Oracle® Reference Architecture

BPM Foundation

Release 3.0

E15462-03

September 2010

ORACLE®

ORA BPM Foundation, Release 3.0

E15462-03

Primary Author: Mark Wilkins

Contributing Author: Dave Chappelle, Bob Hensle, Stephen Bennett, Anbu Krishnaswamy, Cliff Booth, Jeff McDaniel

Contributor: Manoj Das, Barry O'Reilly, Pat Krebs, Clemens Utschig-Utschig

Contents

List of Figures

Send Us Your Comments

ORA BPM Foundation, Release 3.0

E15462-03

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this document?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us at its_feedback_ww@oracle.com.

Preface

Following the approach of Oracle Reference Architecture (ORA) the Business Process Management (BPM) Foundation and Infrastructure documents present the Oracle reference architecture viewed from the perspective of technologies associated with business process. While the ORA perspectives are necessarily grounded in Service Oriented Architecture (SOA) the current combination of BPM with SOA offers significant advantages to IT in supporting the rapidly changing demands of today's business communities.

The ORA BPM perspective document set is divided into two major themes. The first document (this document) of the BPM perspective set builds on the ORA SOA Foundation to create the ORA BPM Foundation. This BPM Foundation document describes the general reference architecture for BPM, associated standards, and architectural principles to be applied to Oracle Fusion BPM technology environments. In addition, this BPM Foundation document shows the interlock with ORA core concepts.

The second document in the BPM perspective set, the ORA BPM Infrastructure, extends the foundation architectural principles to build a set of BPM-specific capabilities and to map them to relevant infrastructure components.

Audience

This document is intended for publication to Oracle customers with an interest in BPM and is targeted towards architects and business specialists. It provides the background material as the basis to discuss BPM solutions between Oracle architects and Oracle customers.

How to Use This Document

This document is intended to be read from start to finish. This document is to be publically available and should be shared with Oracle customers interested in BPM solutions and architecture. The contents of this document should provide first-level answers to questions from customers regarding the architecture for BPM and relationship between BPM and ORA core concepts.

Document Structure

This document is organized in chapters spanning introduction, background, conceptual architecture, standards and technologies, ORA interlock, and appendices. Specifically,

Chapter 1 provides an overview of BPM and the scope of this document.

Chapter 2 is a summary of the concepts and brief background to BPM. It was felt that it is necessary to include some background because BPM has been around, in various forms, for a long time and it is important to know what is different now and why. This chapter briefly offers answers to these questions, but those with experience of past BPM architectural approaches may prefer to jump directly to the next chapter.

Chapter 3 presents the core conceptual architecture for this BPM perspective.

Chapter 4 identifies the industry standards and primary technologies that should be employed to develop a modern BPM solution.

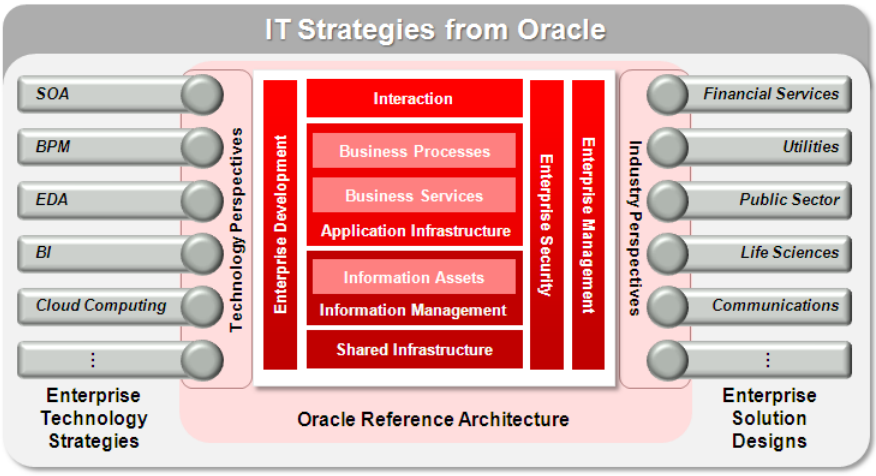
Chapter 5 describes the interlock of this BPM perspective with ORA core concepts.

Chapter 6 is a summary of the document.

Appendix A and Appendix B provide lists of relevant supplementary reading and references.

Related Documents

IT Strategies from Oracle (ITSO) is a series of documentation and supporting collateral designed to enable organizations to develop an architecture-centric approach to enterprise-class IT initiatives. ITSO presents successful technology strategies and solution designs by defining universally adopted architecture concepts, principles, guidelines, standards, and patterns.



ITSO is made up of three primary elements:

- **Oracle Reference Architecture (ORA)** defines a detailed and consistent architecture for developing and integrating solutions based on Oracle technologies. The reference architecture offers architecture principles and guidance based on recommendations from technical experts across Oracle. It covers a broad spectrum of concerns pertaining to technology architecture, including middleware, database, hardware, processes, and services.
- **Enterprise Technology Strategies (ETS)** offer valuable guidance on the adoption of horizontal technologies for the enterprise. They explain how to successfully execute on a strategy by addressing concerns pertaining to architecture, technology, engineering, strategy, and governance. An organization can use this material to measure their maturity, develop their strategy, and achieve greater levels of adoption and success. In addition, each ETS extends the Oracle Reference

Architecture by adding the unique capabilities and components provided by that particular technology. It offers a horizontal technology-based perspective of ORA.

- **Enterprise Solution Designs (ESD)** are industry specific solution perspectives based on ORA. They define the high level business processes and functions, and the software capabilities in an underlying technology infrastructure that are required to build enterprise-wide industry solutions. ESDs also map the relevant application and technology products against solutions to illustrate how capabilities in Oracle's complete integrated stack can best meet the business, technical and quality of service requirements within a particular industry.

ORA BPM Foundation, along with ***ORA BPM Infrastructure***, extend the Oracle Reference Architecture. They are part of a series of documents that comprise the BPM Enterprise Technology Strategy, which is included in the IT Strategies from Oracle collection.

Please consult the ITSO web site for a complete listing of ORA documents as well as other materials in the ITSO series.

Conventions

The following typeface conventions are used in this document:

Convention	Meaning
boldface text	Boldface type in text indicates a term defined in the text, the ***ORA Master Glossary***, or in both locations.
italic text	Italics type in text indicates the name of a document or external reference.
underline text	Underline text indicates a hypertext link.

"SOA Service" - In order to distinguish the "service" of Service-Oriented Architecture from the wide variety of "services" within the industry, the term "SOA Service" (although somewhat redundant) will be used throughout this document to make an explicitly distinction for services that were created as part of an SOA initiative; thus distinguishing SOA Services from other types of services such as Web Services, Java Messaging Service, telephone service, etc.

Architectural terminology - When appearing in this document, the terms view, viewpoint, stakeholder, concern, are used according to the definitions appearing in the IEEE 1471 "Recommended Practice for Architectural Description of Software-Intensive Systems". The term perspective however, does not appear in IEEE 1471 and is defined instead, within the context of ORA, to refer to a particular viewpoint on to the ORA core reference architecture (in this case the technological viewpoint of BPM).

1 Overview

ORA is a product-agnostic reference architecture based on architecture principles and best practices that are widely applicable and that can be implemented using a wide variety of products and technologies. ORA does not include any implementation artifacts for the prescribed architecture. ORA addresses the building of a modern, consistent IT architecture while minimizing the risk of product incompatibilities and obsolescence.

This document set is essentially a reference architecture and infrastructure positioning combined with an assessment and implementation practices. It does not describe how to identify BPM opportunities or what scenarios are appropriate for BPM treatment (see BPM, SOA, and Web2.0 whitepaper for information).

The ORA BPM documents present the ORA architectural concepts from the perspective of BPM, highlighting the specific details of BPM as an elaboration of the ORA core concepts with respect to this technological approach. This ORA BPM extension comprises two documents:

- BPM Foundation: primarily a reference architecture for BPM in IT, including principles, standards, and definition of BPM and its limitations (scope); relationship to ORA.
- BPM Infrastructure: relates the BPM capabilities, as defined by the reference architecture, to the Oracle infrastructure and identifies the role of standards.

1.1 BPM Foundation

The ORA BPM perspective is incremental to the ORA core relying on all underlying descriptions and architectural principles. The relationship of BPM to ORA core concepts is to extend the ORA core concepts to include BPM specific concepts. The underlying ORA core concepts are not reproduced in this perspective document except where it is necessary to show BPM-specific views.

1.2 Scope of BPM Perspective

This document provides a modern architectural foundation for BPM in IT. Although it does not depend on SOA principles, it is fully compatible with SOA i.e. the BPM reference architecture is flexible enough to be used with or without SOA. Of course there are synergies between BPM and SOA that make BPM combined with SOA the preferred option.

As an architectural foundation, the BPM perspective includes a conceptual architecture definition, architectural principles for BPM, and an outline of associated standards and

technologies. It lays the foundation for the next document in the series, the BPM Infrastructure.

This document focuses on the technical and architectural aspects of BPM. Technology is only part of the BPM picture however, and considerable attention needs to be given to business concerns, such as business process re-engineering, process improvement, and the organizational and cultural changes necessary for BPM success. This document does not attempt to offer a approach for the implementation of a BPM strategy, although it is expected that it will be used in conjunction with such a methodology. The strategy for adoption of BPM is covered in Oracle's Approach to BPM.

2

Introduction and Background

Business Process Management (BPM) is defined as a strategy for managing and improving the performance of a business through continuous optimization of business processes in a closed-loop cycle of modeling, execution, and measurement. BPM encompasses the discovery, design, and deployment of business processes; in addition, BPM includes the executive, administrative, and supervisory control of those processes.

So who is responsible for BPM in the enterprise? Although this document does not intend to deal with the organizational issues of BPM implementation (this is covered in the Oracle's Approach to BPM), this is still an important question to consider in the context of an architectural foundation document because IT and "the business" have different perspectives and different concerns. Of course the question is somewhat open-ended because there are many stakeholders in any enterprise initiative, but traditionally BPM was primarily the domain of "the business" (hardly surprising considering the name); however the results of business-driven BPM initiatives often resulted in demands on IT to implement. This "traditional BPM" was popularized originally in the '90s by luminaries such as Champy and Hammer[1] and took the form of Business Process Re-engineering (BRE). The focus of BRE was business process improvement (strictly speaking this is a subject predating Champy and Hammer by nearly a century[2]) that is, making people (workers) and their activities operate more efficiently towards the goals of the business. The manufacturing industry was among the first candidates for "process re-engineering", but the concepts were rapidly adopted by all industry sectors.

Business Process Re-engineering is not what this document is about, however, and BRE is no longer the sole focus of BPM. Soon after BRE was popularized it quickly became a new challenge for IT to address new demands from the business. The resulting disparity between the business and IT in expressing and handling BPM soon became known as the "business-IT gap".

2.1 The Value Proposition of BPM

The core promise of BPM lies in the opportunity to manage, monitor, measure, and improve the way a business runs with the goal of achieving and maintaining optimum performance. Naturally there are many factors contributing to (or detracting from) this utopian idea of optimum performance. Nonetheless, BPM is a tangible approach to continuous improvement in today's technology arena.

---

[1] See Michael Hammer and James Champy (1993) "Reengineering the Corporation: a Manifesto for Business Revolution"

[2] Frederick Winslow Taylor (1911) pioneered "scientific management", publishing guiding principles for business success

2.2 Haven't We Been Here Before?

The promise of BPM is very compelling: optimizing efficiency, taking full advantage of resources, flexibility, and control over business processes, are all critical to success in today's business world. BPM is so compelling in fact that IT is on the third wave[3] of technological solutions to satisfy the need to support BPM through computerization.

Early approaches to BPM were content centric workflow systems involving documents, forms, faxes, etc. Some of the earliest implementations of workflows focused on converting and processing paper-based documents through digitized media, relying on scanners and specialized monitors, with optical storage at the back-end. This was a direct result of business process modeling and re-engineering human participants with limited automation of process flow and management.

The next major phase in BPM recognized the system participants and business partners. Since so much capability was locked inside siloed applications the major technical challenge was integration. Message-oriented Middleware (MoM) and Enterprise Application Integration (EAI) became the core technical solutions integrating disparate applications and systems with a plethora of adapters. These approaches were also extended to include rudimentary workflow modeling and execution. At this stage there were no significant standards governing these architectures and products were mainly proprietary.

In these early stages of BPM development, business process modeling was typically performed by a business analyst and the output was used as a source of requirements definition for software engineering.

One of the major problems with these early efforts to computerize BPM was the starkly different needs of the business specialist and software engineers and the notations and languages used to express them. For example a business process model is concerned with such things as roles and authorities, policies, etc. while a corresponding technical process focuses on mapping and transforming data in messages between systems. Furthermore, the business view of a process is concerned with “what” is happening, while the technical view of a process is concerned with “how”. This gap between the needs and languages of the business and technical communities (the business-IT gap) became the focus of third major effort in addressing the deficiencies in the BPM strategies. In this third wave the Business Process Management System (BPMS or BPM Suite) was created with notations and diagramming interfaces to address the needs of both business and IT stakeholders, executable process models, Business Activity Monitoring (BAM), and the separation of business rules. This new integration of tools and capabilities solved a number of problems and enabled **round-tripping** of models between business, IT, and the process execution platform and provided feedback from the running system.

2.3 Previous Hurdles

Before describing the solutions to the substantial problems facing BPM computerization it is useful to summarize these problems. The following is a list of the concerns that are addressed by the architectural approach described in this document.

- Differing concerns and perspectives of stakeholders (the business-IT gap)
- Systems and applications integration
- Interoperability between tools and application systems
- Separation of human versus system activities

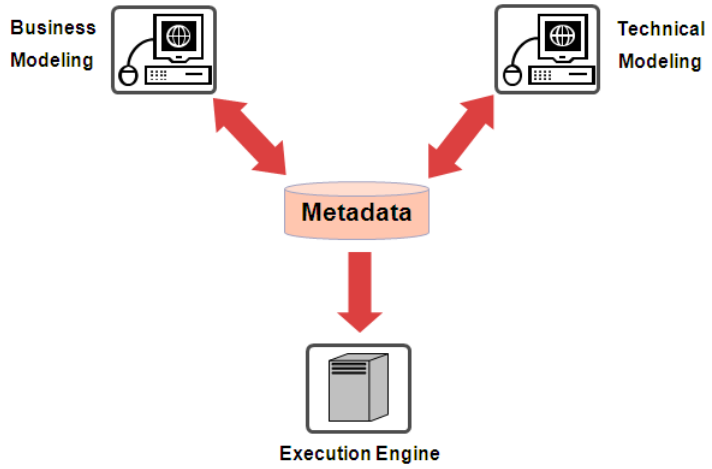
[3] See “Business Process Management: the Third Wave,” by Howard Smith and Peter Fingar

- Exception handling and transaction management

2.4 What Has Changed to Make BPM Work?

One of the major improvements appearing in today's BPMS is recognition of the different needs of the business and IT stakeholders. To address the business-IT gap, the business specialists and IT engineers work with different views of the process models which address their different concerns. Most significantly these models are stored centrally in a metadata repository and are interchangeable so the changes made by one user can be seen by another. In this way the impact of a change by the IT engineer, for example, can be seen by the business specialist using his preferred representation. When the process model is complete it is transferred to an execution engine. This exchange of models through stakeholders and process management systems is shown below in Figure 2–1

Figure 2–1 Model Interchange



This approach is central to the success of BPM and is applied not only to business process modeling and its execution, but also to business rules and forms the basis of the **Business Rules Management Systems** (BRMS).

Originally business rules engines were deployed in isolation, often addressing only very specific problems and always requiring an array of proprietary mechanisms for integration with enterprise applications. Since businesses run on a combination of policies and procedures, BPM technologies have evolved to bring together business rules and process automation to meet these needs in a cohesive manner.

Service Oriented Architecture (SOA) largely addresses the issue of integration (among other things) for both BPM and BRMS. The SOA approach to integration differs from the EAI (or MoM) by providing a common baseline for interoperability rather than adapters for every application (see ***ORA Service-Oriented Integration*** document for more information).

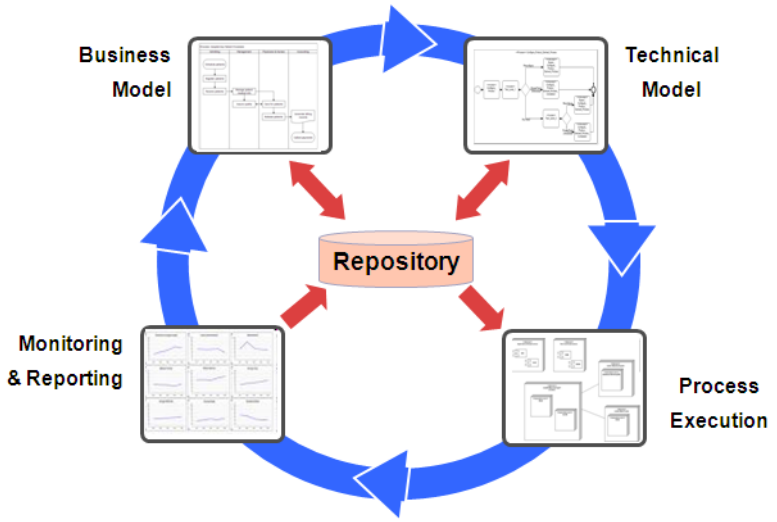
SOA also supports alignment of IT with the business by raising the level of abstraction from technologies to business services. In addition SOA establishes a business vocabulary for IT, simplifying the interpersonal communication between IT and business groups reducing the need for complex mapping of business requirements to software specifications. In particular SOA supports the ability for the business process model to refer directly to executable SOA Services without the need to understand the underlying technical details of those SOA Services.

Standards of course are central to the major improvements in integration, whether between applications or enterprises, and interoperability between tools.

2.5 Business process lifecycle

The term “lifecycle” often conjures images of project implementation and engineering methodologies; however, since BPM is a multi-discipline activity it is appropriate to consider the key steps in the cycle of continuous improvement. In particular BPM is a Model-Driven Architecture (MDA), that is to say, its runtime activities are driven by design-time graphical models. The BPM lifecycle is depicted in Figure 2–2 below.

Figure 2–2 BPM Lifecycle



As we have discussed in the previous chapter, BPM has roots in business process analysis and process re-engineering, originally having little to do with automation, but this new approach to BPM does much more than merely supporting process modeling.

2.5.1 Modeling and Simulation

Modeling is an original and fundamental part of business process analysis. In the context of BPM, modeling refers to the development of a graphical representation of a business process. Ideally the graphical representation should conform to a standardized notation that can be readily understood in an unambiguous way by all stakeholders. In particular the model should readily exchange between business specialists and IT engineers.

Simulation is a theoretical execution of the modeled business process. Typically the business specialist sets the parameters for the simulation describing an environment that mimics a real-world scenario (for example, number of concurrent users, profile of delay between process steps, etc.). The simulation tool (usually an extension of the modeling tool) runs the scenario and the efficiency of the business process can be predicted within boundaries of supplied parameters.

2.5.2 Process Implementation and Execution

In order to take the step from just doing modeling for the sake of business analysis (business process re-engineering, etc.) it is necessary to automate the process: that is to

take it from a diagram to a form that can be executed by a machine. This may be achieved in a number of ways:

- A comprehensive process representation is maintained in a central metamodel repository such that multiple executable and graphical representations can be extracted from, or inserted to, the central store. This approach provides maximum flexibility, but a highly complex metamodel (see Section 4.3, "Business Process Definition Metamodel (BPDM)" for an example).
- The graphical model is maintained in one form while a transformation is required to move it into an executable language. This is the simplest approach, although it can suffer from discontinuity when either one of the underlying metamodels undergoes enhancement to a new version (e.g. **BPMN** v1 conversion to **BPEL**).
- The metamodel for the graphical and executable representations is the same. Here we don't have unlimited flexibility to transform to or from any tool as a core tenet, but graphical and executable forms are guaranteed to be maintained in lock-step (see Section 4.1, "Business Process Modeling Notation (BPMN)" for an example).

Process execution is supervised by a process manager that is responsible for directing process flow, processing flow logic, and may direct message exchange. The process manager does not, however, execute the activities or host the running of the code in the underlying business functions.

2.5.3 Process Monitoring and Optimization

The greatest opportunity for process improvement arising from the closed-loop is in the feedback mechanism available from observing the process running in a live environment. Simulation, mentioned earlier, goes only so far. It is rarely practical to create a simulation environment that exactly matches a real-world situation, especially when human activities are involved in a long-running, complex process.

Monitoring by itself, of course, doesn't deliver the necessary intelligence to change a business process to its optimum form. That requires application of analysis and intelligence. Business Activity Monitoring (BAM) provides the information about the operation of the business process needed to support business specialists and technologists in their pursuit of process optimization. Various other inputs are needed to achieve process improvement, some beyond the scope of mere “activity monitoring” (not least of all the business objectives of the process itself). Some of this additional information may be provided by Business Intelligence; however, this topic is beyond the scope of this document, but can be found in the ORA BI Perspective documents.

2.6 What Is a Business Process?

A process describes a sequence or flow of activities in an organization with the objective of carrying out work. The flow of a process is controlled by decisions, triggers, events, and exceptions, any of which may be influenced by rules or policies.

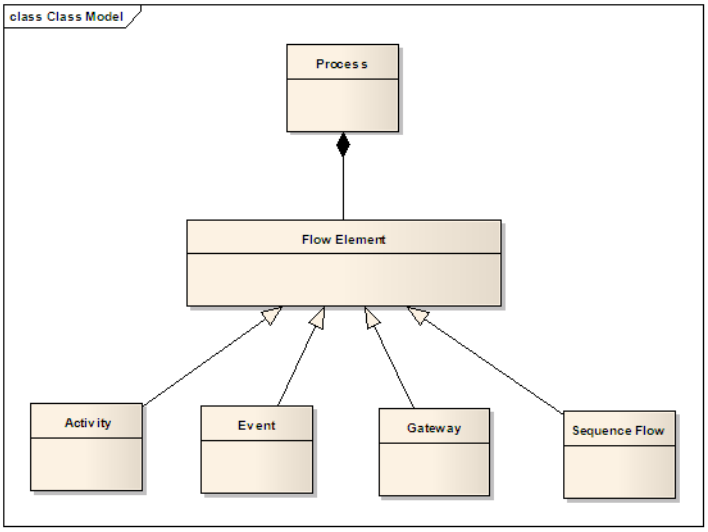
A process has performance indicators which may be compared to actual performance during execution. The activities in a process are performed by system or human actors (or groups conforming to a role) which may be internal to an organization or external (business partners).

2.6.1 Business Process Modeling Terms and Concepts

In modeling terms, a process is depicted as a graph of Flow Elements, which are a set of Activities, Events, Gateways, and Sequence Flow. Figure 2–3 below shows the basic

elements of a process in a simplified meta-model represented in UML. The paragraphs following describe the meaning and purpose of these basic elements.

Figure 2–3 Process Metamodel



2.6.1.1 Process

A Process may be defined at any level from enterprise-wide to a Process performed by a single person. (BPMN uses the term Process specifically to mean a set of flow elements, while it uses the terms Collaboration and Choreography when modeling the interaction between Processes). In BPMN a Process is an abstract concept and does not have a specific graphical notation, instead, it is a set of graphical objects.

2.6.1.2 Activity

An Activity is work that is performed within a Business Process. An Activity can be atomic or non-atomic (compound). The types of Activities that are a part of a Process are: Task, Sub-Process, and Call Activity, which allows the inclusion of re-usable Tasks and Processes in the diagram. Activities represent points in a Process flow where work is performed. They are the executable elements of a BPMN Process.

2.6.1.3 Event

An Event is something that "happens" during the course of a business process. These Events affect the flow of the Process and usually have a trigger or a result. They can start, interrupt, or end the flow. In BPMN, Events attached to Activity boundaries can be set to interrupt, or not interrupt, the Activity (for non-interrupting Events the Activity will continue and the flow will continue without the Event).

2.6.1.4 Gateway

Gateways are the decision points in a process flow. Gateways are used to control how Sequence Flows interact as they converge and diverge within a Process. The term "Gateway" implies that there is a gating mechanism that either allows or disallows passage through the Gateway.

2.6.1.5 Sequence Flow

A Sequence Flow shows the order in which the activities will be performed in the Process. The source and target for a Sequence Flow can be an Activity, Event, or Gateway.

2.6.1.6 Task

A Task is an atomic activity that is included within a Process. A Task is used when the work in the Process is not broken down to a finer level of Process Model detail. BPMN 2.0 defines Specialized Tasks that have standard markers to indicate their type, such as, service task, user task, send task, etc.

In many business workflows, human involvement is required to complete certain Tasks specified in the workflow model. BPMN specifies two different types of Tasks with human involvement, the Manual Task and the User Task.

3

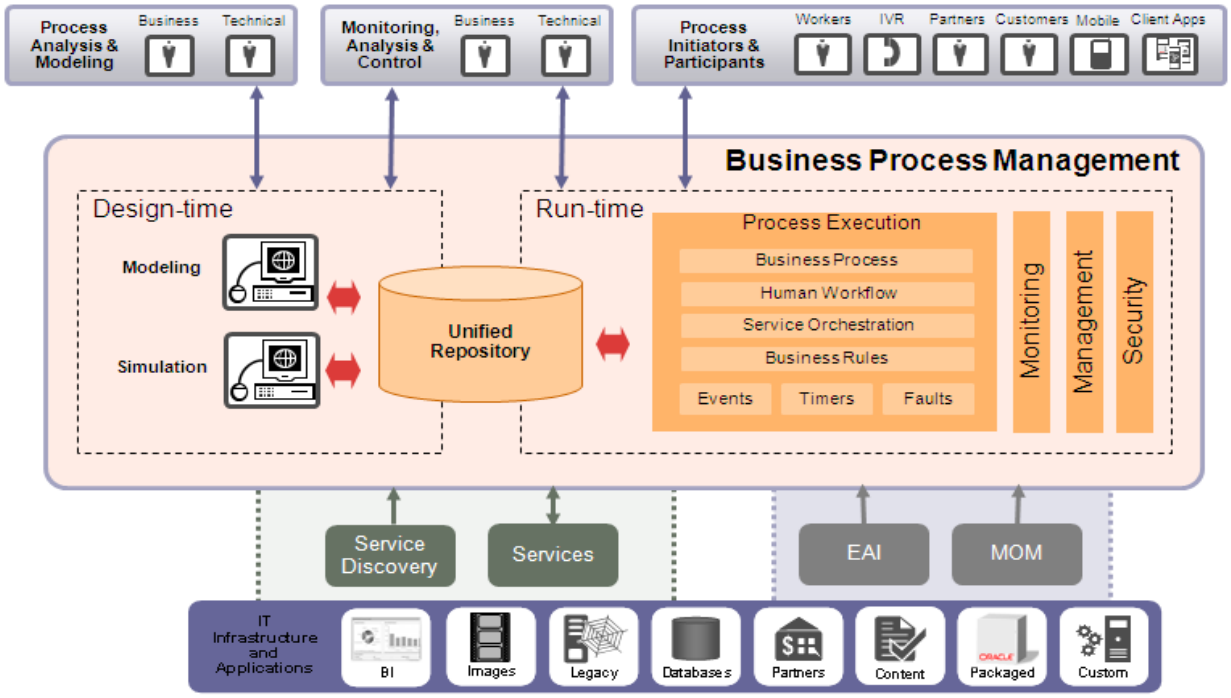
BPM Conceptual Architecture

The conceptual architecture highlights the core capabilities and the primary relationships that belong specifically to a BPM system, unencumbered by implementation details. At this conceptual level of abstraction the BPM architecture can be seen to be comprised of modeling capabilities in the design-time portion with definitions of various types of process orchestrations and business rules conveyed, via some form of repository, to the BPM runtime. The details of how the various forms of process orchestration are stored and transferred, and whether they require transformation to their executable state, are implementation details that can be found in the ORA BPM Infrastructure document.

Figure 3–1 below shows the core of the BPM conceptual model and its relationships to external systems. The underlying (external) integration layer may be provided by SOA or some other approach. The participants in business process system services and their various delivery channels are shown at the top of the diagram. These are the participants and initiators in the business process activities.

The lower portion of the diagram (below the BPM core) shows the systems that provide the business functions that make-up the activities being orchestrated by the process execution system. The arrow from the Process Execution to provider applications is shown here in one direction (process engine using application functionality), but applications are often also initiators of business process, in which case they are represented with initiators and participants at the top of the diagram.

Figure 3–1 BPM Conceptual Architecture



3.1 Modeling

Unlike traditional software engineering BPM relies heavily on models. This is a different paradigm from the requirements analysis, design, implementation, sequence of events of classic software engineering. BPM consists of workflows, business rules, process models, information flows, and integration, all of which should be represented by the integrated models that constitute the BPM solution.

There are a number of model types that are involved in BPM and that should be closely related or, ideally, fully integrated in the same repository.

- Business strategy models are needed to describe the overall enterprise goals and to provide traceability and justification to the BPM solutions. These include Business Motivation Models (BMM), Balanced Scorecard / Strategy Maps. When a close association to these core business models is maintained a number of potential benefits arise including: traceability of requirements, impact analysis for changes, project justification, and Key Performance Indicators (KPIs) that can be derived to measure effectiveness in association with Business Activity Monitoring (BAM).
- Conceptual Business Process Models are typically used at a high-level to depict a process conceptually without regard for actual implementation. These models are used to communicate and understand the basic conceptual process before modeling lower-level models that depict implementation details.
- Organizational models: roles are particularly important when human tasks are included in the business process. These can be modeled using techniques defined in OMG Organization Structure Metamodel (currently work-in-progress).
- Business rules: while the processes provide the procedural flow, the rules (and policies) provide declarative statements of decisions that control the process flow. Rules and policies commonly provide thresholds for service levels, financial approvals, time constraints, etc. A standardized approach for describing rules can be found in OMG Semantics of Business Vocabulary and Business Rules.

- Business process diagrams: the core of the BPM modeling activity, typically made up of swim-lanes, tasks, activities, decisions, and routing.
- Business functional (decomposition) models
- Information models
- SOA Service models

Models have language, notation, and interchange format. Formal modeling requires a consistent, unambiguous notation that is capable of representing both the business aspect of a process as well as the technical implementation details required by the runtime system. Ideally the modeling notation should support separate rendering of business and technical forms that exchange seamlessly between these representations without loss of information or meaning.

Ultimately the technical representations of both business process and rules should contain sufficient detail to be executed in the runtime system.

Architectural Principles

1. Models should be linked to enable traceability from original business motivation to technical implementation of business process.
2. A modeling language should enable bi-directional exchange of process definition between business and technical representations.
3. Business process and rules modeling languages should enable transformation from design-time visual notation to runtime executable representation.

3.2 Model Repository

While in practice the process and rules definitions may occur in various forms they have been grouped as models under one conceptual repository for the purposes of the BPM conceptual model (the logical and physical manifestations are discussed in the ORA BPM Infrastructure document). A number of model types are required for a complete BPM system. The minimum set is shown in the diagram as business process, technical workflow, human tasks, and business rules. Executable models for both process and rules are implied by the arrows conveying the models into the runtime system. Business rules and various process types are described in separate sections of this chapter.

The repository primarily contains metamodel information, i.e. a "model of the model" (just as metadata is data about the data). The information contained in the repository needs to fulfill a number of important functions. The metamodel must contain sufficient information about a process orchestration to satisfy the needs of (and present separate representations to) the business specialist and technical architect / engineer. This means it needs to describe a process while relating it to such things as a business model (business plan, motivation model, etc.) and organizational model on the one hand, while associating it with the technical services (potentially SOA Services) on the other.

Architectural Principles

1. Metamodel must support the (bidirectional) interchange of process details between stakeholders with differing concerns and viewpoints (i.e. business specialists and technologists)
2. Metamodel must include layout and other graphical characteristics for exchange of models between tools.

3. Metamodel must either directly provide an execution language (i.e. the model is executable) or support seamless translation into an executable language.

3.3 Business Rules

Business rules are fundamentally expressions of business policies.

The abstraction of business rules and the expression of business policy are critical to BPM. When rules are embedded in programmatic functions they become monolithic operations that are hard to change, and therefore have limited value and longevity. However, when rules are managed in a declarative form, separately from operational code, they can be better understood and maintained more easily ideally by the business stakeholders.

Architectural Principles

1. Business Rules must be abstracted from programmatic functions and maintained as separate, declarative representations that are both understandable by the business users and executable by the rules systems.
2. Business Rules must be modifiable dynamically and independently of functions and process flows using them.
3. Business solutions that contain decision logic that is likely to change more frequently than the solution itself, or is likely to be reused across solutions, must implement the decision logic using externalized rules.
4. Rules must be designed and deployed in a manner that supports rule changes without code deployments, reboots, or downtime.
5. Rules must be shareable across solutions.

3.4 Process Execution

The conceptual view of process execution, as shown in the diagram in Figure 3–1, encompasses a number of capabilities that a BPM system must ultimately provide. Each of these capabilities is described in the following sections.

3.4.1 Execution Control

The control of process execution is a core capability of the BPM runtime system. Execution control provides the ability to make flow decisions based on the evaluation of business rules, manage the serial or parallel execution of activities, and manage the interaction with the other components of the BPM core according to the specifications of the process model.

3.4.2 Transactions Control

The term transaction refers to the coordinated execution of multiple units of work with the requirement that they all complete or roll-back; that is, if one of the units fails the whole transaction fails and all units are instructed to undo any work they performed so the system as a whole is restored to its original state.

Applications and databases typically use **ACID transactions** (Atomic, Consistent, Isolated, and Durable) that are coordinated by a transaction manger using a **two-phase commit** protocol.

Providing ACID transaction support for atomic business activities and short-lived technical orchestrations is a major benefit over simple pipeline processing systems.

This offers the process designer the capability to declare where transaction boundaries exist, ensuring integrity across technical activities.

In contrast, business processes that orchestrate business activities tend to have their state committed at the completion of each process step. While each step can be an ACID transaction, the process itself can run for days, weeks, or months, such that it is not appropriate for classical ACID transaction management to span the entire transaction. If we were to do this then Process Execution would have the limitation that the resources cannot be used by other users or systems until the transaction either completes or fails (via commit or rollback), which in a business process spanning hours, or even days, is likely to be completely unacceptable.

Long running business processes typically use a different technique called **compensating transactions** when the process is aborted before completing normally. For example, a purchase order is cancelled after a credit check does not pass. In such cases data will be committed to permanent storage, but may subsequently need to be undone. Unlike conventional rollbacks, specific business logic will typically be required to undo a long lived transaction and restore the system to its original state. This type of transaction differs from two-phase commit, because although both types of transactions can result in multiple data stores being updated, compensating transactions permit updates to span a long period of time. Unlike two-phase commit, compensating transactions means that other users or systems will see the changes that have been made even if those changes are later undone.

New transaction protection standards have also emerged to support the needs of BPM. WS-Transaction is a collection of standards including WS-Coordination, WS-Atomic Transaction, and WS Business Activity Framework (see Chapter 4 for more information). These provide support for both short and long-running transactions between Web Services. Atomic transactions are tightly coupled, however, so it is important to ensure that the appropriate transaction approach is used when implementing SOA with Web Services in order to maintain the key SOA benefit of loose coupling.

3.4.3 State Management

Business processes are often long running, for example those involving human tasks, such as an insurance claim awaiting an assessor's decision, or a loan application waiting on an underwriter. Other examples of wait-states in business processes might involve timers (e.g. end of month revenue reporting) or events (e.g. a stock reaches a certain price), etc. (see Section 3.4.4).

These long running processes should not be permitted to consume system resources during periods of (system) inactivity so the process state must be temporarily transferred to permanent storage. This "dehydration" of a process can have serious consequences for in-flight transactions (for example what if a loan applicant decides to modify his loan request while the process is waiting on the underwriter) and requires careful coordination with transaction management (see Section 3.4.2).

3.4.4 Handlers

As we have seen already business processes often stop and wait for certain triggers before proceeding. Triggers can be used to start, stop, pause, or continue a process or potentially even change the path of its flow. Types of triggers include events (e.g. a shipment arrives), timers (e.g. status must be checked at noon every day), and faults (e.g. an unexpected result is returned or processing fails due to an exception). These require special handling for coordination with process control, state management, and possibly transaction management.

3.4.4.1 Events

Traditional flow diagrams (flowcharts) and applications focused on what people and systems are expected to provide during the flow of a business process. However, in reality many other things can happen during the operation of a business process, such as, a customer cancelling an order or a business partner's system going off-line. These events must be handled, first by the modeling notation and ultimately by the process execution system.

Architectural Principles

1. The model and the execution system must be able to describe and handle timeouts, exceptions, and other occurrences in the form of events.

3.5 Monitoring

Monitoring covers a wide spectrum of activities, but in the context of BPM we are concerned with tracking performance indicators that relate specifically to the efficiency of the business process. This type of monitoring is generally referred to as Business Activity Monitoring (BAM), outlined in the previous section. BAM should be expected to monitor such things as business exceptions, number of times a process is run, ratio of successful process runs vs. aborted runs, Key Performance Indicators (KPI), etc.

In general the details of this component of the BPM system is an implementation concern and can be found in the ORA BPM Infrastructure document.

3.6 Security

As with a number of the topics covered in this section, security is a broad general issue having only a small subset of concerns relating specifically to BPM. While the generic concerns of authorization, authentication, access control, etc. apply equally to BPM as for any other technology strategy, only those concerns unique to BPM will be addressed here.

The primary issue requiring special treatment for BPM systems is the question of propagation of authority to perform functions and/or access information. This issue is certainly not new since any distributed processing system has to address the same problem: that is when (system) process control is passed from one system/application to another, or when one system/application invokes another, there is a need for the first to communicate its authority over the second. For example, in an automated process that handles employee vacation requests, the final step in the process needs to update the employee's balance of vacation time in the HR application, but how does the automated process communicate its authority to perform this function? If the BPM system is granted carte blanche over the HR application this would have the potential to lead to serious violations of security. In this example a much more fine-grained permission mechanism is needed.

The initial authority to perform a system function is normally conveyed by a person via login procedure, however this procedure is unsuited to system-to-system communication and another form of security token passing is required.

Fortunately a number of security mechanisms exist already. In particular, SOA fully addresses this issue with a number of well established standards. Similarly, other integration architectures, such as EAI and MoM, address this issue.

There are numerous other concerns that fall under this heading, such as identity propagation, trust, confidentiality, and message integrity. For further information on the broader topic of security consult the ***ORA Security*** document.

3.7 Process Types

The terms workflow, task-list, and business process tend to be used interchangeably, but this document uses them to distinguish between three important categories of processes.

Similarly the word "orchestration" is commonly (and correctly) applied to any of these process categories since it refers to any directed set of activities.

These are described in detail in the following sections.

3.7.1 Technical Orchestration

Technical orchestration refers to process flows that are typically associated with EAI and MoM. In the case of SOA a number of infrastructure components can also provide this form of orchestration, including the Enterprise Service Bus (ESB).

These flows should be short-lived and stateless, otherwise they may lock important resources causing delays for other users and processes. Simple technical orchestrations may be appropriately maintained within a traditional software component, but more complex flows may benefit from the supervisory control of a process manager. While these technical orchestrations do not contain key business process they may constitute a business activity itself which then becomes a participant in a larger business process.

A simple, but familiar, example of a technical orchestration is an on-line e-commerce shopping cart. A web site provides a simple technical orchestration enabling the user to put objects in, change quantities, and finally place an order by navigating to "checkout". A number of basic technical rules are applied during this flow, such as, validating that the cart has objects in it before proceeding to checkout. The technical orchestration and rules seen here were traditionally embedded in applications but, they clearly benefit from a level of abstraction, as in an implementation framework like struts, where the flow and rules are maintained separately from the programmatic code in a distinct, declarative format. As with many web applications the process must be stateless to avoid locking shared resources like the product catalog in this scenario (fortunately the world-wide-web is intrinsically stateless). Generally there is little or no business process in an e-commerce shopping cart and the system would not benefit from the full control of a BPM process manager. The cart is a manifestation of a common business activity that might be called "place an order" however, and this is part of a larger business process.

Architectural Principles

1. The architecture must support short-lived, stateless workflows.

3.7.2 Human Tasks

Many activities (or tasks) must be performed by people rather than computers. These cases arise for various reasons, such as activities requiring research or other intelligent assessment, approval, authorization, or the activity has simply not (yet) been automated. Also, tasks requiring human to human interaction: some tasks will always be performed by a human, such as, providing patient care or conducting an employee performance review meeting.

Human tasks are especially challenging to include in a process automation. Many variables are beyond the control of the systems process manager, such as, the availability of workers to perform the task, the skill level of the worker is highly subjective, and the work product can rarely be adequately validated by the computer system, to mention just a few.

The human involvement in a business process is often an inescapable reality and for this reason much investment has been made in human interfaces and standards (see WS-HumanTask, etc. in Section 4.2.1).

Architectural Principles

1. The architecture must support long-lived, stateful processes when involving human tasks.

3.7.3 Business Process

The term business process is often used incorrectly to describe any type of orchestration, however it is specifically an orchestration of business activities and should not be confused with technical workflow or system processes.

Commonly accepted definitions take the following form: a business process is a set of linked activities performed by people and systems that deliver some kind of business value through a product or service to internal or external customers.

According to Wikipedia

A business process or business method is a collection of related, structured activities or tasks that produce a specific service or product (serve a particular goal) for a particular customer or customers. It often can be visualized with a flowchart as a sequence of activities.

In the e-commerce example, cited earlier in this chapter, we showed the technical workflow of the shopping cart was not a business process in itself, but constituted an ordering activity. The shopping cart is a vehicle for conveying an order within an ordering activity. Ordering is a typical activity in a business process. In this example the order received from the customer via the shopping cart must be processed according to specific business rules, meet service level expectations (e.g. ship within 24 hours), and handle business exceptions (e.g. out-of-stock, returns processing).

Architectural Principles

1. The architecture must support long-lived, stateful business processes.

3.7.4 Service Composition

In service composition (in a Service-Oriented Architecture) the flow of messages between services is controlled by a (declarative) rules based routing mechanism.

More information about service composition can be found in the ORA SOA Foundation document.

3.7.5 Other Types and Terms

There is no strict definition of process types. Instead types and categories must be chosen to meet the needs of the environment. Three types of business processes are defined by the BPMN specification:

- Private non-executable (internal) business processes
- Private executable (internal) business processes
- Public processes

For organizational purposes (rather than the technical categories of the foregoing sections) business processes may be categorized as follows:

- Management processes: the processes that govern the operation of an enterprise. Typical management processes include corporate Governance and Strategic Management.
- Operational processes: processes that constitute the core business and create the primary value stream. Typical operational processes are purchasing, manufacturing, marketing, and sales.
- Supporting processes: processes that support the core processes. Examples include accounting, recruitment, and technical support.

Modeling languages (BPMN and UML in particular) extend this set of process types with some additional terms (e.g. conversations, collaborations, etc.). These additional terms have specific meanings to support the rigor of the associated languages and notations and can be found under the appropriate section in Chapter 4 of this document.

3.7.5.1 Orchestration

Orchestration is the representation of the flow and control of a process. In BPM an orchestration is a managed sequence of activities subject to external control (think of an orchestra being led by a conductor). This part of the definition is significant when contrasting with choreography (see below).

The term orchestration is regularly applied to any of the foregoing types; however, it is useful to make a distinction between the high-level business process orchestration (including "system-centric workflow" and "human-centric workflow") and low-level technical orchestration. This distinction is not clearly defined, but nevertheless, it is useful.

3.7.5.2 Choreography

Choreography describes what occurs between participants in a process. It is a representation of an interaction involving one or more message exchanges between two or more participants. Unlike orchestration, choreography has no central controller, responsible entity, or observer of the process (as in the case of a dance where the participants know how to perform their own steps, but while their interaction was originally planned by a choreographer, there is no director of the final performance).

A choreography is still a type of process, but differs in purpose and behavior from an orchestration. A regular process, or an orchestration process, defines the flow of activities of a participant or organization. In contrast, choreography formalizes the way business partners coordinate their interactions. The focus is not on orchestrations of the work performed by these participants, but rather on the exchange of information (messages) between them.

Another way to look at choreography is to view it as a type of business contract between two or more organizations.

As a definition of expected behavior (basically a procedural business contract) between participants a choreography is useful for organizing interactions between business entities without requiring (1) a single authoritative controller or (2) to divulge the details of individual proprietary business processes (orchestrations).

3.8 Business Process Scope

The scope of a process is determined by the domain boundaries it crosses. These may be either business domains (departments, enterprises, etc.) or system domains (e.g. separate applications or data stores). Scope therefore has implications for concerns like

security and monitoring, as discussed in security (see Section 3.6) a process spanning application systems, business units, or even enterprises will require special considerations for its authority to operate in multiple domains.

If, for example, a process is confined to run within a single function of application (CRM, HRMS, etc.) it has a very narrow scope. In fact some may argue that it is not performing a business process orchestration at all. More likely it is performing a business activity that may involve some kind of technical orchestration.

3.8.1 Intra-Application

Many sets of directed tasks occur within an application, these are called intra-application processes.

Some intra-application processes are not business processes by our definition here, but are instead simple technical workflows making up business activities. However, there are many applications that do perform business processes. A Customer Relationship Management (CRM) application, for example, coordinates business processes "intra-application".

3.8.2 Private (Internal) Business Processes

Private business processes are the most common use for BPM. Within an enterprise business processes span multiple applications, systems, and often business units. In this typically heterogeneous environment, BPM abstracts the business processes from the systems and finds the greatest opportunity to optimize them through BAM and other analytical techniques.

3.8.3 Public Processes

Public processes occur between enterprises. In this category the enterprises involved are generally business partners (e.g. a manufacturer and a distributor), but the relationship could be more obscure, such as a business interaction with a regulatory agency.

In the case of a public process no one entity "controls" the process and the individual participants do not expose their internal business processes. This type of process is described by a choreography (defined in Section 3.7.5.2).

4

Standards and Technologies

The fundamental purpose of most standards specifications is to enable portability and interoperability: where portability is the ability to take design-time artifacts created in one vendor's environment and use them in another vendor's environment, and interoperability is the capability for multiple components to interact using well-defined messages and protocols. This enables users to combine components from different vendors while allowing seamless integration execution.

4.1 Business Process Modeling Notation (BPMN)

BPMN was originally developed by the Business Process Management Initiative (BPMI), and is currently maintained by the Object Management Group since the two organizations merged in 2005. The current version of BPMN (as of January 2009) is 1.2, while major revisions are currently in an advanced state in version 2.0.

Extract from BPMN specification:

'There has been much activity in the past few years in developing Web Service-based XML execution languages for Business Process Management (BPM) systems. Languages such as WS-BPEL provide a formal mechanism for the definition of business processes. The key element of such languages is that they are optimized for the operation and inter-operation of BPM Systems. The optimization of these languages for software operations renders them less suited for direct use by humans to design, manage, and monitor Business Processes. WS-BPEL has both graph and block structures and utilizes the principles of formal mathematical models, such as pi-calculus. This technical underpinning provides the foundation for business process execution to handle the complex nature of both internal and B2B interactions and take advantage of the benefits of Web Services. Given the nature of WS-BPEL, a complex Business Process could be organized in a potentially complex, disjointed, and unintuitive format that is handled very well by a software system (or a computer programmer), but would be hard to understand by the business analysts and managers tasked to develop, manage, and monitor the Process. Thus, there is a human level of "inter-operability" or "portability" that is not addressed by these Web Service-based XML execution languages.'

'Business people are very comfortable with visualizing Business Processes in a flow-chart format. There are thousands of business analysts studying the way companies work and defining Business Processes with simple flow charts. This creates a technical gap between the format of the initial design of Business Processes and the format of the languages, such as WS-BPEL, that will execute these Business Processes. This gap needs to be bridged with a formal mechanism that maps the appropriate visualization of the Business Processes (a notation) to the appropriate execution format (a BPM execution language) for these Business Processes.'

'Inter-operation of Business Processes at the human level, rather than the software engine level, can be solved with standardization of the Business Process Modeling Notation (BPMN). BPMN provides multiple diagrams, which are designed for use by the people who design and manage Business Processes. BPMN also provides a mapping to an execution language of BPM Systems (WS-BPEL). Thus, BPMN would provide a standard visualization mechanism for Business Processes defined in an execution optimized business process language.'

'BPMN provides businesses with the capability of understanding their internal business procedures in a graphical notation and will give organizations the ability to communicate these procedures in a standard manner. Currently, there are scores of Process modeling tools and methodologies. Given that individuals will move from one company to another and that companies will merge and diverge, it is likely that business analysts are required to understand multiple representations of Business Processes-potentially different representations of the same Process as it moves through its lifecycle of development, implementation, execution, monitoring, and analysis. Therefore, a standard graphical notation will facilitate the understanding of the performance Collaborations and business transactions within and between the organizations. This will ensure that businesses will understand themselves and participants in their business and will enable organizations to adjust to new internal and B2B business circumstances quickly. BPMN follows the tradition of flowcharting notations for readability and flexibility. In addition, the BPMN execution semantics is fully formalized. The OMG is using the experience of the business process notations that have preceded BPMN to create the next generation notation that combines readability, flexibility, and expandability.'

'BPMN will also advance the capabilities of traditional business process notations by inherently handling B2B Business Process concepts, such as public and private Processes and Choreographies, as well as advanced modeling concepts, such as exception handling, transactions, and compensation.'

Business Process Modeling Notation (BPMN) is primarily a graphical representation for specifying business processes in a workflow. Version 1 has proven to be a rich notation for modeling business processes and is well established in the business community (with 55 cataloged implementations as of July 2009). While some tools have the ability to transform models between notation formats and into executable process languages BPMN v1 lacks a formal standard repository, interchange, and executable formats. BPMN 2.0 has set out to fill these gaps and provide a complete specification to enable graphical exchange between tools, support various stakeholder views (targeting both business and technical communities), and establish a standard repository.

While BPMN v1 was widely accepted as a the graphical notation for process diagramming, translation to an executable (or interchangeable) form was left to proprietary techniques provided by the BPM Suites.

The vision of BPMN 2.0 is to have a single specification that defines the notation, metamodel and interchange format, providing distinct views for various stakeholders and enabling execution of the process model without the need for writing code (i.e. in a "zero-code environment"). The proposed features include:

- Aligning BPMN with the business process definition meta model **BPDM** (see Section 4.3) to form a single consistent language
- Enabling the exchange of business process models and their diagram layouts among process modeling tools to preserve semantic integrity
- Expand BPMN to allow model orchestrations and choreographies as stand-alone or integrated models

- Support the display and interchange of different perspectives on a model that allow a user to focus on specific concerns
- Serialize BPMN and provide XML schemes for model transformation and to extend BPMN towards business modeling and executive decision support.

BPMN 2.0 provides an interchange format that can be used to exchange Process definitions (both domain model semantics and visual layout) between different tools. A major goal of the specification is to enable portability of Process definitions, such that users can take Process definitions created in one vendor's environment and use them is another. The model interchange format specification supported by the OMG is XML Model Interchange (XMI).

Oracle is a major contributor to the BPMN specification and strongly endorses its use throughout the BPM lifecycle.

BPMN 2.0 and corresponding tools and infrastructure will provide many advantages over other BPM standards and technologies, in particular, providing an integrated repository, multiple views, **round-tripping**, and full lifecycle support in a **zero-code environment**.

4.1.1 BPMN Process Elements

The BPMN process modeling type set consists of Collaboration and Process diagram elements, including all Task types, embedded Sub-Processes, CallActivity, all Gateway types, all Event types (Start, Intermediate, and End), Lane, Participants, Data Object (including DataInput and DataOutput), Message, Group, Text Annotation, Sequence Flow (including conditional and default flows), Message Flow, Conversations (limited to grouping Message Flow, and associating correlations), Correlation, and Association (including Compensation Association). The set also includes markers (Loop, Multi-Instance, Transaction, Compensation) for Tasks and embedded Sub-Processes.

4.1.2 BPMN Scope

The BPMN 2.0 specification extends the scope and capabilities of the BPMN 1.2 in several areas and also resolves known BPMN 1.2 inconsistencies and ambiguities. The key extensions from the original speciation are:

- Formalizes the execution semantics for all BPMN elements
- Defines an extensibility mechanism for both Process model extensions and graphical extensions
- Refines Event composition and correlation
- Extends the definition of human interactions
- Defines a Choreography model

The BPMN specification is intended to support only the concepts of modeling that are applicable to business processes. This means that other types of modeling done by organizations for business purposes are out of scope for BPMN. Therefore, the following broader aspects of business modeling are beyond the scope of BPMN:

- Definition of organizational models and resources
- Modeling of functional breakdowns
- Data and information models
- Modeling of strategy
- Business rules models

Since these types of business modeling either directly or indirectly affect business processes it is important to be aware of the standards (or merely established techniques) and the tools available to support (and ideally integrate) these related models.

4.1.3 BPMN Diagram Types

The BPMN 2.0 aims to cover three basic models of Processes: private Processes (both executable and non-executable), public Processes, and Choreographies. Within and between these three BPMN sub-models, many types of Diagrams can be created. The following are examples of Business Processes that can be modeled using BPMN 2.0:

- High-level non-executable Process Activities (not functional breakdown)
- Detailed executable Business Process
- As-is or old Business Process
- To-be or new Business Process
- A description of expected behavior between two or more business Participants (i.e. a Choreography).
- Detailed private Business Process (either executable or non-executable) with interactions to one or more external Entities (or "Black Box" Processes)
- Two or more detailed executable Processes interacting
- Detailed executable Business Process relationship to a Choreography
- Two or more public Processes
- Public Process relationship to Choreography
- Two or more detailed executable Business Processes interacting through a Choreography

4.2 Business Process Execution Language (WS-BPEL)

WS-BPEL (formerly known as BPEL4WS) is an OASIS specification that draws from Web Services Flow Language (WSFL) from IBM and XLang from Microsoft. IBM, BEA, and Microsoft jointly developed the first version of WS-BPEL in August 2002; other contributors joined to produce version 1.1 in 2003, while the current version 2.0 was formally adopted in 2007.

WS-BPEL uses an XML-based vocabulary to specify and describe executable business processes and is based on WSDL, XML Schema, and XPath specifications.

The purpose of WS-BPEL is to provide an executable expression of process flow between Web Services. In addition to flow a WS-BPEL process defines input and output messages and externalizes its business functionality as a Web Service. In summary WS-BPEL provides orchestration for Web Service composition and produces Web Services.

Often abbreviated to simply "BPEL", WS-BPEL is used within enterprises to standardize application integration between otherwise isolated systems across business units. WS-BPEL is also used between enterprises to standardize and simplify integration between business partners.

BPEL can be used to define both simple and complex business processes offering constructs such as loops, branches, variables, assignments, etc. WS-BPEL is used to direct the invocation of Web Services operations, synchronously or asynchronously,

serially or in parallel. It provides a rich vocabulary with support for fault handling, correlation, long-running transactions, and compensation.

The key features of WS-BPEL are:

- Describes the logic of business processes as a composition of Web Services
- Rich sequencing semantics including parallel and asynchronous processing
- Uses a compensation based long running transaction (LRT) model
- Provides scope-specific fault handling capabilities
- Provides scope-specific asynchronous event handling capabilities allowing time based alerts, as well as out-of-band events such as order cancellation
- Scheduling of activities based on time and/or order of execution
- Uses Web Services as the model for process decomposition and assembly; that is each WS-BPEL process is a Web Service and can be composed as such in other WS-BPEL processes
- Uses XML and XPath for data access and manipulation

The BPEL language requires its activities to be specified using **WSDL**. This typically results in an exclusive relationship with Web Services; however Apache's **Web Service Invocation Framework** (WSIF) provides an approach for specifying other technologies using WSDL and thereby enables integration with WS-BPEL.

Human tasks and activities are not described by WS-BPEL despite the prevalence of human interactions in many real-world business processes. This gap is expected to be filled by BPEL4People which extends WS-BPEL to include orchestration of role-based human activities.

4.2.1 BPEL4People and WS-HumanTask

The OASIS WS-BPEL Extension for People (BPEL4People) is an extension of WS-BPEL for integration of human tasks and activities with automated business processes. It was first proposed in a joint white paper by IBM and SAP in 2005; in June 2007, Active Endpoints, Adobe, BEA, IBM, Oracle, and SAP responded by publishing the BPEL4People and WS-HumanTask specifications. As of February 2009 OASIS was working on standardizing the BPEL4People and WS-HumanTask specifications.

BPEL4People extends WS-BPEL to make provisions for human activities to interact as “services” (fundamentally Web Services). This is achieved through People-Links mirroring BPELs Partner-Links for automated activities. The human activity is defined by the WS-HumanTask specification. Methods for assigning and evaluating human tasks are beyond the scope of the BPEL4People specification.

The core extensions in BPEL4People are people activities and people links. A people activity is a new type of WS-BPEL activity used to define user interactions, i.e. tasks performed by a human user. Work items are created for each people activity and distributed to users within a role designated to perform them. People activities can have input and output variables and can specify deadlines. Within BPEL4People, tasks are defined to specify actions that must be taken by the user. The attributes of a task include priorities, deadlines, and description. Tasks are typically queried, claimed, revoked, and failed through some form of portal interface.

Within the context of a business process, BPEL4People

- supports role based interaction of people
- provides means of assigning users to generic human roles

- takes care to delegate ownership of a task to a person only
- supports scenario as
  - four eyes scenario
  - nomination
  - escalation
  - chained execution

The WS-HumanTask specification introduces the definition of human tasks and notifications, including their properties, behavior, and a set of operations used to manipulate human tasks. A coordination protocol is introduced in order to control autonomy and the life cycle of service-enabled human tasks in an interoperable manner.

4.3 Business Process Definition Metamodel (BPDM)

The BPDM is a metamodel specifying semantics for business process modeling, including modeling capabilities for orchestration and choreography. A metamodel is necessary to define a repository and interchange format for storage, maintenance, and exchange of business models.

The original intent of the OMG was to make the BPDM a pivotal specification for a repository from which any process modeling tool or process execution engine could retrieve a business process definition. Instead the BPMN 2.0 specification team consolidated the definition metamodel and formalized the execution semantics of the modeling notation circumventing the need for a separate metamodel. The BPDM 1.0 metamodel is now superseded by the metamodel definition included in the BPMN 2.0 specification.

The official position from the OMG on the BPDM states, “The OMG BPMN 2.0 final submission includes BPDM concepts. The BPMN 2.0 supersedes the BPDM specification.”

4.4 XML Process Definition Language (XPDL)

XPDL is a business process definition language from the Workflow Management Coalition (WfMC). Unlike BPDM, XPDL combines both graphical model representations and process execution language. Version 2.0 supports interchange with BPMN (version 1) models.

Since Oracle now fully supports the capabilities described by BPMN, BPEL, and Human Task standards, XPDL provides no additional benefit and will not be used by Oracle products going forward.

4.5 Business Rules

Separation and abstraction of business rules is a well established architectural principle and many tools are available to support it; however, few standards exist for this technology domain.

Business rules processing is often based on the Rete algorithm and although this is not a standard per se it is a well established approach to rules execution.

The OMG provides Semantics for Business Vocabulary and Business Rules (SBVR) which defines a vocabulary and rules for documenting the semantics of business vocabularies, business facts, and business rules. In addition it defines an XMI schema

for the interchange of business vocabularies and business rules among organizations and between software tools.

The Java Community Process (JCP) JSR 94 defines a Java interface for rules engines.

4.6 Web Services Transaction (WS-TX)

Web Services Transaction (WS-TX) is a collection of related specifications whose purpose is to define a set of protocols to coordinate the outcomes of distributed application actions. Transaction coordination is critical at various levels of service interactions. Strategies, such as the classic two phase commit, can be used for short-lived technical orchestrations; but more complex methods are often needed in long running business processes in which resources must be committed before the final outcome is known: in these circumstances compensating transactions may be required. WS-TX provides a framework to handle these different scenarios.

WS-TX specifies an extensible framework for developing coordination protocols through the Web Services Coordination specification, which in turn, is a framework used by two coordination types: atomic transaction (WS-AT) and business activity (WS-BA)

WS-Coordination enables scenarios in which a number of applications are required to reach consistent agreement in order to complete an operation. This form of coordination is critical to business process orchestrations.

The WS-Coordination framework enables applications to propagate an activity between services and to register for coordination protocols. The framework enables existing transaction processing, workflow, and other systems to hide their proprietary protocols and to operate in a heterogeneous environment.

WS-Coordination defines the structure required to propagate context between cooperating services; however it does not provide the coordination for transactions between services. It is only a coordination framework requiring other specifications, such as WS-Atomic Transaction and WS-BusinessActivity, for this purpose.

WS-AtomicTransactions provides agreement coordination protocols for the following transactional types:

- completion
- volatile two-phase commit
- durable two-phase commit

WS-AT is used in applications that require consistent agreement on the outcome of short-lived, distributed operations.

WS-Business Activity is used in applications that require consistent agreement on the outcome of long-running, distributed activities. In particular WS-BA provides compensating transaction support to the WS-BPEL specification.

5

BPM / SOA INTERLOCK

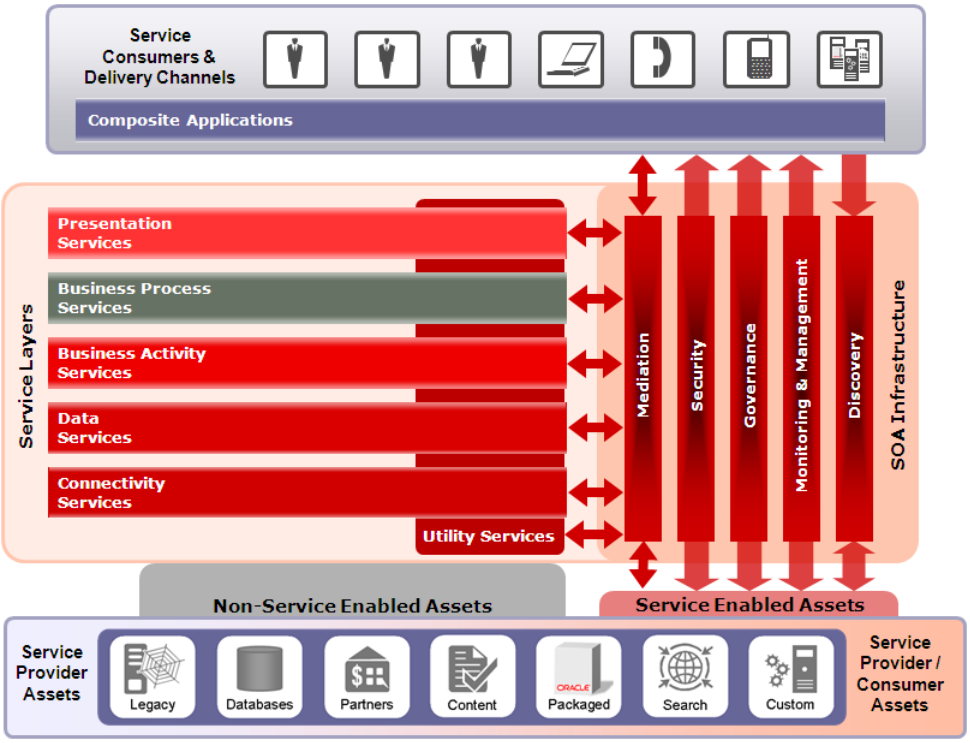
ORA consists of a number of core topics which include integration, security, application computing infrastructure, and monitoring and management. These topics are described in separate documents without respect to technical perspectives such as BPM. The relationships between BPM and other core topics are presented in the ***ORA BPM Infrastructure*** document.

Another technical perspective of ORA is Service-Oriented Architecture. The relationship between BPM and SOA is one of reinforcing benefits i.e. building BPM on an SOA foundation provides greater benefit than either BPM or SOA used in isolation.

5.1 SOA / BPM Interlock

The conceptual architecture for SOA is shown in Figure 5–1 below for comparison with the BPM architecture diagram.

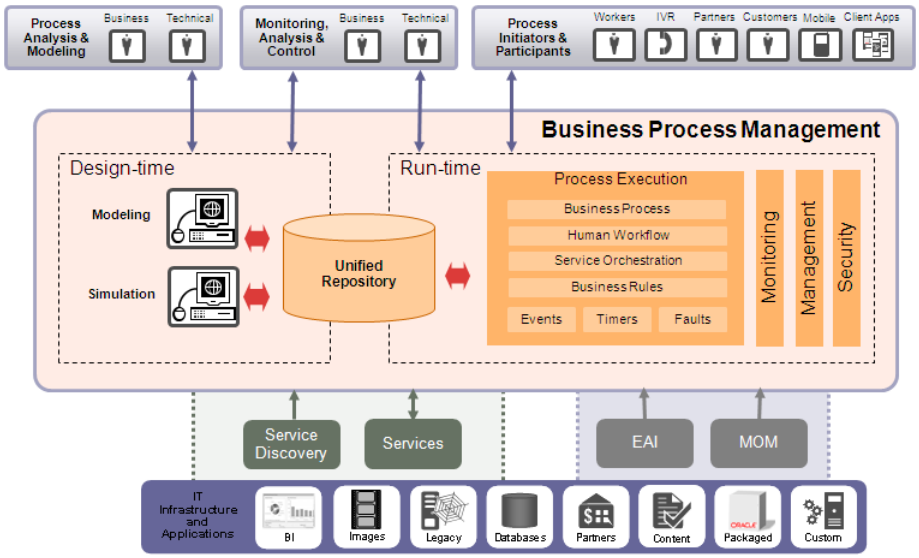
Figure 5–1 ORA SOA Conceptual Architecture



The providers and consumers at the top and bottom of the diagram are very similar to those appearing in the BPM conceptual architecture; for the most part they are the same applications, partners, users, etc. This is because they are the same participants. In the SOA diagram the top-bottom distinction arises to separate the pure consumers of SOA Services (top) from those equipped to interact as providers of business functions (or both provider and consumer when fully service enabled). There is not a hard line between these participants (this is after all a conceptual model), the separation is merely intended to show that there are final consumers benefiting from compositions versus pure system interactions (as in the case of B2B).

In the BPM conceptual architecture diagram (repeated in Figure 5–2 below for comparison) these participants are not anchored as firmly as in the SOA diagram; this is intended to indicate that the relationship with the BPM core is maintained by the underpinning SOA layer.

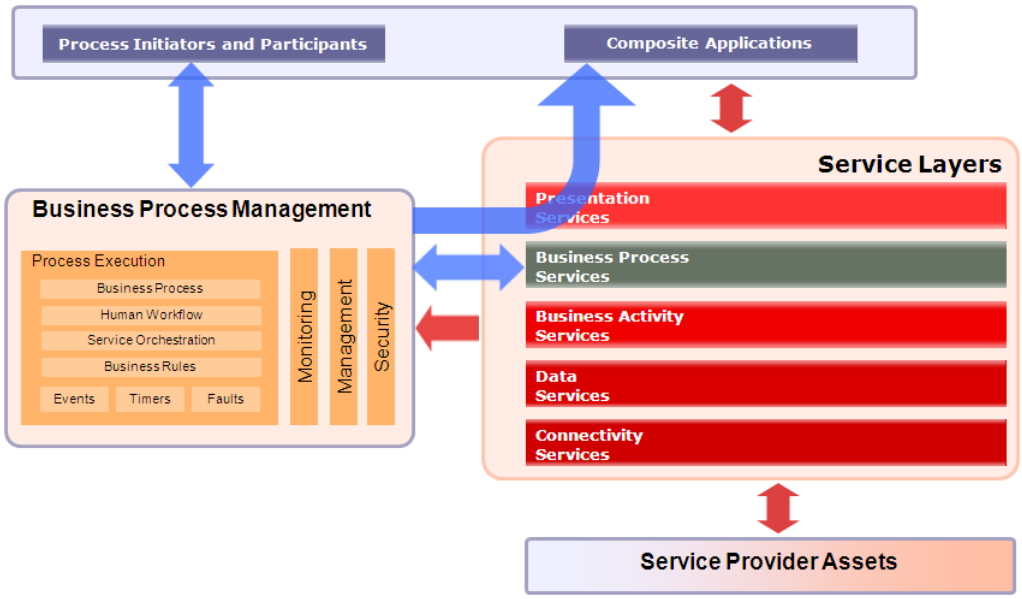
Figure 5–2 BPM Conceptual Architecture (repeated)



Here we see some minor differences in the participants also, for example, “search” is unlikely to provide (or consume) business functions in the BPM system. The human participants play a greater role in business processes, more like the interaction of the SOA Service enabled assets in the SOA diagram; but these human actors are clearly not IT assets and appear instead in the top participant container, while the bi-directional arrows represent their ongoing interaction.

Unlike the SOA conceptual model, BPM is incomplete without a representation of its design-time component, so for the sake of depicting what SOA and BPM offer to each other the two diagrams have been reduced to show only their related parts (see Figure 5–3 below).

Figure 5–3 ORA SOA / BPM Interlock



Here the blue arrows are used to emphasize the BPM contributions to the integrated model: the BPM process engine is directly providing composite applications while also providing and/or consuming business process SOA Services in the SOA model. The shared participants are shown here interacting directly in the business processes. The red arrows indicate the SOA relationships with its providers and consumers while also highlighting the consumption of SOA Services from any SOA layer by the BPM component.

5.2 BPM without SOA

SOA is not a prerequisite for doing BPM. Nor for that matter are Web Services; however BPEL, with its exclusive relationship with Web Services (notwithstanding other integration technologies, namely Apache WSIF) may fall into this category of "without SOA" since Web Services is not an automatic qualification for SOA.

Whatever approach is used for discovery, integration, security negotiation, message exchange and transformation, transaction support, etc. that would otherwise be provided by SOA, these aspects of the BPM infrastructure still need to be addressed. Most likely these will be tackled in a proprietary way, without the growing standards base that now underpins SOA.

5.3 BPM with SOA

BPM and SOA are two complimentary strategies. SOA provides the discipline of SOA Service description in business terms and enables discovery for use in business process modeling. It provides the infrastructure for mediated information exchange and integration between, application silos, business units, and partners. Today's BPM is uniquely positioned to take advantage of these capabilities and together BPM and SOA can go a long way in closing the business-IT gap.

5.4 SOA Extended to BPM

As documented in the ***ORA BPM Infrastructure*** document, SOA alone provides many of the capabilities needed for business process orchestration and business rules

abstraction. These capabilities are not the core benefits that BPM needs from SOA and they must be used judiciously. The SOA Service orchestration capabilities of the SOA mediation component is a useful feature that enables the technical specialist to wire together simple, rules-based flows, using graphical notation, producing declarative execution semantics. It does not however, enable a business user to seamlessly integrate his process model in standardized notation or collect business activity metrics against his graphical flows.

The difference is “technical orchestration” verses “business process orchestration” and while the SOA infrastructure is eminently suited to the former, enabling SOA Service composition and the construction of new business SOA Services, the latter must include the business specialist to realize the ultimate goal of closed-loop BPM.

That said, the “start with SOA” approach is still a good one (as is concurrent SOA and BPM). Some discipline is clearly required to avoid building too much complexity in the SOA Service workflows that may prove difficult to move to the eventual BPM system. Building the service infrastructure in advance does however, provide an ideal foundation for effective BPM.

5.5 Interlock with Other ORA Perspectives

In addition to the interlock between the BPM perspective and core SOA a number of the ORA perspectives are also closely related. In the case of BPM these perspective interlocks should be anticipated for Business Intelligence particularly with respect to BAM. Event Driven Architecture must be considered when designing business process event handlers and the more mature manifestations of BPM will involve intra-enterprise processes and should therefore, take account of the B2B architectural considerations.

A number of these perspective relationships exist and will be covered in detail in future releases of this document and other ORA materials, as the perspectives are developed.

6

Summary

The current “third wave” of BPM shows much promise in addressing the business-IT gap. SOA has addressed core technical problems, BRMS elevates business rules and policy from the realms of code, and the new modeling standard, BPMN 2.0, has set the stage for tools that seamlessly exchange models between stakeholders and execution systems, while incorporating the feedback of BAM in a zero-code closed loop.

This is an exciting time for BPM. We have come a long way from the days of having a fleet of business consultants write a recipe for a re-engineered business process only to find that it will take years for IT to catch-up. We've crossed a number of chasms along the way: from manual workflow supported by digitized media, to integration and workflow between siloed applications with EAI, finally to the BPM Suite with round-tripping between business specialists, systems engineers, and executing systems providing feedback.

The BPMS is still evolving and improving, but the core combination of BPM, SOA, BRMS, and BAM is the universally agreed direction. Given an appropriate infrastructure with a robust and consistent metadata repository, process and rules platforms, and suitable monitoring and analytics, the new wave of BPM is set to change the face of IT.

A

Further Reading

The ***IT Strategies From Oracle*** series contains a number of documents that offer insight and guidance on many aspects of technology. In particular, the following documents pertaining to ORA may be of interest:

ORA SOA Foundation - This document is suggested pre-reading for those wishing to get a deeper background to the SOA aspect of this document. It presents important basic concepts of SOA that are instrumental to building applications for a SOA environment. It covers topics including the components of a service, service layering, service types, the service model, composite applications, invocation patterns, and standards that apply to SOA.

ORA SOA Infrastructure - Infrastructure plays a key role in a successful enterprise SOA environment. The SOA Infrastructure document describes the role of infrastructure and the capabilities it provides. It offers an array of views to define infrastructure for SOA, including logical and physical views, as well as technology and product mapping.

ORA Integration - Many forms of integration exist today and play a key role in enterprise computing. The ORA Integration document examines the most popular and widely used forms of integration, putting them into perspective with current trends made possible by SOA standards and technologies. It offers guidance on how to integrate systems in the Oracle Fusion environment, bringing together modern techniques and legacy assets.

ORA Security - The ORA Security document describes important aspects of security including identity, role, and entitlement management, authentication, authorization, and auditing (AAA), and transport, message, and data security.

ORA Monitoring & Management - A common thread running through many applications, services, and systems is the ability to monitor and manage assets in a consistent and efficient manner. ORA Monitoring and Management offers a framework for OA&M to rationalize these capabilities and help optimize the operational aspects of enterprise computing.

In addition, the following materials and sources of information relevant to BPM may be useful:

State of the Business Process Management Market - this Oracle whitepaper discusses the results of an analysis of the current state of BPM in IT.

B

References

Reengineering the Corporation, a Manifesto for Business Revolution by Michael Hammer and James Champy (Harper Collins, 1993)

Pi-calculus: Communicating and Mobile Systems: the pi-Calculus, Milner (Cambridge University Press, 1999)

Business Process Management: the Third Wave, by Howard Smith and Peter Fingar

IEEE 1471 Recommended Practice for Architectural Description of Software-Intensive Systems

OASIS specifications

Web Services Business Process Execution Language (WS-BPEL)

- WS-BPEL TC Webpage, http://www.oasis-open.org/committees/tc_home.php
- WS-BPEL 2.0, http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf

BPEL4People

- WS-BPEL Extension for People (BPEL4People) 1.0, June 2007, http://www.oracle.com/technology/tech/standards/bpel4people/

Web Services Transaction

- WS-Transaction 1.1, OASIS, 12 July 2007, http://www.oasis-open.org/committees/ws-tx/
- Web Services Coordination (WS-Coordination) 1.2, OASIS Standard, February 2, 2009, http://docs.oasis-open.org/ws-tx/wscoor/2006/06

OMG

- Business Process Modeling Notation (BPMN), http://www.bpmn.org/

Glossary

The following BPM specific terms and abbreviations are included here for easy reference. Please see the ***ORA Master Glossary*** for other terms used in the various ORA documents.

2PC

see Two-Phase Commit.

ACID

ACID is an abbreviation for Atomic, Consistent, Isolated, and Durable which is used to refer to a type of managed transaction in application and database systems.

Business Process Analyst

A Business Process Analyst is an individual within an organization who defines, manages, or monitors Business Processes. They are usually distinguished from the IT specialists or programmers who implement the Business Process within a BPMS.

Business Process

A business process is a set of linked activities performed by people and systems that deliver some kind of business value through a product or service to internal or external customers.

Business Process Management (BPM)

A strategy for managing and improving the performance of a business through continuous optimization of business processes in a closed-loop cycle of modeling, execution, and measurement. BPM encompasses the discovery, design, and deployment of business processes. In addition, BPM includes the executive, administrative, and supervisory control of those processes.

BPMS

An abbreviation for Business Process Management System: also known as BPM Suite. Extending the classic modeling capabilities of early business process related activities the BPMS includes a collection of technologies to enable closed-loop BPM in a cycle of analysis, execution, and monitoring.

BRMS

An abbreviation for Business Rules Management System: extending originally specialized rules engines technologies to provide an accessible, centralized repository for business rules.

Choreography

Choreography describes what occurs between participants in a process. It is a representation of an interaction involving one or more message exchanges between two or more participants. Unlike orchestration, choreography has no central controller, responsible entity, or observer of the process. Choreography is useful for modeling trading partner interactions, forcing them to conform to a common interaction model.

Collaboration

Collaboration (in BPMN 2.0) describes the message flow interaction between a process and external entities.

Compensation or Compensating Transaction

The set of activities that are performed during the roll-back of a transaction (that was not managed by 2PC or other blocking transaction coordination mechanism) to compensate for activities that were performed during the execution of a process.

Functional Model

A tool to break down ("decompose") complex business functions into comprehensible and manageable parts; typically depicted by a hierarchy. Also known as Function Chart, Functional Decomposition Diagram.

Model-Driven Architecture (MDA)

A software design approach for the development of software systems. It provides a set of guidelines for the structuring of specifications, which are expressed as models. Model-driven architecture is a kind of domain engineering, and supports model-driven engineering of software systems. It was launched by the Object Management Group (OMG) in 2001. Under the OMG definition The Model-Driven Architecture approach defines system functionality using a platform-independent model (PIM) using an appropriate domain-specific language. Then, given a platform definition model (PDM) corresponding to CORBA, .NET, the Web, etc., the PIM is translated to one or more platform-specific models (PSMs) that computers can execute.

Participant

A Partner Entity (e.g., a company, company division, or a customer) or a Partner Role (e.g., a buyer or a seller), which controls or is responsible for a business process.

Private Process

A private Process is internal to a specific organization and is the type of process that has been generally called a workflow or BPM Process. Private processes may be executable or non-executable. In BPMN a single executable private Process will map to a single WS-BPEL document.

Process

Any activity performed within a company or organization to accomplish a specific objective. In BPMN a process is depicted as a network of flow objects, which are a set of other activities and the controls that sequence them.

Public Process

A public process represents the interactions between a private business process and another process or participant.

Transaction

A set of coordinated activities carried out by independent, loosely-coupled systems in accordance with a contractually defined business relationship. This coordination leads to an agreed, consistent, and verifiable outcome across all participants.

Two-Phase Commit (2PC)

A transaction management protocol that communicates with the participants in a transaction to direct agreement on readiness, commit, or roll-back operations.

Round-tripping

Round-tripping refers to the ability to pass artifacts seamlessly (and without loss of information) around a cyclic engineering process for the purpose of augmenting them with input from different disciplines at every step. Iterative software engineering practices made round-tripping particularly valuable. In the context of BPM round-tripping refers to the exchange of process models from business specialist to IT engineer and back again (potentially extending the cycle to include the execution engine which in-turn feeds back results of process monitoring), each time refining details of their individual concerns.

XA

A specification by the Open Group for distributed transaction coordination in software systems using the two-phase commit protocol. The XA Specification describes the function of a transaction resource manager in supporting transactional access to various resources.