Establishing a SOA in an Enterprise– a case study (part 1)

(Module 5 Learning Activities)

Prepare for the discussion

- In this example, we will study an enterprise that was faced with some business and technical problems
- When listening to this case, think about the following question:
  - How can SOA address these issues?

Case Study: RailCo Ltd.

- Established in the early 90s

Case Study: RailCo Ltd.

- Established in the early 90s
- The company has gradually grown from a staff of 12 to 40

Case Study: RailCo Ltd.

- Established in the early 90s
- The company has gradually grown from a staff of 12 to 40
- Started out as a brokerage for various railway wholesalers, but then came to specialize in air brakes.

Case Study: RailCo Ltd.

- Established in the early 90s
- The company has gradually grown from a staff of 12 to 40
- Started out as a brokerage for various railway wholesalers, but then came to specialize in air brakes.
  - The narrowed business focus resulted in increased opportunities, as RailCo was able to become a wholesaler in its own right by dealing directly with air brake parts manufacturers.

Technical infrastructure

- 5 employees and 1 manager are dedicated to full-time IT duties
  - responsible primarily for maintaining client workstations and back-end servers.

Technical infrastructure

- 5 employees and 1 manager are dedicated to full-time IT duties
  - responsible primarily for maintaining client workstations and back-end servers.
- Custom development tasks have typically been outsourced to local consulting firms

Technical infrastructure

- 5 employees and 1 manager are dedicated to full-time IT duties
  - responsible primarily for maintaining client workstations and back-end servers.
- Custom development tasks have typically been outsourced to local consulting firms
- Periodic upgrades and maintenance fixes have been the responsibility of in-house staff

RailCo's automated environment (1)

- A two-tier client-server system governing all accounting and inventory control transactions

RailCo's automated environment (1)

- A two-tier client-server system governing all accounting and inventory control transactions
  - Two administrative clerks manually feed this solution with standard transaction document data (primarily incoming and outgoing purchase orders and invoices)

RailCo's automated environment (1)

- A two-tier client-server system governing all accounting and inventory control transactions
  - Two administrative clerks manually feed this solution with standard transaction document data (primarily incoming and outgoing purchase orders and invoices)
  - Receipt and submission of these documents typically initiates corresponding inventory receiving and order shipping processes

RailCo's automated environment (2)

- A contact management system in which customer and business partner profile information is stored and maintained

RailCo's automated environment (2)

- A contact management system in which customer and business partner profile information is stored and maintained
    - This simple application consists of a database fronted by Web-based data entry and reporting user-interfaces

RailCo's automated environment (2)

- A contact management system in which customer and business partner profile information is stored and maintained
  - This simple application consists of a database fronted by Web-based data entry and reporting user-interfaces
  - Users range from managers to administrative assistants and accounting personnel

Business Problems

- Profit margins have been noticeably declining over the past year.
  - Clients have been switching to a competitor providing the same products in a more efficient manner and at a lower cost

Business Problems

- Profit margins have been noticeably declining over the past year.
  - Clients have been switching to a competitor providing the same products in a more efficient manner and at a lower cost
- Further investigation led to the discovery that this competitor has implemented an extension to their existing accounting system, allowing them to perform various transactions online via B2B solutions provided by some of the larger clients

Business Problems

- Profit margins have been noticeably declining over the past year.
    - Clients have been switching to a competitor providing the same products in a more efficient manner and at a lower cost
- Further investigation led to the discovery that this competitor has implemented an extension to their existing accounting system, allowing them to perform various transactions online via B2B solutions provided by some of the larger clients
- A further unpleasant revelation was that RailCo's primary client, Transit Line Systems (TLS), has started an online relationship with this competitor as well

Business Problems

- Profit margins have been noticeably declining over the past year.
  - Clients have been switching to a competitor providing the same products in a more efficient manner and at a lower cost
- Further investigation led to the discovery that this competitor has implemented an extension to their existing accounting system, allowing them to perform various transactions online via B2B solutions provided by some of the larger clients
- A further unpleasant revelation was that RailCo's primary client, Transit Line Systems (TLS), has started an online relationship with this competitor as well
- RailCo is a company with outdated technology automating inefficient business processes
  - Need to better respond to new business trends and automation requirements

RailCo-to-TLS initial Web Services

- To remain competitive and minimize losses, RailCo must upgrade its automation environment as soon as possible.

RailCo-to-TLS initial Web Services

- To remain competitive and minimize losses, RailCo must upgrade its automation environment as soon as possible.
- Its top priority is to participate in online transactions with TLS.

RailCo-to-TLS initial Web Services

- To remain competitive and minimize losses, RailCo must upgrade its automation environment as soon as possible.
- Its top priority is to participate in online transactions with TLS.
- Before our storyline begins, RailCo has already hurried to build a pair of Web services
  - RailCo-to-TLS Invoice Submission Process
  - TLS-to-RailCo Purchase Order Submission Process

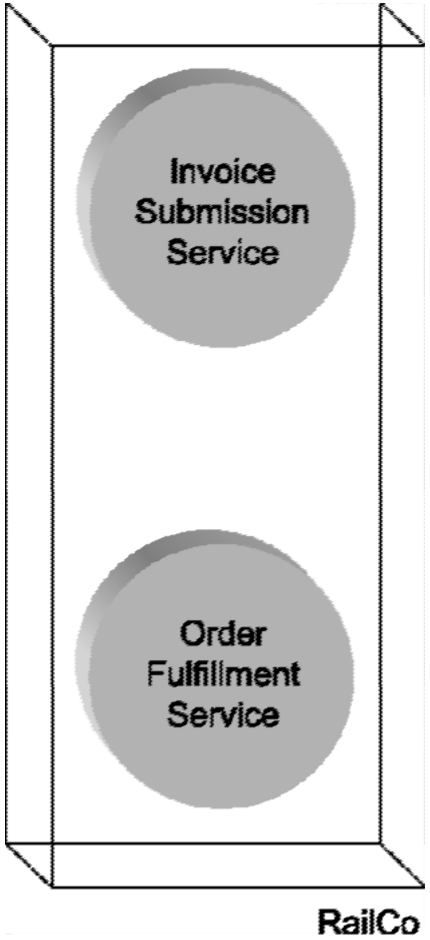


Figure 2.1: RailCo's initial set of Web services, designed only to allow RailCo to connect to TLS's B2B solution.

RailCo-to-TLS initial Web Services

- By that point RailCo runs into some limitations and decides to re-evaluate its environment in consideration of establishing an SOA. Further, RailCo realizes that it must also seek new clients to make up for the lost sales to TLS. This new requirement ends up also affecting the design of its SOA.

Prior solution (1)

- RailCo's accounting solution exists as a two- tier client-server application, where the bulk of application logic resides within an executable deployed on client workstations.

Prior solution (1)

- RailCo's accounting solution exists as a two- tier client-server application, where the bulk of application logic resides within an executable deployed on client workstations.
- Two primary tasks:
  - Enter Customer Purchase Order
  - Create Customer Order

Prior solution (2)

- The completion of each task involves a series of steps that constitute a business process.

Prior solution (2)

- The completion of each task involves a series of steps that constitute a business process.
- This process was originally modelled using standard workflow logic and then implemented as part of a packaged solution.

Prior solution (2)

- The completion of each task involves a series of steps that constitute a business process.
- This process was originally modelled using standard workflow logic and then implemented as part of a packaged solution.
- Within the application, the process may or may not be separately represented by individual sets of programming routines.

Prior solution (2)

- The completion of each task involves a series of steps that constitute a business process.
- This process was originally modelled using standard workflow logic and then implemented as part of a packaged solution.
- Within the application, the process may or may not be separately represented by individual sets of programming routines.
- Regardless, it is compiled into a single executable that provides a fixed manner in which the process is automated.

RailCo's accounting system

- RailCo's accounting system is a classic two-tier client- server application.
- Its GUI front-end consists of a single executable designed for deployment on old Windows workstations.
  - It provides user-interfaces for looking up, editing, and adding various accounting records
  - It also offers a financial reporting facility that can produce a fixed amount of statements with detailed or summarized accounting data

RailCo's accounting system

- RailCo's accounting system is a classic two-tier client- server application.
- Its GUI front-end consists of a single executable designed for deployment on old Windows workstations.
    - It provides user-interfaces for looking up, editing, and adding various accounting records
    - It also offers a financial reporting facility that can produce a fixed amount of statements with detailed or summarized accounting data
- Considering it's only ever had two to three users, there have never really been performance problems on the database end. The now outdated RDBMS (Relational Database Management System) that has been in place for the past decade has been reliable and has required little attention.

Problems with this application have surfaced:

- Operating system upgrades have introduced erratic behavior on some screens, resulting in unexplainable error messages.
    - It is uncertain if these are caused by other programs that have been installed on the workstations.

Problems with this application have surfaced:

- Operating system upgrades have introduced erratic behavior on some screens, resulting in unexplainable error messages.
  - It is uncertain if these are caused by other programs that have been installed on the workstations.
- The workstations themselves have been rarely upgraded and have not kept pace with the hardware demands of recent software upgrades.
  - After the accounting system launches, there is little more the user can do with the computer. As a result, employee productivity has been affected somewhat.

Problems with this application have surfaced:

- Operating system upgrades have introduced erratic behavior on some screens, resulting in unexplainable error messages.
    - It is uncertain if these are caused by other programs that have been installed on the workstations.
- The workstations themselves have been rarely upgraded and have not kept pace with the hardware demands of recent software upgrades.
    - After the accounting system launches, there is little more the user can do with the computer. As a result, employee productivity has been affected somewhat.
- Following a new records management policy and some billing procedure changes, a modification to the overall billing process was imposed on the accounting personnel.
    - Because the accounting system was not designed to accommodate this change, employees are required to supplement the automated billing process by manually filling out supplementary forms.

Discussion

- Fundamentally, this accounting system has been getting the job done.
- However, the actual accounting tasks performed by the users have become increasingly convoluted and inefficient. This is due to the questionable stability of the workstation environments and also because the system itself is not easily adaptable to changes in the processes it automates.
- How can SOA address these issues?

Open Question Points: 10

设置

Fundamentally, RailCo’s accounting system has been getting the job done.

However, the actual accounting tasks performed by the users have become increasingly convoluted and inefficient. This is due to the questionable stability of the workstation environments and also because the system itself is not easily adaptable to changes in the processes it automates.

How can SOA address these issues?

作答

Discussion

- Fundamentally, this accounting system has been getting the job done.
- However, the actual accounting tasks performed by the users have become increasingly convoluted and inefficient. This is due to the questionable stability of the workstation environments and also because the system itself is not easily adaptable to changes in the processes it automates.
- How SOA can address these issues?

SOA can address issues such as these, as follows:

- Service-oriented solutions eliminate dependencies on user workstation environments by delegating all processing to the server- side (as regular distributed Internet applications already have been doing).

SOA can address issues such as these, as follows:

- Service-oriented solutions eliminate dependencies on user workstation environments by delegating all processing to the server- side (as regular distributed Internet applications already have been doing).
- SOA establishes an adaptable and extensible architectural model that allows solutions to be enhanced with minimal impact.

SOA can address issues such as these, as follows:

- Service-oriented solutions eliminate dependencies on user workstation environments by delegating all processing to the server- side (as regular distributed Internet applications already have been doing).
- SOA establishes an adaptable and extensible architectural model that allows solutions to be enhanced with minimal impact.
- Services can encapsulate existing legacy logic providing a standardized API that can plug into larger integrated solutions.

SOA can address issues such as these, as follows:

- Service-oriented solutions eliminate dependencies on user workstation environments by delegating all processing to the server- side (as regular distributed Internet applications already have been doing).
- SOA establishes an adaptable and extensible architectural model that allows solutions to be enhanced with minimal impact.
- Services can encapsulate existing legacy logic providing a standardized API that can plug into larger integrated solutions.
- Further, when building custom service-oriented applications, extensibility can be built into the solution environment, supporting future enhancements, also with minimal impact.

On the way to SOA

- As we know, a process in its entirety can be viewed and modelled as a service. Additionally, one or more processes can be combined to represent an even larger service.
  - For example, the Create Customer Order and Generate Customer Invoice Processes may be combined to form a single Order Processing Process.

On the way to SOA

- As we know, a process in its entirety can be viewed and modelled as a service. Additionally, one or more processes can be combined to represent an even larger service.
  - For example, the Create Customer Order and Generate Customer Invoice Processes may be combined to form a single Order Processing Process.
- We would also expect these processes to be flexible so that they can incorporate processes or resources that exist elsewhere in the enterprise.
  - For instance, we may decide to extend the Order Processing Process to include a sub-process that automatically retrieves the customer's current accounts payable mailing address.
  - This sub-process may already exist as part of a separate Customer Contact Reporting Process.

On the way to SOA

- To implement such a model, we need a technical architecture capable of providing the following:
  - The ability for business automation logic to be partitioned into units so as to properly represent services.
  - The ability for these units of logic to be relatively independent of each other so as to support the requirement for them to participate in different compositions.
  - The ability for these units of logic to communicate with each other in such a manner that their respective independence is preserved.

On the way to SOA

- To implement such a model, we need a technical architecture capable of providing the following:
    - The ability for business automation logic to be partitioned into units so as to properly represent services.
    - The ability for these units of logic to be relatively independent of each other so as to support the requirement for them to participate in different compositions.
    - The ability for these units of logic to communicate with each other in such a manner that their respective independence is preserved.
- The fundamental characteristics of service encapsulation, loose-coupling, and messaging, as realized through service-orientation principles and the Web services technology set, collectively fulfill these requirements through the implementation of a primitive SOA.

With SOA vs without SOA

With SOA

- The logic behind each process would be partitioned into one or more services.

Without SOA

- The bulk of application logic resides within an executable deployed on client workstations

With SOA vs without SOA

With SOA

- The logic behind each process would be partitioned into one or more services.
- If multiple services are used, the execution of the entire process would involve the composition of these services.
  - In this case, each service may represent a sub- process or even a single step within the process that can be executed independently.

Without SOA

- The bulk of application logic resides within an executable deployed on client workstations
- The completion of each task involves a series of steps that constitute a business process.
  - This process was originally modelled using standard workflow logic and then implemented as part of a packaged solution.

With SOA vs without SOA

With SOA

- The logic behind each process would be partitioned into one or more services.
- If multiple services are used, the execution of the entire process would involve the composition of these services.
- In this case, each service may represent a sub-process or even a single step within the process that can be executed independently.
- For example, the Create Customer Order Process may consist of the following sub-processes:
  - Retrieve Purchase Order Data
  - Check Inventory Availability
  - Generate Back Order
  - Publish Customer Order

Without SOA

- The bulk of application logic resides within an executable deployed on client workstations
- The completion of each task involves a series of steps that constitute a business process.
- This process was originally modelled using standard workflow logic and then implemented as part of a packaged solution.

Module 5 case study Summary

- Establishing a SOA in an Enterprise

Establishing a SOA in an Enterprise– a case study (part 2)

(Module 5 Learning Activities)

Lesson Plan

- Case study
  - Background
  - Discussion in groups
  - Sharing session – each group shares their conclusions with the whole class
  - Conclusion

Case study background

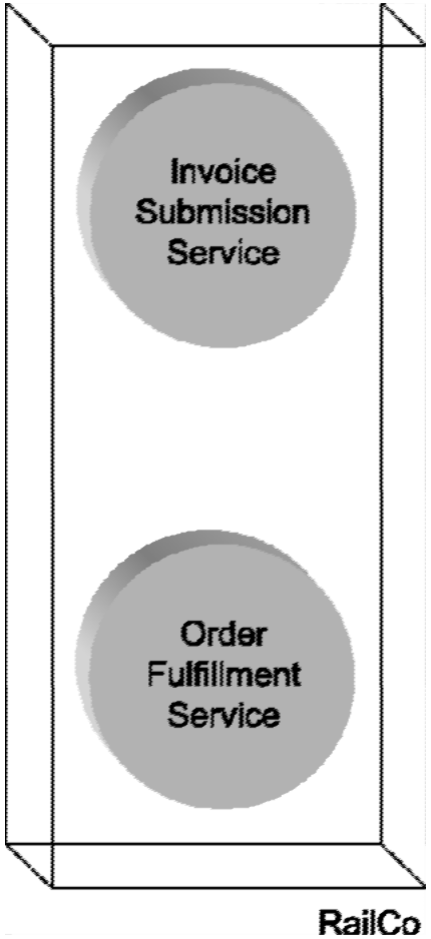
- Please check the document uploaded to the Blackboard

RailCo problems

- RailCo's original goals
  - upgrade its automation systems
  - remain competitive
  - continue its business relationship with its primary client, TLS.
- RailCo had lost TLS as a customer when a competitor managed to provide air brake parts at a lower price while also interfacing with TLS's B2B system.

Railco – initial services

- RailCo rushed to catch up, producing a pair of Web services designed only for use with the TLS system.
- This allowed RailCo to regain its position as a TLS vendor.
- These two initial Web services were
  - Invoice Submission Service
  - Order Fulfillment Service
  - (Another service was added later to interact with the TLS Notification Service.)



RailCo’s initial set of Web services, designed only to allow RailCo to connect to TLS’s B2B solution.

RailCo problems continue

- However, even though RailCo had successfully reconnected with TLS, it had lost its exclusive relationship.
- It now found itself in a position where it had to bid against an aggressive competitor for every purchase order issued; therefore, it was still losing revenue.

RailCo – new goals

- The only way RailCo could avoid significant downsizing was by finding new clients.
  - To accomplish this, RailCo needed to continue pursuing the online vendor marketplace with other transit companies providing B2B solutions.
- It then became evident that RailCo's current set of Web services was insufficient for this purpose.
  - Because they had been designed solely for use with TLS, they were not useful for interacting with other customers that dictated different business and transaction requirements.

RailCo – on the way to SOA

- RailCo was then faced with an important decision
  - either develop a custom set of services for each new client
  - or start from scratch and build a standardized set of services generic enough to facilitate multiple clients.
- It chose the latter option and decided that the best way to achieve this goal was to overhaul its existing environment in favor of an SOA.

RailCo business processes

- RailCo's two primary business processes are:
  - Order Fulfillment (accepting and processing purchase orders from a client) and
  - Invoice Submission (sending an invoice to a client).

RailCo – new design (1)

- RailCo proceeded with a service-oriented analysis that decomposed its business process logic into a series of service candidates. This revealed the need for the following potential services and service layers:
  - A business service layer consisting of two task-centric business services:
    - Invoice Processing Service
    - PO Processing Service
  - An application service layer comprised of four application services:
    - Legacy System Service
    - Polling Notification Service
    - Transform Service
    - Metadata Checking Service

Railco – new design (2)

- RailCo did not have the technology or the budget to invest in middleware capable of providing orchestration. It therefore chose not to pursue centralizing its business logic in an orchestration service layer.

Railco – new design (3)

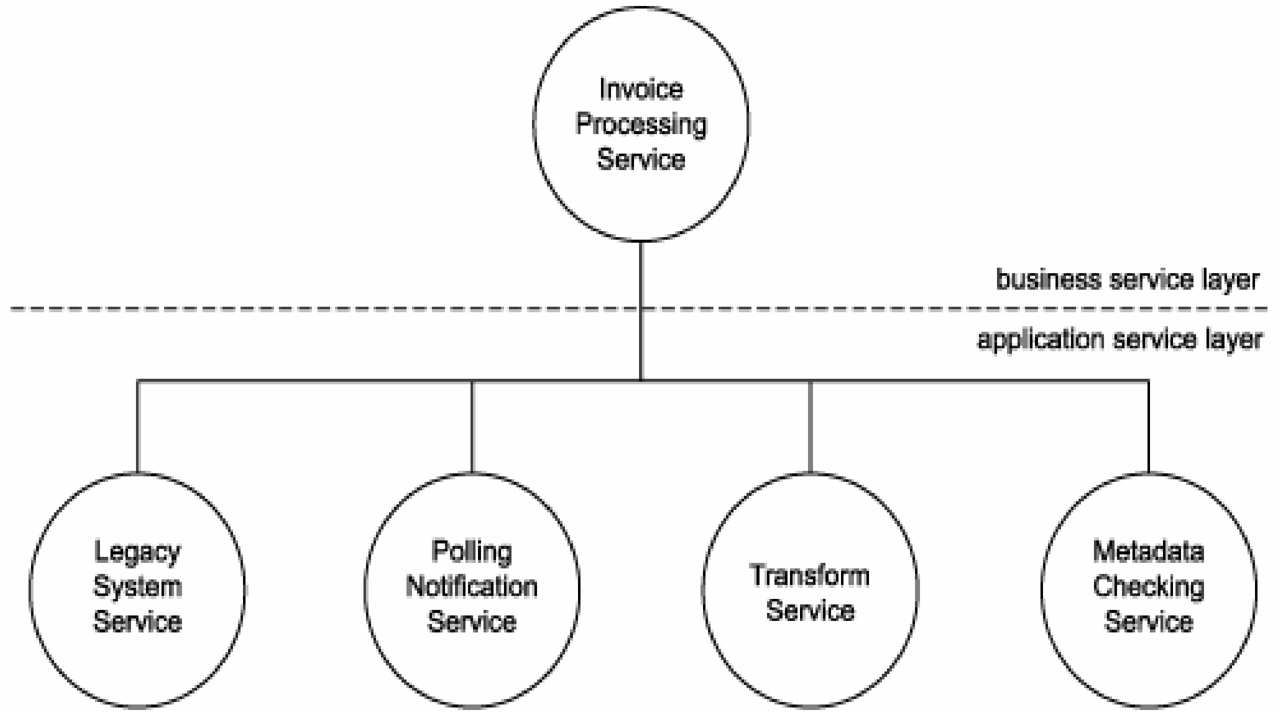
- Instead, it was decided to represent each business process with a task-centric business service that would act as a controller for a layer of application services. The following services were modeled and then designed:
  - Invoice Processing Service (task-centric)
  - PO Processing Service (task-centric)
  - Legacy System Service (application)
  - Polling Notification Service (application)
  - Transform Service (application)
  - Metadata Checking Service (application)

Focus on reusability and extensibility

- Reusability and extensibility in particular were emphasized during the design of its application services.
- RailCo wanted its initial SOA to consist of services that supported both of its current business processes, while being sufficiently extensible to accommodate future requirements without too much impact.

Invoice Submission Process as a service

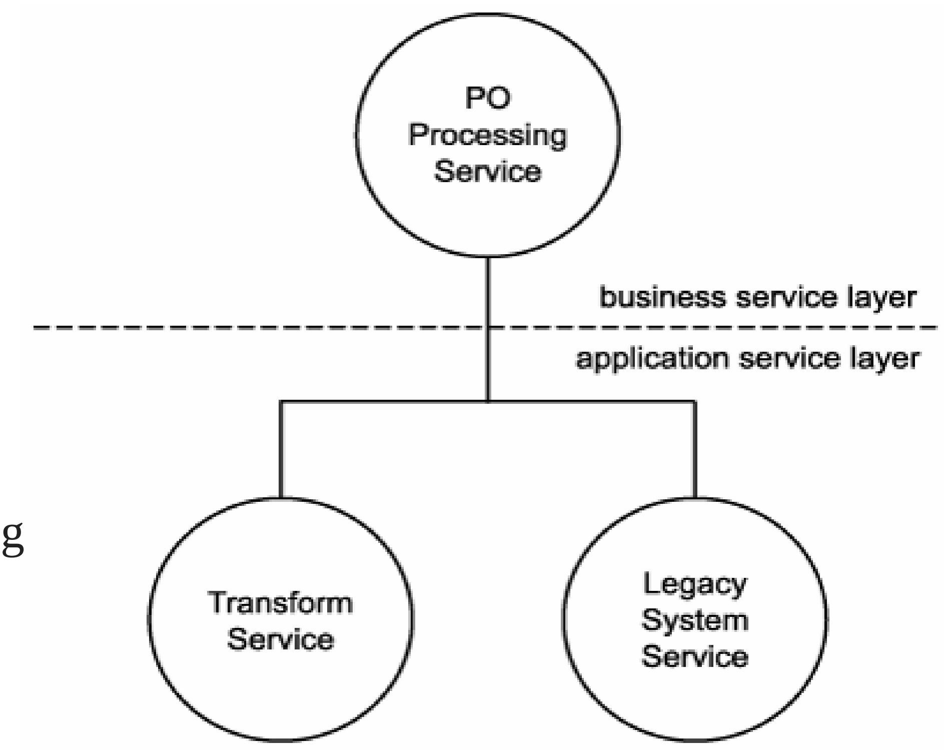
- RailCo was able to compose these services into a two-level hierarchy, where the parent Invoice Processing Service coordinates the execution of all application services


Invoice Processing Service
business service layer
application service layer
Legacy System Service
Polling Notification Service
Transform Service
Metadata Checking Service

RailCo's service composition that automates its Invoice Submission Process.

Order Fulfillment Process as a service

- The can now be automated via the PO Processing Service, which reuses two of the same application services used by the Invoice Submission Process



The Order Fulfillment Process is automated by a PO Processing Service that composes two reusable application services.

RailCo – implementation of SOA

- In the face of some bad news involving the departure of the .NET consultants responsible for delivering their original Web services, RailCo was able to put internal resources to good use. Subsequent to a training effort, the new SOA was created as a J2EE solution.

RailCo SOA – results (1)

- RailCo has fulfilled its original goals by producing an SOA that supports two service-oriented solutions.
  - RailCo can now continue its online transactions with TLS while confidently seeking new customers.
  - Additional clients introducing new requirements can be accommodated with minimal impact.

RailCo SOA – results (2)

- Its standardized application service layer will likely continue to offer reusable functionality to accommodate the fulfillment of new requirements.
- And any functional gaps will likely be addressed by extending the services without significantly disrupting existing implementations.
- Further, should RailCo decide to replace its task-centric business services with an orchestration service layer in the future, the abstraction established by the existing application service layer will protect the application services from having to undergo redevelopment.

RailCo SOA – results (3)

- Upon completing this project, RailCo discovers a side benefit to its new solution environment. By having established the Legacy System Service (which is essentially a wrapper service for its accounting system) as part of its application service layer, it has opened up a generic endpoint that can facilitate integration.
- This provides the potential for RailCo to enable interoperability between its accounting system and its contact management application.
- By allowing these two environments to share data, RailCo can more efficiently take on and service new clients with coordinated contact and financial history profiles.

Discussion

- Consider the case of RailCo introduced in this text as well as in part 1 discussed earlier
- Recall service design principles.
- Evaluate the RailCo solutions in terms of loose-coupling. Present and defend your opinion.

Discussion – 30 minutes

- Discuss with your discussion group in WeChat
  - The groups have been announced in the course WeChat before the class today – please check which group you have ben assigned to and scan the QR code to join
  - Prepare the answer to the question
    - Written format
    - Oral format
- Submit the written answer before the sharing session starts
  - One person submits the draft of the answer to the Rainclassroom (please type the number of your group on the top of your answer)
- Agree who is the speaker for the group, help the speaker prepare
  - One person (or more) prepares to speak in Tencent Meeting (each group should make sure the speaker is agreed upon by the group before we start the sharing session)
- Come back to the Tencent Meeting and Rain Classroom after 30 minutes

We are about to finish discussions

- It is time to submit your answers in the Rain Classroom
- Groups who want to share their answers with the whole class, please let me know

Open Question Points: 10

Setting

Evaluate the RailCo solutions in terms of loose-coupling. Present and defend your opinion.

Open Question is only supported on Version 2.0 or newer.

Answer

Tips to answer this question

- Recall the meaning of loose-coupling service design principle (minimizing dependencies)
- Compare the two designs
  - Initial design of RailCo Web Service – designed specifically for collaboration with one partner (TLS)
  - New design – service developed in more generic way
- Then, think about if the services developed specifically for one service requestor are more loosely-coupled than services developed in more generic way?

Module 5 case study Summary

- Establishing a SOA in an Enterprise