

Erlang

Overview and history



Ericsson Language

Agner Krarup Erlang

What is Erlang?

- Language developed at Ericsson Telecom
- Core language is a simple **dynamically-typed functional programming language**
 - Concurrent, with **light-weight processes**
 - “Share nothing” process semantics
 - Pure **asynchronous message passing**
 - Transparent distribution of processes across machines
 - Mechanisms for in-service code upgrade
 - Large set of libraries (OTP)

History of Erlang

- (mid-80s) Ericsson (Swedish telecom company) set out to find the best language for building the next generation of telecom systems
- Requirements
 - Handling a very large number of concurrent activities
 - Systems distributed over several computers
 - Actions to be performed at a certain point of time or within certain time
 - Very large software systems
 - Complex functionality such as feature interaction
 - Continuous operation over several years
 - Software maintenance (reconfiguration, etc.) without stopping the system
 - Stringent quality and reliability requirements
 - Fault tolerance both to hardware failures and software errors

History of Erlang cont'd

- Initially Smalltalk was considered. Method calls in Smalltalk are “message sends.”
Message passing figures prominently in Erlang's design.
- Soon after, they discovered that the rules for how a telecom system should work could be elegantly expressed in the logic language **Prolog**. *The first Erlang interpreters were written in Prolog and the syntax retains the flavor of Prolog.*

History of Erlang cont'd

- Prolog is a **logic language** - a program consists of facts and queries over those facts are evaluated

```
mother_child(trude, sally).
```

```
father_child(tom, sally).
```

```
father_child(tom, erica).
```

```
father_child(mike, tom).
```

```
sibling(X, Y)      :- parent_child(Z, X),  
                      parent_child(Z, Y).
```

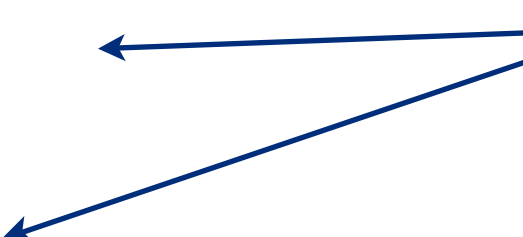
```
parent_child(X, Y) :- father_child(X, Y).
```

```
parent_child(X, Y) :- mother_child(X, Y).
```

```
?- sibling(sally, erica).
```

Yes

Facts



Query



History of Erlang cont'd

- Prolog is a **logic language** - a program consists of facts and queries over those facts are evaluated

- It turns out that the logic features of Prolog were deemed unnecessary for telecom systems so they were removed from Erlang (leaving a functional language)

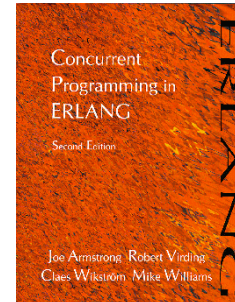
Query

```
parent_child(X, Y) :- father_child(X, Y).  
parent_child(X, Y) :- mother_child(X, Y).  
  
?- sibling(sally, erica).  
Yes
```



History of Erlang cont'd

- Other modifications from Prolog were the addition of concurrency and message passing. The syntax diverged over time.
- Erlang continued to evolve “organically”, growing from 2 to “hundreds” of developers between 1989 and 1997.
- First Erlang book published in 1993, Ericsson attempted to commercialize Erlang.
- Banned within some groups at Ericsson in 1998, released as open source.



Real-World Erlang

- **Whatsapp:**
 - <http://www.dailymail.co.uk/sciencetech/article-2595271/Pride-fall-WhatsApp-boasts-record-breaking-message-numbers-24-hours-service-goes-down.html>
 - 450M users, handles 64B messages per 24 hours
- CouchDB - document store “NoSQL” database
- Facebook Chat
- RabbitMQ - owned by VMWare, message queue server
- ejabberd - XMPP (jabber chat) server
- Amazon SimpleDB
- GitHub
- Wings3d - open source 3d modeling tool
- ... and more: <https://erlangcentral.org/erlang-projects/>

Why Erlang?

- Functional (like OCaml, Scheme, Haskell)
 - Program by evaluating functions to produce results, not by mutation of program state
 - A good match for concurrent programming since concurrent access to shared state (and the required mutual exclusion) are the source of many bugs (races, atomicity violations, deadlocks)

Why Erlang? cont'd

- Transparent distribution of processes across machines
- Efficient implementation of processes, message passing
- Robust error handling support
- Runtime updates

Installing Erlang

- Available on linuxlab.cs.umd.edu (R14B)
 - Newest version is R17
- Windows: <http://erlang.org/download.html>
- Mac: need to build from source
- Web REPL: <http://www.tryerlang.org/>

Erlang Resources

- Getting Started with Erlang (http://www.erlang.org/doc/getting_started/users_guide.html)
- Erlang Documentation (<http://www.erlang.org/doc/>)
- “learn you some Erlang for a great good” (<http://learnyousomeerlang.com>)
- Erlang (CACM) (<http://cacm.acm.org/magazines/2010/9/98014-erlang/fulltext>)
- Erlang for Concurrent Programming (ACM Queue) (<http://queue.acm.org/detail.cfm?id=1454463>)
- Books:
 - Programming Erlang: Software for a Concurrent World (Armstrong)
 - Erlang Programming (Cesarino, Thompson)

