

# Comparing C++ and ERLANG for Motorola Telecoms Software

Phil Trinder  
Heriot-Watt University  
P.W.Trinder@hw.ac.uk

## Abstract

There is considerable folklore suggesting that ERLANG aids the rapid production of robust distributed systems, but only a few rather general studies published. This talk reports the first systematic comparative evaluation of ERLANG in the context of substantial commercial products.

Our research strategy is to re-engineer two C++/CORBA telecoms applications in ERLANG and make comparative measurements of both implementations. The first component is a medium-scale (15K line) Dispatch Call Controller (DCC), and the second a smaller (3K line) Data Mobility (DM) component that is closely integrated with five other components of a radio communications subsystem (RCS). To investigate interoperation costs we have constructed two DMs: a pure ERLANG implementation and an ERLANG/C implementation that reuses some C DM libraries.

We investigate the following six research questions, first considering the potential benefits of a high-level distributed language technology like ERLANG.

### Q1 *Can robust, configurable systems be more readily developed?*

Yes, as detailed below.

*Resilience* The ERLANG DM and DCC both sustain throughput at extreme loads and automatically recover to pre-overload throughput when load drops.

*Availability* The ERLANG DCC remains available despite repeated hardware and software failures, and performance doesn't degrade with repeated failures. The ERLANG DCC resists the simultaneous failure of multiple components and its throughput is not significantly reduced when a small percentage of messages crash the service instance.

*Dynamic Reconfigurability* The ERLANG DCC shows near-linear throughput scaling as resources are added and a near-linear decrease in throughput as resources are removed. Moreover the cost of adding or removing a processor is small.

### Q2 *Can productivity and maintainability be improved?*

Yes, using source lines of code (SLOC) as a metric, both the ERLANG DM and DCC are less than a third of the size of the C++ counterpart. Moreover much of the ERLANG DCC is a reusable generic server. The reasons for the reduced programming effort are that coding for the successful case saves 27%,

high-level communications save 23%, and automatic memory management saves a further 11%. These productivity results are consistent with other studies, and with developer folklore.

Secondly, we consider the feasibility of the ERLANG high-level distributed language technology for realistic telecoms software development.

### Q3 *Can the required distributed functionality be specified?*

Yes, even although low-level distributed coordination aspects are abrogated to the ERLANG implementation, the requisite DCC and DM functionality is readily specified.

### Q4 *Can acceptable performance be achieved?*

Substantially Yes, the ERLANG DMs have acceptable time performance, exceeding the throughput requirements. Indeed the round trip times for the pure ERLANG DM are a third of the C++ DM times, and even the ERLANG/C round trip times are no more than 50% greater. However, the ERLANG memory residency is up to 170% greater due to the (fixed size) 5Mb runtime system.

### Q5 *What are the costs of interoperating with conventional technology?*

Combining the ERLANG DMs with the C RCS test harness, and incorporating the C drivers in the ERLANG/C DM shows that ERLANG components can interoperate with legacy code. As ERLANG components are readily made robust, the robustness of a large distributed system can be incrementally improved by (re)engineering critical components in ERLANG. The additional space cost of the interoperating ERLANG/C DM is small: 15%. However the time penalty for the additional communication with the C driver is high: and the ERLANG/C DM roundtrip times are 4 times slower and maximum throughput is a quarter of the pure ERLANG DM.

### Q6 *Is the technology practical?*

As far as required by the DCC and DM, ERLANG has proved to be a usable technology. We have shown that ERLANG is available on two hardware/operating system platforms, and Sun/Solaris is the RCS product platform. The technology is well supported with training and consultancy, and many useful components are available in the OTP libraries.

We conclude that high-level distributed languages like ERLANG can deliver the required telecoms functionality and performance. Moreover, such languages offer improved robustness and productivity for distributed telecoms software. The RCS product group have responded very favourably to the robustness and productivity benefits of the ERLANG DMs. An ERLANG DM has been installed alongside the development C++ DM, and the product group are considering reengineering other parts of the RCS in ERLANG.