**Meeting Minutes**

**Date:** March 11, 2014

**Start Time:** 3:30pm

**End Time:** 6:00pm

**Members Present:** Drew Aaron, Michael Beaver, Clay Boren,

Chad Farley, Andrew Hamilton, and Travis Hunt

**Members Absent:** N/A

**Topics** **Discussed**

* Architectural Design Traceability
* Detailed Design Traceability
* Backend Data Flow
* Custom Frontend User Interface Component

**Decisions and Actions Taken**

Andrew and Michael worked on traceability for the Architectural Design. All features and functionality in the Frontend are present in the Software Requirements Specification document (SRS) and vice versa. The only change involved renaming the “New File Dialog” component to “New File” and relocating it to under Ancillary Functionality. The Backend architecture met all traceability requirements at this time.

Andrew and Michael also worked on traceability for the current iteration of the Detailed Design. The Data class hierarchy and the Error Detection class have no traceability problems at this time. The maximum machine operations table size within the Library class was changed to a constant value (e.g., MACHINE\_OPS) to facilitate the addition of future instructions. The Byte class was amended to include a string representation of its value. The Memory class gained the BytesToString and the StringToBytes methods. The Register class’s convert method was replaced with the BytesToString and the StringToBytes methods. The SymbolTable class was split into two classes SymbolTable and LiteralTable that inherit from the abstract Table base class. The SymbolTable class’s hash table was reduced in size to 211 entries,[[1]](#footnote-1) and the LiteralTable class’s hash table was sized to 53 entries.[[2]](#footnote-2) Finally, the Pass2Translator class was amended with the ProcessLine method. All classes were updated to adhere to the ASSIST/UNA coding standards. See the attached updated class diagrams.

Chad and Travis worked on the Backend dataflow diagrams. The Frontend user interface will save a temporary file containing the program source code before actual Assembly occurs. This temporary file will be loaded by the Assembler. This gives an extra level of abstraction between the Frontend and the Backend over using just a long string, and it provides the simplicity of reading one line at a time. Output pathways from the Backend to the Frontend user interface will be determined at a later date. See the attached dataflow diagrams.

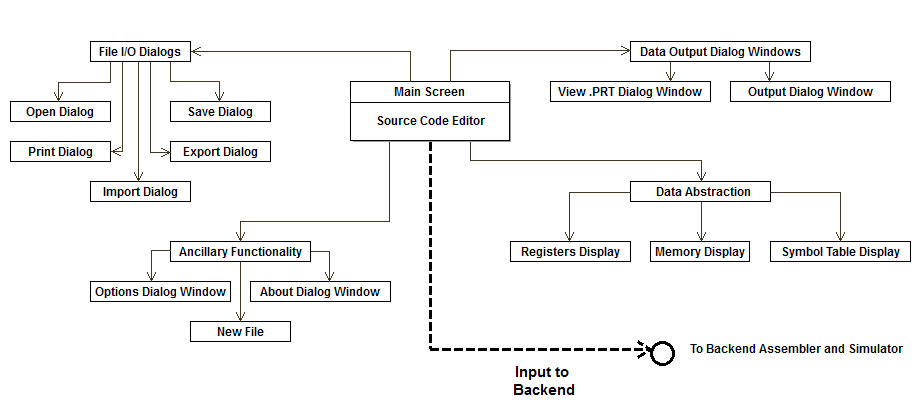
Drew and Clay worked to create the custom Source Code Editor for the Frontend user interface. They researched how to extend the native RichTextBox class. In particular, they need to make sure the content of the Source Code Editor is all uppercase. They also need to add line numbers and column numbers.

The client would like to hold a meeting on March 20, 2014 to discuss the team’s progress thus far, plans for Spring Break, and plans for after Spring Break.

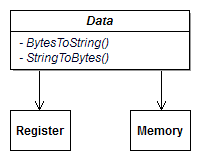
The next team meeting will be March 13, 2014 at 3:30pm in the Commons.

**Supplementary Information**

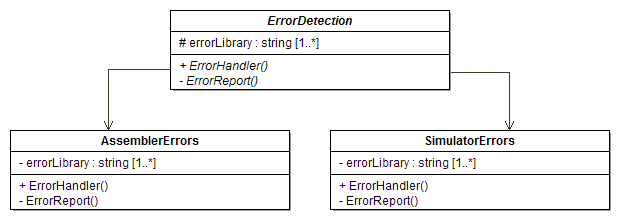
**Updated Frontend Architecture**

****

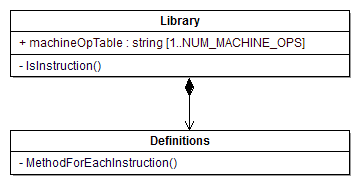
**Updated Data Class Diagram**

****

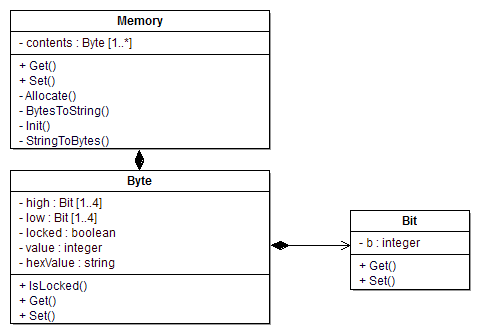
**Updated Error Detection Class Diagram**

****

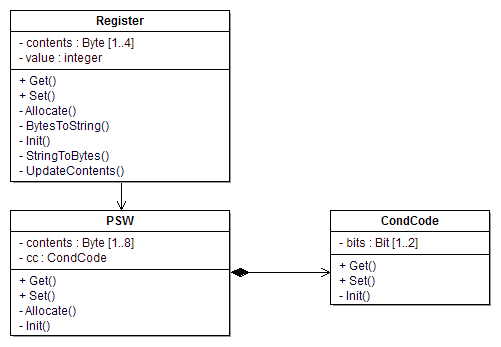
**Updated Library Class Diagram**

****

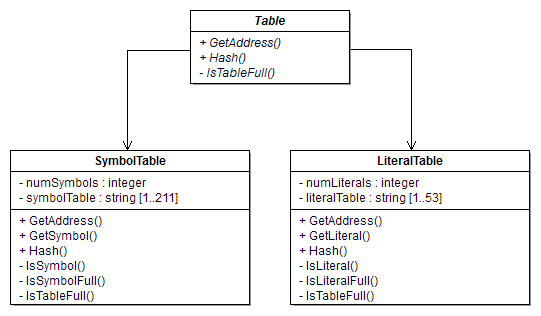
**Updated Memory Class Diagram**

****

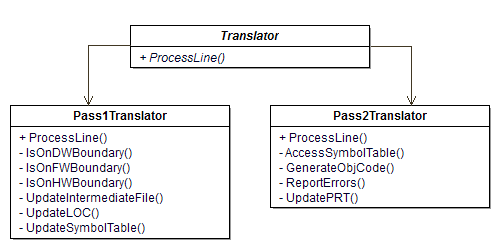
**Updated Register Class Diagram**

****

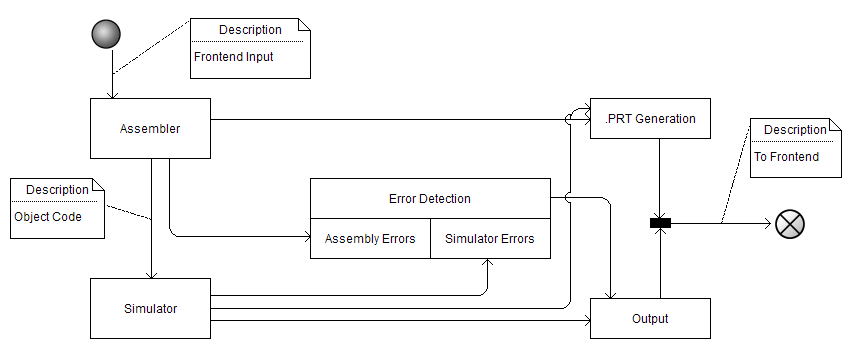
**Updated Table Class Diagram**

****

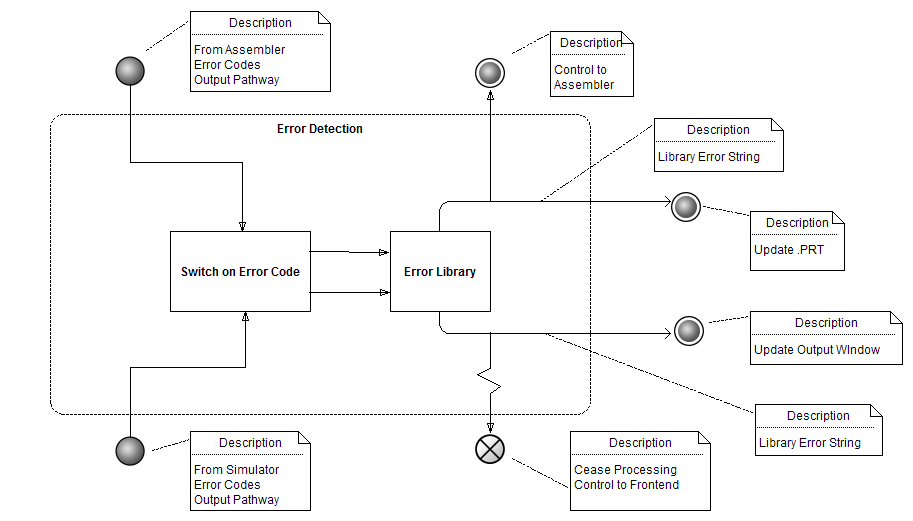
**Updated Translator Class Diagram**

****

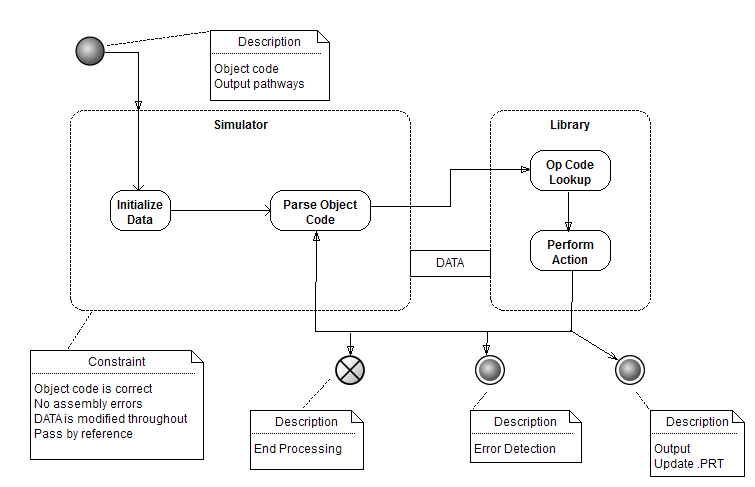
**Backend Dataflow Diagram**

****

**Error Detection Dataflow Diagram**

****

**Simulator Dataflow Diagram**

****

1. 100 symbols \* 2 = 200 + 11= 211 → prime number table size for improved table performance. [↑](#footnote-ref-1)
2. 25 literals \* 2 = 50 + 3 = 53 → prime number table size for improved table performance. [↑](#footnote-ref-2)