binding opinion, meaning that the decision is either positive or negative, but cannot issue a final decision without a review by someone at Level 2.
- Requests over USD $500,000 receive opinions at all levels of the hierarchy, but final decisions are exclusively handled by the board of directors at Level 4.

Consider the following important examples for this scenario:

- An authorized user can cancel any of these instances at any moment, regardless of the stage or step, given that the customer can cancel that application, and there is no need to approve or reject anything. Cancelling an application is not just a matter of terminating the process, because there is a fraud check system for credit applications. All canceled applications are registered and classified according to a set of rules. A canceled credit application can be forwarded to a fraud investigation department.
- There must be the ability to return process instances to any of the previous levels. This can be used in several different use cases. Particularly in situations where the branch employee must attach an extra document and add data to an instance. Despite being possible, that option is not practical.

  Imagine that an application is being decided at Level 4. Delaying the decision and returning it to Level 1 so that the branch employee can attach a document is not practical. The rules need to specify that customers can submit an incomplete application but that some documents can be attached later in the process and that those documents must be attached as soon as they are available.

Those two examples (canceling an application and adding a document) illustrate ad hoc events that must be managed in the structured implementation that is used in this scenario.

4.1.1 Credit approval initial implementation

The following section assumes that the process was implemented as shown in Figure 4-2.

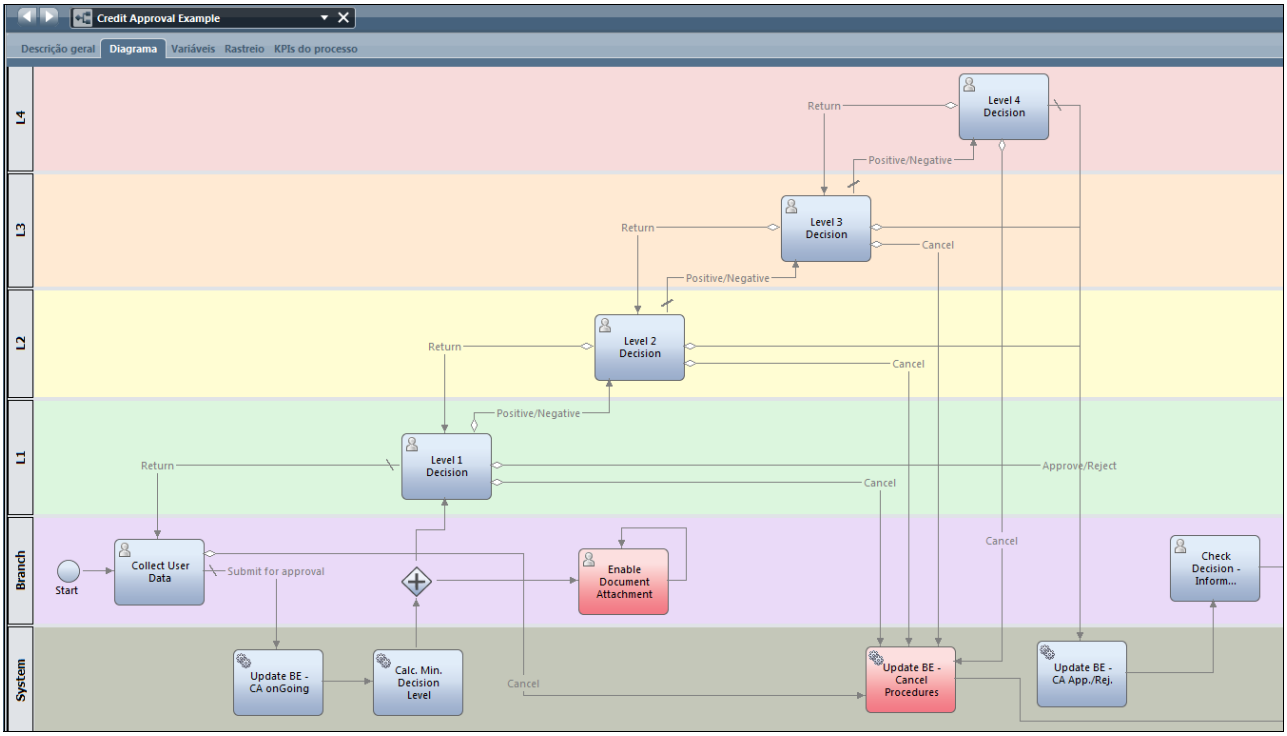


Figure 4-2 Credit approval process implementation

The red boxes in Figure 4-2 show the following activities:

- Enable Document Attachment

  Instead of moving several steps back in the process, you want to give the branch users the ability to add a new document at any step in the process. This example adds a Parallel Gateway that creates a new token, which remains in the branch users task lists while the application is moving through the approval process. When a process finishes, it terminates any active tokens. This design has the following flaws:

  - If you assign this activity only to the branch users where the application was made, the customer cannot go to a different branch to deliver the documents in question. Thus, no task is there to enable the document attachment so that something outside the process will happen.
  - If you assign this activity to employees at all branches, they all have pending tasks for all the applications (made at any branch) during the decision process. Thus, the task lists of the employees can grow huge, which does not facilitate their work. Primarily, employees are responsible for applications made at their own branch (the more frequent situation) and occasionally need to act on applications from other branches.
  - If you want to disable the ability to add extra documents immediately after the decision was made (before the `Update BE - CA App./Rej.` activity), the BPD needs more logic to synchronize (kill) the existing tokens in addition to the one that flows ahead.

- Update BE - Cancel Procedures

  Several arrows arrive to this activity from almost all the human activities on the BPD (with the exception of the Enable Document Attachment Activity). Consider that, for this particular kind of process, the number of activities (either decision levels and events

happening before and after it) is much larger. Thus, this diagram can become much more complex with flows crossing each other. After assuring all the necessary procedures, the instances flow to the terminate step.

4.1.2 Improving ad hoc aspects over the initial implementation

You can use different styles to handle ad hoc events. Figure 4-3 illustrates this process with a different design.

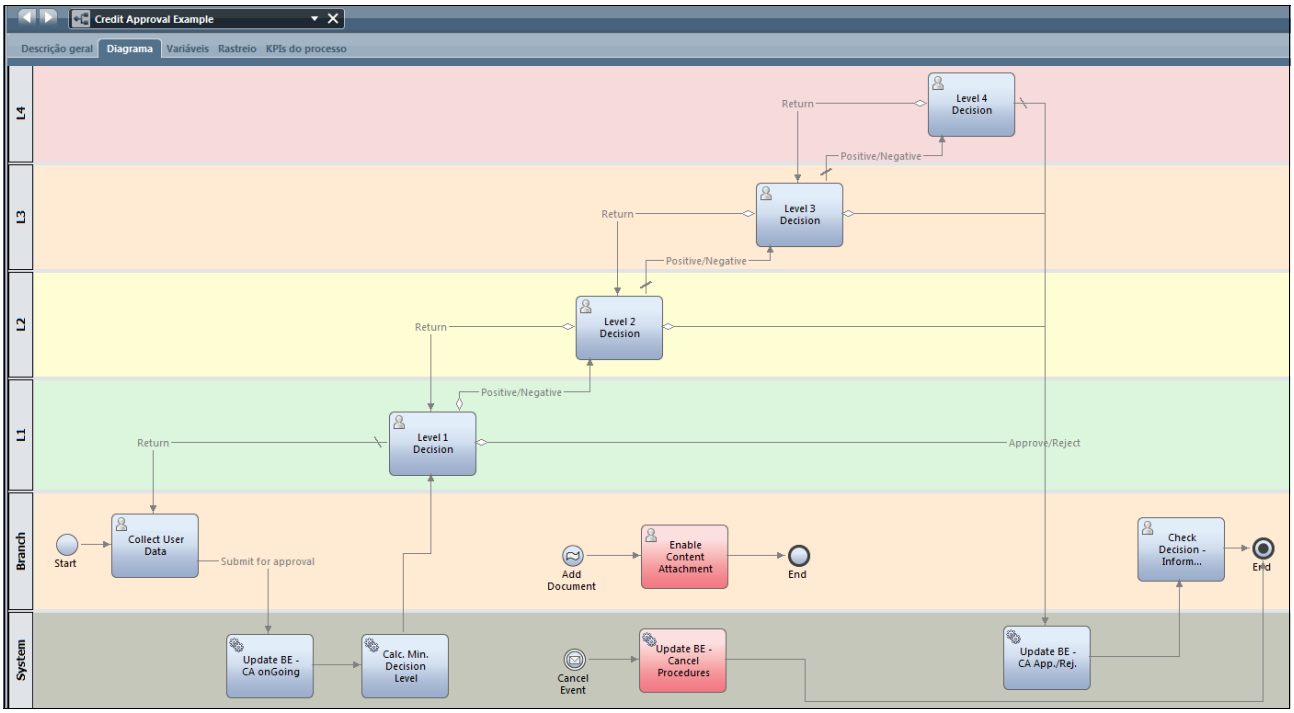


Figure 4-3 Improving the initial implementation

Note the following activities:

- Enable Document Attachment

  With the Ad Hoc Start Event, a user can go to the Portal, search for the intended instance (that can be assigned to someone else), and add a new token to it. Now the user has an activity that enables the content attachment.

  This design has several advantages over the previous design:

  – Any authorized user (ad hoc tasks can be restricted to specific users or roles), regardless of the branch or where the application was made, can add documents to the instances.

  – You can limit creation of those tasks to a certain interval of the instance's life without extra flow synchronization logic (from activity or moment A to activity or moment B).

  – Only the user who needs to add the content has a task on the task list (after adding the ad hoc task).

- Update BE - Cancel Procedures

  For this activity, you use a Message Intermediate Event that will be triggered by a undercover agent (UCA).

That method has several advantages over the previous design:

- The number of situations where the process can be canceled is not a factor. The design is cleaner and much easier to read.
- The cancellation can be triggered from any component anywhere in the environment. If you want to track the trigger of this cancellation, basically provide an input parameter to the UCA that states where it was triggered from.

These are two examples of how you can handle ad hoc aspects around processes using IBM Business Process Manager assets and functionality. By using the examples, you can enrich your structured process with ad hoc capabilities without losing its major structured design.

4.1.3 Implementing task reassignment

Consider an employee, named John, at a particular location. John spends a part of each day reviewing credit applications. He is not the regional direction manager.

- Today John started working at 8 am, and he analyzed and provided opinions and decisions about several credit approval processes.
- At almost 1 pm John is ready for lunch, but he decides to handle one more high priority credit application before leaving.
- However, immediately after starting the credit review task, John receives an important call, and for personal reasons, he leaves the office and will not return for two days.
- The task he began before leaving the office is still assigned to him.
- The next morning, the service level agreement (SLA) for the credit approval that John was working on has expired and a warning is sent to the employees working at the branch where credit was requested.
- They determine that the credit approval process is held up because John has started a task but has not completed it.
- Some action must be taken to avoid the process waiting for John to return.

The usual solution for a situation like this is to use the administrative capabilities of the BPM environment, which enable an authorized user to transfer ownership of the task from John to some other user or group. For this scenario, though, assume that you have more specific requirements beyond a simple transfer of control:

- You might want an enhanced administrative interface that provides augmented search or some specific business rules around a transfer of control.
- You might have a specific approval scheme that is imposed for certain types or classes of credit approval (for example, high value credit approvals).
- There can be a specific team that handles transfers of control in a particular way for the organization.

You can implement facilities such as this by providing an ad hoc section of the process to handle the transfer of control for any task in the process.

4.1.4 Handling ad hoc events on separated processes

In the previous scenario, you can implement an event-driven subprocess to handle task reassignment activities. But another approach is to implement an entirely separate process, which has several advantages:

- The original process is not complicated by this additional function.
- You might enable a newly implemented Task Transfer Process to handle cancellations for multiple processes instead of just the credit approval process.
- You can easily pass the responsibility for handling the cancellation to a different set of users, for example a set of central administration staff.
- There is need to take specific account of reassignments in the credit approval process.

Using this scheme, you implement a process that can be used to transfer the task that is currently assigned to John (or any other user in similar situations) and provide a help desk team to support the process, as follows:

- This process passes a key to the instance (or a lookup capability), the name of the currently assigned user (John), and the name of the new user or group for the reassignment.
- After verifying that there is a task being executed by the user in that process for more than one hour, it asks the help desk user to choose one of the following options:
  - Return the task to the original list of users to whom it had been assigned
  - Transfer the task to a particular (different) user
  - Reject the request
- The process uses the BPM system API to reassign the task and records information about why the reassignment took place. This record gives all the information to anyone querying the reasons for a reassignment at a later time.

This method can be a cross process, meaning that any task for any process can be transferred following this pattern.

4.1.5 Extending the task transfer process

At present, you do not have many more benefits than you might have by using the built-in administration capabilities. But, you can do so much more.

The help desk user is responsible for reassignment of a task between owners as described. You will add an approval step, so that no transfer is made without an approval from some senior person. This step is important for credit approval because there are steps that might be subject to fraudulent activity.

This process provides a useful audit capability to show when and why a transfer was made and by whom.

Assume that the bank's rules are as follows:

- A request for a task transfer can be sent to the help desk by any employee. Typically the requests are sent by colleagues of the employee to whom the task is currently assigned or from the branch where the original request was made based on some set of SLAs.
- The requests must be approved by some other employee that has a higher level of approval than the current user, unless that process instance is already at Level 4, in which case the request is accepted by someone at the same level. If sent without this authorization, the request is rejected and returned to its creator.

- Before sending the request to the help desk team, the employee asking for the transfer must send a request to a user who can authorize the transfer and indicate to whom the task must be transferred.
- The authorizing user can accept that request or can indicate an alternate assignee. However, the authorizer can simply reject the request without giving any alternative. In this case, the requester is responsible to resubmit the request for approval from the same or another authorizer, with a new alternative assignee.

The process of authorization requests to transfer tasks can be as simple as an email exchange between all participants with mail logs that provide auditing. However, this type of solution can be challenging to audit and maintain. To accommodate this, you add the approval processing to the start of the Task Transfer Process as follows:

- The list of people who might create a transfer request is unclear. Ultimately, any employee might do it.
- The list of people who will authorize transfer requests is unclear. The authorization must be from a higher approval level. There are two choices here. The system can determine some suitable, ranked person to provide the approval, or the approver might be explicitly named.
- The number of iterations between the creator and the authorizers is unknown and will depend basically on the decisions made by the authorizers.

Figure 4-4 illustrates this process.

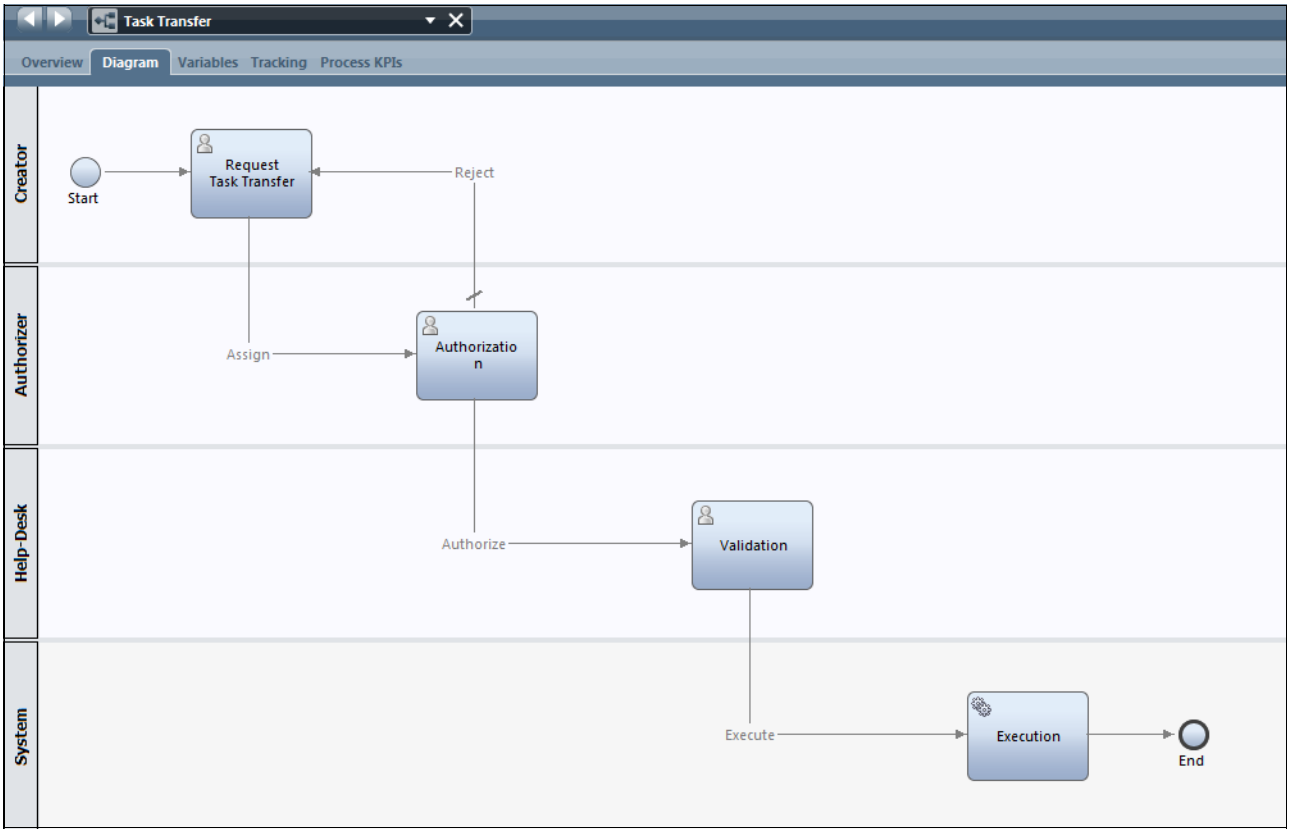


Figure 4-4 Task Transfer process

Now, instead of a set of activities that might have been ad hoc—driven by email, instant messaging (IM), phone calls, or corridor conversation—you have provided a trackable, auditable, and visible process to handle them. The process also provides a structure to

enable better understanding of what is happening with this set of activities. At some later time, you can choose to modify the process, perhaps making it less ad hoc. Alternatively, if the first implementation serves the purposes of the organization, you can keep it exactly as described. Regardless, you have a process where you can keep record of these authorization requests. In addition to auditability, you can measure SLAs for the help desk and the other actors (creators and authorizers).

There are other advantages of implementing such processes on a BPM system. Assume the scenario where the authorization process is supported by simple exchange of email or IM between the parties. Note the following aspects:

- Typically, when an authorizer accepts a request, the reply will include *OK* or *Authorized*. But this approval is less likely to be made available to the help desk, making it more difficult to properly track approvals.
- The help desk team will receive requests that are not correctly authorized, and they will have to reject these request, causing unnecessary resubmission.

The BPM implementation improves the following aspects:

- After being properly authorized, the process can be forwarded automatically to the help desk.
- The help desk receives only pre-authorized requests. They just do a validation. and if in doubt, request clarification from the requester.

Both cases improve the SLAs for both the Task Transfer Process and the process that it is supporting (Credit Approval). So starting from a small and mostly ad hoc process, you ultimately have a structured, ad hoc process that supports something that normally is seen as a simple administrative task.

4.1.6 Conclusion

The objective to follow a structured process model is quite reasonable, but it is challenged by the emergence of ad hoc activities. You can address these ad hoc aspects directly within the process using the capabilities that are provided in BPM. Alternatively, you can compartmentalize by separating off the ad hoc aspects into their own processes and keep the core process as structured and clean as possible.

Your choices are somewhat limited by the particular ad hoc activities that are required; there will be instances that are addressable only within the process and that is reasonable.

These examples are fairly straightforward and are intended to illustrate the challenges that occur within a real, practical implementation. There are many others examples, some of which are complicated and involved in their solution. The key point is to be aware that solutions to ad hoc challenges are available within the BPM environment and can be used to solve these challenges.

4.2 Using events for a more ad hoc approach to process implementation

Consider a situation where a structured process does not exist. Or imagine a situation where you do not know if there is a structured process. Organizations frequently face these kinds of issues, and often do not pursue BPM as a viable solution for their business needs because of the following challenges:

- Time or cost does not permit doing a full process discovery.
- The process spans many groups, organizations, or even companies, so visibility of the activities is limited.
- Steps and tasks can happen in a different order for each iteration.
- The initial value might only be in gaining visibility into what is already happening.

Each of these points is a valid reason for taking a different approach to BPM. In fact, these challenges are good examples of why ad hoc processing is important to the enterprise.

4.2.1 A simple example

This simple example is of a person's morning routine. In the morning, a person performs certain regular tasks but perhaps not in the same order every day. In addition, one person's morning routine likely differs from that of another person's morning routine, although many of the steps are the same.

The morning routine

Imagine that you want to model the Morning Routine Process. You can identify the following possible steps that might or might not happen for a given person on a given day:

- Brush teeth
- Go for a jog
- Eat breakfast
- Get dressed
- Take a shower
- Check email

If you have a way to detect the occurrence (or absence) of each step, you can start to document your observations. In this example, you assume that you are observing the behavior of three separate individuals. Your monitoring on day one might yield the results shown in Figure 4-5 on page 43.

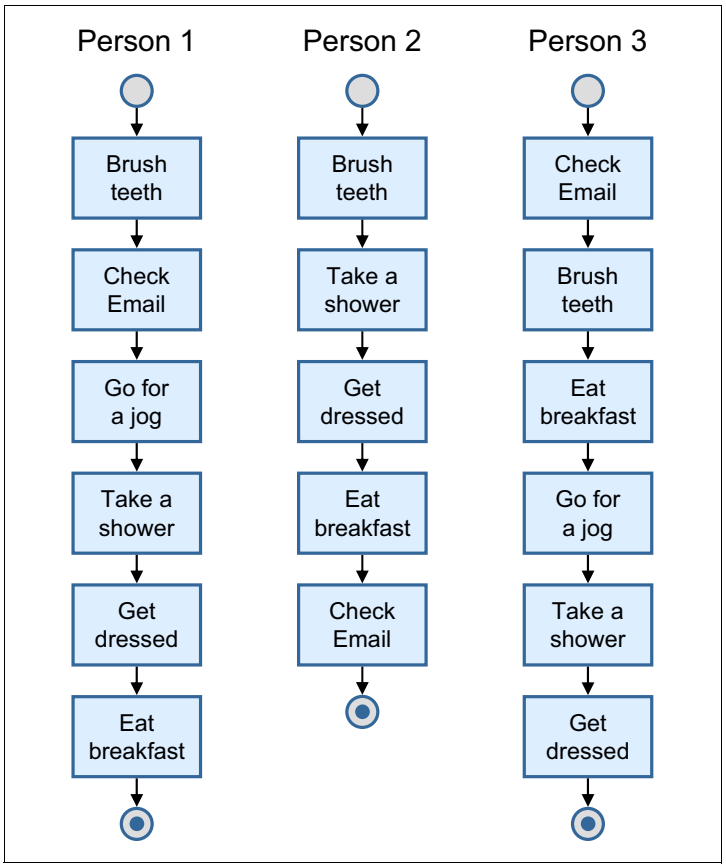


Figure 4-5 Morning routine

From even this basic set of information, you can make observations:

- First, all three individuals followed a different sequence of events.
- Second, you can see that the complete set of steps is not always followed, such as Person 2 skipping the morning jog that day.
- Third, you can start to see patterns emerge, such as getting dressed always follows taking a shower (although clearly the more data you have the more confidently you can make that statement).

Another aspect of these observations which is extremely important is the element of time. Knowing exactly when things happen in addition to the order is a key piece to this approach. In this morning routine example, assume that you can capture when each step started, as shown in Figure 4-6.

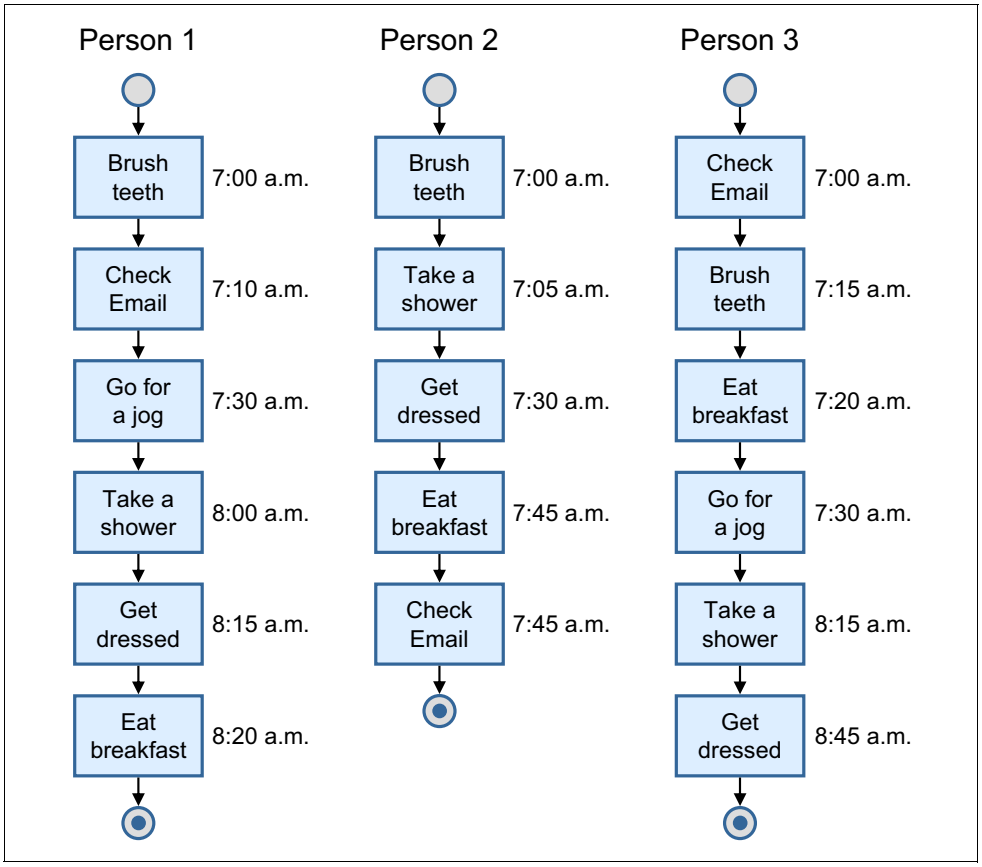


Figure 4-6 Morning routine with start times

You can now make additional observations about these routines:

- First, it is clear that Person 2 completes the routine much faster than the others.
- Second, you can start to see how long each step takes and can possibly start to understand and improve on some of the times. For example, it appears that Person 3 takes much longer to shower than the others.
- Finally, you can also start to understand the nature of the process for each individual. For Person 2, you can assume that eating breakfast and checking email happen in parallel rather than sequentially.

The evening routine

Similar to its morning counterpart, you can imagine a certain set of steps occurs during the evening routine:

- Eat dinner
- Watch TV
- Brush teeth
- Read
- Go to sleep

Figure 4-7 shows an example of what you might discover if you monitor these behaviors for the same three individuals.

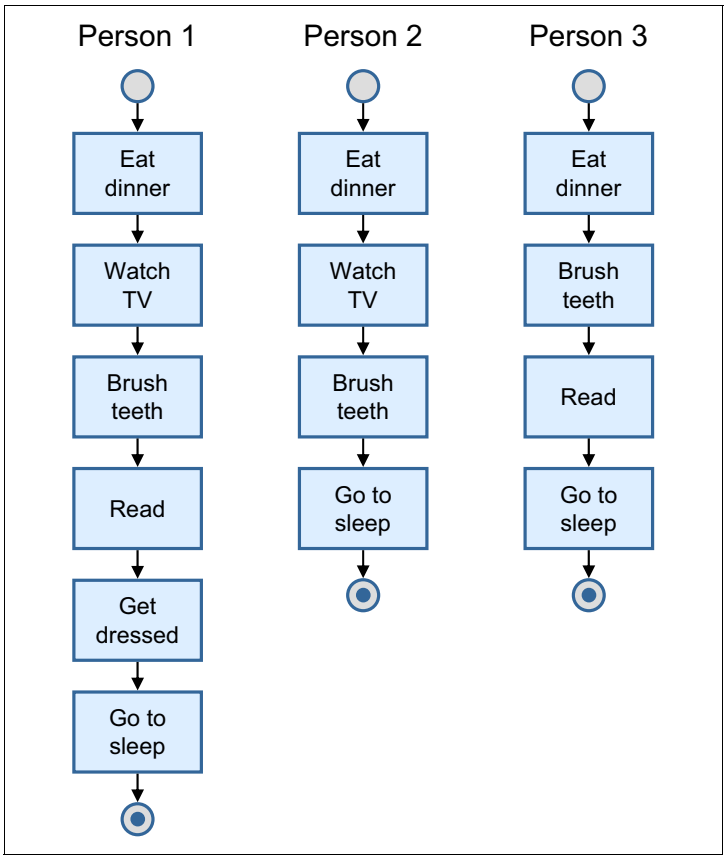


Figure 4-7 Evening routine

In contrast to the morning routine, although not all of the steps are always taken, they appear to occur in generally the same order each time. You can determine what steps always happen and use those steps as core components of a flow. Thus, you can potentially start to model a more structured process around these activities.

4.2.2 Examples in business

This concept of using events to discover, manage, and drive business process can be extended to apply to many real-world business scenarios. Two excellent examples are discussed in Chapter 3, “Applying the ad hoc spectrum to your business” on page 21.

The first of these is the airline scenario. This airline example reflects reacting to changes to the business (events) in a flexible way. In addition, because the process now consists of these independent, decoupled decisions, it is no longer restricted to a certain order of processing. Therefore, if one plane lands before another because of a schedule change, the business process can adapt and respond accordingly using the most up-to-date information.

The second business example describes a leasing approval process. In this situation, the leasing approval process is a fairly consistent, predictable business process. Because of a need for understanding, the process can be initially instrumented before discovery takes place. By collecting information about the process through events, the business can manage certain areas (one step taking too long) and can begin recording and documenting the general flow of the process to model and optimize the process in the future.

4.2.3 Instrument, and then manage

Even with the simple example of the morning routine, there is evidence that you can gain visibility and efficiency simply by observing what is already occurring. Then, you can possibly react to certain instances where the time between steps is longer than normal. In addition, this reaction can be done without changing the way the current process is running.

In business, for process improvement to take place, you must understand and document the way things are done today. If that does not happen, how will you know if you have improved? Ad hoc business process is no different. By instrumenting the key events in an organization, a non-intrusive event processing tool can be implemented to gather information about these events, create better visibility, and start to add event-specific rules to sense and respond to potential situations.

4.2.4 Instrument, and then discover

In the example of the evening routine, a sequence of steps can be discovered from a seemingly unordered set of events. This fictitious scenario allows you to extrapolate this process to situations in business where the same conditions exist. Imagine a purchase approval process that is done mostly manually. Although you might not initially understand the order in which the events occur, you will likely know the important steps, or milestones, that must occur (initial request, management approval, budget approval, and so forth). Without interfering in the existing flow you can instrument these steps, and over time determine if there is a predictable, repeatable process occurring.

Often, organizations begin with this approach and will incrementally evolve to a more traditional business process environment after the process is better understood and quantified. There is a natural transition from an event-based solution to a solution that is both event and process-based. In fact, it is often recommended that using both allows for a heterogeneous processing environment able to support any kind of business process.

4.2.5 Conclusion

There is a need for multiple entry points in the adoption of business process management. One traditional approach is to document and discover the process prior to attempting anything that touches the existing, running environment. However, many cases can benefit from detecting the events first and then using those events for visibility, discovery, and enforcement. This ability to incrementally grow the application and to incrementally learn about the process also is a key aspect of ad hoc business process management.

4.3 Extending BPM support to all processes in the business

When you look at the processes that exist within an organization, you tend to focus on those that stand out. The previous chapter focused on airline scheduling, credit approval (banking), and tax processing, but there are myriad others that you might consider. What tends to be forgotten are the less visible micro processes that are usually small, informal, ad hoc in nature, and often built to be short-lived, and then discarded.

This set of processes is often constructed with email, IM, phone calls, or even personal visits as the interaction mechanism and uses constructs such as spreadsheets to transfer data between participants. Routing is entirely ad hoc and is to whomever is available by the interaction mechanism that is available at the time. Although these micro processes are

small, they are generally numerous, and you might represent the distribution of processes in the organization as shown in Figure 4-8.

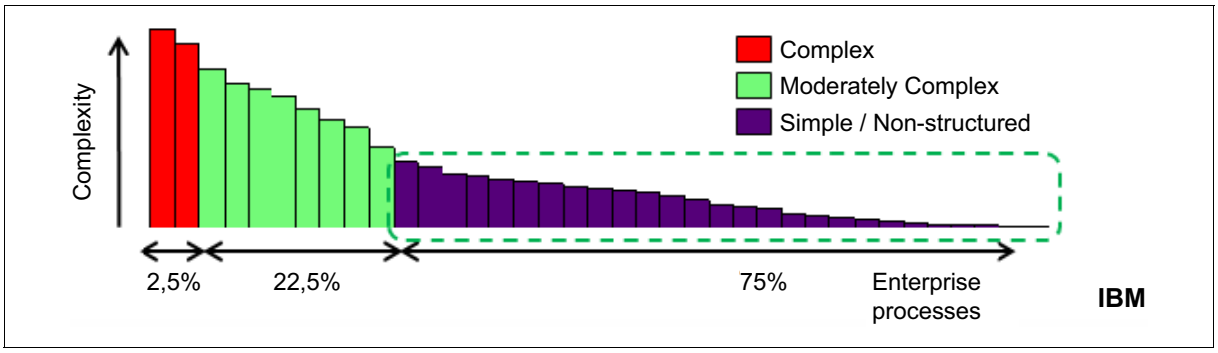


Figure 4-8 Distribution of processes in an organization

The processes that you tend to focus on are in red (complex), but those in green (moderately complex) and purple (simple, non-structured) tend to get left to their own devices. So, what if you used a different approach and tried to find a more formal solution for the “long tail” of processes on the right side of the diagram. This approach has the following main objectives:

- Provide rapid support for the process that includes some level of self service.
- Have a simple BPM implementation rather than no implementation at all that provides the opportunity for visibility into the process.
- Allow for ample scope for improvement within the first implementation.

This section highlights the benefits of such an approach and provides suggestions about possible implementations, mainly using the ad hoc facilities that are available in BPM.

4.3.1 Main benefits of this kind of approach

Here are the main benefits of this approach of extending BPM to all processes in the business.

More complete coverage of the total set of processes

Providing at least some basic level of support for the maximum number of processes, with the resulting gains in control and management is generally preferred over the alternatives. Having some implementation supporting a process can help to more easily measure SLAs and check the historical data about its flow within the organization.

Figure 4-8 shows the relationship between the well known, more complex and sophisticated processes within the organization and the numerous simple processes. Although the first objective of investing in a BPM system is to address the organization’s more critical and significant processes, providing a minimal level of support for the long tail does provide some value.

Improved pipeline and demand management with less pressure

When starting a BPM initiative, an initial set of processes are typically in the pipeline, which causes a challenging timetable for implementation. As the processes are implemented, there is often a rapid growth in the pipeline and no more capacity to implement. There will be budget constraints, limited availability of skilled teams, and so on. This implementation leaves a potentially long line of processes waiting for implementation and the promise of process improvement.

If you can find a way to provide this at least minimal level of BPM support for processes accompanied by some version of self-service then departments can have a way to implement the simpler processes that make up the long tail. Your self-service aspect might even lead to a reduction in the load on BPM teams.

Provide a wider range of BPM implementations and improve overall success rates

Many implementers of BPM, despite best advice to the contrary, choose large, complex projects for their initial implementations as the opportunity to show grand success and return on investment (ROI) for their projects. This approach has significant risk and, in some instances, leads to significant disillusionment with the BPM approach if these projects are not successful.

If, in parallel with those complex implementations, you can provide support for the long tail processes, then you have the opportunity to spread the risk, show quick wins, and provide balance with the other projects.

Start small with less well-known processes with basic implementations that can later be enriched

One aspect of long tail processes is that either they are not well-defined or there is no general agreement on their definition. Instead of trying to describe and model the process in detail, a minimalist implementation might suffice for the initial stage. This can provide valuable monitoring information, showing how the process actually unfolds. You are following the styles described previously: instrument, then manage and instrument, then discover. See 4.2, “Using events for a more ad hoc approach to process implementation” on page 42.

Take advantage of the BPM infrastructure to give rapid ROI

The acquisition and setup of a BPM solution, including all the necessary software and hardware components, is not a trivial investment. The more processes that can be supported with this environment the better, and the more the cost is spread across the processes. If you can provide an environment where a greater number of processes are supported quickly, there is less pressure on the small number of complex processes to see ROI so rapidly.

4.3.2 Implementing a solution for generalized support of processes

Suppose you have a scheme to address the primary set of processes within the organization. These are the processes that are of primary importance and they will follow the regular process development methodology. Now, assume that a second set of processes did not get addressed for several reasons: lack of time, resources, capacity, or a good design. You might categorize this second set as follows:

- Complex, sophisticated processes, generally involving both human steps and system integrations, that you want to implement in the regular way but were not addressed. (These are the red and green processes in Figure 4-8 on page 47.)
- Less complex processes, often with highly ad hoc natures and without any structure or order. Often these processes consist only of human tasks; there is no system integration. (These are the purple processes in Figure 4-8 on page 47).

The objective is to find a way to support these processes. Consider a generic, ad hoc implementation model. This model relies on the knowledge workers who are executing the process steps to provide the information required to progress each process instance to its next step. What might be the minimum set of information required to make this routing work?

You can start with the following minimum information:

- Forward to another user or set of users and provide a list of options with the following special cases of this style:
  - Forward (return) to the previous participant
  - Forward to the process instance creator
- End the process instance (finish)

So, your generic process consists of a simple model with a loop around a single human task, as illustrated in Figure 4-9.

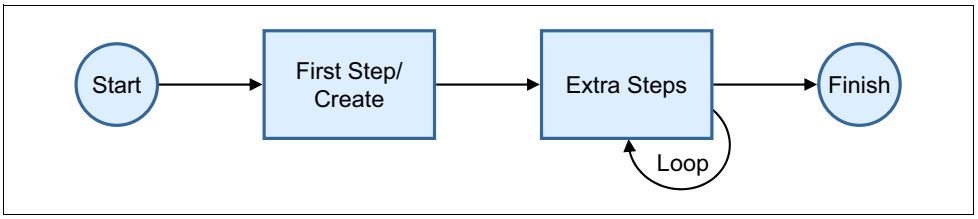


Figure 4-9 Loop around a single human task

Consider the following important aspects:

- One part of your set of processes is complex, involving more than just human tasks. For a scheme like this to work, the knowledge workers involved in the process must assume the responsibility of handling the systems integrations in some other way, outside of the generic process.
- What is the value here? Although it might seem that you provided something simple here, you gain a key capability in addition to email, instant messaging, phone calls, and so on: visibility into the process. Simple, human tasks can report on status, wait time, execution time, which user, how many steps in each process instance, and so on.

  Most organizations have no good data on these types of processes, and this visibility into the process enables the organization to start to get a handle on what happens in these processes. When you have that visibility into the process, you can start to manage and optimize. Another advantage of having this information is that when or if you decide to have a more formal implementation of a particular process, you will already have information for how the process executes.

Consider the following characteristics that you might implement:

- To implement a model sufficiently generic that can support any process in the organization, it will not have an application-specific data structure. Each user is allowed to create an instance of a process, textually describe its request or need, and forward it to anyone else. So the most that you might expect is a comment variable, implemented as a string.
- You want to enable documents to be attached to the process instances, through the organization's Enterprise Content Management (ECM) environment or in some more ad hoc way.
- You can add a timer event that enables a notification to a supervisor if the process SLA is exceeded. The supervisor might be a predefined user or dynamically configured for each process instance.
- You can provide a set of other UIs, such as these examples:
  - Process creation about where the initial process setup is performed
  - Task expiration which is provided to the supervisor when a task expiry notification is generated

4.3.3 Two distinct approaches for generic processes

Consider the various styles of generic processes.

Ad hoc template working as base model for processes

For this style, you create a process application that serves as a template for other processes. The generic process works as the basic pattern for the implementation of those that follow it. The creation of a new process from this template is straightforward: you create a duplicate of the process template with a new name, and then provide only a few customizations, such as these examples:

- Change the window that contains the process name, giving greater context to the users.
- Change or set the value of the process SLA.
- Set the user identified as Supervisor.

You can quickly create processes from the template, supporting the organization's needs. After some amount of time using this template, some additional customization is likely necessary, for example adding items:

- Adding new fields to the UIs, perhaps as a result of system integration
- Adding a specific task to execute at the end of the process by users of a particular department followed by an integration with another application

These might be capabilities that you always knew about or they might be items that you learned about from capturing monitoring information for this process over time. For example you might observe that for the majority of process instances, a particular task is always executed at the end of the process. So, you can change the process in question, incorporating these changes and enhancing your process that started out as a generic template.

Generic engine

For this approach, instead of making copies of a model, you assume a single process implementation on which all generic processes will run. This approach implies that you will have a mechanism to customize each instance of the generic process engine so that it provides the required runtime characteristics. For this to work, you need the following key components:

- A generic process application on which the generic processes will be instantiated
- A process instance configuration mechanism

  Because each process has specific settings, instantiated on a single model, these settings must somehow be described and stored. You must specify the following information:

  - The name of the process to be inserted into the UI
  - The name of each step in the process
  - Timeout value for the process, which can be for the entire process or per task
  - The supervisor user ID

  Many options are available for this data store. You might use a relational database, a flat file, a spreadsheet, and so on. Suppose that you use a database. In this case, a table might hold the information and each process instance is provided with the name of the table as some form of initialization parameter.

- Configuration utility

  So that business users can configure this information, you want a utility that populates the chosen data store. In addition, you might add capabilities such as who is permitted to configure a new process instance type and enable or disable for a process type.

4.3.4 Advantages and disadvantages of each approach

The template approach, instantiating that same template for each process individually, has the following advantages:

- Individualized processes

  The process is distinct from other processes. Simple items, such as having a saved search that is shared by the users, are straightforward. Also, any problem with this process does not affect any other process copied from the same template.

- Ability for process evolution without losing the core features (and origins)

  Incorporating new features and adding to the flow is straightforward and does not affect other processes while maintaining its history (previous snapshots, active instances, and so forth).

- Because the process is copied from a template and then (possibly) customized, you need skilled process development resources to build the coped process.

The generic engine approach offers the following advantages:

- Shorter time to deploy new processes over the engine

  The descriptive configuration through the configuration utility will normally take less time than making a copy of a process application, customization, validation, and promotion between environments. Further, t no skilled process development resources are required to build the configuration.

- A new feature added to the engine is immediately available to processes

  For example, the construction of a new SLA analysis scoreboard, integration with an ECM system (enabling content store and retrieve), or addition of default documents attached to a process instance are immediately available to all the processes that are executed on the engine. Some of these changes require additional configuration.

- Maintaining uniformity of behavior among all processes

  This might qualify as an advantage or disadvantage depending on the objectives. If you want all processes to be uniform, having a generic process engine is ideal. If you want the ability to have distinct processes, then uniformity becomes a disadvantage.

4.3.5 Conclusion

This section started by describing two different but complementary approaches. You can develop high priority processes using a structured modeling approach and handle the remaining processes in a generic, ad hoc way, even if the basic nature of the process really needs a more structured approach with complex capabilities such as systems integration. You gain several important advantages from the generic approach, in particular gaining visibility into the large population of ad hoc (long tail) processes that are present in most organizations.

5

Ad hoc process examples

This chapter provides the following examples of ad hoc process patterns:

- Well-defined processes with ad hoc actions
- Each instance of the process flow is user-defined

5.1 Well-defined process with ad hoc actions

The Message Events provided by the Business Process Modeling Notation (BPMN) provide a convenient mechanism for modeling ad hoc patterns within a well-defined process. The example in Figure 5-1 shows a business process diagram (BPD) that uses event subprocesses to implement ad hoc patterns that might arise during the flow of a well-defined process instance.

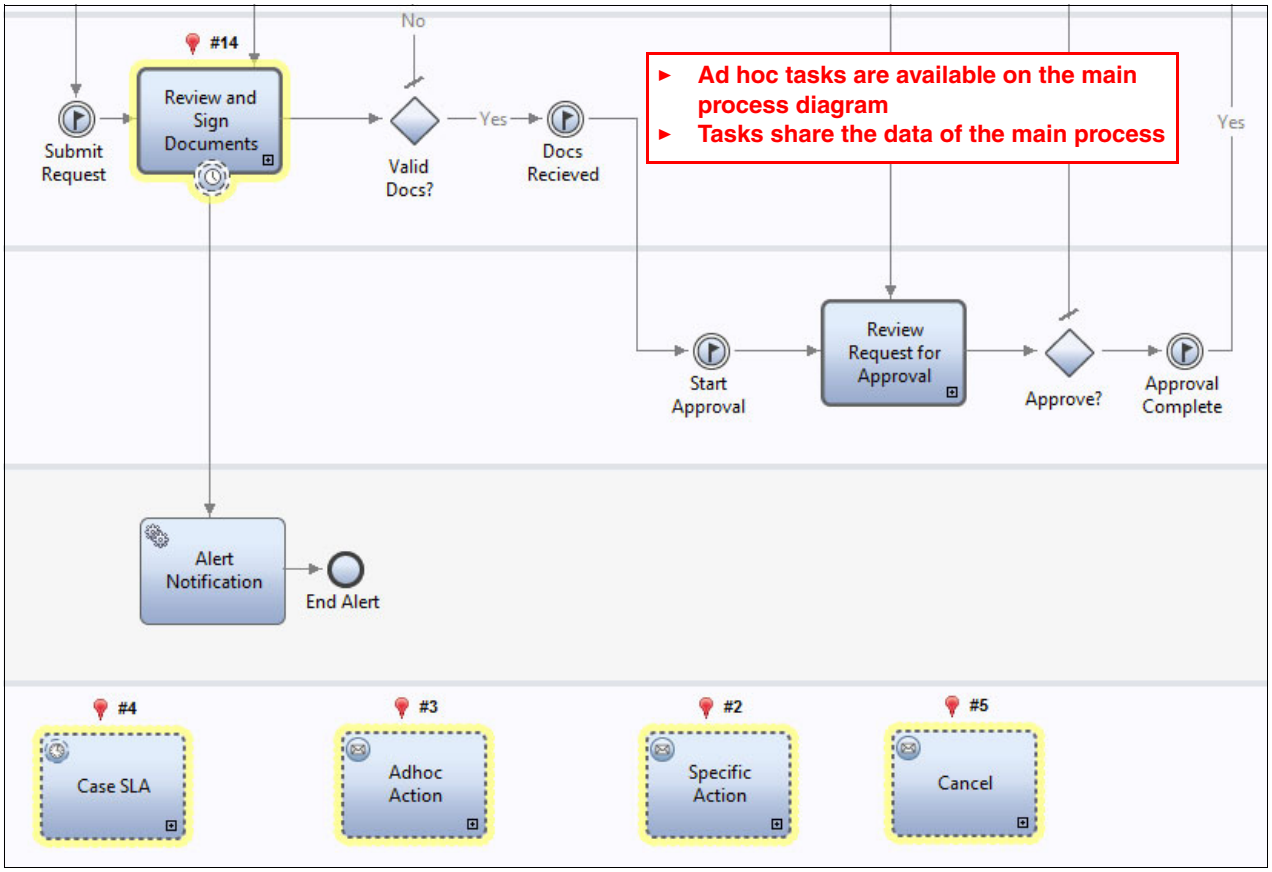


Figure 5-1 Ad hoc actions within a well-defined process

Figure 5-1 shows the BPD within the Process Inspector view of IBM Business Process Manager. The process inspector uses the yellow *halos* and the red *tokens* to show active components that are waiting for events. In this diagram, and in the regular path through the process (shown at the top), the highlighted components at the bottom of the view are event-driven subprocesses, that can be invoked by a suitable event at any point in the life of the main process. The ad hoc activities are implemented inside the Event Sub Processes, allowing as complex a set of activities as required to be implemented.

This is a good time to introduce the concept of *share business objects*, which are important because your well-defined process activities and your ad hoc activities typically refer to the same set of process instance data. For example, an allowed ad hoc action might allow a customer service representative to update a contact phone number. It is important that this data change is immediately visible to the main process flow.

Figure 5-2 shows a business object that is defined as a shared object.

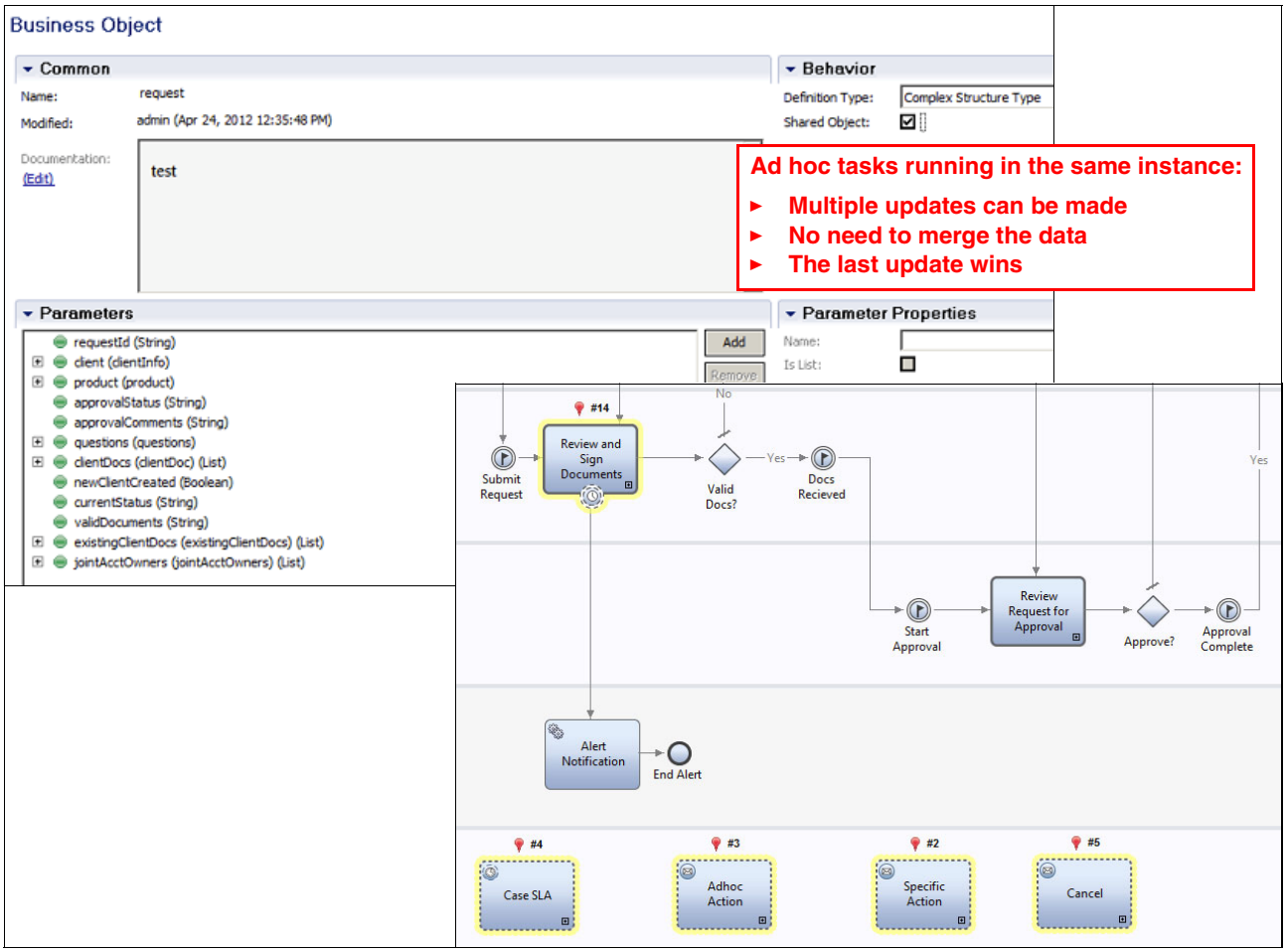


Figure 5-2 Defining a shared business object

Variables that are defined using shared objects are automatically shared for the activities where they are used. Automatically sharing these variables allows the activities to simultaneously update data without having to manually merge the data into the object. Using shared objects is a preferred practice in places where one or more ad hoc activities are using the same business data objects.

The BPMN also supports the Intermediate Message Event (IME). Using this event is another way to asynchronously wait for an event that can happen at anytime. Figure 5-3 shows an example of using an IME to wait for receipt of a document.

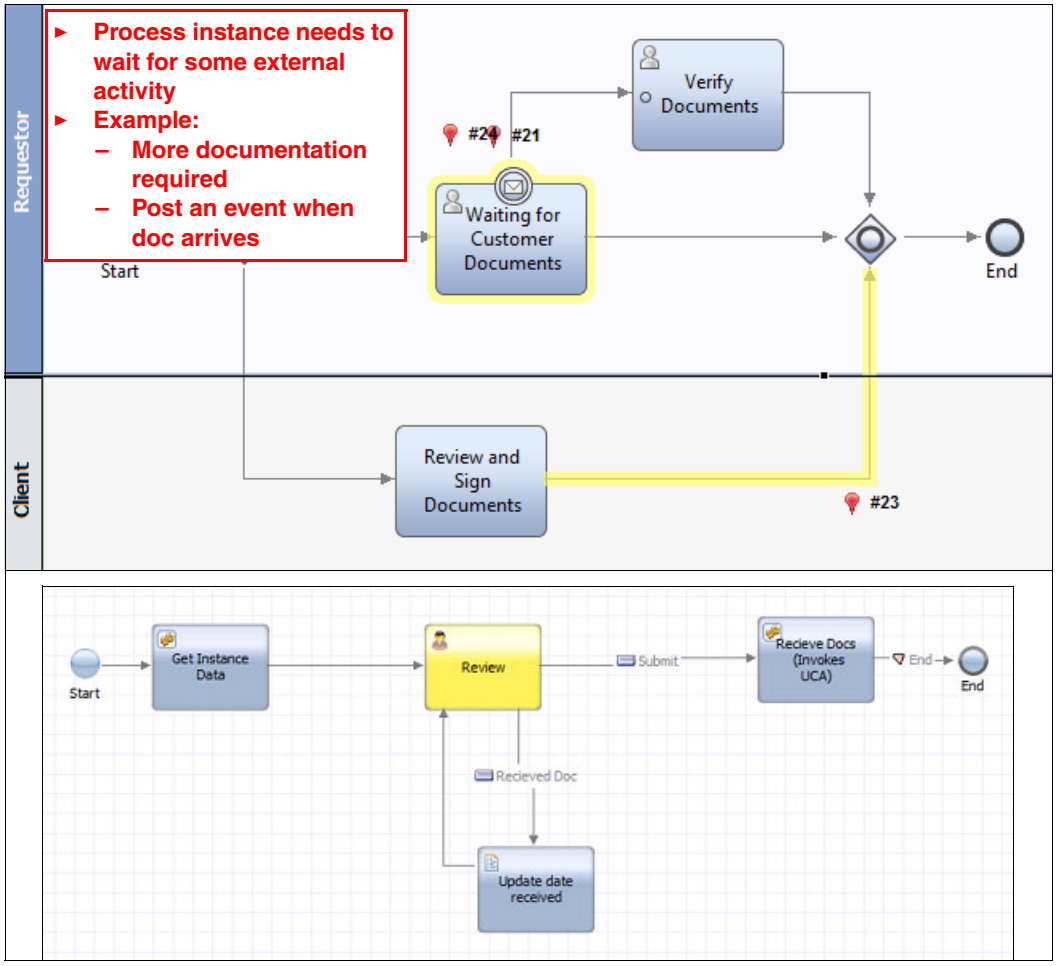


Figure 5-3 Using Intermediate Message Events to wait for an ad hoc activity to complete

In this example, the process is waiting for the receipt of the proper documentation. An ad hoc activity is implemented that allows the user to attach a document. This activity also sends a message to the process, which enables the process to continue. Figure 5-4 shows an example of the interface for this activity.

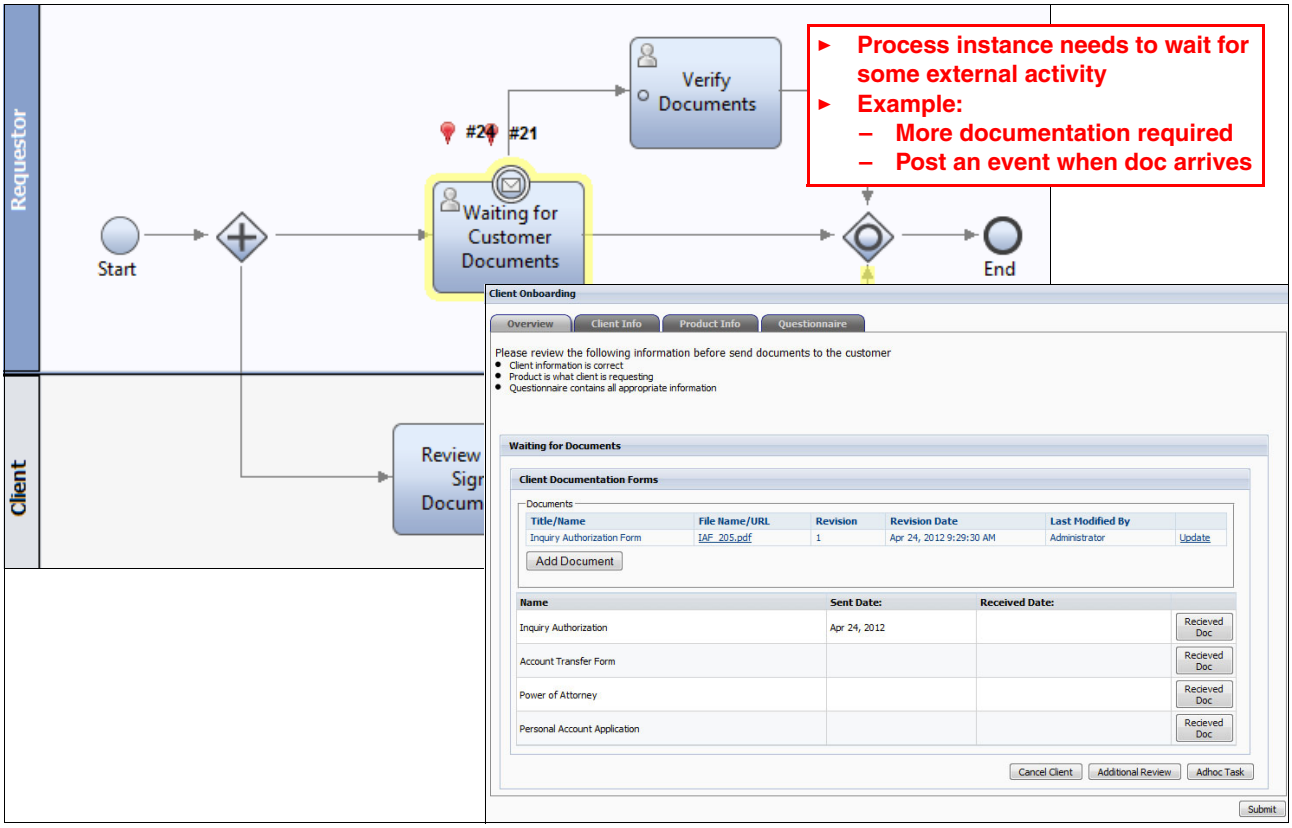


Figure 5-4 User interface used to send events to a process

5.2 Completely user-defined process

Processes that are completely ad hoc can be implemented by a BPD that acts as a *state machine*. Rather than existing as a picture of a well-defined process flow, the BPD in this case is a picture of a generic process that knows how to interpret a randomly defined set of steps, as shown in Figure 5-5.

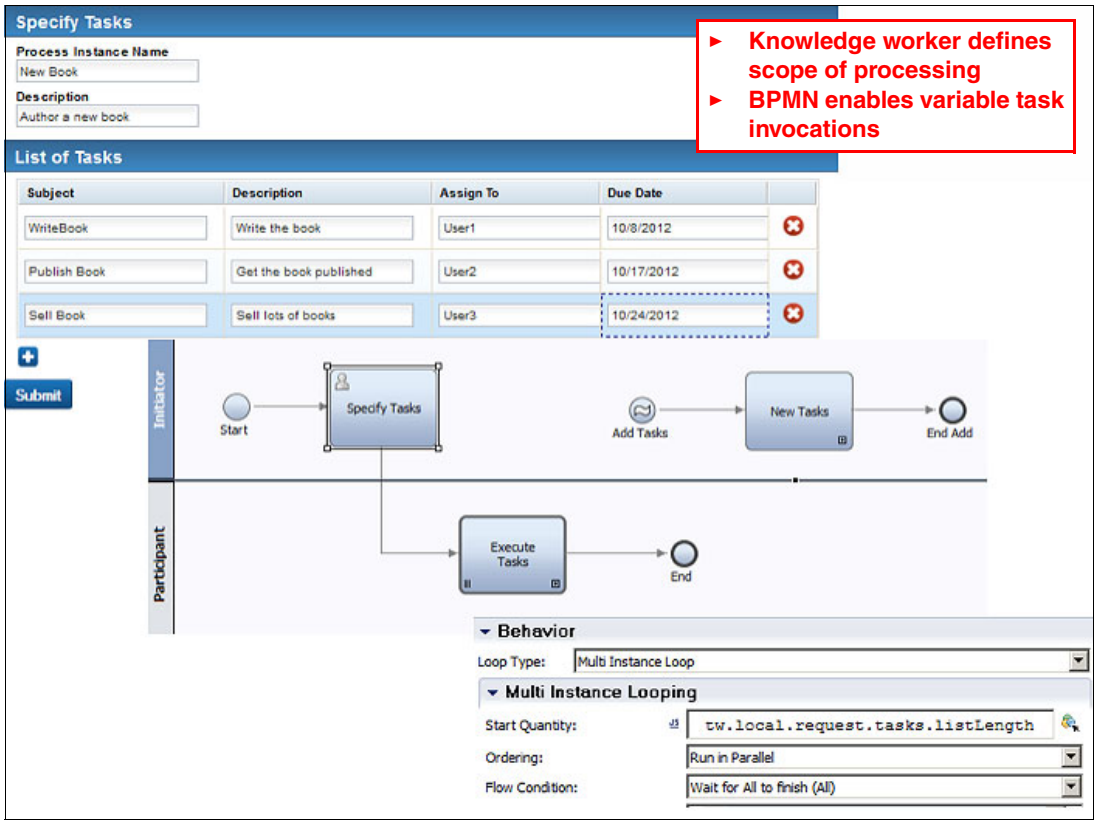


Figure 5-5 Process to execute completely user-defined process

6

Case study: How Banco Espírito Santo approached ad hoc processes

This chapter provides a case study for how Banco Espírito Santo (BES) approached ad hoc processes.

6.1 Overview

Banco Espírito Santo (BES) has more than 140 years of history of creating value for shareholders, clients, and employees. It is the largest private financial institution in Portugal by market cap, with an average of 20.3% share in the Portuguese banking market. The foundations of BES trace back to the nineteenth century (Figure 6-1).

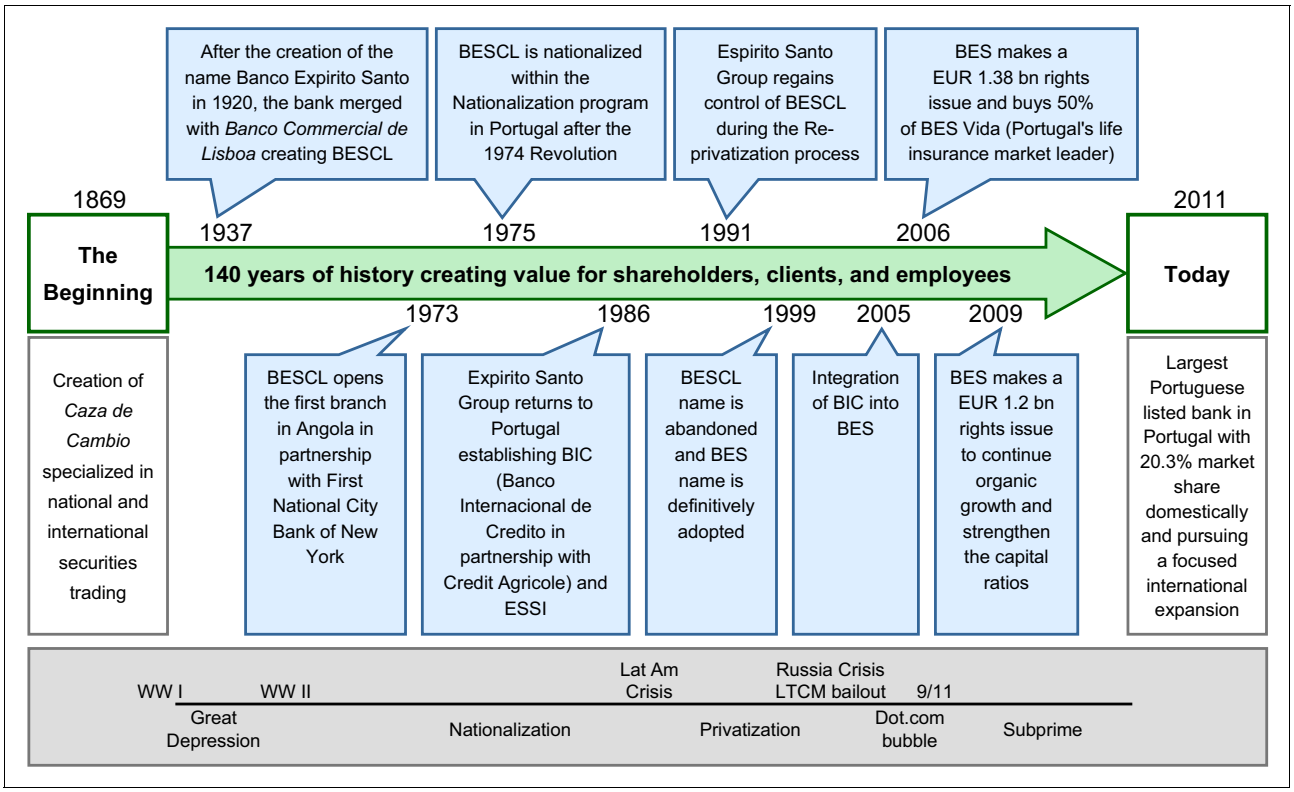


Figure 6-1 BES history

BES was founded in 1869, with a unique story of organic growth, a strong franchise, and a wide international presence in 25 countries and four continents, with special focus in the Portuguese speaking countries (Figure 6-2).

Figure 6-2 BES international presence

6.2 The BPM path followed by BES until 2008

BES started its business process management (BPM) journey in 1998, when the products in this domain where known as *Workflow* solutions. After a proof of concept with Staffware in 1999, BES licensed IBM Lotus® Notes as the tool to support future development.

Mortgage was the first large scale core process and was deployed in production in 2001. Then, BES licensed IBM WebSphere MQ Workflow and developed the following processes:

- Discount Letters (domestic) in 2002
- Discount Letters (international) in 2004
- Mortgage Servicing in 2005

These processes had excellent acceptance throughout the company, providing high value to the lines of business involved and to the project sponsors. Further, the projects revealed the importance of such solutions for the company. The success of these projects was such that the high-level sponsors within the bank were willing to continue to invest in the implementation of more processes to support the various business lines, with priority given to the most important processes.

On the downside, however, these early processes had long development cycles, despite the considerable size of the development teams. Thus, implementation costs were high. Also, IT could not develop processes in parallel. So, it was clear that the bank had a throughput problem.

6.2.1 BES Digital and the BPM revolution

In 2006, the company decided to review its overall strategy for BPM and launched a program called *BES Digital*. The objectives of this program were to identify issues with the current process, specify the issues, and then define new goals for the future.

Figure 6-3 describes the major issues identified in 2006, both from a business and technology perspective.

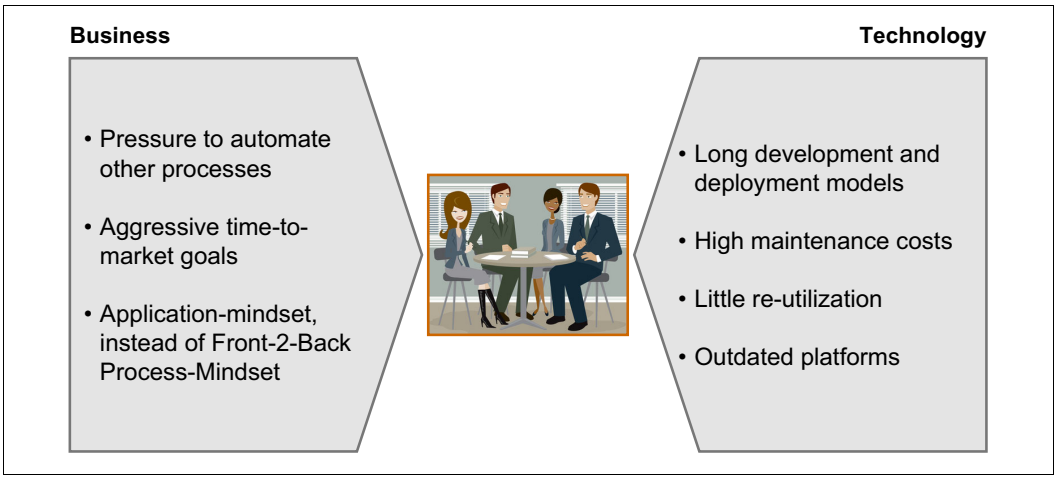


Figure 6-3 Business and technology issues

BES Digital also identified the following set of goals:

- Improve operational efficiency
- Reduce operational risk
- Improve service quality
- Gain agility
- Automate, integrate, and coordinate the following items:
  - End-to-end processes
  - Consistent business information
  - Improve time-to-market
  - Reduce costs
  - Reuse
  - Maximize business and IT capabilities

After joint discussions involving both the business and IT, the conclusion was that to achieve the goals, choosing a new development platform would be necessary. The bank analyzed more than 15 products and solutions, including pure-play BPM products, Enterprise Content Management (ECM) products that included a significant BPM component, and Enterprise Application Integration (EAI) products that had followed a BPM approach. After several meetings and presentations to get more knowledge about these products, BES issued an request for proposal (RFP) to three companies. After analysis of the responses, clarifying meetings, and a proof of concept, BES decided to license the usage of IBM Lombardi Teamworks®.

Thus, in early 2007, BES started a new phase in its BPM journey. The teams working with previous technologies were trained with Teamworks (at that time in version 5.5.5). Rules were defined for the project development teams to assure that they were aligned with the current preferred practices so that there would be coherence across the parallel development lines.

The teams were built and aligned with the project (process) pipeline, and the infrastructure was created for the supporting environments (development, Q&A, and production).

In June 2007, BES deployed the Personal Credit Approval process, which was the first production deployment for a process made with the new platform (at that time Teamworks 5.5.7). This deployment was followed by the deployment of several other processes, such as Credit Card Approval, Complaints, and so forth. The speed of development (and time to value) was higher than before. So, projects became less expensive and the throughput for processes deployed in production was higher than before.

This process of development and deployment showed a significant improvement compared to the past. In the two years following, the development team grew, following the principles of agile development, and many more processes were implemented.

6.2.2 Good, but not good enough

As the development team grew, however, the BPM teams became “victims” of their own success. The added value that the implemented processes brought to BES were so obvious that the demand for the development of new processes increased even more. Although there was a greater capacity to deliver new processes, it was clear that this capacity was still not sufficient. The way the development strategy was set would not be sufficient to cover the needs that were identified in the time required.

Figure 6-4 illustrates the processes that were developed, since the BPM journey started, until end of 2008.

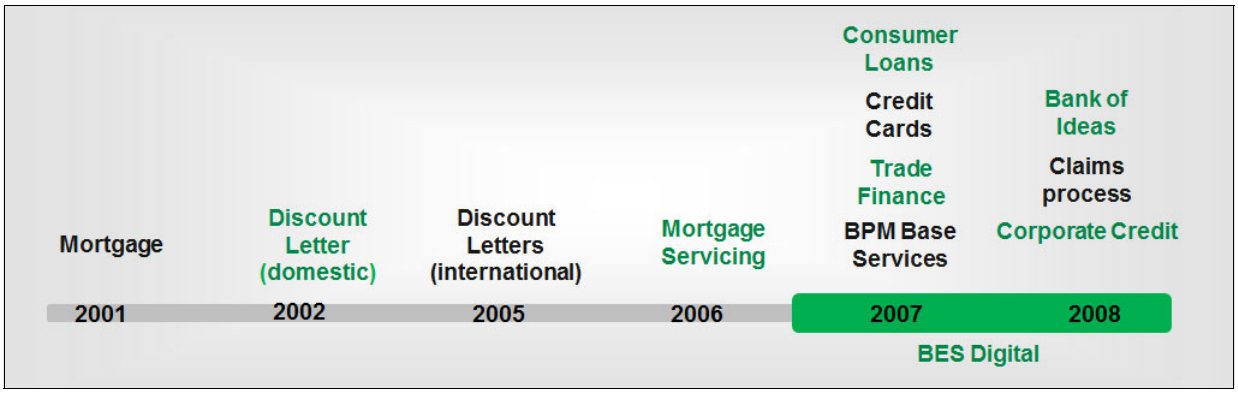


Figure 6-4 Processes developed

Despite the increase in the ability to develop and deploy new processes, demand significantly exceeded capacity. It became clear that it would take many years, still with high development costs, to cover all the needs. So, again, BES decided it was important to review its BPM strategy and identify alternatives that would address and resolve these constraints.

6.3 The turning point: Vision and strategy

Upon reflection, BES defined a new vision and strategy for BPM starting in 2008. There had to be even more business involvement in the process modeling stage and more use of the deployment capability. Ideally the business should be autonomous in the development of simpler processes, using for this a library of reusable services, provided by IT.

It did not make sense for BES to apply the same methodology in the development of complex and structured processes to that applied to the more ad hoc or simple processes. If BES could get quick support for all the processes in the enterprise, several benefits might be expected:

- Faster complete coverage and support for all the processes (improving the delivery capacity)
- Better pipeline and demand management with less pressure
- Better knowledge about lesser known processes with basic implementations that can later be enriched
- Better advantage of the investments made in the solution needed to support the complex processes

BES had all the motivations to follow this kind of approach, covering a higher number of processes in the enterprise as quickly as possible.

Assuming that not all processes have the same complexity, the processes were classified into three levels. This classification is made according to a set of criteria, including these items:

- Flow complexity
- Integrations with other applications and processes
- Process data and UI complexity

Table 6-1 illustrates how these criteria vary in each level.

Table 6-1 Process criteria

Level	Flow complexity	Integrations	Process data and coach complexity	Who develops it
Complex	High	Many	High	IT
Moderately complex	Medium (approximately 50 - 100 activities)	Some predefined	Medium	Business or IT
Simple	Low	None	Low	Business

6.3.1 Establish a Center of Excellence

The BES BPM program always had a strong sponsorship at the highest level within the enterprise. This established program proved to be an important factor in the success of the initiatives. However, in 2008, BES decided to give more visibility on this sponsorship, to strengthen the involvement of all parties and to provide specific attention to governance. This decision resulted in the creation of a BPM Center of Excellence (CoE) that proved to be critical to the success of the initiative.

The CoE followed these basic principles:

- Reports directly to the Bank Board and include the participation of one of its members.
- Includes the participation of IT and the Business areas in the bank that are responsible for the processes, from discovery to implementation and optimization (internally called an *Organizational Department*).
- Ensures that IT manages all initiatives related to BPM, from pipeline management to compliance, with a set of rules that ensures uniformity between the implementations.
- Defines the frameworks that support the initiative and its objectives, rules, and constraints.
- Aligns the implementation of each process on existing frameworks.

In practice the CoE has, among other things, to ensure compliance and alignment with the defined strategy and vision.

6.3.2 Define the frameworks

Following this vision and objectives, three levels of frameworks were defined, one for each of the three complexity levels previously described and according to the following rules:

- Level 3: Complex (internally called *Integrated*)
  - Development on this framework is for complex processes with a high level of integration, large volumes of data and complex user-interfaces.
  - Although the business is heavily involved, the process design and implementation is provided by IT.
  - Development cycles at this level are typically in the range of 3 - 12 months.
- Level 2: Standard (internally called *Procedural*)
  - Development on this framework is for processes of medium complexity, with few integrations and less complex user interfaces (UIs).
  - The process modeling, including UI design, is done by the Business with minimum IT participation. IT provides a full range of services to be available as a palette so that business modeling can be kept at a high level.
  - System integration, similar to other more complex and reusable services, is developed by IT.
  - An agile and iterative development approach is used with shorter development cycles varying in the range of 1 - 3 months for each iteration.
  - Processes are validated by IT at the end of their development to assure an accurate implementation. IT is also responsible for process deployment.
- Level 1: Simple (internally called *Generics*)
  - Development on this framework is for the simpler processes. These processes follow a standard model that minimizes implementation effort for both the process flow and the UIs.
  - Process development is in the hands of the business and is completely autonomous (from IT) from definition to deployment. Thus, IT manages only the capacity that is required to support execution in production.
  - These processes must be devoid of integration with other applications except for special cases where those integrations are proved to be highly reusable.
  - The cycle from process design to deployment should be just a few days.

By default, and regardless of process complexity, processes must be implemented in the Level 1 framework. This method might seem a contradiction, given that there two other frameworks, but this approach has to do with the process development lifecycle as follows:

- A complicated process can be divided into several parts, which individually can be considered simple tasks or subprocesses. If that division was not possible, agile and iterative deployments would not have the success that they have.
- Instead of immediately starting with a more complete and complex process implementation, start with a simpler process to better understand that process and to gather effective metrics based on actual behavior. Then, you can decide whether implementing the process in other frameworks is worthwhile.
- Exceptions can be decided by the CoE. The BPM CoE can make this assessment and decide on the future implementation (which might be to not develop the process). If the CoE decides to implement the process on one of the other frameworks, it will typically (but not necessarily) be in the Level 2 framework.
- A process implemented in the Level 2 framework might later be refined and evolved into the Level 3 framework.

When analyzing cataloged processes that are awaiting implementation, it became clear that the “long tail” diagram (Figure 4-8 on page 47) illustrated BES well. In fact, the number of simple cases exceeded all expectations at the bank, representing about 80% of the total. A further look at these simple processes showed that most of them were ad hoc. Generally, the organization knew that these ad hoc processes were happening, but in most cases defining the ad hoc processes as a structured flow provided no recognized added value. In most of these cases, the intention was to keep the knowledge workers who were interacting with the processes deciding what should happen next, such as the following examples:

- Send the instance to another user or department (meaning the users working on that department).
- Forward (that is, return) the instance to the previous participant.
- Forward the instance to the process instance creator.
- Terminate the process.

So, these processes fit perfectly into a dynamic workflow. Following the previous description of the Level 1 framework, the following distinct types of processes were available:

- Simple processes that are naturally ad hoc and will never evolve to a different implementation
- More complex processes that were decomposed, simplified, and are opportunistically implemented on this framework but that have the potential to be implemented in the other frameworks in the future

6.3.3 Development and management of the frameworks

Now the frameworks can be categorized into three levels.

Level 3: Complex

In 2007, with the early development on Teamworks, the bank set up a process development team that was focused on what the bank called *Base Services*. These services were considered to be reusable across the different processes. In practice, Base Services were the development assets that were not specific to any “process domain” (later called *Process Application*). On later Business Process Manager versions, Base Services were called *Toolkits*.

This framework includes the following service and features:

- Integration with an Enterprise Content Management (ECM) platform
- Integration with an Authorization Management Application
- Integration with the Enterprise Application Integration (EAI) platform
- Logging, tracing, and auditing services
- Error handling features

In addition to these Base Services, a strong set of governance rules were defined to be applied by all the teams members regarding the use of the services in the development cycle. Since 2008, this framework has undergone few changes, because its profile is already in line with future objectives.

Everything within this framework was provided and controlled by the IT team, and the use of the framework is exclusive for IT. For the processes developed on top of this framework, the Business Departments participate in requirements specification, testing, and acceptance but never interact directly with the Business Process Manager development tools.

Level 2: Standard

The idea behind this framework was to create a set of high-level services that expose a set of features to the business users so that they could design and implement the processes with the lowest possible technical knowledge. In practice, some Base Services developed within the Level 3 framework were encapsulated to hide their complexity. So, the framework offers a simpler usage of those Base Services. In addition, the following services were developed to enable basic BPM technical aspects to be easier to implement:

- Automated data INSERTS on process specific databases
- Input field validation and formatting

The business teams used this framework to model the processes, design the UIs, and define the data entities so that IT provided the following support:

- Development, maintenance, and tuning of the framework itself
- Technical support of business users who implement the processes, especially for new business users who need training on the tools and methodologies
- Final certification of the developed processes to ensure acceptable production deployment

The framework was made available in 2008, and business teams immediately started the first process development led by the business users.

Level 1: Generics

This framework defined the following requirements:

- Targeted for simpler processes. These processes must follow a standard model that minimized the implementation effort, either from the point of view of the process flow or the UIs.
- The business must be completely autonomous from its definition to deployment, meaning that IT manages only the capacity that is required to support the execution in production.
- These processes must be devoid of integration with other applications except for special cases where those integrations are proved to be highly reusable.
- The cycle from process design to its deployment should be of just a few days.
- Based on these requirements and analyzing viable options to achieve them, the strategy chosen for the implementation of this framework was based on the creation of a meta process (called the *Generic Engine*). All the processes implemented in this level would execute on that Generic Engine.

Basically this framework is composed of the following major components:

- Generic Process Engine (GPE)
- GPE Supporting Database
- GPE Configuration Utility

The section that follows describes the goals, usage, and the way that these components work together to provide a single framework that supports hundreds of different processes with thousands of new process instances created per day. Consider the following key aspects:

- The processes that run on top of this framework can have an implementation time of just one day. This time frame includes the process configuration on the framework, validation by IT, and deployment in production.
- All the configuration is done by business users, without any IT participation.
- IT validates only the estimated amount of new daily instances for new processes being deployed, assuring that they can be supported by the production infrastructure.

6.4 How Generics environment got its start

Figure 6-5 illustrates, in a simplified way, the main components of the Generics environment and the way these components relate to each other.

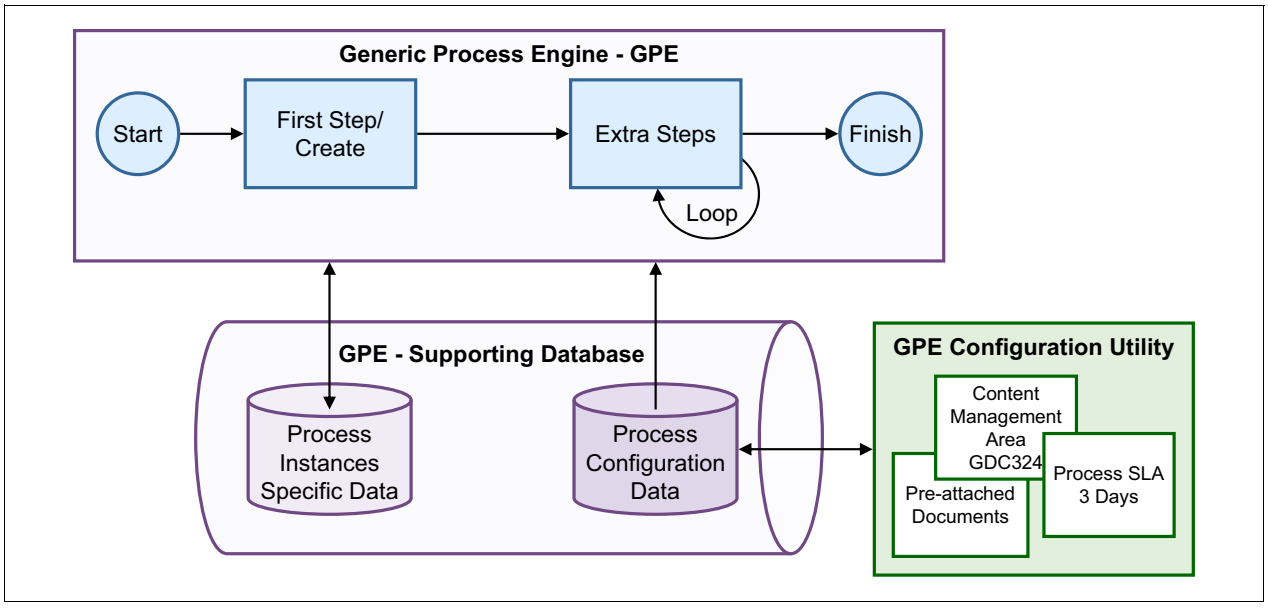


Figure 6-5 Main components of the Generics environment

6.4.1 Generic Process Engine

The Generic Process Engine (GPE) is basically an IBM Business Process Manager process application that consists of a main business process definition (BPD) that is designed to provide the simple flow illustrated in Figure 6-5. This BPD provides this flow by obtaining the specific behaviors and details for each process according to the parameter settings stored as configuration data of the supporting database. At a later point, this data is enriched to offer a wide range of behaviors based on the attributes of each process that is stored in that database.

When a user creates a new instance of a given process that is configured on this platform, the GPE reads the configuration data for the process from the supporting database. The instance behavior is then controlled by the GPE based on that data, from the start of the instance (for example, validating whether the user who create it is authorized to do so), its flow (for example, the SLA that is required for the process) until it ends (for example, who can finish it).

In Figure 6-5 on page 68, the meta process seems relatively simple. However supporting the various features that are required of the environment led to a substantially more complex implementation for the Process Application.

Figure 6-6 illustrates the first level of the BPD that is built to support all of these features.

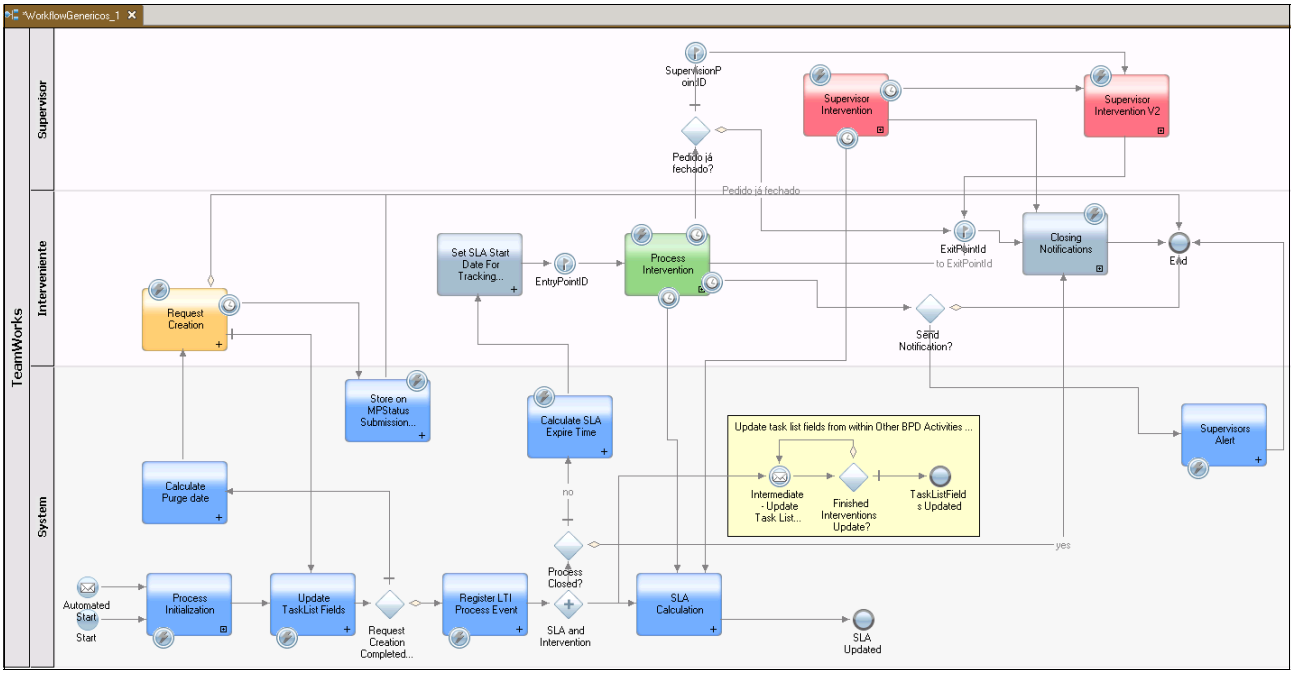


Figure 6-6 First level of the process diagram

The loop shown on Figure 6-5 on page 68 (to support the multiple steps in a process instance) is not visible in Figure 6-6 because it is implemented as a lower level BPD that implements the Process Intervention activity (the green box). At the end of each step of this loop, the actors decide what the next step will be.

This engine currently supports hundreds of processes with thousands of instances created daily.

Initially, the engine supported a set of features for its first version and another set of less critical features were to be developed after. The first version of Generics was deployed in late 2008 and those improvements came immediately afterwards. Given the success of Generics within the organization, an even richer major version (called V2) was planned and developed in 2012, which was deployed at the end of the year.

Generics V2 included the following major enhancements:

- In addition to the first and extra steps loop, Generics includes a simple graph that is designed and configured for each process. That graph describes the states that can be followed by a particular instance. For each instance, the actors can choose the next path from the available options in the graph. Some of the things that today are configured for the process will be defined at a finer grade for the state level.
- Using the GPE Configuration Utility, you can define data structures that can then be used inside a process configuration. With this, Generics can have structured data available to activities, text comments, and attached documents (that are available in V1). Visibility control and editing definitions are also made through the configuration utility that are then applied by the engine.

Version note: The figures and information in the sections that follow are based on Generics V1.

The development team established a clear, well-defined boundary for how the environment can grow and for the features and usage the environment should or should not have. There are no major developments planned in the future. As explained previously, the environment was built using existing BPM assets and toolkits, and this build environment was a key factor in minimizing the development effort and maintenance between the GPE and the other processes.

The following figures show some of the UIs that are available on the Generics environment. Each generic process has three primary tabs (on the left side):

- Case
- Attachments
- Monitor

The Case example in Figure 6-7 shows an instance of a process that is used to ask for a credit card limit change approval. This example is a good example of a business process that, not having any other implementation, was quickly deployed in Generics, thus providing a more formal structure than email or fax.

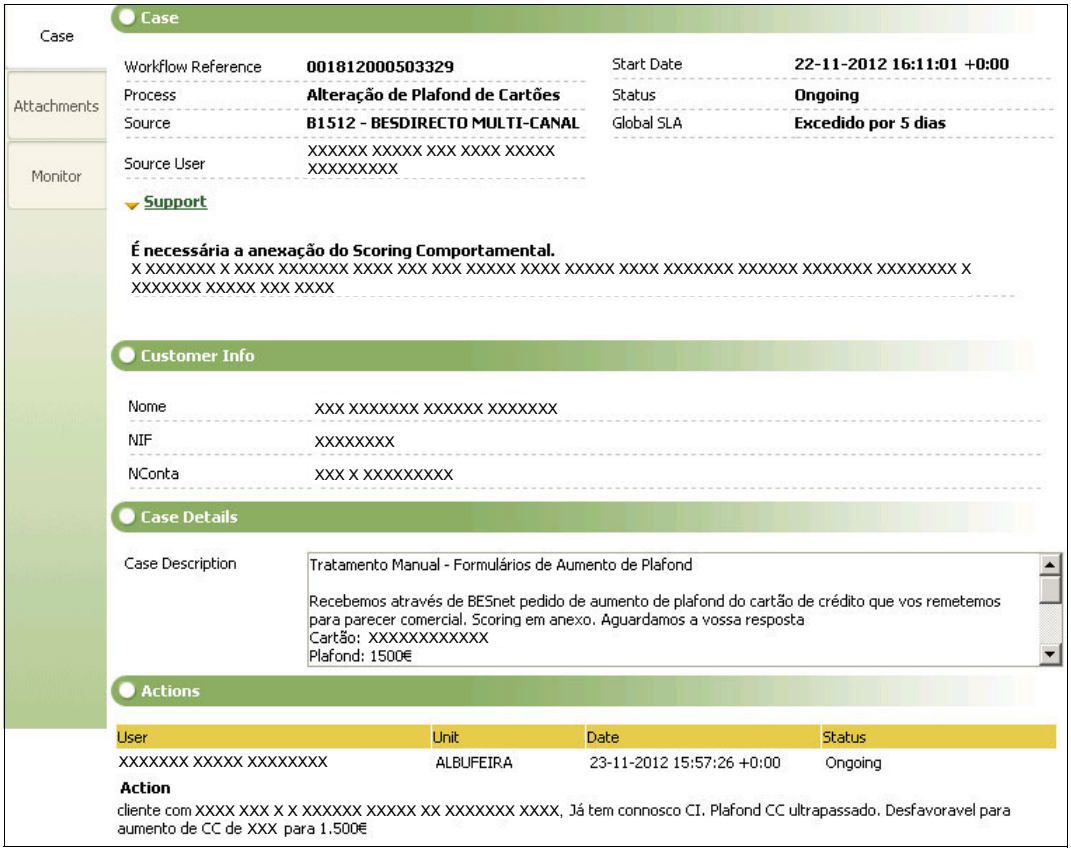


Figure 6-7 Case for credit card limit change approval

The top portion of the panel provides generic data about the process, its creator, and the Global SLA (which is expired in 5 days). It also includes guidance (Support) stating that a particular scoring document is attached by the requester (the text is in Portuguese). After that, a section includes basic customer data that results from an integration reused by the Generics framework. (Because this is a real-world example, some of the data is obscured.) Next, the Case Details shows text entered by the creator followed by a list of actions (in this case, only one element) with the track of the actors and the comments that are recorded by each.

Figure 6-8 shows the Attachments tab of the UI with the list of documents already attached to this instance.

Figure 6-8 Attachments tab

Figure 6-8 shows how the ECM integration that was built with the first complex processes is easily reused within a Generics process implementation. All of the documents are stored on the ECM system and exposed by the processes. The UI could be populated directly from the processes (that is, they are predefined) or can be attached as a part of the process. For example, a document is edited, saved as a part of a process step, and then saved back to the ECM system.

Finally, Figure 6-9 shows the Monitor tab.

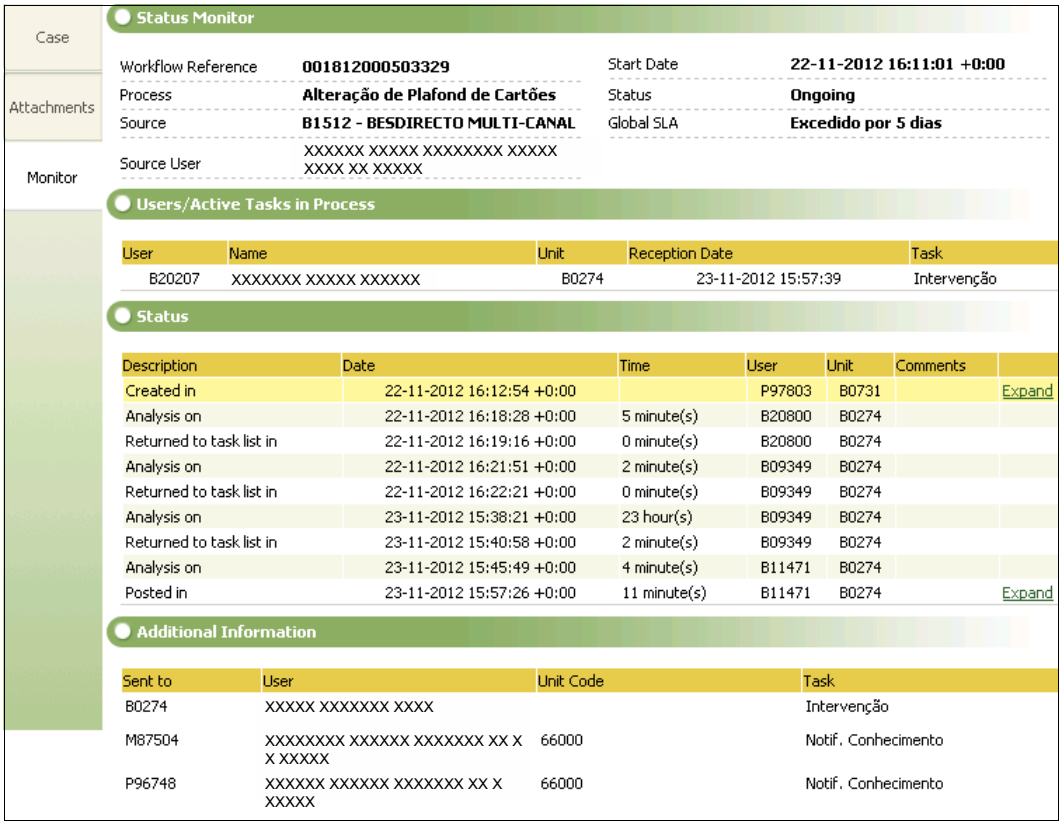


Figure 6-9 Monitor tab

This UI is used to expose the active tasks for the instance and also all the previous relevant functional steps (a trace of previous activities similar to the Social stream is available in IBM Business Process Manager). This tab shows the actors and the time spent on each step.

With these UIs the process data is visible during its execution and also after it (maintained for historical analysis). It is also easy to check for the performance of each actor and how and where the SLAs are broken.

So the engine is the primary component, making the instances behave as they should based on the configuration data, which is described in the next section, and exposing all the UIs for the running and completed instances.

6.4.2 GPE Supporting Database

The GPE Supporting Database has the following main components:

- Process configuration data
- Process instance specific data

Process configuration data

As described previously, the GPE implements the behavior for each process instance based on the configuration data for that particular process. Process configuration data is where that data is stored. Several examples of the attributes, and behaviors implied from them for each instance of each specific process, are as follows:

- Who can create new instances (for example, anyone, only specific departments or users)
- Confidentiality that states who can query the process data (for example, anyone, the creator and actors, and so forth)
- Global process SLA
- Activity SLA
- Supervisor that states who is notified or acts if the SLA expires
- What to do when the SLA expires (for example, escalate the process to the supervisor or only notify the supervisor)
- ECM area where the process attached documents shall be stored
- Support text that designates a message that gives context to the creators and actors about what is expected to accomplish during its execution
- Who can finish live instances (for example, any actor or only the creator)
- Who is notified when a process finishes (for example, no one, the creator, the supervisor, and so forth)

Figure 6-10 illustrates a simplified table to store the process configuration data.

ProcessID	Process Name	Description	Process SLA	Supervisor	ECM Area
xxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxx	xxxxx	xxxxxxx
xxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxx	xxxxx	xxxxxxx
xxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxx	xxxxx	xxxxxxx
3333	Credit Card Statement copy	Used to ask copies of monthly statements for credit cards. Shall be assined to the Credit Cards department and after validation a copy of the statement shall be retrieved from the ECM system, attached and sent to the requester.	1 day	J10175	GDC3311
xxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxx	xxxxx	xxxxxxx
xxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxx	xxxxx	xxxxxxx
xxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxxxxx	xxxx	xxxxx	xxxxxxx

Figure 6-10 Process configuration data

For example, the row for the process that has ID 3333 lists the values for each attribute that describe what is expected for each instance of this process:

- SLA of 1 day
- Supervisor is user J10175
- ECM area to store the instances attachments is GDC3311

Process instance specific data

One of the implementation decisions made by the development team early on in 2007, when working on the initial Teamworks development projects, was that each process application should have its own database. That decision was being followed on the implementations with previous BPM platforms, and it was recognized as a critical success factor for the following reasons:

- Business discipline and the organizational mindset is an important facet of BPM, but ultimately software is running. Performance is always a factor, and being able to optimize and tune data access is critical for the overall process, for general platform performance and especially for the user experience. Some of the BES processes have huge amounts of data associated with each instance. So, structuring that data and the way it should be stored and accessed is a key consideration.
- After each process instance is completed, the process data and everything related with it must be available for years (mostly for legal or audit reasons). However, maintaining millions of completed process instances in the BPM database only to be able to query that data is neither efficient nor recommended. Having that data outside the BPM product and being able to understand and report the process flow even years after it has been completed is important, particularly in a regulated industry such as banking. This method also enables the purging of completed instances from the BPM database minimizing issues because of a bloated database.

For Generics, there is a single process application and, therefore, a single database where the process-instance-specific data from the various supported generic processes are saved. Figure 6-11 illustrates a simplified master table to store the process instances data. It shows two instances of the process with ID 3333.

ProcessInstanceID	ProcessID	Creator	CreationDate	ECM Doc	EndDate	LastActor	FailedSLA
xxxxxxxxx	xxxx	xxxxxx	xx-xxx-xxxx	xxxxxxxxx	xx-xxx-xxxx	xxxxx	x
xxxxxxxxx	xxxx	xxxxxx	xx-xxx-xxxx	xxxxxxxxx	xx-xxx-xxxx	xxxxx	x
393827461	3333	J10090	11-Nov-2012	AXD123123			T
xxxxxxxxx	xxxx	xxxxxx	xx-xxx-xxxx	xxxxxxxxx	xx-xxx-xxxx	xxxxx	x
xxxxxxxxx	xxxx	xxxxxx	xx-xxx-xxxx	xxxxxxxxx	xx-xxx-xxxx	xxxxx	x
xxxxxxxxx	xxxx	xxxxxx	xx-xxx-xxxx	xxxxxxxxx	xx-xxx-xxxx	xxxxx	x
531232512	3333	B01271	26-Jul-2011		26-Jul-2011	J12321	F
xxxxxxxxx	xxxx	xxxxxx	xx-xxx-xxxx	xxxxxxxxx	xx-xxx-xxxx	xxxxx	x
xxxxxxxxx	xxxx	xxxxxx	xx-xxx-xxxx	xxxxxxxxx	xx-xxx-xxxx	xxxxx	x
xxxxxxxxx	xxxx	xxxxxx	xx-xxx-xxxx	xxxxxxxxx	xx-xxx-xxxx	xxxxx	x

Figure 6-11 Simplified master table

So, during the execution of these two instances, the GPE, reading the configuration data for that Process (3333), provides a dynamic workflow (according to the database configuration) and collects data with details of each task for each process instance.

Having this data, allows you to perform the following tasks:

- Check the volumes (created process instances, tasks executed, and so forth) for the organization as a whole or for specific branches, groups of branches (commercial directions), and departments.
- Check what is the “normal” flow for these instances inside the organization, and use that when modeling the process in a different framework.
- Evaluate how the SLAs are being accomplished, and do it also at different levels within the organization (department, user, and so forth).

6.4.3 GPE Configuration Utility

An engine reads configuration data from a database that is designed to hold the relevant data for the behavior of each process. So, you need a business-specific utility that enables the business to store that configuration data on that database. This utility is implemented by the GPE Configuration Utility. Its objective is to enable process configuration that acts somewhat like a “back-office” for process configuration and then populates the Process Configuration Data accordingly. This utility is designed for business users and exposes the different Generics functions in a simple and understandable way.

The figures in this section show that UI and groups of functions exposed on the GPE Configuration Utility.

Figure 6-12 shows the process catalog inside which the processes are configured.

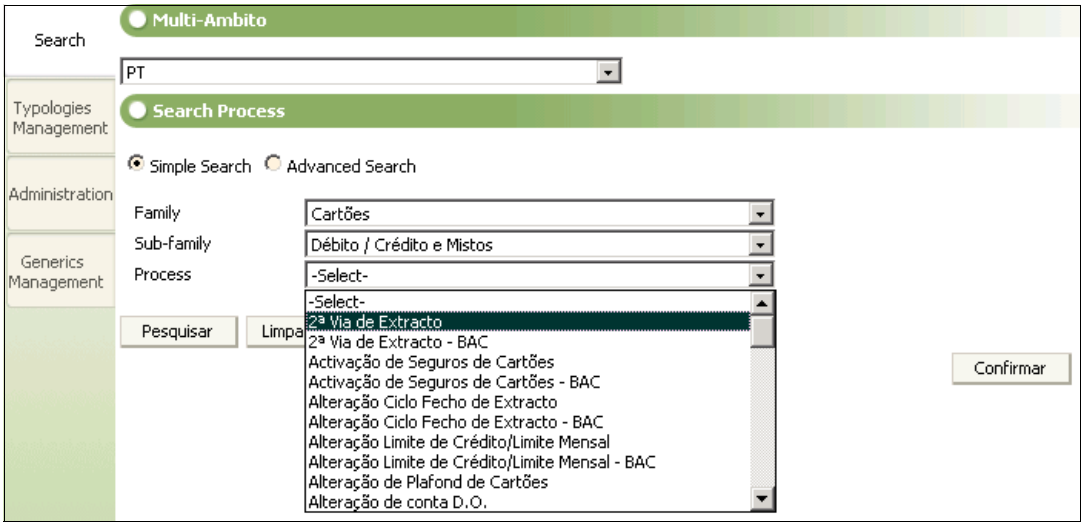


Figure 6-12 Process catalog

The Generics platform supports multi-language and multi-time zone capabilities. BES operations already uses the Portuguese and Spanish capabilities on a shared (single) “instance.” This function can be expanded to any other operation on a different country by configuring the language resources accordingly. Then, you can determine whether to run the Generics platform on the same instance or a new instance created in a different location.

The processes are classified under *families* and *subfamilies*, following the company’s process tree as generally known by the business. The contents shown are in Portuguese, but it translates as follows:

- Family: Cards (Cartões)
- Subfamily: Debit/Credit and Mixed (Débito/Crédito e Mistos).
- Process: The selected process is used to ask for a copy of the monthly statement for the card (2ª Via de Extracto).

The following figures use an already configured process. Creating a new process starts by selecting where to create the process inside this catalog (the family and subfamily).

Selecting an existing process gets the current configuration data of the process, as shown in Figure 6-13.

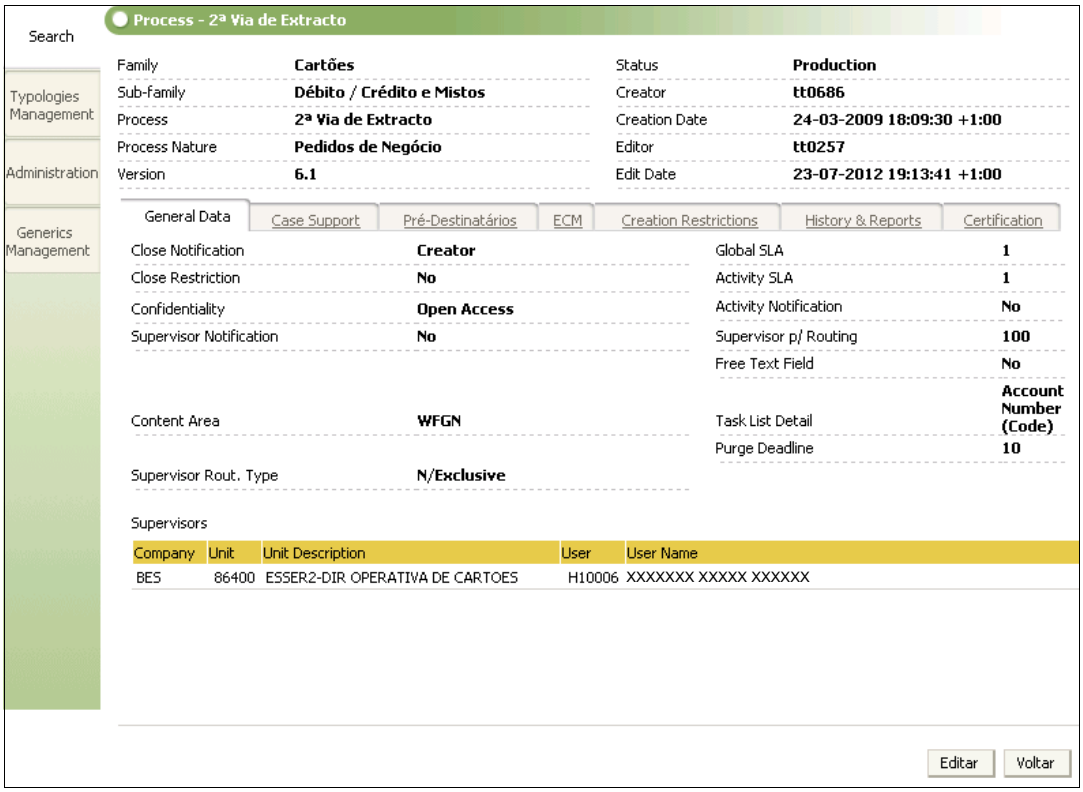


Figure 6-13 Selecting a process

The process configuration displays in several blocks of data, shown as horizontal tabs. Figure 6-14 shows the General Data tab. In this UI, you can change the process SLA, its supervisor, confidentiality, and other configuration values.

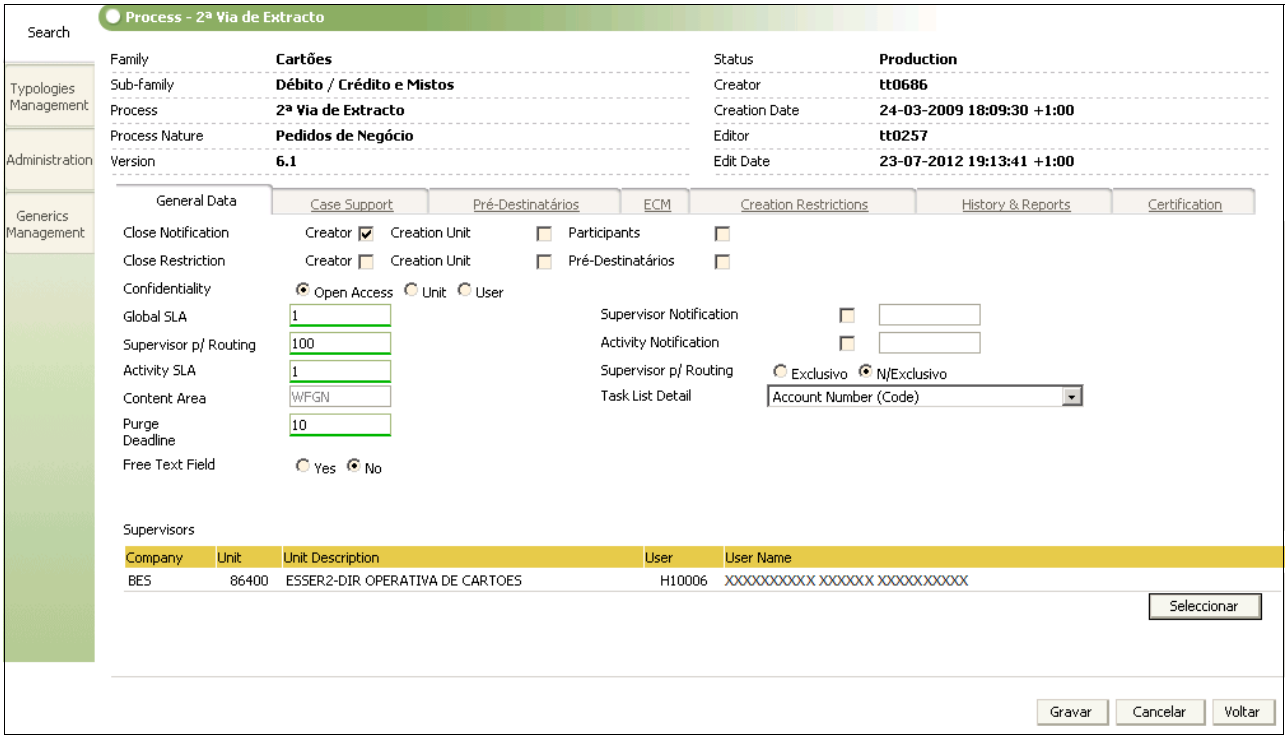


Figure 6-14 The General Data tab

Figure 6-15 shows supporting text that is used as guidance for the creators and actors. In addition, a set of pre-built sentences work as shortcuts for anyone who wants to include them in the text box when executing tasks.

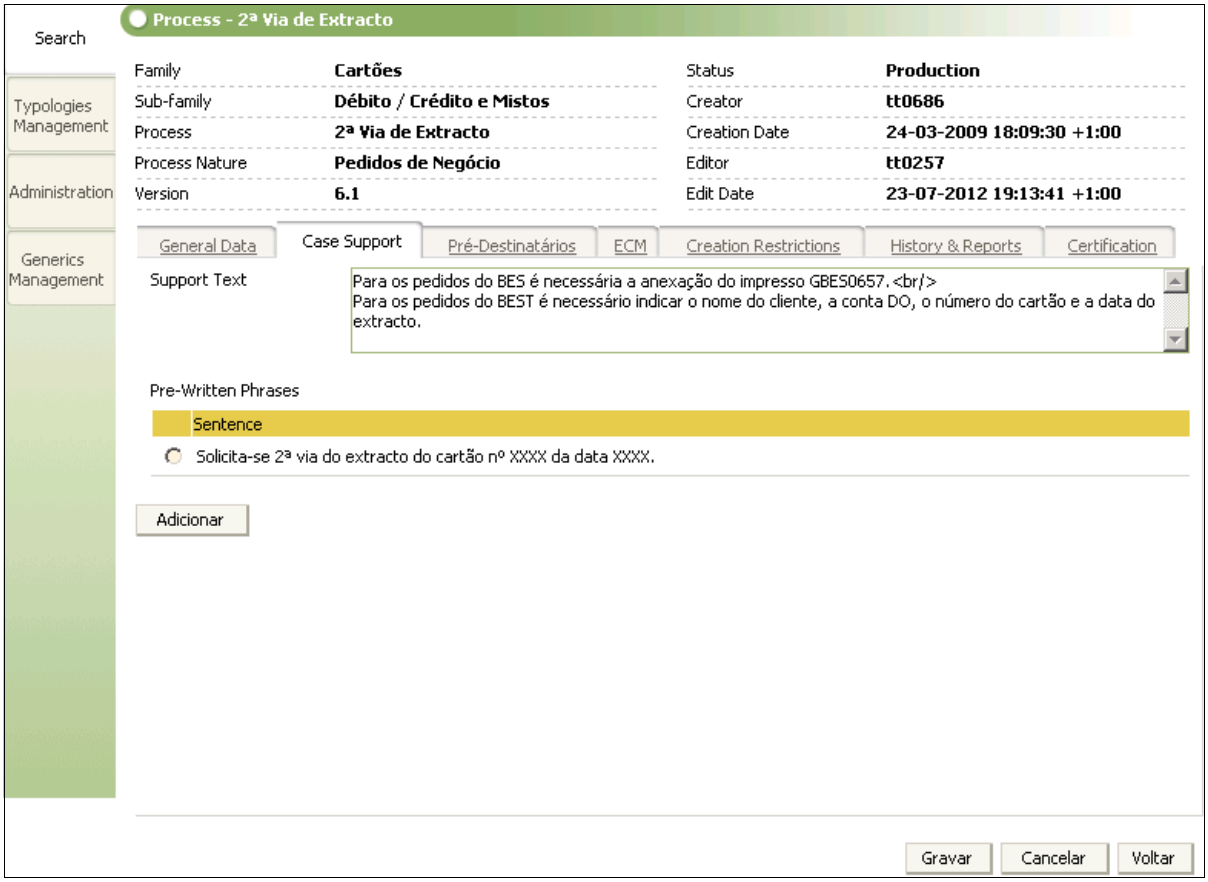


Figure 6-15 Editing process support data

Figure 6-16 shows the predefined set of actors that will receive tasks after the creator submits it. If not defined, the creator must explicitly indicate the actors on creation. This creation works only for that first step of the “loop,” meaning that after the first step, any user needs to explicitly indicate the actors for follow-on steps.

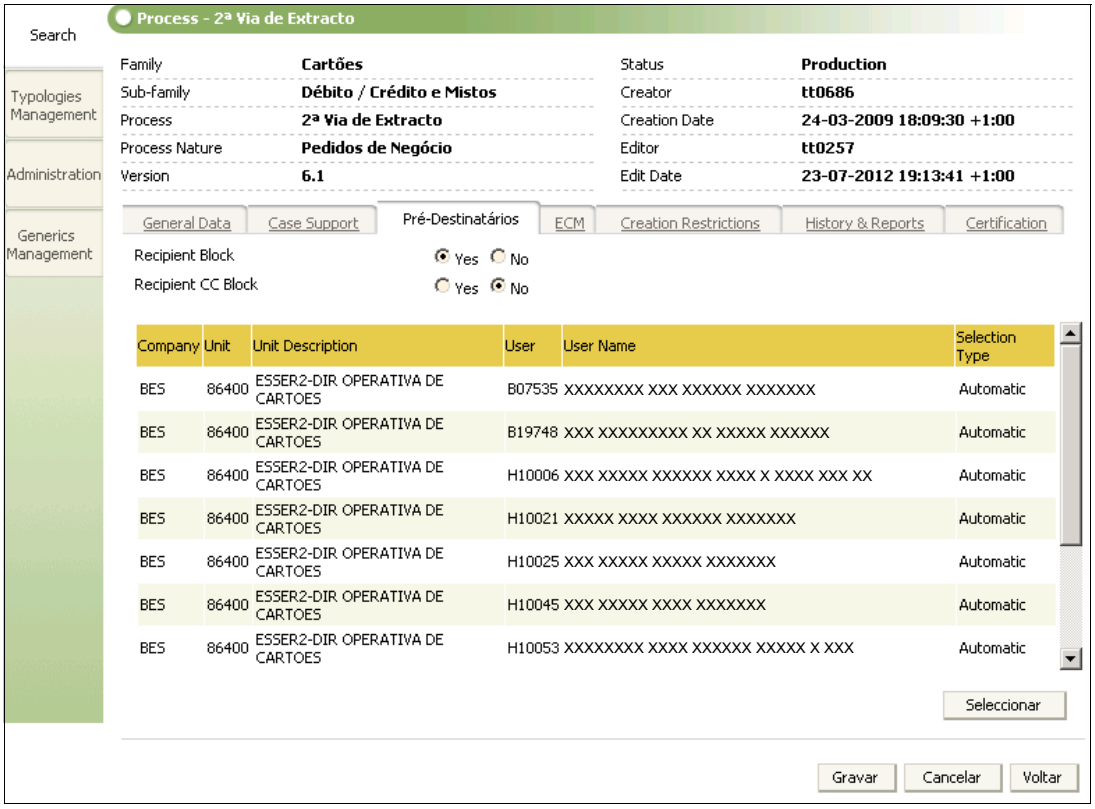


Figure 6-16 Predefined actors

Figure 6-17 shows the ECM configuration block. In addition to the ECM area where the attached documents are stored, attachments can be identified that are attached automatically to be each instance.

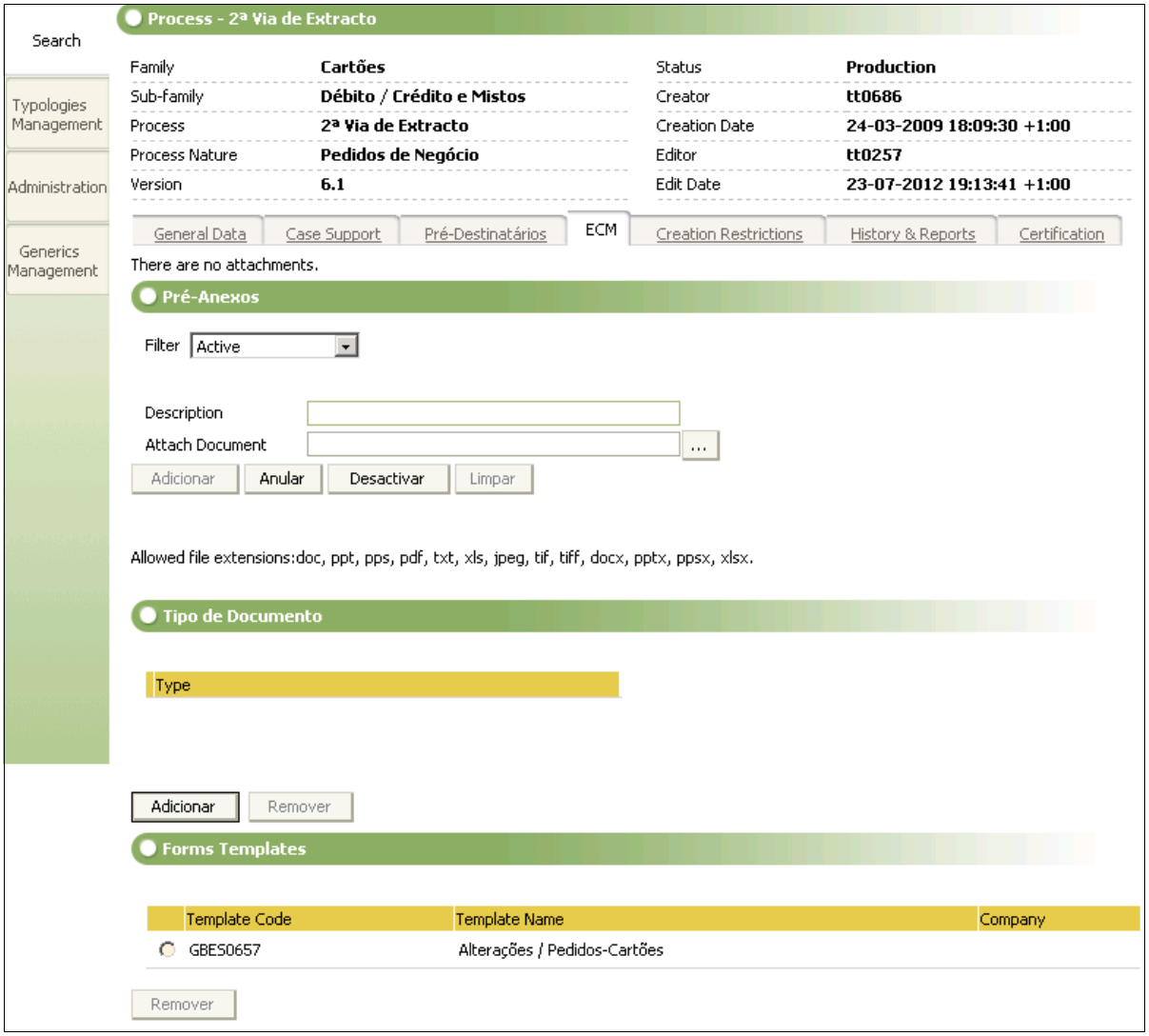


Figure 6-17 ECM configuration block

Finally, the authorized creators are configured (Figure 6-18). This step can be skipped if any user can create instances for the process. The figure shows the list of units (branches and departments) where new instances can be created.

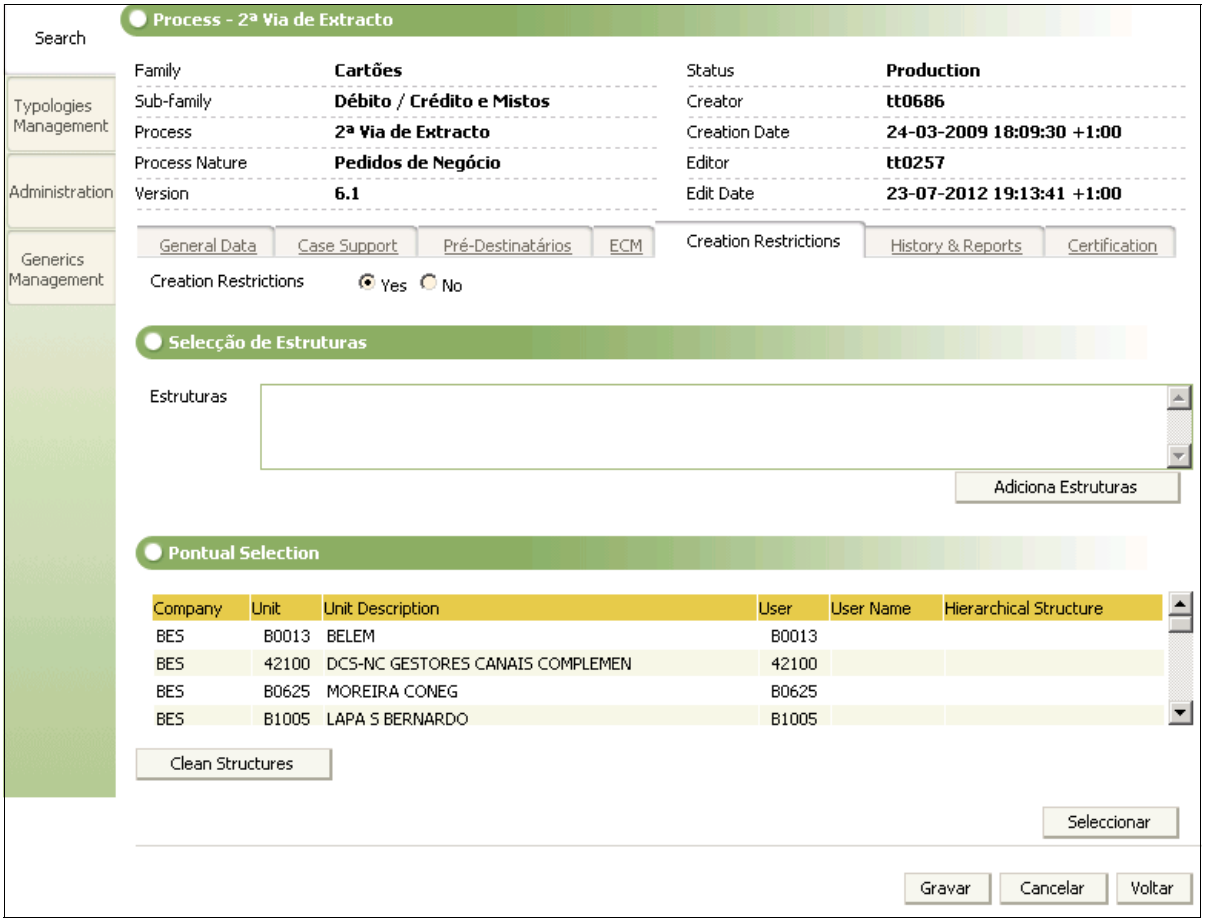


Figure 6-18 Configuring authorized creators

The BES decision was that only the Organizational Department uses this utility. So, there is a further template that is well known by all the other departments, and that template is used when they need any new Generics process to be defined. That template maps the same functions that are supported and exposed in the Configuration Utility so that collecting that data is a simple, rapid exercise. This function turns the Organizational Department into the owner of the Generics configuration, assuring that all the process configuration data is validated and that the deployment occurs as wanted and needed.

6.5 Generics rapidly increased

Figure 6-19 shows the number of Generics processes that have been deployed in production per year since 2008.

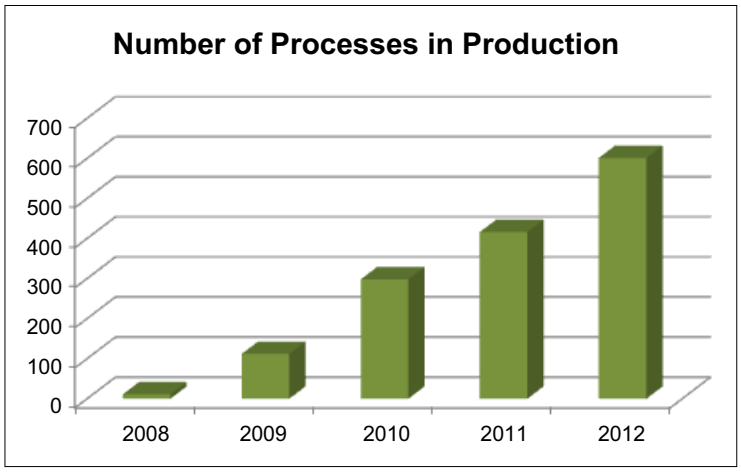


Figure 6-19 Generics processes in production per year

These numbers were collected in November 2012, before the deployment of Generics V2, with which a huge number of new processes were expected to be deployed. It was estimated that there would be approximately 1,000 processes deployed in early 2013.

As discussed previously, the platform supports the Portuguese and Spanish BES operations and is ready to support any others. The Spanish operation started using it in 2010, after it was updated to support multi-context, multi-language, and multi-time zone operations.

Figure 6-20 shows the number of Generics process instances created per year.

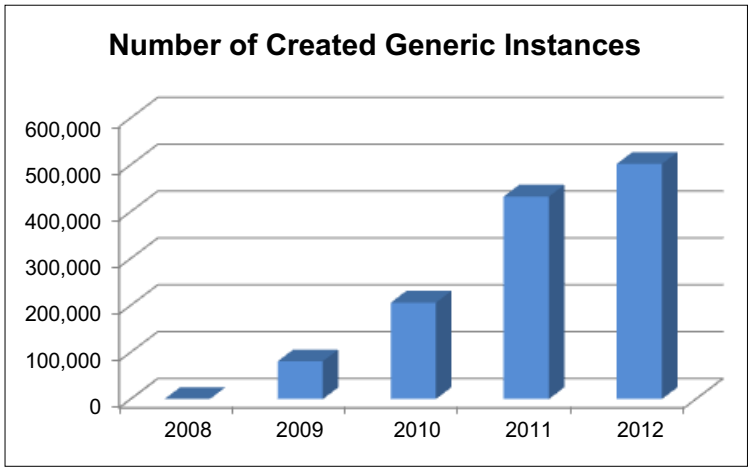


Figure 6-20 Generic process instances per year

For 2012, BES was expected to have around 600,000 Generics process instances created on IBM Business Process Manager. (The value shown for 2012 represents the number at the end of November.) That number represents about 50% of the total number of process instances created for all BES processes, shown in Figure 6-21.

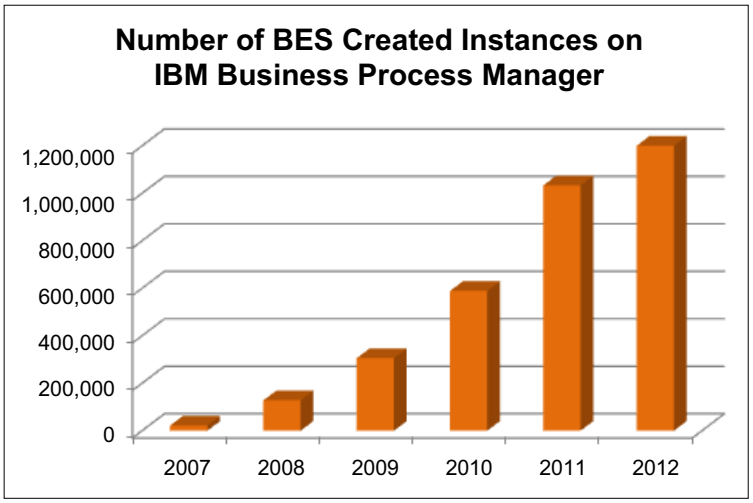


Figure 6-21 BES created instances on IBM Business Process Manager

Add to this number the process instances created on deprecated platforms that will be migrated to IBM Business Process Manager V8 during 2013. The numbers show that from 2009 to 2010 the number of Generics instances doubled, and from 2010 to 2011 they doubled again. Expectations for 2013 is that the number will increase again, because with V2, a considerable number of processes will be added to run on top of the Generics platform.

6.6 The benefits of Generics to BES

BES has recognized the following benefits from adopting this Generics approach:

- Visibility, measurement, and control

  By definition, if a process is running in the organization, some part of the business needs that process and sees value in it. Further, if no better method exists to support a process, the process is likely being implemented as emails, faxes, or even phone calls between the actors. And none of these “off the grid” activities are easily visible or trackable.

  The immediate advantage that BES got from Generics is visibility. Making the processes visible makes them measurable, and if you measure a process, you can control it.

- Better performance and SLAs

  When people know that their performance in a given set of tasks is (or can be) measured, they are more interested and are motivated to perform these tasks. An added bonus is that this motivation is mostly a natural consequence, and little additional effort is required beyond gaining the initial visibility.

  In addition, having measurement and control means that we can know what is outside the boundary of the SLA or at risk of being so. This information can help the organization to achieve better performance and incrementally improve efficiency to meet SLAs.

- Fast and easy process support

  The simplicity and speed with which processes can be deployed without any IT participation (in its configuration) means that the Generics solution is an excellent tool

for the business. It has now become business (users from any departments asking for new processes) talking with business (the Organizational Department). In a matter of hours, or at most days, a process can be configured and deployed.

Basically, this set of benefits works together with visibility. Having more supported processes, because doing so is simple and fast, provides more visibility and control to the business.

- Risk mitigation

  Having some level of process support with more visibility, control, and better performance provides an improved ability to manage the business, which can significantly reduce risk.

- Process design lifecycle based on numbers

  Because BES is measuring volumes for a larger set of process types, decisions about which processes to move to other frameworks is based on real numbers. This information makes a significant difference in the way BES has managed the process development lifecycle and the priorities around that lifecycle. This method makes it much more straightforward to determine which processes are really impacting the business and how.

6.7 Examples importance and usefulness of generic processes

This section provides two examples that illustrate the importance of generic processes.

6.7.1 Leasing company merger

In 2010, BES merged with a leasing company, and there was an immediate requirement to support that company's processes. A diverse set of processes were associated with the leasing lines of business, such as leasing for auto, industry equipment, office equipment, real estate, and so forth, and each of these lines of business had its own unique set of characteristics. There was also a set of basic processes supporting the general leasing business, such as simulations, contracts, insurance policies, contractual guaranties, and so on. In all, there were more than 30 processes with different levels of complexity and criticality. Many of these processes were immature. The merger was likely to affect the majority of the processes.

Considering all these circumstances, BES decided to implement all of these processes as Generics. Doing so provided an immediate, initial level of support and enabled data to be collected for each process. This data provided a better understanding and visibility on each process while the processes became more stable and mature inside the organization.

After some time and with the existing data collected on Generics for each of those circumstances, processes were assessed and it was straightforward to determine which processes should remain as Generics and which processes required a deeper and more complete implementation using one of the other frameworks. A set of these processes were then implemented on top of the standard framework and after some time, only a few were implemented on the complex framework by the IT team.

6.7.2 Credit processes servicing

Credit Approval processing has been at the center of the BES BPM activities since the beginning. In addition to the core Credit Approval processes, many ancillary processes are key to the overall credit approval activity. Some of the Credit Approval activities result in contracts that are valid for many years (even decades). Many situations can occur in that time that require appropriate handling by these ancillary processes. However, the IT team did not give the implementation of these processes a high priority. After Generics were made available, these processes were implemented quickly, and the overall credit applications process benefited from the more formal availability of these ancillary processes.

6.8 Managing generic processes

Management of generic processes can be divided into business and IT team activities.

6.8.1 Business team activities

Since the first BES BPM implementations, one of the major objectives was to provide business teams with the tools to understand how their processes were behaving evolving and inside the organization. It was important for them to understand how each process was running daily and how to collect the historical data that aids in understanding process evolution so that they could offer ways to decide how to improve it. The implementation of Generics kept to this simple path.

Scoreboards

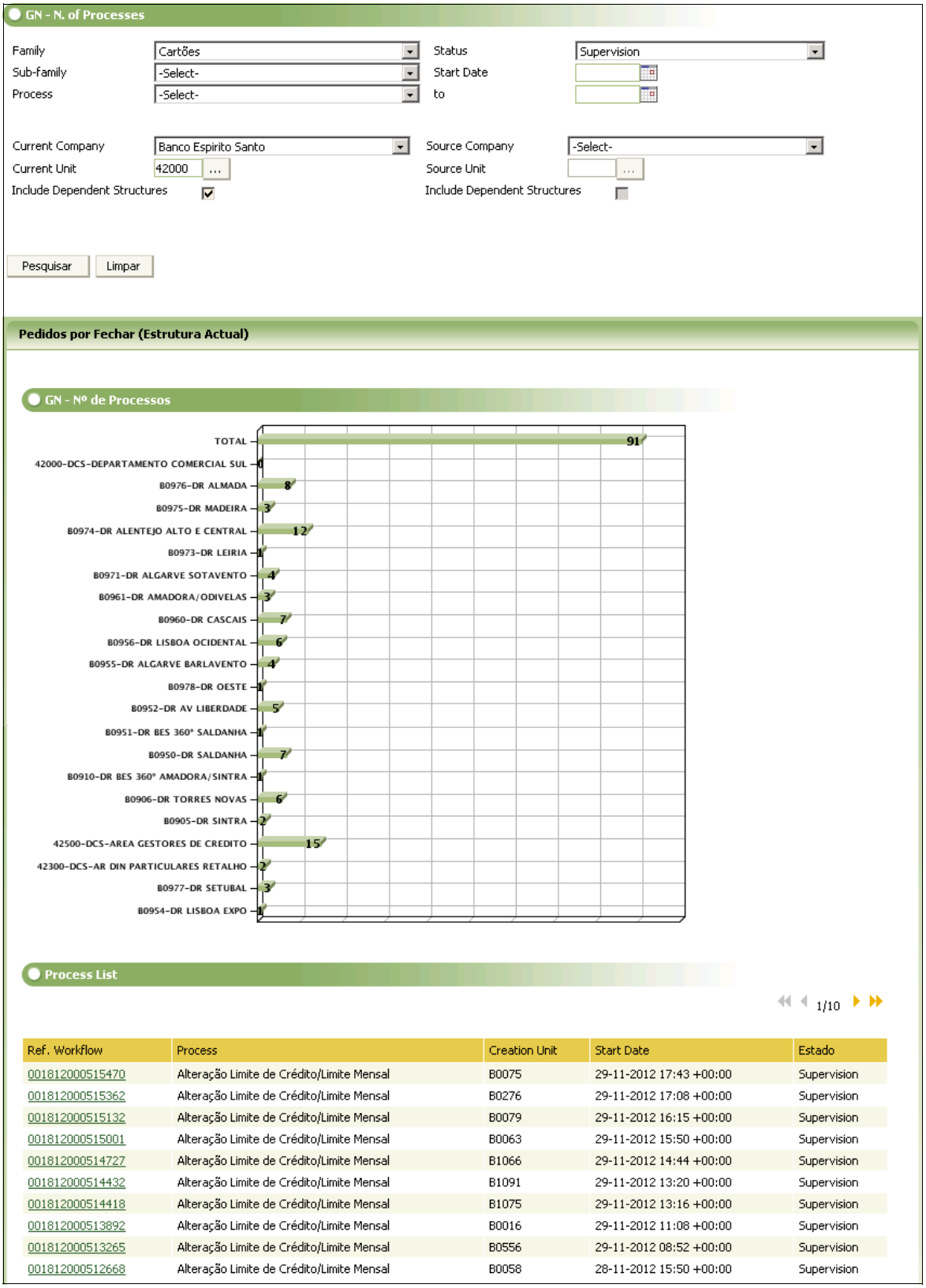
Scoreboards were built with the IBM Business Process Manager reports and scoreboards capabilities and included the following purposes:

- SLAs and key performance indicators (KPIs)
- Department, branch, and team workload
- General data and volumes

Some scoreboards can be used for more than one purpose, depending on the filters that are applied.

The figures in this section show examples of existing scoreboards and how they are used by the business.

Assume that a manager from the Cards department, who is responsible for the general control of all the processes around the cards line of business, wants to know the status of the SLAs for that department at a particular moment. The manager can use a scoreboard, such as the one shown in Figure 6-22 on page 86.



Ref. Workflow	Process	Creation Unit	Start Date	Estado
001812000515470	Alteração Limite de Crédito/Limite Mensal	B0075	29-11-2012 17:43 +00:00	Supervision
001812000515362	Alteração Limite de Crédito/Limite Mensal	B0276	29-11-2012 17:08 +00:00	Supervision
001812000515132	Alteração Limite de Crédito/Limite Mensal	B0079	29-11-2012 16:15 +00:00	Supervision
001812000515001	Alteração Limite de Crédito/Limite Mensal	B0063	29-11-2012 15:50 +00:00	Supervision
001812000514727	Alteração Limite de Crédito/Limite Mensal	B1066	29-11-2012 14:44 +00:00	Supervision
001812000514432	Alteração Limite de Crédito/Limite Mensal	B1091	29-11-2012 13:20 +00:00	Supervision
001812000514418	Alteração Limite de Crédito/Limite Mensal	B1075	29-11-2012 13:16 +00:00	Supervision
001812000513892	Alteração Limite de Crédito/Limite Mensal	B0016	29-11-2012 11:08 +00:00	Supervision
001812000513265	Alteração Limite de Crédito/Limite Mensal	B0556	29-11-2012 08:52 +00:00	Supervision
001812000512668	Alteração Limite de Crédito/Limite Mensal	B0058	28-11-2012 15:50 +00:00	Supervision

Figure 6-22 Scoreboard

The top of Figure 6-22 on page 86 shows a set of filters where the family/subfamily and process can be selected. For this particular example, only the family was selected. For the process status, the processes that are already out of SLA are selected. Thus Supervision is the selected status.

This scoreboard is built to show the total number of processes with the selected state on the first line. In this case, there are 91 processes in Supervision status. After this, the scoreboard shows a line for each unit (departments and branches within the selected hierarchy) showing the number of processes for each. This information provides an idea of the departments with a higher number of processes that are out of the SLA. Drill-down capability is available for each row to show the same information but only for the selected row. For example, if you select the **B0950 - DR SALDANHA** row, you can drill down into that area and see the relevant units showing how the seven processes that are from the SLA are distributed.

The bottom of Figure 6-22 on page 86 shows all processes that were considered in the numbers discussed previously. For this particular example, there is a (paginated) list of the 91 processes that are from the SLA. Selecting any of these processes by clicking the line (and with the correct permissions), you can query all the process data.

Figure 6-23 on page 88 illustrates one of the following situations:

- The manager drills down by selecting the row for the unit.
- The person responsible for that particular unit (B0950) was called to try to dispatch the work for the delayed processes. This person needs to understand to whom the unit is currently deployed and who has the unit in the task lists.

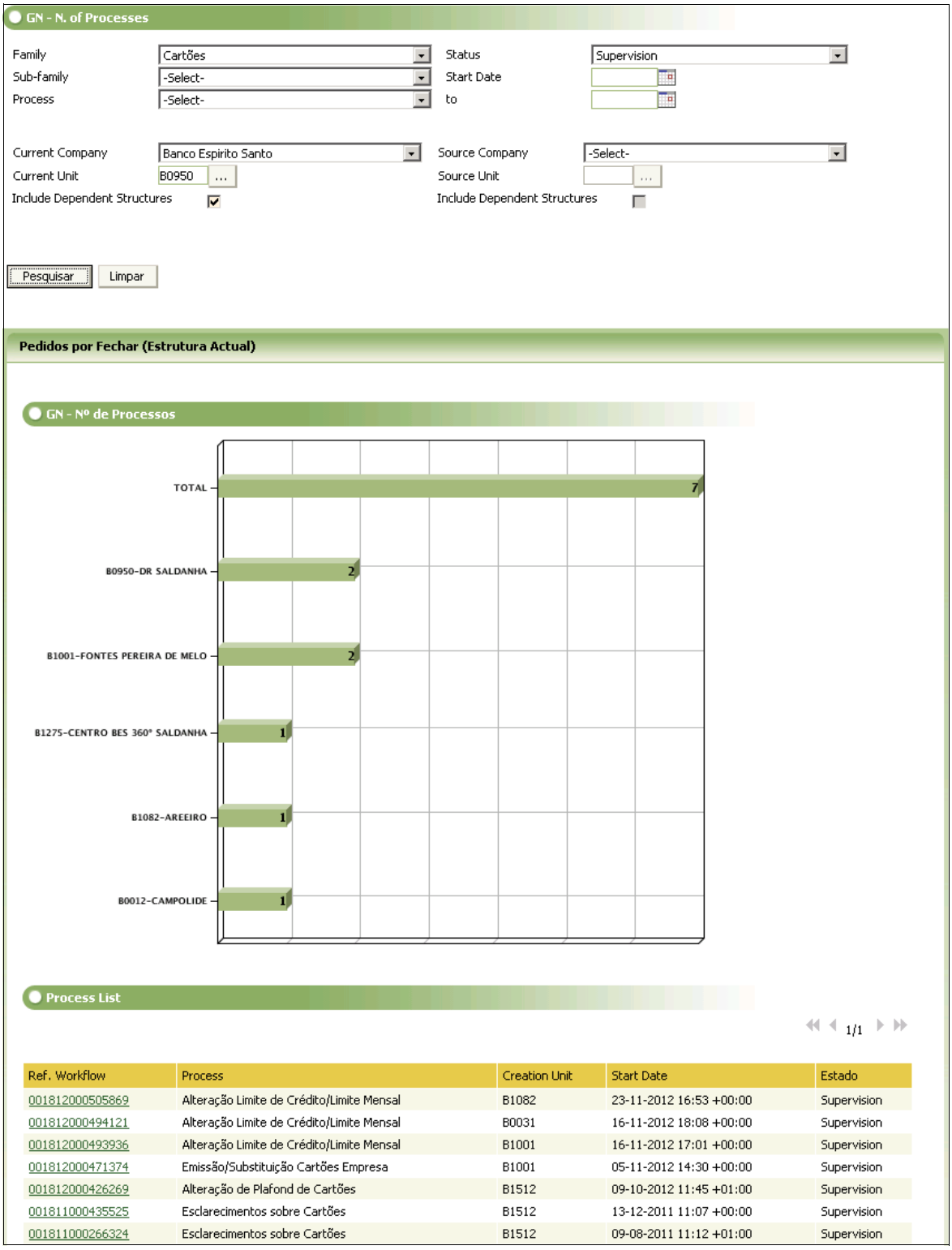


Figure 6-23 Scoreboard

Now any of the two people can select the sub-units (of the unit selected at the top level) where those processes are currently waiting for dispatch.

Figure 6-24 shows a different scoreboard that is used for process instance volume metrics. For this example, the CoE wants to understand whether a particular process (the request for a copy of the credit card monthly statement) has high volumes and whether some deeper analysis is required. In this situation, this particular scoreboard can be useful.

Figure 6-24 Scoreboard

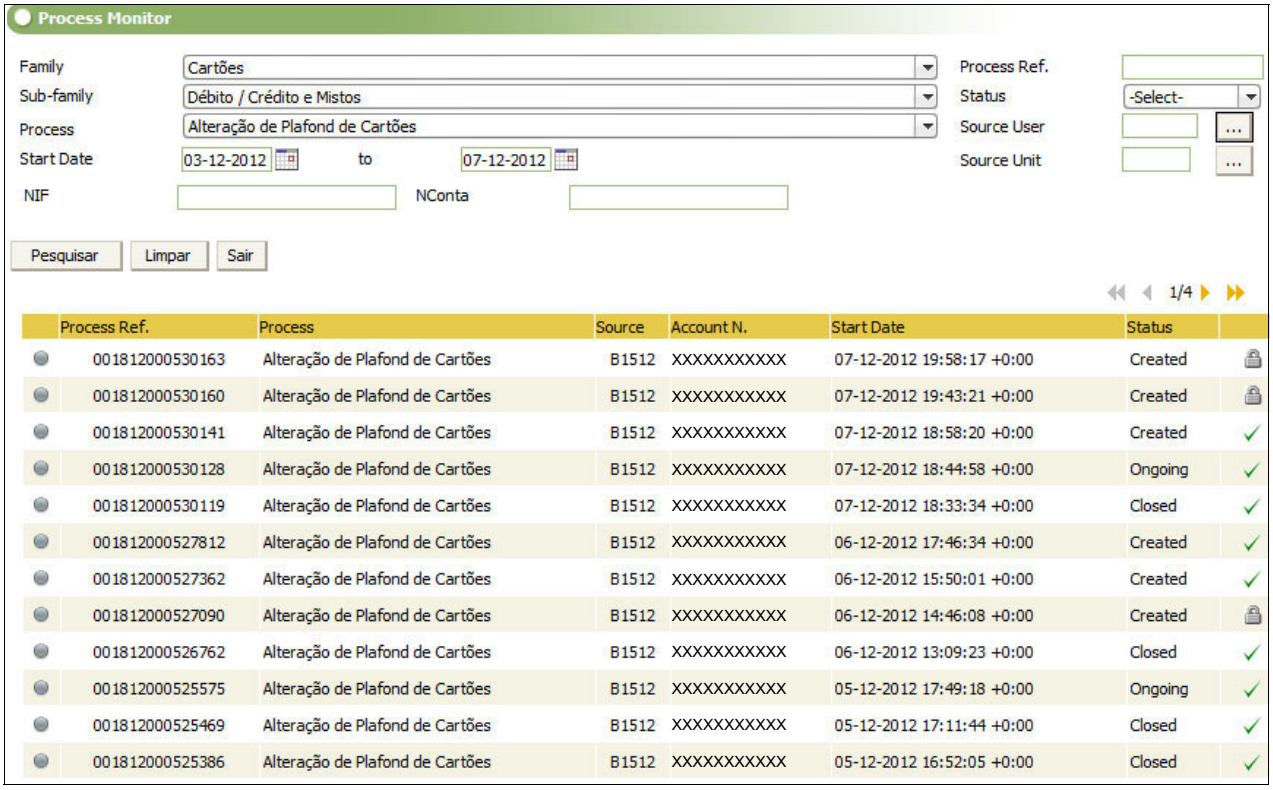
You can also drill down for instance of 2012 and check the monthly volumes for it, as shown in Figure 6-25.

Figure 6-25 Scoreboard

With this data, the CoE can check, for this particular process, whether the number of instances per month is low. From this information, the CoE can determine that unless something else justifies a different implementation (such as risk or legal factors), this process can continue to be supported on the Generics platform.

Online reports

Another tool can be used to provide the business with Generics data. This tool enables authorized users to query the GPE supporting database to view process instance specific data. It shows data for each process instance that is returned by the query. Figure 6-26 shows a query for credit card limit change-request process instances that were created over a specific interval (in this case, one week).



Process Ref.	Process	Source	Account N.	Start Date	Status
001812000530163	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	07-12-2012 19:58:17 +0:00	Created
001812000530160	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	07-12-2012 19:43:21 +0:00	Created
001812000530141	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	07-12-2012 18:58:20 +0:00	Created
001812000530128	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	07-12-2012 18:44:58 +0:00	Ongoing
001812000530119	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	07-12-2012 18:33:34 +0:00	Closed
001812000527812	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	06-12-2012 17:46:34 +0:00	Created
001812000527362	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	06-12-2012 15:50:01 +0:00	Created
001812000527090	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	06-12-2012 14:46:08 +0:00	Created
001812000526762	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	06-12-2012 13:09:23 +0:00	Closed
001812000525575	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	05-12-2012 17:49:18 +0:00	Ongoing
001812000525469	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	05-12-2012 17:11:44 +0:00	Closed
001812000525386	Alteração de Plafond de Cartões	B1512	XXXXXXXXXXX	05-12-2012 16:52:05 +0:00	Closed

Figure 6-26 Online reports

So, online reports allow BES to list those instances with the selected status. BES can drill down into each instance and view the complete history for this instance up to this point in time. This data is available immediately when the instance is created and continues to be available after the process is complete. There is a selection of search criteria, including status, case reference, customer data, and so forth.

Daily data exports

All the scoreboards and reporting data capabilities discussed previously are (near) real-time views into the system. BES can use these reports for operational analysis but cannot use them for historical perspective, that is, to go back to some point in time. For example, if BES wants to know the SLA status for a particular process on the last day of the previous month, these tools cannot provide that information. However, this type of requirement is supported through another avenue.

A set of files is produced on a daily basis that is distributed to a set of entities and systems, as follows:

- Data warehouse

  All data that is processed is stored in a data warehouse. So, BES can retrieve SLA information from the data warehouse.

- Business departments

  Interested departments receive the daily files and can build a variety of reports using them. So, if they have a particular need that is not covered by the existing reports and scoreboards, they generate custom reports from the files. This information is also used by business departments to get a better view of a specific process and how it is being used within the organization.

6.8.2 IT team activities

Although the IT team uses the monitoring utilities that were described previously (particularly for information about process volumes), a separate set of utilities exist for infrastructure management. These utilities are fundamental to understanding information such as the growth, over time, of resource consumption of the Generics related processes. This growth does not depend exclusively on the volume of new processes per day. For example, assume that there are issues with the process itself and there is a performance degradation for instances as a result of a full table scan on a table that is growing over time. The number of new processes per day, and the actions that those processes take, can remain fairly constant, but the overall resource consumption will gradually increase as the average response times for the UIs increase. So, it is key to understand the overall system behavior and the effects introduced by each process. In fact, these scoreboards are used for all the processes running on the IBM Business Process Manager platform, but the following examples are focused on Generics use cases.

Number of UI requests

Figure 6-27 shows a scoreboard with the following information:

- Daily number of user requests across all Generics Process Application UIs (vertical blue bars)
- Daily average response times on the UIs (red line)
- Trend lines for user requests (blue) and average response times (red)

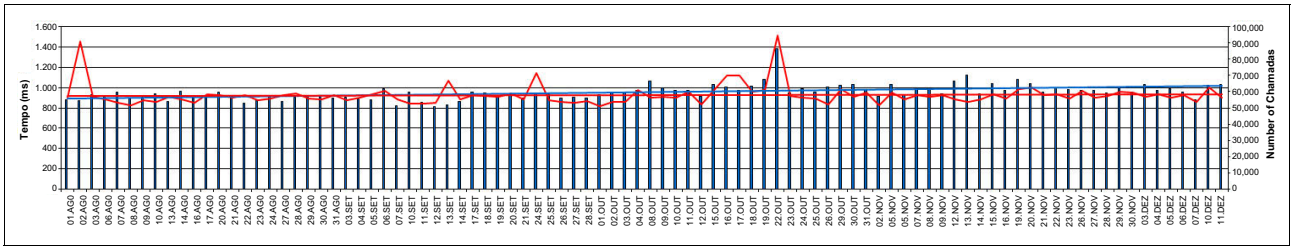


Figure 6-27 Number of UI requests

The left vertical axis indicates the average UI response time in milliseconds and the right vertical axis indicates the number of user requests on Generics UIs. This information indicates even a slight increment on the number of requests with stable response times. This same scoreboard has been used to anticipate problems and to confirm improvements on several processes.

UI response times

Figure 6-28 shows a scoreboard used to analyze the Generics UIs response times per day, organized by categories. In this case, response times are within a certain threshold.

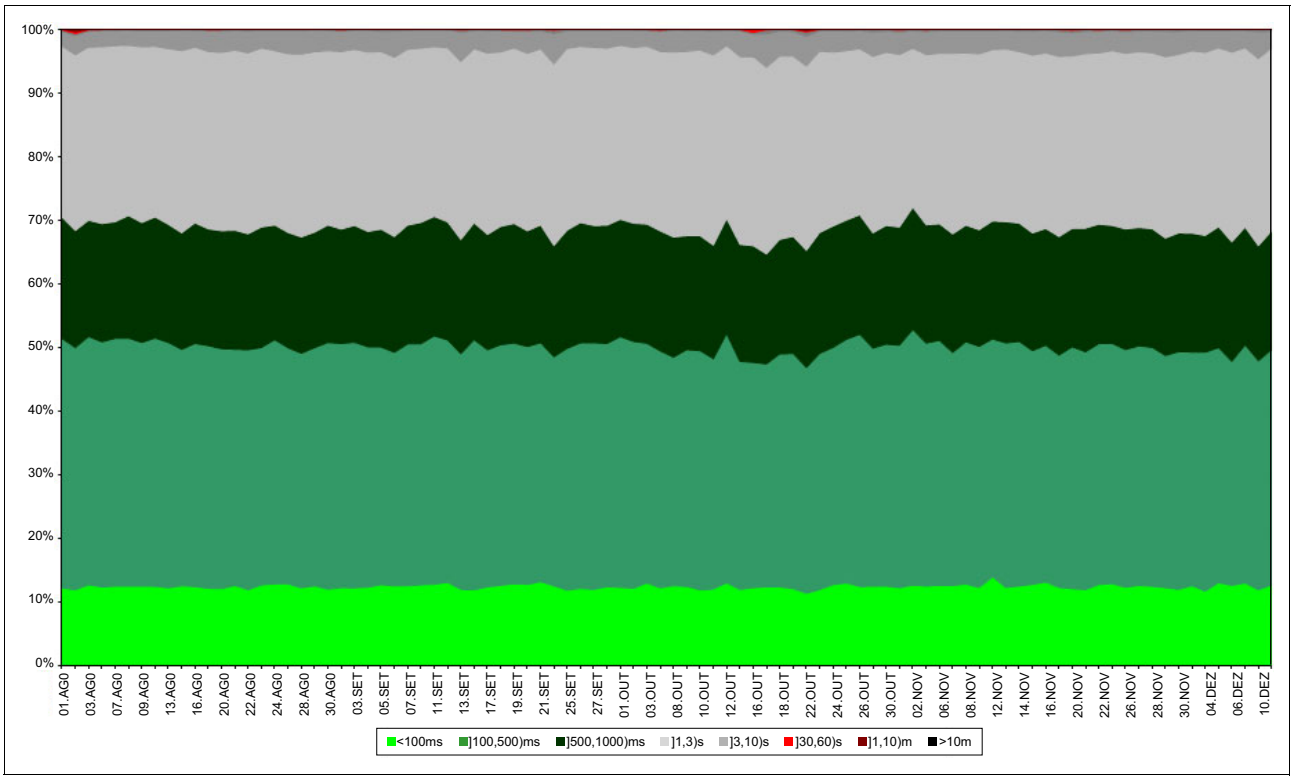


Figure 6-28 UI response times

So, in addition to the average response time, the response times analysis can go deeper. The legend enables checking that, for example, approximately 70% of the total requests are answered in under one second, and the number of requests answered in over three seconds is less than 5%. In addition to trends, the impact of changes on the process assets is also reported.

Server-side resource consumption

Figure 6-29 shows another important scoreboard where the server-side execution times are grouped per process application. This information gives an idea of the relative resource consumption for each of those process applications.

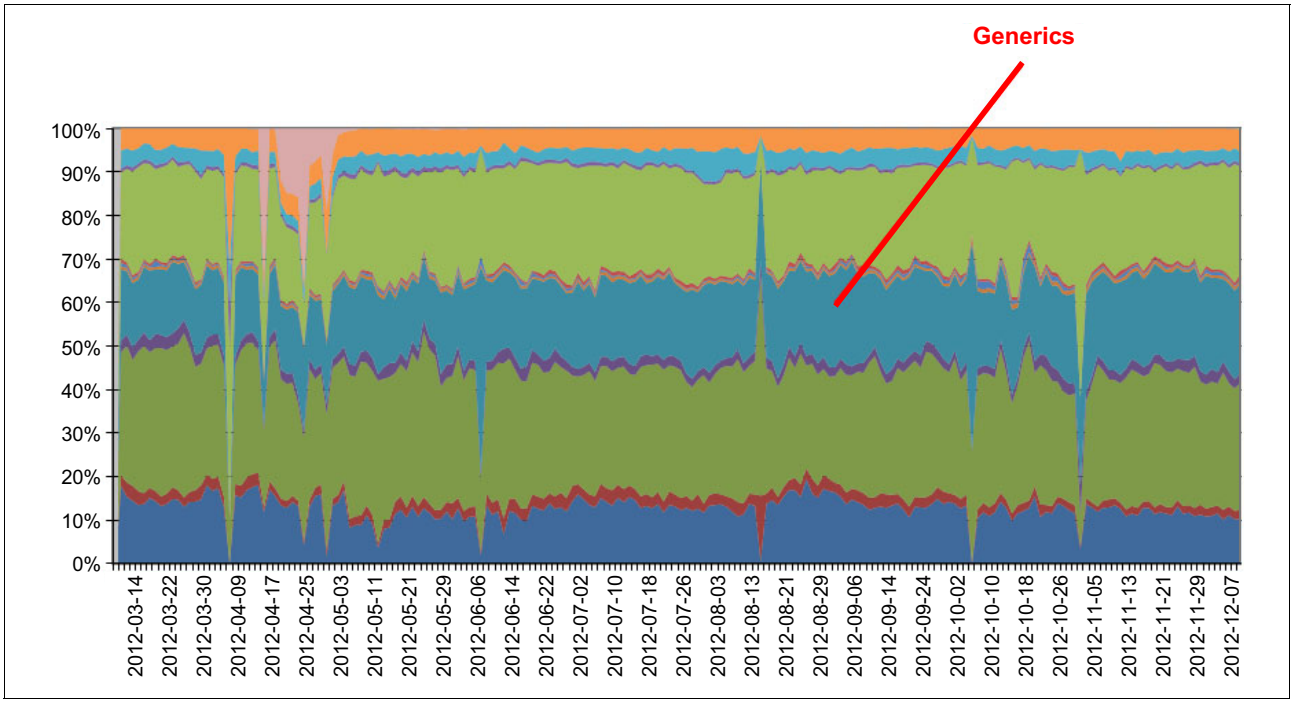


Figure 6-29 Server-side resource consumption

This view provides the following conclusions:

- Generics account for about 50% of the total number of new instances created per day.
- The load that the Generics processes impose on the system is approximately 20% of the total. This load is considered normal because the Generics UIs are not complex and there is a small number of activities per instance.
- The complex processes, with highly complex UIs, are much resource demanding and for a much lower number of instances.

This scoreboard is also important in helping to estimate resource needs (CPU, RAM, and so forth) to support each new process application or the impact of any special activity for the existing ones. Any anomaly or abnormal usage of resources is also reflected and its impact can be measured.

Another important aspect for IT is historical data mining, with which IT tries to understand the typical paths for a process, and use that when processes are being implemented on different frameworks. At present, this data mining is not automated, but BES is working with IBM Research on a joint effort to discover behavioral and statistical models from Generics data. Although the initial goal is to provide insight into historical Generics processes, the hope is to eventually deliver insight into, and help improve, the outcomes of live Generics processes.

6.9 Conclusion

After choosing a new BPM platform in 2006, changes were clearly occurring and the bank was getting closer to what it needed. However, also clear, was that the journey was not yet over. The success of each implementation generated greater demand for more process implementations in the organization.

Generics processes did provide several benefits:

- Support for ad hoc processes
- Rapid implementation of processes
- Organizational autonomy of BPM

Perhaps the most significant benefit is from the autonomy that Generics processes offer to the business teams and the large scale roll out (the BPM democratization) that results from this autonomy. Anyone inside the organization is now a potential BPM user for methods that might be for small processes, such as intra-departmental administrative functions, or for processes directly related to the bank's customers.

The Generics V2 implementation will bring a new set of features and important enhancements that mean even more benefits to the business.

7

Next steps

The paper discusses various aspects of ad hoc processing in the context of business processes. It concentrates on two main areas:

- The set of ad hoc capabilities that are built into the BPMN environment.
- By extension, the ad hoc capabilities that are available in the IBM Business Process Manager environment.

The paper also illustrates the ad hoc approaches to process implementation with generic processes and event-driven processes. Depending on your point of view of these discussions, you might consider the following potential next steps:

- Ad hoc capabilities are not for you, and you will continue with structured and well-ordered business processes only.
- Ad hoc capabilities look useful, and you want to determine how to make appropriate use of them.
- You are completely sold on using ad hoc capabilities as much as possible.

There are two further considerations:

- Whether to use ad hoc processes will often be decided on a process-by-process basis. For different scenarios within the organization, different paths might be applicable or preferred. Thus, a single organization might fit into more than one, or even all, of these categories.
- Often, the situation might not be what you want to do, but what your environment is capable of supporting. For instance, although an appealing approach might be to give users access to ad hoc requests, such as cancelling a process instance, providing instant messaging between knowledge workers, or having knowledge workers dictate the order of task execution, this approach might not be practical for the class of users involved or the capabilities that you currently can deploy. For example, such capabilities might be restricted to supervisors.

Regardless of your opinions and conclusions, the question at this point is how to make forward progress. That is the focus of this chapter.

If you decide that ad hoc processing is not appropriate for either your set of processes or your organization as a whole, nothing is wrong with this decision. Although this paper advocates for the use of ad hoc capabilities, there is nothing wrong with deciding that they are not applicable to a process or organization. The structured and ordered approach to business processes is a perfectly reasonable way to proceed.

The important consideration is that you make this decision after gaining an appreciation for the capabilities that ad hoc processing can provide and not simply dismissing ad hoc processing without consideration. For processes that fit into this category, there is nothing new to do and there are no next steps. Existing processes need no further action and new or planned processes do not require altering their design. However, you must follow the leading practices for process development. Those practices are not described here because they are covered in many other publications.

From an overall viewpoint, consider the following general advice for next steps:

- Making a start is important. The message is not to overanalyze; projects that do overanalyze tend to never get started. You can begin with something simple, for example an ad hoc event that cancels a process instance or that waits for a document to arrive, and then gain experience and move on from there.
- BPM is an ideal environment on which to implement capabilities in an incremental way. So adding more incremental capabilities in a step-by-step manner and at a pace that suits the particular scenario is reasonably straightforward.
- You can exercise this style of incremental development for one scenario at a time or for multiple scenarios simultaneously. Whichever style you employ, the objective is to think consistently for the set of scenarios that exist within the organization. Make sure, for example, that common capabilities (such as the aforementioned cancel process instance scenario) are implemented in a consistent style for the set of process models.

7.1 Using ad hoc capabilities built into Business Process Manager

Getting started with IBM Business Process Manager ad hoc capabilities is a matter of figuring out how to use them, where to use them, and whether using them is appropriate. You will notice various approaches for each process model that you work with, although over time, some patterns will likely emerge, as with customer facing processes that always need a cancel capability.

The next steps for this category of ad hoc processing are not much different from any other style of BPM development. The capabilities used are a part of the BPMN development palette and are an integrated part of the IBM Business Process Manager development environment. Thus, you do not have to do anything particularly different from the set of tasks that you do today (assuming that you are using the IBM Business Process Manager development environment).

The capabilities that are available are illustrated in the Banco Espírito Santo (BES) case study discussed in Chapter 6, “Case study: How Banco Espírito Santo approached ad hoc processes” on page 59.

You can also find information in the IBM Business Process Manager Information Center to get started with using these capabilities:

http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0m1/index.jsp?topic=/com.ibm.wbpm.main.doc/ic-homepage-bpm.html

Be aware of the following set of technical BPM aspects to successfully implement ad hoc capabilities with IBM Business Process Manager:

- Events and event handling

  Much of the ad hoc processing within BPM is implemented with ad hoc events. These events are the way that *out of band* activities are passed into a process instance. So, be sure you understand the types of events, how they are modeled, and what infrastructure is necessary to enable them. For example, you need to know how to insert events into the BPM environment to make use of the event handling capabilities.

- Message-based processing within IBM Business Process Manager

  Many (but certainly not all) events arrive in the IBM Business Process Manager environment through messaging. So, you need to understand how IBM Business Process Manager supports and works with a messaging environment (for example, Java Messaging Service).

- Event subprocesses

  Many events are handled by event subprocesses. You need to understand how to model these types of events.

- Document events

  These types of events are a new style that can be handled by IBM Business Process Manager. These events require some configuration in the target content management system. Thus, some knowledge of this environment is required. (At the time of writing, only the IBM FileNet® Content Manager environment is supported.)

- Dynamic workflows

  You can build much ad hoc capability into a process model by designing flexibility into it. Rather than leading the knowledge workers from one specific task to another, you can build the process model so that it is straightforward for knowledge workers to navigate between tasks they want. No specific ad hoc skills are required here. It is more a matter of the style that is used to construct the process by giving more choices to the users.

- User interaction

  Users can interact with one another in multiple ways while executing a process instance. They can use instant messaging (where that is integrated into the IBM Business Process Manager Process Portal), they can use task collaboration, and they can use the social stream to share information about the process.

  In some ways, this type of process is the most challenging aspect of ad hoc processing from an enablement and execution viewpoint, because these facilities are used by users. Consequently, each and every user who wants to or must make use to them must be enabled (educated); there is a challenge of scale, user turnover. and inappropriate use. One approach to this set of challenges might be to enable only a small set of users and then to see how organically the knowledge spreads from those users to others.

7.2 Non-traditional BPM approaches

This section discusses the non-traditional BPM implementation styles from previous chapters, in particular generic processes and event-driven processes.

7.2.1 Generic BPM processes

The following general styles of generic processes are available:

- Use of a generic process as a template for other, similar processes
- Use of a single process that has the capability to execute different process types, which are controlled through external configuration

The first process, although technically a generic process, is still too general for a BPM environment. The challenge, in terms of getting started, is to find a suitable set of processes for which to build a generic template and then build out of that template. After you have a set of processes, you can look for the common elements and then build out your template. Too often, of course, this process happens in a natural unplanned manner, where you notice that a particular, existing process can be a good base from which to implement a new process model.

For the second style of generic process, keep in mind the following additional considerations:

- Determine that you really do have a need for a generic processing environment.

  Much initial effort is involved in implementing a generic process; be sure you are implementing for the correct reasons. Look back at the case study, and review the reasons that Banco Espírito Santo had for implementing their Generic Process Engine environment. You do not need to have all of the reasons that the company had, but this case study provides some clues as to whether you are on the correct track.

- Consider the set of capabilities that you want to provide.

  Determine the set of capabilities that the generic process will support. This set is generally more than the IT team wants to provide and less than the business users want to see. In keeping with the “start simple and grow” mantra, the leading practice is to begin with an implementation that is reasonably limited in scope. This approach makes the generic process easier to build and makes the configuration mechanism easier to build. Both the generic process and the configuration file can be made more sophisticated over time.

- Think about how you will implement the external process configuration.

  You have many choices for how to implement the external process configuration. There is no specifically correct choice. There are spreadsheets, flat files, databases, business rules, and anything that matches the following high-level criteria:

  – The mechanism must be appealing to and usable by the constituents (usually business teams) that will complete the configurations.

  – The mechanism must be a simpler and more appealing alternative to the BPM tool that is used to build a process, or you might as well use that and build a custom process definition.

  – The mechanism must be independent of the IT teams. Generally, a major goal for the generic processes is to provide relief (or a way around) the limited numbers of IT resources. Whatever mechanism you choose must not involve significant IT resources or you have not gained much.

For example, in the case study, Banco Espírito Santo used a database as the process configuration mechanism. Banco Espírito Santo also had a requirement that the business

users can easily build a process configuration. So the company built a facility for defining the process description in a spreadsheet format and built a utility to act as a front-end system to the configuration database.

7.2.2 Event-driven processes

The second ad hoc style illustrated is event-driven processes where no obvious, formal, end-to-end BPM environment exists, and there is a set of application parts that are loosely coupled together by some set of events. As noted in previous chapters, although you have no formal model for the process, often a model exists, although you do not know what it yet.

The difficult part of making progress with event-driven processes is in instrumenting the activities so that you can make progress with managing the process and discovering more about it. Instrumentation allows you to capture metrics on which application parts run, when they run, how long they run for, and so forth. This method is the first step in discovering what the event-driven process actually does (that is, which application parts are used and in what order) and its performance characteristics.

The following approaches are most useful to instrumentation:

- Implement instrumentation in the application parts.

  Although this approach is possible, generally it is the least appealing option. This process requires customization for each of the parts or as many parts as possible. You do not necessarily need to instrument every application part; this is certainly a case of what you get back being closely linked to what you put in. But, this approach is not a scalable way to do things and is often not even a possibility for some application environments. If you choose this option, you do not need to instrument each application part and you do not have to do it all at once. You can get useful information even if you instrument only a subset of the total.

- Instrument the events.

  The events that bind the event-driven process together can be intercepted and instrumented from a single place. A common way to implement the events is with messages. Intercepting messages with an enterprise service bus (ESB) and providing monitoring data can be fairly straightforward. This method becomes less easy when the events are of different types, although many ESBs (such as IBM WebSphere Message Broker) support multiple events types and can become challenging when no easy way to intercept them is available. So, if you have a level of control over what is used to implement the events, pick something that can be intercepted by a component similar to an ESB.

- Insert a BPM layer.

  The messages are intercepted by a BPM layer, and the built-in instrumentation is used. This approach is largely the same as the second option. However, this approach overtly introduces a BPM layer into the mix rather than some other interceptor. This approach is unpopular with some organizations because one of the possible objectives of the event-driven style is to avoid a BPM implementation.

7.3 Resources to help you get started

You can find more information at the following sources:

- The BPM samples exchange includes many useful examples that are oriented around the ad hoc capabilities in BPMN. These resources all provide sample code, detailed explanations, and information about how to build a sample for yourself.

  http://bpmwiki.blueworkslive.com

- IBM Business Process Manager information center

  http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0m1/index.jsp?topic=/com.ibm.wbpm.main.doc/ic-homepage-bpm.html

Abbreviations and acronyms

BAM	business activity monitoring
BES	Banco Espirito Santo
BPD	business process definition
BPM	business process management
BPMN	Business Process Modeling Notation
CoE	Center of Excellence
EAI	Enterprise Application Integration
ECM	Enterprise Content Management
ESB	enterprise service bus
GPE	Generic Process Engine
IBM	International Business Machines Corporation
ICU	intensive care unit
IM	instant messaging
IME	Intermediate Message Event
ITSO	International Technical Support Organization
KPI	key performance indicator
RFP	request for proposal
UI	user interface

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic 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
- *Creating a BPM Center of Excellence (CoE)*, REDP-4898

Online resources

These websites are also relevant as further information sources:

- IBM Blueworks Live:

  https://blueworkslive.com

- IBM Sample Exchange for Process Solutions for toolkits:

  http://bpmwiki.blueworkslive.com/display/samples/SAMPLE+EXCHANGE+HOME

- IBM Business Process Manager Information Center:

  http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0m1/index.jsp?topic=/com.ibm.wbpm.main.doc/ic-homepage-bpm.html

- BPM samples exchange:

  http://bpmwiki.blueworkslive.com

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Empowering your Ad Hoc Business with IBM Business Process Manager

Realize the benefits of BPM when dealing with ad hoc processes

Take advantage of IBM Business Process Manager tools

Read a real-world success story of a BPM implementation journey

In the context of daily business, ad hoc processes are those activities and events that occur within an organization's operations that typically are undocumented or unmonitored. At times, these ad hoc processes can seem chaotic and unpredictable. In many cases, these "off the platform" processes represent an opportunity for you to realize visibility into your organization operations. By taking advantage of the benefits of business process management (BPM) and IBM Business Process Manager solutions, you can bring order and stability to these business processes and improve the organization's agility in order to stay adaptive and competitive.

This IBM Redpaper publication presents examples and a case study that illustrate how having a choice of where on the ad hoc spectrum you operate your business is both necessary and vital to producing better outcomes and achieving agility. You need agility to stay relevant and to survive. The intent of the prescriptive framework in this paper is to give you the confidence and motivation to choose how much business agility you want and to begin achieving it. This paper is intended for Executive Sponsors, Team Leaders, Lead Architects, and anyone interested in adding business agility and ad hoc processes to their enterprise.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

REDP-4995-00