



**Robert L. Bogue**  
MS MVP, MCSE, MCSA:Security  
317-844-5310  
Rob.Bogue@ThorProjects.com

## How SharePoint and SOA Fit Together

---

What do you get when you take SharePoint, a wildly popular product, and Service Oriented Architecture (SOA), a widely misunderstood but sought after technique for improving the agility of an IT department, and put them into a single article? Hopefully you get a popular article where people poke their heads in to learn if this is another combination like peanut butter and chocolate, or if it's more akin to spinach and strawberries.

Services Oriented Architecture is a perspective on how to build systems that emphasize reusability, loose coupling, and standards-based approaches. SharePoint is a specific portal technology that is focused on getting information into the hands of the users who need it as quickly and as simply as possible.

In this article, we'll explore how SOA and SharePoint work together to help organizations that are struggling with minimizing costs and maximizing benefits.

### SOA as Plumbing

I once had a house with plumbing so bad that it nearly needed to be ripped out entirely and replaced. The galvanized pipe that was used in the construction of the plumbing had quite literally been rusted through over the years. I awoke one morning to find an arc of water shooting over a light on my back porch.

After a few thousand dollars and a few weeks I had nearly completely new plumbing for the entire house. The problem was that other than the basic fundamental of having water pressure again there was very little — if anything — that I could point to that showed that all of the work had been done. When the plumber asked for his check, I paid him, but somewhat reluctantly since I felt like I got very little visible changes for the work that had been done.

A push towards SOA is much like the experience I had with my plumbing. Very little of what we see on the surface changes during a SOA initiative. Very little is known about how SOA is going. Sure, there's evidence that meetings are taking place and people are talking about SOA in the halls, but measuring real progress is very hard to do.

### SharePoint as a Faucet

When viewed in the same analogy, SharePoint is the faucet. It's the item that I reach out and touch when I interact with my plumbing. When I changed out the old plumbing for new plumbing, I didn't change the faucet. It was the same one I always used. So even when all of the underlying pipes had changed, nothing had changed from an interface perspective. Conversely, I recently changed a faucet for my wife. While the underlying plumbing hasn't changed the user interface, the faucet, has.

Instead of barely noticing the difference for thousands of dollars for a few hundred, there were radical visual changes — changes that were quickly apparent and appreciated.

SharePoint is the visual face to your infrastructure no matter what that infrastructure is. Its ability to consume Web services and display content allows users to see what is going on in the back end.

SharePoint and SOA fit together just like the pipes and the faucet.

### SOA as a Platform

Obviously, a Services Oriented Architecture (SOA) strategy is not plumbing. It is a strategy where you bring together disparate systems in a way that is loosely coupled and structured. SOA takes a different view of loosely coupled systems than traditional programming methodologies.

Development has evolved through several different methodologies. Procedural development focused on developing software into procedures that created reusable units of code. Procedures are layered so that one procedure calls another until the unit of work is so small that it is actually completed by the procedure. Procedures moved development forward from the unstructured mess that existed before. Procedural development focused on making code reusable within the same system. Most developers could readily make the leap into procedural development.

Object Oriented Development (OOD) focuses on coupling the data and the code necessary to work on that code. Objects are then arranged in relationship to each other. OOD relies upon small units of cohesion and loose coupling between different objects. Loose coupling means that changes in one object don't negatively impact objects that use that object. In other words, small pebbles dropping into water remain small pebbles dropping into water — they don't turn into the first domino in a chain reaction. OOD is focused, as procedural development was, on code within the same system and occasionally on systems that support the same remote object technology. Unfortunately, fewer people were able to completely understand object oriented design principals. The result is that not everyone made the leap into object oriented development.

Several variants of object oriented development have emerged, in part due to the challenge of getting developers to understand how to do OOD. One variant, test-driven development, is favored by agile methodologies. Test-driven development (TDD) focuses on creating tests for the kinds of behaviors that an object should exhibit.

A different view of test driven development is a fundamental perspective underlying SOA. Instead of developing tests and coding to the tests, SOA emphasizes contract first development. This form of development, which is similar to TDD in the way that it doesn't start with developing the code, is called contract-based development. This methodology defines the contract that will exist between the consumer of the object (or service) and the provider of that service. The contract is the arrangement of the data that will be sent and received by each party.

SOA manages these contracts into a cohesive network of messages that move from one component of code to another; whether this network of contracts is created with a contract-first development approach or whether those contracts were introduced into existing processes in order to support the creation of a SOA.

The SOA methodology premise is that once a contract is defined it can be consumed by any component capable of generating the request and understanding the response specified by the contract. SOA creates a cross-system level of reuse that previous approaches didn't attempt.

### **SharePoint as a Consumer of Contracts**

Once a contract has been established — once the interface to a service is known — it is possible for other components to consume those services. SharePoint is a platform designed to consume information so it can be displayed. It is a tool specifically designed to facilitate the visualization of information. In other words, it is designed to consume the same kind of information that SOA generates.

SharePoint ships with several Web Parts designed to help integrate to existing systems. Web Parts like the XML Web Part and the page viewer Web Part are designed to consume content from other systems and display them as a part of SharePoint.

Trivial custom Web Parts can be built, which implement the contract necessary to consume a SOA service, or alternatively a generic Web Part can be created that will allow you to quickly configure the Web Part to consume any service in the SOA library. In either case, assembling a visual interface for SharePoint on top of a SOA architecture is quick, easy, and effective.

As a point of fact, one CIO described SharePoint as "The user interface for our SOA effort. We've been working for 14 months to get SOA off the ground and SharePoint is converting the vision to a reality." SharePoint has the power to put a face on to an otherwise faceless initiative.

## Conclusion

If you're committed to doing a service oriented architecture initiative within your organization; if your group has seen the vision that SOA can provide; then adding SharePoint to the mix can help others see that vision as well. If you're looking for some visible part of the plumbing that you're putting in, consider adding SharePoint to give the users a way to interact with your hard work.

## About the Author

Robert Bogue, MCSE (NT4/W2K), MCSA:Security, A+, Network+, Server+, I-Net+, IT Project+, E-Biz+, CDIA+ is the president of Thor Projects LLC. He has contributed to more than 100 book projects and numerous other publishing projects. He was recently honored to become a Microsoft MVP for Microsoft Office SharePoint Server. Before that Robert was a Microsoft Commerce Server MVP and before that Microsoft Windows Servers-Networking MVP. Robert blogs at <http://www.thorprojects.com/blog> You can reach Robert at [Rob.Bogue@thorprojects.com](mailto:Rob.Bogue@thorprojects.com).