GSAW 2008 Tutorial A:

Building Solid Ground System Architectures: Architecture Principles and UML Views

Length: Full day

Overview:
Defining an effective software architecture is essential to the successful development of distributed ground system software. The definition of an architecture that meets the project's needs and the communication of this complex set of information can be a daunting task for even the most experienced software architects.

This tutorial provides an overview of software architecture views and principles. Initially the tutorial describes the development of component and deployment views based on IEEE 1471 and UML. Next, software architecture principles such as abstraction, separation of concerns, coupling and cohesion, and modularity are discussed in the context of software architecture. For each principle, the tradeoffs and issues are illustrated using the previously described architectural views.

The techniques and principles described in this tutorial are based on a significant experience base architecting large software systems including several satellite ground systems. As a result, the approaches, principles, and examples described here form a practical basis for software architects working on such systems. The tutorial includes hands-on exercises to give students a chance to apply the techniques for themselves.

Instructor: Jeff Garland, CrystalClear Software, Inc.

Biography:
Jeff Garland has worked on many large-scale software projects over the past 20+ years, in many different domains, including telephone switching, industrial process control, satellite ground control, and financial systems. He has served as both the lead architect and a member of the architecture team on several of these projects. Mr. Garland holds a Master's degree in Computer Science from Arizona State University and a Bachelor of Science in Systems Engineering from the University of Arizona. He is currently President and Principal Consultant for CrystalClear Software, a consulting firm that specializes in the development of software architectures for large-scale systems.

Who Should Attend:
Basic understanding of UML. Experience as a software developer. Interest in software architecture. Experience with large software systems beneficial.