



PeopleSoft Enhancements with SOA and Mashups



PeopleSoft Enhancements with SOA and Mashups

Service Oriented Architecture

Introduction:

Service Oriented Architecture (SOA) has become the big buzzword in the tech industry these days. All the major enterprise software and middleware vendors are fundamentally changing their current offerings and building new products around the concepts of SOA. But what does this new push to SOA in the enterprise mean to you and your organization? Why do you even need to consider SOA for the future of your enterprise? How do you get there once you do decide you are ready to take the steps towards making your enterprise SOA ready? The focus of this white paper is to raise the awareness of what SOA can do for your organization and how EMS-IC can help you move your ERP systems like PeopleSoft towards Service Oriented Architecture.

What is SOA?

First, let's start with what SOA is not. A common misconception of SOA is that it is SOAP (Simple Object Access Protocol) based web services. While SOAP currently plays a large role and is perhaps the most popular technology used to expose services, SOA is in fact just an architectural design methodology that can be implemented using a variety of technologies.

Sometimes SOA services are exposed as RESTful services where the contract between the client and the service is a little more loosely defined but also doesn't suffer from the sometimes-bloated nature and complexities of SOAP based services. The SOA methodology could also be implemented using messaging services like JMS or with higher-level orchestration languages like Business Process Execution Language (BPEL).

SOA is completely centered on exposing and consuming services in both the internal enterprise as well as the external Internet enterprise. SOA differs fundamentally from other methodologies of sharing information and re-use that classically have been used in the enterprise. Functionality re-use typically took the form of a custom library developed in a particular language like, for example, Java.

One of the hurdles with re-using functionality at the system level is that custom libraries or services are centered on a specific technology or language and use specific bindings that were not necessarily standard across all platforms. Dealing with this hurdle would sometimes force organizations to create duplicate functionality, services, and sometimes even data in all the various technologies represented in their enterprise and as such negate the whole premise of true re-use. As the enterprise evolves, having this patchwork quilt of "copy and paste" technologies and duplications of services and data makes it much more difficult and costly to react effectively and timely to organizational level changes.

Service Oriented Architecture seeks to solve this classic problem by breaking down enterprise level functions into simple services that provide a specific set of functions. As more and more

functionality is exposed this way, upstream systems that rely on those services can mix and match those services in whatever manner is appropriate for the task they are seeking to solve. If a particular piece of data or functionality is not available, instead of creating it themselves, the data or functionality is implemented as a new enterprise service that other departments in the organization can also leverage.

SOA also takes away the need of the upstream client to be aware of the nuts and bolts of how the service is built. Using SOAP as a communication layer, clients of the service can use whatever technologies are appropriate for the problem they are solving. The client is no longer required to have a lower level of support for the technology the service is using to expose data or functionality. In essence, protocols like SOAP is greatly reduce the impedance mismatch between systems.

What does service mean?

Consider the following scenario of how the phone company's 411 service is used. The phone company provides the 411-phone service so that customers can quickly pick up a phone, dial a short number, and ask an operator for directory information. The caller of the 411 does not need to know how that information is stored, how it is retrieved, or what technologies are used to organize, store, and retrieve that information. The customer simply picks up a phone and requests for information from the service using a standard means of communication. As simple as this example is, it describes the basic premise of what a Service Oriented Architecture strives to accomplish.

Before the advent of SOA and technologies like SOAP it was much more challenging to provide services to your enterprise clients both internally and externally. For example, if your organization wanted to create an enterprise level directory service and expose that core service to multiple departments within your organization, you would have to purchase the raw directory data from the phone company in whatever format they provide. Before you could use that data in your enterprise, you would also need to create a new system to import that data into and create mechanisms that expose that functionality. Sharing that functionality and data across a diverse landscape of technologies, platforms, and protocols represented in your enterprise presents its own set of challenges.

Another potential issue with this approach is your relationship with the directory service provider is more tightly coupled. In order to remain up to date with current directory information, data must be periodically refreshed and if the data provider changes the format of the data, your enterprise is now forced react to changes in the data provided by an outside entity that is not privy to your enterprise needs, schedule, or evolution.

What if instead of having to do all that work yourself, you could have the same ease of integration with your organization's enterprise as you do when you pick up a phone and call the 411 directory service? What if you could just let the phone company do what they do best and you can focus on what your organization does best? SOA provides the same level of interaction conceptually as you achieved when you picked up the phone (the interface) and using English (the protocol), asked the directory service operator for information (request and response).

In the enterprise SOA version of our 411 services, using the Internet (the interface) with SOAP Web Services (the protocol), those same directory services could be provided (request and response) directly to your upstream systems without your organization having to house and maintain that capability. You also no longer have to create new infrastructural systems to import the data into and share with the rest of your enterprise.

Since the SOA service is based on an industry standard protocol, integration is also simpler. Now you are back in the better situation where you do not need to care about the details of the directory service. Accessing that information directly through a self-describing service frees you up to move on to the work your organization does best.

Breaking Out Of the Silos

SOA methodology also seeks to break down the silo mentality where each department within an organization has their own applications, data, and lifecycles to manage. Duplication of effort, code, and even data are downsides of the silo mentality. While it may be easier to manage a particular project within a department, it is not always in the best interest of an organization to allow each department to not at least consider the enterprise needs as a whole. Silo mentality can also lead an organization down the path of having to manage and maintain multiple solutions built on many different technologies and methodologies making it a major challenge to react to future changes.

Just like how the example of the 411 service demonstrated utilizing functionality exposed by external services instead of bringing them in house, creating and exposing internal services specific to your enterprise is also beneficial to the overall growth potential of your organization. It is not a trivial task thinking in terms of simple services that do a very specific set of functions. But doing so opens up a new world of functional possibilities that you may not have even thought of when you first decided to create a service for your enterprise.

Orchestration and Mashups

Orchestration and Mashups are what can result from breaking out of the silo mentality of application design. Business processes exposed in their more basic forms can now be orchestrated and combined or "mashed-up" using industry standard middleware technologies like Business Process Execution Language (BPEL) and Enterprise Service Bus (ESB) or any number of other middleware technologies that are SOA aware.

Having the ability to orchestrate or mashup all the various exposed enterprise services across an organization brings a new level of power to the enterprise. Creating new applications and processes off of existing services is now easier to do and does not necessarily require long development life cycles to build, manage, and maintain those new SOA based applications. As the tools continue to evolve and become easier to use, the holy grail of having anyone be able to easily alter the behavior of an enterprise system and create new functionality is closer to a reality than ever before.

PeopleSoft and SOA

Historical Approach To Enhancements

Traditionally, enhancements to PeopleSoft have been done using Oracle tools like PeopleCode and AppDesigner. While these tools are extremely powerful and well supported by Oracle, the future direction of PeopleSoft enhancements is moving towards externalizing enhancements using SOA techniques. Adopting the SOA approach to PeopleSoft enhancements and reducing the amount of enhancements done in PeopleCode helps to mitigate the risk that those enhancements will be impacted by future upgrades and can also serve to reduce the costs involved with determining those impacts.

Consolidation Of Functionality

One of the main reasons organizations choose to use PeopleSoft is that it has an enormous level of proven functionality and a large customer base. Organizations can store and manage their information in PeopleSoft and know that they are not using a technology that will evaporate like so many of the other vendors and their technologies. Having the backing of a large, well respected company like Oracle is big plus. However, because PeopleSoft is so powerful and so full featured, it many times is overwhelming to organizations and their users.

Modules purchased by an organization include far more features and functionality then they really need to satisfy their requirements. Since other organizations did need those features, PeopleSoft needed to include them in order for their other clients to be satisfied with the level of functionality.

The main issue with this for many organizations is that they are forced to deal with multiple screens and data entry points that have no bearing on their organization. It costs organizations money to personalize the application to their needs from a user friendly UI to multiple integration touch points both inside and outside the enterprise.

Once the application is enhanced and modified this situation leads to another approach in managing enhancements inside of PeopleSoft. It now becomes more strategic, how does the customer adopt another approach to minimize customizations while getting their application to mimic their business process.

Enhancing PeopleSoft using SOA

Oracle is now offering a different approach to enhancing PeopleSoft. Using standard SOA technologies like SOAP, PeopleSoft functionality can be leveraged and enhanced with fewer evasive changes to PeopleSoft. Exposing PeopleSoft components via SOAP based web services, the SOA approach allows developers to leverage PeopleSoft as a service but not be so tightly coupled to the underlying PeopleSoft code. Since the component is now exposed as an abstracted service, changes to the supporting code behind that service can be done without impacting the service itself. Oracle has basically stated that if you use the SOAP services, the SOAP service itself will not be changed. The net effect is that organizations can now choose

when they are ready to move on to new features that may have been added by the recent PeopleSoft upgrades.

In the true spirit of SOA, integration with other systems in an organization is now easier due to industry-backed standards. PeopleSoft components can now be easily re-used across the entire enterprise eliminating the need for each upstream client to have to build and maintain their own separate integration points. Internal data can be combined with external data to provide a whole new level of functionality to organizations end users. This mashup of services can be consolidated into a single application which has the nice side effect of less training for end users, less frustration, and less infrastructure integration across multiple solutions.

Information can now be easily shared across multiple PeopleSoft instances and modules. For example, sharing and combining data across HR, Student, and Financial modules in PeopleSoft is not only possible, but also encouraged. Creating portal dashboards that provide a quick glance into data across the various PeopleSoft modules and instances can greatly reduce the amount of time and effort it takes to get to the data that you are interested in. Creating custom views of that data across PeopleSoft systems is also much easier to do now that all the PeopleSoft modules can be exposed as services.

PeopleSoft Mashup

The term mashup in Wikipedia is defined as “combining data from more than one source into a single integrated tool”. While this concept is nothing new, with the advent of SOA, the mechanics of realizing this concept in the enterprise has become easier to obtain and less evasive to the sources involved. By exposing PeopleSoft functionality using SOA techniques, data from different sources can now be combined in a more ad hoc fashion.

A real world example of a PeopleSoft mashup could be combining information from the HR module with outstanding travel advanced and travel reimbursement in the Finance module to create an analytics report to help control costs. Another example could be bringing democratic information like race, sex, etc. together with benefits plans information alongside total benefits deductions from payroll.

For higher education customers of PeopleSoft, the ability to combine campus solutions people information with payroll and payables activity by student to discover how much each student is getting paid in total, is an extremely valuable mashup made possible by enhancing PeopleSoft using SOA principles.

EMS-IC and SOA

Bringing intelligence to the chaos

EMS, with its decade of experience in PeopleSoft and other ERP solutions, is now offering through its Intelligent Chaos division (EMS-IC) a partnership with organizations to assist in moving towards a SOA middleware approach to PeopleSoft enhancements. With the help of our expert consultants and EMS-IC SOA technologies grown from real world solutions,

organizations can now completely change their PeopleSoft experience by tailoring it to their specific business needs without sacrificing the power of the PeopleSoft platform and all the modules it contains. No longer are enhancements only done in PeopleCode or done in none supported fashion. Any technology that supports the protocol standards found in a modern SOA middleware infrastructure can now be used to create the exact features organizations need to effectively run their organization.

The EMS-IC Approach

The EMS-IC approach to bringing SOA methodologies to an organization revolve around a customer centric approach. What this means is we analyze your requirements and with our in-depth knowledge and experience with both PeopleSoft and SOA middleware, craft the exact solution that your organization needs.

Completely custom rich clients can be created to put a brand new face on the PeopleSoft and streamline the user's experience and productivity. Using modern technologies like Adobe Flex to author GUIs, EMS-IC brings a whole new level of richness to the users. No longer are users forced to deal with multiple input screens that have no meaning or use to their organization. Those numerous screens can effectively be removed from their day-to-day business process. Specific business logic can also be housed outside of PeopleSoft and without the constraints of PeopleCode capabilities.

Adopting SOA brings flexible and powerful new capabilities to PeopleSoft and is being widely encouraged by Oracle for the benefits SOA brings to its customers. Oracle provides unlimited support for exposing PeopleSoft functionality to the enterprise in a SOA fashion. Doing so gives the client much more control over enhancements and increases flexibility in implementation approaches as well as minimizing issues with future upgrades to PeopleSoft.