JOURNAL of PRACTICAL CONSULTING

The Case for Business Agility in the SAP Dominated Enterprise

Probal DasGupta
SAPBUREAU Services Worldwide, Inc.

Business agility can be achieved in a SAP dominated enterprise by creating process-centric systems through the use of Business Process Management (BPM) techniques and tools. Such a system would be deployed through the Model Driven Architecture (MDA) methodology on a Service Oriented Architecture (SOA) platform. We should seek to (a) create Process Centric Systems by using BPM to converge business applications towards process-oriented enterprise computing by capturing all process-knowledge via modeling, (b) embrace a SOA-based open systems architecture using SAP's Enterprise Service Architecture (ESA) as the blueprint for implementing a service oriented business architecture through SAP NetWeaver, and (c) adopt MDA as our system deployment methodology by using Model Driven Architecture (MDA) as the modeling-based software development approach that ultimately deploys the processes on the computer. SAP's Enterprise Services Architecture (ESA) already embraces SOA and BPM and can be extended to utilize MDA concepts. ESA strategy is about better aligning IT with the business. But ESA goes far beyond a technical theme or fad. ESA is the dawning of new agile, process centric focus within IT organizations.

To thrive – or even to survive – in today's competitive marketplace, companies need to continuously balance differentiation for top-line growth and productivity for bottom-line profitability. To succeed, they need a flexible IT architecture that enables them to quickly implement new, innovative business processes, without changing core business engines.

Today SAP has emerged as the undisputed leader in Enterprise Software and while we cannot vouch for the veracity of their advertisement that states, "The Best Run businesses run SAP," it is certainly true that over 90% of Fortune 500 companies do run SAP. This article, therefore, focuses on the exploration of how business agility can be achieved in the SAP dominated enterprise.

Strengths of Classic SAP

What Classic SAP (R/3 or mySAP ERP) does really well is automate core and non-core processes that are essential for running a business but not necessarily for differentiating it. These are called "Best Practices" and comprise all the things that companies need to do in order to be a functioning business entity, such as inventory management, invoicing, order-to-cash, and so on.

Journal of Practical Consulting, Vol. 1 Iss. 1, 2006, pp. 2-6

ISSN 1930-806X

Building Business Differentiators using Classic SAP

When it comes to differentiating processes, one needs to build something that reflects this uniqueness. With Classic SAP, this means custom development using ABAP and related coding facilities – consuming time that you may not have and money that you can ill afford.

As a result, business agility becomes a primary concern. The question arises: Even after you have mapped your differentiating processes on to SAP, how quickly can you adapt to changing business conditions and adopt newer ways of conducting your business when the need arises? The answer is: Not very quickly.

Also of concern is the fact that, in a fast paced marketplace, what is differentiated today becomes non-core and ubiquitous rather quickly because all competitors begin emulating the leader. So whether you are the market leader or not, you are soon left with a lot of custom code ABAP-type for non-core processes, which now has to be maintained forever – at a cost you can not justify for non-core processes.

Enterprise Services Architecture

Enter SAP Enterprise Services Architecture (ESA), the "blueprints" for service-oriented business solutions enabling the "in-time enterprise" that is always in tune with the ruling business conditions. Service Oriented Architecture (SOA) makes this nimbleness possible, and SAP NetWeaver, which is central to SAP's ESA strategy, is built entirely on the SOA platform.

Service Oriented Architecture (SOA)

SOA allows business processes to be packaged as a service. When a new business rule arises, it is released as a service. From this pool of business function services, all core and non-core processes are composed. This is what SOA is all about: the service components continue to be composed and re-composed (i.e., re-used) in building processes that are core and differentiated today and might be non-core tomorrow.

Open Systems Framework through SOA

Service Oriented Architecture (SOA) should be our architectural platform of choice, because event-driven service oriented architectures are most adaptable to change over time, and adoption of open systems standards retains the greatest flexibility and eschews any kind of dependence.

SOA achieves loose coupling among interacting software agents. It enables our mature component expertise to use SAP middleware to modularize systems and applications into "services" (that are discrete business components with well-defined interfaces), and then to combine and recombine them in ways that meet your needs within your SAP framework.

With SOA we supplant old Enterprise Application Integration (EAI) techniques with the assistance of Web services standards. These standards simplify the creation and consumption of tasks through Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and extensible Markup Language (XML). Working with standard protocols eliminates much of the complexity of EAI so that companies can focus on improving their services instead of getting bogged down by integration difficulties. SOA promotes open systems by defining all functions or services using unified service description language and well-defined public interfaces, thereby making the solution architecture platform-independent. The ubiquitous and abstracted nature of the services also supports flexibility in choosing which service implementation to use in a particular application.

SAP is Leading the Charge in SOA

SAP is the foremost among Enterprise applications in adopting SOA across the board and building its new middleware, NetWeaver, on an SOA platform. SAP's Enterprise Service Architecture (ESA) actually envisions a world order based on a "Service Layer" where SAP already has 30,000+ services. SAP, in fact, extends the SOA paradigm into its Service Oriented Business Architecture (SOBA) world vision where 50+ cross-industry scenarios are based on 200 mega-processes and use 30,000+ web services. SOBA web services already cover legal rules of 110+ countries across 23 industry verticals.

Service oriented architectures (SOA) have been considered best practice for decades, from the Open Systems Interconnect (OSI) Reference Model adopted by the International Standards Organization in 1983, to the Distributed Computing Environment (DCE) developed by the Open Software Foundation, to CORBA. However, the widespread adoption of SOA had previously been constrained by lack of agreement over interface standards. Web services finally solve that problem. It is the first interface standard that is universally accepted by all vendors, and will work with any programming language on any platform (with the caveat that the vendors stick to the standard).

SAP NetWeaver

Since SAP NetWeaver packages business processes as services, and these service components continue to be composed and re-composed (i.e., re-used) in building processes, NetWeaver is more of an "application compositor" platform than an "application integration" platform.

Although it does integration very well through its x/Apps interfaces that can integrate all .NET, J2EE and legacy applications into SAP, NetWeaver is well differentiated from traditional EAI (Enterprise Application Integration).

EAI keeps monolithic applications the way they are and allows them to "talk to" one another through messaging. Portals represent another level of integration. But from neither EAI nor portals do we get the advantages of "composition" that SOA (and, therefore, NetWeaver) provides.

Process-Centric Systems

This new ability to quickly compose processes by using (and re-using) pre-existing services enables us to shift the focus to processes and create Process-Centric Systems, as opposed to Data-Centric Systems that still proliferate.

A process-oriented approach makes no distinction between work done by a human and a computer and directs focus solely on the business aspects.

Process-centric solutions create business value in many different ways:

- Greater business agility through quicker process modification.
- Better systems through process clarity.
- Tighter process flow (reduced number of processes or reduced number of steps in processes) through better understanding of work flow.
- Higher productivity per person as a result of tighter processes.
- Reduction in elapsed process time as a result of tighter processes.
- Higher automation brought about by process clarity.
- Reduced cost per transactions within processes (as a result of higher productivity, reduced process time and higher automation).
- Improved quality/reduced errors in processes.
- Improved coordination across departments/geographies.
- Higher employee satisfaction through process clarity.
- And a host of associated benefits.

The most valuable is Business Agility, which is the most coveted goal of any business in today's fast changing world. That is why creating Process Centric Systems should be our principal strategic direction.

Business Process Management

We should focus upon the Business Process as the intersection between business and technology and use advanced BPM (Business Process Management) techniques to converge business applications toward process-oriented enterprise computing. This will unify process thinking across business and IT disciplines.

BPM for Complete Process Modeling

Business Processing Management should be our central technique to understanding, analyzing, designing, documenting, implementing, and maintaining business processes on a computing platform.

Using BPML (Business Process Modeling Language) as a meta-language for modeling business processes, XML as a meta-language for modeling business data, UML Diagram Exchange Specification as a graphic notational standard, BPSM (Business Process Semantic Model) for understanding the underlying semantics, all the business processes of an enterprise can be rigorously captured in an intelligent and malleable format. All definitions and schematics are stored in a Repository, which can be queried through the Business Process Query Language (BPQL) based on the Simple Object Access Protocol (SOAP).

BPM Toolkit

One should use leading BPM tools such as Filenet, AWD (Automated Work Distributor), Staffware, Ultimus, Savvion, and so on.

We are currently involved in a strategic partnership with ProcessWizard Limited in a joint development effort to make their BPM tool ProcessWizard compliant with AcceleratedSAP methodology, in addition to its current SCOR-compliance. We also use ProcessWizard for all enterprise modeling tasks, but particularly for SCM implementations since it has built-in SCOR templates.

MDA-based Software Development and Deployment

Based upon the solid foundation of BPM (Business Process Management) as the method of understanding and capturing all process knowledge via modeling, we use Model Driven Architecture (MDA) as our software development approach that ultimately deploys the processes on a computing platform.

The MDA approach defines the application (i.e., processes, business rules, and procedural details) and the data in the form of machine-readable models. First comes the *Computation Independent Model (CIM)* that presents the business functionality without any reference to computers. Then comes a representation of the system without any reference to the platform on which it will run. This is called the *Platform Independent Model (PIM)*. Next comes the *Platform Model*, which is a description of the hardware/software platform without any reference to business functionality. Now that the platform is defined, the PIM specs are applied to the platform to create the *Platform Specific Model (PSM)*.

The *Unified Modeling Language (UML)* is the standard modeling language used in MDA for visualizing, specifying, and documenting software systems. *UML 2* integrates a set of concepts for completely specifying the behavior of objects, the UML action semantics. The *Meta-Object Facility (MOF)* technology provides a model repository that is used to specify and manipulate models, thus encouraging consistency in manipulating models in all phases of the use of MDA.

The ultimate aim of MDA is to move the focus from technology to business by generating executable code directly off the business semantics encapsulated within the models. Instead of writing program code, all time is now expended in writing business specs. Instead of maintainable code, we now have maintenance business process models.

The ultimate benefit of using MDA is to help reduce complexity, lower cost, make applications immune to the changes bought about by new technology, and hasten the introduction of new applications (read “business agility”).

About the Author

Probal DasGupta heads a technology company focused on extending SAP's reach into the U.S. middle market by utilizing the archaeological principles of "Anastylosis" in software engineering. The resultant product called "instantSAP" is still under development. DasGupta was the *de facto* CIO of the ISPAT group, known today as ACELOR MITTAL, the world' largest steel manufacturer, until 1993 when he left to form his first technology startup in the U.S. called AmeriSOURCE Consulting. Thereafter, he has spawned, nurtured and grown other successful companies, SAPBUREAU being the latest.

Email: probal.dasgupta@sapbureau.com.
