**HiPerVison Projects on Shelton Vision's Textile Inspection Systems: Backend web-based interface**

Malavika Reghunathan Nair

14 July 2014

A preliminary report submitted in fulfilment of the requirements for Course CO7501- Individual Project, Department of Computer Science, University of Leicester.

[Word count = 1682]

**Summary:**

This project involves working on the Shelton WebSPECTOR surface inspection system which is currently used in various industries to inspect materials for defects. The system uses a combination of vision hardware (lights, cameras and electronics) and software. There are two primary software platforms. The first is the front-end (written in C#) where the operator interface, defect analysis and system co-ordination is done. The second is the back-end (written in C++) where the image processing and defect detection is carried out.

The outline aim of the project is to explore the possibilities of using existing web technologies and services to provide a web-based interface to allow engineers to interrogate the back-end remotely to troubleshoot any faults in identifying defects and adjust parameters.

**Table of Contents:**

1. Introduction
2. List of Requirements
3. Technical Specification
4. Background material
5. Project plan

**1. Introduction**

The aim of the project is to provide a web-based interface to allow engineers to interrogate the back-end remotely to troubleshoot any faults in identifying defects and adjust parameters.

The main challenge of this project is that it is client-server programming and to create a platform-independent web service that can communicate with C++ applications and a client-side web application. The existing system works under the Microsoft windows XP operating system. It has been developed over 15 years with a mixture of Visual Basic 6, C and assembler libraries, SQL procedures, and WiT. It will be re-written in C++ (back-end) and C# (front-end).

From time to time there is a need for an engineer to interrogate the back-end to see why it is not detecting defects and adjust parameters. These parameters are linked to the product database. The information the engineer sees is a mixture of material images, process graphs and other displays. The engineer is able to control the material position during a material playback mode.

The project will be supervised in cooperation with the client, who will provide the requirements and technical support based on their understanding of their needs. The project thus combines technical and academic challenges with that of interacting with a real client, e.g., requirements may change during the project.

This project is a feasibility study where the main task is to evaluate the different technical solutions.

**2. List of requirements (objectives):**

**High-Level Requirements:**

Essential

* Review the available technologies
* Create a web service in Java to communicate with the existing C++ backend and MSSQL Database.
* Create client side using Javascript and JQuery
* Enable a protocol for being able to change database fields that the web server and client can both access. The change will then be seen on the client displays

Recommended

* The definition of a more complete API for the backend

Optional

* An extended client side prototype exploiting the API

**Detailed Requirements:**

-Create a web page (client side) with

a) Buttons that tell the server they have been pushed.

b) Data field that

i) can change a field in a database when and update button is pressed.

ii) not be directly linked to the database but request that the server updates the database (for very remote use).

c) Upon a button push, display a stream of images being sent from the server

d) Upon a button push, display a graph using one of the following methods..

i) Display graphs pre-generated.

ii) display graphs from vector and axis scale data. (preferred).

**3. Technical Specification**

Considerations:

-The client side could be implemented using jQuery and Javascript. **jQuery** UI has a series of ready-made, great-looking user interface widgets and a comprehensive set of core interaction helpers designed to be implemented in a consistent and developer-friendly way. With all this, the amount of code that you need to write personally to take a project from conception to completion is drastically reduced [1].

High-level Architecture:

Figure1.1 illustrates a high-level view of the scenario.

Database (SQL Server)

Result query/update

Client Side Web Application (Part of this project)

Camera

Get image details

HTTP Request/Response

**(XML)**

*Fig 3.1: High-level architecture*

**4. Background material**

The web is a distributed, dynamic and large information repository.[3]Services are self-contained modules-deployed over standard middleware platforms- that can be described, published, located, orchestrated and programmed using XML based technologies over a network.[4]

Three basic components of Web services are-

* **SOAP** (Simple Object Access Protocol)
* **UDDI** (Universal Description, Discovery and Integration)
* **WSDL** (Web Services Description Language)

SOAP defines a communication protocol for web services. WSDL enables service providers to describe their applications. UDDI offers a registry service that allows advertisement and discovery of web services.[3]

Discovery Service

Query Publish

Service

Provider

Service

Requestor

Bind & Interact

*Fig 4.1: Service Oriented Architecture*

The steps how a **client-server** works can be illustrated as follows:

1. The client program wraps the information into a SOAP message
2. As an HTTP POST request, this SOAP message is sent to the web service
3. The web service unwrap the SOAP message and make it understandable to the application. The application process the request and return the required result
4. The web service again packs the result into a SOAP message and sends it back to the client as a response to the HTTP request
5. The client program now unpacks the SOAP message and gets the result.

Two most widely used approaches for web service development are SOAP and REST (Representational State Transfer).REST has been accepted widely as a simpler alternative to SOAP and WSDL based web services. REST defines a set of architectural principles by which you can design Web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP by a wide range of clients written in different languages [8].

There are many different platforms for developing services and their clients. One such environment can be set up using the following:

* + - **Eclipse:**[**https://www.eclipse.org/downloads**](https://www.eclipse.org/downloads)
    - **Java**: **http://www.java.com/en/download/index.jsp**
    - **Oracle Weblogic 12c :http://www.oracle.com/technetwork/middleware/weblogic/documentation/index.html**
    - **Jersey Framework: https://jersey.java.net/**

A major requirement in this project is to integrate the web service written in Java with C++ applications. A Java application may communicate with a native application through a TCP/IP connection or through other inter-process communication (IPC) mechanisms [5]. Java Native Interface is an alternative solution for that. But the disadvantage of JNI is that the type-safety feature of Java programming language cannot be guaranteed. a crashed native application does not immediately terminate the Java application with which it communicates over TCP/IP [5]. Sometimes, however, it is necessary for a Java application to communicate with native code *that resides in the same process*. This is when the JNI becomes useful [5].

**Java Native Interface (JNI)** is a powerful feature of the Java platform. Applications that use the JNI can incorporate *native code* written in programming languages such as C and C++, as well as code written in the Java programming language [5]. JNI is a two-way interface than can invoke native code and vice versa [5]. Figure 4.2 illustrates the role of the JNI.

Native Application and Library

JNI

Java Virtual Machine Implementation

Java application and Library

Host Environment

*Fig4.2: Role of JNI [5]*

There are several frameworks and technologies available for web development. Listed below are a few among them:

**Spring Framework** is another framework which can be used to develop web interface. The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform [6]. It can be used to build a RESTful web service, consuming the RESTful Web service, access relational data using JDBC etc.

**JavaServer Pages (JSP)** technology enables Web developers and designers to rapidly develop and easily maintain, information-rich, dynamic Web pages that leverage existing business systems[7].JSP is an extension of Java Servlet Technology.  Together, JSP technology and servlets provide an attractive alternative to other types of dynamic Web scripting/programming by offering: platform independence; enhanced performance; separation of logic from display; ease of administration; extensibility into the enterprise; and, most importantly, ease of use [7].

1. **Project Plan**

Please find the Gantt chart below:



**References:**

[1] “JQuery UI 1.8. The User Interface Library for JQuery”,by Dan Wellman

[2]”Apache CXF Web Service Development-Develop and deploy SOAP and RESTful web services”, by Naveen Balani and Rajeev Hathi,PACKT PUBLISHING, BIRMINGHAM-MUMBAI

[3]”Foundations for Efficient Web service Selection”by Qi Yu, Athman Bouguettaya, SPRINGER US, 2010

[4]”Web Services & SOA Principles and Technology”, by Michael.P.Papazoglou, SECOND EDITION,PEARSON

[5]”Java Native Interface,Programmer’s guide and Specification”, by Sheng Liang, ADDISON-WESLEY

[6] Spring Framework [online], Accessed date [11/07/2014]

Available at:<http://projects.spring.io/spring-framework/>

[7] JavaServer Pages Overview [online], [Accessed date: 12/07/2014]

Available at:<http://www.oracle.com/technetwork/java/overview-138580.html>

[8] RESTfulWeb services: The basics [online], [Accessed date: 14/07/2014]

Abailable at: <http://www.ibm.com/developerworks/webservices/library/ws-restful/>

[9]Why JChartFX?[online],[Accessed date:14/07/2014]

<http://support.softwarefx.com/jChartFX/article/2501037?showassample=true#!2501236>