CoordInspector

Universidade do Minho
DI-CCTC
Braga, Portugal

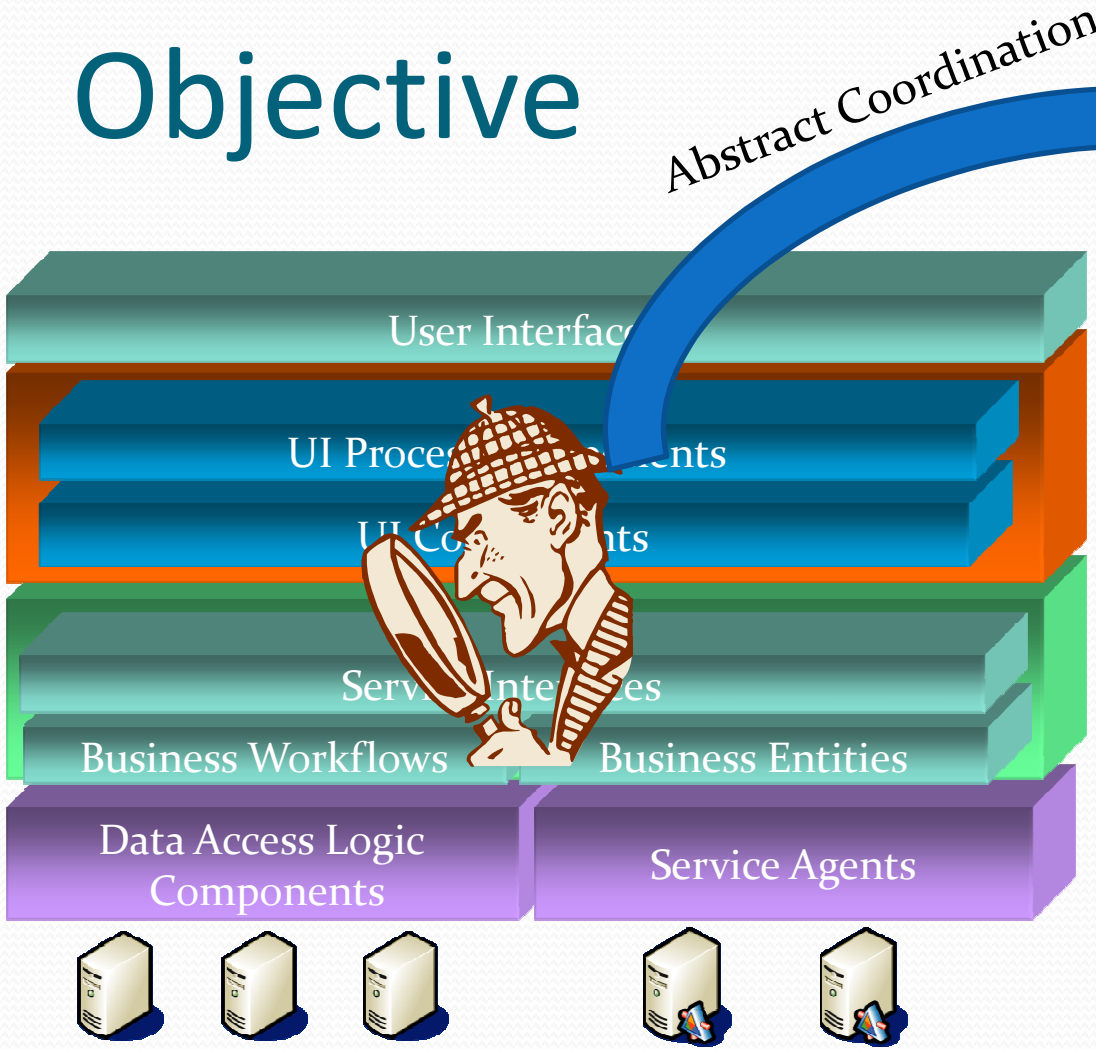
SCAM'08 – Beijing, China– 29 Sept 2008

Nuno F. Rodrigues – nfr@di.uminho.pt
Luís S. Barbosa – lsb@di.uminho.pt

Motivation

- Most modern software applications are built from combinations of existing components, services, and systems.
- Coordination logic is often deeply entangled with the rest of the code
- There are numerous advantages in capturing the internal system logic and the coordination logic separate
  - No need to change the internal system components whenever the interaction changes
  - System components just have to deal with what they are suppose to do, and not with acknowledgment of messages, treating coordination exceptions, etc.
  - System components become more flexible and usable in other contexts.
  - The burden of Concurrency is passed to where it belongs, ie, the Coordination Logic.
  - There are several visual editors and formal languages aiding the use of specific Coordination Languages.

Objective



BPEL

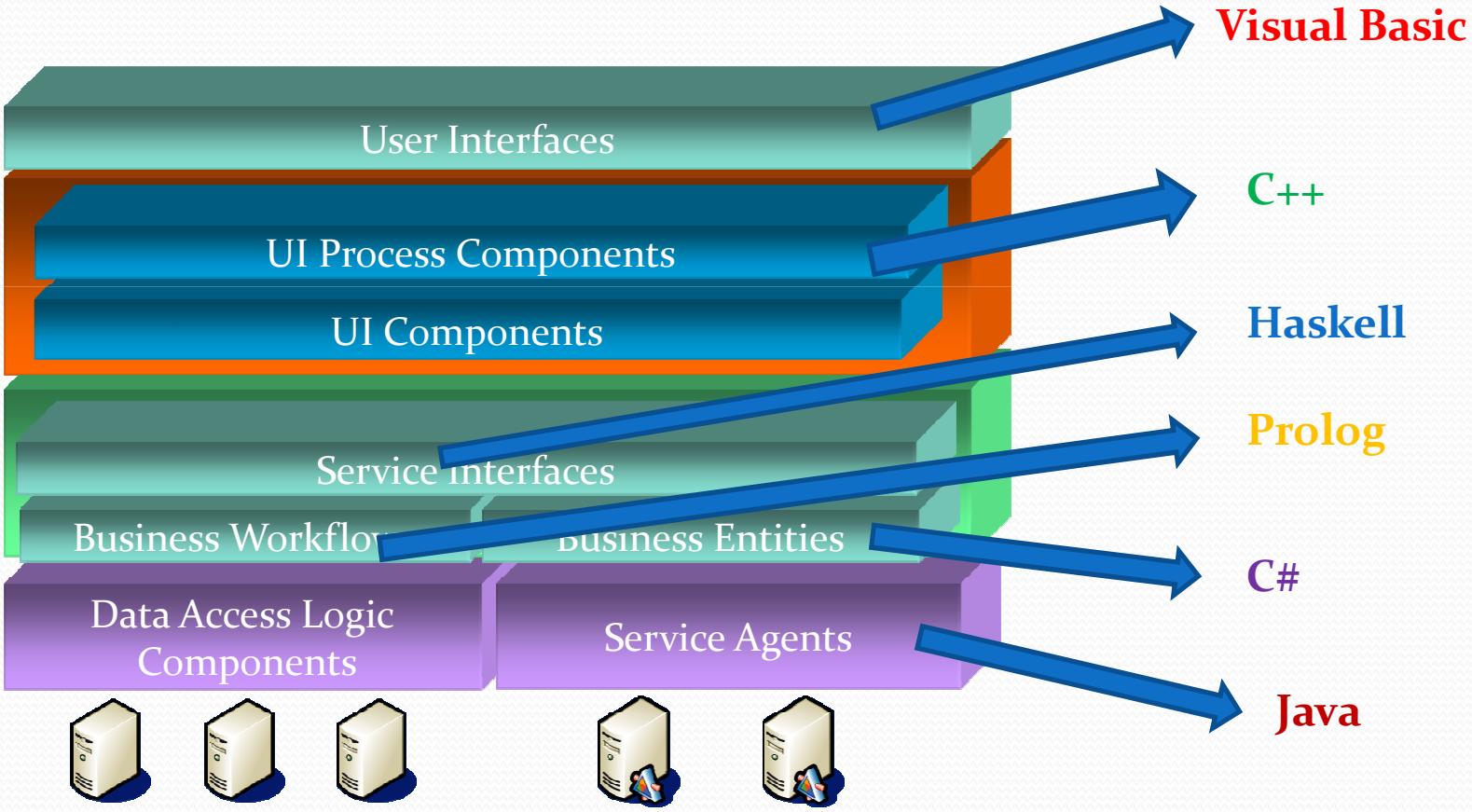
```
<process>
        <flow>
                    ...
```

ORC

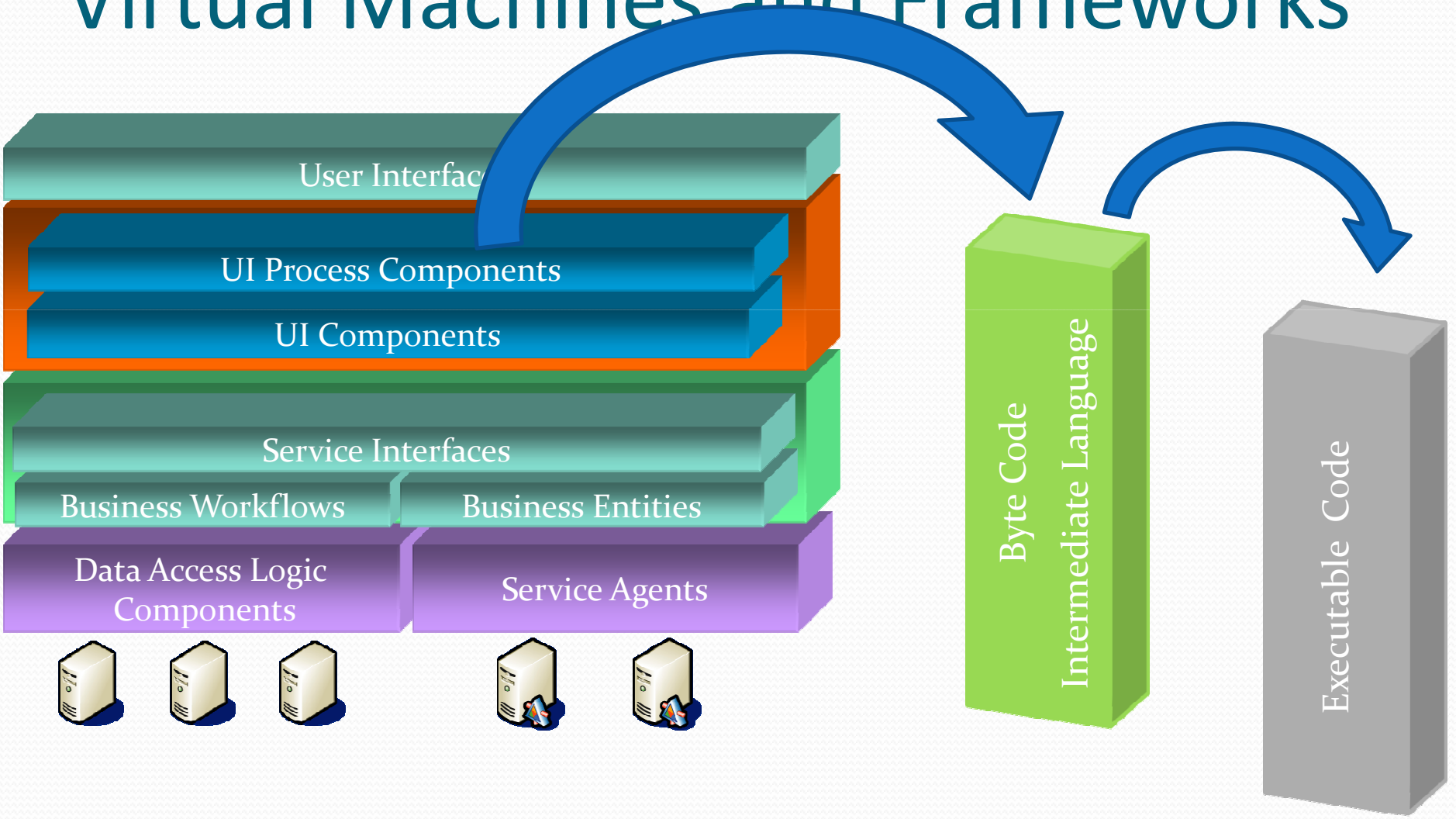
```
P(d, x) :=
    (CNN(d)|BBC(d))
     > x > email(a,x)
```

- Related Work
  - Soap Message Mining
  - Log files analysis

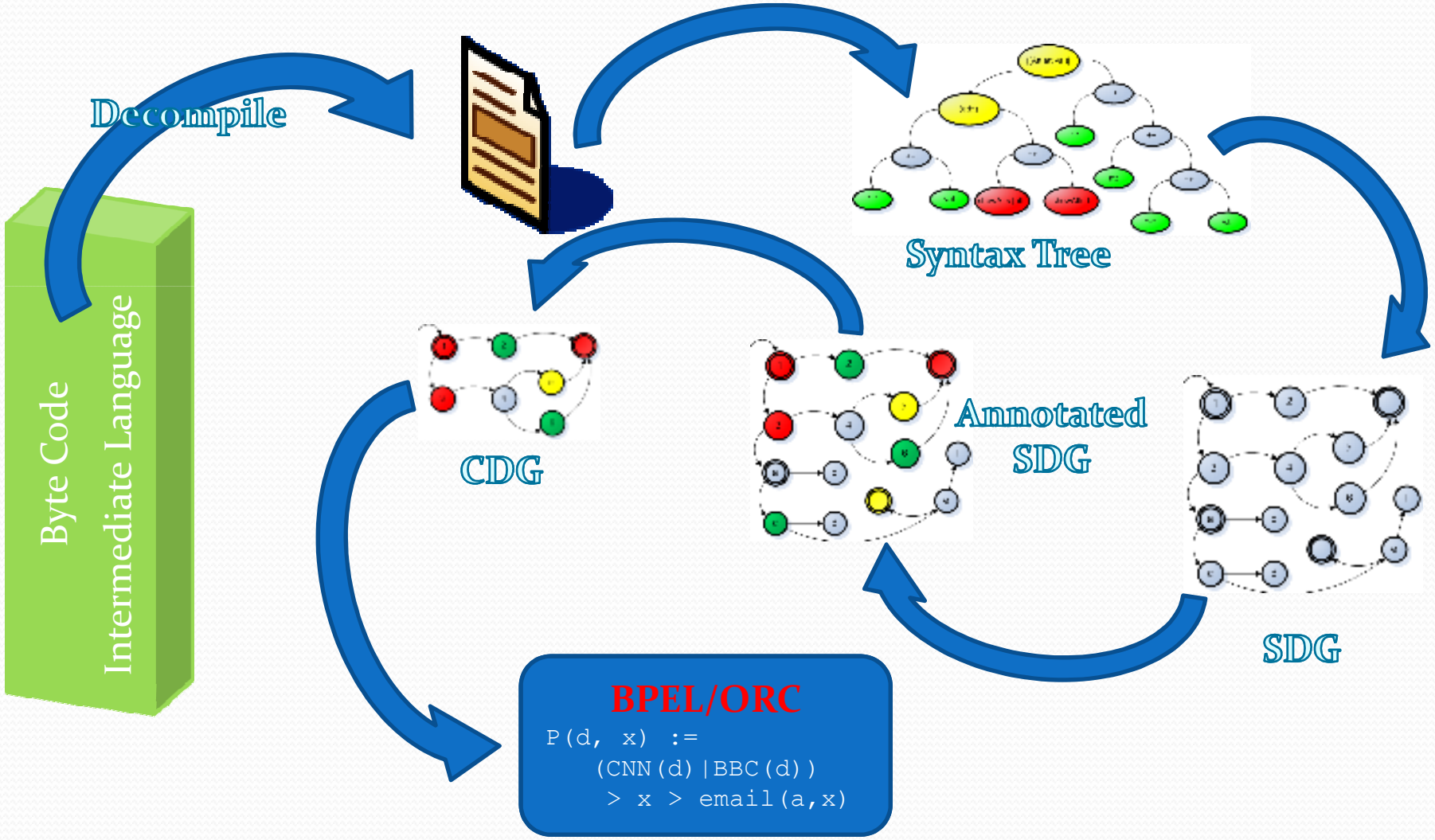
Where to look it for?



Virtual Machines and Frameworks



How can we look it for?



Thank you

http://www.di.uminho.pt/~nfr/tools/tolls.html

Universidade do Minho

DI-CCTC

Braga, Portugal

Nuno F. Rodrigues – nfr@di.uminho.pt

Luís S. Barbosa – lsb@di.uminho.pt