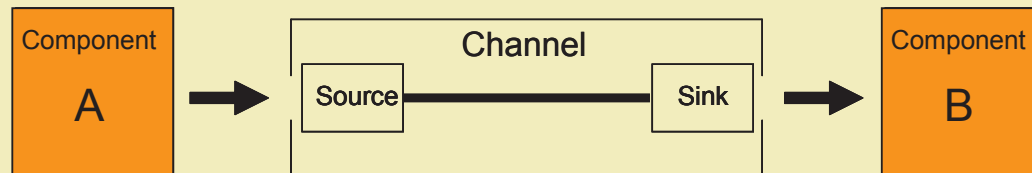


The MoCha Framework

Juan Guillen-Scholten

Centre for Mathematics and Computer Science (CWI)
Kruislaan 413, 1098 SJ Amsterdam, Netherlands

Mobile Channels



A channel consists of two distinct ends: Usually (source, sink), but also (source, source) and (sink, sink). Components write to the source-end, and read from the sink-end. The communication is anonymous, the components do not know each other. Channels can be synchronous or asynchronous (FIFO, set, bag, etc). Channel-ends are mobile.

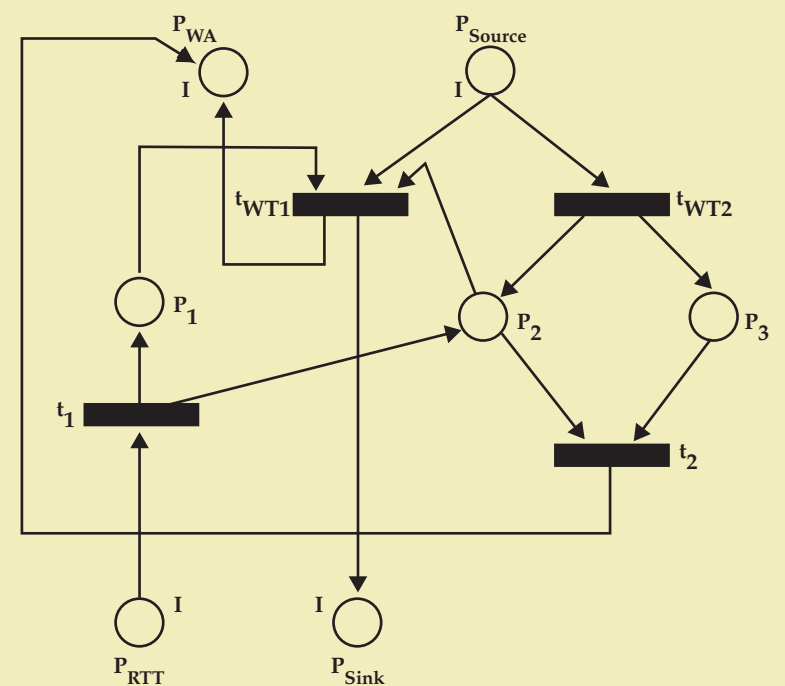
MoCha-pi

An exogenous coordination **calculus** that is based on mobile channels. Our calculus is an extension of the well-known pi-calculus. The novelty of MoCha-pi, is that its channels are a special kind of process that allow other processes to communicate with each other and impose exogenous coordination through user defined channel types. Also new, is the fact that in our calculus channels are viewed as resources.

$$K(l, r) \stackrel{def}{=} \mathcal{R}^l \mid \mathcal{R}^r \mid K'(l, r)$$

$$K'(l, r) \stackrel{def}{=} l(x).r(\lambda).(\bar{l}\langle\lambda\rangle \mid \bar{r}\langle x\rangle).K'(l, r)$$

A Synchronous Channel Type in MoCha-pi

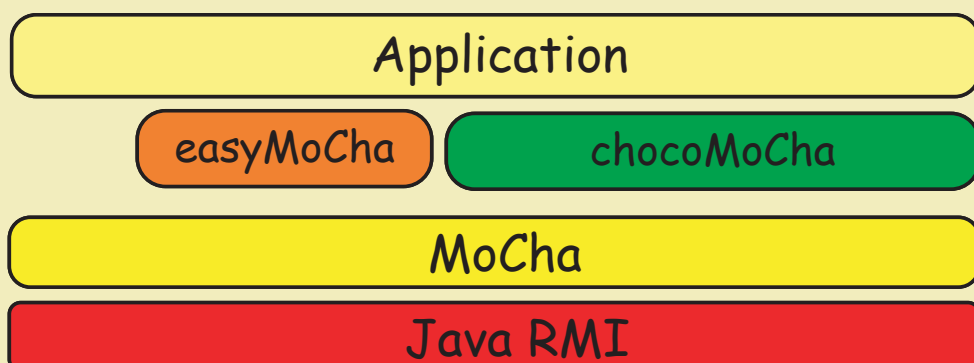


A P/N Lossy Synchronous Channel.

Modeling with Petri Nets

We use **Petri Nets** for modeling the communication and coordination aspects of systems/applications that use our mobile channels.

The MoCha Middleware



MoCha is a **middleware** with the aim of providing applications with directed, anonymous, **distributed** and **mobile** communication **channels**. It is implemented on top of the Java Remote Method Invocation (**RMI**) package.

Juan Guillen-Scholten
Joint work with:
Farhad Arbab (CWI, LIACS)
Marcello Bonsangue (LIACS)
Frank de Boer (CWI)

Questions? Comments? Suggestions?: juan@cwi.nl