

# A Choreographic Formal Model of Communicating Systems

—Immigration on Formal Methods—

Emilio Tuosto @ GSSI

# So far...

- An idea of FMs

## Leonardo da Vinci

“ Ma prima farò alcuna esperienza avanti ch'io più oltre proceda, perché mia intenzione è allegare prima l'esperienza e poi colla ragione dimostrare. ”

## eM's (bad) translation

“ Before proceeding further, I will first get some experiment, because my intention is to first understand the experiment and then to explain it with the intellect. ”

- Concurrency vs Parallelism
- Shared-memory

# Message-passing

Pink Floyd

“Is there anybody out there?”

# A glimpse of Erlang

```
ping(N, Pong_PID) ->
  Pong_PID ! {ping, self()},
  receive
    pong ->
      io:format("Ping received pong~n", [])
  end,
  ping(N - 1, Pong_PID).
```

```
ping(0, Pong_PID) ->
  Pong_PID ! finished,
  io:format("ping finished~n", []);
```

```
pong() ->
  receive
    finished ->
      io:format("Pong finished~n", []);
    {ping, Ping_PID} ->
      io:format("Pong received ping~n", []),
      Ping_PID ! pong,
      pong()
  end.
```

## Semantics

- Message passing
- FIFO buffers **[[mailboxes in Erlang's jargon]]**
- Spawn of threads

# A glimpse of Erlang

```
ping(N, Pong_PID) ->
  Pong_PID ! {ping, self()},
  receive
    pong ->
      io:format("Ping received pong~n", [])
  end,
  ping(N - 1, Pong_PID).
```

```
ping(0, Pong_PID) ->
  Pong_PID ! finished,
  io:format("ping finished~n", []);
```

```
pong() ->
  receive
    finished ->
      io:format("Pong finished~n", []);
    {ping, Ping_PID} ->
      io:format("Pong received ping~n", []),
      Ping_PID ! pong,
      pong()
  end.
```

## Semantics

- Message passing
- FIFO buffers **[[mailboxes in Erlang's jargon]]**
- Spawn of threads

## Asynchrony by design

Erlang is an embodiment of the well-known **actor model** of Hewitt and Agha...dates back to '73!

# Friendlier representations

## Local behaviour: communicating machines



CFSMs (Brand & Zafiropulo 1983!): FIFO buffers as well

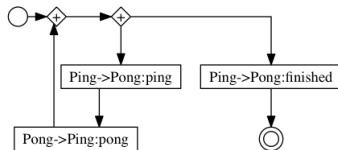
# Friendlier representations

## Local behaviour: communicating machines



CFSMs (Brand & Zafiropulo 1983!): FIFO buffers as well

## Choreography: global graph



... “synchronous” distributed workflow (Deniélou and Yoshida 2012)

# A glimpse of Erlang

```
ping(N, Pong_PID) ->
  Pong_PID ! {ping, self()},
  receive
    pong ->
      io:format("Ping received pong~n", [])
  end,
  ping(N - 1, Pong_PID).
```

```
ping(0, Pong_PID) ->
  Pong_PID ! finished,
  io:format("ping finished~n", []);
```

```
pong() ->
  receive
    finished ->
      io:format("Pong finished~n", []);
    {ping, Ping_PID} ->
      io:format("Pong received ping~n", []),
      Ping_PID ! pong,
      pong()
  end.
```



# A glimpse of Erlang

```
ping(N, Pong_PID) ->
  Pong_PID ! {ping, self()},
  receive
    pong ->
      io:format("Ping received pong~n", [])
  end,
  ping(N - 1, Pong_PID).
```

```
ping(0, Pong_PID) ->
  Pong_PID ! finished,
  io:format("ping finished~n", []);
```

```
pong() ->
  receive
    finished ->
      io:format("Pong finished~n", []);
    {ping, Ping_PID} ->
      io:format("Pong received ping~n", []),
      Ping_PID ! pong,
      pong()
  end.
```

Q:

Is this program correct?

# A glimpse of Erlang

```
ping(N, Pong_PID) ->
  Pong_PID ! {ping, self()},
  receive
    pong ->
      io:format("Ping received pong~n", [])
  end,
  ping(N - 1, Pong_PID).
```

```
ping(0, Pong_PID) ->
  Pong_PID ! finished,
  io:format("ping finished~n", []);
```

```
pong() ->
  receive
    finished ->
      io:format("Pong finished~n", []);
  {ping, Ping_PID} ->
      io:format("Pong received ping~n", []),
      Ping_PID ! pong,
      pong()
  end.
```

Q:

Is this program correct?

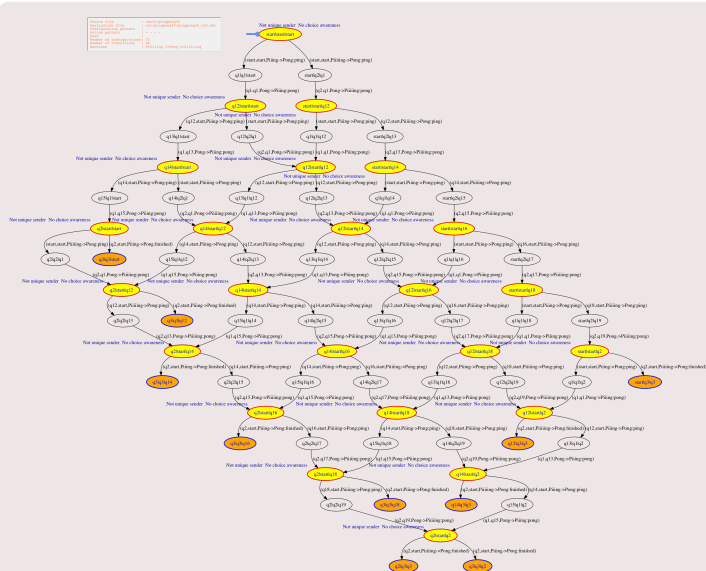
A:




No!

Exercise:

find the bug

Send ping-pong to shell !!! ... I mean, use ChoSyn



-  Brand, D. and Zafiropulo, P. (1983).  
On Communicating Finite-State Machines.  
*JACM*, 30(2):323–342.
-  Guanciale, R. and Tuosto, E. (2016).  
An abstract semantics of the global view of choreographies.  
In *Proceedings 9th Interaction and Concurrency Experience, ICE 2016, Heraklion, Greece, 8-9 June 2016.*, pages 67–82.
-  Tuosto, E. and Guanciale, R. (2018).  
Semantics of global view of choreographies.  
*Journal of Logic and Algebraic Methods in Programming*, 95:17–40.  
Revised and extended version of [Guanciale and Tuosto, 2016].