# Enterprise Services Repository: Importance in Enterprise SOA Architecture

## Applies to:

SAP NetWeaver Exchange Infrastructure experts, eSOA architects

## Summary

This Article explains about importance of Enterprise Service Repository to support strategic implementation of Enterprise SOA. How Enterprise Services Repository helps companies to allow customers and partners to effectively consume web services from an open and centrally accessed location like UDDI.

**Author:** Rajesh Babu

**Company:** HCL Technologies Ltd

**Created on:** 12 June 2007

## Author Bio

Rajesh Babu is currently working as SAP NetWeaver consultant with HCL Technologies Ltd. He has a total 3.6 yrs of IT experience in SAP. He has been involved in implementing XI integration processes for the past one year. Prior to integration implementations he had been involved in ABAP development.

## Table of Contents

## Introduction to ESOA

In a globally competitive marketing world to drive innovative business processes with out breaking the existing applications (business functionality), companies need to move towards service oriented architecture. This makes enterprise applications as service enabled to meet business changes and support future processes. Partners and customers can consume these services repeatedly across various applications.

SAP name for service oriented architecture is Enterprise SOA which enabled by open Net weaver Platform. SAP Netweaver04's provides composite application framework to compose various applications at the top of underlying enterprise application architecture, and then enable them as services. An Enterprise service aggregates web services which combines business logic and helps in executing end-end business processes. Enterprise services allow applications to expose as web services (based on WSDL). In order to efficiently use web services across various applications and customers, it is essential that companies need to store these services centrally in one repository.

## ESR is the Heart of ESOA

To support strategic service oriented approach SAP provides a new Enterprise Service Repository. ESR is a central repository of information that contains all the services. ESR is a container, stores all the underlying Meta data of application objects like service interfaces and descriptions. The global data types, interfaces and business processes maintained in Enterprise service repository which can be reuse where needed. The first implementation of ESR and its associated editors come in SAP Netweaver04's, which has been evolved from Netweaver Exchange Infrastructure Integration Repository. Enterprise service repository is a design time repository of service objects for ESA. You can model all service design objects for a process and can reuse the data types and service objects which are already maintained. All enterprise services are published in a central Enterprise service repository so they can be used by any body who really need that service includes customers and partners.

## Two Ways of Creating Enterprise Services

There are essentially two ways of creating enterprise services **inside-out** and **outside-in** approaches.

### Inside-Out Approach:

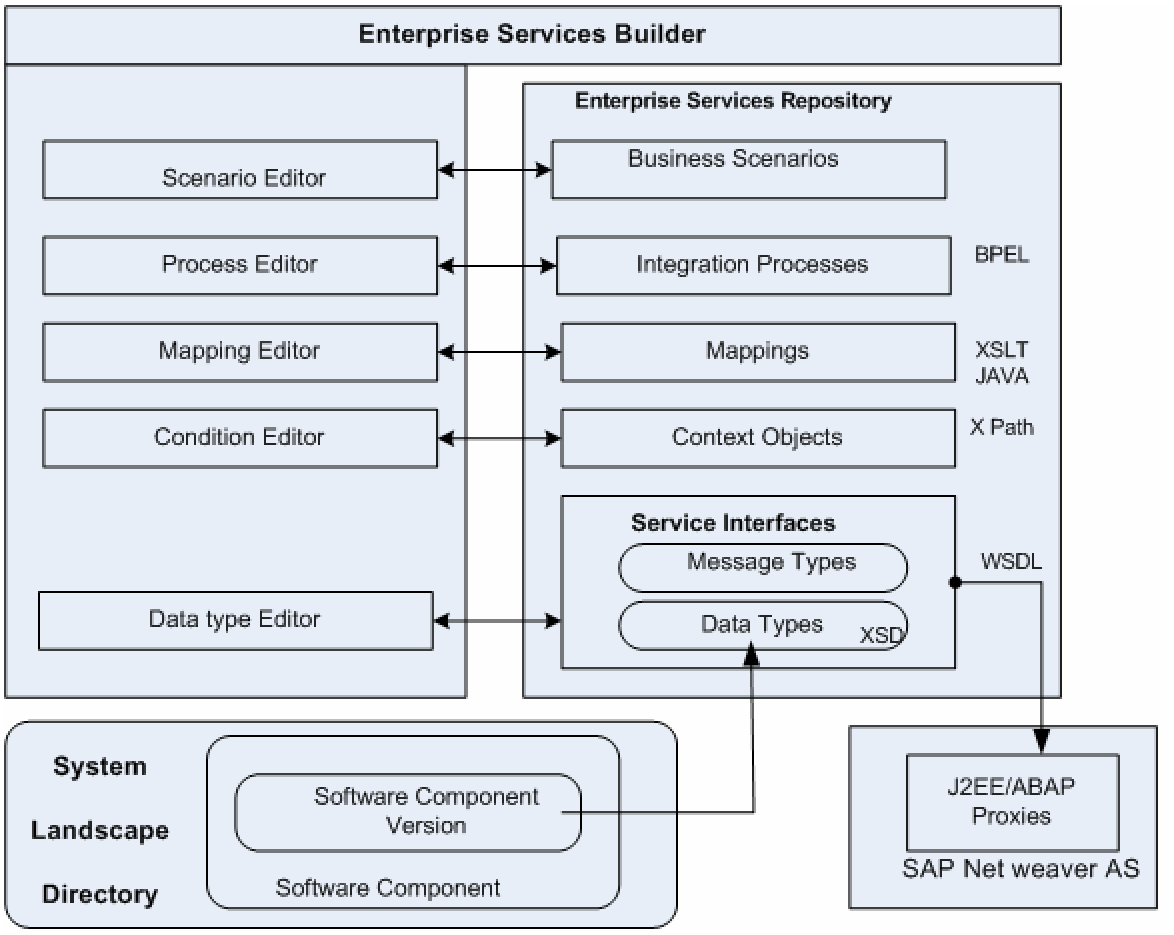
This approach refers with existing application functionality, you start with application development and then service enables that functionality in turn into web service using web service wizard. Inside out development takes an existing business methodology no need to start from business process modeling. You can take an existing application like a BAPI in side and then expose it as web service. Inside out development is helpful and easy approach, it doesn't require any thing new from out side.

### Outside-in Approach:

This approach takes process-oriented, model-driven patterns. Instead of starting with application development, you start with examining business processes and need to implement a strategic business methodology which provides best value. You can then start with modeling service objects like data types and services interfaces, or you can reuse data types which already exist. Out side development starts from out side, understanding business needs and then you can work with application development. This is where Enterprise Service Repository comes into picture. You model data types, interfaces in repository through Enterprise Service Builder. If something you need is not there in repository, you model it there or import it into the repository.

## The Architecture of Enterprise Service Builder and Repository

Enterprise service repository has inherent support from Enterprise service Builder. You can model service design objects through Enterprise service builder that provides various editors for modeling data types and service interfaces. Sometimes all the objects might not available in ESR which needed, then you have to create from scratch or can import from development workbench (ABAP development workbench/ Net weaver developer studio).



Graphical Representation of Enterprise service Builder and Enterprise service Repository.

### Advantages of ESR

Having a central repository offers several distinct advantages, including:

- Orderly Development
- Reuse
- Ease of Development
- Model driven Development
- Service Orchestration

Orderly Development: Having central repository offers an orderly development process. In past a developer looks at a problem and invents solution. There are many ways to solve a problem which leads to lack of uniformity and transparency. Enterprise service repository is based on governance, that allows companies to promote standards like global data types and it brings out uniformity. An orderly approach ensures business value and flexibility to change in response to market conditions without fear that making changes in one area will not break integrations in other area.

Reuse: The presence of central repository enables reuse of elements across different applications. When you are modeling you can scan the repository to see any of the objects can be reused or adopted. You can avoid reinventing and then reduce the entire life cycle. In addition to reuse, services can be adapted and repurposed. The  repository allows adding to services what you needed. Because of the adherence to open standards services created in other systems can also reuse by importing into repository.

Ease of Development: Developing Enterprise services involves the creation of XML files, Web services descriptions. The tools of enterprise service repository generate all XML and WSDL entities automatically when you model data types and interfaces. There is no need to write XML and WSDL files since ESA is solidly built on open standards.

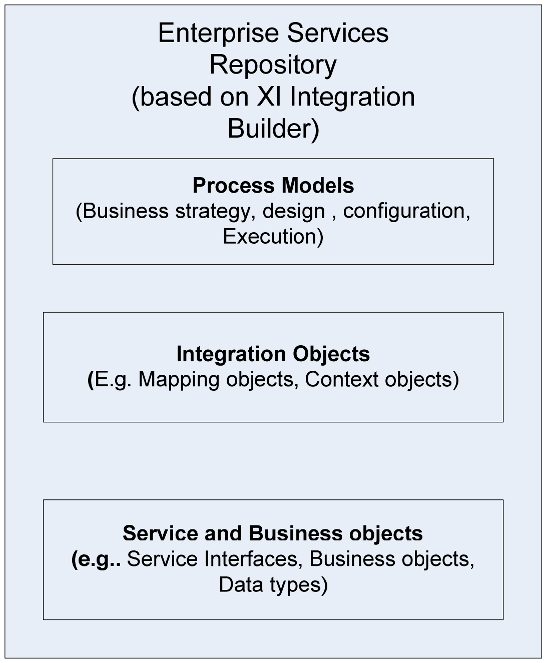
Model-driven Development: The central Enterprise service repository enables a new approach to application development i.e. Model-driven development, in which you start with business processes rather than application development. Enterprise service repository allows editors to model data types, interfaces and then to create strategic services that will support business processes. You will then start generates proxies for services.

Service Orchestration: The Enterprise service repository also plays an important role in Service orchestration. It represents a central modeling layer, which guides implementation from high level business models right down to callable enterprise services. No matter whether you are reusing services or business objects from SAP applications, those created by third party or your own custom development the repository acts as a central hub for coordinating all of these activities and ensuring uniform development, order, and efficiency.

## Elements in ESR

Enterprise service repository has evolved from Net weaver XI Integration Repository. It includes service objects, integration objects, and process models.

## The Main Elements of ESR:


Enterprise Services Repository (based on XI Integration Builder)
Process Models (Business strategy, design , configuration, Execution)
Integration Objects (E.g. Mapping objects, Context objects)
Service and Business objects (e.g.. Service Interfaces, Business objects, Data types)

Data Types: Data types are lowest-level element from which we construct services. Data types are schema objects which defines the structure of XML messages. Data types can be nested. They can be create from data type editor or can import into repository. The use of global data types ensures standardization and interoperability.

Message Types: message types are defined based on Data types that describe actual payload of XML Message. Message types are design time representation of the messages that are exchanged at runtime. Message types are used for message mapping purposes (value transformations from source format to target format).

Operations: Operations are specific actions that a service can perform at runtime. An operation can be synchronous or asynchronous. A synchronous message waits or blocks until it get response from receiver. An asynchronous message is sent and may receive no response.

Service interfaces: service interfaces are high-level representation of XML metadata. They contain Meta data descriptions of messages and operations used at run time. Once you create service interface it contains underlying data type, message type and generates WSDL file. The exported WSDL document can be published either to an UDDI or exchanged with partners in another way (such as HTTPS or encrypted mail).

Integration objects: The presence of design objects like data types, message types and interface objects in Enterprise service repository are used to implement flexible integrations among business partners. Data types are used by for translations between two systems or two XML vocabularies, such as Rosetta Net, an XML standard fro high-tech industries.

Mappings are among integration objects used in integration scenarios. Message mappings transform sender message to the form of receiver. Interface mapping registers a pair of interfaces which used in integration scenario and specify message mapping s to be used at runtime.

Process models: You can model business processes at design time based on process editors in Enterprise service repository. In future versions of Enterprise service repository incorporates ARIS modeling tool which allows dynamic process models. Unlike static models you can import or export dynamic models as needed.

## Conclusion

To achieve a globally competitive market strategy, companies have been moving on to service oriented architecture that enables helpful and flexible solutions to response with market conditions with out breaking the existing functionality and providing support for future applications. The success of ESA relays on proper usage of Enterprise services (technically Web services). So companies required to publish Web services at a central and open place (like Enterprise Service Repository) where customers and partners can be flexibly call whenever they really needed.

## Related Content

For related information, visit:

https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/5e144eb2-0c01-0010-aaa4-fa0e0c12aab2

https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/3b41fe03-0701-0010-798c-b51c54656cc6

https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/706005a3-3bd6-2910-91ae-a2016239bdcf

https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/b5bbf165-0701-0010-a29b-99b86599945d

## Disclaimer and Liability Notice