System-of-Record Architecture for Process-Driven Solutions

Note: This whitepaper was originally written for Lombardi, prior to the IBM Acquisition of Lombardi in 2010. It has since been edited and updated to replace “Lombardi” with “IBM” and “Teamworks” with “IBM BPM”. Other than that, the contents of this white paper have not been altered. More importantly, the concepts and thought-leadership provided here-in by Fahad Osmani continue to be very valuable in today’s editions of IBM BPM.

(Superficial editing described above was done in 2012’3Q)
# Table of Contents

Table of Contents ............................................................................................................................ 2
Introduction ...................................................................................................................................... 3
A New Vocabulary ........................................................................................................................... 3
Process vs. Business Data – Where to draw the line?................................................................. 5
Business Data SOR Architecture Principles.................................................................................... 7
Typical Architecture of a Process-Centric Solution ........................................................................ 8
Evolving to the Recommended Solution Architecture ................................................................. 13
A Case Study .................................................................................................................................. 15
SOR Architecture Checklist ........................................................................................................... 16
For Additional Help ......................................................................................................................... 18
Introduction

Two key questions arise in each and every business process management (BPM) project: “What data is needed for this process application?”, followed by “What systems-of-record (SORs) master the data that’s needed?”

It may be surprising that sometimes there isn’t an official system-of-record for some of the critical data that is central to running the business. Other times, it’s no surprise – the lack of a system-of-record may be recognized as the root of the process problem!

Even when systems-of-record do exist and are well understood, process improvement applications will very often enrich or augment the business data that is accessed in the SORs. Where does this new data get stored?

Business Process Management System (BPMS) platforms like IBM BPM provide built-in data management support for process-driven solutions. Should the BPMS be used as a system-of-record instead of an existing system? Should a new system-of-record or repository be created? If so, under what circumstances?

This whitepaper provides some insights about the data requirements in typical process improvement applications, and recommends practical strategies for system-of-record architecture in these process-driven solutions.

A New Vocabulary

As new ideas and philosophies of process improvement are proposed, one can naturally draw upon traditional software architecture and design tenets to define and describe BPM solutions. However, a new vocabulary and design perspective can be helpful to articulate requirements, design and solutions that are uniquely process-centric.

By using exclusively traditional software design artifacts and vocabulary, there is often a disconnect between IT and business stakeholders regarding what is being articulated by BPM experts as the proposed solution and what is actually understood by traditional IT and business groups. For example, when one person refers to a ‘process application’, another person may automatically assume that the process application includes a ‘repository of business data’ related to the process. A new vocabulary and design perspective around process solutions would separate the process-application from the business data repository in a way that such distinctions would be clear early on, and any design implications resulting from such distinctions could be dealt with at an early stage of the implementation cycle.

Let’s define a few terms that will become the fundamental building blocks of our new vocabulary for BPM solutions.

Business Process

A business process coordinates various role-based tasks and activities in the pattern of a process. It can also coordinate system activities in support of these user tasks, quite often
coordinating and interfacing with traditional stack applications, acting in some sense as a stack coordinator for the purpose of completing a higher level business process. A business process involves asynchronous flow of work, incurring ‘dwell times’ during the tasks and activities due to the human participation in the business process. A business process deals with historical and transitional process data and can act as a capture/render medium for real time business data, although it is usually not meant to be the source repository of that business data.

**Transactional Application**

A *transactional application* is the master source of some or all of real time business data, and acts as the capture/render medium for this data as well. It is defined by its ability to provide ad-hoc and real-time access to this business data in a true “CRUD” capability, allowing its users to Create, Read, Update, and Delete any part of this data at any time. Normally, transactional applications implement “short” atomic transactions that complete (or rollback) quickly – so every access shows a “real-time” view of the business data.

**System of Record (SOR)**

A *System of Record (SOR)* is a data store, housing information in a structured fashion, allowing for retrieval and updates as needed for its purpose. In most cases a SOR is a database. In more modern SOA environments the data repository is accessed through an abstraction layer such as .NET or Java so that detailed knowledge of the underlying storage schema is not required. For the purposes of this paper the implementation of an SOR is less important than understanding its role in the solution.

**Process Data**

*Process Data* refers to the body of information pertaining to the basic lifecycle functions of any business process. An example of this sort of data would be things like Start Times, Wait Times, Key Performance Indicators (KPIs), Service Level Agreements (SLAs). The Process Data SOR is intrinsic to the BPMS system. This data can be used to improve the business process both when it is viewed in aggregate, and when the specific details of one instance of the process are viewed to gain specific insights.

While the values of the process data will differ for each solution, the types of data collected for a process should be the same across all process solutions.

**Business Data**

*Business Data* refers to any information related to the detailed attributes of the specific business process. BPMS components (tasks, business process model, KPIs, SLAs, escalations etc) access and manipulate this data, but it is not intrinsic to the BPMS. Examples would include things like Incident Resolution Time, Type of Issue, Name of Originator, etc. in a process such as Trouble Ticket Management. The Business Data SOR should be distinct and disconnected from the Process Data SOR as it will need to support various other consumers and subscribers to its data. This data is generally used to determine the specific state of a particular instance of a process.
While some processes may use similar business data it is much more common that each process will have its own set of data collected as part of the process interactions.

**Instance Reporting**

This division of Business and Process Data is often at odds with requirements from the users of a process solution. Frequently what is requested is an unfiltered view into the live business data of the solution, overlaid with the process data, quite often with search filters to allow for selective retrieval. These types of requirements will be referred to as *Instance Reporting*.

The point of such views and interfaces is to allow the user to view and perhaps even update business data along side with the workflow controls and tasks presented by the BPMS. The tactical nature of these requirements can often lead an inexperienced or undisciplined implementation team to blend together the SORs for Process Data and Business Data. Such an approach is not in line with a robust, long term strategy for process delivery.

**Trend Reporting**

In order to build trend-level views of how each business data attribute varies for an activity across a large number of process occurrences, graphical reports and dashboards are needed to visualize the historical data so that the trends can clearly be seen over measurement periods. We shall call these artifacts *Trend Reporting*.

**Process vs. Business Data – Where to draw the line?**

For the purposes of this paper, Process Data can be thought of as data that can be captured for any business process independent of the specific business case the process is designed to solve. It is data that a person familiar with business processes can understand. It does not require process specific knowledge to be interpreted.

Business data is specific to a particular process or set of processes, and may not have much meaning to someone unfamiliar with the details of the process. It is important to note that many BPM solutions require a blending of both types of data to meet their visibility requirements.

**Why Keep Process Data Distinct from Business Data?**

Most businesses have a large number of internal and external business processes that are mission critical. In order to improve the business, the efficiency and quality of these processes should be continually improved.

One of the first things needed to determine the most important process to focus on first, out of all the possible processes, is to develop a consistent view/vocabulary around the processes and the overall solutions that support them so that consistent metrics and indicators can be applied across the board.
Collecting similar data across all implemented processes helps enforce this abstraction. This process data is generally collected by the process application and consumed by the following audiences:

- People participating in the process
- People managing the process participants
- People evaluating the performance of the process

Since individuals may fall into each of these 3 categories for different parts of various processes it is useful to have a common view of process data. This allows a person to understand what is meant when talking about process performance even when processes differ widely. Keeping this data separate from the business data avoids “burying” these useful commonalities within solution-specific data. At the same time, it allows the solution to hide the solution-specific data from individuals and systems that would only be confused by its presence.

It is also important to note that Process Data is almost always historical. This is a record of what happened in the process over time. It is rarely a real-time view of the data.

Why Keep Business Data Distinct from Process Data?

Business data is frequently of use to individuals and systems not directly involved in the process itself. It is data that is collected while the process is being performed. Separating this business data from the process data provides several advantages.

Business/Strategic Advantages

- Disambiguation from generic process data.
  The most important advantage is that it separates the business data from the process from which it was collected. In general, processes change much more rapidly than the data the process acts upon. By nature, business processes are largely sequential, one activity followed by another. Each activity exposes the potential to view or change business data. By having separate SORs for process and business data we eliminate the need for the business data to change or be reorganized as the process is modified and matures.

- Audit trail.
  Versioning business data and documents over time is a specialized capability that has its own tools and requirements. By building a separate SOR for business data, greater flexibility is attained in selecting the design and tools that will best address these requirements without being bound to the BPMS system.

- Expose the business data to external (non-BPMS) systems.
  An additional advantage is that a separate SOR for this business data can be used by a wide variety of data consumers (including the process itself). And the separate SOR can
be optimized for how the data will be accessed and used. As most Business Data is transactional this can be taken into account in the SOR’s design. Often, an SOR pre-exists and is already exposed to such external systems.

Technical/Tactical Advantages
Finally, storing the Business Data in a separate SOR allows for several technical advantages such as the following:

- Permits ad-hoc changes to the data by both systems and individuals if required.
- Allows the use of reporting tools and mechanisms independent of the process solution.
- Permits correct level of access to the data by non-process participants according to legal/technical requirements.
- Separates implementation concerns (such as data locking) from the process itself.

It is important to note that Business Data is almost always available via a real-time view. It should reflect the values that are available to all process participants and any other data users at the current time.

Business Data SOR Architecture Principles
The very nature of a Business Data SOR is that it is transactional as opposed to historic. On the other hand, a process application creates an historic archive of Process Data.

The transactional nature of the SOR can tempt developers and business owners to adopt classic application design patterns that do not fit well with a process system. A developer may feel ‘entitled’ to such typical features as the ability to insert, update, or delete any business data at any time and perform ad-hoc queries against all live data. This is powerful and dangerous in the face of a guided and sequential process system.

The SOR should always be built or validated by someone in a Data Architect role. A Data Architect will use appropriate standards-based tools to design and implement a Business SOR based upon business process requirements. They will provide standards-based CRUD interfaces to business data objects.

The inclusion of a SOR relies on some basic design assumptions.

1. What happens when the SOR evolves? The assumption here is that a stable API hides these changes from the process.
2. An SOR introduces synchronization issues. The process system must reliably load and save the business data at the start and end of each activity, as well as at any point where the user closes the activity for later completion.
3. An SOR also introduces inconsistency issues. Since other applications can change the data it is possible that critical business process data may not exist at the time of task execution (an order could be canceled, a line item modified, a part back-ordered). A process system must reliably lock the business data that it has read out of the SOR until
it saves it again (at the end of the activity, or when suspending and postponing the activity, etc)

4. By nature, BPM and process applications are sequential. The existence of SORs can lead to 'change anywhere/anytime' solutions (traditional web applications). These two are contradictory to each other and the process system must always be used for process-centric solutions that have a pattern and a life cycle with a defined beginning and an end.

Typical Architecture of a Process-Centric Solution

Process driven projects are usually driven by upfront analysis, followed by implementation work via process-applications, with business data captured along the way ending up in some business system-of-record. A high level view of such a project might look as follows.

At a more detailed level, the typical process-centric solution consists of the major component-groups (shown below), not all of which need to be developed with process technologies (as shown by the BPMS border). In the sub-sections below, we shall examine each of these component-groups in greater detail.
Typical Process Project

**Analysis, Definition**

**Process Application #1**

- **Report Screens**
  (see 4.2 Custom User Interfaces for Process)

- **Process Screens**

- **Workflow Logic**

- **Process Data Definition**
  (see 4.1 Process Components)

- **Business Data Definition**
  (see 4.1.3 Integrations for Process Logic and UIs)

**BPMS**

- **Process SOR Data**

- **User Registry**

- **Hybrid B/SOR Data**
  (see 4.3 Business Data SORU)

- **Admin Screens**
  (see 4.4 Custom User Interfaces for SORU)

**Legacy DBMS**
Process Workflow Components

The process workflow components consist of the following items:

**Process Logic**
These artifacts consist of process models and workflow diagrams, decision gateways and other control structures that model a business process along with all its lifecycle control points. Each BPMS system provides notation and graphical objects to map out these activities and the controlling sequential flow of logic between them. This is squarely within the realm of BPMS technology and attempting to do this with generic tools when process is available would be akin to reinventing the wheel.

**Process UIs**
These are subcomponents of the artifacts in Process Logic. At various significant points in the lifecycle of business process, human interaction is required with the solution. This interaction is modeled in one or more UIs which are connected directly to the Process Logic as individual activities and tasks. These UIs can either be built directly and internally within the BPMS system or externally invoked, but are still tied to the Process Logic.

**Integrations for Process Logic and UIs**
Both the Process Logic and Process UI artifacts will sometimes need information from systems external to the process software in order to drive them forward. In some cases,
this information is for view purposes only, and in others, it is meant for distinct and targeted updates (as opposed to ad-hoc and all-encompassing view/updates, which, as we shall see, are an entirely different component-group in the overall process-centric solution).

Custom User Interfaces for Process

These components are non-workflow related UIs which are to be used by process workflow participants. They are used for on-demand viewing (but not editing) of process and business information, some of which may come from the BPMS system (process data) and some of which may come from the Business Data SOR (business data).

Since this view is to be used by participants in the workflow, it needs to be tied into the workflow UI framework, restricted by the same rules under which the workflow UIs must operate (sequential and pre-patterned flow of information) and thus cannot be used for ad-hoc data maintenance.

An essential point here is that there should be no process-data edit/update capability in these UIs, otherwise the native workflow design and capability – which is relying on being the master of this information – will not be able to function correctly.
An example of these UIs can be search screens custom-built to be deployed via the process workflow solution that are to be used to search for an view details on individual streams of work.

Business Data SOR

There are two main component groups here. The Business Data SOR is one.

The other is all the access points and integrations to this Business Data SOR in the form of generic integrations such as the ones shown below.
Custom User Interfaces for SOR

These components are non-workflow related UIs to be used by users other than workflow participants. They are used for on-demand viewing and updating of information in the Business Data SOR.

Since these views are not used by participants in the process workflow, they do not need to be tied into the workflow UI framework, nor do they need to be restricted by the same rules under which the workflow UIs must operate (sequential and pre-patterned flow of information) and can be used for ad-hoc data maintenance.

An essential point here is that the Business Data SOR should not contain any Process Data. The BPMS system is designed and optimized to be the master of Process Data. Ad-hoc editing of process-data fundamentally undercuts the notion of an automated process solution.

Evolving to the Recommended Solution Architecture

Now that we have our terminology straight and basic building blocks laid out, how do we move forward? A company may have an existing solution to address a business-process need. This existing solution might not fulfill any of the requirements for actually measuring how well it is solving the process issue, or may not be flexible enough to keep up with the evolving needs of the process, or may fall short for any number of reasons that can all be boiled down to, ‘it is not a process application’.

There are a number of as-is scenarios to consider.

Legacy Business Data SOR Exists

In this scenario, the process-driven project is going to interact with a legacy system-of-record for business data. This legacy system may have:

1. Outdated schema that does not address the needs of the new process-driven solution
2. Outdated interfaces for access (e.g., file transfers or “green screens”)
3. Outdated system of implementation (e.g., mainframe-based)

In order to evolve to a process-driven solution from such an existing scenario, we can consider multiple approaches:

**Rip and replace**

In this approach, the existing SOR is targeted for a sunset event that coincides with the launch of the new process-driven solution. All the existing functionality of the legacy system, including storage, access mechanisms, housekeeping and SLA with consumer-applications must be replicated in the new SOR. The new SOR must also incorporate the need for new business data and access to such data as part of the process-driven solution.

All traditional best-practices and precautions implied with sunsetting a legacy application are pertinent here.

**In-Place Upgrade**

The existing SOR is upgraded to comply with any additional data it needs to capture and provide without changing the underlying implementation or data-access architecture behind it.

**Hybrid**

The existing SOR is left completely unchanged in terms of the data it captures and the systems it serves. Any additional requirements surfaced by the process-driven solution are placed in a new system-of-record with its own access layer and is meant to exist side-by-side with the legacy SOR.

**No Business Data SOR Exists**

In this scenario, the process driven solution needs to interact with an entirely new business data SOR that has not existed previously in the host environment. This business data has not been captured in any electronic format before by the company and the process is being automated for the first time.

In order to evolve to a process-driven solution from such an existing scenario, we need to take into account more than just the needs of the specific process-driven solution at hand. This Business Data SOR might well serve as the central repository of business data for various other non-process-driven applications and in addition, might actually be used as the SOR for additional process-driven solutions.

A fine line must be walked here in terms of avoiding a very specific design for the SOR that narrowly serves the process-driven solution only, or an over-engineered design that allows for future functionality that may never happen.
A Case Study

A representative example of a customer deployment that characterizes the correct use of the IBM BPM-recommended architecture is one of the world’s leading event management / travel companies.

In this example, the customer authored their Sales and Fulfillment process for which there was no single system-of-record – instead there were multiple systems that mastered data. IBM BPM was used to automate what had been a manual process, and was expected to fill gaps that existed in the infrastructure. There was little concept of process data. A best practice that was not followed in the deployment was creating reports during the process authoring. The reports were written subsequent to the deployment, which also necessitated some data analysis to determine the best alignment of certain source data after-the-fact.

Once the process had been deployed, the customer’s situation was characterized by the following patterns:

1. Business data housed in existing Systems-of-Record continued to be reported from those systems. The IBM BPM application was implemented such that information would be written back into the legacy architecture to drive various business reporting.

2. Process visibility (current state of tasks, ownership, actions) was established by creating a dashboard in the IBM BPM application that aligned to the customer’s organizational hierarchy. This dashboard allowed managers and individual contributors to gain insight about the current state of their process, and compare that real-time information to historical data. Implementation of the dashboard required additional data processing to translate legacy process data (data collected prior to the dashboard creation) into the manager's view of the process lifecycle. (This was necessary because best practices were not followed at the outset of the project.)

3. Process performance data (cycle time, throughput) was made visible via both MS Excel and customer scoreboards. In this example, cycle time and throughput metrics are visible in a performance scoreboard and the Lean Sigma group accesses the performance data in order to analyze aspects of control and variability in the process.

Key decisions that the customer made that influenced the above decisions:

- IBM BPM is the system of record for some of the business data, all of the process data, and all of the performance data because of long term plans to sunset legacy systems.

- The existing SORs were not modified to handle newly identified business data required by the process implementation. This required that IBM BPM become the SOR for the data that fell in this gap. If new SORs were created or the existing ones modified to handle this data, those systems could have mastered all of the business data.

- The customer has a large investment in the BI tool from Business Objects, and is leveraging the data in IBM BPM and other systems for BI reports. Only the process visibility reporting has been authored in IBM BPM.
The customer has long term plans to drive all historical data into a data warehouse, so that there is better access to this measurement information.

Without rehashing the benefits already outlined in the whitepaper, the customer was helped significantly with the efficiencies and alignments created by this architecture. The flexibility to change legacy systems, leverage existing reporting technologies, and obtaining quick access to process and performance reporting are but a few of the advantages.

**SOR Architecture Checklist**

The following are some basic questions you can pose for your project to see if your existing architecture is line with BPM SOR best practices.

1. Does your BPM solution utilize a SOR?
   - Every BPM solution should have identifiable Systems-of-Record.

2. Is the Business Data SOR different from the Process Data SOR?
   - The Business Data SOR should be different from the Process Data SOR.

3. Is the Process Application in sole control of the Process Data SOR?
   - The process application and BPMS software should be the sole entity in control of the Process Data SOR.

4. Is the Process Application in sole control of the Business Data SOR?
   - The process application should NOT be the only controlling entity of the Business Data SOR. The B-SOR is meant to have any number of consumers, all brokered by the B-SOR’s transactional view/update mechanism.

5. Is there a built-in transactional update mechanism around the Business Data SOR?
   - Ideally, the B-SOR should have its own transactional view/update mechanism (stored procedures, views, functions or a set of Web Services, etc). Putting this responsibility in the hands of the first consumer of the B-SOR (typically the process application) causes a confusing mix of domains and best-fit.

6. Is the process application in charge of maintaining transactional integrity around the Business Data SOR?
   - This should NOT be the case. While the process application is capable of maintaining the B-SOR’s transactional integrity for its own accesses, it cannot possibly anticipate all other future consumers of the B-SOR and their transactional needs. A dedicated and self-contained broker is the best practice for this.
7. Is the Business Data SOR’s transactional logic self-contained and maintainable?
   - All transactional logic implemented by the designated access broker to the B-SOR should be easily maintainable and more importantly, self-contained. It should not be implemented by a solution that fulfills this as a secondary function.

8. Is the business logic mixed with the transactional logic?
   - The business logic of the process solution should be kept DISTINCT and SEPARATE from the transactional logic used to maintain the B-SOR.

9. Are there non-process-centric views available into the Business Data SOR?
   - Ideally, there should be more than one view into the B-SOR aside from whatever screens in the process application utilize it. The reason for this is that non-participants in the process will want access to this information and the increased number of B-SOR consumers will improve the transactional integrity of the B-SOR view/update broker.
For Additional Help

The IBM BPM homepage is http://ibm.com/bpm

Live-recorded YouTube Tutorial Demos can be found here: http://IBMBPM Demos.com

The following free PDF eBook is a great end-2-end reference guide for beginners and veterans, found here: http://www.redbooks.ibm.com/abstracts/sg247973.html?Open