Executive Summary

IEEE Real-Time Distributed Systems Communication (POSIX 1003.21)

PNUMs 11A, 11B, and 11C

Background: This effort supports the continued standardization efforts of the draft "Real-Time Distributed System Communication Application Program Interface (API)" (IEEE POSIX P1003.21). The results of this effort further the development of POSIX 1003.21 to becoming an approved standard. POSIX 1003.21 is listed as an emerging standard in section 2.3.5 of the DoD *Joint Technical Architecture (JTA)* version 1.0 (and section 2.2.3.5.1 of the first draft of version 2.0, dated 18 July 1997). The activities herein are intended to elevate the status of POSIX 1003.21 in the JTA. Additional effort is needed to coordinate: a common standard for Avionics Operating Systems with the Joint Strike Fighter program, IEEE Fault Management Group, and other DOD working groups related to the Joint Technical Architecture Weapon Systems Annex.

POSIX 1003.21 provides an interface to a message-passing mechanism for distributed systems. The intent of POSIX 1003.21 is to allow applications to be portable across different POSIX 1003.21 implementations. Capabilities of POSIX 1003.21 which support real-time performance issues include: asynchronous operations, prioritized messages (integer and/or deadline), bounded blocking operations, explicit buffer management, and support for multiple communication models.

The working group has conducted a formal IEEE ballot on a Language-Independent Specification (LIS) and is preparing the LIS for a recirculation ballot. The working group also plans to document the following:

- Ada language binding,
- C language binding,
- Formal specification of interface semantics, and
- Results of POSIX 1003.21 technology implementation.

The successful completion of POSIX 1003.21 is fundamental to the real-time distributed systems community. POSIX 1003.21 has been developed to specifically meet the needs of the real-time distributed systems community. As such, it is intended to be used by application developers. However, it is sufficiently complete so that it may be used as a basis for building other, system-like applications. In other words, one could build a system management application on top of POSIX 1003.21. A dynamic load balancing system function could be an example of a system management application. In addition, there is consideration being given to the use of POSIX 1003.21 as a basis for the development of a real-time Common Object Request Broker Architecture (CORBA) implementation.

Objectives:

- Document draft standard,
- Ballot the LIS,
- Document formal specification of interface,
- Document draft of Ada language binding,
- Document draft of C language binding,
- Ballot the Ada language binding,
- Ballot the C language binding, and
- Document technology implementation.

Subtask Descriptions:

- Documenting and Balloting of the Draft Standard: The P1003.21 working group is documenting the draft standard addressing real-time distributed systems and resolving issues raised during the balloting of the draft POSIX 1003.21 standard.

- LIS for IEEE ballot: A Language Independent Specification (LIS) provides a common interface that can be used to define the semantics that apply to all language bindings (e.g. Ada and C). A major advantage of an LIS over a specific language binding is that it is presented in a programming-language neutral form. Hence, the LIS is free of any constraints imposed by a particular programming language. In addition, an LIS reduces the risk that a specific binding precludes the implementation of features in another language. As a result, a properly-defined LIS allows the language bindings to take advantage of features of the programming language.

- Language bindings: Although an LIS provides the functionality of an interface, an LIS does not provide an interface bound to a specific programming language. Language bindings define how the LIS interface is bound to a particular programming language. The P1003.21 working group will define two language bindings for the P1003.21 LIS.
  - Ada language binding: The working group is committed to developing an Ada binding for this standard. The working group will use the results of the technology implementation effort at RTIS as the basis for this work. The working group has initiated some preliminary work on an Ada95 language binding.
  - C language binding: The working group is committed to the development of a C language binding. The working group will use the results of the technology implementation effort at the SEI as the basis for this work.
- Formal specification of interface semantics: It is crucial that the semantics of the LIS be properly specified. To ensure this, the working group is developing a formal specification of the interface semantics using the Z notation. The Z notation is a formal specification language based in mathematical theory. The working group has already realized benefits from this work in that it has exposed areas in which the LIS was inconsistent. The working group will publish this formal specification as part of the LIS.
- Technology Implementation Objectives: In prior years, the SEI and RTIS have implemented prototypes of the P1003.21 functionality. The SEI's prototype was written in C, whereas RTIS' was written in Ada. For FY 1997, both the SEI and RTIS will update the APIs of their prototypes to keep the interface compliant with the LIS. This work will ensure the working group that changes to the interface are still implementable. All implementation experience developed during this effort will be made freely available to the public. This information may be used by any interested implementor for development of commercial products. In addition, the SEI prototype will be made freely available to the public to accelerate the development of COTS products by commercial vendors.

RTIS would extend a P1003.21 prototype being developed for the Lockheed JSF program and deliver to the OS-JTF draft IEEE standards applicable to this project. These standards may include IEEE Fault Management Standard, IEEE Check Point Restart, Society of Automotive Engineers AS-5 Avionics Standards. Currently, this prototype demonstrates the viability of using a commercial standard interface for JSF and evaluates performance characteristics of different network interconnects. The JSF prototype implements only a subset of the P1003.21 functionality. RTIS also actively participates in the AS-5 (Avionics Systems) subcommittee of the Society for Automotive Engineers (SAE). This subcommittee is currently defining an avionics profile for the POSIX APIs (including P1003.21). The P1003.21 prototype extension would add those capabilities of P1003.21 required by the SAE avionics profile that are not supported by the JSF prototype.

Motivation and DoD Applicability: The original view of POSIX was that of a single workstation operating system. Much of P1003.1 and its real-time extensions were of this scope. For this work, the concept of source code portability was very important. However, as POSIX has matured, its scope has increased to distributed systems where the concept of interoperability becomes more important.

Most DoD weapon systems and combat control systems are distributed. A distributed system consists of multiple programs executing on more than one processor. The processors in a distributed system may or may not be similar. In addition, these systems often have real-time requirements. Upon completion of the work in P1003.21, industry can use a formal standard to develop implementations that meet DoD needs for real-time distributed systems. In this sense, P1003.21 is critical. Formal standards such as P1003.21 reduce the cost of DoD systems by increasing the portability and thus increasing reusability.

Any distributed DoD program can benefit from the P1003.21 standard. DoD programs that fall into this category include, but are not limited to: Joint Strike Fighter, SC-21, Crusader Mobil Artillery, and Aegis Combat System.