Implementing a service-oriented architecture for earth resources information

Gavin Stilgoe, Linda Bibby, Boris Rudoy[1]
[1]Department of Primary Industries (Victoria), Melbourne, Australia, Gavin.Stilgoe@dpi.vic.gov.au
[1]Department of Primary Industries (Victoria), Melbourne, Australia, Linda.Bibby@dpi.vic.gov.au
[1]Department of Primary Industries (Victoria), Melbourne, Australia, Boris.Rudoy@dpi.vic.gov.au

INTRODUCTION

Earth Resources Division (ERD), within the Department of Primary Industries in Victoria (DPI), is the steward of vast geoscience data and knowledge holdings in a range of corporate databases on behalf of all Victorians. Since 2006, ERD has attempted to change how it delivers its information, so that it delivers as much public information as possible, in a way that provides both syntactic and semantic interoperability. The solution is based on a service-oriented architecture layer that exposes data from a range of existing and new applications across multiple technology platforms (Figure 1).

IMPLEMENTATION

The SOA layer is composed of standard W3C compliant web services, OGC Web Feature Services, Web Map Services and an Enterprise Service Bus. The layer sits over existing and new systems across multiple platforms (JEE and .NET).

The aim of the SOA layer is to recognise common business functions, and allow any other systems access through this business logic, without the need for the continual re-development of that logic, or copies of data in separate business applications. The strength of this approach is that it allows us to present a common corporate schema of our data, even though it is currently being sourced from many disparate systems. Implementation of SOA will allow ERD to align data assets to business services in a standard, flexible and architected fashion.

A major new project commenced in 2006 that created a JEE-based system for information associated with geological mapping. The new application has helped ERD to improve data quality by removing as much free text as possible from the data, and integrating spatial and aspatial applications. The new application relies heavily on controlled vocabularies and schemas to handle the science and quality requirements, which in turn impacts existing systems.

As the Vocabularies and Schemas became more sophisticated to handle science and quality requirements, we had to change our non-JEE applications. We used a SOA approach, with an abstraction layer to deliver the data and functions between the data layers and the applications, to use the information contained in the new vocabularies, and allow for schema changes overtime without requiring more changes in the older applications. ERD non-JEE applications (OpenROAD) can now access web services, and these services have been expanded so they are not only data transfer mechanisms, but also provide access to functionality that exists in the non-JEE applications, like vocabulary functions ('is the Jurassic Period younger than the Cambrian Period?'), and transformation functions ('return Latitude/Longitude in GDA94, given Latitude and Longitude in AGD66').

The services have strong science benefits, such as the ability to merge data from traditionally disparate industries. For example, boreholes from the minerals sector and wells data from the petroleum sector to support regional 3D modelling. The minerals vs. petroleum sector focus and associated difference in IT systems common throughout government means that this is the first time we are able to present the data together in a seamless way.

Our clients will be able to access services directly to obtain data they want, in addition to the existing products available via our Portals (e.g. GeoVic). The benefit for us in this approach is that we don't have to create and maintain a Portal that meets all conceivable use cases, a task that is neither practical nor possible. Instead, we can provide a portal that consumes our services to satisfy many of our clients with a few key use cases. In addition, clients can build their own applications that consume these services to meet their requirements.

Ability to capitalise on agility of ICT development is another important benefit of the SOA implementation. Clear architecture and separation of concerns will make agile methodology welcome. We can re-use components, which in turn reduces cost and allows rapid return on investment, creating a highly adaptable environment to changing requirements.

CHALLENGES

We learned that the most important requirements for SOA had nothing to do with technology! What we needed most was a clear vision of what the end looks like, a method to get there and a strong leader to keep the team focussed on the big picture. SOA implementation is complex, and requires skilful software engineers and business analysts with a good grasp of modern methodologies.

SOA also added a new dimension to our system design that at times proved difficult for both developers and users to comprehend, which affected how quickly we were able to implement. In addition, once we decided to implement SOA, the question of how to implement created vigorous debate. This was a completely new concept for us, and there were a lot of opinions as to the best way to achieve our goal. The SOA approach creates a lot of overarching documentation, which must be read in conjunction with other documents. This was not something developers who are used to working on focussed development were famililar with, or readily able to adapt to. The implementation also suffered from staff turnover, which affected buy-in and eroded the knowledge base of the group.

Some existing applications (built on OpenROAD) and COTS solutions (ArcGIS and DBMap) do not easily lend themselves to SOA, or would require massive program changes to implement services. Our approach for these applications is to use an open source data integration tool (xAware) to enable web service access to data via a data mapping process, rather than a programmatic approach. This approach provides a consistent interface to all applications, and allows us to create web services from existing applications without the need to write and maintain code, or to change the underlying applications.

There is a constant tension between using systems and applications that are fit-for-purpose and a single solution that minimises the number of systems and applications we need to maintain. Our ideal is to move most of our business systems to JEE with a Google Web Toolkit front end, to allow for rapid development on a common platform. However, there is also a need to integrate specific applications, such as an ontology manager for our controlled vocabularies or specific GIS software for sophisticated cartographic activities, which increases the complexity of the underlying systems.

Although ERD has a long way to go to realise its dream of all public geoscience data available as web services, the Services approach offers unprecedented opportunities for clients external to DPI to access public data. It could pave the way for Victorian Government to embrace SOA across all datasets using a simple model. Unlocking large datasets for use in many ways, as part of normal business, is a goal worth striving for.