SOA Composition in Telecom Information Systems

Hongbin Yang

School of Management Science and Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China,

Abstract. In this article, we provide an approach to create a Conceptual level System Component Model specific to Service Oriented Architecture (SOA) in telecom. As a byproduct, SOA specific Architectural Decisions, and an SOA specific Architecture Overview Diagram will also be generated. This paper focuses on the preparation of the conceptual SOA Component Model to facilitate that selection.

Keywords: *Service Oriented Architecture (SOA), Enterprise application integration (EAI), Conceptual schema, Business integration*

1. INTRODUCTION

The telecommunications industry has seen rapid and continuous change in the last few years， Telecom carriers have over a number of years created a complex web of interconnected applications each with its own means of communicating with other systems. As new systems are introduced, they too must be interfaced with each of these legacy applications, making every step more difficult to perform and more costly. This is a particular problem for today’s wireless telecommunication industry.

A well-implemented SOA solution can achieve this. SOA seamlessly joins business-to business applications to allow systems to “talk” to one another without frontiers and without separate P2P interfaces. For telecommunications firms that can mean creating end-to-end revenue stream stability that not only addresses revenue/cost leakage but also helps to recover money more quickly.

2. UNIQUE ASPECTS OF SOA TO TELECOM BUSINESS

Subsystems do not provide core business functions, but rather enable the interaction between core business applications, components, business partner systems, and other business unit systems; to provide one or more integrated business systems. Accordingly, SOA are focused specifically on the details of interactions between these end points. An SOA solution defines a framework to the following extent:

- **Facilitate Interaction between End Points** - The most common goal of SOA is to enable interaction between heterogeneous applications.

- **Enable Distributed Operation of Business Components** - The architecture should allow business components to be distributed throughout the network, as needed. This can be useful for both situations with heterogeneous components, and components that natively work well together (a federated set).
- **Insulate End Points -** Ideally, end points should not require knowledge of consumers, messages/business objects, or how a request is satisfied. Since many business applications were designed in isolation from a wider business system context, they fit the criteria very well. A well designed SOA solution allows presentation of business functionality as anonymous services, where the location and provider of the service does not need to be known by the requester of the service.
- **Provide an Infrastructure that is Extensible and Scalable -** The SOA solution should be flexible enough to support ready addition of new end points, as well as expandable to accommodate increases in business workload.
- **Enable the Reuse of End Points** - Ideally, each application and end point service should have a single interface point, or at least a limited number of interface points, for all interactions. As new interactions are added, they should use the established interface point; thereby promoting re-use, synergy, and a simplified design.
- **Facilitate the exploitation of existing Infrastructure Services** - The SOA solution should formulate common internal access to infrastructure services to facilitate meeting security, directory, application management, etc.; requirements imposed by the business system.
- **Enable a Phased Approach to Implementation** - The SOA solution should facilitate development of short-term solution increments that lay the groundwork for implementation of the complete architecture.

3. RESEARCH TECHNIQUE

Our reference architecture covers two major elements: Business Rules and Enterprise Service Bus. More fundamentally, SOA solutions deliver savings of development costs and can reduce operational in comparison to P2P solutions. This paper focuses on the preparation of the conceptual SOA Component Model to facilitate that selection.

3.1 Make Initial Architectural Decisions

Note that prior to initiating product selection the component model will need to be elaborated to a specification level. Both product selection and component model specification are well documented already in the Method, and those techniques are not replicated here. The only SOA unique aspect to specification of the component model, is the use of the SOA Attributes Blueprint as the consolidated source of parameters for the model.

The SOA Conceptual Architecture is expressed through a number of work products. These are tabulated below. With the exception of those indicated, composition of SOA is dictated by the existing techniques for these work products.

Table 1. Conceptual Architecture Work Products

Work Products Involved In Conceptual Architecture	**Work Products focus of this technique**
ARC 100 Architectural Decisions	X
ARC 101 Architecture Overview Diagram	X
ARC 118 Change Cases	
ARC 108 Component Model	X
ARC 102 Reference Architecture Fit Gap Analysis	
APP 011 System Context	
ARC 107 Architectural Template	
ARC 301 Current IT Environment	
ARC 111 Deployment Units	
ARC 119 Nonfunctional Requirements	
ARC 310 Standards	
ARC 117 Viability Assessment	
APP 303 Detailed Gap Analysis	
ORG 006 Future Organization Design	

SOA conceptual component model composition is achieved in five basic steps, as depicted below. These steps are:

1. Make initial SOA decisions
2. Identify required Business services
3. Partition the subsystem into smaller subsystems, where required
4. Identify SOA categories present, to align components with available product configurations
5. Apply integration architectural patterns to coalesce and confirm the required components.

Throughout these steps, architectural decisions are being made and confirmed.

3.2 Service Partitioning Example

The Company has a number of subsystems defined for their SOA solution:

- Subsystem A – a B2B integration broker, to handle direct interaction with partners
- Subsystem B - a general asynchronous message-oriented SOA hub
- Subsystem C - a subsystem with synchronous interaction to facilitate synchronizing partner master file information. In this subsystem, the queues are set up as a synchronous relay for transferred data.

- Subsystem D – an ODS/Data Warehouse subsystem, that needs batch ETT capabilities. The ODS receives near-real-time inputs of operational data from the general hub subsystem.

Table 2. Subsystem Service Category Work Products

Subsystem	End Points	Model	End Point EAI Services	Central Services
A	• VPN Server	Hub	• Thin adapters, object-oriented	• Format & Transform • Routing (B2B)
B	• Web App Server • Order Entry • CRM • ERP • Production • ODS	Hub	• Thin adapters, with some XML conversion	• Queuing • Format & Transform • Routing – content-based • Process Management
C	• Order Entry • CRM • ERP	Hub	• Thin adapters, with some XML conversion, to facilitate synchronous interaction	• Process Management
D	• ODS • Data Mart	Hub	• Thin adapters, SQL oriented	• Batch Scheduling • Format & Transform • Queuing

The SOA overview diagram for telecom information system depicts the partitioning of B2B interaction functionality from internal integration functionality, and a similar partitioning of operational data store / data mart integration functionality.

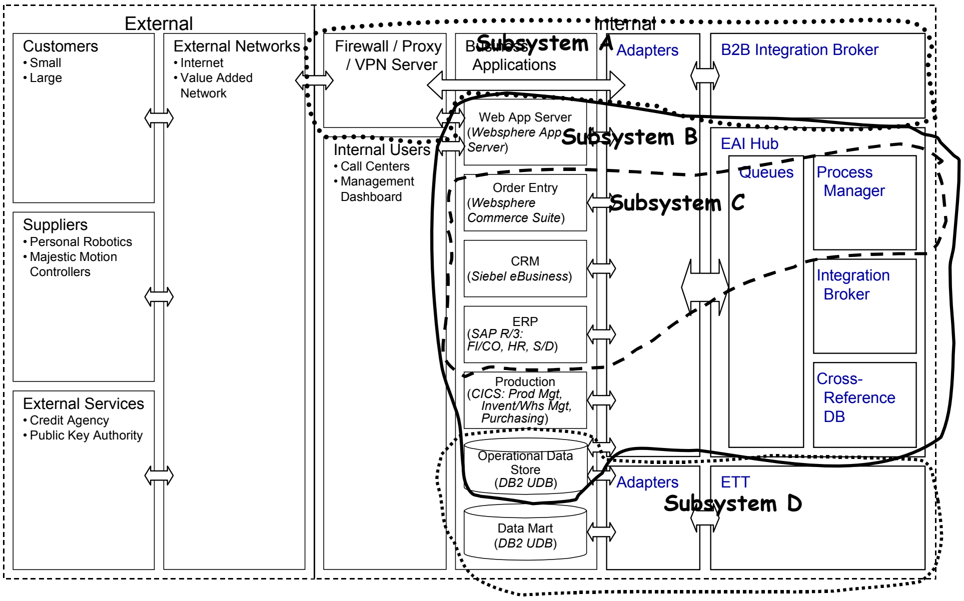

Figure 1. Architecture Overview Diagram – Telecom SOA Conceptual

3.3 Service Oriented Architecture Decisions Background Information

Utilization of the SOA Infrastructure-In telecom company, there are usually many different departments or groups, with different ideas about how to facilitate interaction across their business applications. Subsystems often have a broad scope, with the potential to benefit multiple departments and groups, and in these cases, the total benefit to the enterprise depends on incorporation of the solution by all of these different factions. There is a great risk that the solution will not be incorporated or perpetuated, if the different groups do not clearly understand the expectations related to their utilization of the SOA solution. The Architecture Decisions can be an important instrument for clarifying these expectations.

SOA Functionality within Business Applications-As the need for inter-connectivity has become recognized as a critical enabler for business applications, it has become common to find functionality living within the domain of business applications. There are four ways in which this usually is found:

- Vendor-provided business applications with modules
- Vendor-provided business applications with internal "workflow" functionality
- Custom business application, with custom services added
- Applications built on platforms that have native services

Packaged applications, and especially those in the ERP, CRM, and SCM areas, will often have a module dedicated to SOA functionality. Some of these SOA modules can be fairly sophisticated, to the point of resembling a decent message broker. Normally, these modules are optimized to focus on the message types, and interaction types that serve the needs of the package, and are not created with the richness found in products provided by vendors.

Custom applications that have accompanying custom services usually do not try to do more than what is usually found in a fat adapter: format and transform services, with perhaps rudimentary queuing, and routing services. Normally, these services target interaction with a specific few of the customer's other applications, and the intention was not to design something that could scale for enterprise-wide use. There are a few ambitious IT shops that have built from scratch a sophisticated broker device, and sometimes this is a standard that a customer will want to maintain in their collection of components.

The SOA aspects native in application servers are usually object broker and/or process/workflow services. This native functionality is optimized for use by components riding on the platform, and are usually are not intended by the vendors to be extendable for enterprise-wide service.

Business Application Insulation-Application insulation (sometimes called location independence) strives to limit the degree of knowledge one business application has of another, and specific goals and approaches related to this are important to capture. Application can be divided into two major categories:

- Physical Aspects: The degree to which one application knows the physical details of where another application resides.
- Logic and Data Aspects: The extent to which one application knows of another's internal logic, data schemas, or other capabilities/needs. (e.g. whether or not SAP is to have direct knowledge of how another application produces new sales orders)

4. RESEARCH CONCLUSIONS

SOA frameworks allow customers to more readily integrate software applications, take advantage of market opportunities and gain competitive advantage. The potential benefits of SOA are evident to telecom business but many firms struggle to achieve these when SOA solutions are not implemented guided by a coherent and well-defined Conceptual Architecture or are hindered by other obstacles such as:

- The issues involved in the complex world of governance. SOA touches on all systems and all parts of the organization making it an extremely sensitive operation and one difficult to get wide acceptance for. Again an internal sponsor alleviates these difficulties
- The implementation of the SOA solution who sticks to a standard template rarely fits with the firms existing systems and infrastructure or answers their needs

REFERENCE

1. *Web Services Description Language (WSDL) Version 2.0* , W3C Org. (2003). http://www.w3.org
2. J. Snell, D. Tidwell, and P. Kulchenko, *Programming Web Services with Soap*, (O'Reilly, 2002).
3. D. Krafzig, K. Banke, and D. Slama, *Enterprise Soa: Service-Oriented Architecture Best Practices* (Prentice Hall, 2006).
4. E. Newcomer and G. Lomow, *Understanding Soa with Web Services* (Addision-Wesley, 2006)
5. R. Sharma, B. Stearns, and T. Ng, *J2EE Connector Architecture and Enterprise Application Integration* (Prentice Hall PTR, 2001)
6. IBM Corp, *An EAI Solution using Websphere Business Integration* (2003).
7. K. Hammer, Web services and Enterprise Integration, *EAI Journal*. Volume 11, (2001).
8. J. Matjaz, N. Ramesh, R. Leander, and S.J. Basha, *Professional J2EE EAI Peer* (Information Inc., 2001).
9. G. Hohpe and B. Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions* (Addison-Wesley Professional, 2003).
10. F.A. Cummins, *Enterprise Integration: An Architecture for Enterprise Application and Systems Integration* (Wiley, 2002).
11. *Web Services Description Language (WSDL) Version 2.0* W3C Org., (2003). http://www.w3.org