In defining the elaboration phase in RUP, Rational states the following:

“While the process must always accommodate changes, the elaboration phase activities ensure that the architecture, requirements and plans are stable enough, and the risks are sufficiently mitigated, so you can predictably determine the cost and schedule for the completion of the development.”

We have identified several assumptions that must be evaluated in order to convey an assurance level of stability, anticipate scheduling and cost requisites (especially for education of staff in proprietary languages and specific communication protocols), and to assist in formulating an accurate BOM. These assumptions are as follows.

1. The terminal can communicate sufficiently to and from the UART.
2. The UART can communicate sufficiently to and from the BS2P40 Microcontroller.
3. The Microcontroller program can be created sufficient enough to accomplish the task of counting eggs.
4. The communication protocol RS232 can be converted to RS485 and back sufficiently.
5. The communication protocol RS485 can be driven sufficiently by the high-level language chosen for the primary application, housed on the existing POS (the terminal).

These have been identified as contingent of the following DP’s associated with their corresponding FR’s, which is an excerpt from the group’s FR-DP decomposition.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FR –DP Decomposition

|  |  |  |  |
| --- | --- | --- | --- |
| FR2.1  | Shall run on Windows OS  | DP2.1  | Utilize Existing POS  |
| FR2.1.1  | Primary Application  | DP2.1.1  | .Net or Java Based Application  |
| FR2.1.1.1  | Shall have a Bridge to Communication Driver  | DP2.1.1.1  | Java: JNI , C++:Link C driver; both utilizing windows.h  |
| FR2.1.1.2  | Shall have a Hardware Polling Process  | DP2.1.1.2  | .Net or Java Control Structure Implemented to Poll Modules  |
| FR2.1.1.3  | Shall have a State Logic Process  | DP2.1.1.3  | .Net or Java Control Structure Implemented as a State Machine  |
| FR2.1.1.4  | Shall have a GUI Build/Update  | DP2.1.1.4  | .Net: Windows Forms, Java:Swing  |
| FR2.1.2  | Shall utilize a Serial Communication Driver  | DP2.1.2  | C driver created to drive communication  |
| FR2.1.3  | Shall utilize the USB Standard  | DP2.1.3  | Software USB capable  |
| FR2.1.4  | Shall utilize the RS485 Standard  | DP2.1.4  | Convert communication signal to RS485  |
| FR2.2  | BS2P40  | DP2.2  | Utilize the BS2P40 Microcontroller  |
| FR2.2.1  | Shall have a RS485 to RS242 Convertor  | DP2.2.1  | Hardware Converter for RS232 to RS485  |
| FR2.2.2  | Shall utilize an UART  | DP2.2.2  | Utilize provided UART  |
| FR2.2.3  | Shall utilize Communication Software  | DP2.2.3  | Hardware Implemented  |
| FR2.2.4  | Shall have Counting Software  | DP2.2.4  | PBASIC Control Structure to count eggs from hardware device  |
| FR3  | Data Access Layer  | DP3  |  |
| FR3.1  | Primary Application  | DP3.1  | .Net or Java Based Application  |
| FR3.1.1  | Shall utilize a File Writer  | DP3.1.1  | .Net: TextFileWriter, Java:FileWriter  |
| FR3.1.2  | Shall utilize a File Reader  | DP3.1.2  | .Net: TextFileReader, Java:FileReader  |
| FR3.1.3  | Shall have a Parser/Tokenizer  | DP3.1.3  | .Net or Java based object to parse .ini files  |
| FR4  | Persistence Layer  | DP4  |  |
| FR4.1  | Primary Application  | DP4.1  | .Net or Java Based Application  |
| FR4.1.1  | Shall utilize Configuration Settings  | DP4.1.1  | Configuration txt file  |
| FR4.1.2  | Shall have Log Files  | DP4.1.2  | log txt file  |
| FR4.1.3  | Shall have State Values  | DP4.1.3  | Configuration txt file  |
| FR4.2  | BS2P40 Microcontroller  | DP4.2  |  |
| FR4.2.1  | Shall have an Address  | DP4.2.1  | PBASIC Object stores address  |
| FR4.2.2  | Shall have State Values  | DP4.2.2  | PBASIC Objects can remember state  |

 |

Please note that colors are arbitrarily chosen purely for aesthetics and should not be considered to have deeper meaning.

Though the following information was requested in table form, I believe that the attached flow chart will more specifically help identify how our group, as per each individual, is verifying these assumptions, and in turn identifying individual progress of individuals (as per requested).

These assumptions have been broken down further into segments in the flow chart to assist in verifying these assumptions, and which group members are tackling which segment. Note that other tasks are being accomplished in parallel, so this in no way identifies total labors of all members.

Also, note that some functional requirements, such as DP3.1.1, DP3.1.2, and DP2.1.1.1, will be implemented naturally through the developmental environment chosen, and to add clarity to the DSM and Design Matrix, these cells have been left gray to illustrate that they are not applicable to the coupling concerns of the respective matrices. These functional requirements will be met through the Design Parameters listed, and as listed in the Design Parameters they will be implemented by the structure listed included in the language chosen. This is also reflected at a high level view in the Application Architecture V3.0. It should be easily understood through the Design Parameters which structure native to the language will be utilized.