

EASA and Service Oriented Architecture (SOA)

Many IT projects require multiple software applications to communicate with each other. These can be running on different hardware platforms and are often from different solution vendors.

The best practice for meeting this requirement is to expose each applications interface as a service to hide both the underlying implementation and hardware infrastructure. Web services are the most common choice of this service and have the benefits of being both vendor and platform neutral.

In order to manage the interaction between these services, SOA architectures are being used. These provide mechanisms for security, quality of service, and orchestration of the overlaying web services.

EASA can benefit SOA systems in two significant ways:

- 1) Wrapping existing software to expose it as a web service
- 2) Rapid creation of sophisticated web enabled user interfaces.

Wrapping Software

Creating new software that appears as a web service is straight forward. Exposing existing software as a web service is more complex. Rewriting the older software can be an option but is typically expensive and error prone.

A better solution is to wrap the software in a layer which exposes the web service and this is the approach that EASA takes.

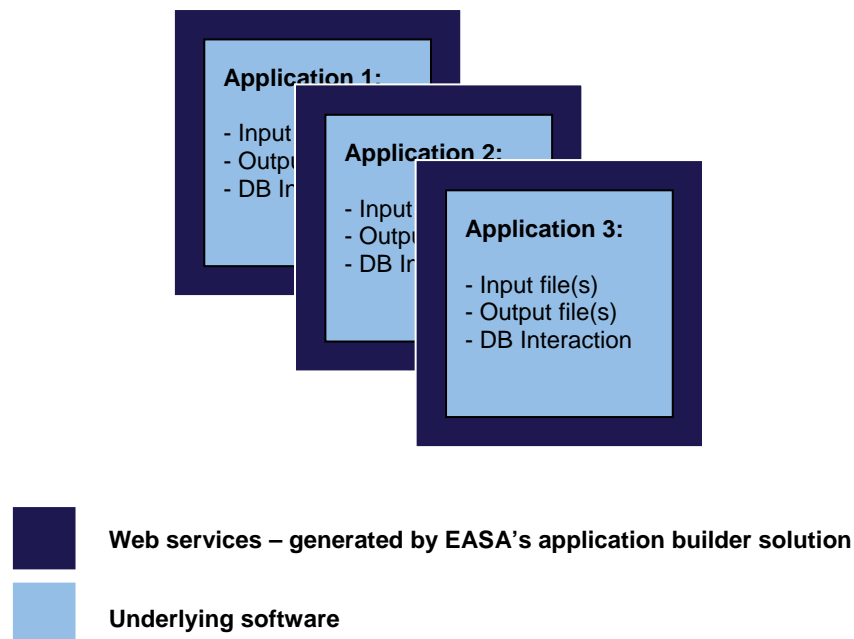


Fig 1: Wrapping applications as web services

EASA is particularly powerful in wrapping software that takes significant time to run, for example high end modelling tasks such as Monte Carlo simulations or solutions of non linear systems. A case study exploring this class of SOA problem is given in Figure 2.

EASA does not run computationally intensive calculation on the application server that is exposing the web service, but instead queues and executes the calculation on separate dedicated computing resources.

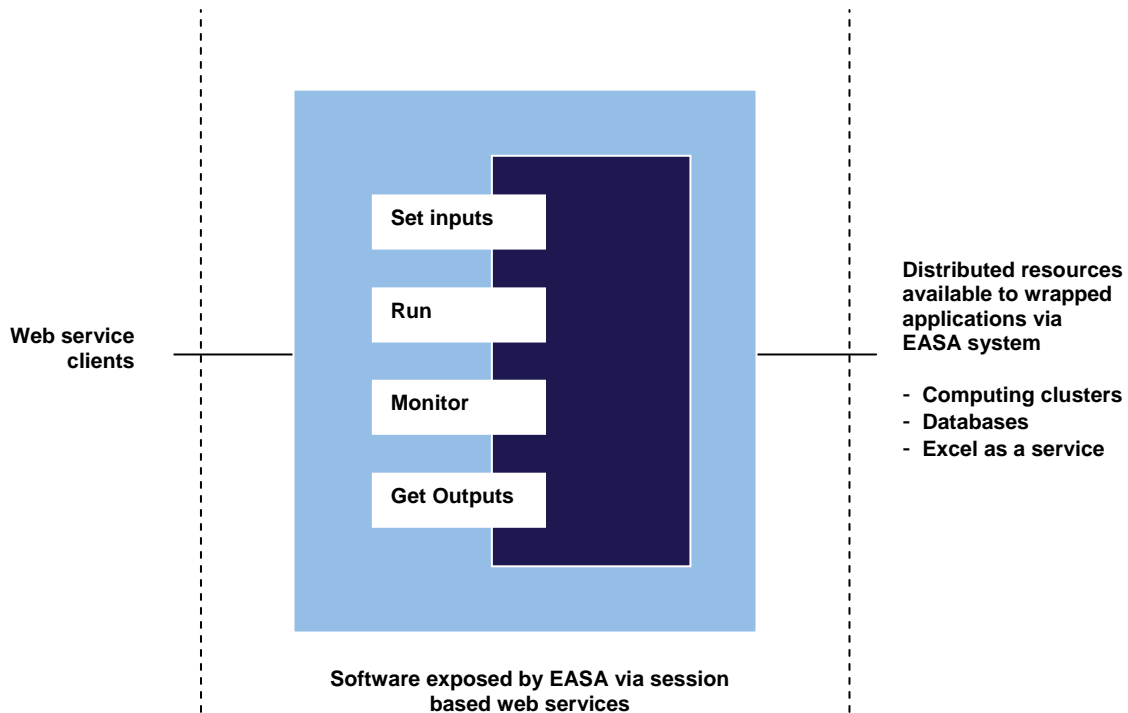


Fig 2: Services exposed by the wrapping process

Therefore the web service client does not need to wait for the analysis to complete, but instead can make web service calls to periodically check on progress and obtain interim results.

For example, in an engineering simulation, the wrapping process could allow for error checking to take place on the analysis output file in “Real time” and information regarding the runtime status could be fed back to the web service client.

Sophisticated User Interfaces

As the number of customer specific web services available through an SOA grows, so does the potential for creating highly customized user interfaces to meet specific business needs.

EASA allow components to be created that represent specific parameterized web service calls. These codeless components can then be combined to form high quality web enabled user interfaces with built in support for error checking, diagramming, multi-language support, save/load state etc.

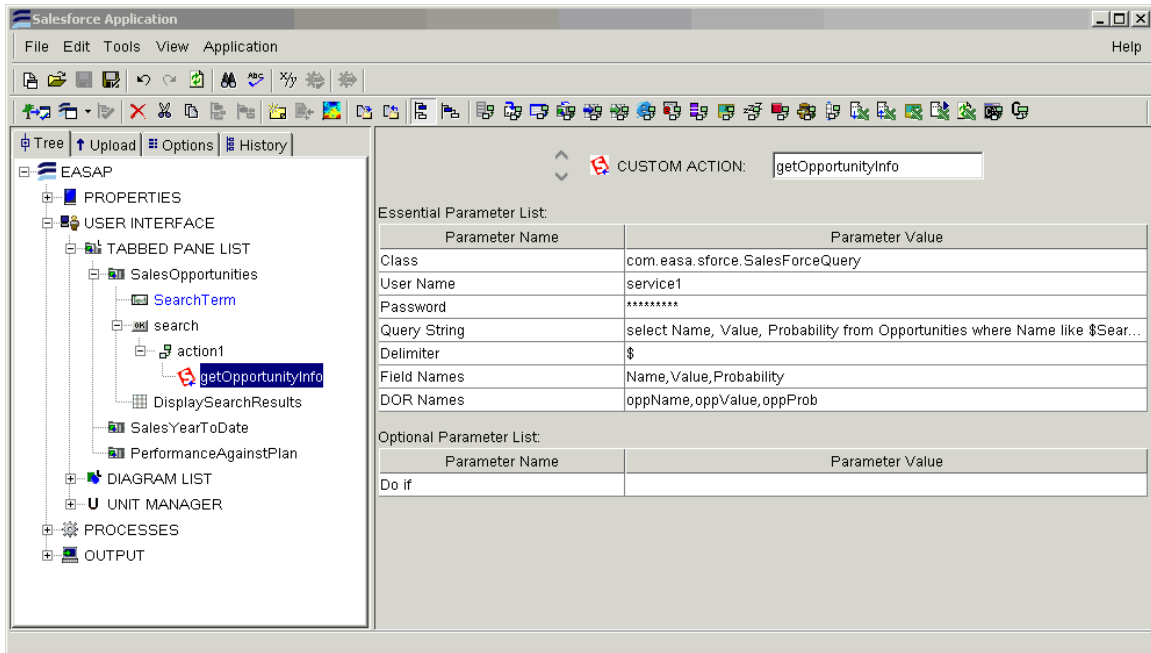


Fig 3: Application builder tool for creating GUI's from multiple web services

The user interfaces are easy to maintain and enhance in line with emerging business needs. Different groups within the business can have their own user interfaces that are tailored to their specific requirements, allowing companies to increase both the usage and value of their overall SOA investment.

References

1. Understanding Enterprise SOA – Eric Pulier, Hugh Taylor
2. Scaling SOA with Distributed Computing – DDJ Journal, Nov 2006