

Architecture Documentation Questionnaire

Thank you for agreeing to participate in this research exercise. The purpose of this research is to investigate architecture documentation in the context of system integration.

The rest of this document outlines a role-playing exercise, then provides a set of architecture documentation for you to use, and finally asks you to use that documentation to answer some questions.

We ask you to not read ahead in this document, or skip questions with the thought of coming back to them later. You can print the document, if you would prefer to work with hardcopy. Please try to answer each question based on your experience and the information we have provided up to that point in the exercise.

You can withdraw from this study at any time, for any reason. Our only request is to please send an email to jklein@sei.cmu.edu indicating that you will not be completing the exercise.

Before we start, we need to collect some information about you, to help us with our research data analysis. Please provide your answers in the table below:

Demographic Information

Question	Your Answer
How do you identify your professional position (e.g., software architect, enterprise architect, etc.)?	
What type of organization do you <u>currently</u> work for (e.g., commercial product development, consulting, government, military, etc.)?	
If you have <u>past experience</u> working in other types of organizations, please list those here.	
How many years of professional experience do you have <u>developing or integrating software systems</u> ?	
What is the approximate number of system integration projects that you have worked on?	

Architecture Documentation Questionnaire

This is a role-playing exercise. You will play the role of an architect at a fictitious company called “Social Travel”. Your business sells travel packages, and provides a number of social networking capabilities to allow your users to connect and share information with each other.

You are responsible for a collection of integrated enterprise systems:

- *TravelPhotos* – provides photo storage, tagging, and sharing with other Social Travel users.
- *TravelStuff* – an online retailer for travel-related gear
- *TravelIns* – marketing travel insurance, such as trip cancellation coverage, international medical coverage, and emergency rescue/evacuation coverage.
- Enterprise-wide Social Features – a back-end system for cross-cutting features such as Profiles, Friends, Tags, Recommendations, Sharing, etc.

Your company has just acquired a smaller company called “Adventure Builder” that specialized in selling adventure travel packages. You need to integrate some of their systems with your enterprise systems, to realize three new capabilities:

New Capability #1 – Social features

This capability adds social features to Adventure Builder catalog browsing. A user can see which trips her friends have taken, see comments about trips from other users, see trip photos from other users, share plans and itineraries with friends, etc.

The architecture approach to achieve this capability will store the social data repository (i.e. user profile, tags, sharing links, comments, etc.) outside of the Adventure Builder system. The Adventure Builder system must provide keys (or IDs) for data inside Adventure Builder that the social repository can link to. We also need to insert new widgets and elements into the customer UI to allow access to the social features.

New Capability #2 – Common payment processing interface

This capability changes the Adventure Builder payment processing interface to use the same interface as is used by the existing TravelStuff systems.

The architecture approach to achieve this capability is to completely replace Adventure Builder’s “Bank” interface.

New Capability #3 – Add cross-sell features

This capability changes the Adventure Builder user interface to cross-sell products from TravelStuff and TravelIns when user books trip. For example, based on the destination, offer relevant clothing products from TravelStuff and insurance from TravelIns against a hurricane forcing the trip to be cancelled.

The architecture approach to achieve this capability is to make a request to existing SocialTravel cross-sell business logic trip with itinerary information, and then insert new widgets and elements into the customer UI to display the cross-sell offers.

In order to scope the work to develop these new capabilities, you need information about the Adventure Builder system. Below, please list three questions that you have about the Adventure Builder system architecture. These questions do not have to be the highest priority, or listed in any particular order. We are trying to understand how you approach this problem.

	Your Questions:
Question 1:	
Question 2:	
Question 3:	

Architecture Documentation Questionnaire

As often happens in a corporate acquisition, you do not have direct access to your counterpart architect at Adventure Builder. In this scenario, as soon as the deal closed, the Adventure Builder architect cashed in her stock and left on a trip around the world.

However, before she left, she prepared a documentation package for you that addresses concerns related to integrating an existing system into an enterprise or into a system of systems (SoS). This documentation refers to the as-is existing system as a “constituent system” of the SoS.

This documentation has five models, summarized here:

Constituent System Stakeholders/Concerns Model: Maps stakeholders in the constituent system to their concerns about that system. This helps you understand the scope of the constituent system, architecture drivers, and who will be impacted by changes that you make to the constituent system to allow it to join the SoS.

Constituent System Execution Time Context Model: Describes any runtime interactions between the constituent system and external systems, including request/response, data exchange, message passing, and error/exception handling.

Constituent System Code Context Model: Identifies external modules (libraries, packages, development tools, etc.) that the constituent software depends on, along with the type of dependency.

Constituent System Interface Information Model: Information elements within the constituent system that are of interest to the SoS (e.g., a SoS that deals with geo-location might have concepts like position, elevation, and direction). Note that this data may not be accessible through existing external interfaces of the constituent system.

Shared Resource Model: Component(s) that represent a resource used by the constituent system and by other external systems. These resources include processor computing cycles, memory, disk space, network interfaces, network bandwidth, files, databases or repository, virtual infrastructure, and system physical resources such as a display, radio link, or sensor. This model also includes component(s) in the constituent system that use each shared resource.

The models in the documentation package are shown on the following pages. There is a lot of detail here – skim it for now, and some later questions will ask you to look at it more closely. Again, you can print this document if you want to.

Some of the models have traceability references back to the Adventure Build architecture description, which is available at https://wiki.sei.cmu.edu/sad/index.php/The_Adventure_Builder_SAD, but you should not need to refer to that directly.

Architecture Documentation Questionnaire

Model: Constituent System Stakeholders/Concerns

Table 1 – Constituent System Stakeholders/Concerns Model for the Adventure Builder System

Stakeholder	Concerns	Traceback to AB Architecture
Product Manager	Modifiability – add new business partners quickly	QAS1
Product Manager	Usability, Performance – purchase action latency	QAS2
Product Manager Operations	Latency under load	QAS3
Operations Finance	Reliability - Purchase requests to OPC are idempotent. Usability – Successful purchases are always acknowledged to customer.	QAS4
Legal Information Security Product Manager	Security – Payment processing transactions are secure and meet all internal policies and regulatory compliance requirements.	QAS5
Operations Information Security Product Manager	Denial of Service attack is detected and handled.	QAS6
Operations	24/7 availability. Failures detected and notification issued within 30 seconds.	QAS7

Model: Constituent System Execution Time Context

Table 2 – Constituent System Execution-Time Context Model for the Adventure Builder System

External System	Interfaces to the external system and interface properties	Traceback to AB Architecture
Bank	Interface: CreditCard Service/SOAP/Adventure Builder invokes request	Top Level SOA View Primary Presentation
Airline Provider	Interface: AirlinePO Service/SOAP/Adventure Builder invokes request Interface: Web Service Broker/SOAP/Adventure Builder receives request	Top Level SOA View Primary Presentation
Lodging Provider	Interface: LodgingPO Service/SOAP/Adventure Builder invokes request Interface: Web Service Broker/SOAP/Adventure Builder receives request	Top Level SOA View Primary Presentation
Activity Provider	Interface: ActivityPO Service/SOAP/Adventure Builder invokes request Interface: Web Service Broker /SOAP/ Adventure Builder receives request	Top Level SOA View Primary Presentation
User's Email Client	Interface: SMTP/SMTP/external configuration file	Top Level SOA View Primary Presentation

Also, the top-level workflow diagram is applicable here to show how these interfaces are used in practice. This is shown in Figure 1.

Architecture Documentation Questionnaire

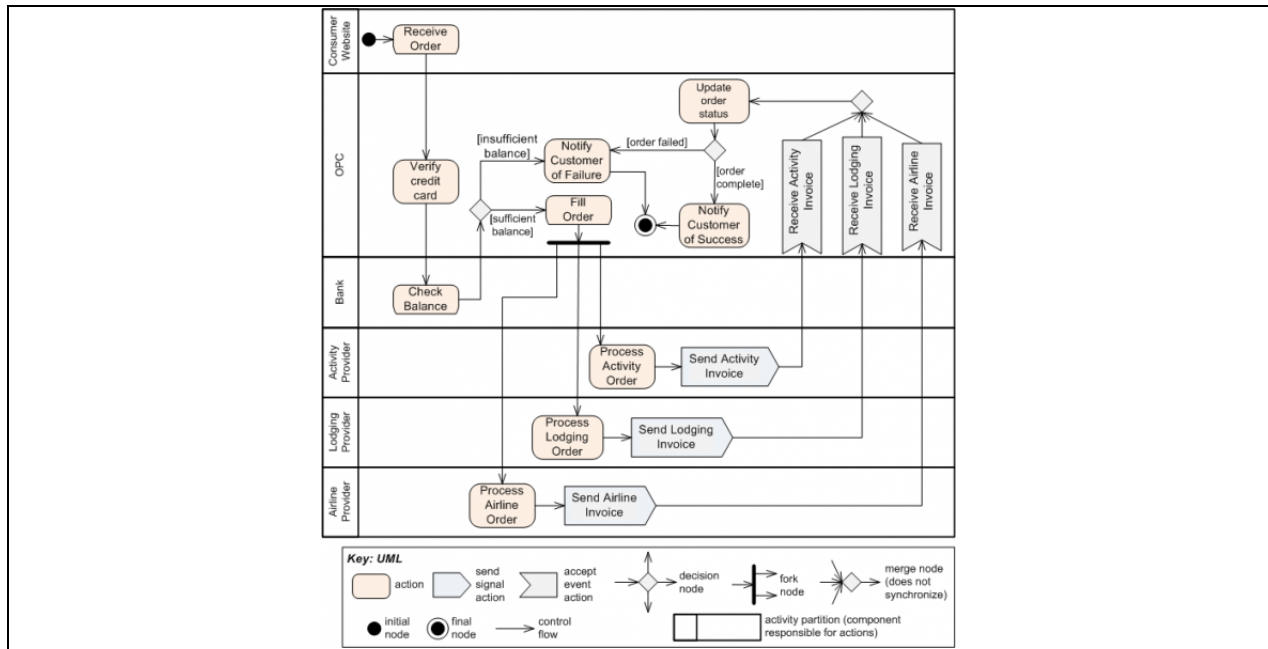


Figure 1 – Workflow behavior diagram from Adventure Builder Architecture

Model: Constituent System Code Context

Table 3 – Constituent System Code Context Model for the Adventure Builder System

External Module used by AB System	Properties	Traceback to AB Architecture
gwt (Google Web Toolkit)	Dependency Type: Build (generates Javascript for execution); Version unspecified; Open Source	Top Level Module Uses Diagram
waf (Web Application Framework)	Dependency Type: Build?; Version "Java Blueprints"; Open Source	Top Level Module Uses Diagram
wsdls	Dependency: build; Version unspecified; license unspecified.	Top Level Module Uses Diagram

Constituent System Interface Information Model for the Adventure Builder System

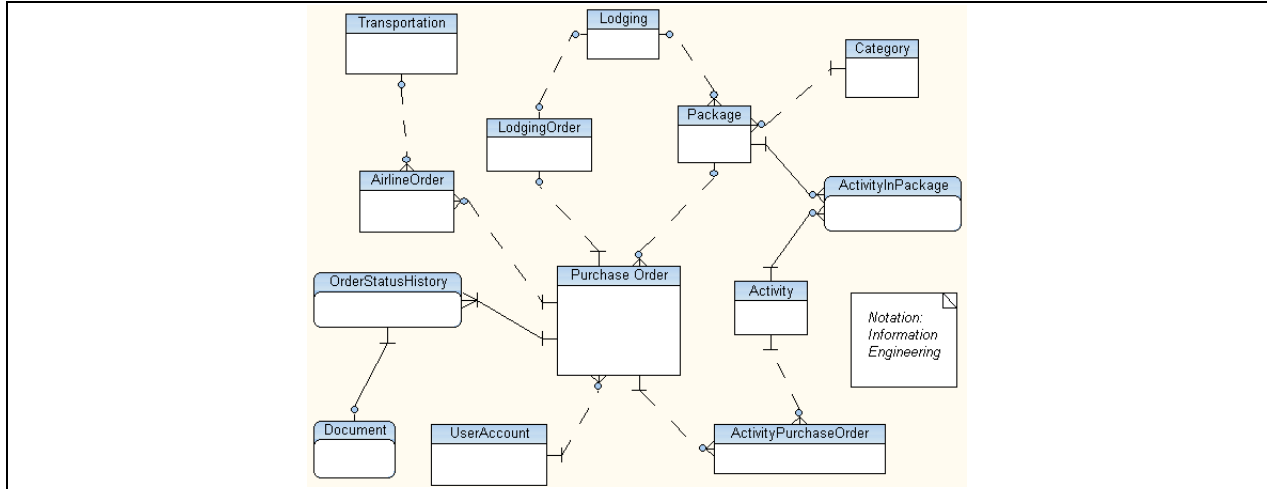


Figure 2 – Part 1 of Constituent System Interface Information Model for the Adventure Builder System (From Adventure Builder Data Model View)

Our approach for new capability #2 requires us to replace the Bank interface. Part of the information model for the Bank interface is shown in Figure 3.

Architecture Documentation Questionnaire**Table 4 – Element Catalog for Part 1 of Interface Information Model**

Information Element	Description
PurchaseOrder	Aggregate of transportation, lodging, package, and activity orders.
UserAccount	An end user of the AdventureBuilder application. We store email id, password, and contact info.
AirlineOrder	Aggregate of purchased transportation entries.
Transportation	Each transportation entry is a flight available for booking in our travel packages. For each one, we record: name, departure and arrival airports, days and times, airline name, flight number, rate, cabin class.
LodgingOrder	Aggregate of purchased lodging entries.
Lodging	A hotel, guesthouse or B&B that can be used for lodging in our travel packages. For each type of lodging, we store name, description, location info, room description, rates.
Package	A travel package available in our catalog. A package specifies lodging and a list of activities. Attributes of a package include name, description, rate per person, category and a representative image to show to the user.
Category	A category of adventure travel packages. Examples: island packages, mountain adventures. This categorization helps the user to browse through our catalog of packages. Category data consists of a name, description and a representative image to show on the user screen.
Activity	An adventure activity available. Examples: snorkeling, fishing, bird watching, rafting, surfing. Activities are available in selected packages. Information stored for each activity include: name, description, rate, and a representative image to show to the user.
ActivityInPackage	This entity represents the many-to-many relationship between activity and package. It simply lists the activities in each package (“join table”).
ActivityPurchaseOrder	Aggregate of purchased activity entries (“join table”)

<p>Type CreditCard. This type is used to store the credit card information of the user.</p> <p>String cardExpiryDate String cardNumber String cardType</p>

Figure 3 – Part 2 of Constituent System Interface Information Model for the Adventure Builder System**Model: Shared Resource Model**

In the stand-alone Adventure Builder system, there are no resources shared with any external systems.

In the new SoS, we have several resources that now will be shared. These are shown in Table 5. For each resource identified in Table 5, the source of the information in the Adventure Builder Architecture Documentation is shown.

Architecture Documentation Questionnaire**Table 5 – Shared Resource Model for the Adventure Builder System**

External Resource	Adventure Builder Resource Usage	Resource shared with	Traceback to AB Architecture
Bank Interface (accessed through Firewall)	Validate credit card for every customer purchase (Call/Return)	Other SocialTravel.com applications (Call/Return)	OPC C&C View and Deployment View
Adventure Order Processing DB (executes on srv-dbopc)	The Order Processing Component uses this for consumer account data, consumer purchases and external invoices (R/W)	Consumer Website (mostly read for authentication, write only at account creation and update) Social features (read – need to characterize workload) Cross-sell (read – once per purchase)	OPC C&C View and Deployment View
Consumer UI (executes on srv-web1 and srv-web2)	Primary browse and purchase workflows.	Social Features – insert new content for tagging, sharing, etc. Cross-sell – insert cross-sell features.	Top Level Uses View and Install View and Deployment View

Architecture Documentation Questionnaire

Now you are going to use the models provided in the view of the AB system to answer some questions. We have a few questions related to each of the new capabilities. Obviously, in a real integration project, there would be a multitude of questions: Here, we attempt to cover a small sample of these questions to assess the utility of the models.

In each answer, please note which models you consulted to make your decision. If you cannot find sufficient information in the models to make your decision, please state this (these questions were formulated without consulting the models).

First, let's consider our approach to achieving the first new capability: adding social features to the AB system. These social features will create runtime dependencies between the AB system and the Social Travel system.

Question	Your Answer
1. The AB system has 7x24 availability. The Social Travel systems have had recent outages, and there is ongoing work to improve availability. Which AB stakeholders do you need to engage with to understand the AB availability requirements?	
2. You need to expose the social features in the AB web user interface, which uses gwt (Google Web Toolkit). The Social Travel system uses V2.7.0 of the gwt. Which version does AB use?	
3. Is there an external programmatic interface to access the User Account and Purchase Order data elements?	

Next, let's consider the second new capability, which requires us to replace the payment processing interface on the AB system. AB calls this interface the "Bank" interface.

Question	Your Answer
4. The inputs to the new payment processing interface are: Card Type, Card Number, Card Expiration, and Card Security Code/CCV. Can the current AB system provide all of these elements?	
5. Using the new payment processing interface may impact the purchase completion latency of the AB system. Which stakeholders do you need to consult with about the performance requirements?	

Now, let's consider the third new capability, which adds cross-selling to the AB purchase user interface. Our approach is to have the AB system make a request to a Social Travel system when the purchase order has been placed, the necessary details about the travel order (e.g., traveler's gender, destination, date, activity type, etc.) and the Social Travel cross-sell business logic will return a list of five items to offer to the traveler.

Question	Your Answer
6. Does the AB system have existing request-response interfaces with external systems? If so, what protocols/technologies are used for these interfaces?	
7. Is the traveler's gender available in the AB system?	
8. Does an AB purchase order contain the activities that are included in the adventure?	

Architecture Documentation Questionnaire

Finally, let's go back to the three questions that you listed at the very start of the exercise. Try to use the models to answer these questions – if one of your questions duplicates one of the previous questions, you can just note that here.

In each answer, please note which models you consulted to make your decision. If you cannot find sufficient information in the models to make your decision, please state this.

Question	Your Answer
Your first question (There is no need to copy the question again here, unless you find that helpful)	
Your second question	
Your third question	

Two wrap-up questions:

Question	Your Answer
How much time did you spend on this exercise?	

Do you have any comments on the exercise? Did the role-playing questions represent the types of questions that an architect responsible for this integration might ask? Did you notice any gaps in the coverage of the models that was not exposed by the questions? Do you have any comments about the utility of the models in this viewpoint?

Your Answer:

This is the end of the exercise. Thank you for your contribution. We will share the results with you as soon as we complete the analysis.