



An Oracle White Paper
July 2012

Reaching Process Excellence with Oracle Process Accelerators

Using Best Practices to Accelerate BPM Value Creation

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Executive Overview.....	3
The Evolution of BPM: From Success to Process Excellence ...	3
The Path to BPM Process Excellence.....	5
The Oracle Answer: Process Accelerators.....	6
PA Best Practices: Getting “Under the Hood”	12
PA Best Practices: Process, Services	13
<i>BPMN Pattern: Top-Down Business Decomposition.....</i>	13
<i>BPMN Pattern: Leverage Pre-built Routing Patterns</i>	15
<i>BPMN Pattern: Business Fine-Tuning with Business Rules</i>	16
<i>BPMN Pattern: Start with Business Indicators, Get to Dashboards</i>	18
<i>Services Pattern: Define Business Service Contracts.....</i>	19
PA Best Practices: User Interface Component Architecture	21
View Pattern: Plug into OOB Oracle BPM Human Tasks / Workflow...	22
View Pattern: Reuse Page Fragments	23
Controller Pattern: Reuse Page Flows	24
Model Pattern: Place Business Logic in the Appropriate Layer	25
Putting it all together: TRM UI under the PA Reuse Pattern lens.....	27
Pattern: Application Component Modularization	28
Conclusion	30

Executive Overview

Oracle Process Accelerators help organizations reach process excellence faster. Process Accelerators are business process solutions developed with Oracle Business Process Management (BPM) Suite 11g. Process Accelerators can be deployed as-is, or extended to meet customer-specific requirements. In addition to expediting time-to-value for BPM deployments, Process Accelerators embody best practices and serve as blueprints for organizations that are developing process driven solutions with Oracle BPM Suite. Organizations adopting Process Accelerators not only improve the business process targeted by the Accelerators they deploy, but also have a unique opportunity to reach maturity in their process management initiative faster and with lower risk by applying Process Accelerator Best Practices and Patterns.

The Evolution of BPM: From Success to Process Excellence

Organizations increasingly recognize the value of using Business Process Management (BPM) software, as seen by the accelerating rate of adoption due to BPM's transition to mainstream enterprises. The promise of greater efficiency, business visibility, and agility in process management is what drives organizations to invest in BPM. The current trends show a growth in BPM at an increasing rate, with organizations moving their focus from lower infrastructure layers to higher layers of abstraction. This change in focus is driven by the need to execute processes end-to-end, beyond the commoditized processes that can be supported by simply configuring within the boundaries of packaged applications. BPM is effectively a bridge into a future in which apps are powerful yet flexible, optimized to execute against a specific business process yet fully extensible and ready to plug-and-play into other apps.

Once an organization decides to invest in the BPM model to improve operational efficiency, it needs to espouse a new paradigm where success is not simply measured by the ROI of a single initiative, but where results are consistently delivered over a long period of time. After all, the BPM end game is not just to solve a single process, but to reach "process excellence": execute efficiently against multiple process management initiatives, delivering successfully against a growing portfolio of business processes.

What are the determinants that lead to process excellence? Plenty of attention has been devoted – and rightly so – to the importance of change management, leadership support, and business involvement in kick-starting the BPM model.

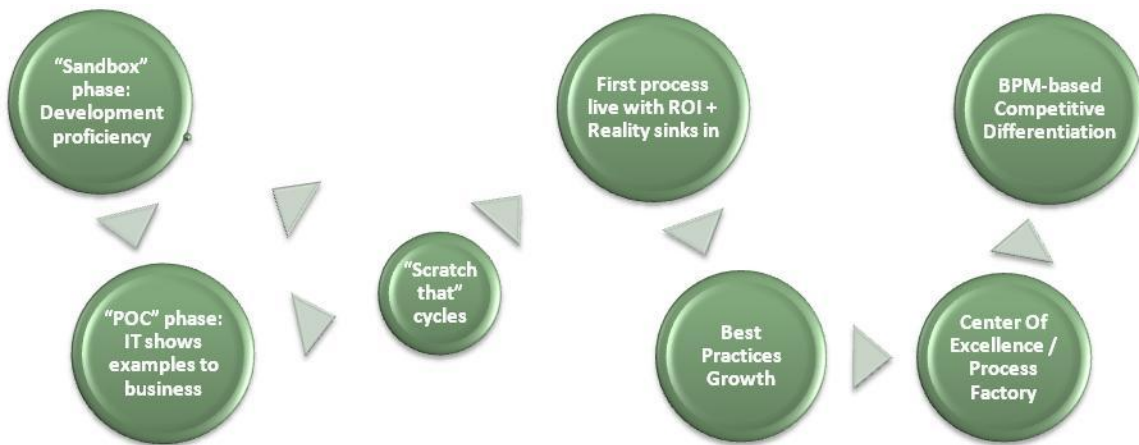
Here we focus on how to help organizations reach, once BPM has been accepted, into a high-throughput joint Business-IT delivery model. The end results we are aiming for are the following, all together enablers of process excellence:

- *Business masters a requirements-to-value BPM generation model.* It's easier to see an application and suggest how to configure it rather than starting from a "blank sheet of paper". While BPM typically targets processes that are not standardized into a packaged application (even if some steps in the process may be), Business can still move efficiently from visualization to solution by adopting a model that merges the efficiency of a pre-built solution with the flexibility of an iterative BPM delivery model.
- *IT masters BPM skills.* Enterprises thinking about BPM strategically know they need to be serious about developing internal competencies long term. Most IT organizations want to leverage existing Java or .NET skilled resources and train them to be productive and cooperative BPM experts. Moving from a procedural and object-oriented to a business process oriented model can be facilitated by establishing patterns. These patterns can help IT not only speed up the development, but also partner with the Business, for example using BPMN to design business processes rather than flowcharts.
- *External and internal business steps are integrated in an end-to-end process.* BPM is anything but an island. Very few business-changing solutions are standalone. Integration can become a very expensive portion of BPM project delivery, especially in iterative improvement with a lot of moving parts. Having a truly integrated BPM and SOA strategy is a must, as well as having pre-integrated scenarios that include applications to start from.

Looking at the enablers above, the necessity for BPM vendors to provide organizations (at all levels of BPM maturity) with pre-built solutions based on a best practice delivery model becomes obvious. Organizations can use these pre-built solutions and the methodology that comes with them to establish a common business - IT model around how to iteratively improve BPM solutions.

The Path to BPM Process Excellence

How can organizations reach process excellence faster, becoming “BPM Factories”, delivering repeatable and iterative success? To answer this question Oracle has collected information from customers, partners, and analysts, and also internally, among the Oracle teams using BPM to deliver next-generation solutions and to choreograph Oracle apps. We have observed different patterns, with organizations going through the maturity phases represented in the picture below:



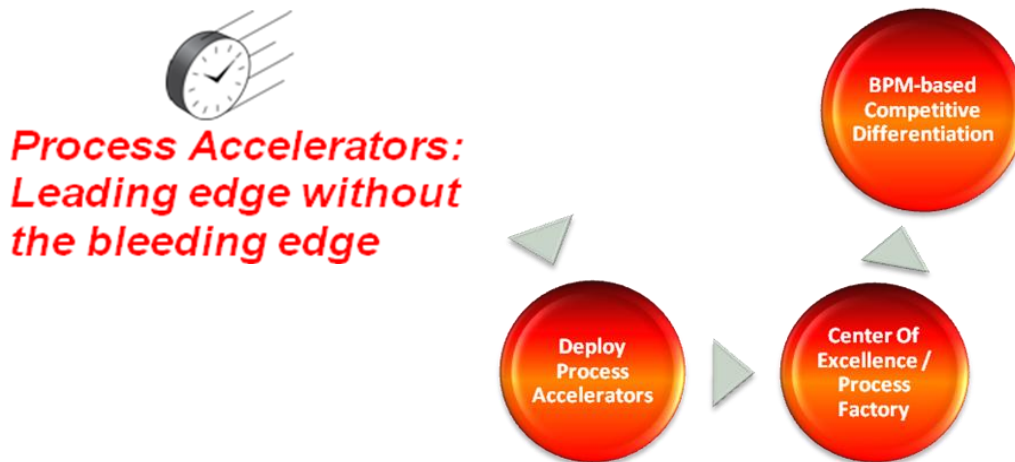
Typically, the path to process excellence is not a straight line. Organizations learn to harness BPM gradually, and in many cases go through what is shown above as “Scratch that” cycles, during which they realize that scaling in complexity and delivering enterprise-grade solutions and consistent success requires better models. Most organizations eventually reach success, generate consistent ROI, and proceed to become “process factories”, using BPM for incremental competitive differentiation.

An iterative process is very appropriate for BPM, with Business and IT working together at all times, with Business driving requirements iteratively and IT injecting architectural patterns, engineering for production, and reusing existing assets. How can BPM vendors help Business and IT accelerate progress towards such an iterative and architecturally sound delivery model?

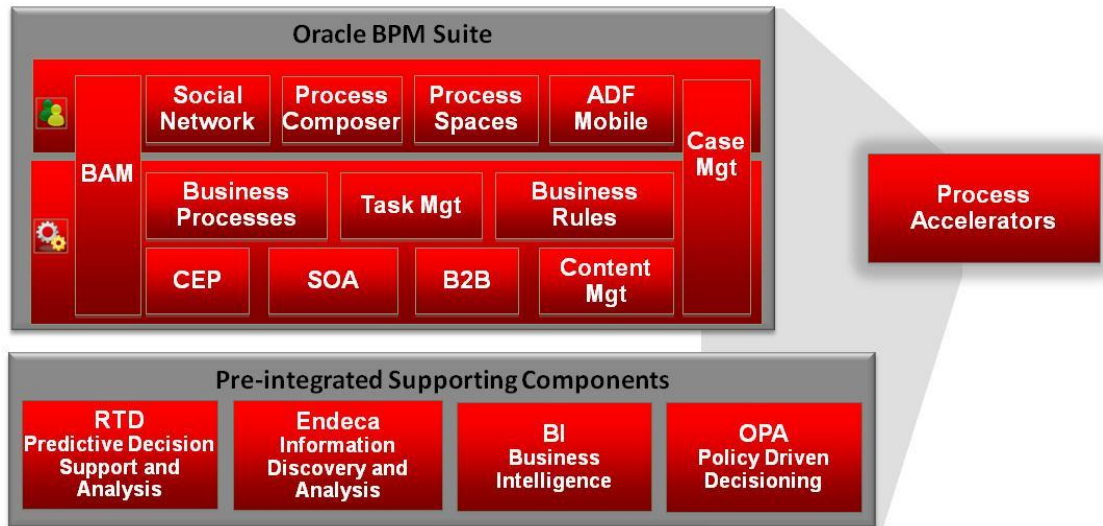
The Oracle Answer: Process Accelerators

Oracle's analysis of the path to process excellence has led to adding powerful BPM product functionality for both the IT (BPM Studio) and the business user (BPM Composer). It has also indicated that the most successful organizations are those with a high level of BPM proficiency and bold enough to target critical industry business processes, or the true drivers of sustainable differentiation in their industry.

The challenge for Oracle was how to shorten the customer path to excellence and allow every customer to fully leverage the power of the Oracle BPM Suite, in other words to offer all customers a "leading edge without the bleeding edge". The answer has been to invest heavily in BPM Process Accelerators.



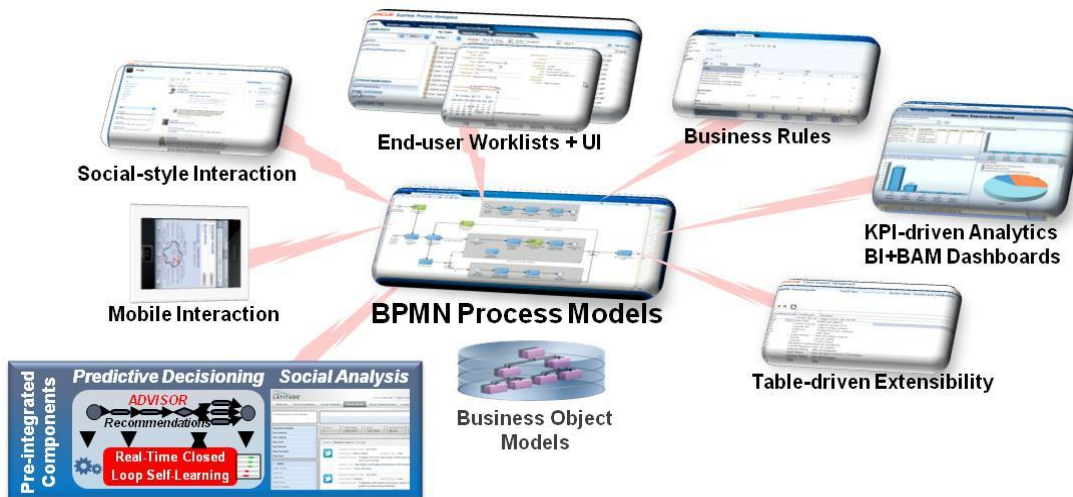
Process Accelerators (PA) are business process solutions developed and tested by Oracle, built on top of Oracle BPM Suite. Oracle BPM includes modeling and simulation tools (BPM Studio, Composer, Process Simulation), Business Rules, Human Tasks and Workflow, Business Activity Monitoring (BAM), Process Spaces for collaboration (built on top of Oracle WebCenter). BPM also layers on top of Oracle Service-Oriented Architecture (SOA), leverages Oracle Application Development Framework (ADF), the multi-channel JSF-based UI framework used also within many Oracle apps, including Fusion CRM; and Unified Content Management.



A set of very powerful tools, and well integrated. Besides, PAs leverage, when required by the business process at hand, both pre-integrated components such as Oracle Real-Time Decisions (RTD), Oracle Endeca, and Oracle Policy Automation (OPA), and services provided by other Oracle applications, such as Siebel CRM. All these pieces are essential to deliver complete solutions faster.

Overall, Oracle BPM delivers a punch, a very powerful and complete set of all-essential functionality. However, with power comes responsibility. Control is needed to harness the power, knowing when to use each tool to assemble solutions that are really flexible, reusable, and rapid. That's when PA best practices and patterns come into play, a model that business and IT can use to cooperate and thrive. This also helps IT adopt good architecture and working models for the new BPM tools and paradigm.

PAs demonstrate how the different capabilities of the Oracle BPM solution come together in a compelling fashion to address customer requirements end-to-end: the picture below illustrates some of the key components included in the PA framework.



The key principle of PAs is that they are ready-to-use, yet designed with extensibility in mind. They are not only valuable solutions per se, but also serve as a reference framework of best practice patterns that can be broadly used in any BPM implementation.

Process Accelerators target both critical industry processes and horizontal areas. When exploring “The Path to BPM Process Excellence” we have seen how organizations generate ROI with BPM when they target processes that are core to their business model. However, it is also risky to target these processes right away, before the organization has aligned both IT and business with less complex undertakings. Industry-focused PAs allow tackling core industry processes earlier, since the embedded best practices decrease the risk of running into the typical bumps in the road to process excellence.

PA Customer Value Proposition

- **Time to Sustained Value:** Reach Process Excellence faster and with lower risk
- **Adapt to Fit:** Reuse and extend pre-built solutions to your unique requirements
- **Process Optimization:** Monitor KPIs, refine
- **Best Practices:** Leverage embedded design patterns for effective change management
- **Enable Collaboration:** Align Business and IT around a common delivery model
- **Expand BPM Value:** Automate more aspects of business – industry specific and horizontal

Oracle continually evaluates possible candidates for future Process Accelerators that are the most pragmatic and valuable. These solutions respond to business requirements for a critical process in an industry and embed experience of domain experts from Oracle, design partners, and customers. Implementing organizations are likely to extend and modify Industry PAs as required by their specific requirements, adding their own “secret sauce”, or differentiating best practice. After all, the advantage of BPM is flexibility, and PAs are architected to be evolved.

For Oracle customers, the option to deploy Process Accelerators raises the potential value they can derive from an 11g rollout, by extending the benefits of BPM to a larger number of employees, and speeding up development”

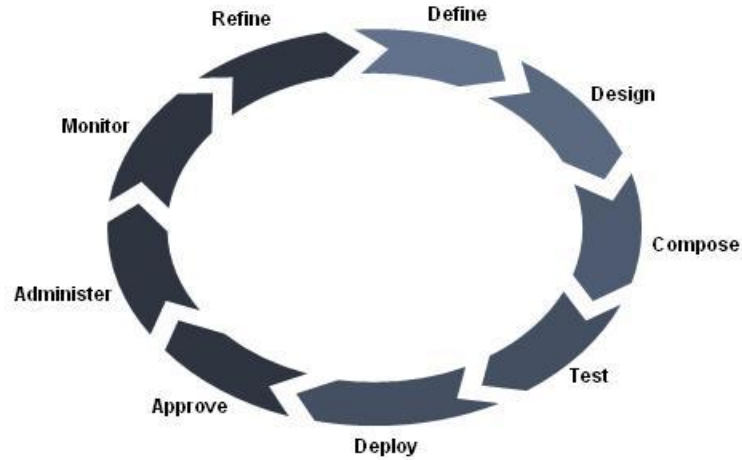
Somak Roy, Ovum O100125-024

Customers save significant time and effort even when heavily changing PAs. Some industry PAs may be standalone solutions, yet many integrate with existing Oracle Applications through generic interfaces that use standard business objects.

In addition to core industry processes, there are many horizontal business processes that are common to many organizations but usually not critical enough to justify the investment of scarce resources to perform them better. These processes therefore continue to be performed in ad-hoc fashion using email or spreadsheets, thus saddling organizations with higher costs, reduced productivity, and limited visibility and agility. Some examples include travel request management, document routing and approval, internal service request and delivery, and employee onboarding.

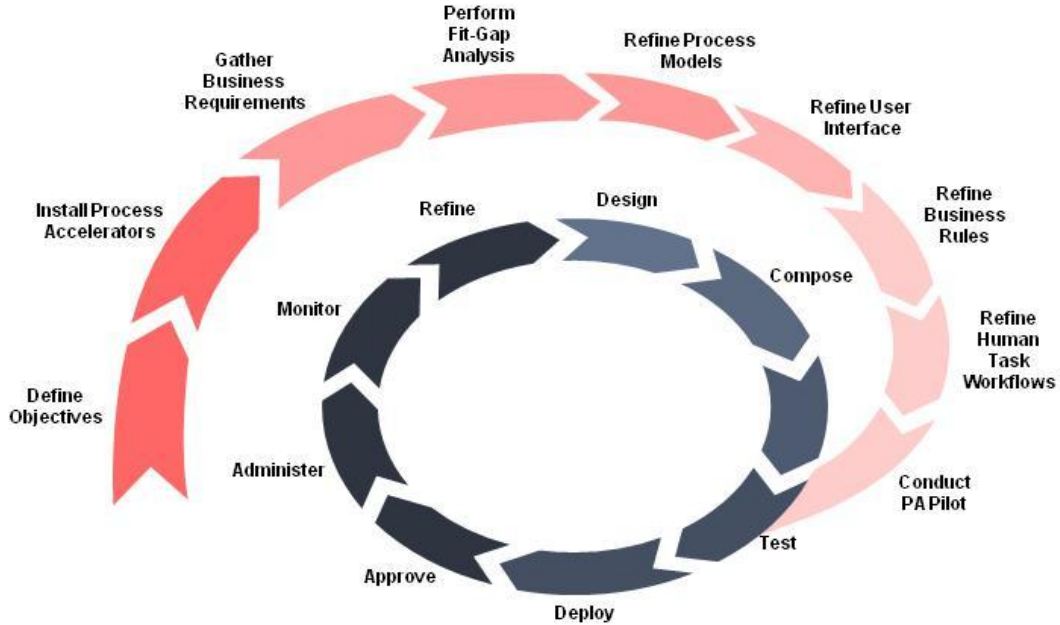
Horizontal PAs target these processes, and allow adopting organizations to both capitalize on the BPM investment, and to potentially increase their experience in delivering BPM solutions by targeting lower complexity and criticality (and therefore less risky) processes. Horizontal PAs still embed PA best practices, and therefore can help organizations set up a balanced and efficient joint IT - Business delivery model.

Such a process solution delivery cycle is here represented:



The delivery cycle starts with the identification and definition of the target business processes, and includes BPM composition (including all necessary services and UI), test, and deployment. The key concept here, regardless of the specific phases, is that this is a cycle: organizations in search of process excellence are expected to measure business process KPI and metrics, monitor business performance, refine the process, re-define, re-design, and re-deploy: they can't stop after one cycle. The delivery model needs to be fast yet solid enough to enable them to address the core business processes that are critical to compete within their industry.

PAs are also starter sets to help customers set up an efficient IT-Business delivery model faster. Here's a conceptual representation of how organizations can use PAs to "kick-start" the delivery model spinning speed.



In the model above customers select PAs relevant to their process scope and can immediately compare their business requirements to PA's requirement scope. They install one or more PAs so that business users can "touch" the PA functionality. Based on their priorities they can then refine KPIs, Business Rules, human workflow and conduct a production pilot before deploying. Every element reused by customers is structured according to best practices, and designed for extensibility. As a result, the organization naturally adopts best practices essential to an efficient BPM model.

PA Best Practices: Getting “Under the Hood”

Access to PA artifacts is critical to enable a leap to immediate productivity with BPM. When organizations have access to solid artifacts together with embedded best practices and a step-by-step methodology, they can also accelerate their transition from BPM “newbie” to leading edge, with considerably less risk to end up on the “bleeding edge”.

We have covered until now more the theory than the practice. It’s now time to get deeper under the hood of PAs to discover concrete best practices. We will mostly use Travel Request Management (TRM), a horizontal PA, as a case study for the example practices covered in this white paper.

Some of the best practices below could be summarized as “use the best tool for the job”. After all, best practices are not only about capturing very complex patterns that only architects are trained to recognize behind the business requirements. Especially when dealing with a complete toolset like Oracle’s BPM Suite, the simple identification of the path of least resistance to get to maximum results categorizes as best practice. These best practices embed experience in delivering results with a product that empowers both business and IT.

The best practices presented here are divided in two groups: Process/Services for BPM/SOA-related patterns, and User Interface-related patterns. While more patterns are used when delivering PAs, these two groups already show how important is to control all aspects of a BPM implementation. A BPM solution is composed of process, services, and user interface (among other components), and a complete BPM suite must provide the power to deliver all components with ease.

PA Best Practices: Process, Services

Let's examine first patterns in two areas:

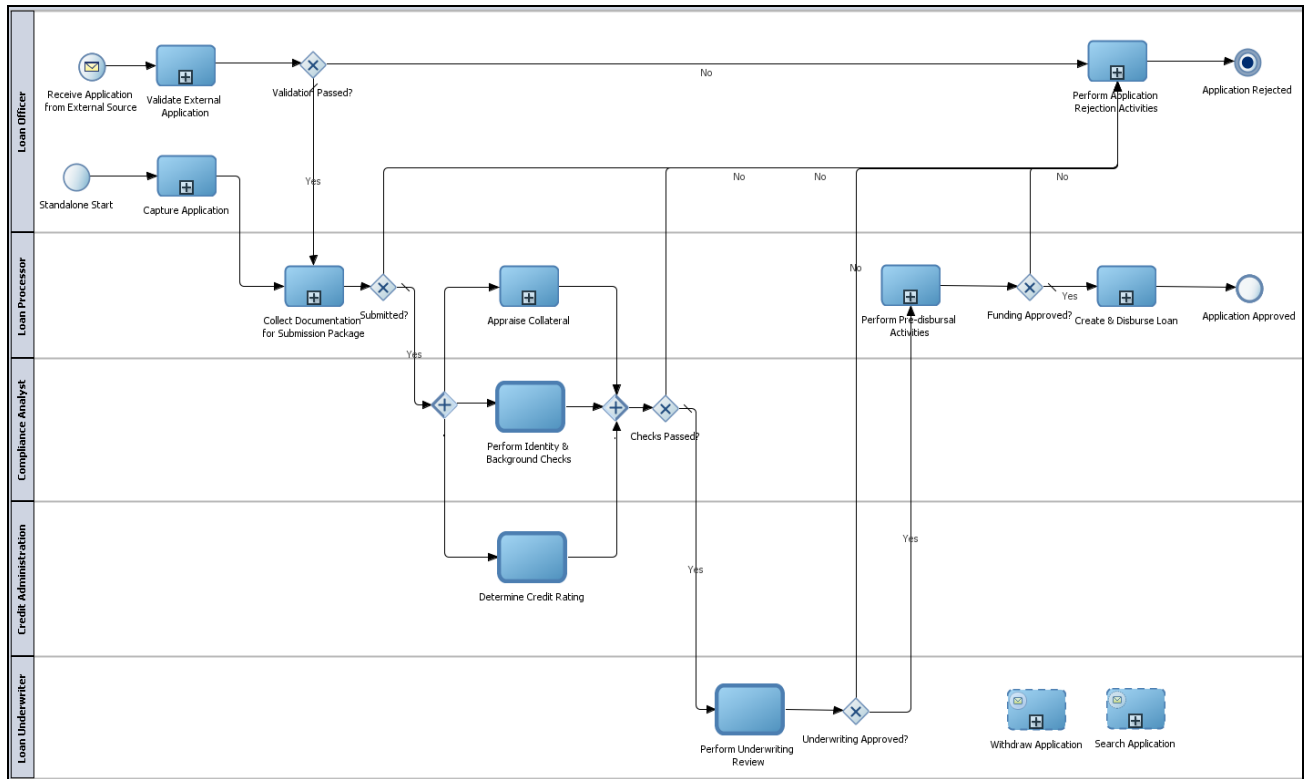
- *BPMN Patterns*: Top-Down Business Decomposition
- *Services Patterns*: Define Business Service Contract, use in BPMN

BPMN Pattern: Top-Down Business Decomposition

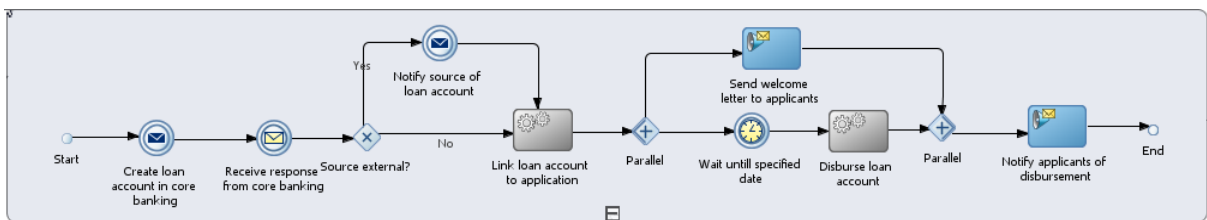
We have defined specific standards for modeling business process in Oracle Process Accelerators. BPMN 2.0 already includes three “conformance classes” for process modeling: *Descriptive*, *Analytic*, and *Common Executable*. The specification defines these classes in terms of what elements and attributes must be supported by a tool that claims BPMN 2.0 compliance. Conformance classes can be seen as different level of abstraction, from the highest level, *Descriptive*, to the lowest level, *Executable*. More specifically, *Descriptive* and *Analytic* conformance classes focus on the elements and attributes that are visible on a diagram and exclude hidden elements that are used primarily for execution. The *Common Executable* class adds elements and attributes required to execute a process.

When implementing PAs, we have started from BPMN 2.0 and have added rules that further constrain the freedom that comes from having a rich modeling representation. These rules cover naming conventions, use of symbols, common patterns, layout, exception handling, and other important topics. Here we don't want to necessarily present the specific rules, but rather focus on the guiding principles, in particular on those around the business level / descriptive model that help Business and IT stay in sync from requirements to production:

- *Maintain the top BPMN model 100% business-related*. There is no reason to introduce any logic beyond business logic in the top-level description of the business process. No matter how complex the BPMN flow is, it must be described in business terms first. It's a straightforward principle; however, given that IT is generally more fluent with BPMN than the business, there is a tendency to include functional aspects or even execution aspects into the top model. There is no need to be afraid of losing content, details can come later, and documented at the level where they are needed. A litmus test for the top-level business flow is to think about whether we'd feel comfortable presenting it to the executive that owns the underlying business process. If it's too much then it needs to be abstracted, for example by hiding details within subprocesses. Here's a top level BPMN flow for Loan Origination, very understandable by business users.



Oracle BPM 11g is very conducive to a top-down, business-driven process modeling, since it includes both a business modeling tool, Oracle Business Process Composer, and BPM Studio, an IT-oriented tool. Oracle BPM 11g support of BPMN 2.0 standard around Subprocesses provides the ability to easily decompose from business level to executable/IT level. As an example of decomposition, the details behind Create and Disburse Loan task in the above BPMN process are shown below:



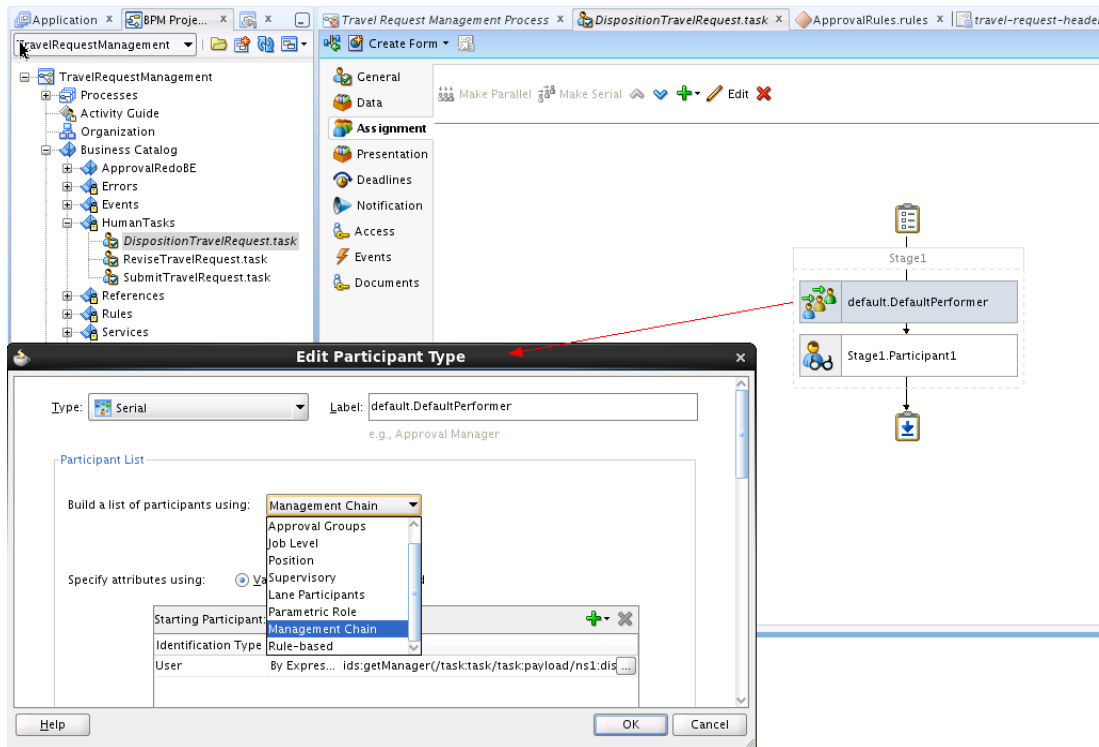
Technical details, for example around integration with Core Banking systems, are here specified.

- *Gateways are for business routing.* This principle derives from the one above. The purpose of BPMN gateways is often misinterpreted: they look like decisions in flowcharts, therefore a place where the value of a variable is checked and if-then-else decisions made based on that. However, BPMN models are not flowcharts. A BPMN gateway in higher levels is where business conditions are checked. Underlying variables are defined later. A good test of whether a gateway belongs to the top level is whether it leads to different tasks performed by different business users or roles.
- *Understand your Business Objects first.* In our experience the best way to design and also to communicate the high level business process is first and foremost to understand and describe the business object central to the business flow. After all, BPMN process work upon business objects such as Incident Report, Banking Loan, or Travel Request. While we may all think we know what the business object is, it's extremely useful to agree on a common definition, list key attributes with sample values before embarking too deep into designing the process that will drive the business object transformation. This becomes even more important in Case Management scenarios, where the business object drives the business process.

BPMN Pattern: Leverage Pre-built Routing Patterns

BPMN models include both automated/system tasks and human tasks. Human tasks include actions such as approve, reject, or provide more information. One of the sources of complexity when designing a complex human task workflow is determining who the next person down the line that needs to work on a task is. The average complexity of organization structures is increasing, driven for example by multiple departments and by the management hierarchy within each, particularly when a matrix reporting style is adopted. Determining the assignment rules dynamically based on these structures becomes essential.

Oracle BPM Suite facilitates this task by dynamically building the list of participants based on patterns defined in the Human Task suite component. The screenshot below shows the Assignment tab in a human task (TRM's DispositionTravelRequest).

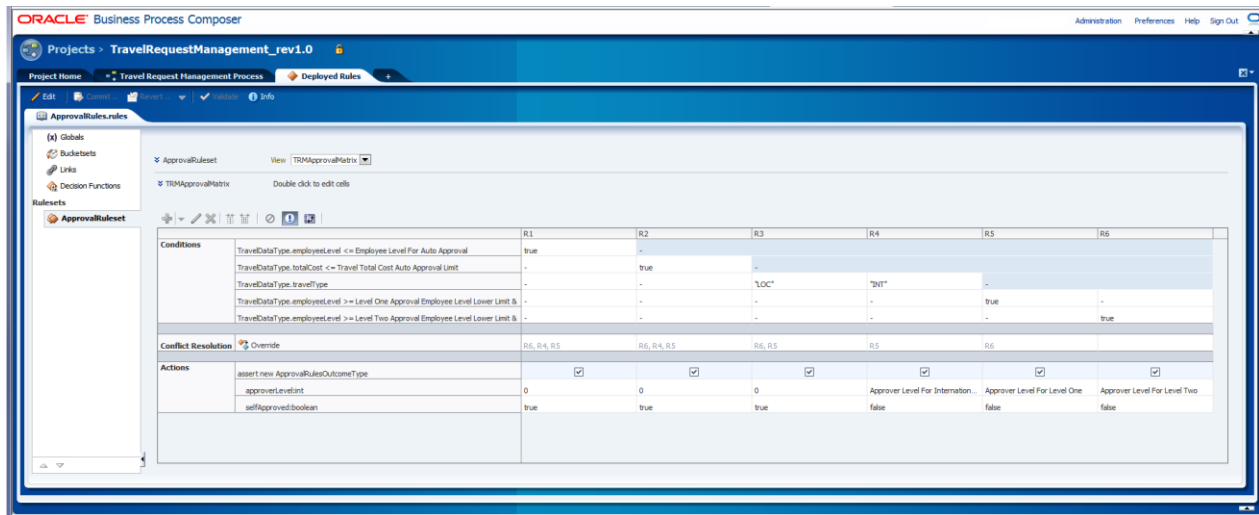


In the example above the list of participants is built based on the “Management Chain”. It can also be noted that the participant type is “Serial”, indicating that the approval will go up the management chain in serial order, rather than in parallel or only one level up (“Single”). “Rule-based” can also be selected if a business rule needs to be used to dynamically select the list of participants’ type, based on process information at task execution time.

This Oracle BPM pattern is essential: the alternative is to create logic to traverse organizational structures within the BPM flow, which then makes it more of a flowchart than a business process, and most importantly makes it inflexible.

BPMN Pattern: Business Fine-Tuning with Business Rules

Business rules should not be the exclusive domain of IT: they should be put in the hand of business to dynamically fine-tune the behavior of the process based on the changing business conditions. Oracle BPM Suite has adopted this principle and exposes business rules in BPM Composer, designed for use by business analysts. The screenshot below shows an example business rule, specifically of a decision table used in TRM to determine the level of approval required.



Complex business logic is well suited to be defined in a decision table like this, where the conditions in the matrix on top determine the actions that are going to be taken in the matrix on the bottom. For example looking at the column “R4”, if Travel Type is International, then the approver level is “Approver Level for International Travel”, which for this organization has been defined as Senior Vice President. This happens to be the policy that has been configured for the initial internal Oracle deployment of TRM, making Oracle itself the first customer to adopt TRM, a clear commitment to “walking the talk” of PAs.

A good pattern is to take decisions within rules and then using a gateway right after the rule to take different paths in the process depending on the decision. Analytics and rules expressed in business terms enable refinement of the business process by fine-tuning rules based on system behavior. In TRM the “Employee Level for Auto Approval”, “Level One Approval Employee Lower Limit”, and other levers can be adjusted to balance expense control and approval overhead, based on analytics measures such as expenses to date with forecasted travel cost by organizational group.

Name	Value
(x) Employee Level For Auto Approval	2
(x) Travel Total Cost Auto Approval Limit	200.00
(x) Approver Level For International Travel	3
(x) Level One Approval Employee Level Lower Limit	3
(x) Level One Approval Employee Level Upper Limit	5
(x) Level Two Approval Employee Level Lower Limit	6
(x) Level Two Approval Employee Level Upper Limit	7
(x) Approver Level For Level One	2
(x) Approver Level For Level Two	3

Rules can be changed in a Web browser while the process is running, and results monitored in real-time with Business Activity Monitoring (BAM), without involving IT or re-deploying the process.

BPMN Pattern: Start with Business Indicators, Get to Dashboards

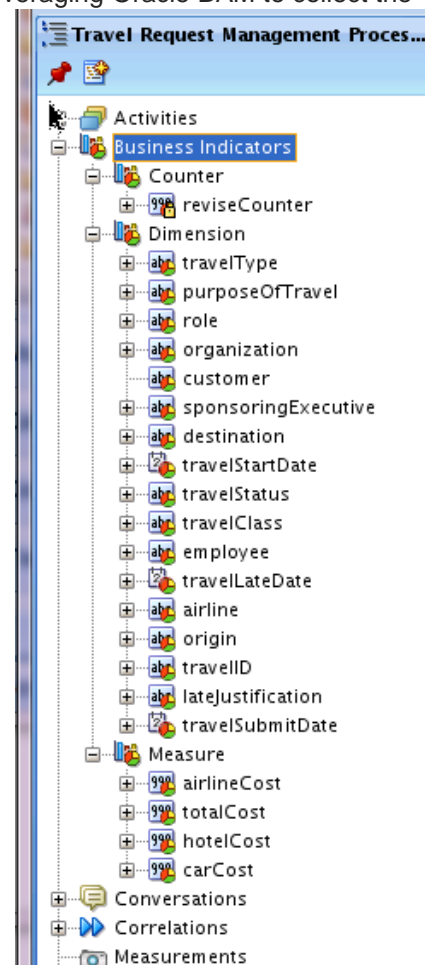
Dynamic refinement of the business process is driven by business decision makers that monitor dashboards. Too often dashboards are addressed in later iterations, rather than evolved with the task flow logic. Dashboards naturally evolve after deployment, when users see the first versions, understand how to better control the process, and define new requirements.

There can be a “chicken-and-the-egg” issue, whereas business, once a new interesting measurement is identified, wants to go back in time and look at historical data around that new measure, although data may not have been captured up to that point.

Process Accelerators address this issue by defining a set of Oracle BPM Business Indicators that goes beyond strict dashboard requirements, and by leveraging Oracle BAM to collect the data generated by the process. Once the business indicators are defined, BAM will start collecting data that will be surfaced as soon as dashboards are materialized. PA Dashboards can therefore be easily extended to use these business indicators using BAM’s point-and-click BAM dashboard definition functionality.

Business Indicators can be measures, dimensions, and counters.

- Measures are used for aggregation, for example in the TRM PA for controlling air, hotel, car, and total cost, aggregated by department, cost center, manager, etc.
- Dimensions are used for partitioning: by purpose of travel, dates, origins, destinations, employee, airline, etc.
- Counters keep track of the number an event occurs. In TRM a counter to track the number of revisions a Travel Request goes thru has been defined.



The Oracle BPM Suite documentation contains more information around business indicators and Measurements, another powerful feature that allows tracking process information within process portions.

One of the advantages of Oracle BPM Suite is that it includes BAM: the BPM process feeds BAM, BAM comes with predefined process dashboards and allows business analysts to quickly define new dashboards based on the set of business indicators that BPM feeds to BAM. A set of business indicators come out-of-the-box, and others are can be easily added, all of them automatically fed to BAM (you just need to “open the gate”).

Each PA adds a set of indicators and dashboards designed around the business processes in scope, based on the following principles:

- Identify KPI and Dashboard requirement earlier in life-cycle to correctly identify and capture required business indicators
- Identify key events and milestones at the modeling stage
- Use BAM to tap into external systems
- Use BAM Alerts to monitor faults
- Use analytical tools with best fit - BAM for operational monitoring and BI for historic and trend analysis
- Define role-specific dashboards, without trying to satisfy different users with same set of Dashboards: it's OK to have the same widget show up in multiple tabs, and apply filters appropriate for each business role.

Services Pattern: Define Business Service Contracts

BPM offerings need to embed technical instructions about what needs to happen in each node/task. When external services are called, technical details may make understanding what really happen in the service difficult. The developer may find it tempting to just point a BPMN task to existing technical services, even when these services may not be at the same level of abstraction as business tasks in BPMN. A best practice is to explicitly define new business

services, on top of existing technical services. This prevents exposing to BPMN the entire technical/data model, leaving to the BPMN task the onus of filtering out the right information. We should only provide what the process needs through explicitly defined services that become “Business Service Contracts”. The advantage of introducing this layer of service abstraction is not only clarity, but also flexibility: should the underlying model change (i.e., change attributes, or swap out the entire application that provides the services), the contract provides stability to the business process. Ideally, the service contract must be backward compatible. If backward compatibility can’t be achieved easily, different “versions” of the service can be created, keeping the old service contract in place.

In the Oracle TRM process Accelerator a separate business service layer has been created, called TRMModel. Creating a set of business services adds layers of abstraction that can become difficult to control, unless a fully fledged SOA tool is used to administer them. In the Oracle strategy, the BPM offering is built on a services-oriented architecture. This simplifies creation and usage of business services: point-and-clicks on BPMN task implementations translate into underlying bindings that associate a service with the task.

PA Best Practices: User Interface Component Architecture

Travel Request Management (TRM) helps organizations manage approval and tracking of travel requests. The central business object is the travel request that an employee submits for approval. The approval flow is determined on the fly based on various factors such as type of travel, the employee level, and travel purpose. The screenshot below shows the “Submit Travel Request” screen that the employee fills in to start the business process.

ORACLE Travel Request Management | Logged in as jcooper | Home | Help

Submit Travel Request | Submit | Save | Cancel

Request ID: TR000046 | Status: New | Sponsoring Exec: William Faulkner | Customer: Best Co | Reference ID: | Submitted By: jcooper | Traveler: James Cooper | Title: Loan Agent 1 | Email: jcooper@emailExample.com | Cost Center: | Organization: BPM | Org VP: |

* Employee ID: jcooper | * Travel Purpose: Customer Visit | * Travel Type: Domestic | * Origin: SFO | * Destination: EWR | * Travel Justification: Strategy meeting

* Trip Start Date: 7/31/2012 | Trip End Date: 7/2/2012 | Submit Date: 7/16/2012 | Late Justification: ..:

Cost Estimates (USD)

Airline	594
Hotel	280
Car	106
Total	980

Error
Trip End Date before today's date
Example: 10/11/2012

Submit one trip segment for a single round trip. Submit multiple segments for multiple destinations.

Segment	Origin	Destination	Start Date	End Date	Total Cost Estimate
1	SFO	EWR	7/31/2012	8/2/2012	980

Figure 1: The Submit Travel Request Screen

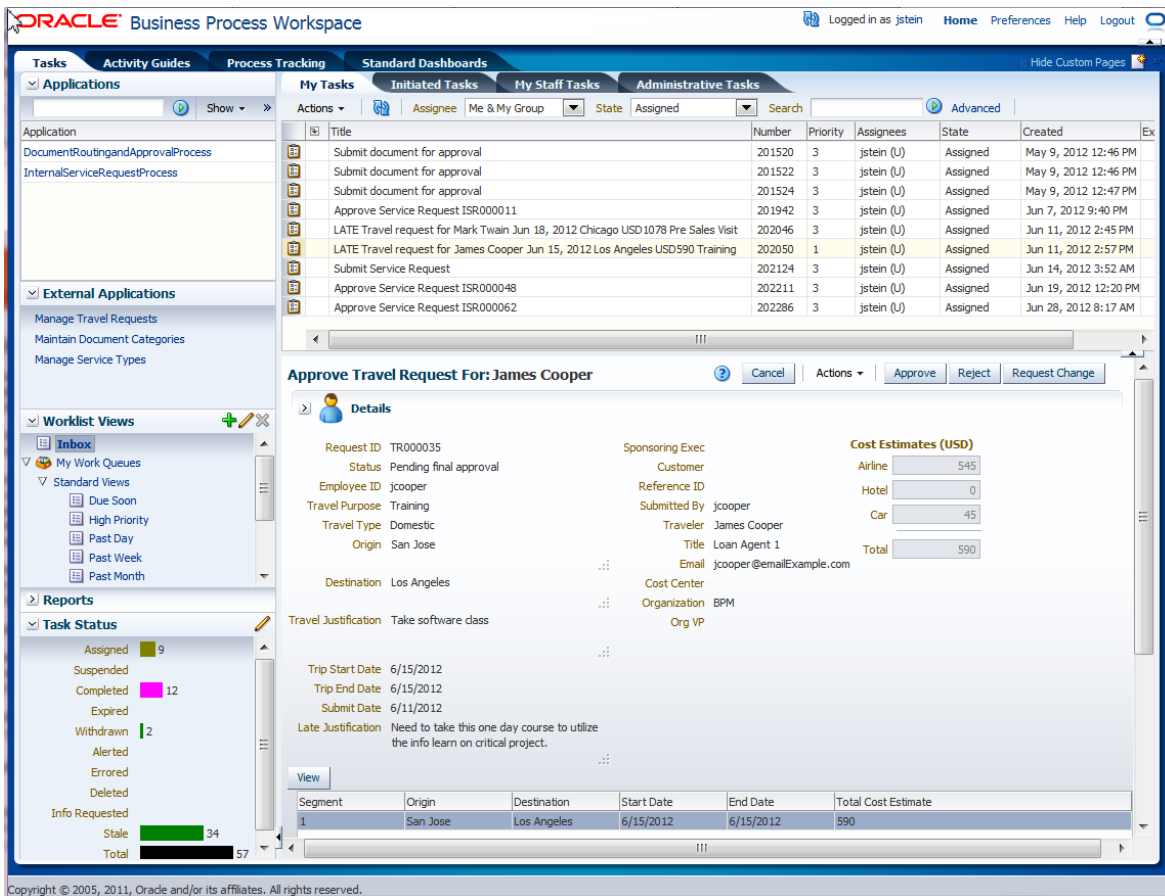
Looking at the screen above, the different components of the Travel Request can be identified, such as its header (the area containing the general travel request information on the top), and the trip segments (the table on the bottom of the screen). Also shown above is field validation in action.

Travel Request is a business object of moderate complexity. However, even a deceptively simple supporting user interface (UI) like the above can be used to show how the UI in a TRM app can be implemented according to best practices that extend BPM flexibility to the UI layers. This influences the level of functionality, flexibility, and TCO.

Let's now explore some of the best practices and patterns that hide behind this apparently simple UI. We'll inspect them according to the Model-View-Controller (M-V-C) model, traversing them in V-C-M order, from the most visible business to the core backend aspects.

View Pattern: Plug into OOB Oracle BPM Human Tasks / Workflow

A very basic yet essential best practice is to use what's available in the out-of-the-box BPM product. The Oracle Business Process Workspace is used here as the access point to three PAs, TRM, Document Routing and Approval (DRA), and Internal Service Request (ISR). Why rebuilding when the Workspace provides access to all tasks by role?

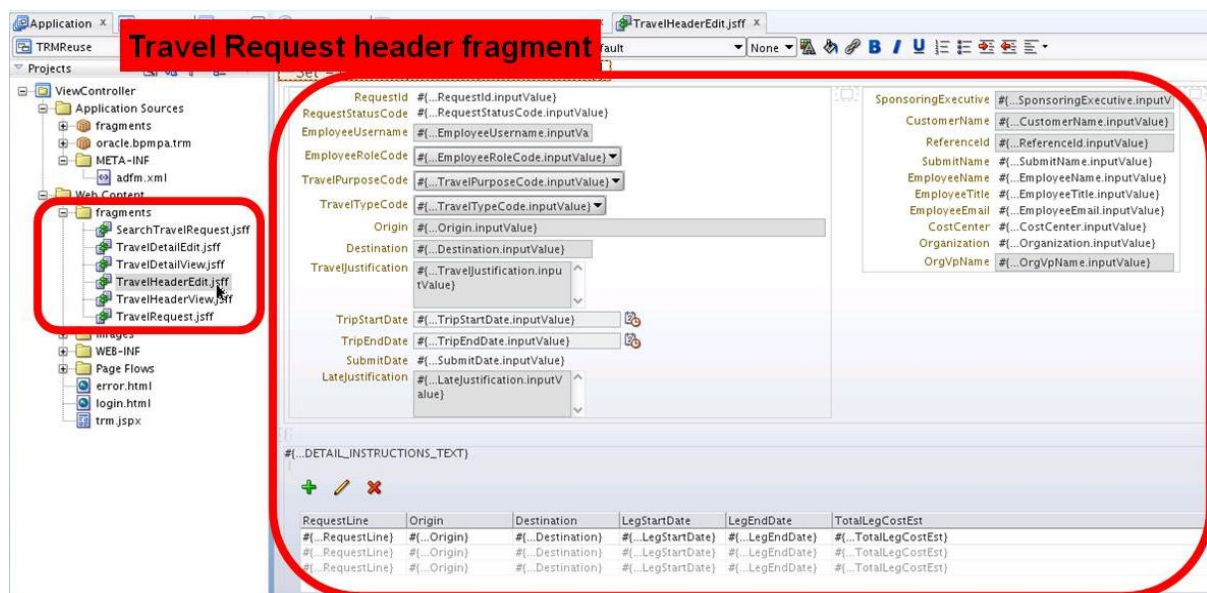


Copyright © 2005, 2011, Oracle and/or its affiliates. All rights reserved.

Any additional application exposed to the user is added to applications areas, where the user can initiate new requests that will spin off business process instances in the BPM system, in the case of PAs new BPMN process instances.

View Pattern: Reuse Page Fragments

Page Fragments are view components defined once and reused wherever needed in other pages. Defining and using fragments favors UI reuse, saving implementation cost while making the UI flexible and maintainable. Looking back at “*Figure 1: The Submit Travel Request Screen*” on page 21, a travel request “header” is immediately identifiable as a candidate to become a page fragment, since all UI screens that display travel requests will include the header fields. The picture below shows page fragments in the Oracle Application Development Framework (ADF).

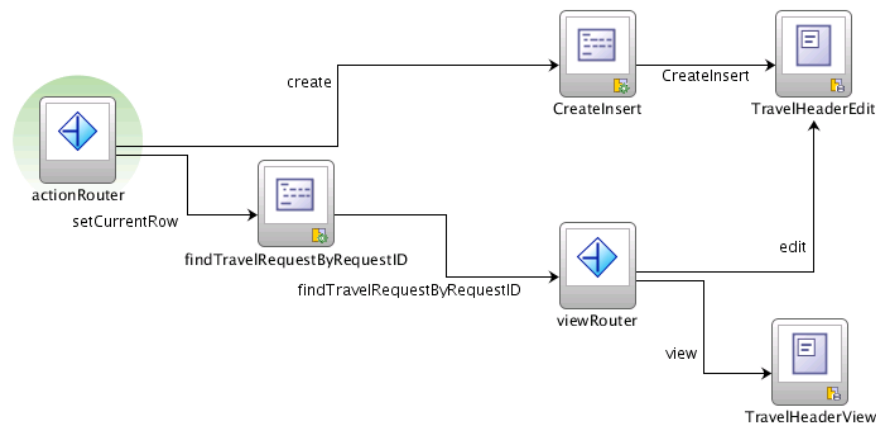


You may wonder whether a UI framework like ADF, enabling quick creation and drag-and-drop reuse of page fragments, is a must, or even belongs with BPM. In our view UI design is critical to BPM success, since it's the most prominent aspect that process participants see and work with. There is arguably no more appropriate use of powerful UI frameworks than in BPM solutions, especially human-centered ones, where the same business object is created, escalated, approved, rejected, expanded, completed, recycled, and where different business users have different authority and access at different point in times: a lot of screens, but a lot of commonality across them.

Controller Pattern: Reuse Page Flows

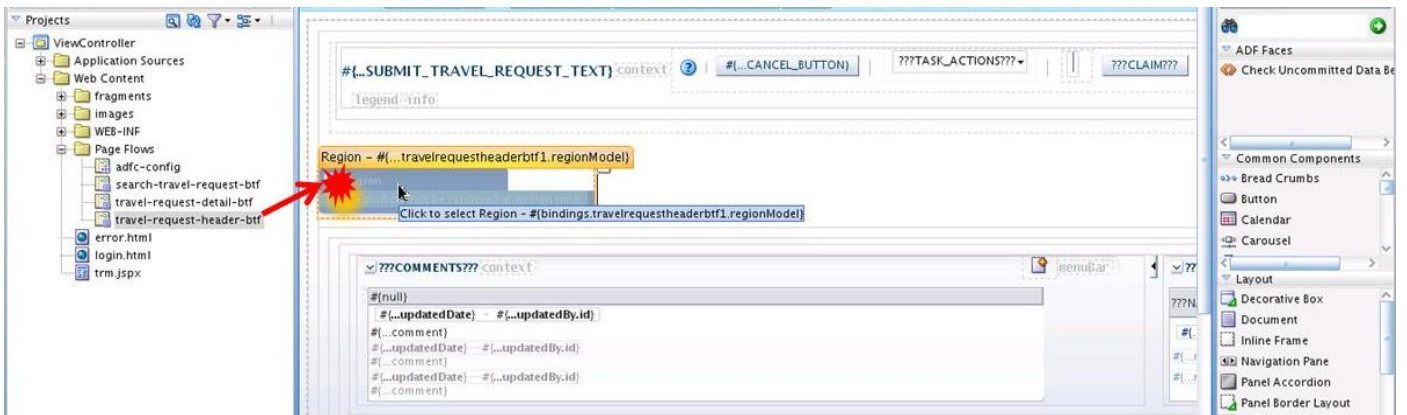
Reusing UI components like fragments is important, but not enough. Web applications usually have more than one page, with navigation from one page to the next determined on the fly. For example, you may press a Submit button on travel request page to create a new travel request, or you may click on an existing travel request and access a view-only screen with details.

While all this navigation logic can be coded, and it may seem easy to just do so, that's not best practice. IT architects know that the Controller portion in the M-V-C model has been created mainly to avoid navigation logic proliferation. When developing BPM solutions, IT needs tools that make "doing the right thing" easy. The screenshot below shows the Oracle answer to this issue: Task Flows.



Oracle ADF uses Task Flows to graphically design the UI flow: you can drag-and-drop pages into the Task Flow, then connect them with arrows (control flows). Underlying, ADF generates all the necessary code so you don't need to write it.

Once the task flow is defined you can drop it into a UI page. The picture below shows how the travel request header task flow can be dropped in the Submit Travel Request page, which will then inherit all the travel request header navigation. ADF takes care of generating all the necessary bindings on the fly.

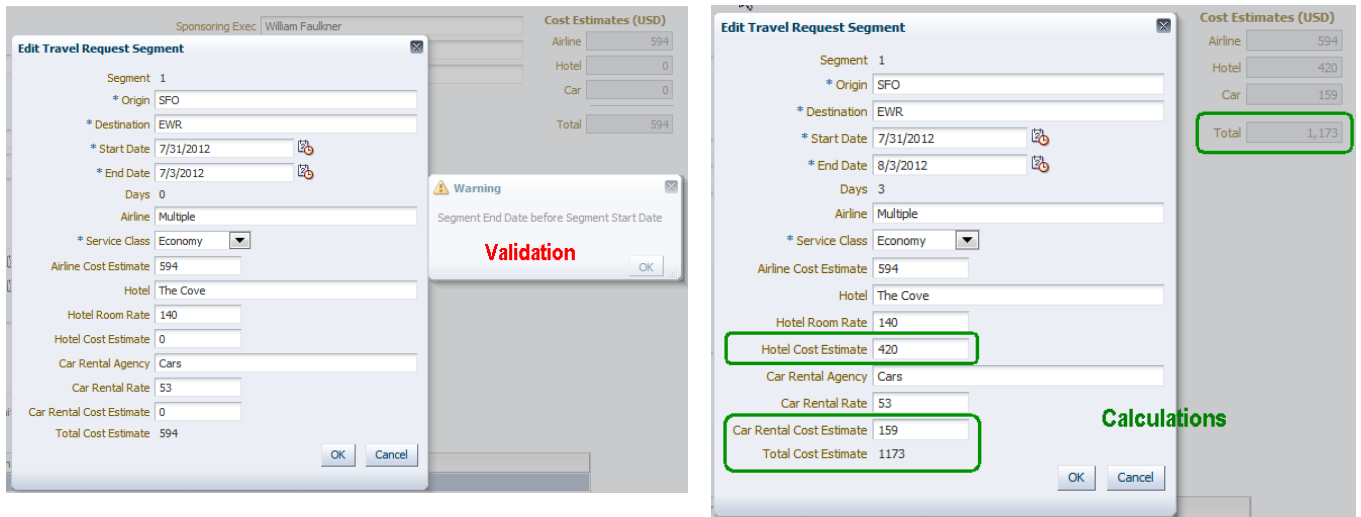


The screenshot also shows on the right the traditional UI components, such as button, panels, etc., that can also be added to the page by drag-and-drop. TRM already includes a fully working UI, but you can easily extend TRM or other PAs using ADF. Business and IT can iteratively add functionality to satisfy additional business requirements. Many configurations are also made directly available to business users via Oracle BPM Composer.

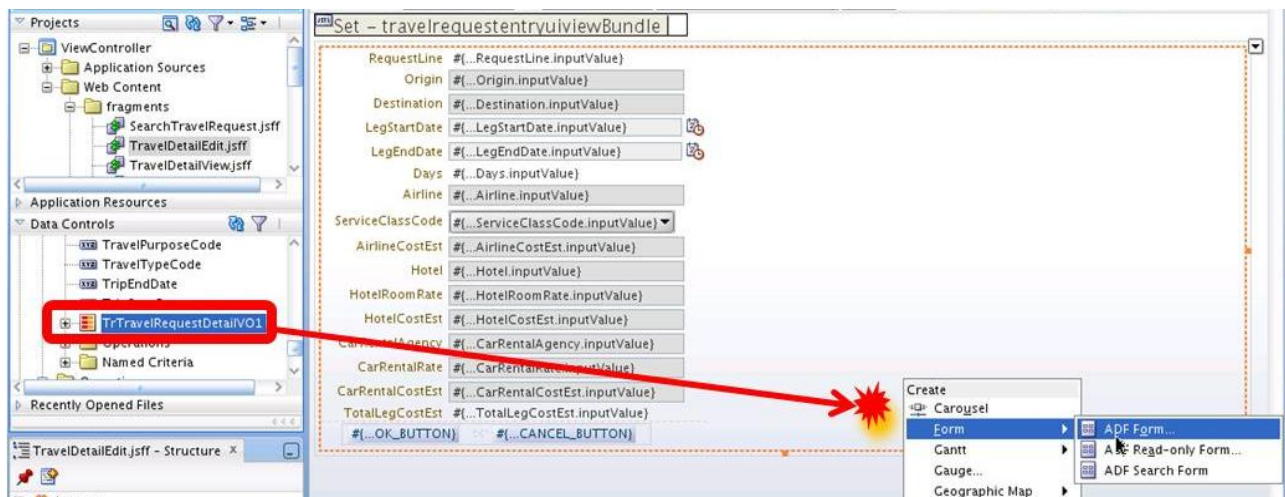
Model Pattern: Place Business Logic in the Appropriate Layer

The next two screenshots highlight calculations and validations in “New Travel Request Detail”, a popup used to collect travel leg information in TRM. As the user enters data in the UI, validation and calculation are performed, for example the end date is checked to be not before the start date, and the hotel cost estimate calculated based on the number of days and the daily rate. Simple stuff, but that if not properly structured can lead to low reuse, high testing cost, and lack of flexibility. In this case having core business object validation within the UI would force duplication of logic, for example between the Cost Estimates in the Submit screen and the calculation in the popup.

In this scenario, keeping business object logic together with business validation makes a lot of sense: when the business object changes then the validation has to be changed in only one place, not separately changed in every place at higher layers.



Oracle BPM Suite delivers ADF-Business Component (ADF-BC), a tool that allow you to add business logic and validation at the “Model” level, for example at the DB Schema level. ADF-BC allows creating Data Controls just by pointing to the model / business object definition. These controls can then be added into UI pages by drag-and-drop. Such a simple action will not only create all fields in the UI corresponding to the business object attributes, but also carry all validations associated with those attributes. The screenshot below shows how a data control that represents the Travel Request details is dropped onto the UI template of the page fragment reused in the “New Travel Request Detail” popup. The result (a new ADF Form is created) is already shown (including fields and the OK and Cancel buttons).

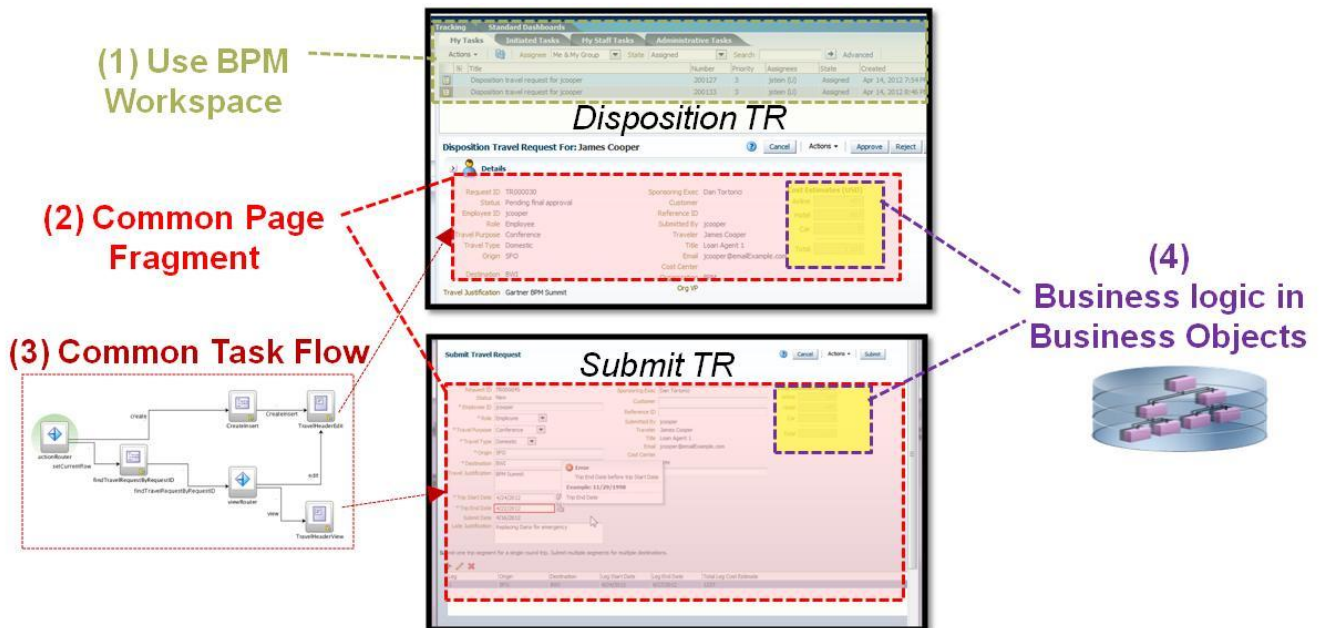


The page fragment carries all validation and calculations, and can be embedded in all screens that deal with Travel Request details. You just decreased the complexity of UI implementation by a factor of x when x screens are using this functionality, multiplied by the number of times the definition of the business object is going to be changed in the future based on new, improved, and updated business requirements. Underlying bindings between UI and business logic are established for you and saved in XML files that hard-core developers can view at their leisure.

The decision where business validations should be placed depends on many factors, such as reuse, proximity, and performance. Product capability like ADF-BC helps making the right choice easy. After you have defined the Travel request as a business object in ADF-BC by just pointing at the underlying object (ex.: database table), you can define Business Rules for each attribute. The advantage in this scenario is that all linked UIs will inherit the business object validation. At a later time you may want to add more validations, which you can just add in the business object. PAs provide many concrete examples of how you can extend business objects based on your specific business requirements.

Putting it all together: TRM UI under the PA Reuse Pattern lens

It's time to look back at our TRM UI and summarize the reuse best practices / patterns that are "under the hood". The picture below represents two TRM screens, "Submit Travel Request" that the employee uses to inform of an upcoming business trip and request approval, and "Disposition Travel Request" that the manager uses to approve, reject, or request more info. The Disposition screen shows most of the fields of the submit screen, but view-only. To complete the Disposition screen, only some buttons to Approve, Reject, or Request Change need to be added. The two screens are very similar because they are separate steps that work on the same business object, Travel Request: a very common pattern in BPM.



Implementation of Disposition screen without patterns would end up creating a separate component, therefore at best with cut-and-paste code reuse. Issues would pile up especially when business requirements change, with changes that need to be applied in many places. Instead, PAs uses a model that leverages, among others, the following patterns:

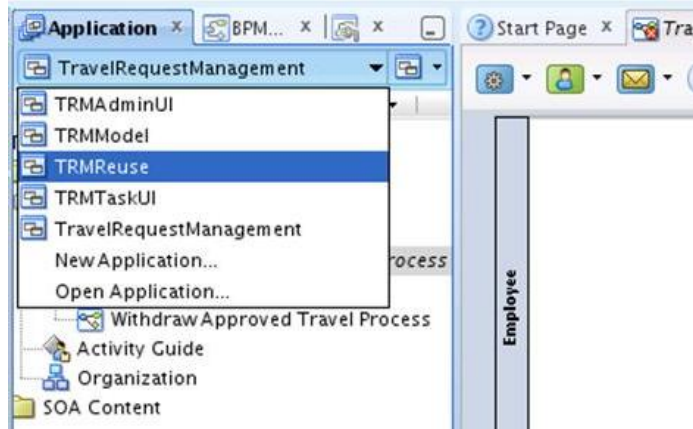
- 1) Both screens can be accessed via the BPM Workspace, based on the business role
- 2) Common page fragments are reused (see area highlighted in red)
- 3) A common task flow allows reusing the navigation logic in different pages/regions (example: in a page to view and in a page to edit)
- 4) Validations and calculations are defined only once, in the underlying ADF-BC objects (see yellow highlights)

Patterns like the above help align IT and Business around a model that delivers value quickly, yielding flexible solutions built on top of Oracle BPM components.

Pattern: Application Component Modularization

When you apply best practices, components start proliferating. That's what you want: build component libraries and frameworks. It then becomes important to partition UI functionality into

reusable components. That's the reason why in TRM, we have isolated reusable components in a separate application that can be easily shared across the BPM Suite (TRMReuse).



The screenshot above also shows other application partitioning examples in TRM: business-facing UI grouped by business role groups, for example for Administrators (TRMAdminUI) and users performing tasks (TRMTaskUI). Partitioning functionality in clearly defined and consistently used modules helps making the right choices when it comes to reuse vs. rebuild. There are plenty of other best practices in SOA service partitioning which apply to the BPM development environment at large.

“Process Accelerators represent a good place to start in BPM, and for those with experience, a great way to reset and learn about the evolving architecture of process automation. The Process Accelerators best practices described in this paper will help process improvement teams drive value more rapidly. Accelerators also provide an opportunity for teams to learn how to evolve processes more easily while promoting greater leverage, reusability and extensibility.”

Maureen Fleming, IDC

Conclusion

The difference between organizations that consistently deliver successful process solutions at low cost and those that obtain low ROI from BPM lies in their IT-Business delivery model. Oracle Process Accelerators improve the learning curve encountered by both IT and Business teams, serving as a launching pad for executing against critical business processes, developed with both business and extensibility best practices in mind. Process Accelerators embed best practices that help business generate value and IT deliver at low cost with reusable components. Bottom line, customers adopting Oracle Process Accelerators best practices and frameworks can reach process excellence faster.



Oracle Process Accelerators
July 2012

Author: Cesare Rotundo
Contributing Authors: Chuck Jones, Vinay
Pamadi, Kartheek Lenkala, Ralf Müller,
Bhagat Nainani, Manoj Das

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com



Copyright © 2012, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0112

Hardware and Software, Engineered to Work Together