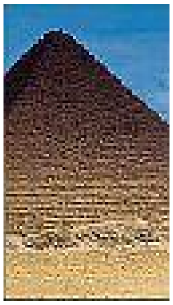

Pyrrhus Software

Enduring Solutions

info@pyrrhusoft.com
training@pyrrhusoft.com

Modeling with SAE AADL

In the world of Systems Engineering, there is a lot of confusion on what is in hardware and what is in software. In addition, there is a desire in some domains for a generic operating architecture where all systems (eg, weapons systems) can play together **à** plug and play weapons. One aspect of the world of complex systems is describing the architectures and then analyzing them to determine if it is feasible to actually build the system. In addition, the use of a reference architecture as platform for the generation and maintenance of families of systems is also desirable.

Research has been going on since about 1989 to come up with an architecture description language or modeling language. For example, UML provides a level of software systems modeling as does XML. The architecture description language (ADL) world is pushing the software model further into the world of systems that include both software and hardware components. For example, suppose that you need to construct a communications system on an aircraft that actually makes use of several different hardware stacks to transfer the data through the aircraft. At the highest level of modeling, this looks like a simple client-server interaction between two components that plug into a bus. Further analysis of the system allows you to expand the bus to find that it is actually an execution platform that plugs into another bus. The architecture description language enables system's engineers as well as software engineers to build systems of various components, including systems of systems, to determine the feasibility of data flow, interconnections, timing, performance, etc.

The Architecture Analysis & Design Language (AADL) is a textual and graphical language used to design and analyze the software and hardware architecture of real-time systems and their performance-critical characteristics. The language is used to describe the structure of such systems as an assembly of software components mapped onto an execution platform. The language can describe functional interfaces to components (such as data inputs and outputs) and performance-critical aspects of components (such as timing). The language can describe how components interact, such as how data inputs and outputs are connected or how application software components are allocated to execution platform components. The language can also describe the dynamic behavior of the runtime architecture by supporting the modeling concept of operational modes and mode transitions. The language is designed to be extensible to accommodate analyses of the runtime architectures that the standard language does not completely support. Extensions can take the form of new properties and analysis specific notations that can be associated with components.

This 2-day course is designed for software engineers who interested in learning about the features and facilities available in AADL to model real-time, safety-critical systems.