Position Paper for W3C/OMG Workshop on Distributed Objects & Mobile Code

Victor L. Voydock
BBN Systems and Technologies
email: voydock@bbn.com
2/28/96

Corbus is a CORBA compliant distributed object system based on the BBN Cronus system. It should be of interest to Workshop participants for the following reasons:

- It is mature, proven technology based on 15 years of distributed object computing research and development. It is used in a number of DoD applications.
- It has more capabilities than commercial CORBA products
- It is a good vehicle for exploring new areas of distributed systems R&D. It is owned by the government. There is access to the source code.

We would be happy to make a Corbus presentation at the Workshop and/or participate in group discussions about issues and experiences in distributed object computing. Corbus has a number of capabilities that are available in few, if any, commercial ORB products. These have proven useful for the development of real-world applications. We look forward to discussing our experiences with implementing and using these capabilities with other practitioners in the field. The capabilities include:

- Robust, proven wide area network communications substrate (dynamic status checking of ops in progress, not timeouts. Years of tuning based on real world experience)
- Support for *clusters* – autonomous administrative domains (allows scaleable object location, controlled sharing of services)
- Cluster-wide Interface & Implementation Repositories
- Replicated objects (*mechanism* transparent to application developer. Developer sets *policy*, trades off availability vs consistency. Can detect out-of-date replicas and inconsistent replicas)
- Remote access to databases (send SQL, get back results. Oracle, Sybase & Informix supported)
- Permanent, globally unique object references (can move objects without invalidating object references. Important when storing object references in persistent storage)
- Multithreaded servers
- Remote monitoring of servers
- Graphical system administration tools
- The ability to make asynchronous method invocations without having to use the DII
- The ability to interactively invoke the methods of a server (so can test server before client code is written)
- Ability to log all Corbus message traffic (useful for debugging complex applications)
- Common LISP language bindings (non-standard)
- Implementations (non-standard, since developed before CORBA existed) of emerging CORBAservices – naming, lifecycle, persistence