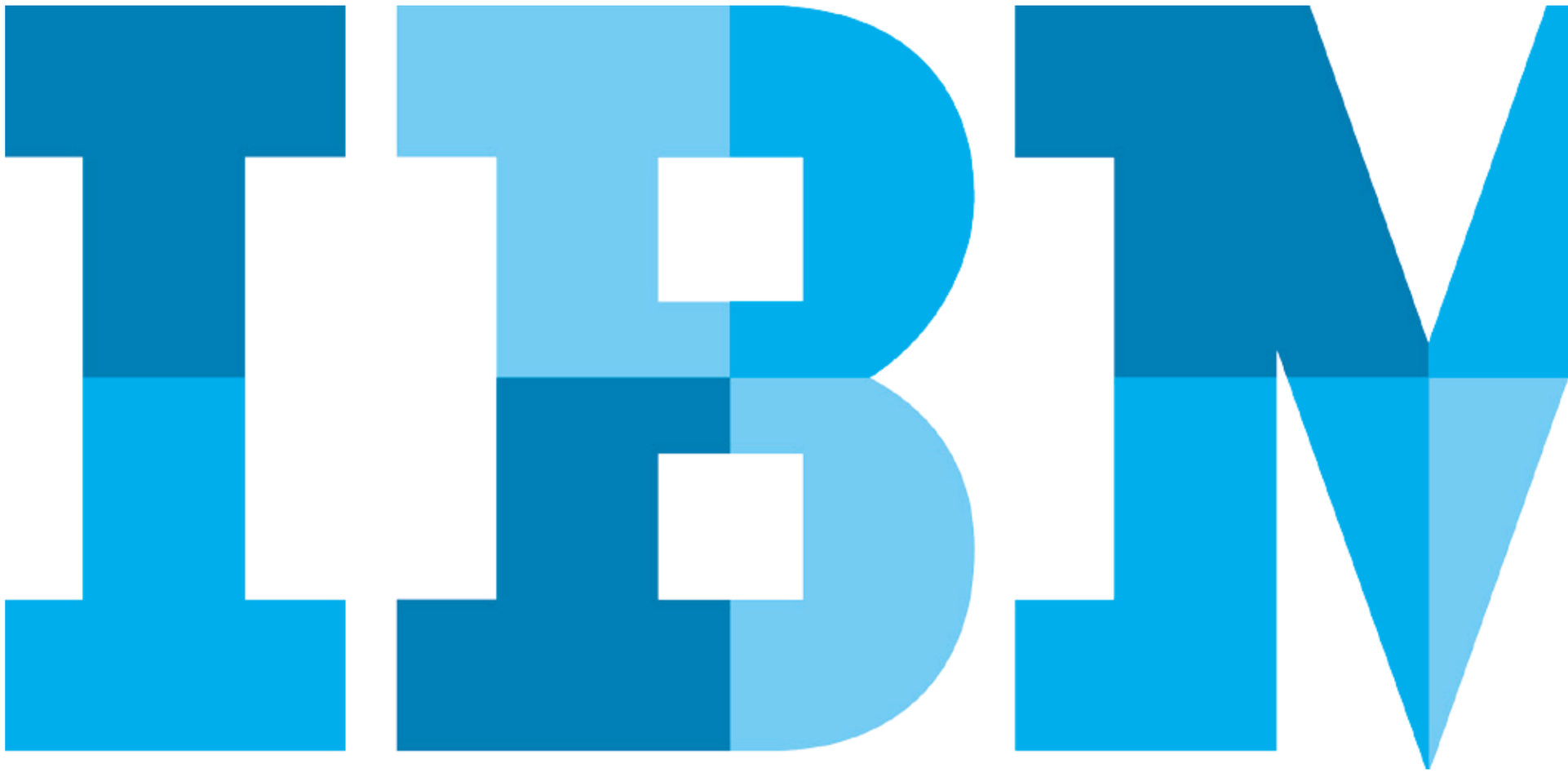
IBM Analytics

IBM Banking Process & Service Models

Support for BIAN



IBM

Introduction

Banks typically have complex IT systems compared to other types of organizations. Part of the reason is that products and services that are designed, developed, offered, and serviced by a bank are fully manifested as IT constructs.

Traditionally, banks have used an application-centric approach to develop IT solutions to business problems. This has resulted in a large number of IT applications, each supporting some subset of business functions, sometimes overlapping. In reality, business processes of the bank cut across these applications, necessitating highly complex integration requirements.

To complicate things further, business requirements constantly change, resulting in new applications being developed to meet these requirements. These new applications are then integrated with existing applications, adding to the complexity.

The bottom line: a bank is left with IT systems that are inflexible, inefficient, and out-of-alignment with business needs. This is the context for IT transformation in banking. The goals of the transformation initiatives are to reduce complexity, improve agility, standardize, and achieve business-IT alignment.

From a business perspective, some of the key factors that drive the transformation initiatives include the need to comply with new and increasing regulations, the need to become a customer-responsive enterprise, the need to support omni-channel customer interaction, and the need to operate in a boundary-less economy.

A perfect storm scenario emerges when you couple the business drivers with the technology drivers. While the business drivers provide the push for sustained competitiveness, technology advances provide the pull. These technology advances include component-based modeling (CBM), service-oriented architecture (SOA), model driven development (MDD), standardization, business process and business rule management systems (BPM/BRMS), mobile and social computing platforms, analytics, cloud and big data. In addition, availability of pre-defined service content and structure, like IBM® Banking Process & Service Models (BPS), help jump-start SOA value ‘out of the box’.

While there are plenty of business and IT drivers for transformation, there are significant challenges such as dealing with legacy IT applications in the enterprise. For most banks, a “rip and replace” approach is a non-starter. The only feasible approach is to adopt an incremental, progressive approach to transformation. This implies that the bank needs to manage a hybrid environment of new SOA components co-existing with legacy applications, which poses many integration challenges.

Bottom-up and top-down approaches to transformation are used to deal with these challenges.

Transformation

A bottom up transformation approach focuses on the current set of applications in the enterprise and uses SOA to identify how these applications can be incrementally componentized.

The advantages of bottom-up approach are:

- You can continue to use the legacy apps
- It is less risky
- It often provides faster time to market
- It is less expensive

The disadvantages are:

- It might prevent true reengineering
- Tie the new process too closely to the limitations of the current environment

In contrast, a top down approach starts with an SOA-based solution design to the business problem, develop components that are driven by this design, and integrates these components with legacy applications as appropriate.

The key advantage of this approach is that it provides the ability to design and implement business processes without the constraints imposed by the legacy applications. But it does require the design and development of the new service components.

In both of these scenarios, we are looking to ensure that the modeling process retains the component and service-based properties of separation of concerns, loose coupling and encapsulation linked to a component-based banking business model.

Enterprise Architecture

discipline defines and maintains the architecture models, governance and transition initiatives needed to effectively co-ordinate semi-autonomous groups towards common business and IT goals. Enterprise architecture helps to link an enterprise's business strategy to its change programs through the definition of:

- Architecture models to capture the business' intended structure (through a business architecture) and to provide a clear specification of how multiple projects and programs must exploit information technology (through common, and explicit, IS and IT architectures)
- Mechanisms, such as architecture governance and transition planning, to help plan, coordinate, and control all parts of the business, ensuring they all pull in the same direction

Listed below are the components of Enterprise Architecture:

- Business architecture
- Enterprise data, message, process, and service models
- SDLC methodologies and tools
- SW infrastructure standards
- HW infrastructure standards
- Governance

The approach that is discussed in this paper is a specific instance of enterprise architecture for banking, with BIAN Service Landscape as the business architecture and course grain component services and BPS as the fine grain enterprise service, message, and process models.

BIAN

BIAN (Banking Industry Architecture Network) is a standards development organization addressing Service Oriented Architecture (SOA) in financial services. BIAN is a global, open, independent and unique community where banks, software providers and system integrators are collaborating to define a common SOA framework for the banking industry.

The BIAN Service Landscape is a reference framework that contains all identified BIAN Service Domains. Its purpose is to provide a mechanism for quickly identifying and selecting non-redundant Service Domains based on business events. Different criteria can be used to classify and organize Service Domains that would result in different layouts of the standard set of BIAN Service Domains. BIAN uses a primary Service Landscape view based on agreed categorizations that have been in use by the BIAN membership.

A Service Domain defines a unique and discrete business capability. The Service Domains are the 'elemental /canonical building blocks' of a service landscape. Any business activity or event can be represented by a suitable collection of one or more Service Domains working together in collaboration. There are about 300 identified BIAN Service Domains of which about 200 can be identified as Core Banking functionality while the rest can be considered as Business Support Functions.

IBM Banking Process & Service Models (BPS)

BPS is a pre-defined content and structure rich set of models that are designed specifically for financial institutions. It is regularly enhanced and extended to align with the global requirements for risk and compliance and optimally allow for the development of more efficient straight through processing solutions. By using BPS, the approach to transformation can be dramatically shortened, with consequent savings in time and money for the organization.

BPS is a result of leading-edge practices and engagements and have been validated through their use within many of the world's major financial services organizations. It plays a critical role in the definition of service-based architecture. It is only through analysis of the processes that support the operations of a financial institution that the service candidates that will best support those processes can be identified. Process analysis also provides essential information about the context of those services, capturing requirements governing the applications that call services within the architecture, and the human roles in the organization that interact with those applications.
BPS has in excess of 400 business processes that can support and fast track organizations in their process reengineering. It also identifies candidate business services that support the business processes.

These reusable elements are analyzed further as Service Capabilities. The Service Capability definitions isolate the proposed service definitions and allow the capturing of additional requirements such as the type definitions that support the inputs and outputs of the service, the interactions between services and the delegation patterns that ultimately support the business of the financial institution. The requirements that are captured within BPS can be used to construct a detailed design model of the actual services that are required to support the project. These are component-based models that catalog services, and the interaction between services in the form of collaborations. This paper shows that the relatively new BIAN-defined Service Domains can be mapped to and implemented by using the long established and widely adopted BPS models.

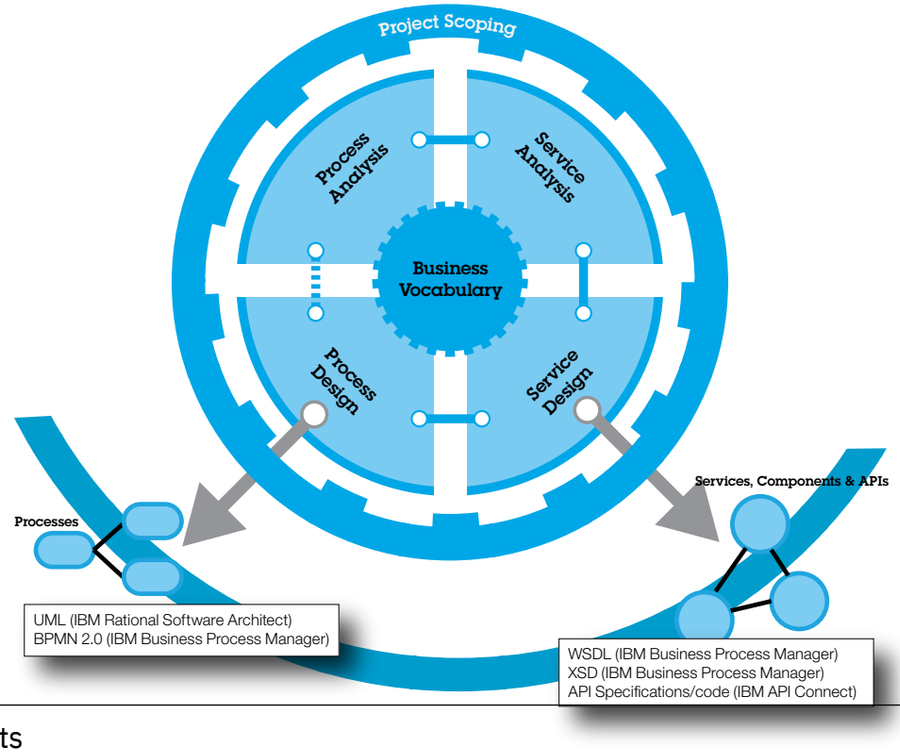


Figure 1.BPS Components

BPS Components

Process Models

Analysis Process Model
The Analysis Process Model provides the business specification of the processes, which is used to underpin the initial analysis activities in determining the optimum subset of processes for a project from a business perspective.

Orchestration Process Model
Provides the design level technical specification of the processes, which forms the basis for any downstream implementation of executable processes.

Service Models

Service Analysis Models

Business Object Model (BOM)
The Business Object Model provides analysis-level technology independent class models that enable traceability between automated business process requirements and downstream SOA IT analysis representation.

Service Design Models

Interface Design Model
The Interface Design Model (IDM) provides a design for the development of components, types, interfaces, and data transfer objects for an enterprise-wide business services-based architecture.

Web Service Design Model
The Web Service Design Model (WSDM) provides a design for the development of participants, service interfaces, and messages appropriate for an enterprise-wide business services-based architecture.

Transfer Object Model
The Transfer Object Model (TOM) defines Transfer Objects, which are simple UML data types that are derived from the IDM class model. Transfer Objects are used to explicitly define, at modeling time, the structure of data that will be used in:

- Deployed services
- Data objects in executable process models

Item Definitions in the Orchestration Process Models in RSA are linked to Transfer Objects to define their structure. TOM provides a clean separation between the IDM class model and the Transfer Objects. This separation enables better management of Transfer Objects, such as:

- Governance over multiple versions of Transfer Objects
- Governance over multiple Transfer Object hierarchies intended for different target systems
- Control over who can modify Transfer Objects

Design Model Datatypes
The Design Models Datatypes (DMD) provides a mechanism for defining model datatypes and enumerations that are used by the Service Models.

BIAN Service Landscape and Alignment

The BIAN Service Landscape is a reference framework containing BIAN Service Domains, each of which describes a discrete business capability, which is organized for ease of access.

The BIAN Service Landscape has three levels of classification for the business capabilities it describes:

- Business Areas – Group together, at a high level, a broad set of capabilities.
- Business Domains – Collections of capabilities within the Business Area.
- Service Domains – Unique and discrete business capabilities.

Each BIAN Service Domain represents a generic business capability for which BIAN seeks to define the standard canonical semantic services for the banking industry.

For this paper, we will look at the Service Domains within the Service Landscape, and how they relate to the BPS.

BIAN and its members believe that SOA is the best technology for internal and external interfaces to produce consistent definitions, levels of detail and boundaries through collaboration.

In recent releases of BPS, a pivotal modeling activity has been the creation of a BIAN-BPS Alignment. This alignment reflects the linkage between BIAN components and BPS model elements.

There are two separate alignment artifacts provided with BPS, showing alignment of the BIAN Service Landscape with the BPS Analysis Process Model (APM), and with the BPS Business Object Model (BOM).

A pivotal modeling activity in the BIAN-BPS framework was the creation of a BIAN-BPS Alignment Model. This alignment model reflects the linkage between BIAN component business scenarios and APM.

This was an essential first step that is highlighted here because it exemplifies the challenges of what the nature of the synergy might be and to what extent it will become over time. In both the top-down and bottom-up scenarios, the bidirectional pathways that are relied on this semantic and structural cross road to identify existing assets as candidate services and specify the creation or modification of candidate services.

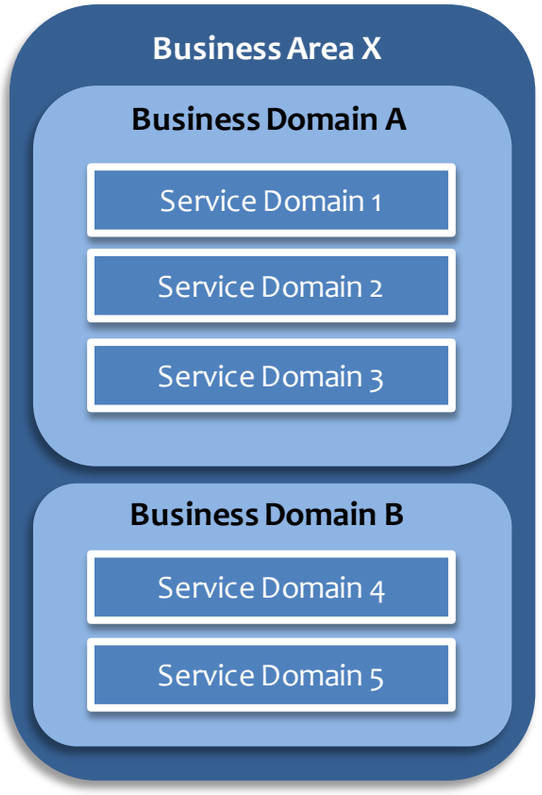


Figure 2.BIAN Service Landscape classifications

BPS Support for BIAN

BIAN was created to establish, promote and provide a common framework for banking interoperability issues and to become and to be recognized as a world-class reference point for interoperability in the banking industry.

BPS supports financial institutions in achieving their strategic objectives by the adopting and using the best practices and open standards advocated by IBM's vision of a service-oriented architecture. This allows financial institutions to focus on business process improvements through the utilization of our 'leading practice' templates that fast track the development of 'to-be' business scenarios.

Business process reengineering also facilitates the better alignment of business to IT and the optimization of straight through processing solutions, where appropriate, with resultant efficiencies and cost reductions. This is achieved through the definition of services that are independent of line of business and product and can be reused across the enterprise.

BPS supports fast tracking institutions in their BIAN transformation strategies by offering organizations the means to reengineer their solutions and put business drivers at the heart of their strategic objectives.

BPS contains mapping coverage for a subset of the BIAN Service Landscape, specifically the current banking-specific Service Domains.

BIAN to BPS Mappings

The BIAN to BPS Mappings are available for entitled customers. The mappings are provided as a spreadsheet containing:

- Mappings between BIAN Service Domains and APM Processes and Global Tasks
- Mappings between BIAN Service Domains and BOM Service Capability Operations

APM and BIAN Service Landscape

For a given BIAN Service Domain, a mapping to the APM provides the processes and tasks that describe the high-level process behavior that is associated with a particular Service Domain. The activity flow (Process) models include the activities that are performed when the service is executed as well as the control flow and data flow defined over the set of activities.

BOM and BIAN Service Landscape

The BOM describes the set of service signatures, in the form of Capability Operations that are at the same level of detail as the Service Operations defined under the BIAN Service Domains.

The BOM also provides detailed structural definition in the form of class diagrams for data objects that are identified in the BIAN Service Landscape and modeled in the BOM.

BIAN Service Operations help define the boundary of business components in the business architecture whereas the IDM services define the interfaces of IT components in an SOA solution. IDM provides a sequence diagram for composite services helping to identify the constituent atomic services and the order in which they take place.

Benefits of the BIAN-BPS mappings

We anticipate the following business and IT benefits and contributions by using the BIAN-BPS mappings that are discussed in this paper:

- Consistency of business terminology and definitions – Results in closer alignment between business and IT analysis and design project deliverables
- Predictability to ensure success – Results in organizations creating more predictable and innovative business models
- Quality as a foundation of trust – Results in more integrated results where data, process and IT services are aligned
- Audit and governance as a foundation for business control – Results in consistent project roll-out across enterprise
- Modularity resulting in flexibility – Results in adoption of a service-oriented architecture approach
- Fostering collaboration – Results in organizations fosters collaboration across lines of business and defining a common catalog of processes and services
- Going beyond silos – Results in an enterprise view of data, process and service analysis and design artifacts with an underlying agreement on data definitions such as ‘customer’, ‘product’ etc.
- Moving forward incrementally – Results in incremental development and structured project roll-out
- Definition of correct service granularity – Results in definition of correct service level granularity that is more closely aligned to levels of granularity in BPM.

Methodology

The BIAN/BPS methodology provides an end-to-end set of steps which articulate the end-to-end methodology for solution design and implementation using the BIAN Service Landscape and BPS.

Model Scoping

BIAN Service Landscape artifacts are used to retrieve a subset of process, business object, and service models from BPS that are relevant for the selected business scenarios.

- Identify the associated BIAN Business Scenario matched to the business requirements. BIAN Business Scenarios model an informal flow of business interactions between Service Domains.
- Use the BIAN-BPS Alignment mapping to identify relevant BPS processes and/or services.

Example: Using the BIAN Business Scenario, “New Customer Setup”, and the BIAN-BPS Mappings, circa 70 APM Processes and 80 BOM Service Capabilities are in scope for this Scenario, based on the Service Domains used by that BIAN Business Scenario.

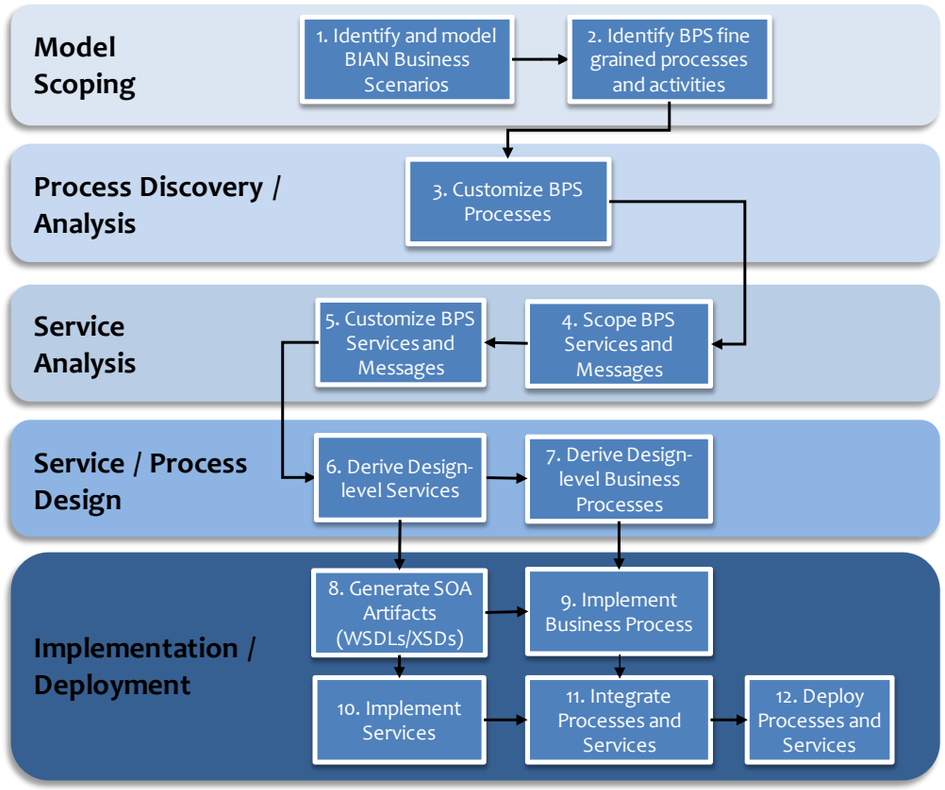


Figure 3.BIAN - BPS Methodology

Process Discovery / Analysis

Business processes are formalized and adapted and shared across the community using cloud-based tools, primarily to communicate and gain consensus on the scope.

- With the activity boundaries identified, identify fine grain business processes and tasks, based on BIAN and BPS domain knowledge.
- Review and validate completeness of process and activity descriptions, information/data attributes, control flows, decision points, and roles.
- Perform any customizations as needed. Reuse as many of the existing assets as possible.

Example: From the 70 Processes identified in the Model Scoping phase, only a subset are relevant for New Customer Setup, such as “Administer Customer Details” and “Identify US Account”. A detailed review of the relevant Processes is done during this phase; then identification and addition of customizations is performed.

Service Analysis & Design

Business processes are refined as model driven source code is engineered as service and solutions leading to software architecture delivery.

- Scope out project level Business Object Model (BOM)/Interface Design Model (IDM)/Web Service Design Model (WSDM) elements from the conceptual level analysis models.
- Validate the scoped models against the business requirements that are identified in the process discovery.
- Identify business capabilities in the BOM and associated service operations (create new ones if necessary).
- If required, identify a collaboration corresponding to the operation in the IDM model to build a composite service.
- Link the collaboration diagram to the service operation through a realization of UML.
- Using the BOM as the analytical object model, identify the control object, and other business objects and validate its attributes.
- Define lifecycle model of the control object using the BPS fine grain process models.

Example: The “Administer Customer Details” Process, identified in the previous phase, contains a task “Retrieve Full Customer Details”, which has an associated BOM Capability Operation of the same name, and that Capability Operation has an associated design-level WSDM Service Operation, “retrieveFullCustomerDetails”. Analysis and updates to those model elements are performed in this phase of the project.

Service Implementation

Analyzed and designed service domains and service operations are realized in a runtime environment through an enterprise service bus.

- Transform UML model, developed in the Service Analysis and Design phase, to SOA runtime artifacts. These artifacts include WSDL, XSD definitions of the services, as well as BPEL flows for straight through processing, BPMN for process implementation, skeleton Java code. Bring in the generated artifacts into IBM Integration Designer for service realization.
- Identify existing application capabilities and service touch points. Expose the application functionality as APIs in service bus. (Various technologies exist, depending on the nature of the application, for exposing application functionality as services such as enterprise application adapters, POJO invocation, web services, message queues, CICS, IMS, etc.)

- Develop mediation flows and service/message maps from the canonical model to existing applications through the exposed application API.
- Develop composite flows as BPEL micro flows or service orchestration in enterprise bus.
- Bind service invocations to application end points. Use a UDDI service registry as necessary for service lookup and dynamic service invocation. Unit test service implementations.

Example: The designed Services from previous phases, for example, retrieveFullCustomerDetails are transformed as part of the Process export step, described below, and them implemented in an implementation environment such as IBM Integration Designer.

Process Implementation

Business processes are refined for implementation details such as service invocation targets, user interfaces and human workflow.

- Export the designed business processes from IBM Rational® Software Architect and import them into IBM BPM Process Designer. Bring in service definitions (WSDLs) and message model (XSDs) from Service Realization step into Process Designer.
- Adorn processes with implementation details such as service invocations, UI screen flows (coaches), human workflow.
- Develop UI screens and screen flows using the message model brought from Service Realization step. Define attribute bindings to screen widgets.
- Wire service tasks to business services developed in enterprise service bus. Develop simple message maps from the UI objects to service messages only if needed. Bind the invocations to service end points.
- Simulate and test the business processes.

Example: The design processes, for example, Administer Customer Details, are exported from the design environment to an implementation environment, such as IBM Process Designer and implemented, tested, and deployed.

IBM Corporation
IBM Analytics
Route 100
Somers, NY 10589

Produced in the United States of America
December 2018

Please Recycle

IMW14891-USEN-01