Evaluating Distributed Functional Languages for Telecommunications Software

J.H. Nyström  P.W. Trinder  D.J. King
Heriot-Watt University  Motorola

January 23, 2002
Overview

• Goals
• Motivation
• Project plan
• Telecoms software characteristics
• High-level language properties
• First test implementation
• What’s next
PROJECT GOALS

Commercial
Convince Motorola that ERLANG constitutes a viable technology for implementing distributed telecoms software.

Scientific
Investigate the impact of language constructs on distributed software construction.
Q: Has ERLANG not already been shown to be viable telecoms technology by numerous successful applications?

A1: No, it has been shown that companies with deep understanding and experience of building such systems and can successfully use ERLANG.

A2: Yes, but what are the properties of ERLANG that makes it so suitable for implementing distributed applications?
PROJECT PLAN: COMMERCIAL

- Construct three applications in ERLANG of increasing size and realism, i.e., closer to a real product.
- Show that the applications have the essential properties required of a telecoms software with additional benefits.
Evaluating Distributed Functional Languages for Telecommunications Software
Telecoms software characteristics

- Distributed
- Reliable
- Highly available
- Rapid production

Evaluating Distributed Functional Languages for Telecommunications Software
Telecoms software challenges

- Fault-tolerant
- Scalable
- Resilient
- Dynamically adaptable
Erlang Workshop, August 29, 2003

**High-level language aspects**

- Dynamic typing vs. static typing
- Strict vs. non-strict evaluation
- Explicit vs. implicit messaging and distribution

Evaluating Distributed Functional Languages for Telecommunications Software
Dispatch Call Controller (DCC)

The server part of dispatch control within a mobile telephone network.

Crucial functionality:

- Dynamic scalability
- Resource reclamation
- Fault tolerance
- Soft real time performance
First test implementation

Evaluating Distributed Functional Languages for Telecommunications Software
What doesn’t the system handle

- Single point of failure.
- No dynamic code upgrade.
- Simplistic service.
What does the system handle

- Fully fault-tolerant in the service nodes.
- Scales.
- Resilient.
- Service nodes can be added or removed dynamically.
 Evaluating Distributed Functional Languages for Telecommunications Software
Evaluating Distributed Functional Languages for Telecommunications Software
What now?

- Remove single point of failure.
- Hot-code upgrade.
- Realistic service.
- Measure performance impact of
  - dynamic reconfiguration
  - fault-tolerance
  - hot-code upgrade
WHAT’S NEXT?

Motorola Interaction

- Determine the next application.
- Disseminate results

Scientific

- Reimplement DCC in GdH.
- Comparison with C++/CORBA and JAVA/RMI implementations.
New project

Commercial and Scientific objectives

First results promising

Challenges: Find the right application and the right people