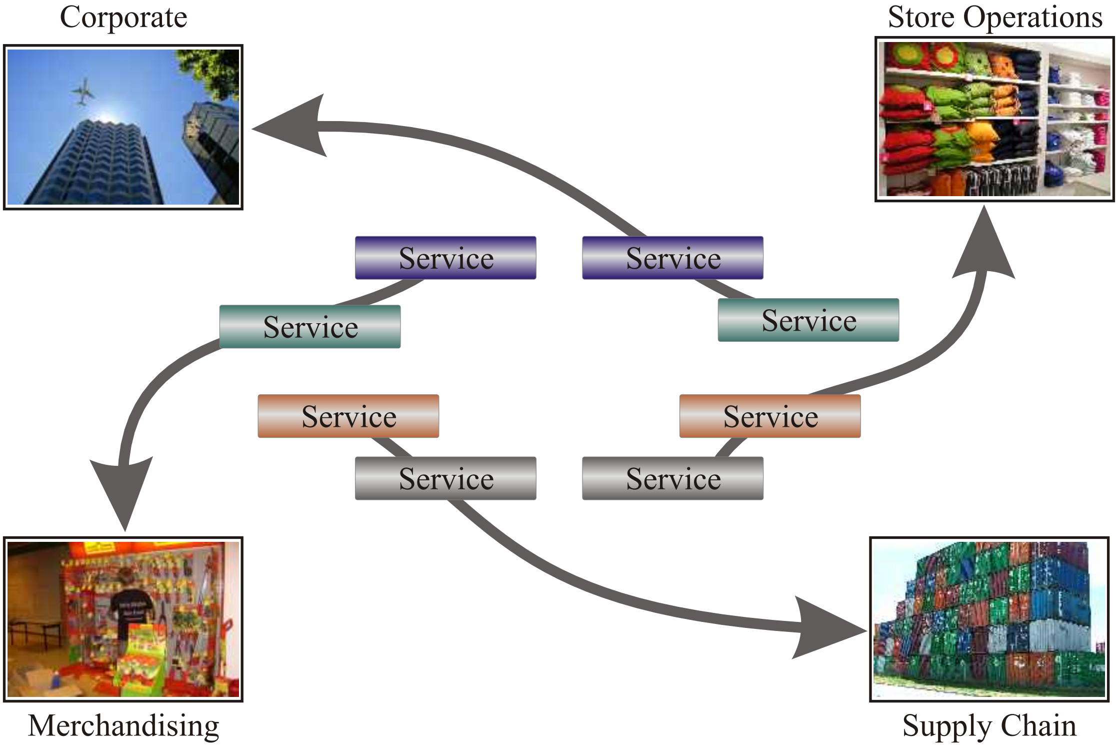
Torry Harris

SOA

SOA in Retail



Corporate

Store Operations

Service

Service

Service

Service

Service

Service

Service

Service

Merchandising

Supply Chain

Table of Contents

Introduction

The retail industry is marked by four significant characteristics

- A high frequency of change in management reporting and service offerings to meet competitive needs posed by increasingly larger purchasing avenues available to the consumer, ranging from internet purchasing, price comparison tools, to speciality serviced stores
- Inventory control and management requiring automated communication in real time between a large number of vendors spread across the world and multiple points of sale
- While retail environments within an enterprise have certain common features, the need to distinguish them by offering store to store variations based on the needs of the community which a store serves is becoming increasingly common.
- The need to provide an integrated, consistent cross-channel experience to consumers

These requirements stress the constitution & response times of the IT environment, in terms of the time available to modify the technical environment to cope with the changes introduced by operations that are bound by competitive pressures and seasonal requirements. The disparate standards maintained by a wide vendor community with whom retail has to maintain a real time data exchange, adds to the challenge. The need to construct individual solutions at the local level to meet the needs of a community, while using corporate assets and compliance with the larger enterprise norms, demands re-use of technical assets together with customisation at the point of sale.

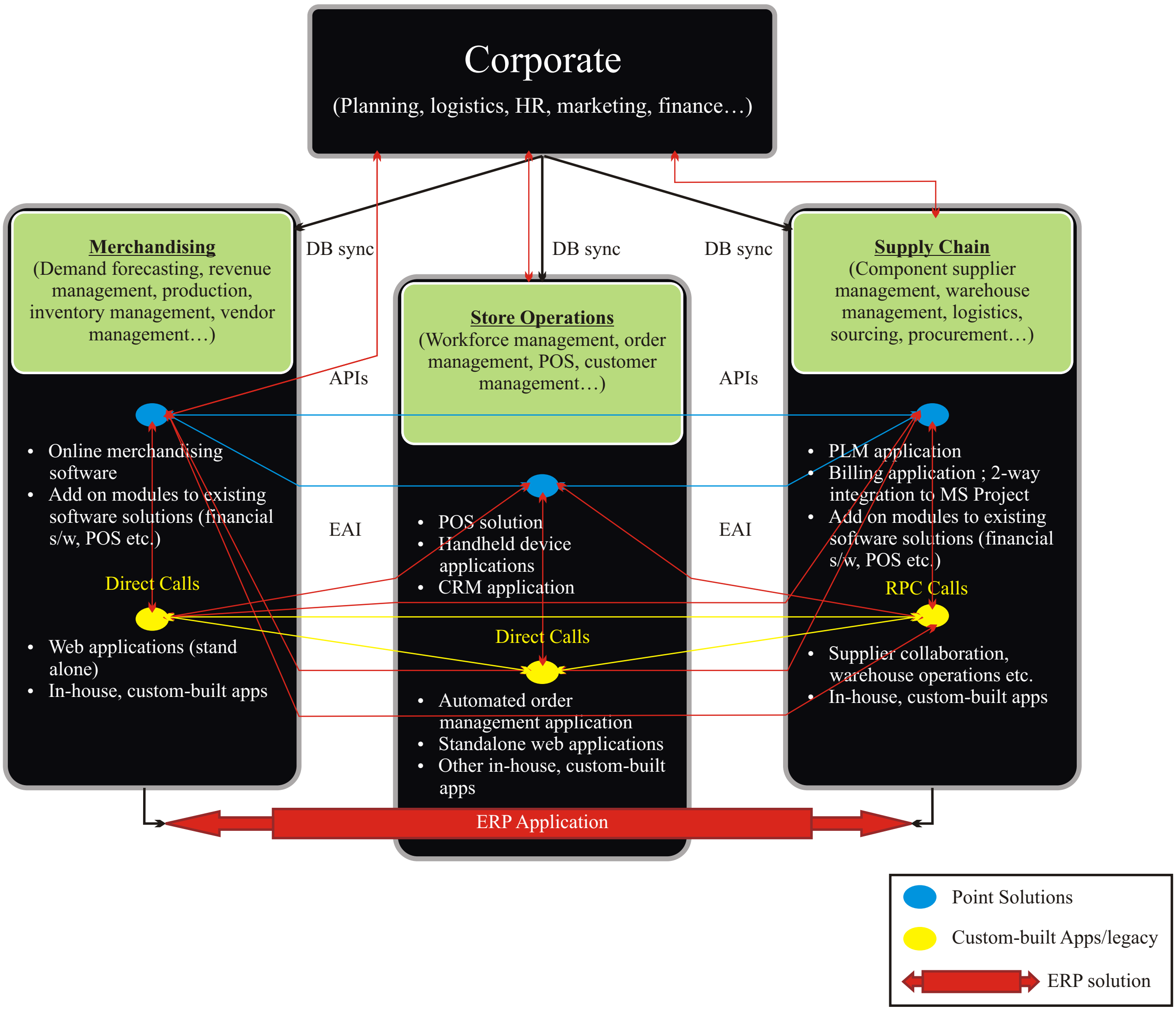
SOA provides such methodology to enable an integrated interaction platform in a flexible yet cost-effective manner, allowing incremental lightweight changes, replacing the big-bang massive overhauling approaches. The approach allows better leveraging of the information already contained in existing systems, thereby adding value to investments. Technology can be deployed incrementally through applications created by combining multiple services (well-bounded sets of functionality fulfilling a business need) through SOA. It enables cost savings and the flexibility to change quickly, that stem from consolidating redundant logic and data. According to Forrester Research analyst Randy Heffner in his February 2007 report titled ‘Planned SOA Usage Grows Faster Than Actual SOA Usage’, flexibility and costs are not the only drivers for SOA. The drivers for SOA adoption include easier application and cross-platform integration (including access to legacy applications), improved software quality, implementation of changes required to meet a government mandate, better reuse and software management, application rationalization and promoting standardization.

In this paper, we review the principles of Service Oriented Architecture and specifically how these principles relieve the above challenges and align people, processes and data through the use of open standards and a loosely coupled architecture.

Current State

As the business environment and consumer expectations changed, retailers developed applications and systems as the needs arose. Initial objectives were primarily to minimize manual processes and facilitate information flow at the point of sale. As such, siloed applications became the norm. To do away with a blend of manual processes, incompatible purchased systems, applications developed in-house and a legacy computing environment, e-business was adopted, but with limitations on efficiency of external collaboration, data centralization and channel optimization. For instance, a corporate level eCommerce implementation would not accommodate the needs of store systems or the store systems and processes were not structured to change with shopper's changing needs.

A typical non-SOA architecture adopted in the past is illustrated below, with siloed functions and a legacy infrastructure, where each functional department has its own applications and databases. The subsequent sections detail how retail functionality could be achieved through using the SOA architecture and redesigned in a more granular reusable way.


Corporate
(Planning, logistics, HR, marketing, finance…)
DB sync
DB sync
DB sync
Merchandising
(Demand forecasting, revenue management, production, inventory management, vendor management…)
Store Operations
(Workforce management, order management, POS, customer management…)
Supply Chain
(Component supplier management, warehouse management, logistics, sourcing, procurement…)
APIs
APIs
Online merchandising software
Add on modules to existing software solutions (financial s/w, POS etc.)
Direct Calls
Web applications (stand alone)
In-house, custom-built apps
EAI
POS solution
Handheld device applications
CRM application
Direct Calls
Automated order management application
Standalone web applications
Other in-house, custom-built apps
EAI
PLM application
Billing application ; 2-way integration to MS Project
Add on modules to existing software solutions (financial s/w, POS etc.)
RPC Calls
Supplier collaboration, warehouse operations etc.
In-house, custom-built apps
ERP Application
Point Solutions
Custom-built Apps/legacy
ERP solution

As can be seen, the needs of each functional department are addressed through independent applications, leading to a number of point solutions and custom, legacy functionality that are then integrated through APIs/EAI/DB calls/ERP etc. leading to a heterogenous, spaghetti environment. The functions are aligned with separate databases, resulting in an unreasonable demand on time, effort and resources to synchronize data across functions. To list a few typical challenges of such architecture:

- A number of autonomous business entities - independent and discrete that lack good connectivity
- A heterogenous computing environment with multiple software solutions running on multiple platforms that were never intended to work together
- Over-customized packaged solutions that cannot be upgraded owing to the 'tweaks’
- Delays in delivering information to each function across channels (no real-time information), performance slowdown owing to complex integration issues and end-of-day synchronizations
- Lack of/ limited visibility into data across functions, multiple versions of the data / divided data
- Lack of standards-based interfaces – difficult external collaboration
- Redundant processes, duplication of functionality

Translated to retail terminology, this could mean:

- High IT maintenance costs of disparate platforms
- Excess inventory (lack of visibility, monitoring, real-time information), out of stocks
- Marketing disparity among channels (online, brick-and-mortar), no single view of the customer
- Scattered customer information – difficult to capture, analyse, roll-out loyalty programs
- Difficult to coordinate promotions owing to integration challenges, irrelevant up-sells and cross-sells
- Delay in realizing changing business goals or in monitoring them

In short, the need for an integrated solution that replaces the siloed approach between the different departments took on greater importance. The term 'integration' took on a new significance because the business climate required that what would be integrated today would need to be changed even before the integration was complete.

A service-oriented architecture enables such integration in a flexible, reusable manner allowing quicker, consistent responses to meet customer demands. It allows efficient collaboration through combining services that are implemented using open standards, thereby increasing maintainability and reuse, with no clutter to cut through.

The subsequent sections in this paper detail how data and software can be orchestrated as services to bridge the different functional silos in an evolutionary manner, to deliver timely, relevant information across functions and channels.

The SOA way

Considering design today needs to evolve throughout the product lifecycle, information would need to pass from post-sale (stores), back to pre-sale (design) systems to enable feedback loops, linking information in a flexible, inter-operable manner, breaking down functional silos. SOA enables combining services into business processes that are dynamic enough to be executed across boundaries. In fact the very purpose of adopting SOA is to enable dynamic changes within the function, across functions and across the ecosystem of partners. A well implemented SOA strategy leads to a service-oriented Enterprise (SOE), allowing flexibility at all levels – infrastructure, applications, organization and external collaboration. A SOE is usually the guiding vision while embarking on a SOA journey.

Prior to detailing an SOA implementation, it is important to recognize that owing to the diversity of the retail market, a one-size-fits-all approach to an SOA implementation may not work. The approach should be customized according to the functionality and needs of the retailer.

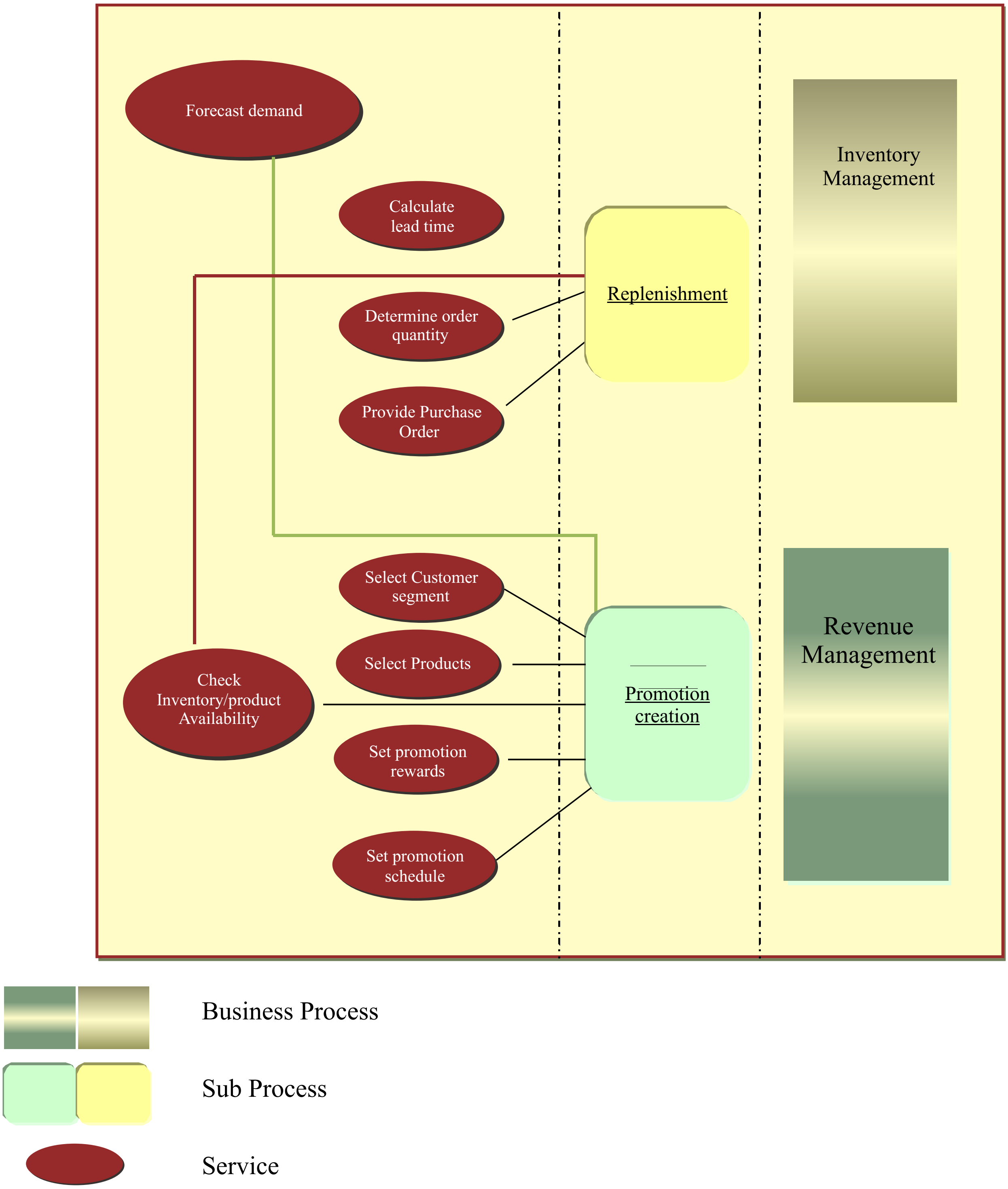
Below are some key features of a Service-oriented architecture:

- Distils large software applications/ logic into smaller free-standing, self-contained units of code known as services that do not need to know the underlying implementation technology.
- Services can be composed/ combined to form a business task. Business tasks can be combined to form a business process. Business processes come together to form enterprise applications.
- Uses web-services that are standards based (messages travel via a set of globally standardized and accepted protocols) for communication (popular choice for implementing an SOA as it supports SOA principles).
- Enables building on existing investments through the years rather than scrapping software and starting from scratch. Legacy application logic can be encapsulated and exposed through a common, standardized communications framework.
- Allows one to choose best-of-breed environments for specific functionality. No matter how proprietary the software, if it supports the creation of web services, one can create a service interface layer for it to talk to other service capable applications.
- Allows abstraction of business logic and application logic with the result changes introduced in one layer does not impact the other, providing the much needed IT flexibility in a dynamic organization

A simple definition of SOA that summarizes the above points is available on WhatIs.com *(A service-oriented architecture (SOA) is the underlying structure supporting communications between services. In this context, a service is defined as a unit of work to be performed on behalf of some computing entity, such as a human user or another program. SOA defines how two computing entities, such as programs, interact in such a way as to enable one entity to perform a unit of work on behalf of another entity. Service interactions are defined using a description language. Each interaction is self-contained and loosely coupled, so that each interaction is independent of any other interaction)*. It goes on to illustrate the concept through an example: of purchasing online at Land's

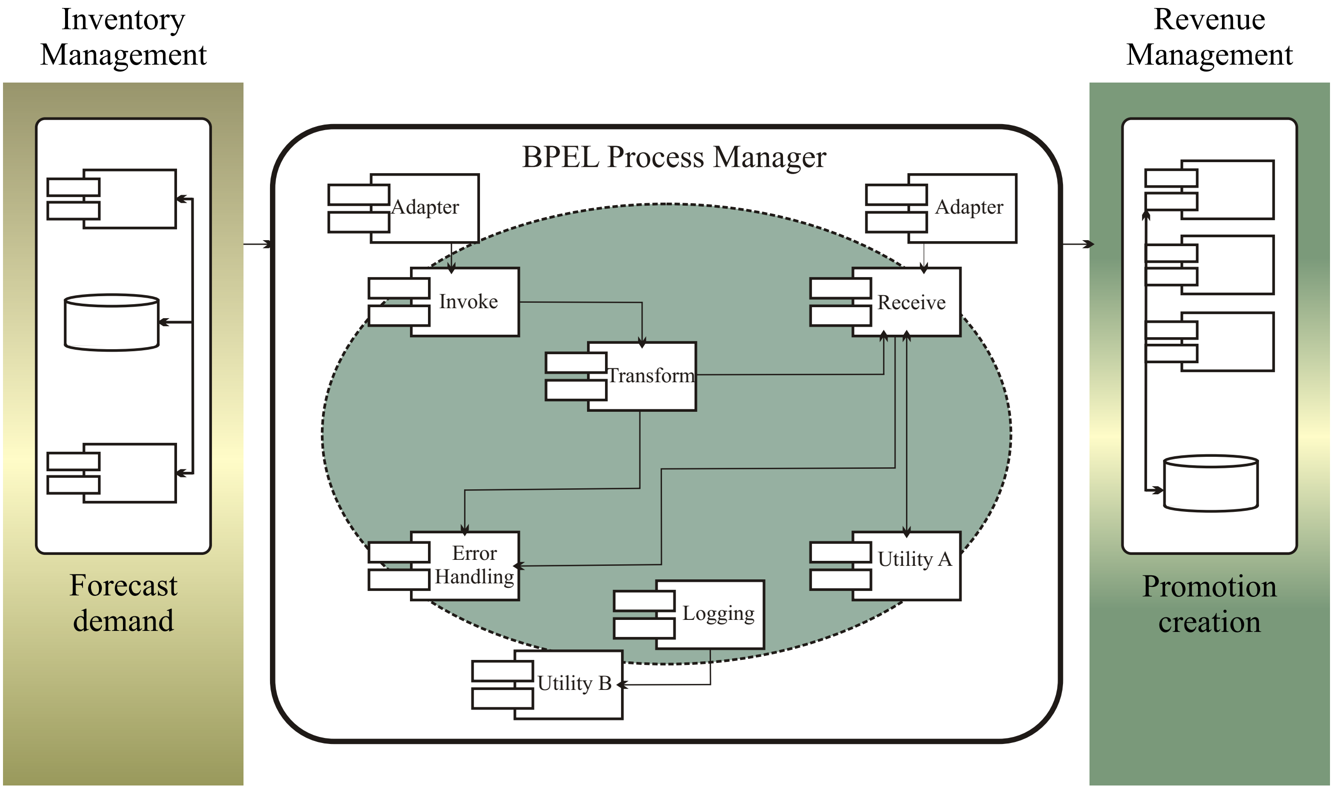
End. An example below provides more clarity on the concept.

We could model two of the core retail processes in SOA – Inventory Management and Revenue Management as in the diagram below. Only one sub-process is considered for each of these business processes to keep the representation simple. The two business processes share common services - 'Forecast demand' and 'Check inventory availability', in order to complete the respective functionalities.

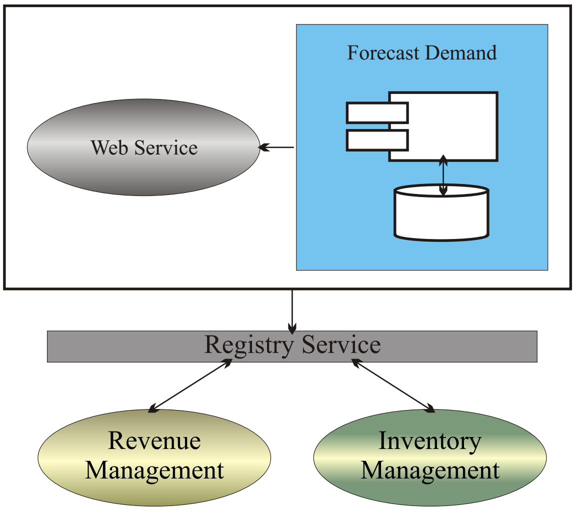


1. Each sub-process / business process can be broken down into free-standing services that are interoperable as depicted in the diagram above. Each business process requests IT functionality from the underlying systems by calling a service.
2. The services could be combined/orchestrated and reused in any order to form a business task / process. The 'Check inventory availability' service will be orchestrated differently for each of the sub-processes 'Promotion creation' and 'replenishment'. You would check the available inventory prior to replenishing stock. You would promote a product to increase sales in a particular category after checking the inventory. Similarly, the demand forecast service would be stitched to the needs of each business task.
3. All services are registered in a service registry to avoid redundancy.
4. The structure above enables separation of business logic and application logic, allowing changes in each, without affecting the other.
5. If a new sub-process, say marketing were to be involved in a promotion, they may want to forecast the demand for a particular product, in which case, they would reuse the available services and orchestrate them differently

A typical orchestration of reusable services, threading and sequencing, using Business Process Execution Language (BPEL) is depicted below. This enables encapsulation of workflow logic for a specific process. An orchestration service layer enables standardization on centralized business process abstraction.



A simple representation of how a typical service would be exposed through a registry is depicted below. The forecast demand service would be published through a registry to be accessed by the two business processes.



Typically, the steps involved in an SOA implementation would be:

- Defining the roadmap – identifying systems, the functionalities to be grouped and the step-by-step implementation process
- Consolidating business processes in the middle layer
- Encapsulating business logic in easy-to-manage services
- Ensuring reusability through services
- Ensuring security at each level
- Defining and implementing governance processes
- Building templates and checklists as required

Tasks would include:

1. **Determining where to start**

   This would involve understanding where we are today, what will need to change to achieve the desired end state and what impact (financial, time, resources etc.) that change will have. Determining where efficiency needs to be improved first, is usually the optimal way to start the implementation. For example, if the current focus is on improving the store systems to sell more efficiently, then it would be logical to start from determining the applications that focus on 'selling more efficiently' and listing out the functionalities that should be targeted first to convert into services.

2. **Documentation**

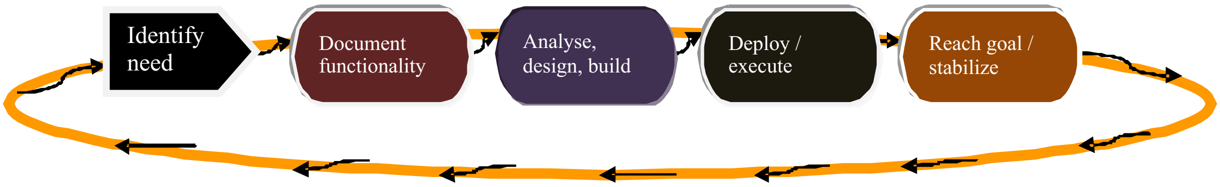
   Documenting the use-cases will assist in better understanding the scope of work. It will also enable consolidation of functionality and eliminate redundant applications. The documentation would include inputs, outputs, interfaces, dependencies (could be other use-cases in other applications), logic and structure. This will allow one to form an initial estimate of the timelines and effort involved.

3. **Develop capabilities internally**

While it is important to choose the right service provider (Refer section on 'Assistance in Implementation') that provides specialization and niche services in SOA, it is equally important to have an internal team that is focused on the architecture and the broad goals to achieve for migration to an SOA. We could call this the 'central architecture group'. This group would work along with the service provider (and other internal teams – analysts, delivery group etc.), providing the required support and direction to the provider to achieve business goals.

4. **Migration of 'As-is' to 'To-be':**

Migrating to an SOA is preferably an iterative process, where changes are introduced in a phased, milestone based, gradual step-by-step approach. The migration starts with one functionality at a time that is service-oriented, progressed through testing and execution until the project is complete, then the next functionality/need is tackled. The enterprise gains knowledge and experience with each iteration, continually building up SOA skills and maturity. Depicted as below:

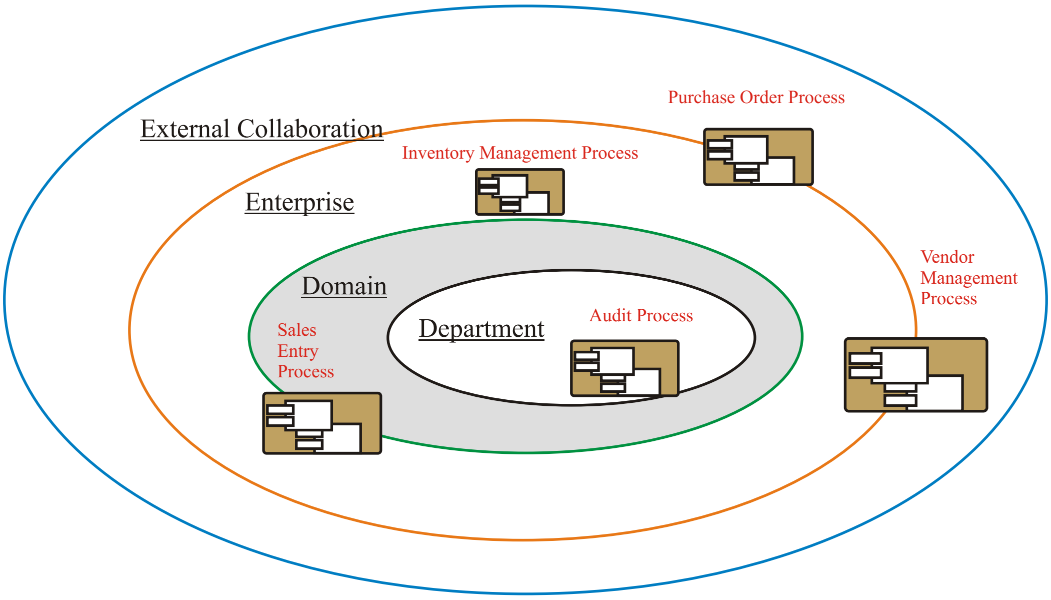


During such a migration, typically there would exist hybrid architectures within the enterprise that are part legacy and part service-oriented. With each iteration, a number of reusable services are built up with the result, for new projects, existing services could be reused or additional new services created thereby drastically reducing the time, effort and complexity of the IT landscape.

More details on the reasons to adopt SOA and the steps to adopt them are available in our whitepaper "A guide to SOA implementation" (http://www.thbs.com/soa_whitepapers.html).).

SOA and Reuse in Retail

One of the core benefits of adopting SOA is reusability of technical assets and the cost savings that stem from such reusability. All services may not be reusable, however, a systematic approach to determine the components/services/processes that could be reused at each level – external collaboration, enterprise level reuse, domain level and department level reuse, would be a beneficial exercise in the analysis phase.



External Collaboration

A vendor management process, used by an enterprise to manage multiple vendors that offer products/raw materials and services is a good candidate for reuse. Business performance could be improved by gathering information about the different vendors and using the data gathered to determine the optimum vendors to retain thereby sourcing the best products and services. New vendors could be connected to the enterprise dynamically through exposing a standards-based vendor management functionality, saving considerable time and effort through reuse.

Enterprise level reuse

A good example of a reusable process at the enterprise level would be the 'inventory management process'. For instance, if we consider the two functions merchandising and store operations in an enterprise, the movement of inventory at different stores would determine the merchandise to be stocked at each store. Business process optimization could be achieved through granularizing the inventory management process into services, depending on the needs of the two functions to promote reuse.

Domain level reuse

The sales entry process could be a potential candidate for reuse across store operations. This category of services would span across a line-of-business (LoB)

Department level reuse

Common utility functionalities could be reused across applications in the department. Functions such as logging, error handling, notification, authentication etc. could be independent services that are reused by a number of internal applications.

Establishing an approach towards the adoption of a reuse program is essential prior to defining the roadmap for implementation. Developing a catalogue of reusable processes enables effective structuring of services and leads to a cost-effective architecture.

A few essential Do's for an SOA implementation:

- A good practise would be to involve all departments throughout the implementation process. Strong partnerships between departments allow more expectations to be managed about the project. Maintenance could become monumental if the implementation is not carefully coordinated and the services not centralized where possible.
- The risks associated with large rollouts and end-of-project testing could be reduced through using the step-by-step iterative approach for implementation.
- Using SOA governance principles early on, to monitor and track is usually a good practise.
- Product investments in SOA are usually driven by business goals - proficiency by a vendor in a specific sector/ area is important. A number of products provide overlapping functionality, hence, assessing the need and matching it to the products' strengths is essential.
- The solution is designed to provide a clear separation of concerns, allowing binding to take place at will (loose coupling). It is essential to keep it simple, to enable quick implementation (training then becomes a non-issue).
- Rather than as a one-off project, establishing an internal division to improve systems on a regular basis is usually the norm. This division comprises of members from the business community and the technical IT community, to get a combined insight on the business needs and the implementation aspects.
- While making legacy applications SOA-compliant, appropriate load-balancing would have to be ensured to enable legacy back-ends to cope with increased usage volumes.

Business functionalities that can be exposed as services

Some of the common business processes and their services that can be modelled based on SOA:

Business Process	Sub-Process	Example services in a Sub-Process
Inventory Management (planning and optimization)	• PO process • Receipt process • Allocation process • Distribution process • Transfer Process • Stock optimization process • Demand planning process • Forecasting process • Replenishment process • Style coordination process • Physical count process • Inventory freeze process	• PO process: • Generate serial numbers • Map line items • Assign, track, update costs • Approval, notification • Record shipments • Bin Lookup • Audit-control • Report generation • Multi-currency transactions • Line-item level tracking • etc…..
Revenue Management	• Price-change simulations process • Promotion creation process • Allowances process • Vendor discounts process • Markup creation process • Markdown creation process	• Price-change simulations: • Item role assignment • Link items • Create, apply business rules • Approvals and feedback • Pricing lifecycle planning • etc…..
Sales Management	• POS system • Sales entry process • Sales audit process • Stock ledger process • Consignment sale entry process	• Stock ledger process: • Opening Inventory • Cost of Sales • Comparatives • Net transfers • Performance gross profit • etc…..
Supply Chain Management	• Demand management process • Warehouse management process • Transportation management process • Trade logistics management process • Reporting process • Supplier Relationship Management • Sourcing • Service Management	• Supplier Relationship Management process: • Contract management • Spend analysis • Supply planning • Commodity management • Procurement • etc…..

Business benefits

A few benefits of a SOA implementation in the retail sector are listed below:

- Better customer information management: In multi-channel retailing, customer information capture from multiple channels, analysis and use through consistent service reuse help achieve better knowledge of the customer at a reduced cost, facilitating customer loyalty programs and promotional offers
- Linking the different channels: Helps link the online world with the brick-and-mortar outlets' applications through loose-coupling and open standards, enabling the addition of new channels dynamically. Increases chances of a sale and provides for better up-sell and cross-sell opportunities.
- Enables support for best-of-breed applications that can expose a service interface (without having to over-customize them), thereby leveraging the investment made in automated solutions.
- Allows for scaling at marginal cost. New initiatives could be implemented through additional services or service re-use.
- Legacy investment is phased out gradually, while reusing legacy applications across disparate environments through web services and frontier based enterprise service buses or other network based hardware that enable protocol and message transformation at the local frontier inside a wide spread federated environment that has native requirements within it
- Helps optimize supply-chain through improved cross-application integration.

Assistance in implementation

SOA is an evolution, an ongoing conversion into services that are reusable and increasingly perform independently. As additional business needs surface, the iteration repeats, leading to an agile enterprise. Often a review of services after the go-live stage reveals areas that can be re-looked at immediately in terms of enabling better reuse for upcoming projects. Hence, it is important to choose a partner/service provider that you see compatibility with in spirit as in ability. While rigidity is unwelcome in any service relationship, it is alien and contrary to the concept on which SOA is built. Hence the willingness to embrace change, to serve on the fly, are traits that while desirable in all relationships, are mandatory in choosing meaningful partners and products to implement SOA.

There are a number of product vendors, offering a range of products right through the analysis, design, implementation and test phases. They often have carefully chosen, approved implementation partners that are certified on the product and greatly simplify the implementation and integration needs. In fact, many of the service partners are certified to work on a number of competing products, thereby enabling the enterprise to better evaluate and implement a product based on the business needs.

Since SOA is about integration of services without destruction of technical assets, a number of niche players in the middleware and integration space offer specialized SOA implementation skill-sets and services.

Conclusion

Retail researchers estimate that consumers who shop at the same banner whether online or at a brick-and-mortar store – spend 14% more annually than consumers who shop via only one channel. As the number of service points increases, so do the integration demands. SOA-compliant applications significantly lower the cost and effort of cross-application integration, through a vendor-neutral communications framework, providing intrinsic interoperability and agility to an enterprise.

An SOA centric approach including a top down and bottom up approach in the retail industry is perhaps essential to meeting the customer demands and challenges posed in this domain that are more susceptible to change than in any other vertical.

In the words of Edsger Djikstra in his classic work "The Humble Programmer"

The vision is that we will be able to design and implement the kind of programs that are now straining our programming ability, at the expense of a few percent of man years of what they now cost to build and run…

This statement was relevant in the early seventies when it was first written and is perhaps even more compelling and relevant today. SOA, more than any other methodology, helps to shorten the effort and complexity of change. All we have done is given a name to a solution that began long ago, and distilled it for easier and widespread consumption.

Torry Harris Business Solutions Inc, a US based services provider with a large base of technologists located in the UK, India and China has provided cost effective solutions at a design, development and support level to a variety of enterprise clients across the world since 1998. The company specializes in integration, distributed computing, and its focus on SOA is a result of nearly a decade of expertise gathered in the middleware space. The company has partnerships with almost all the leading SOA and integration product vendors. SOA, involving the creation of autonomous parts of a solution, lends itself admirably to the cost effective model of offshore service collaboration. A separate white paper entitled "SOA Implementation with an offshore partner" available for download, explores this model in a more detailed manner. Further information about the company and a variety of white papers on SOA are available at www.thbs.com/soa.

For more information, write to us at soa@thbs.com.