

An Implementation of Java Native Interface (JNI) to Replace a Socket

Brandon Hessler
Penn State - World Campus
Enterprise Integration
SWENG568 Section 1
Program: Software Engineering M.S.

April 11, 2017

Contents

| | | |
|----------|---|----------|
| 1 | Project Introduction | 2 |
| 1.1 | Business View of the Integration Needs | 2 |
| 2 | Workflows or Process Flows Design (e.g., Integration logic across integrated applications) | 2 |
| 2.1 | Data Flow | 2 |
| 2.2 | Control Flow | 2 |
| 3 | Integration Design (i.e., Approach to implementing the identified integration logic) | 2 |
| 3.1 | Architecture Models (An architectural model is an overall structure of your systems, which provides the general relationship of highlevel components across integrated applications.) | 2 |
| 3.2 | Functional Models (A functional model is essentially a list of specific functions to support a given integration need. It provides an overview of the essential functions supported by integrated applications and how they are related.) | 2 |
| 4 | Implementation and Programming | 2 |
| 5 | Bibliography | 2 |

1 Project Introduction

1.1 Business View of the Integration Needs

2 Workflows or Process Flows Design (e.g., Integration logic across integrated applications)

2.1 Data Flow

2.2 Control Flow

3 Integration Design (i.e., Approach to implementing the identified integration logic)

3.1 Architecture Models (An architectural model is an overall structure of your systems, which provides the general relationship of highlevel components across integrated applications.)

3.2 Functional Models (A functional model is essentially a list of specific functions to support a given integration need. It provides an overview of the essential functions supported by integrated applications and how they are related.)

4 Implementation and Programming

5 Bibliography

- JAX-WS

I used JAX and the @WebService annotations in order create a document style web service. JAX-WS is an easier way for me to implement the SOAP style functionality but with an easier time than doing so without an API. JAX-WS is an easy and secure way for Java to implement the RESTful Web Service that it seems most people use these days.

- wsgen

Using wsgen was not an easy thing to accomplish. I had to spend a lot of time trying to throw different commands at this in order to get it to work. The biggest problem was that I was not calling the wsgen function from the correct place, I was calling it from the code path instead of the compiled code path (in the out folder). This generated the two java classes I needed in order to aid with the deploement of the Web Server. One is able to accomplish the same task by writing it by hand (I realized this after) and it would have been much less time consuming in my case. But for a complicated Web Server

with a lot of calls, I could see how this could speed things up tremendously. Screenshot of me running this is in Figure ??.

- w3c dom

I almost decided not to do this step since I already had the web service working and could take things from a file and send it over the Web Service before I started to use it. But it didn't quite sit right with me only having a hard coded string that would be sent. I instead wanted to use the XML file that was given to us in the lesson to find the correct student to send over the network. Once I decided this I needed to learn how to traverse an XML document in Java without parsing it into one or more objects. Obviously this would have been much easier, but most likely an unnecessary step in the real world. It makes much more sense to look through the student data that is already in the XML file (produced by whomever) and send the requested data as is so that they can parse what they need from it. Through my research I learned that the easiest way to do this was with the w3c dom package, which lets you traverse the Nodes of the XML file and get the requested data. That data is then sent over the Web Service and received by the client.