Subsystem Analysis
Technique Paper
Version 2.0
September 2008
SOMA Core Team

Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved

Document History

Document Location

Revision History

Date of this revision:	Date of next revision *(date)*

Revision Number	Revision Date	Summary of Changes	Comments
1	07/01/08	Developed the first draft	Sella Ganapathy
2	09/18/08	Incorporated feedback from SOMA core team	Sella Ganapathy

Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved

Table of Content

Figures 4
1 Introduction 5
2 Purpose 5
3 Description 5
3.1 Transition from Business to IT 5
3.2 Steps in Subsystem Analysis 7
3.3 Leverage Existing Assets 8
3.4 Identify Subsystem Dependencies 8
3.5 Identify Service Components 9
3.6 Assign Services to Service Components 10
3.7 Decompose Service Components into Functional and Technical Components 10
3.7.1 Functional components 11
3.7.2 Technical components 11
4 Usage 12
5 Key Consideration 12
6 Estimation Consideration 12
7 Impact of not using this technique 12
8 Related Techniques 12
9 Roles, Inputs, Outputs and Work product 13
10 Resources 13

Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved

Figures

Figure 3-1 Subsystem Analysis context in SOMA 6
Figure 3-2Transition from Business to IT 7
Figure 3-3 Steps in Subsystem Analysis 8
Figure 3-4 Subsystem Dependencies 8
Figure 3-5 Functional and Technical Components 10

Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved

1 Introduction

This technique paper provides guidance to use one of the capability patterns in SOMA namely Subsystem Analysis. In the context of SOMA a subsystem can be defines as a means of implementing a set of related business functions using information technology with well defined boundaries. This technique paper describes how to identify subsystems and subsystem dependencies and how to specify the components that comprise the subsystem. This paper is targeted for practitioners including project managers who are engaged in executing projects where SOMA Method is used.

2 Purpose

The purpose of this technique paper is to identify subsystems based on functional areas found during the Functional Area Analysis, identify subsystem dependencies and to define functional and technical components within the subsystems.

3 Description

3.1 Transition from Business to IT

Subsystem Analysis is one of the capability patterns that SOMA recommends during the specification phase. The following figures gives the context in the overall of specification phase.

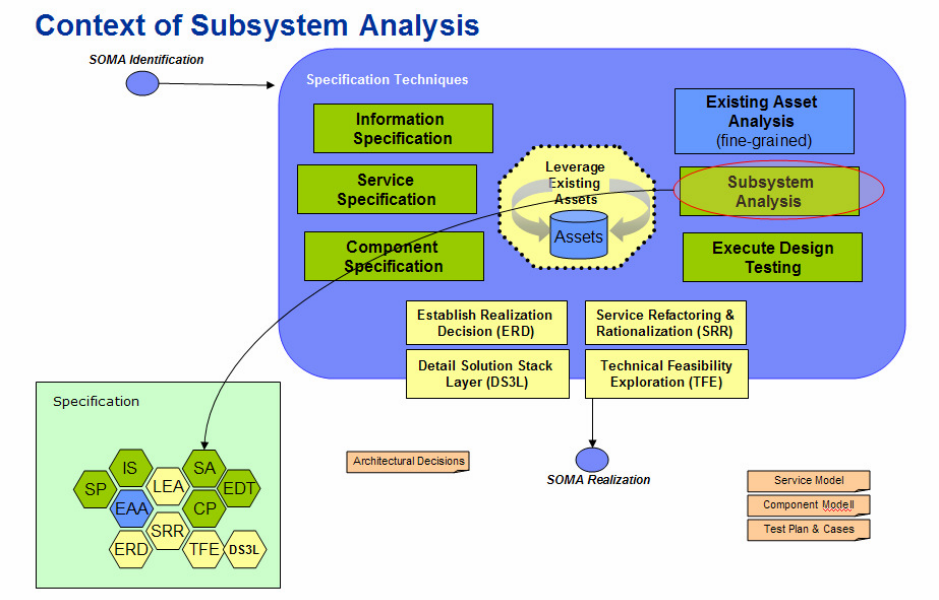


Figure 3-1 Subsystem Analysis context in SOMA

Functional Areas that are defined from the Domain Modeling and Decomposition technique provide a logical grouping of different functions from business perspective. Examples of functional areas are Customer Service, Claim Processing, Accounts Payable, Retail banking etc. 'Functional Areas' and 'Functions' are commonly used terms in the business world. In the IT world it is used less and there is an equivalent term that people are more familiar with. It is called 'Subsystem'. A subsystem can be defines as a mini-system with well defined boundaries and implements a set of related business functions using information technology. Many subsystems make up a large system. This can be a module in a packaged application, for example Accounts Payable Module, Sales Order module, Quotation module etc.

In SOMA, when we want to implement various functions of a Functional Area, we do that by defining and realizing different subsystems using information technology. In order to identify and define these subsystems, we can make use of the Functional Areas available to us from the business world. When we enter subsystem analysis, we make a conscious decision to map a functional area to, usually, one subsystem (but can be more than one, if necessary).

This mapping is the bridge from the business domain to the IT domain. Here, we identify the coarser-grained service components that will realize a subsystem. The ramification of this mapping is that the service component chosen will have the responsibility of implementing the functionality of the services identified and delivering the Quality of Service (QoS) required. The services identified become the preliminary interfaces on the façade of the associated service component.

Figure 3-2Transition from Business to IT shows the mapping from the Business Domain, through functional areas to sub-systems and components.

Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved

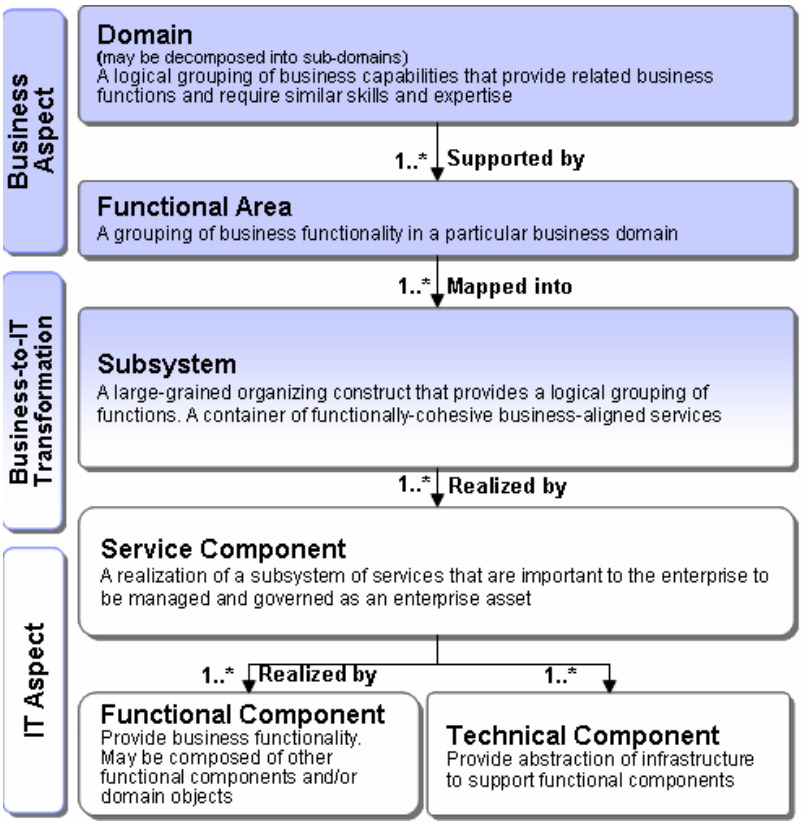
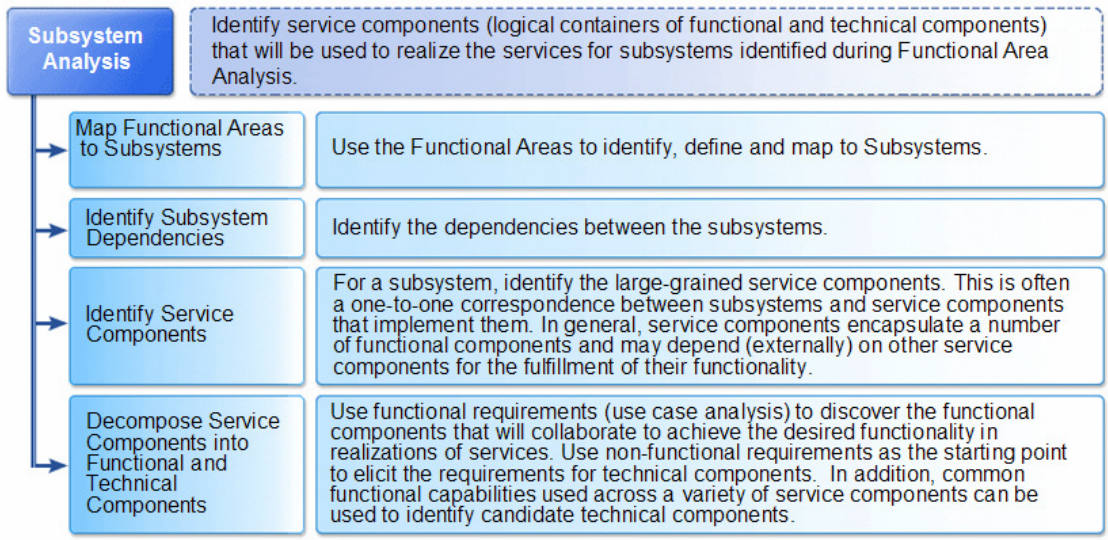


Figure 3-2Transition from Business to IT

3.2 Steps in Subsystem Analysis

Figure 3-3 Steps in Subsystem Analysis shows the steps required in the Subsystem Analysis activity.



Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved

Figure 3-3 Steps in Subsystem Analysis

3.3 Leverage Existing Assets

Leveraging existing assets is a capability pattern that SOMA uses very frequently. It is context based and in the context of subsystem analysis it is about discovering and using existing assets in order to identify relevant subsystems.

Survey the existing client environment and discover, for the given scope, what application systems are used and what components are used to provide support to business. For example an organization may use a CRM package can have a module to provide functions related to customer service. This can be considered as a potential subsystem. Similarly an automobile dealer may use two different packages one for making new car sales and the other for selling services. Each can be one subsystem.

3.4 Identify Subsystem Dependencies

This activity of Subsystem Analysis is to formulate the dependencies and associations between subsystems. A subsystem that relies on a service from another subsystem is said to be dependent on that subsystem. This includes dependencies on data that may be accessed via services that provide the required data.

In Figure 3-4 Subsystem Dependencies, the Financial Transaction Management subsystem depends on three other subsystems namely, Account Administration, Policy Administration and Account Transaction Management in order to provide functionalities to service consumers.

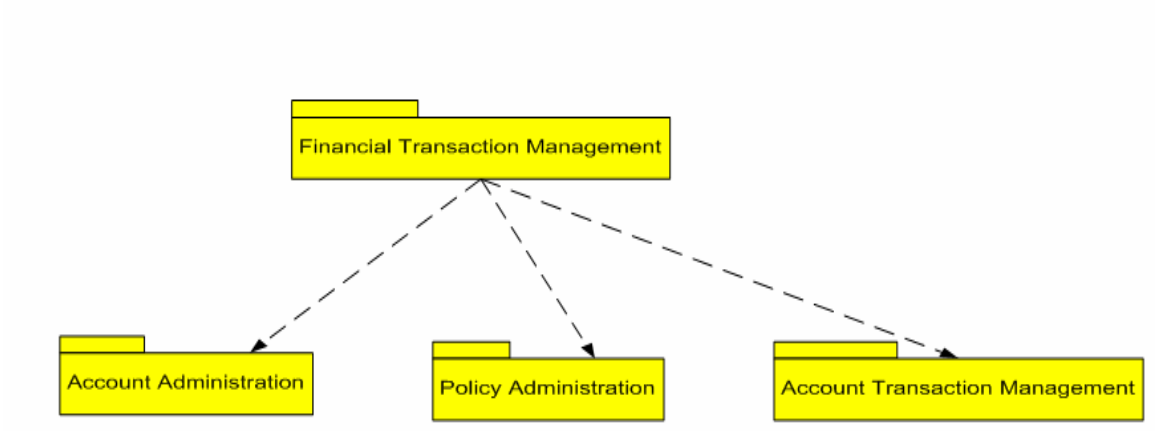


Figure 3-4 Subsystem Dependencies

When the Financial Transaction Management subsystem is accesses to obtain services, this subsystem will in turn access one or more of those three subsystems that it depends on to provide a complete response back to the caller.

Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved

3.5 Identify Service Components

Service Components are the entities that expose services to the enterprise at large. Service components are grouped into subsystems, usually because they have related business functionality.

Service components are comprised functional and technical components. A functional component provides some functionality required by the business, for example, rate a customer's credit worthiness. Technical components provide the "plumbing" to applications, for example, event handlers, logging, authentication, etc.

The rationale for service components is that it may be an enterprise-scale asset. There is funding, governance and maintenance associated with the promotion of a component to service component status. Infrastructure management must be in place to ensure availability, load balancing, security, performance, versioning and overall health of the service component.

The reader may ask "What are the criteria by which something becomes a service component versus a functional component?" In general, service components are larger-grained units that encapsulate a number of functional components and may depend (externally) on other service components for the fulfillment of their functionality.

Service components, as a whole, provide the functionality corresponding to that required by a subsystem and usually are mapped one-to-one with subsystems. Functional components can be used across several subsystems or service components and are functional units corresponding to the notion of business objects in object-oriented analysis and design (OOAD).

It is convenient to separate and group cohesive units of functionality into functional components that can be designed and implemented as individual units. We do this separation in the service realization section and often concurrently with the specification section.

In Figure 3-5 Functional and Technical Components, the Account Transaction Management subsystem is designed to be realized (implemented) by one service component called Account Transaction Management service component (same name as the subsystem as this name conveys its functionality).

Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved

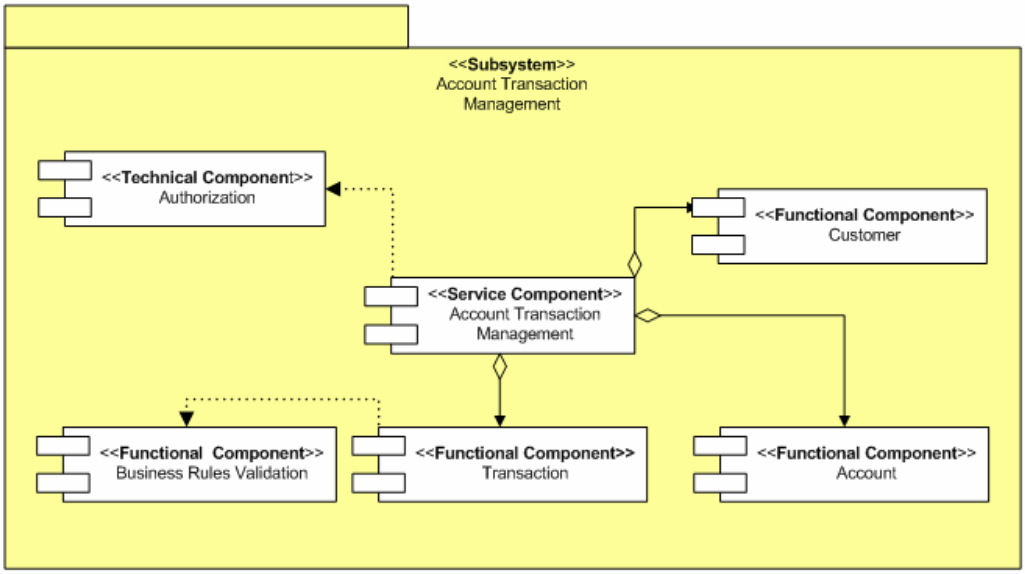


Figure 3-5 Functional and Technical Components

3.6 Assign Services to Service Components

After identifying and defining the responsibilities of each service component we know what functional area and functions that each service component realize (implement). Now the logical next step is to relate service components to services. We have used different SOMA identification techniques to identify services and used Service Litmus Test to arrive at services that need to be exposed. Now these exposed services are going to be implemented by service components. So we need to create a relationship between services and service components or in other words we need to assign services to service components for service realization.

Since the services are already categorized in Service Hierarchy, and this categorization is based on functional areas, there is a natural affinity that can easily be observed between service components and services as both are derived from functional areas. Use this hierarchy structure and map the services to respective service components.

3.7 Decompose Service Components into Functional and Technical Components

In general, service components are larger-grained units that encapsulate a number of functional components and may depend (externally) on other service components for the fulfillment of their functionality. Service components, as a whole, provide the functionality corresponding to that required by a subsystem. Service components need

Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved

to be decomposed into functional and technical components in order to be implemented.

In decomposing a service component into its constituent functional and technical components, we have delegated the functionality provided by the service component to fulfill the subsystem's functional responsibilities. Functional components supply the business functionality required, while technical components provide generic functionality such as authentication, error handling, auditing, logging, etc.

It may be noted that both functional and technical components are internal to a service component and no interfaces are defined at either at functional or technical component level. All interfaces are defined and created at service component level.

3.7.1 Functional components

The composition of these functional components into a larger-grained service component is not merely structural; it also involves the definition of flow, that is, the collaboration of the functional components to provide functionality to support the business processes. As seen earlier, the functionality of these business-related components is enabled through the services (implemented by the component's finer level object or legacy system structure) defined.

It is important to note that this step includes traditional OOAD activities. We have a focused and well-partitioned scope to direct the object design. In traditional object-oriented design, we tend to create larger more dependent object graphs, whereas if subsystem analysis follows the identification of functional areas within the business, we have a very clearly defined scope to focus on and direct our design energies towards. These results in a set of more loosely coupled object models (class diagrams and sequence diagrams triggered by system use cases).

In the Account Transaction Management service component that we identified earlier the following are examples of functional components:

- Customer
- Transaction
- Account
- Rule Engine

3.7.2 Technical components

The composition of technical components into larger-grained service components occurs in the same fashion as functional components.
Technical components such as authentication, logging and reporting may be used across business processes. These common components are needed to form the infrastructure to support the functional components.
In the Account Transaction Management service component, the following is an example of a technical component:

- Authorization

Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved

4 Usage

This technique is a key technique in the SOMA specification phase. Unless subsystems, service components and their constituent functional and technical components are defined we will not be able to decouple services from their providing operational systems. Service components along with functional and technical components populate the Service Component layer in SOA Solution Stack layer diagram. (Show example)

5 Key Consideration

When identifying subsystems, take into considerations the following aspects:

a. Geographic locations (if a function area is implemented in two different locations, then we may consider one subsystem for each location)
b. Organizational responsibilities (if a functional area is split and managed by two different managers, then we may consider to create two subsystems if the split is logical and likely to continue)
c. types of systems (if a functional area is split and implemented by one or more different IT systems, then each IT system can become a subsystem)

For example, let us consider a functional area that has been defined during Identification as Accounting Functional Area. When we review the client environment, we find that the functions within this functional area are done by two different systems. One IBM mainframe application that does the General Ledger and another Unix based application that does the capital budget accounting. Since these two systems are maintained by two different groups, it makes sense to map this functional area to two subsystems one for General Ledger and one for Capital Budget accounting.

6 Estimation Consideration

The time is required to do this technique will depend upon number of functional areas to map and number of operational systems that we need to study in order to abstract them into subsystems, service components and in turn functional and technical components.

7 Impact of not using this technique

If Subsystem Analysis is not used, there may be no traceability from the business requirements and organization to the IT domain. This could lead to difficulties in maintenance and the ability to reduce redundant applications and processes.

8 Related Techniques

Subsystem analysis is closely related to Functional Area Analysis and Component Specification capability patterns.

Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved

9 Roles, Inputs, Outputs and Work product

Roles: SOA Architect
Inputs: Service Model, Component Model, Functional Area Description, Application Architecture
Outputs: Component Model, Service Model

10 Resources

Property of IBM (C) Copyright IBM Corp. 2008 - 2011 All Rights Reserved