Next Generation Operating Environment

Bruce Trask
Director of Software Engineering,
SDR Products
CORBA Middleware Misconceptions

- CORBA based Middleware
  - Too big …
  - Too slow …
  - Too heavyweight …
  - Not designed for small scale systems …

- PrismTech Response:
  - It’s possible to run the next generation of CORBA middleware efficiently in even the most resource constrained environment such as a DSP or FPGA

….. and this presentation will demonstrate this!
CORBA Middleware

- **GPP - Enterprise**
  - Large memory capacity
  - Rich system resource support
  - Operating Systems supports POSIX APIs

- **GPP - Embedded**
  - Limited memory capacity
  - Basic system resources available (tasks, semaphores, mutex, condition variables)
  - Operating Systems may support POSIX APIs

- **DSP**
  - Very constrained memory capacity < 2Mb
  - Basic system resources available (tasks, semaphores, locks, interrupts)
  - Usually no POSIX APIs

- **FPGA**
  - Small amount of logic in FPGA
  - Implemented CORBA features to support SCA
  - Low level transport layer
Next Generation CORBA on DSP & FPGA

- DSP platform
  - Significantly reduced footprint ORBs can fit in on-chip memory.
    - C ORB deployment < 100k (TI TMS320V5510)
  - Modular ORB functionality
    - Pick ‘n’ Mix approach to building ORB function
  - Pluggable transport layers allowing smaller and faster transport implementations
    - ETF based transport TCP, UDP, Shared Memory, Rapid I/O, Raw Ethernet
  - Pooling of system resources (threads, memory)
  - System characteristics obeying real-time requirements

- FPGA platform
  - ICO an Integrated Circuit ORB
CORBA on DSP – Key Challenges

- Size
  - Choice of implementation language and CORBA binding
    - Apart from direct assembly, C and C++ most common languages used for DSP programming
    - C and C++ language most commonly supported by compilers
    - 3rd party drivers for devices most commonly available as C libraries
CORBA on DSP – Size

- C over C++?
  - C++ can be and is supported
  - C++ in general produces larger code footprint
  - Some compilers have difficulty with certain aspects of the C++ language.
  - C produces smaller code
  - All language facilities well supported by most C compilers.
  - C language is very portable
CORBA on DSP – Size

- **Objects**
  - C++ implemented using class, attributes and methods
  - C implemented using struct, members and functions taking struct as argument
  - In general compilers produce more code in C++ than for the equivalent OO approximation in C

- **Exceptions**
  - C++ supports exceptions. As such CORBA exceptions usually mapped to native exceptions with often considerable additional size implications
  - C CORBA exception simply implemented as a member in a CORBA::Environment struct passed as an argument to all CORBA function calls
CORBA on DSP – Size

- C++ support libraries and runtime
  - C++ implementations typically make use of generic C++ support libraries (such as STL)
  - C implementation makes use of customised internal functions for the same functionality
CORBA on DSP – Size

Binding Issues

- C++ binding more extensive than C with more support operations defined on core and generated classes
- C++ language binding defines a whole set of additional type support
  - _var, _mgr, _out
- C binding uses simpler type mappings.
  - ‘Any’ a good example.
    - C++: implemented as class with 60 defined methods
    - C: implemented as struct with 3 accessor functions
Transport

- Communication driver implementations can contribute considerably to overall footprint.

- Example some TCP/IP stack implementations can add 100s of KBs to footprint e.g. BF3Net on TI BIOS consumes 135KB, NDK on TI BIOS consumes 288KB.

- Choosing a transport implementation wisely will reduce footprint significantly e.g. Link Handler for OSEck on TI DSP consumes approximately 20KB.
C language and CORBA binding a good choice for DSP applications simply on size alone

- The size difference between the same essential set of functionality can be of the order of $5:1$.

- e*ORB C and C++ on Red Hat 9 Linux compiled with gcc 3.2
  - C libec_poa.so 29 kbytes
  - C++ libe_mpoa.so 105 kbytes
CORBA on DSP – Key Challenges

- **Performance**
  - Typically stub and skeleton optimisations only a small part of the overall performance characteristic
  - Issues such as buffer management, GIOP marshalling and transport efficiency offer more scope for optimisation
  - No real reason why C++ implementation should not be as performant as C. Performance measurements have shown this to be the case
  - Transport choice is fundamental in order to meet throughput and latency targets
CORBA on GPP – Footprint Metrics

- **C++ ORB**
  - **ORB:** OpenFusion e*ORB SDR v1.0 C++ Edition.
  - **RTOS:** VxWorks 5.4, Tornado 2.0, ccpc 2.7.2
  - **Hardware:** MCPN750 – 1342A 367MHz, 64MB RAM
  - Basic Client and Server Used In ORB Footprint Measurements, void void
    - Results taken from OpenFusion® e*ORB SDR v1.0 C++ Edition Footprint Metrics – Version 1.0 Whitepaper

<table>
<thead>
<tr>
<th>Image</th>
<th>.text (Bytes)</th>
<th>.data (Bytes)</th>
<th>.bss (Bytes)</th>
<th>Static Total (Bytes)</th>
<th>Dynamic Heap (Bytes)</th>
<th>Total ORB Overhead (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel</td>
<td>961924</td>
<td>57528</td>
<td>23984</td>
<td>1043436</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kernel + Server</td>
<td>1229388</td>
<td>65296</td>
<td>25088</td>
<td>1319772</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kernel + Client</td>
<td>1161436</td>
<td>63128</td>
<td>24816</td>
<td>1249380</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server Side ORB</td>
<td>267464</td>
<td>7768</td>
<td>1104</td>
<td>276336</td>
<td>49704</td>
<td>326040</td>
</tr>
<tr>
<td>Client Side ORB</td>
<td>199512</td>
<td>5600</td>
<td>832</td>
<td>205944</td>
<td>23064</td>
<td>229008</td>
</tr>
</tbody>
</table>
CORBA on GPP – Performance Metrics

- **C++ ORB**
  - **ORB**: OpenFusion e*ORB SDR v1.0 C++ Edition.
  - **OS**: Red Hat Enterprise Linux Server, WS Release 3 (Taroon Update 3),
  - **Hardware**: Intel Pentium 4 CPU 3.20GHz, 2GBytes RAM, TCP/IP transport.
  - Client and Server roundtrip timings for simple data types

  ![Single Data Type](image)

  - Socket latency = 20.02 micro seconds, therefore ORB overhead represents around only 6% of total round trip time
    - Results taken from OpenFusion® e*ORB SDR v1.0 C++ Edition Footprint Metrics – Version 1.0 Whitepaper
CORBA on GPP – Performance Metrics

- C++ ORB
  - **ORB**: OpenFusion e*ORB SDR v1.0 C++ Edition.
  - **OS**: Red Hat Enterprise Linux Server, WS Release 3 (Taroon Update 3)
  - **Hardware**: Intel Pentium 4 CPU 3.20GHz, 2GBytes RAM, TCP/IP transport.
  - Client and Server roundtrip timings for sequence of octets

Socket latency = 20.02 micro seconds, therefore ORB overhead represents around only 6% overhead of total round trip time
- Results taken from OpenFusion® e*ORB SDR v1.0 C++ Edition Footprint Metrics – Version 1.0 Whitepaper
CORBA on GPP – UDP Transport

- **C++ ORB**
  - **ORB**: OpenFusion e*ORB SDR v1.0 C++ Edition.
  - **OS**: Red Hat Enterprise Linux Server, WS Release 3 (Taroon Update 3)
  - **Hardware**: 2 x Intel Pentium 4 CPU 3.20GHz, 2GBytes RAM, 1GB Ethernet, UDP transport
  - Client and Server data throughput – sequence of Octets (1KB data packet) – e*ORB DIOP implementation requires all data to fit in maximum of 1KB packet

- **Results**
  - Data throughput of 51MB/sec, 400Mb/sec possible at 30% packet loss
CORBA on GPP – Shared Memory Transport

- **C++ ORB**
  - **ORB**: OpenFusion e*ORB SDR v1.0 C++ Edition.
  - **RTOS**: QNX 6.3.0
  - **Hardware**: Intel Pentium 4 CPU 3.20GHz, 2GBytes RAM, POSIX Shared Memory Transport
- Client and Server roundtrip timings for sequence of octets
## C ORB Footprint Metrics on DSP

- **ORB**: OpenFusion e*ORB SDR v1.0 C Edition
- **RTOS**: TI BIOS
- **Hardware**: TI 6416, 1MB on chip memory, TCP/IP Transport
- **Client side ORB footprint measurement – MinimumCORBA “Out-of-the-box” configuration**

<table>
<thead>
<tr>
<th>LIBRARY</th>
<th>CODE</th>
<th>DATA</th>
<th>UDATA</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Client ORB Contribution</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ec_iio.lib</td>
<td>3328</td>
<td>81</td>
<td>48</td>
<td>3457</td>
</tr>
<tr>
<td>ec_orb.lib</td>
<td>46848</td>
<td>2643</td>
<td>509</td>
<td>50000</td>
</tr>
<tr>
<td>ec_os.lib</td>
<td>3328</td>
<td>162</td>
<td>28</td>
<td>3518</td>
</tr>
<tr>
<td>ec_tcp.lib</td>
<td>5248</td>
<td>499</td>
<td>280</td>
<td>6027</td>
</tr>
<tr>
<td><strong>Total Image Size</strong></td>
<td></td>
<td></td>
<td></td>
<td>63002</td>
</tr>
</tbody>
</table>

| **Transport BF3Net** |       |       |       |
| bf3netlan_wi.l64    | 134528 | 948   | 180   |
| **DSPBIOS & Runtime Support** |       |       |       |
| bios.a64            | 16160  | 1450  | 1120  |
| csl6416.lib         | 1696   | 452   | 344   |
| rts6400.lib         | 32384  | 1045  | 2716  |
| rtdx64xx.lib        | 3520   | 84    | 0     |
| **Total Image Size** |       |       |       |
| **Total Image Size** | 259629 |       |       |
C ORB Footprint Metrics on DSP

- **C ORB**
  - **ORB**: OpenFusion e*ORB SDR v1.0 C Edition
  - **RTOS**: TI BIOS
  - **Hardware**: TI 6416, 1MB on chip memory, TCP/IP Transport
  - Server side ORB footprint measurement – MinimumCORBA “Out-of-the-box” configuration

<table>
<thead>
<tr>
<th>LIBRARY</th>
<th>CODE</th>
<th>DATA</th>
<th>UDATA</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server ORB Contribution</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ec_iop.lib</td>
<td>3328</td>
<td>81</td>
<td>48</td>
<td>3457</td>
</tr>
<tr>
<td>ec_orb.lib</td>
<td>53664</td>
<td>2805</td>
<td>526</td>
<td>56995</td>
</tr>
<tr>
<td>ec_os.lib</td>
<td>3328</td>
<td>162</td>
<td>28</td>
<td>3518</td>
</tr>
<tr>
<td>ec_poa.lib</td>
<td>16128</td>
<td>1016</td>
<td>292</td>
<td>17436</td>
</tr>
<tr>
<td>ec_tcp.lib</td>
<td>5248</td>
<td>499</td>
<td>280</td>
<td>6027</td>
</tr>
<tr>
<td><strong>Transport BF3Net</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bf3netlan_wi.l64</td>
<td>134528</td>
<td>948</td>
<td>180</td>
<td>135656</td>
</tr>
<tr>
<td><strong>DSPBIOS &amp; Runtime Support</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bios.a64</td>
<td>16160</td>
<td>1450</td>
<td>1120</td>
<td>18730</td>
</tr>
<tr>
<td>cs16416.lib</td>
<td>1696</td>
<td>452</td>
<td>344</td>
<td>2492</td>
</tr>
<tr>
<td>rtdx64xx.lib</td>
<td>3520</td>
<td>84</td>
<td>0</td>
<td>3604</td>
</tr>
<tr>
<td>rts6400.lib</td>
<td>32704</td>
<td>1061</td>
<td>2724</td>
<td>36489</td>
</tr>
<tr>
<td><strong>Total Image Size</strong></td>
<td></td>
<td></td>
<td></td>
<td>284404</td>
</tr>
</tbody>
</table>

*Note: BF3Ne is a transport protocol used for communication in this setup.*
C++ ORB Footprint Metrics on DSP

- **C++ ORB**
  - **ORB:** OpenFusion e*ORB SDR v1.0 C++ Edition
  - **RTOS:** TI BIOS
  - **Hardware:** TI 6416, 1MB on chip memory, TCP/IP Transport
  - Client side ORB footprint measurement – MinimumCORBA “Out-of-the-box” configuration

<table>
<thead>
<tr>
<th>LIBRARY</th>
<th>CODE</th>
<th>DATA</th>
<th>UDATA</th>
<th>OTHER</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Client ORB Contribution</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_giop.lib</td>
<td>81792</td>
<td>2158</td>
<td>200</td>
<td>5908</td>
<td>90058</td>
</tr>
<tr>
<td>e_iior.lib</td>
<td>704</td>
<td>101</td>
<td>56</td>
<td></td>
<td>861</td>
</tr>
<tr>
<td>e_orb.lib</td>
<td>100352</td>
<td>4412</td>
<td>852</td>
<td>8589</td>
<td>114205</td>
</tr>
<tr>
<td>e_reactor.lib</td>
<td>4160</td>
<td>100</td>
<td>8</td>
<td>492</td>
<td>4760</td>
</tr>
<tr>
<td>ec_os.lib</td>
<td>2720</td>
<td>142</td>
<td>14</td>
<td></td>
<td>2876</td>
</tr>
<tr>
<td>e_tcp.lib</td>
<td>37120</td>
<td>1235</td>
<td>168</td>
<td>2189</td>
<td>40712</td>
</tr>
<tr>
<td><strong>Transport NDK</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hal_eth_mxf.lib</td>
<td>8698</td>
<td>56</td>
<td>384</td>
<td>1536</td>
<td>10584</td>
</tr>
<tr>
<td>hal_ser_stub.lib</td>
<td>64</td>
<td>0</td>
<td>0</td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>hal_timer.lib</td>
<td>416</td>
<td>12</td>
<td>28</td>
<td></td>
<td>456</td>
</tr>
<tr>
<td>hal_userled.lib</td>
<td>32</td>
<td>12</td>
<td>4</td>
<td></td>
<td>48</td>
</tr>
<tr>
<td>netctrl.lib</td>
<td>4832</td>
<td>267</td>
<td>372</td>
<td></td>
<td>5471</td>
</tr>
<tr>
<td>nettool.lib</td>
<td>31808</td>
<td>971</td>
<td>845</td>
<td></td>
<td>33624</td>
</tr>
<tr>
<td>os_sem.lib</td>
<td>13856</td>
<td>878</td>
<td>420</td>
<td>136668</td>
<td>151822</td>
</tr>
<tr>
<td>stk.lib</td>
<td>82848</td>
<td>1854</td>
<td>1429</td>
<td></td>
<td>86131</td>
</tr>
<tr>
<td><strong>Total Image Size</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>288200</td>
</tr>
</tbody>
</table>

- **DSPBIOS & Runtime Support**
  - bios.a64 | 22016 | 1534 | 1120 | | 24670 |
  - csl6416.lib | 6368 | 896 | 692 | | 7956 |
  - lnkrtdx.a64 | 1344 | 92 | 24 | | 1460 |
  - rts6400.lib | 39136 | 1093 | 2733 | | 42962 |
  - rtdx64xx.lib | 3520 | 84 | 0 | | 3604 |
| **Total Image Size** | | | | | 80652 |

- **Total Image Size** | 622324
C++ ORB Footprint Metrics on DSP

### C++ ORB
- **ORB**: OpenFusion e*ORB SDR v1.0 C++ Edition
- **RTOS**: TI BIOS
- **Hardware**: TI 6416, 1MB on chip memory, TCP/IP Transport

**Server side ORB footprint measurement – Minimum CORBA “Out-of-the-box” configuration**

<table>
<thead>
<tr>
<th>LIBRARY</th>
<th>CODE</th>
<th>DATA</th>
<th>UDATA</th>
<th>OTHER</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server ORB Contribution</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e_giop.lib</td>
<td>81792</td>
<td>2158</td>
<td>200</td>
<td>5908</td>
<td>90058</td>
</tr>
<tr>
<td>e_iilop.lib</td>
<td>704</td>
<td>101</td>
<td>56</td>
<td></td>
<td>861</td>
</tr>
<tr>
<td>e_mpoa.lib</td>
<td>83456</td>
<td>2351</td>
<td>356</td>
<td>24206</td>
<td>110369</td>
</tr>
<tr>
<td>e_orb.lib</td>
<td>104064</td>
<td>4640</td>
<td>876</td>
<td>8589</td>
<td>118169</td>
</tr>
<tr>
<td>e_reactor.lib</td>
<td>4160</td>
<td>100</td>
<td>8</td>
<td>492</td>
<td>4760</td>
</tr>
<tr>
<td>ec_os.lib</td>
<td>2976</td>
<td>142</td>
<td>14</td>
<td></td>
<td>3132</td>
</tr>
<tr>
<td>e_tcp.lib</td>
<td>37120</td>
<td>1235</td>
<td>168</td>
<td>2189</td>
<td>40712</td>
</tr>
<tr>
<td><strong>Transport NDK</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>368061</strong></td>
</tr>
<tr>
<td>hal_eth_mxf.lib</td>
<td>8698</td>
<td>56</td>
<td>384</td>
<td>1536</td>
<td>10584</td>
</tr>
<tr>
<td>hal_ser_stub.lib</td>
<td>64</td>
<td>0</td>
<td>0</td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>hal_timer.lib</td>
<td>416</td>
<td>12</td>
<td>28</td>
<td></td>
<td>456</td>
</tr>
<tr>
<td>hal_userled.lib</td>
<td>32</td>
<td>12</td>
<td>4</td>
<td></td>
<td>48</td>
</tr>
<tr>
<td>Netctrl.lib</td>
<td>4832</td>
<td>267</td>
<td>372</td>
<td></td>
<td>5471</td>
</tr>
<tr>
<td>nettool.lib</td>
<td>31808</td>
<td>971</td>
<td>845</td>
<td></td>
<td>33624</td>
</tr>
<tr>
<td>os_sem.lib</td>
<td>13856</td>
<td>878</td>
<td>420</td>
<td>136668</td>
<td>151822</td>
</tr>
<tr>
<td>stk.lib</td>
<td>82848</td>
<td>1854</td>
<td>1429</td>
<td></td>
<td>8631</td>
</tr>
<tr>
<td><strong>DSPBIOS &amp; Runtime Support</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>288200</strong></td>
</tr>
<tr>
<td>bios.a64</td>
<td>22016</td>
<td>1534</td>
<td>1120</td>
<td></td>
<td>24670</td>
</tr>
<tr>
<td>csl6416.lib</td>
<td>6368</td>
<td>896</td>
<td>692</td>
<td></td>
<td>7956</td>
</tr>
<tr>
<td>lnkrtdx.a64</td>
<td>1344</td>
<td>92</td>
<td>24</td>
<td></td>
<td>1460</td>
</tr>
<tr>
<td>rts6400.lib</td>
<td>39136</td>
<td>1093</td>
<td>2733</td>
<td></td>
<td>42962</td>
</tr>
<tr>
<td>rtdx64xx.lib</td>
<td>3520</td>
<td>84</td>
<td>0</td>
<td></td>
<td>3604</td>
</tr>
<tr>
<td><strong>Total Image Size</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>736913</strong></td>
</tr>
</tbody>
</table>
CORBA on DSP – Performance Metrics

- C ORB
  - ORB: OpenFusion e*ORB SDR v1.0 C Edition.
  - OS: Client side: Red Hat Enterprise Linux Server, WS Release 3 (Taron Update 3), Server Side: TI BIOS
  - Hardware: Client Side: Intel Pentium 4 CPU 3.20GHz, 2GBytes RAM, TCP/IP transport, Server Side: TI 64161MB on chip memory, NDK TCP/IP Transport
  - Client and Server roundtrip timings for sequence of octets

Socket latency = 138 micro seconds, therefore ORB overhead represents around only 8% overhead of total round trip time
Shared Memory Performance Metrics (In Parameter)

- **ORB**: OpenFusion e*ORB SDR v1.1 C++ Edition.
- **RTOS**: QNX 6.3.0
- **Hardware**: x86, Pentium P4, 3.2 Ghz

![Graph showing time in microseconds against octet sequence for TCP/IP and PXSHMIOP]
Shared Memory Performance Metrics (Out Parameter)

- **ORB:** OpenFusion e*ORB SDR v1.1 C++ Edition.
- **RTOS:** QNX 6.3.0
- **Hardware:** x86, Pentium P4, 3.2 Ghz

![Graph showing performance metrics for different octet sequences and time in micro seconds]
Shared Memory Performance Metrics (Inout Parameter)

- **ORB**: OpenFusion e*ORB SDR v1.1 C++ Edition.
- **RTOS**: QNX 6.3.0
- **Hardware**: x86, Pentium P4, 3.2 Ghz
Shared Memory Performance Metrics (Return)

- **ORB**: OpenFusion e*ORB SDR v1.1 C++ Edition.
- **RTOS**: QNX 6.3.0
- **Hardware**: x86, Pentium P4, 3.2 Ghz

![Graph showing performance metrics]

- **TCP/IP**
- **PXSHMIOP**
Integrated Circuit ORB

- SCA compliance is maintained to the FPGA
- Transport overhead is reduced
  - 100x faster than S/W proxy implementation
Why use an ICO?

- Software developers generally limited CORBA’s use, in most cases, to implementations executing on General Purpose Processors (GPPs).
- CORBA benefits becoming important as system complexity drives the use of DSPs and FPGAs.
- Pros and Cons of running CORBA in a FPGA embedded processor versus direct use of VHDL language bindings.
  - as portability, reuse, and elimination of software proxies).
Why use an ICO?

- Use of an embedded processor in an FPGA for the sole purpose of supporting middleware is impractical
  - FPGAs that support embedded processors are at the high end of the price range
  - Running an ORB in an FPGA embedded processor uses significant memory resources
    - Wasting valuable FPGA resources on CORBA functionality
  - Choice of industry standard embedded processors is limited
    - Most are custom processors locked to particular FPGA vendor
  - Embedded processor performance is typically poor
    - Most embedded processors are clocked at significantly lower speeds than commercially available GPPs and DSPs
Why use ICO?

- ICO uses no block memory and currently uses fewer than 3000 logic cells and is portable between FPGA families.
- ICO improves portability and re-use in SCA-based applications since the coupling between software on GPP/DSPs and FPGAs is eliminated.
  - Coupling via Memory Mapped I/O, for example.
- Eliminates the need for proxies/ adapters.
  - Reduces overhead, latency.
  - Increases throughput.
- Eliminates the need for complex hardware abstraction layer protocols (Supports direct access to SCA components running on H/W).
  - Spectra tools auto-generate VHDL delivering H/W SCA “components.”
- Applications in security-related areas where the assurance of large software applications (such as ORBs) is suspect.
The latency through the ICO from the time a GIOP message arrives at its inputs until the response message is ready at its output is about .200ns.
SCA Operating Environment (OE)

- **GPP - Enterprise**
  - EORB C++
  - TAO C++
  - JacORB Java
  - RtOrb Ada

- **GPP - Embedded**
  - EORB C++
  - EORB C
  - TAO
  - RtOrb Ada

- **DSP**
  - EORB C++
  - EORB C

- **FPGA**
  - ICO
SCA Operating Environment (OE)

- Implementation of SCA 2.2 specification
  - CORBA components using e*ORB C Edition
    - Common platforms: Linux, Integrity, VxWorks, TI DSP/BIOS, OSE
  - Small footprint
    - Approximately 1Mb
  - Portable
    - Abstraction layers for system specific APIs (non POSIX)
- Core framework elements written in C
  - CF::DomainManager
  - CF::DeviceManager
  - CF::ExecutableDevice
  - CF::FileSystem
  - CF::File
  - CF::ApplicationFactory
SCA Operating Environment (OE)

- Spectra Tools generates
  - CF::Resource
  - CF::ResourceFactory
  - XML domain profiles
- Spectra Tools generators
  - C++ component generation supported
  - C component generation in development
- Lightweight Services
  - Name
  - Log
  - Event
XML Parser - Size

- XML Parser contributing ~ 400K to overall image size
- More future optimisation of XML library planned
Core Framework – Size

- Total Library sizes for Core Framework
  - libcf_impl.so
  - libcf_logservice_c.so
  - libcf_logservice_s.so
  - libcf_xml.so
Benefits of using CORBA in SDR

- CORBA is small and fast enough to efficiently support communication across the whole signal processing chain, including FPGA and DSP environments.
- Efficient CORBA implementations like e*ORB impose little overhead on top of the underlying performance of the transport.
- Choice of transports in ORB is critical to meeting performance criteria.
  - ETF allows for custom transports to be easily supported.
  - ETF allows for multiple transports to be configured in and used in the same system.
- The language neutrality of CORBA allows OE to be written in C (very low footprint) but still support waveforms written in other languages such as C++ and ADA.
Benefits of using CORBA in SDR

- If CORBA is not used …
  - You’re on the road to a poor man’s CORBA
    - Still have to solve the same issues in a proprietary way
      - Transports
      - Message formats
      - Marshalling/Unmarshalling of types
      - Call dispatch
  
- The benefits of CORBA are substantial
  - Facilitates implementation of portable waveforms. A key goal of the JTRS program
  - The use of standards based middleware like CORBA and SCA enables greater tool integration, supporting faster development through MDD and generative programming techniques