**Programming Language Proposal**

At this time the development team wishes to propose the Visual C# language as the main programming language for this project. Microsoft’s Visual C# language is inherently supported by Microsoft’s Visual Studio environment, which is compatible with the team’s established Team Foundation Server versioning protocols. While Visual C# has drawbacks, the team believes it is a viable programming language for this project.

**Strengths**

The greatest strength afforded by Visual C# is its familiarity. Visual C# can be viewed, in simplistic terms, as a fusion of Java and C++. Since the team is familiar with either or both languages, there will be almost no significant retraining necessary. Furthermore, Visual C# is capable of running C++ code, which is particularly useful if a team member wishes to implement code in C++ before converting it.

As its name implies, Visual C# is a visual, GUI-based language. As such, it will be relatively easy to build the ASSIST/UNA GUI with the tools provided by Visual Studio. When the team needs to develop its own GUI components, the Visual C# language shall prove quite useful and adaptable. Visual C# also has the benefit of being less restrictive on coding. For example, pointers are more easily implemented in C# than in C++ or similar languages. Also, Visual C# is capable of concurrency and threading. Should the team decide that threading is necessary for this project, this feature will be particularly useful. Finally, if the need arises in the future, Visual C# is also useful for implementing server and network functionality.

**Weaknesses**

Like all programming languages, Visual C# has drawbacks. Namely, Visual C# code can be rather messy and unintuitive. However, this problem may be easily remedied with constant practice. Also, unlike C++, Visual C# is not particularly well-suited for Object-Oriented Programming. This should not be too difficult to overcome since most of the team is fairly experienced with the OOP paradigm. Like C++, memory management in Visual C# can be problematic. Hence, the team will need to be especially cautious when using dynamic memory.

**Conclusion**

The Visual C# programming language is the perfect candidate for this project. Visual C# offers the familiarity of Java and C++ without too much compromise. Moreover, Visual C# is a GUI-based language, which will make designing and implementing the ASSIST/UNA frontend a relatively simpler task. While Visual C# does have notable weaknesses, the team believes, at this time, that no other alternative is viable for this project.

Dr. Patricia L. Roden, *Client* Travis Hunt, *Team Leader*

Michael Beaver, *Technical Writer* Andrew Hamilton, *SQA*

Drew Aaron Clay Boren

Chad Farley