Integrated Operational Reporting in BI - A Process Oriented View

Applies to:

SAP BI 3.5 and above

For more information, visit the Business Intelligence homepage.

Summary

The white paper discusses the concept of an integrated operational reporting in BI that supports cross-functional integration along a process. It uses two key processes in the organization - Procure to Pay and Order to Cash to illustrate the concept and discusses the critical success factors for this approach.

Author: Anand Jha

Company: Deloitte Consulting

Created on: 03 April 2009

Author Bio

Anand Jha is an experienced Sr. Consultant with Deloitte Consulting based out of the San Francisco office. He has a rich set of experiences in the Business Intelligence domain.He holds a bachelors degree in engineering from Delhi University and a Masters in Management Science from Stanford University.

Table of Contents

Introduction

Operational reporting in BI is gathering momentum and many companies have incorporated it as part of their BI strategy. While many important criteria are considered when designing an operational reporting architecture, an often missed point is the consideration of cross-functional integration in the design. As a result, users have to spend a significant amount of time in manual integration thereby reducing their productivity The white paper discusses the concept of an integrated operational architecture in BI that supports cross-functional integration along a process. It uses two key processes in the organization - Procure to Pay and Order to Cash to illustrate the concept and discusses the critical success factors for this approach.

Operational Reporting in BI

Today, many companies are adopting SAP BI as the primary platform for their operational reporting needs; a shift from the traditional use of BI for strategic and tactical decision making that catered to the executive and middle management. Operational BI demands near real time data, fast response times, good formatted reporting capabilities and flexibility to meet the dynamic business needs. Recent trends such as SAP's innovation in near real time data extraction technologies, the SAP BI accelerator and the Business Objects acquisition are all geared to meet these demands and support the evolution of BI for operational reporting.

A seldom discussed but equally important requirement for operational reporting is the functional integration between different subject areas. A transactional system, such as SAP ECC, provides seamless integration for document flows in a process. Users can navigate from one document to another document in the flow effortlessly. An unmet need however, is the ability to do this navigation for a list of documents. For example, if the user has identified a list of purchase orders based on certain criteria and would like to view all the goods receipt documents for these purchase orders, it will require them to do so one at a time, consuming a lot of time. Typically custom reports are built in the transactional system to cater to these needs. When moving to BI, these integrated custom reports can provide significant challenges if the integration points and the solution are not given considerable thought in the design phase. Organizations that are building or enhancing their operational BI reporting strategy, should consider the cross-functional integration as a critical aspect in their BI design.

Need for a Process Oriented View

Why do we need to take a process oriented view for building the integrated solution? Typically, BI requirement workshops and architecture designs are segregated by subject areas such as Finance, Procurement, Sales and Distribution and Inventory Management. While subject area focused design is robust in its silo, it often misses the integration points with preceding and subsequent subject areas in the process flow. A focus on requirements and design along the entire process is necessary to ensure that the integration points are taken into consideration and included. The integration points need to be called out explicitly in the design and agreed to by the BI subject area designers and the ECC process designers. Depending on the nature of the document relationships, several integrations options may exist. The key is to think about the various alternatives, bring them to the table and decide on the optimal one for each relationship.

Procure to Pay Process – The Integration Points

The procure-to-pay process usually begins with a user creating a purchase requisition. Depending upon the business rules and approval processes of the purchasing organization, purchase requisitions are subsequently converted to Purchase Orders. Purchase orders and purchase requisitions integrate in both directions at a document and line item level and the bi-directional integration is helpful for users to drive the reporting from either direction. An example of such a scenario is to identify the list of purchase requisitions which were converted to purchase orders and see the purchase requisition and purchase order in a list report with details from both the documents in a single line. The single line reporting can be achieved by having a concatenated characteristic in both the models which can then be leveraged in the front-end for integration purposes using either v-lookups or report to report interfaces.

Once the purchase order is created, goods or services are received against it and subsequently invoice receipts are created. The quantities and amount of goods and invoice receipts are present in the purchasing data model as part of standard content; however the actual material document numbers and invoice receipt numbers are not. The document level integration needs to be considered. Depending on the specific business rules, several options may exist. One option is to bring the purchase order number and line item onto the material document number when designing the inventory material movement subject area. Inventory management as a subject area has standard content for summarized stock reporting by various categories. The document level integration for bringing in the purchase order number will involve designing a separate data model that houses the material document and line item details. Similarly, the purchase order number and line item can be built onto the invoice receipts model. Once the purchase order and line item number is present in the material movement and invoice receipts model, the reporting integration, depending on the scenario, can be handled in the frontend. An example is the ability to list a set of purchase orders based on certain exceptions such as purchase orders with invoice receipts but no goods receipts and list the corresponding invoice receipts. This allows identification of parked invoices which fail the three way match because goods receipt did not take place. This reporting scenario can be achieved by having a concatenated purchase order and line item characteristic in both the models which can then be leveraged in the front-end for integration purposes, using either v-lookups or report to report interfaces.

Invoice receipts once approved posts to the general ledger. The general ledger invoice document has several integration points in this process – invoice receipt, goods receipt/material document and purchase orders. These integration points should be incorporated in the general ledger model design. In addition to the above integration, the general ledger invoice document has a corresponding clearing document which contains the payment information which needs to be considered for the final piece in the overall integration. Another layer of complexity in this final piece is cross company code transactions where integration has to be handled between the invoice and the clearing document, which belong to separate company codes.

Let's consider an end to end reporting scenario in which the user wishes to see purchase orders with goods and invoice receipts details and the payment details for the vendor. We will assume it's a cross company code transaction in which the payment company code is different from the purchasing company code. In this scenario, the report needs to source its data primarily from three subject areas- purchasing, general ledger and payables. The purchasing and general ledger data model is linked by the purchase order and line item, and will provide the order and receipt details. The general ledger invoice and the clearing document are linked by the cross-company code number to the payables subject area for payment details. Data from all the three areas can be integrated in the frontend using either v-lookups or report to report interfaces.

Order to Cash Process – The Integration Points

ORDER TO CASH

Sales Order | Delivery | Inventory Movement | Billing | General Ledger | Account Receivable
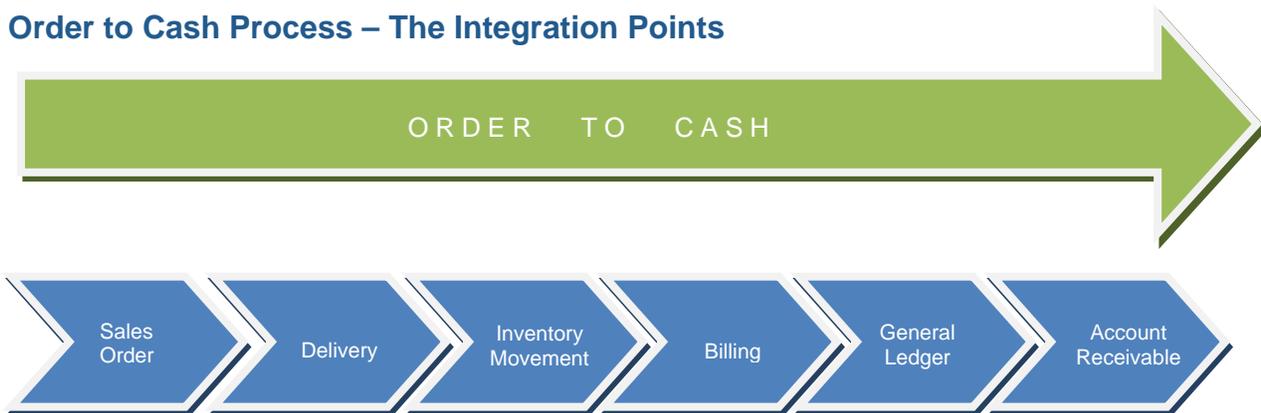
The creation of a sales order triggers the order-to-cash process. In some businesses, quotations may precede orders hence you may want to consider quotes if applicable in your scenario. Quotes and Orders integrate in both directions analogous to the purchase requisition and purchase order relationship. Since sales orders can result in multiple deliveries and goods issue documents, the integration with deliveries and goods issues are best handled in the subsequent documents.

Delivery documents can be enhanced with both the sales order and the goods issue document on it. The bi-directional integration provides the ability to see the details of sales orders and material documents that are not present on the delivery information model. The goods issue document integrates with the delivery document and the sales order number and this integration should be handled in the material movements model, which as noted earlier, will be a custom data model housing the material document details. A common reporting requirement is to list the sales orders with the deliveries and goods issue document with details from each document. The integration point is the concatenated characteristic of sales order number and line item. Details from the respective documents can be integrated in the front-end using v-lookups or report to report interfaces.

The billing document which is generated after the goods issue is done, integrates with the sales order number and line item as well as with the goods issue document. Billing documents, unless blocked, post to the general ledger as accounting documents. The general ledger invoice document has several integration points in this process – billing, goods issue/material document and potentially the sales order number. These integration points should be considered in the model design. In addition to the above integration, the general ledger invoice document has a corresponding clearing document which contains the customer payment information which needs to be considered for the final piece in the overall integration. Similar to payables, consider cross company code transactions where integration has to be handled between the invoice and the clearing document that belong to separate company codes.

Critical Success Factors

The BI Integration Lead	From a project management perspective, all the subject area designers should be coordinating with each other and have a high level understanding of the entire process and the integration points. One of the functional designers should have the specific responsibility of an integration lead and ensure that the integration points are present with consistent metadata definitions in all subject areas. The integration lead also needs to call out all the components where the integration is built in the backend data model and where it is expected to be carried out in the front end.
Keep them involved: The business user and the process subject matter expert	While the business users are part of the requirement phase and sign-off, they often come into the picture only during quality testing or user acceptance testing. It is critical to validate some of the key integration scenarios using playback sessions with business users and ECC process designers when in development to bring out any unknowns. This also helps build confidence in the integration logic. Any changes to the design happen early on during the development as opposed to quality or user acceptance testing phase. Hold as many sessions as you can to validate!
Keep an eye out for Performance	Since the integration happens at a line item level in most cases, the individual subject areas will tend to have very high granularity leading to performance impacts. The architecture design can mitigate these impacts by incorporating logical partitioning depending on data volumes by subject areas. Layering the architecture for summarized and operational reporting is another way to minimize the impact on users of summary reports. Due to the presence of many document level details especially in the general ledger, dimension cardinality and the load performance should be tested thoroughly.
Passing on the Baton - Knowledge Transfer	You have built it and delivered it, but if you do not have the right client resources for knowledge transfer, the solution will not be sustainable. Identifying the right resource early on is critical so that the knowledge transfer happens throughout the project lifecycle. The client resource will be a peer to the BI integration lead who understands not only the information per subject area but its relationship across all the areas in the process.

Conclusion

In summary, cross-functional integrated reporting in an important requirement for operational users, though it is often missed upfront in the design and comes up only in the final stages of a BI implementation. Take a pro-active approach to address it using a process view to design since it enables a forcing mechanism for addressing all the possible integration points. While we have discussed two of the most common operational processes, the principle can be applied to any end to end process. Keep the critical success factors in mind as you use this approach and you will have a set of happy users at the end of your engagement.

Related Content

For more information, visit the [Business Intelligence homepage](#).

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.