



A SOA visualisation for the Business

Table of contents

1	Introduction	3
1.1	Abbreviations	3
2	Some background information	3
2.1	The organisation and ICT infrastructure	3
2.2	Five layer SOA architecture	3
2.3	Additional terminology in the SOA architecture	3
3	Implemented SOA visualisation	3
3.1	Central repository	3
3.2	EAI Wiki	3
3.2.1	Chain register	3
3.2.2	Component register	3
3.2.3	System register	3
3.2.4	Public Service register	3
4	Future developments	3
4.1	Integration with performance monitoring	3
4.2	Migration to stable environment	3
5	Lessons learned	3

1 Introduction

In this article, I would like to share some experiences with a visualisation for the business in a SOA environment. MediaWiki is the basis for the visualisation tool because it is a wide spread and well known environment. For this purpose, the normal static MediaWiki is enhanced with some dynamic features to support hyperlinks for the inter-relations between the entities in the SOA environment.

Chapter 2 will give some background information on the environment where this project has been carried out. Chapter 3 explains the solution which is implemented in the past year whilst chapter 4 presents some ideas for future developments. Chapter 5 finally presents some lessons learned.

1.1 Abbreviations

Abbreviation	In full	Description
BE	Back End	A system out of the EAI scope which provides services for front end systems. Usually, it provides a coherent set of services for a specific domain.
BEA	Back End Adapter	A component within the EAI scope with a collection of services which communicate directly with a back end system.
EAI	Enterprise Application Integration	EAI is the strategy, a set of methods and tools for integrating all systems across an enterprise. It provides a common framework for communication and data coordination between applications.
FE	Front End	A system out of the EAI scope which uses the EAI services, indirectly provided by back end systems. Usually, it enables end users to obtain and maintain information stored in back end systems, e.g. the organisations web site.
FEA	Front End Adapter	A component within the EAI scope with a collection of services which communicate directly with a front end system.
FS	Functional Specifications	In this organisation, the Functional Specifications are laid down in a set of word documents based on defined templates. In the EAI department, they are generated by the design tools used.
GFA	Generic Front end Adapter	A generic Front End adapter that offers EAI services on a generic interface (usually SOAP) that is available for all Front Ends.
SOA	Service Oriented Architecture	According to Gartner, a Service Oriented Architecture is a logical architecture where definitive business functions of an application are exposed for programmatic access via a well-defined formal interface, with some means of identifying and locating the function and the interface when it is needed.
SOAP	Simple Object Access Protocol	SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment.
XML	Extensible Markup Language	See " http://www.w3.org/TR/xml ".
VCS	Version Control System	A Version Control System enables versioning, reliability and team work.

2 Some background information

This chapter will give some background information on the organisation and ICT infrastructure relevant for this article.

2.1 The organisation and ICT infrastructure

The organisation in which this project has been implemented is a telecom operator with a large IT department with a focus on the market and a centralised Corporate Services department. A team of about 50 people is concerned with the design of SOA services with an EAI middleware infrastructure.

New development is organised in projects executed by multi-disciplinary teams. The first analysis for a project is done by a team of business analysts from different departments. They base their analysis on business requirements, the information available from previous projects and existing documentation. Mainly for this purpose and maintenance it is important that the documentation describing the existing services and components is accurate and clear.

Because of the different levels of quality of the existing set of documentation, a unification project has been defined to unify the information on the set of services available. The results of this project are presented in the next chapter.

2.2 Five layer SOA architecture

In our view, the SOA architecture can be viewed as a set of five layers as shown in Figure 1. This figure shows a model of the ICT environment viewed from the SOA perspective used in the EAI department.

The explanation of the five layers as used in the EAI environment in this organisation is as follows:

- **Front End**
The Front End (FE) layer represents the front end systems of the ICT environment. Each front end system in this scope can access one or more services in the EAI through a front end adapter.
- **Front End Adapter**
The Front End Adapter (FEA) layer represents the services within the EAI scope which offer their services to the front end systems. A FEA service always invokes another service, either on the Composite layer or directly towards a Back End Adapter service.

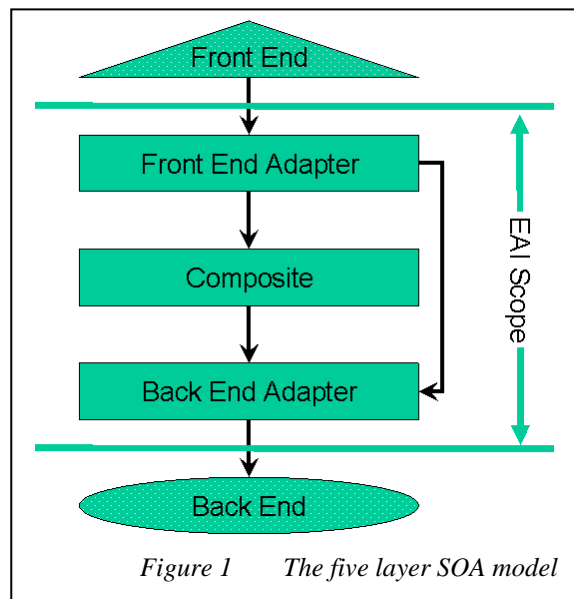


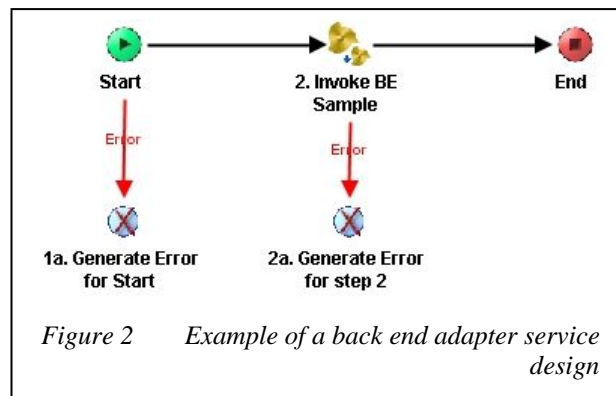
Figure 1 The five layer SOA model

- Composite**
 The Composite layer represents the services within the EAI scope which combine the functionality of other services. A Composite service always invokes another service, either again on the Composite layer or directly towards a Back End Adapter service.
- Back End Adapter**
 The Back End Adapter layer represents the services within the EAI scope which represent the services offered by the back end systems. A FEA service always invokes another service, either on the Composite layer or directly towards a Back End Adapter service.
- Back End**
 The Back End layer represents the back end systems of the ICT environment. Each back end system in this scope can be accessed by one or more services in the EAI through a back end adapter.

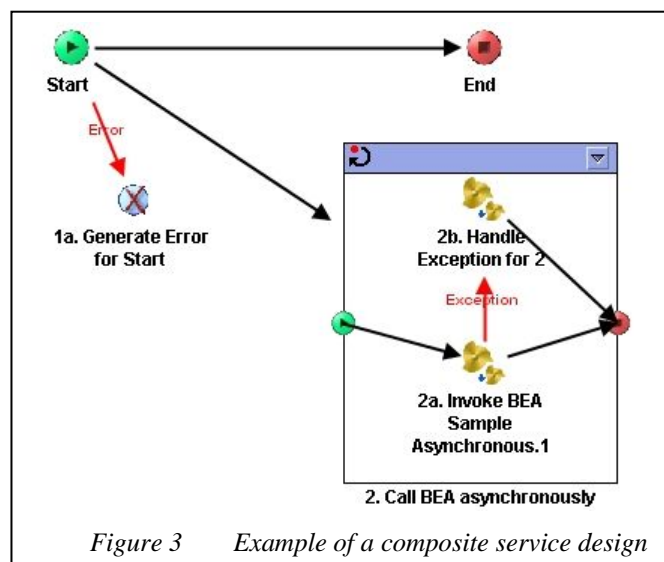
2.3 Additional terminology in the SOA architecture

In the SOA architecture used in this organisation, there are some more terms which need to be explained to understand the solution presented in the next chapter.

- Service**
 A Service is a functional entity with an interface and internal implementation. The interface defines the request message to invoke the service and the response message which contains the result from the invocation. The internal implementation defines how the response is constructed from the request message. Figure 2 shows a simple example of a back end adapter service with a Start and End node and an invocation of another service "BE Sample". Every step in the service design can generate an error which is described separately.



- Composite Service**
 A Composite Service is a service without a direct communication to an external system. It combines the functionality of several other services into a single service. An example of a Composite Service is shown in Figure 3 which asynchronously invokes a back end adapter service (BEA Sample).
- Chain**
 A Chain is a combination of systems and services which together provide a consistent set of functionality. Usually, such a chain is the result of



a (successful) project. An example of a chain is Number Portability which is a combination of all services related to the functionality of porting a mobile number from one operator to another operator.

- **Component**

A Component is a combination of related services which can be deployed together. An example is the set of back end adapter services which all communicate with the same single back end system.

Figure 4 shows an example of a number of services and their relations with other services, components and systems. This set can be seen as a "Sample" chain. The "GFA" services are part of the Generic Front end Adapter component. The composite "PS" services are combined in the "PS Sample" component. The "BEA" services are combined in the "BEA" back end adapter component for the "BE Sample" back end system. Please note that the service "PS Sample" invokes two "BEA" services.

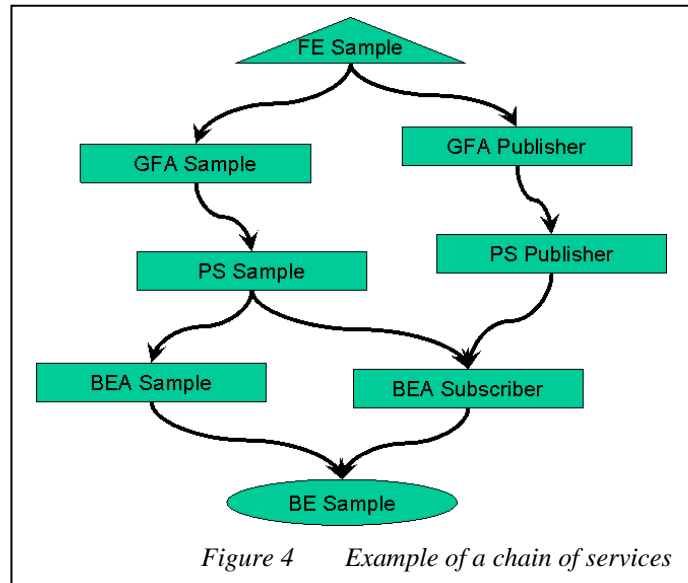


Figure 4 Example of a chain of services

3 Implemented SOA visualisation

The Five Layer SOA Model presented in the previous chapter (Figure 1) is very useful as a reference for developing services in a consistent and incremental way. In this article, I concentrate on the visualisation of the information. The design process uses a central repository to store the relevant information. This central repository is the basis for a number of presentations. Both the central repository and the EAI Wiki as visualisation tool are explained in the next sections.

3.1 Central repository

Figure 5 shows the environment of the Central Repository. The information in this repository is maintained by the EAI Design team. They add new services and update the designs of existing services. The Unification project is started to convert 'old' style designs into the format of the Central Repository. The information itself is stored in a Version Control System (VCS) to enable versioning, reliability and team work. A dedicated exporter tool is developed to generate Functional Specifications (FS) in word documents and the business view in an EAI Wiki which is explained in the next section.

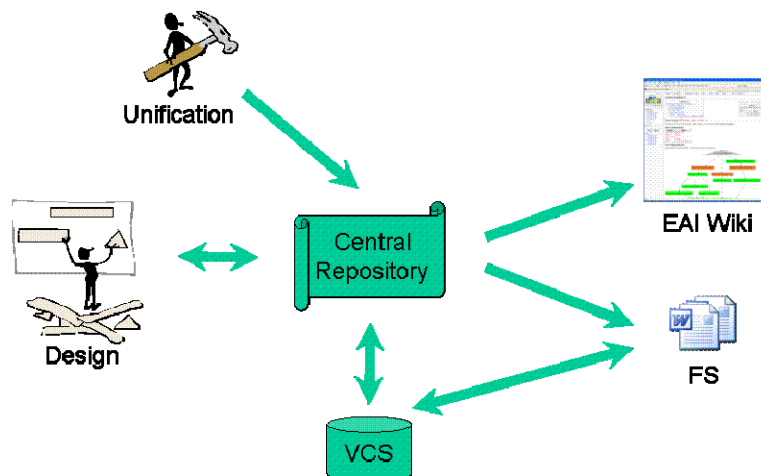


Figure 5 The Central Repository

3.2 EAI Wiki

A Wiki is a web based graphical user interface commonly known from the Wikipedia web site. There are however also a number of organisations who use a Wiki as a basis for sharing knowledge within the organisation. Together with the open (source) environment and ease of use, we decided to use a MediaWiki to present the SOA information.

The main feature of the EAI Wiki is that most information is (hyper) linked to other information and can be reached by clicking on the hyperlink. In an overview of related services (see Figure 4 for an example), each entity is clickable and leads to the page with more information on that entity. The functional specifications and other relevant documents are stored on the Wiki server and can be accessed via an hyperlink.

In the organisation, a number of stakeholders are identified, each with different needs and focus:

1. Quality Assurance needs to know the relations between the different services to determine the effect of a request.
2. Business Analysis needs insight into the existing set of services and their status in order to determine the effort to develop a new chain.
3. Development likes to know the relations between different services to be able to determine the effect of a change in one service in relation to other services.
4. Management wants to base their decisions for e.g. budget on a correct and complete set of information.

The next sections give an overview of the pages which are currently implemented in the EAI Wiki.

3.2.1 Chain register

The Chain register is a page which shows a clickable list of all Chains present in the EAI Wiki. It is automatically generated from all Chain pages currently available in the EAI Wiki. The Chain page itself contains information on the chain like a description, version history, classifications and a picture with the services involved (see Figure 4 for an example).

3.2.2 Component register

The Component register is an automatically generated page with a list of all components present in the EAI Wiki. The structure of a Component page is similar to the Chain page with a description, version history, classifications and a similar picture but then with the focus on all services which are part of the particular component shown.

3.2.3 System register

The System register is also an automatically generated list but then of all System pages. It shows 2 lists, one for the Front End Systems and one for the Back End Systems. Again the structure of a System page is similar as the Component and Chain pages. This makes the use of the Wiki easier.

3.2.4 Public Service register

One of the characteristics of a service is its visibility: Protected (only available for use by EAI services) or Public. The Public Service register shows a generated list of all services which are available for Front End Systems. With this register, the business customers can have insight into the functionality already available for possible re-use. The structure of a Service page is again similar to the other pages.

4 Future developments

As the previous chapter described the solution currently implemented, this chapter will elaborate on some ideas for future developments.

4.1 *Integration with performance monitoring*

Within the organisation, another team is working on a monitoring system to show the usage of the EAI entities. They monitor the EAI bus in real-time to collect data over the services invoked. The data is stored in a local database and processed to enable a presentation of sensible information.

However, they can only show information on single entities, there is no information to correlate the entities. This correlation is present however in the information stored in the EAI Wiki. So a next step would be to combine these two sets of information to enrich both environments. This way, it would not only be possible to show the load of a single service but also the load of a complete chain in relation to other chains.

4.2 *Migration to stable environment*

As stated in the previous section, the monitoring project stores its information in a database. This is a system in a stable environment with a designed data model and presentation technology. The EAI Wiki however is using a MediaWiki with underlying database without a clear data model. This route was chosen because the technology and knowledge was available and easy to expand at that time.

Currently however, it is hard to extract the information needed from the EAI Wiki database, e.g. to support the monitoring project. So a next step is to define a data model for the information stored in the EAI Wiki and create a new application to present this information. In short, the EAI Wiki environment is nice to build a prototype but for a maintainable and usable system, a different environment is desirable.

5 Lessons learned

The main benefits of the MediaWiki used as visualisation tool are:

1. Visibility. The EAI Wiki is an attractive presentation mechanism. Colleagues who use it, are eager to share it with others and thus spread the news.
2. Accessibility. The EAI Wiki is accessible for everyone with access to the intranet; it is web based and easy and intuitive to use. Relations between entities can be followed by hyperlinks to traverse the contents.
3. Possibility to add contents by other users. Each EAI Wiki page contains a link to a "Notes" page. As other users traverse the EAI Wiki pages, they can add remarks to that page on its related "Notes" page. An automated feedback mechanism notifies the EAI Wiki content manager when information is added.
4. Complete overviews. The Chain page shows the overview of ALL services related to the chain; and similar for the Component and System pages. When a certain service is changed, it is clear which services are affected.
5. Simple page updates. The main content of the EAI Wiki is generated from the central repository which ensures a consistent presentation.
6. Dynamic page creation. Using MediaWiki page templates, most EAI Wiki pages are generated dynamically. Each service only defines the entities it is using which implies that a relationship is only defined once. All relations are processed when a page is generated. This means that the impact of a newly defined relation is visible immediately.

However, some of the drawbacks of the MediaWiki are:

7. Complex data model. The data stored in the EAI Wiki is processed from structured import files into the MediaWiki internal database structure which is relatively complex.
8. Performance. As each page is generated dynamically based on the information present at that moment, the presentation of the EAI Wiki is relatively slow.
9. Accessibility. As the EAI Wiki is accessible for everyone with access to the intranet; it is hard to prevent unwanted use and updates of crucial data. Also when a new set of data is imported from the central repository, manual updates are lost.

All in all, at the moment we can conclude that the EAI Wiki is a success in the organisation. This can be deducted from the questions which arise from colleagues in other departments who have seen the EAI Wiki. The main question is "When will all information be available in the EAI Wiki?"