

RIA cooking SOA

Emerging Face of Internet Technology

No, RIA is not the girl next door, it is the name of the new emerging technology that we will be building and using more and more over the next 5 years. RIA stands for Rich Internet Application, basically in short for providing rich user experience ala great desktop application over the internet without the need to install anything on the client machine.

And SOA is not the beans we are talking of which produces oil to meat substitute, but short for Service Oriented Architecture, in short of designing software applications, more so the logic end of the application into several loosely related (coupled) stand alone units that perform specific *service* on request. My favorite childhood game of Lego is what comes to my mind while thinking of SOA, like with same set of 110 parts of 'lego' you can create a car or a plane or a house etc; whatever put together makes sense and also not all 110 are used always. Of course, the lego example is over simplified version, but I can not help imagine it the same way. SOA is promised to be the logical solution to flexibility and cost reduction from re-use of codes for today's corporations.

So how does both relate together, how is RIA involved in the cooking of SOA? Well, the nature of both RIA and SOA makes it logical partner, SOA as a loosely coupled component of the application system, rarely have a front end, and RIA's capability to be independent of the Mid and back end makes it logical choice, not that we cant build it with HTML front end, but 100% HTML front end has its own inherent shortcomings to make it not the ideal choice for corporate applications.

Let us now try to explore some of the promises of both the technology in little more details, after all we business people have to have a benefit of using it, we do not just want to jump in because it is new.

Sometimes ago in one of our R&D brainstorming meetings I along with the team listed down a set of issues that today's internet applications have and when we came across RIA, it seems to be the answer to most of them. Let me list down the possibilities from using RIA:

- **Rich UI:** have you ever wished that your hotmail or yahoo mail was as easy to use as Outlook; well RIA is the answer to those wishes. Now with RIA you do not have to wait for the list of address to load while you are trying to find the name of the friend while on the phone, it's a click away like in your outlook.

- **Zero Touch Install:** basically the idea is, nothing has to be installed on the user PC for it to have the Rich UI. Imagine that, your hard disk does not get filled up, your memory does not need to be updated every year and the problem of application installation error etc all gone.
- **Reduce need for IT support:** Though not eliminated, RIA promised to reduce the dependence on the onsite IT support. As the application is not installed on the user PC, there is nothing related to the application that can go wrong on individual PC, everything can be managed remotely, new version getting to be available without any effort on the user end. So no more long wait for the download of patches and installing them. For a large organization this can result in substantial savings.
- **Supports Disconnected Use:** Think of Outlook mail application again, you can check your mail online or offline, you can write new mails outside the office and when you connect to the internet the synchronization happens. RIA promises this possibility with other enterprise applications also, on the Laptop, on the PDA, several applications can reside while on disconnected mode and allow the users to continue their work even offline, once in the network, everything gets updated. But even I am still waiting to see a successful implementation of this feature.
- **Platform Independence, well short of:** as its still an emerging technology, not all form of it does support this feature, but ideally RIA applications are suppose to be independent of the client operating system. So anyone in the organization using any type of PC can use corporate applications without worrying about if it will run.

All the above are the features and possibilities of RIA, but please do not forget RIA is the front end of the application, there is still the Database at the backend and processing of the data to the business logic at the middle end to provide meaningful inputs to the front end of the application. There comes the concept of SOA. Not that you need SOA to make a RIA, but it is the other way around where using a RIA can make using SOA easy in the organization.

SOA a powerful concept as it is nothing totally new. It has existed over years, just that the use of it has not been much or at all in the recent web technology based systems. Then the question is why do we need to use it, what are the benefits of using this SOA? Let us try to list down the promises of using SOA successfully in an organization:

- **Reuse:** the key benefit is the ability to reuse the service build for one application over several other future applications that requires the same service. It is not just cutting and pasting the same code as some might think, but more like just pointing the data to be processed by that service and come back without even touching the running service in a server in the organization.
- **Efficiency:** The efficiency of building and modifying an application to respond to the business need can improve tremendously with reuse of services. We can possibly build a new application with reusing several of the existing services of multiple applications and just building the front end and Database.
- **Loose Technology Coupling:** The interesting part of this feature is that we can have several services build on different technology running in different servers and a new application can be made by combining them. The key feature of the architecture is the ability of using services build on different languages, so .net, java and perl all can provide services to build a single application. XML is the language that manages the exchange of information between different services and to provide the output to RIA. XML is technology independent and can interact over several platforms and technology as long as the protocol is followed.
- **Application Integration Ease:** Provided the applications we are trying to integrate are following SOA, integration can become much easier than how it is now.
- **Division of responsibility:** with effective adaptation of SOA for the enterprise, the responsibilities can be made more clear-cut, there will be one group who focuses on building the services as and when required. Another group is the 'consumer' of the services, basically this group focuses on the business more and focuses in using the services to create optimal business systems.
- **Reuse/continuation of use of Legacy Application:** so in SOA we are not only talking about reuse of services build for SOA but also possibility of using the existing legacy applications in the organization that still serves the requirement. So we do not need to junk the old applications just because we are using any new technology, this can end up saving millions and plenty of headache. Through a technique called wrapping legacy systems can be made to run seamlessly with new services as part of the system.

Ultimately the above two technology by itself and through the combination can bring better return on investment from technology and make applications more user-friendly and adoptive to business needs. More and more the focus can be to the deliver of requirement as per the need of business rather then adapting to what technology can provide. It is towards business to win back some of the power over technology through the use of technology.

Should there be issues you want clarified regarding the subject matter, please do not hesitate to contact the author:

*Prof. **Arup Maity**, PMP, MBM
President & CEO – Blastasia, Inc.
Professor, Asian institute of Management
Member PMI
E-Mail: arup@blastasia.com*