CORBA for Telecom Systems
Fact or Fiction?

Douglas C. Schmidt
schmidt@cs.wustl.edu

Washington University, St. Louis
www.cs.wustl.edu/~schmidt/

September 29

Problem: Lack of Real-time Middleware for Telecom

- Many telecom applications require QoS guarantees
  - e.g., call-processing, network management, wireless systems
- Building these applications manually is hard
- Existing middleware doesn’t support QoS effectively
  - e.g., CORBA, DCOM, DCE
- Solutions must be integrated

Candidate Solution: CORBA

Goals of CORBA
- Simplify distribution by automating
  * Object location & activation
  * Parameter marshaling
  * Demultiplexing
  * Error handling
- Provide foundation for higher-level services

Limitations
- Lack of QoS specifications
- Lack of QoS enforcement
- Lack of real-time programming features
- Lack of performance optimizations
Overview of the Joint Real-time CORBA Submission

- **Features**
  - End-to-end priority propagation
  - Protocol properties
  - Thread pools
  - Explicit binding
  - Mutex IDL

---

End-to-End Priority Propagation

- **Features**
  - Priorities can propagate end-to-end
    * Supports heterogeneous RTOS priority mappings
  - Supports priority inheritance
  - Servers can also dictate priority

---

Protocol Properties

```c
interface ProtocolProperties {
    typedef struct {
        IOP::ProfileId protocol_type;
        ProtocolProperties orb_protocol_properties;
        ProtocolProperties transport_protocol_properties;
    } Protocol;

typedef sequence <Protocol> ProtocolList;

interface TCPProtocolProperties
    ProtocolProperties {
        attribute long send_buffer_size;
        attribute long recv_buffer_size;
        attribute boolean keep_alive;
        attribute boolean dont_route;
        attribute boolean no_delay;
    };
```

---

Thread Pools

- **Features**
  - Pre-allocate threads and thread attributes
    * Stacksize
    * Static threads
    * Maximum threads
    * Default priority
  - Applicable at both the ORB and POA level
Explicit Binding

CORBA::Object explicit_bind (in CORBA::Object o, in CORBA::PolicyList policies);

- Features
  - Enables pre-establishment of connections
  - Priority-banded connections
  - Private connections
  - Protocol policies

Mutex IDL

- Features
  - A portable Mutex API
    - e.g., lock, unlock, try_lock
  - Necessary to ensure consistency between ORB and application synchronizers
  - Locality constrained

The ACE ORB (TAO)

- TAO Overview
  - A real-time, high-performance ORB
    - Leverages ACE
      - Runs on POSIX, Win32, RTOSs

- Related work
  - U. RI, Mitre
  - QuO at BBN

Concluding Remarks

- Developers of distributed, real-time telecom applications confront common challenges
  - e.g., service initialization and distribution, error handling, flow control, scheduling, event demultiplexing, concurrency control, persistence, fault tolerance

- Successful developers apply design patterns, frameworks, and components to resolve these challenges

- ORBs are an effective way to achieve reuse of distributed telecom software components

- The next generation of ORBs will provide much better support for real-time QoS
Web URLs for Additional Information

- **More information on TAO**: [www.cs.wustl.edu/~schmidt/TAO.html](http://www.cs.wustl.edu/~schmidt/TAO.html)
- **TAO Event Channel**: [www.cs.wustl.edu/~schmidt/JSAC-98.ps.gz](http://www.cs.wustl.edu/~schmidt/JSAC-98.ps.gz)
- **TAO static scheduling**: [www.cs.wustl.edu/~schmidt/TAO.ps.gz](http://www.cs.wustl.edu/~schmidt/TAO.ps.gz)
- **TAO dynamic scheduling**: [www.cs.wustl.edu/~schmidt/dynamic.ps.gz](http://www.cs.wustl.edu/~schmidt/dynamic.ps.gz)