Goals

Abstract Goals

* provide an evironment for executable specifications of distributed software systems

* connect different specification- and implementation-techniques

* let the modeler chose the appropriate technique

Concrete Goals

* distributed Petri Net simulation using Design/CPN

* calling Java objects from Design/CPN

* calling Design/CPN nets from Java

Expected Applications / Benefits

* Java libraries can be reused

* easy network- and GUI-programming delegated to Java

* CPNs can be utilized for embedded distributed algorithms

* connecting the worlds of Petri Nets and Object-Orientation

Technical Features

* start any number of simulators on distributed machines

* move arbitrary tokens between different instances of the simulator anywhere in the net / Internet

* create arbitrary Java objects and invoke any Java method from a Design/CPN simulation run

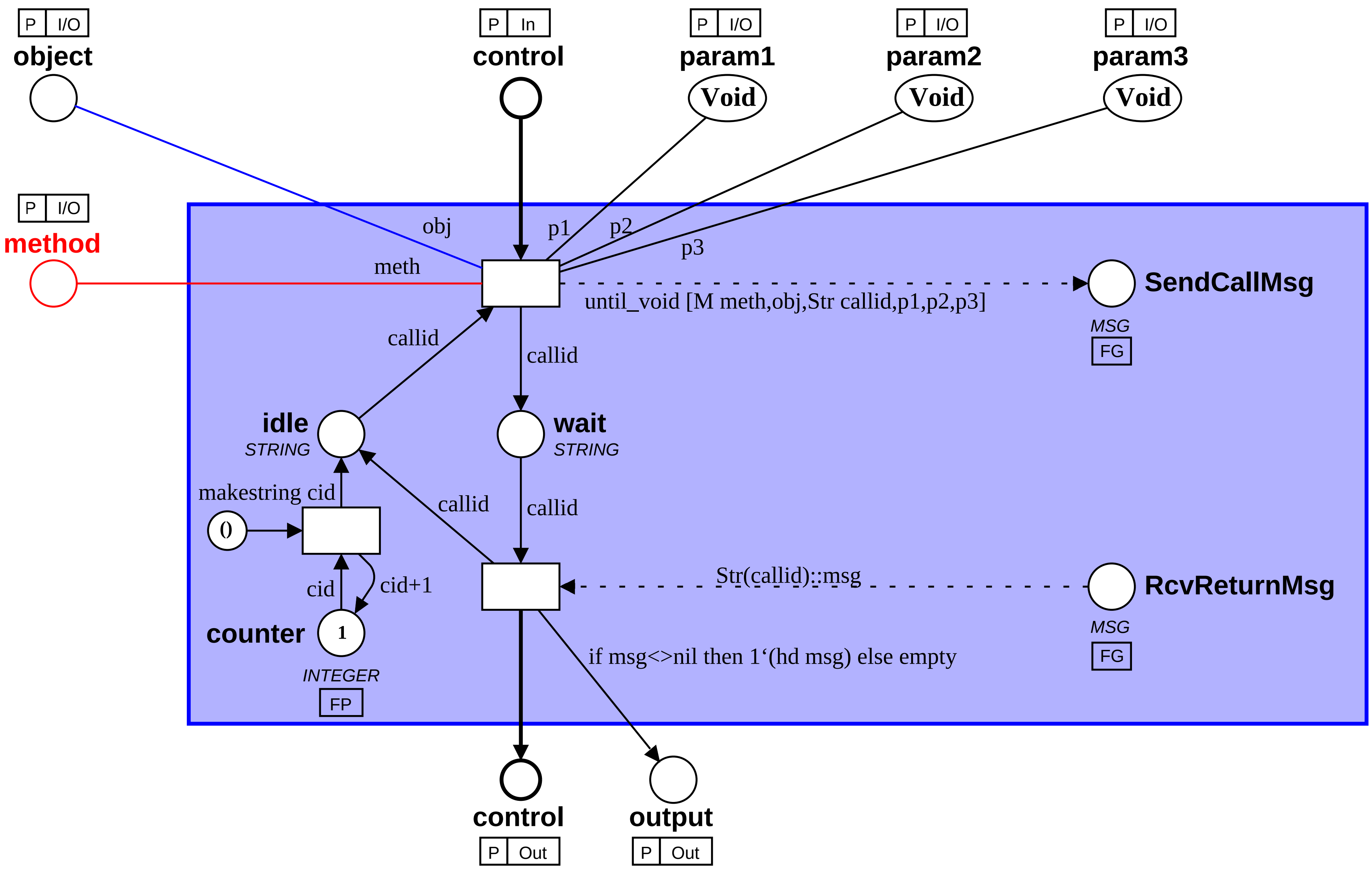
* provide CPNs that can be called from Java

* (almost) no distinction between remote CPN- and Java-objects

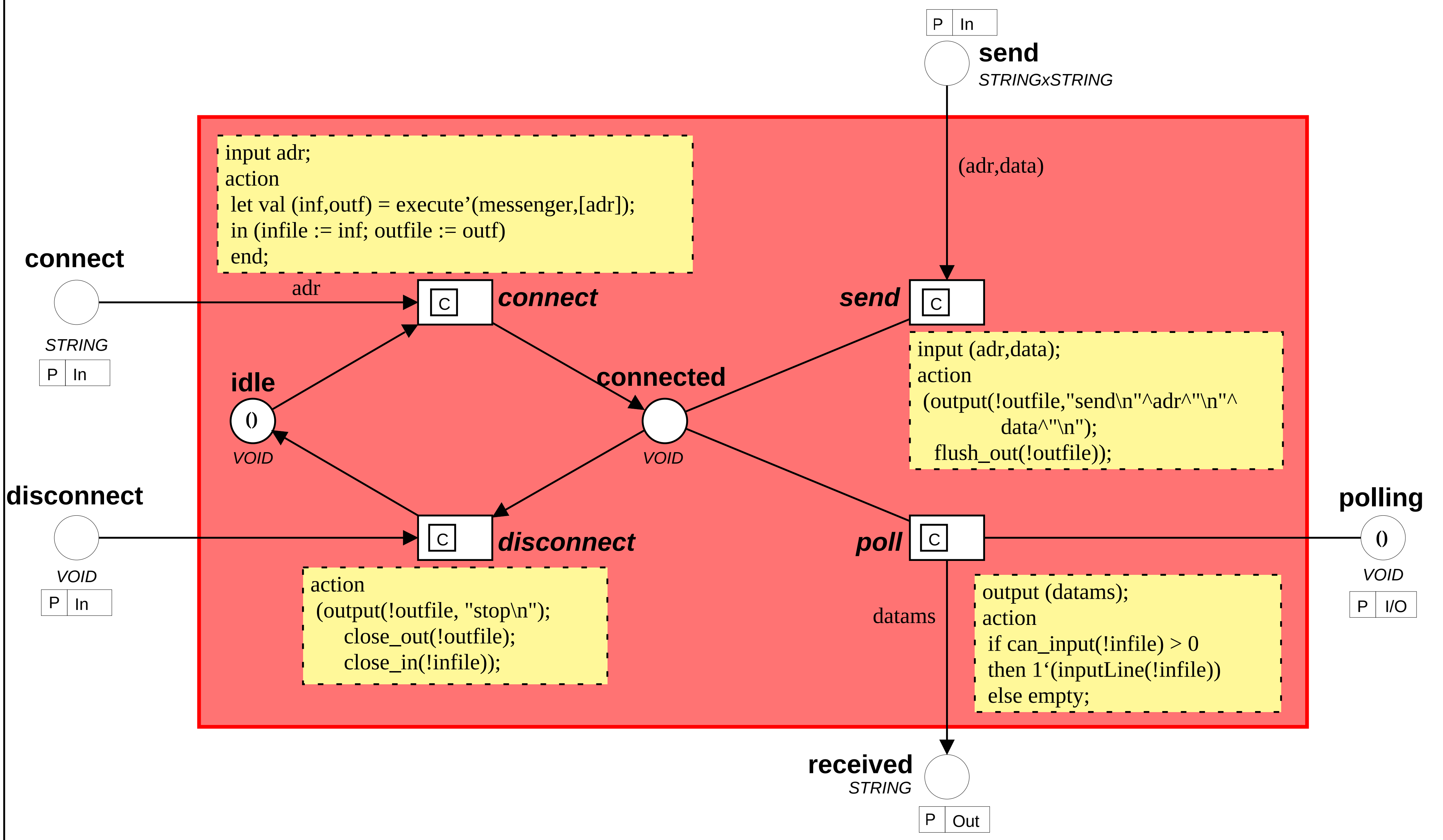
* easy to integrate into already existing Design/CPN models

* drawback: performance (polling)

The Call Page



The I/O Page



P In
send
STRINGxSTRING
(adr,data)
input adr;
action
let val (inf,outf) = execute'(messenger,[adr]);
in (infile := inf; outfile := outf)
end;
connect
adr
C
connect
send
C
STRING
P In
idle
()
connected
input (adr,data);
action
(output(!outfile,"send\n"^adr^"\n"^
data^"\n");
flush_out(!outfile));
VOID
VOID
disconnect
polling
C
disconnect
poll
C
()
VOID
P In
VOID
P I/O
action
(output(!outfile, "stop\n");
close_out(!outfile);
close_in(!infile));
datams
output (datams);
action
if can_input(!infile) > 0
then 1`(inputLine(!infile))
else empty;
received
STRING
P Out

A Framework for Interacting Design/CPN and Java Processes

Olaf Kummer, Daniel Moldt, Frank Wienberg
University of Hamburg, Germany

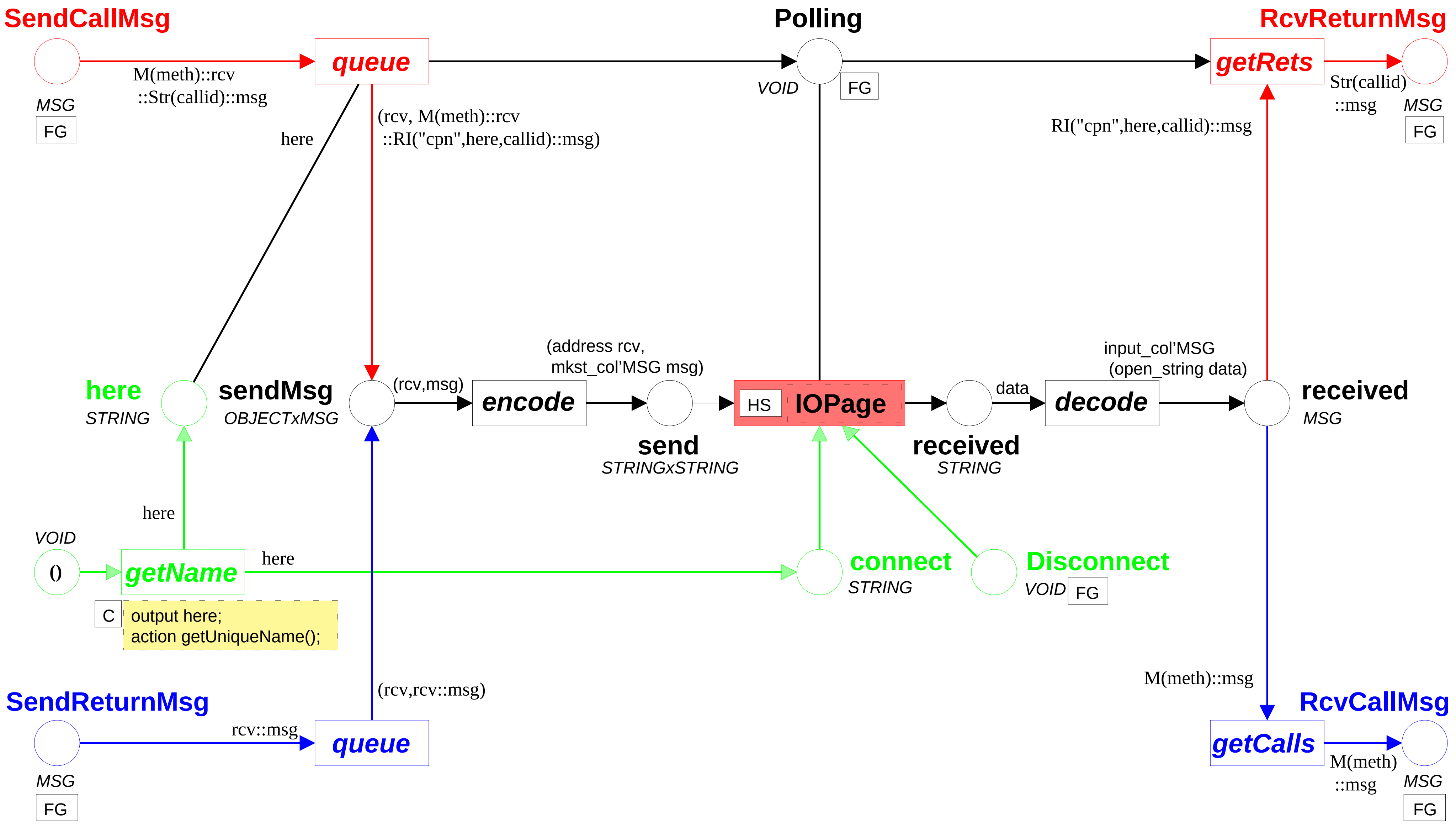
Research Topics of our Group

* techniques: Coloured Petri Nets, object-oriented Petri Nets, cyclic orders, rewriting, (linear) logic, AI

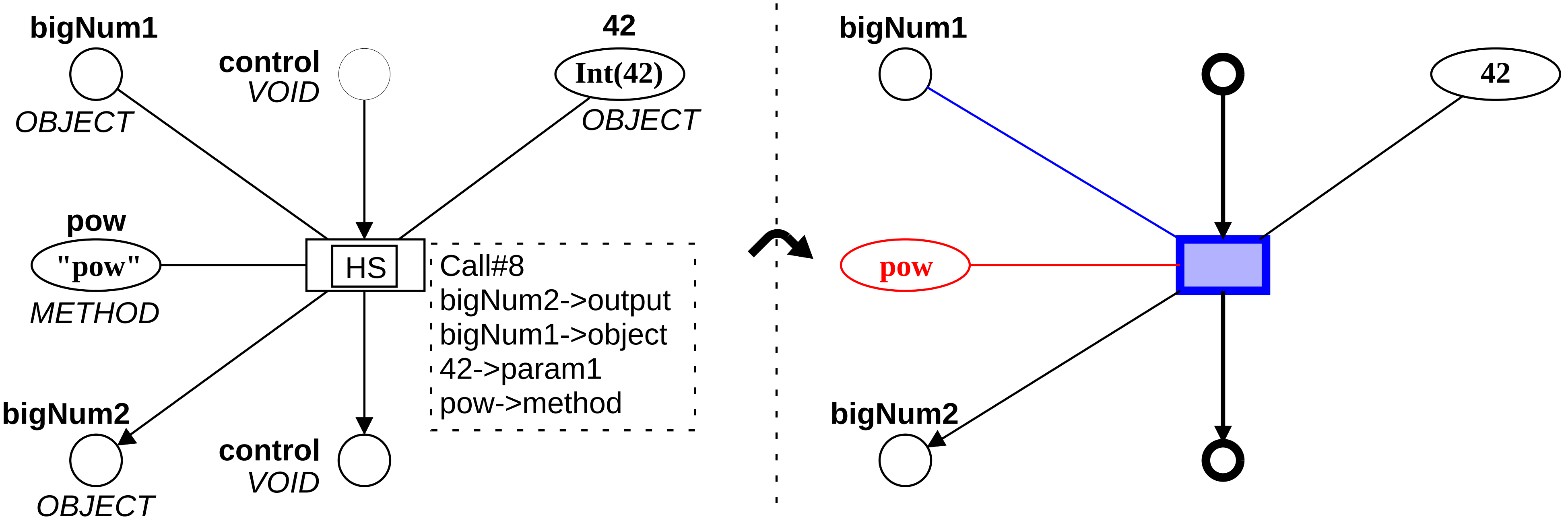
* methodologies: SA, OOA / OOD, specification with CPN, agent-oriented analysis

* tools: generation of OOPN, graphical editor, simulator, Java-Bridge

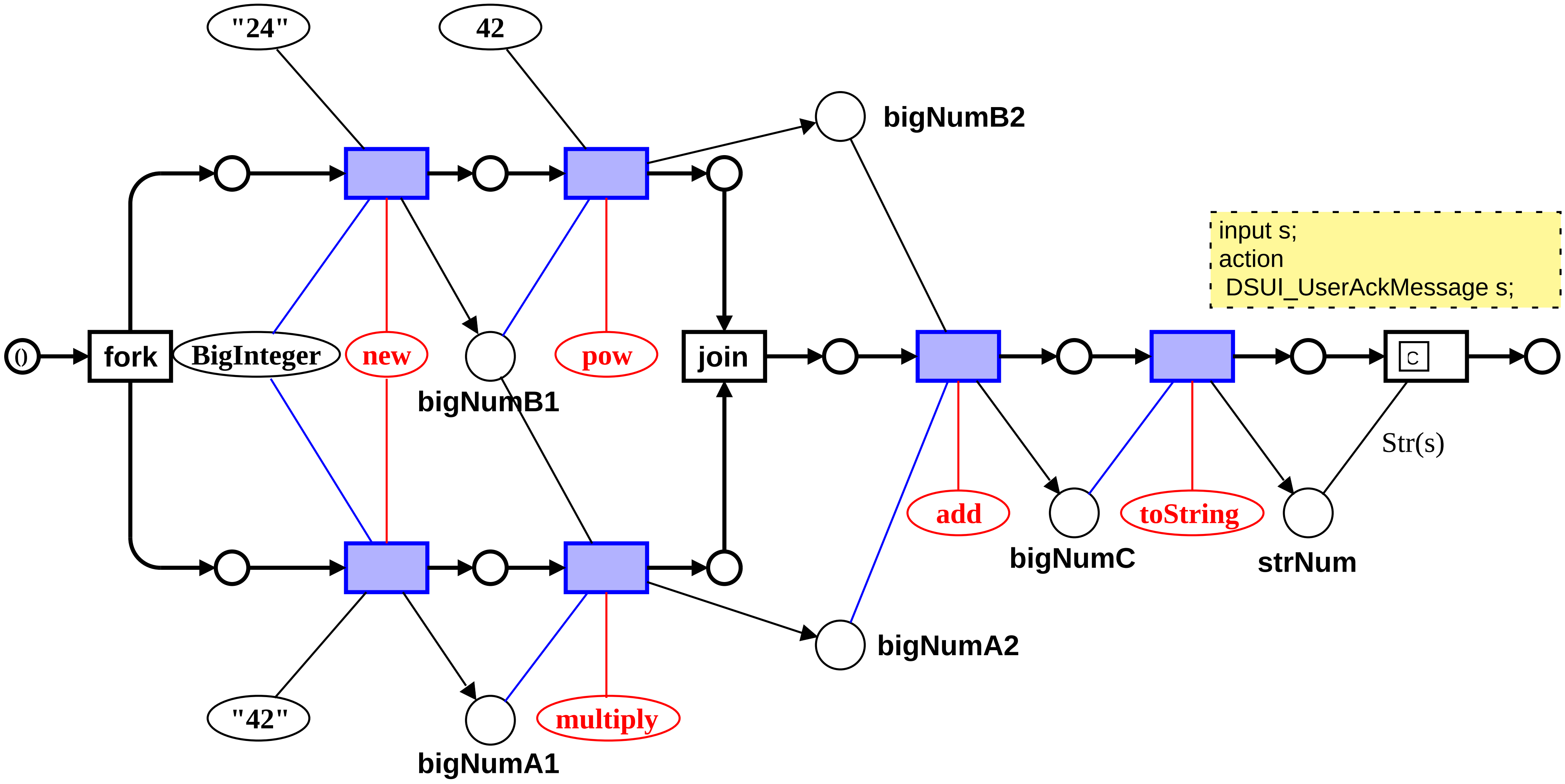
The Message Handling Page



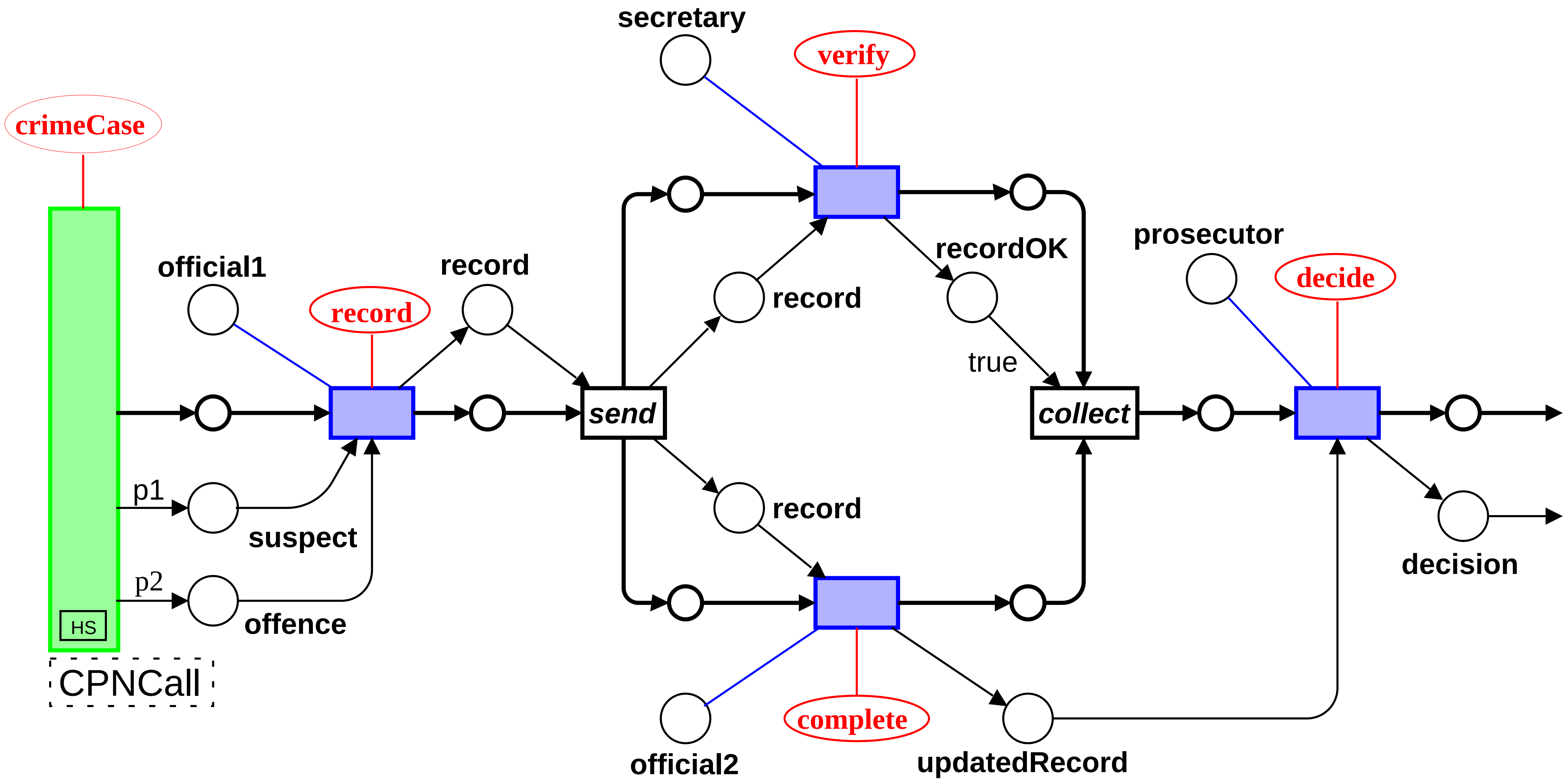
A CPN Notation for Method Calls



An Example Net Calling Java



A CPN that can be Called



A Subpage for Calls to a CPN