HIGH-ASSURANCE CORBA FOR SOFTWARE-BASED COMMUNICATIONS

Kevin Buesing
(Objective Interface Systems, Herndon, VA, USA, kevin.buesing@ois.com)
Victor Giddings
(Objective Interface Systems, Herndon, VA, USA, victor.giddings@ois.com)

ABSTRACT

A high-assurance software-based communication system is one in which there is a high consequence attached to system failure. In other words, a high-assurance software-based communication system is a safety-critical and/or security system for which failure is not an option.

Creating a high-assurance, safety-critical certified application is a non-trivial effort. The criteria such systems must adhere to results in a system designer having a limited choice of operating systems, programming language, development tools, third party libraries, etc. Consequently, today's systems integrators are demanding software tools that meet safety-critical criteria.

This paper will describe the efforts to develop a high assurance profile for the Common Object Request Broker Architecture (CORBA). The constraints on the languages, tools and operating systems used in creating high-assurance distributed systems will be described. In particular, the language profiles, the consequent IDL mapping profiles, and the CORBA profiles will be discussed.

As a result of the creation of high-assurance CORBA, developers will have the middleware available to create software-defined radios which can meet the most stringent requirements of certification, including DO-178B Level A certification.

1. INTRODUCTION

Safety-Critical and High-Assurance systems today require software that must meet stringent criteria, focused in three main areas: reliability, safety and security. Traditionally these systems have been custom designed. Custom designed systems are difficult and impractical to expand and maintain. The industries that build these types of systems are looking to COTS vendors for solutions.

Several COTS vendors have produced High Assurance and or Safety-Critical real-time operating systems (RTOS). The availability of these products will create a demand for the same level of robustness in middleware. For high assurance and safety critical systems the RTOS is only part of the solution. Designers of these systems are looking to middleware for assistance in solving problems that middleware in other industries have historically addressed. The CORBA, Minimum CORBA and Real-Time CORBA specifications provide a solid foundation to begin addressing high assurance and safety critical middleware needs.

2. WHAT IS HIGH ASSURANCE?

High assurance software systems have extremely high consequences attached to system failure. During system design, specific artifacts must be produced to validate proper system functionality. The fields which typically require a high assurance system include: flight control systems, secure communication devices, medical surgery equipment, unmanned aerial vehicles, military command and control systems, and nuclear reactors. This type of system is also expanding into other fields such as the automotive industry and voice over IP (VOIP) phone systems. To the FAA high assurance means that the system is allowed one failure per one billion hours of operation. One billion hours is actually 114,077 years.

3. ACHIEVING HIGH ASSURANCE

Achieving high assurance in a software system requires many things. The design and development process must include meticulous traceability of requirements to code. It also requires a strong commitment to quality assurance. The design process itself may need to be monitored by a third party. In addition the code written must conform to a predictable and provable subset. The code produced must be compiled and linked with verified tools. Finally the language run-time and the operating system that it runs on must be predictable. If any link within the chain is broken the system will not achieve high assurance. The primary goal when designing and developing a system of this type is to keep it simple. Strict adherence to simplicity and limiting the code size can greatly assist the evaluation and certification effort. The overall goal of the design and development process is to allow the evaluation of the

Proceeding of the SDR 05 Technical Conference and Product Exposition. Copyright © 2005 SDR Forum. All Rights Reserved

resulting software. Currently certification varies greatly depending upon the target industry.

4. EXISTING INDUSTRY STANDARDS

The industries involved in high assurance and safety critical software have produced a variety of certification requirements based upon their own industry. The FAA has adopted RTCA DO-178B certification for airborne systems and equipment. Achieving this certification encompasses the entire project. The avionics industry has also produced the Avionics Application Software Standard Interface or ARINC-653 standard. This standard addresses time and space partitioning in order to prevent cascading failure of applications. Internationally the ISO-15408 or Common Criteria for Information Technology Security Evaluation standard has been produced. The US Government has also created the DCID 6/3. This standard is for the protection of sensitive compartmented information within information systems. It specifically outlines procedures for storing, processing and communication of classified intelligence. In addition safety evaluation is done on a system not on the building blocks of the system. This results in a limited ability to re-use past designs and discourages commercialization. The high assurance object request broker (ORB) must be applicable to different industries and must not hinder existing industry safety and security standards.

5. PLATFORMS AVAILABLE

Several platforms for use in high assurance and safety critical systems are available. They are typically referred to as a certifiable/certified RTOS. They are designed to conform to one or more of the standards discussed previously. Thee RTOSs are under consideration for the proof of concept of the high assurance CORBA ORB (Green Hills Software: Integrity-178B, LynuxWorks: LynxOS-178 and Wind River Systems: Platform for Safety Critical ARINC 653). Additionally the high assurance CORBA mapping needs to also consider the impact of running on a multiple independent levels of security/safety (MILS) Separation Kernel.

6. HOW TO ACHIEVE A HIGH ASSURANCE ORB PROFILE

The high assurance ORB profile that is currently being worked on in the OMG is focused on an overall goal: to allow evaluation of software that has a CORBA ORB in it for safety critical and high assurance systems. This ORB profile does not include details on how to achieve a high quality development process. It does not cover which high quality tools should be used. It also avoids any discussion on certification since this varies greatly by industry. It leaves these details to the certification standards for each industry. The high assurance ORB profile focuses on two language subsets and on how to keep the ORB simple to allow evaluation and certification.

6.1. Keep It Simple

The ORB must be “simple” since certification cost greatly exceeds development cost. Therefore the code size of the ORB run-time must be reduced. In order to achieve this, the ORB must have limits set on its functionality. For example the CORBA standard for ORB shutdown is quite difficult to implement and for many high assurance systems the shutdown process is simply to turn the device off. Therefore, the complete shutdown specification in the CORBA standard can simply be removed. Similarly the ORB must eliminate most if not all dynamic behavior allowing system resources to be allocated at program initialization and not dynamically. The resources include thread creation, memory allocation, run-time symbol resolution, run-time path resolution and transport connections.

6.2. Languages and Subsets

The CORBA product must also facilitate the generation of code that can be used in a safety critical system. The code generated must conform to a certifiable programming language subset or profile. In order to simplify certification the IDL compiler must generate code in the target language that is smaller than current CORBA products. In order to achieve the size reductions for the generated code certain IDL types will need to be restricted. At the same time the target language generated code must be highly optimized for size.

The profile of both the interface development language (IDL) and the idl compiler’s generated code must meet stringent requirements. Several candidate language subsets have been identified. Ada has the SPARC subset and the Ravenscar run-time restrictions. The motor industry software reliability association (MISRA) C subset has also been identified. Finally C++ has had discussions of possible safe subsets as well. The two safe subsets being considered in the high assurance ORB mapping are Ada and C++.

The high assurance ORB mapping needs to consider all of the language profiles to facilitate interoperability between languages. Several language features can not be used. For example all dynamic binding must be avoided. This means virtual inheritance or virtual functions will need to be limited or avoided. In addition this also requires the elimination of native exceptions in the language run-time. Many standards for certification and accreditation require code traceability. The requirement eliminates the use of templates and multiple inheritance. Finally the generated code must consider memory management, any IDL types that do not have memory constrained limits will need to be eliminated or severely constrained.

6.3. IDL Subset

The subset of the IDL language will be based on the ability to map to the identified safe language subsets. Therefore IDL features requiring certain programming language features will be removed from the profile or will need their language mappings changed in order to meet the target language subset. Different programming languages have different mappings for given IDL constructs. For example fixed types map to native types in Ada but in C++ they map to an ORB generated class. In order to achieve interoperability between languages all of the target languages language mappings will need to be analyzed to allow for a common interoperable IDL subset.

The following IDL subset is a work in progress; Octet, Boolean, Char, Enumerated Types, Short, Unsigned Short, Long, Unsigned Long, Long Long, Unsigned Long Long, Float, Double, Array, Structures, *Strings*, *Sequences*, *Unions*, *Any*, *Fixed*. The types in italics are problematic with high assurance systems. It could be possible to constrain the types in italic to achieve high assurance. A particularly troublesome data structure is an Object Reference. According to the CORBA standard an Object Reference has an unknown maximum size. As stated earlier types must have memory constrained limits. If the object reference was eliminated clients could not contact servers!

7. CONCLUSION

Although considerable challenges remain significant progress has been made in defining a High Assurance CORBA standard. It will be possible to complete a CORBA subset suitable for high assurance implementation. This implementation will retain interoperability within the subset and additionally will offer the advantages of CORBA, Increased portability, time to market and location transparency.