

Scala Enthousiasts

The Actor model and Akka

Christophe Pache

Crossing-Tech
EPFL

February 16, 2012



- Some theory about the Actor model and Akka
- Use case : Nao by Nicolas
- Examples with Nao by Nicolas



This is an introduction by dummies for enthusiasts
Philipp Haller, from TypeSafe, is interested in giving a presentation
more advanced features
(Future and Promise)



Actor model

The Actor model provides efficient concurrent computation where every object is an actor communicating with each other with messages



An actor consists of:

- an independant process
- a mailbox to communicate

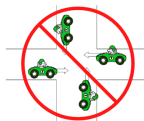


This enables synchronous and asynchronous communication



Actor model advantages

- High level concurrent computation
- Sequential execution, one message at a time
- Simplified distribution of processes
- Avoid deadlocks through timeout



Akka provides more than only Actors, among the most important:



▸ registry



▸ routing



▸ remote



▸ supervision



▸ STM

Actor model and Akka

Akka provides more than only Actors, among the most important:



▸ registry



▸ routing



▸ remote



▸ supervision



▸ STM

And also Plain Old SCala Object integration in ▸ Typed Actors

Registry

- store every created Actor
- lookup using ids or types



uuid	id	Actor



► Akka features

- Routing actors
- Actor pool size fixed or dynamic



► Akka features

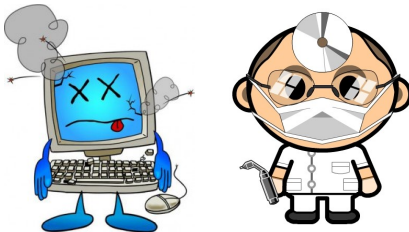
- Server managed
- Client request accessing registered actors
- Transparent



► Akka features

Supervision

- Let it crash
- Life cycle management : Permanent or not
- Restart hook



► Akka features

Software Transactional Memory

- Shared references
- Transactions
- Coordination



► Akka features

POSCO integration

- Class and Interface
- Factory for creation
- Transparent



► Akka features

Akka provides high-level concurrent computation with
(among most important):

- POSCO integration
- load balancing
- supervising



And now....

- Everyone is now Akka enthousiast?
- Practice and you will

