



Creating Tools for Capture, Manipulation and Visualisation of Data for SOA Project Scope, Cost and Effort Estimation

DEPARTMENT OF COMPUTER SCIENCE
The AUSTRALIAN NATIONAL UNIVERSITY

Student: Prashant Grover (u4325940)

Supervisor: Liam O'Brien, NICTA

Course Code: COMP8790

Organizations: The ANU and NICTA

BACKGROUND

To understand the project, we need to familiarize ourselves with, and understand, certain Key terms that will be used frequently during the lifecycle of this project. These are:

- Service Oriented Architecture
- Services
- Legacy Code/Legacy Components
- Applications

Service Oriented Architecture: SOA is a model plan used to provide a standard communication mechanism for interaction between Services, Applications and Legacy Components. It is very efficient in optimizing this process in terms of scope, cost, effort, scheduling, agility, adaptability and reusability. The architecture and framework enables composition, discovery and invocation of services. This approach helps bring out the functionality of Legacy Components, contained within an organization, to users outside the organization.

Services: These represent Business Tasks that possess reusable functionality. Customer Lookup Services, Credit Check Services, Item Lookup Services and Inventory Check Services would be appropriate examples to describe this definition. Services are very useful for 2 main reasons:

- a. They can be globally distributed across several organizations and
- b. They can be reconfigured into new Business Processes

Business Processes can be classified into 3 types- Management Processes, Operational Processes and Supporting Processes.

To create a Service we are required to gather data regarding four aspects:

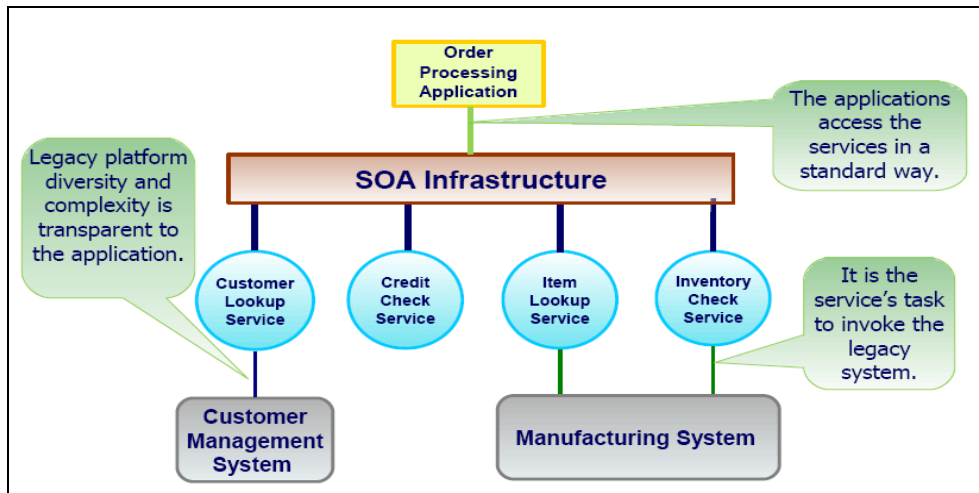
1. A detailed definition of the service supported by a low level process model.
2. List of performance indicators addressing parameters such as cost, effort, rate of performing a function
3. Link to the information model of the organization depicting information owned by the particular service and the information that is referenced by it. This would include information it creates, deletes, updates or reads.
4. Other Services that depend upon the functions defined by the Service.

Services should be loosely coupled such that any changes made within the Service should have minimal or no effect upon the Application that references it.

Legacy Code/Legacy Components: Legacy code is defined as Pre-existing code that has been serving an organization within the organization for a number of years. Legacy Code is code that could be replaced or refactored but is typically expensive and cumbersome for the organization to undertake. It could potentially also be incompatible with certain peripheral systems such as the current operating system in use. Example- Customer Management Systems, Manufacturing Systems.

Applications: These are software, present outside the organization, that reference Services. It is these Applications that indirectly reuse the functionality of Legacy Code Components via Services. Example- Order Processing Application, Invoicing Application.

Schematic of the SOA Approach



To apply the SOA approach, an organization needs to perform certain tasks. These tasks are in the form of Project Types.

They include, but are not limited to: Service Migration, Service Integration, Service Development, Application Development, SOA Infrastructure, SOA Governance and SOA Architecture Analysis.

The latter three pertain to Management of the Architecture. Typically an organization would perform a certain combination of these Project Types to fulfil its SOA initiative.

Because of the vast amount of the data required to be collated and analyzed, each of these project types further requires certain artefacts, commonly known as lists and tables, to be created. The data collected within these Artefacts would include, but again not be limited to, data related to identification, analysis, development of services or existing legacy systems, determination of cost, effort, risk, level of difficulty, testing and so forth. At the moment, there is no software tool that exists to perform the data collection, manipulation and analysis efficiently.

PREVIOUS WORK UNDERTAKEN

SMART (Service Migration And Reuse Technique), as the name suggests, was a method undertaken to perform Service Migration. It was developed by the Carnegie Mellon Software Engineering Institute for the Department of Defence to analyse the conversion of their Legacy Code Components into Services.

The SMAT-AUS Team at NICTA, Canberra, Australia is also trying to build a systematic and logical framework that integrates methods, tools, data, cost factors and functions to support scoping, cost, effort estimation and has tried to identify activities needed to be undertaken. It has been performed for Service Integration but effort is being made to extend this technique to other Project Types as well.

CURRENT PROBLEMS

Presently there are 3 main problems that need to be addressed with data collection and analysis while undertaking any one of the project types:

- a. It is cumbersome for users to input data in spreadsheet format (excel sheets)
- b. Understanding and Interpreting data is quite problematic even for expert users. This is quite obvious considering the vast amount of data required to be collected and the large number of spreadsheets corresponding to artefacts at any given instance.
- c. There is no graphical representation to correlate and compare data within the artefacts to derive accurate analyses and conclusions.

TOOL REQUIREMENTS

Data Collection

The tool shall allow for-

- creation of a project/file to organise the data for a customer
- selection of a project type from a predefined set
- creation of additional project types
- selection of a table (artefact) from a predefined set
- creation of additional tables (artefacts)
- addition of columns to tables
- removal of columns from tables
- storage of the data for later retrieval

Analysis

The tool shall allow for-

- browsing of data within the tables
- sorting of tables by column (ascending and descending order for both alphabetic and numeric data)
- summation/totalling of various columns such as those for cost or effort (especially in the tables such as Integration Alternative and Service Integration Strategy)

Data Visualisation

- The tool shall generate a 2D graphical representation of a particular artefact (e.g. system/service table)
 - With icons representing the different types of entities in the graph
 - Edges representing the connections between the entities
- The tool shall display details of a particular entity (service, system) when the entity is selected (clicked on)
- The tool shall display details of a particular edge (relation such as service integrating to system) when the edge is selected (clicked on)
- The tool shall allow for specific columns to be selected from a table for printing
- The tool shall allow for the data in tables to be printed neatly with all information in columns to appear on printed pages (portrait and landscape)

Optional Requirements

The tool shall allow for-

- reading and writing of tables and data in XML format
- reading and writing of tables and data in CVS format
- generation of summary table information (e.g. legacy system x service with specific details of cost, risk, effort, difficulty, etc. in separate tables)

TIME SCHEDULE

Week	Dates	Activities
1-2	21/7-3/8	1. Meeting Potential Supervisors 2. Finalizing Supervisor at NICTA
3	5/8-10/8	1. Initial Meeting 2. Understand Background 3. Define Requirements
4-10	11/8-28/9	1. Initial Project Presentation 2. Start Implementation 3. Continue Implementation with inputs from Supervisor
Semester Break	29/9-12/10	1. Continue Implementation
11	13/10-19/10	1. Evaluate Results
12	20/10-26/10	1. Final Report Write-up 2. Final Presentation
13	27/10-2/11	1. Review Report 2. Submit Final Report

References

1. SMART: The Service-Oriented Migration and Reuse Technique - Grace Lewis, Ed Morris, Liam O'Brien, Dennis Smith, Lutz Wrage
2. Service Oriented Architecture 2008 Determining Life Cycle Costs and Assessing Quality of Service in SOA Projects – Liam O'Brien
3. Tables courtesy SMAT-AUS NICTA, Canberra, Australia
4. [http://en.wikipedia.org/wiki/Service_\(systems_architecture\)](http://en.wikipedia.org/wiki/Service_(systems_architecture))