9A.3 A new production process based on a loosely coupled architecture

Peter Trevelyan, Alan Radford, Graham Mallin, Jeremy Tandy, Bruce Wright *
The UK Met Office, Exeter, UK

1 INTRODUCTION

Over the last few months the Met Office IT Architecture Team have been investigating the use of a service oriented architecture (SOA) to deliver the new IT infrastructure required to support the Met Office business. This has lead to a number of key projects being undertaken in order to demonstrate the effectiveness of such an approach. While these initiatives stand alone and provide business benefit, the greater agility and flexibility required of the Met Office in the future can be enhanced by the use of a full ‘end to end’ SOA. This project was designed to be a vehicle for demonstrating a full SOA while taking the opportunity to investigate and prove key core technologies for the future.

Fundamental to these core technologies is a ‘net-centric’ approach which will aim to use web-based standards and services where possible. In addition, a core requirement will be to use as much open source software as possible.

The project was a ‘proof of concept’ R&D activity which was not intended to deliver operational components but provide a platform for future developments in this area.

There are number of issues/problems in the whole area of production that have been outstanding for a number of years, mainly because they have proved very difficult to solve. A list of the top ten issues are itemised below:-

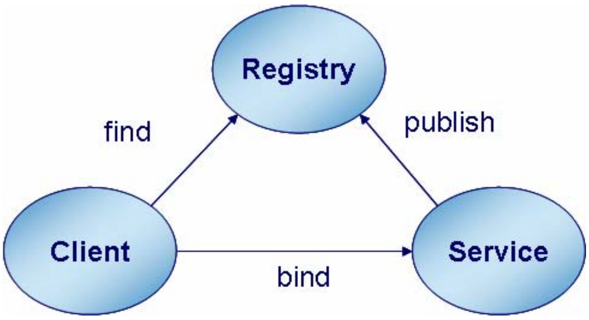
1) The ability to be able to display a set of basic products from a set of standard building blocks.
2) The ability to be able to create/prototype new products quickly and simply i.e. without application programming effort.
3) The ability to be able to monitor who gets what and when.
4) The ability to be able to employ GIS functions as part of a valued added addition to any product including exporting of data in GIS formats.
5) Ability to provide route (or line), point or area specific value added products based on customers assets / infrastructure
6) Perform a dynamic threshold analysis e.g. roads flash red when weather exceeds certain thresholds.
7) The ability to incorporate customer data into products such as sensor data or customer specific algorithms.
8) The ability to integrate the provision of a climate data into products e.g. an anomaly chart.
9) Product consistency i.e. visibility of where a product is sourced and a non programmatic change of data source.
10) Handle data of different resolution with no change to any software including high resolution GRIB data.

The objective of the SOA project was to develop a strategy that will be directly aimed at solving some or all of these problems. Once this strategy has been proved then a number of follow on projects will be spawned with the specific purpose of implementing this approach and hopefully ensure that we do not go down technological bolt holes and therefore waste time and money.

2 SERVICE ORIENTATED ARCHIETURE CONCEPTS

SOA is an Architectural Paradigm for software systems. Service Oriented means that components wait in a state of readiness and may be “called” or “invoked”. SOA has roots in Object Oriented (OO) programming methodology. The following diagram illustrates the “publish-find-bind model”

* Corresponding author address: Peter Trevelyan, Met Office Fitzroy Road, Exeter, Devon, EX1 3PB United Kingdom. +44(0)1392 886521, email: peter.trevelyan@metoffice.gov.uk



The SOA "Publish Find Bind"Model

2.1 The main components of a S.O.A

The main components of a S.O.A are:-

1) Service registry:

2) Service description:

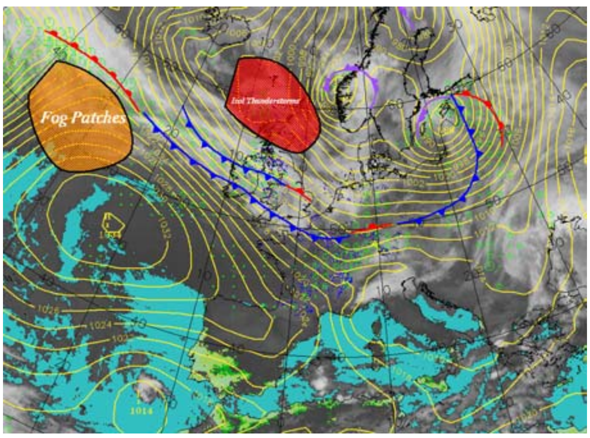
3) Client that searches the Service registry

4) A binding of the client to the service:

Service oriented architecture offers the promise of delivering business agility, but implementing SOA is journey. It cannot be accomplished in a single project. The key is to have a plan that delivers value in the near term, and continuing agility in the long term.

2.2 What do we mean by a "Service"?

Essentially a service is work done by an entity that benefits another. Granularity is important as this will dictate the flexibility and overhead of the service to service messaging. We have taken a fairly large scale approach, and to understand this we need to look first at our business processes. Consider the following graphic. It has a map background some contours, sensible weather objects created by a forecaster and a legend including a logo. Each of these layers required a service.



A Typical Chart Showing the Layered Structure

2.3 What can S.O.A Deliver:

Achieving efficiencies in our current application portfolio can free up additional money and resource for delivering new business value. By adopting a Service-Oriented Architecture (SOA) for new development, we can adopt a standards based approach to enable the re-use of existing application functionality to support new business processes, thus reducing development time.

2.4 How can we encapsulate a service?

We need to create an environment that allows software to "plug in" and this is where web services fulfil their potential. Web Services are designed to allow applications to interoperate by passing standard messages across a network using standards based interface. These interfaces can take a variety of forms and in this project the web services standards were taken directly from the OGC ("Open Geospatial Consortium) and will be discussed later.

3 PROJECT SCOPE

In order to facilitate the key objectives above, the following tasks were required. This task list will of course be refined during the project:

1) Identify key external partners with which to work;

2) Model the business processes that make up the product generation suite.

3) Design and build a SOA to deliver an end to end prototype product generation system.

4) Modelling a subset of meteorological features from the Lightning domain (Lightning observations, Lightning risk areas, radar, satellite and Lightning sensitivity) in XML using the GML profile.

5) Asses the feasibility of representing other meteorological features such as fronts, observations, contours and gridded data in GML; specifically:

   - Outline meteorological domain model expressed in UML; and
   - Express representative objects of the model in GML.

6) Investigate how easy the integration of existing Met Office legacy assets within the SOA would be.

7) Integration of existing Open Geospatial Consortium services such as Web Feature Service (WFS), Web Mapping Service (WMS), and the Web Coverage Service (WCS).

8) Produce reports and presentations:

- Comparing and contrasting the manageability, supportability, performance, agility and flexibility of the SOA with that provided the existing production infrastructure.
- Identifying how elements of the Lightning system can be most effectively be migrated to an SOA;
- Highlighting whether the proposed architecture has been de-risked by implementing key components of the solution and verifying their suitability (e.g. performance);
- Identifying where open-source software can deliver appropriate components for the architecture;
- Lessons learnt;

4 MODELLING THE BUSINESS PROCESS

The business process behind product creation is well understood, but is poorly modelled using our current IT architecture. The following outlines what the processes are and how they will be modelled using a registry and a SOA. In order to model our production process we need to:-

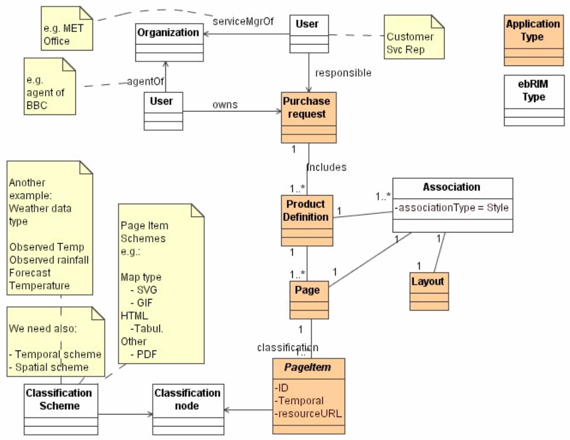
1) Create an itinerary that will drive the production process by extracting not only customer driven information, but in terms of the services needed to create the content. The information model needs to be supported by services that allow for "multi rule" queries and be properly governed i.e. content needs to be "registered"

2) Create an information model that will provide information to support a set of queries that together will support a range of business functions. These fall broadly into two categories; first to drive the process chain managed by the ESB (Enterprise Service Bus) and secondly provide information to the service/account managers etc. For the prototype project we have chosen to use the ebRIM registry model from OASIS.

3) Model documents in terms of what content is required e.g. text, graphics etc and the services that need to be invoked (including the service binding) in addition to the associated final document layout and format.

4) A "customer portfolio" that contains a list of all "page manifests" required by a customer (including associated page layout).

The following diagram is a sample of the UML used to model the information contained within the registry.



The UML Used to Model the Document Definition.

The registry is at the heart of the new production process and further work will be carried out in order to refine the information model. A client based on the WRS (Web Registry Service) that allows simple queries and changes to made against the registry.

5 OGC INVESTIGATION

The OGC have developed a number of services that enable the transfer of features, maps, grids etc in a standard way across a network using Internet protocols. The following web service interfaces have been defined by the OGC.

5.1 The Web Coverage Service (WCS)

This supports electronic interchange of geospatial data as "coverage's" – that is, digital geospatial information representing space- varying phenomena. A WCS provides access to potentially detailed and rich sets of geospatial information, in forms that are useful for client-side rendering, multi- valued coverage's, and input into scientific models and other clients.

5.2 The OGC Web Map Service (WMS):

This supports the provision of images using a standard interface that allows the client to specify geospatial and temporal contexts. This is very powerful as used properly it solves much of the geo-referencing problem. The term images cover any information that can be mapped i.e. rendered.

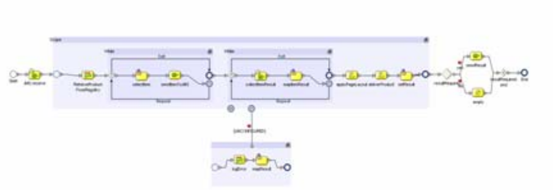
5.3 The OGC Web Feature Service: (WFS)

Enables a client to retrieve and update geospatial data encoded in Geography Markup Language (GML) from multiple Web Feature Services.

The WFS and WMS services were used within the project and the WCS was looked at.

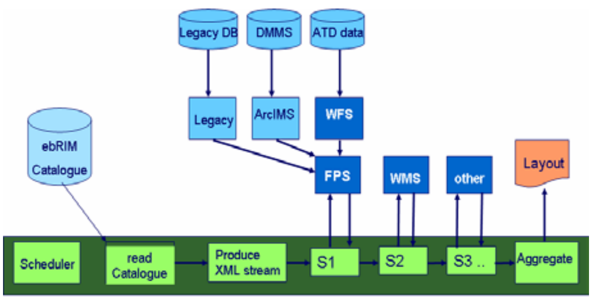
6 S.O.A ARCHITECTURE

The heart of the architecture is the concept of invoking, chaining and orchestrating services. Each service is used via an interface and the outputs from all the services aggregated to create the desired content. Products are layered and each layer created or obtained from a specific service e.g. radar image, contoured PMSL data etc. Orchestration was undertaken by an ESB ("Enterprise Service Bus") using OGC web service interfaces. These processes were modelled graphically:



A Simple Model of Service Orchestration:

A registry held all information required to initiate a production process and a number of existing assets were reused after having been web enabled using OGC Web Services.



ESB (Enterprise Service Bus)

The services (denoted by S1, S2 etc) are characterised by the interface or binding i.e. the information they require and the information they return. These "service binding" contracts are held in the registry that is based on the ebRIM standard, rather than the more traditional UDDI. There are a number of reasons for this, but the ebRIM has been adopted by the OGC and has a richer data model than UDDI.

The process of events goes something like this:-

1) The scheduler reads the catalogue and looks for products that need to be created.

2) The product definitions are read as a series of XML documents that describe the services needed to create the content.

3) The XML "contracts" are parsed and the appropriate temporal information added. These XML "messages" are passed along the ESB and the relevant services invoked in turn. Service 1 (S1) refers to a feature portrayal service (FPS) that fronts an ESRI map server and lightning data. The other services i.e. S2 and S3 are generic.

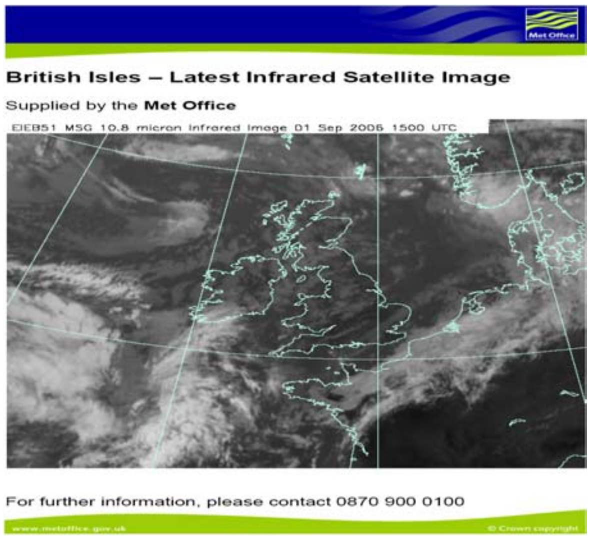
4) The content is aggregated into one image (if this is what was required) and a page layout applied to create the document.

5) Any error messages are sent to a monitoring system.

6) The page ready output is disseminated to the customer in their preferred format.

7 LAYOUT AND RENDERING:

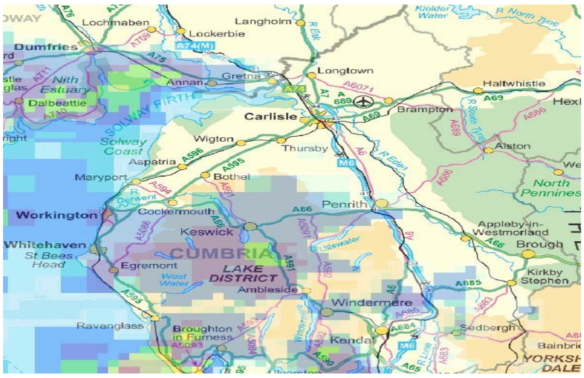
A number of possible options were considered, but for the project the use of "Open Office" and "Open Document Format" proved to be very powerful. The "open Document Format" ODF is XML based and the "Open Office" API was to write a layout and rendering engine. The following picture was produced using Open Office



8 INITIAL RESULTS

8.1 SOA as Architecture:

All elements of the architecture were built and tested to form a complete production chain. The architecture proved to be very flexible and images from a number of sources were merged to form content that would have been very difficult to create using our existing frame work. Products such as the one below although looking very simple would have been very difficult to create using our current infrastructure. By using a SOA it has become a relatively easy task.

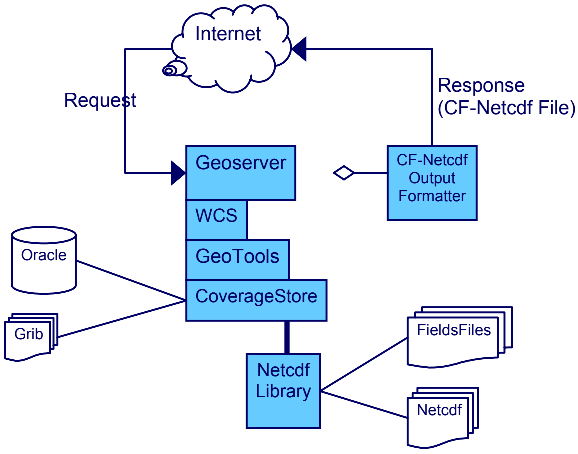


A Radar Image Overlaying High Res Map

8.2 OGC Web Services:

The OGC web services proved to be very powerful and using existing applications (ESRI's ArcIMS) was easy and quick. Web enabling legacy applications proved to be relatively easy and have tremendous advantages. Previous attempts at radically changing our architecture were unsuccessful as it meant "slash and burn" approach, but the SOA permits evolution rather than revolution.

Some work has been carried out with the Web Coverage Service using the Open Source project Geoserver. Using Geoserver gridded data held on the mainframe has been made available across the network as NetCDF files, even though the original data is held as indexed GRIB files. The fields' selection uses all four dimensions so allowing realistic sub selection of grids. This work was not particularly difficult to carry out, but the benefits are large. The following diagram depicts the main components of the architecture.



WCS Deployment using Geoserver

8.3 Registries:

Storing the full information model (customer and document description in terms of services and service bindings) in the registry has taken some real thought, but it has become clear that such an approach has tremendous potential and will mean that the content and layout of documents can in many cases be made by simply changing the registry. Below is an example of a radar image overlaying a high resolution map.

8.4 Benchmarking:

The performance of the SOA is vital to its adoption and work is being done to understand how much of an overhead the ESB is. Initial results are encouraging and there is plenty of scope for load balancing.

8.5 SOA and Modelling Production

The modelling of the production process was quite straightforward as SOA allows a much better "real world" abstraction of the production process. One of the real benefits of SOA is that it greatly simplifies quite complicated and convoluted architectures.

8.6 Data Integration and SOA

It has become apparent that a lot of time has been spent in simply aggregating data. There have been a number of attempts to solve this in the past, mainly through the adoption of data formats. Whilst this has some benefit it still does not make data available across the whole organisation i.e. it is more than often parked in a specific silos. Web services (enabled through standard interfaces e.g. OGC) have transformed the whole data aggregation problem and the policy in the future will, wherever possible, to keep the data at source and access it across the network. This of course has bandwidth issues and work is taking place to ensure that the network enabled databases support a number of services such as data grid sub-setting and the rendering of data e.g. creating images of pressure contours. Indeed most requests for data are in truth requests for images of the data.

9 FUTURE WORK:

In order to bring this work to operational maturity further work needs to be undertaken. A set of pilot projects will be set up to take the work further solving some real business problems. One area in particular that will be looked at is web delivery as much of the really hard work in presenting data over the web has been the difficulty of data aggregation. With proper use of SOA the data aggregation will be solved, once and for all, (performance not withstanding) and much more effort can be concentrated on the presentation. The advent of "Service Orientated Clients) e.g. Adobe's Flex and a WPF (Windows Presentation Foundation) provides a natural way of exploiting a SOA and creating a true rich internet application. Another natural consequence of this work is combining data from different sources e.g. temperature data and route in order to create an added value layer e.g. routes colour coded by temperature. Up to now this has been the sole province of GIS systems, but in the SOA world the services (in this case a geoprocessing service) can be accessed across the network. Evolving standards such as the WPS (Web Processing Service) should help with this work.