CMU 17-655 & 17-755 Architecting Enterprise Systems Fall 2012

Web Scraper P1 Write-up Due: 26th September

# Architectural Drivers

## Business Constraints

In this project the business constraint assumed was that the application should be delivered quickly, because the faster the result of this application comes, the faster we will determine whether we want to continue in this business or not. In this project, we based our decisions on being fast on delivery, because without a fast solution, we might waste time on a business that might have no potential.

## Technical Constraint

In this project we assumed that the infrastructures of the existing systems (Sales, infrastructure, and inventory) are Java based or at least easily integrated to Java. So our application will be written with Java.

### Quality Attribute

The quality attribute that was assumed was modifiability. To articulate modifiability further we used the six-parts scenario.

#### Modifiability

##### Scenario

|  |  |
| --- | --- |
|  | Modifiability |
| Source | ACME bicycle |
| Stimulus | Wants to add more details to the XML |
| Environment | After initial deployment |
| Response | The new details can be added |
| Measure | In less than one day |

##### Business relation

This quality attribute is derived from the fact that ACME Bicycle is new to the field, and they might realize that they need to add more details about their products. The data representation should be flexible in adding more detail.

# Process of execution

## Content of the package

The packaged file contains the following:

The applications: which contains the source files and build script.

XML schema: which is the schema of the XML data model of our applications.

ReadMe.docx: this file.

## Executing Application 1

In your file system command-line, traverse to the root of the application. Then run the following ant command:

ant runApplication1

This command will execute the compilation and running of the application.

When the application runs, it will show a list of all the brands with an index for each of them. Choose the index of the brand you want to get results for. After that the execution of the application it will produce an XML file with the needed content. The file will be located in build/jar/brand-<the brand you chosen>.xml.

Note: when a user writes his input, it does not echo back in the command line.

## Executing Application 2

In your file system command-line, traverse to the root of the application. Then run the following ant command:

ant runApplication2

This command will execute the compilation and running of the application.

When the application runs, it will merge all the files that were produced in Application 1 and put the result in build/jar/unified.xml file.

## Executing Application 3

In your file system command-line, traverse to the root of the application. Then run the following ant command:

ant runApplication3

This command will execute the compilation and running of the application.

When the application runs, it will read the file build/jar/unified.xml and it will produce brand names to the user. Then the user has to write the name of the brand. If the brand name is entered, then the application will produce an HTML file in build/jar/output.html which contains a human readable form of the models of that brand.

If there are any problems with the execution, please do an ant clean command to clean up and then try again.

Note: when a user writes his input, it does not echo back in the command line.