

New Tracks for Dynamic Component Software

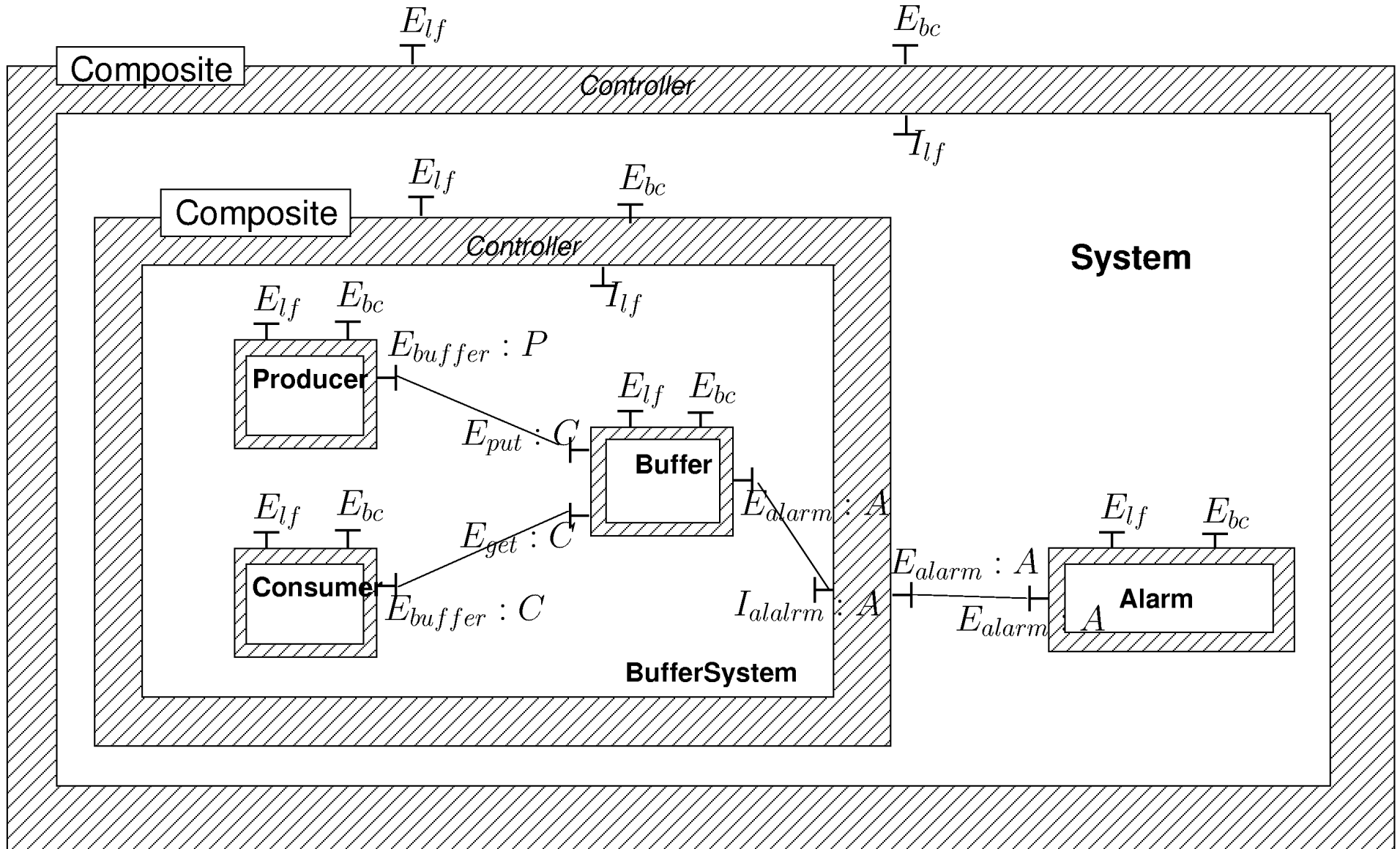
Antonio Cansado

INRIA Sophia-Antipolis

State-of-the-art

- Component Models
 - Fractal
- Component Languages
 - Java/A
 - ArchWare project

Fractal



Fractal / Java

- ADL + IDL + code

```
<component name="Firewall">
  <interface signature="IAccess" role="server" name="IAccess"/>
  <interface signature="IReconfiguration" role="server" name="IReconfiguration"/>
  <interface signature="ILogin" role="client" name="ILogin"/>
  <component name="WebServer">
    <interface signature="IRedirect" role="server" name="IRedirect"/>
    <content class="WebServer"/>
    <controller desc="primitive"/>
  </component>
  ...
  <binding client="AccessPolicy.IRedirect" server="WebServer.IRedirect"/>
  <controller desc="composite"/>
</component>
```

```
public interface IReconfiguration {
    void enablePortBlock(int ip);
    void disablePortBlock(int ip);
}
```

Fractal / Java

```
public String[] listFc() {
    return new String[] { "s" };
}

public Object lookupFc(final String cItf) {
    if (cItf.equals("s")) {
        return service;
    }
    return null;
}
```

```
pub package org.objectweb.proactive.examples.components.helloworld;
import org.objectweb.fractal.api.control.BindingController;

public class ClientImpl implements Runnable, BindingController {
    private Service service;

    public ClientImpl() {
    }
    public void run() {
        service.print("hello world");
    }
}
```

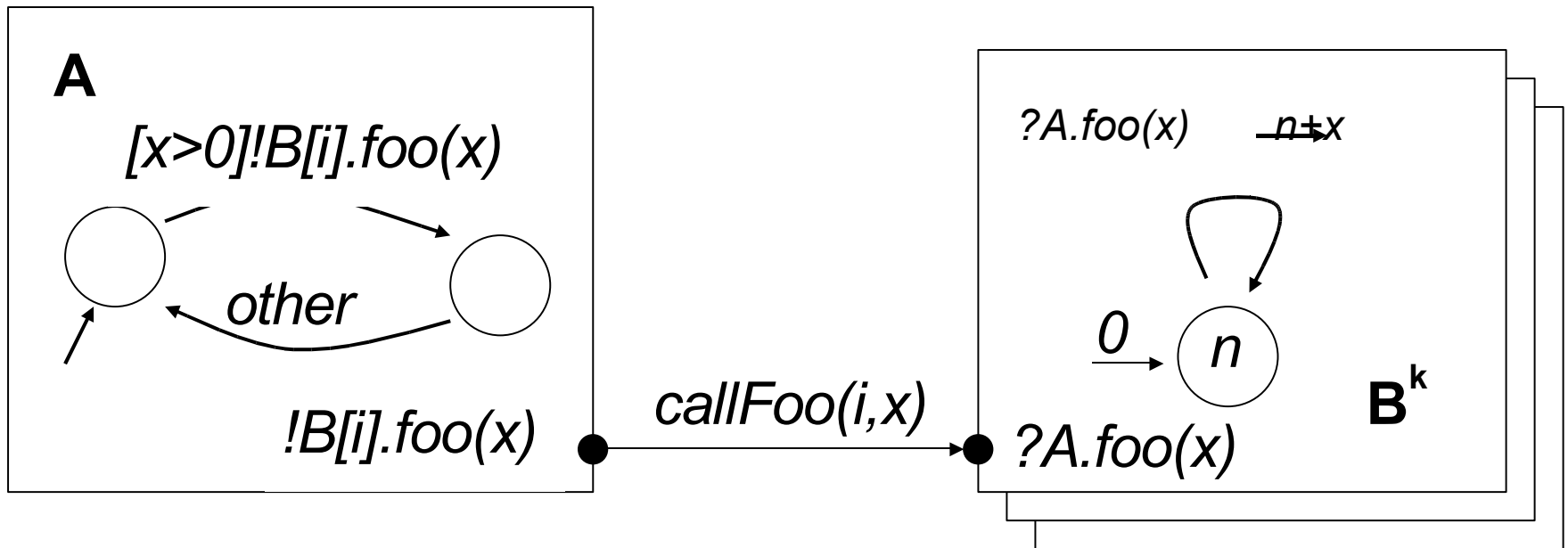
Fractal / Java

- ADL + IDL + code
 - ADL: XML
 - IDL: Java Interfaces
 - Code: Java sources
- **Fractal is a model, not a language**
 - No component primitives in Java
 - Overhead in specification / implementation
 - Confusing?

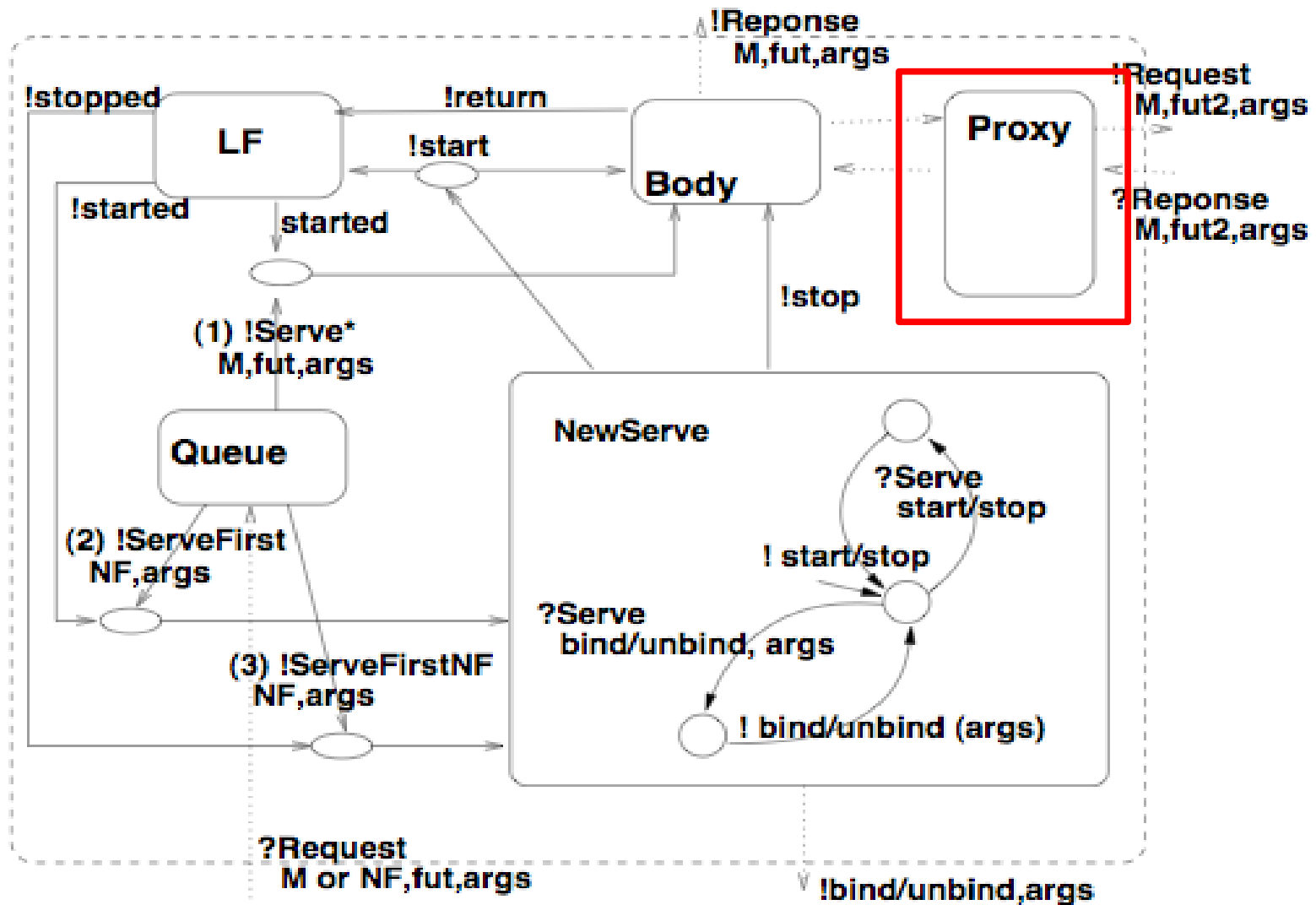
Fractal / pNets

- ADL + IDL + behaviour
 - ADL: XML
 - IDL: Java Interfaces
 - Behaviour: pNets

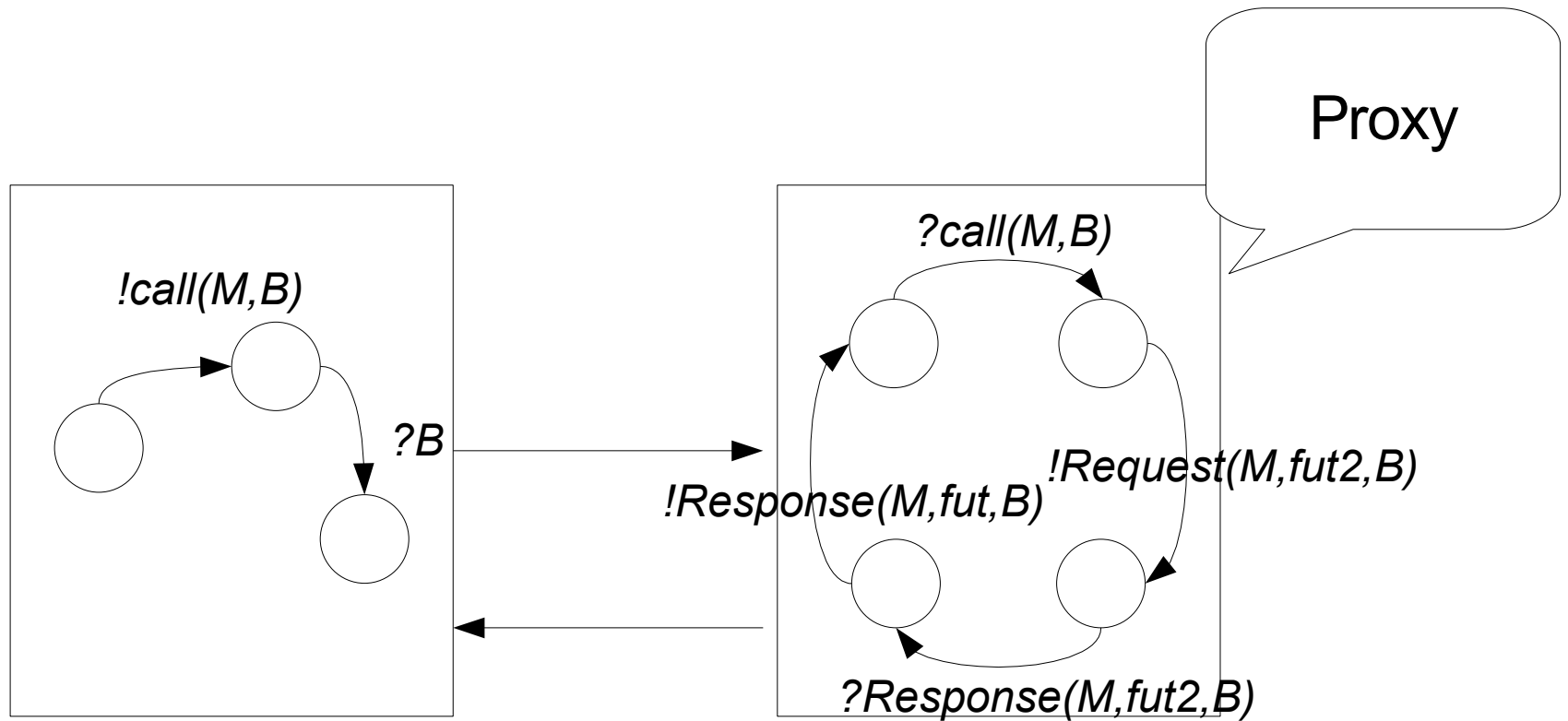
pNets



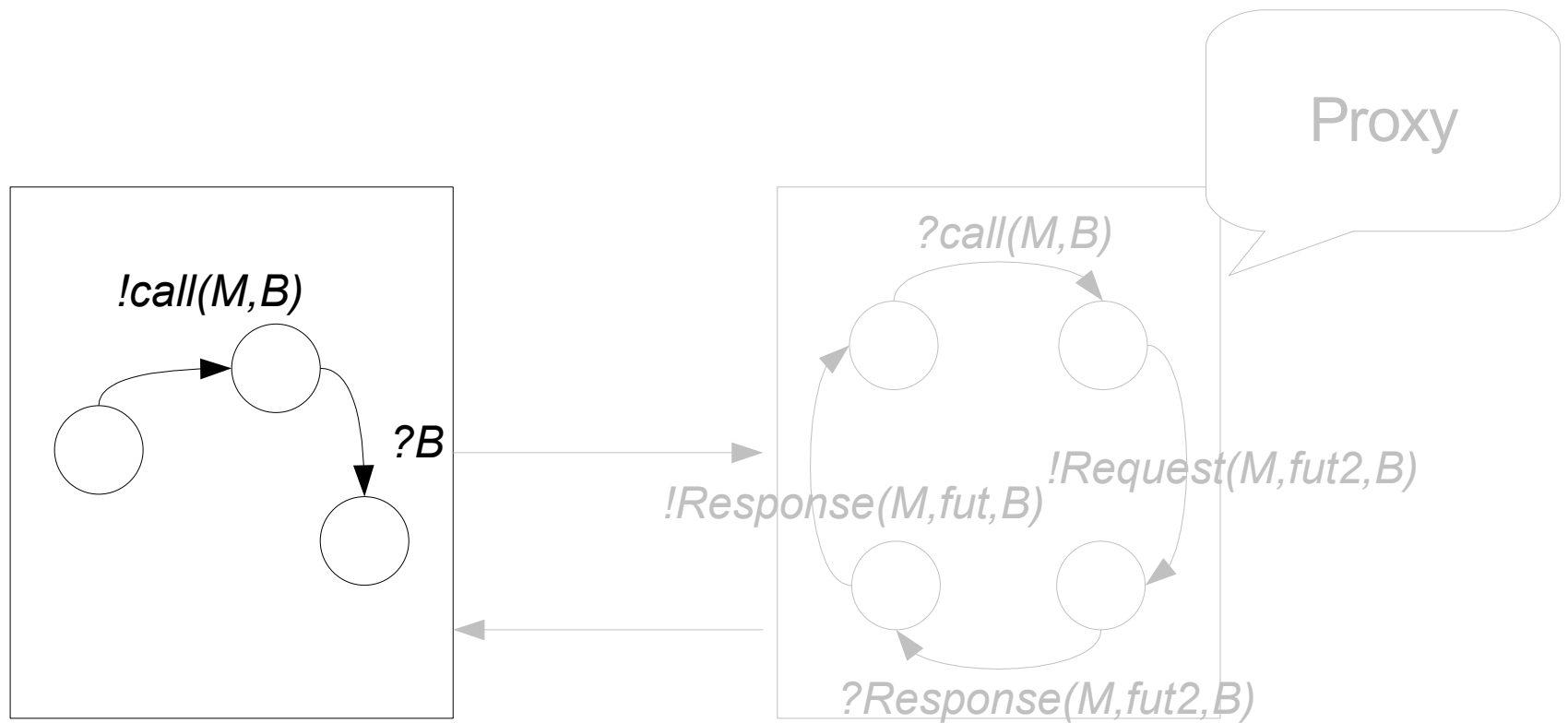
Overhead in Specification



Overhead in Specification



Overhead in Specification



pNets

- Encodes behaviour, but it is not meant to be a system specification
- **No component primitives in pNets!**
 - Up to the user to stick to the model
 - Overhead in the specification

Java/A

- Augment Java with component primitives
 - Components
 - Interfaces
 - Connectors / Ports
- Cleaner code / specification

Java/A Example

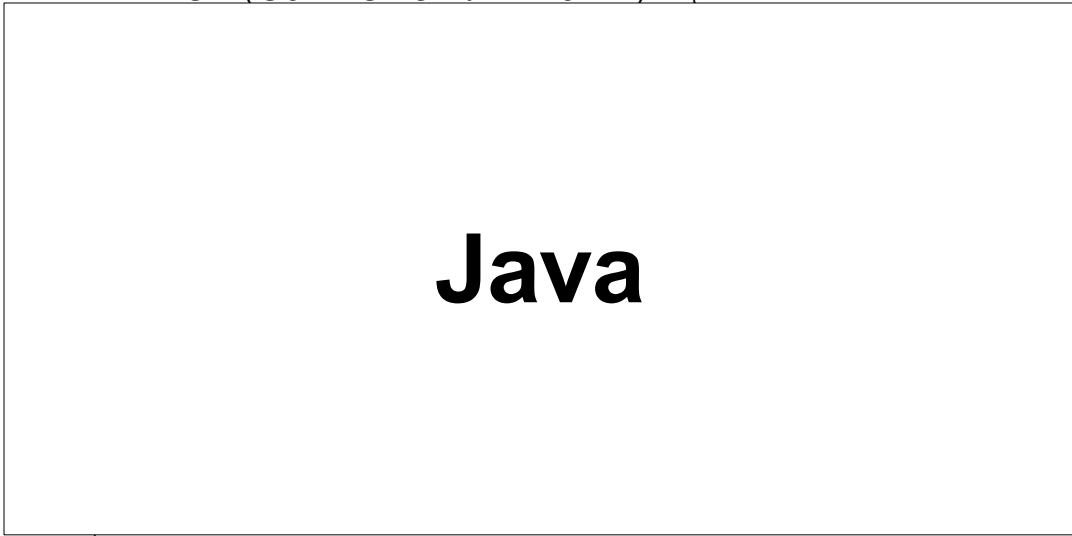
```
simple component Bank {
  Map balance = new HashMap();
  Queue pending = new LinkedList();
  BA current = null;
  Set verifieds = new HashSet();

  dynamic port BA {
    provided {
      signal verifyPIN(IBAN iban, int pin);
      signal withdraw(IBAN iban, Money amount);
    }
    required {
      void pinOk();
      void pinNotOk();
      void withdrawOk();
      void withdrawNotOk();
    }
    <! // protocol of BA ... !>
  }
}
```

Java/A Example

```
port BC {
  provided {
    void pinOk();
    void pinNotOk();
  }
  required {
    void verifyPIN(IBAN iban, int pin);
  }
  <! // protocol of BC
  states {
    initial Initial;
    simple Idle;
    simple Verifying;
  }
  transitions {
    Initial -> Idle;
    Idle -> Verifying { effect verifyPIN(); }
    Verifying -> Idle { trigger pinOk(); }
    Verifying -> Idle { trigger pinNotOk(); }
  }
  !>
}
```

Java/A Example

```
void verifyPIN(BA incoming, IBAN iban, int pin)
    implements BA.verifyPIN(IBAN, int) {
pending.offer(new Object[]{incoming, iban, pin});
synchronized (pending) {
    while (current != null) {
        
        }
    }
catch (javaa.exception.ConnectionException e) {
    e.printStackTrace();
}
}
}
```


Java/A

- **Too close to Java for real verification**
 - Abstract away data manipulation
 - Not suitable as a specification

ArchWare project

- Architecture Description Language (PI-ADL)
- Architecture Analysis Language
- Architecture Refinement Language

ArchWare project

- Architecture Description Language (PI-ADL)
 - Higher-order PI-calculus
 - Run-time viewpoint
 - Structural and Behavioural Architecture-centric constructs
 - UML 2.0

Dynamic Architectures in PI-ADL

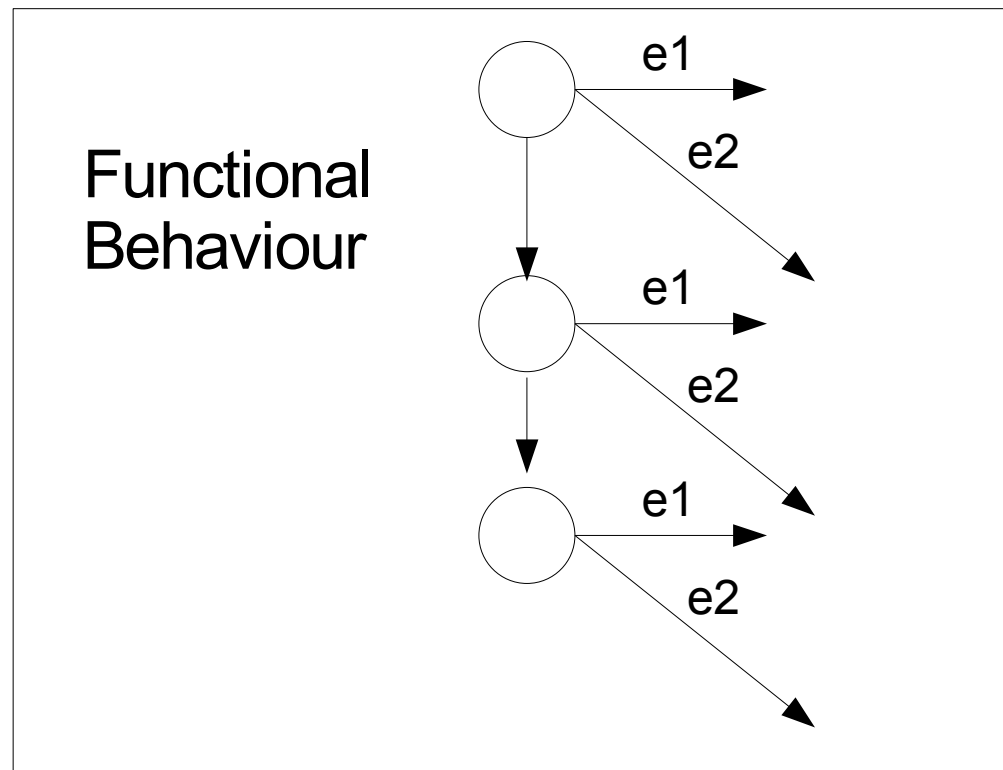
```
component FilterService is abstraction(eos : Natural) {
  ...
  crossOutMultiple is abstraction(p, eos : Natural) {
    via inFilter receive x : Natural.
    if (p <> eos)
    then   if (x mod p) <> 0
           then
             if isConnectionUnified(outFilter)
             then { via outFilter send x.
                    crossOutMultiple(p, eos) }
            else   compose{ ps is PipeService(eos)
                          and fs is FilterService(eos)
                          and cm is { via outFilter send x.
                                      crossOutMultiple(p, eos) }
            } where { outFilter unifies inPipe
                    and outPipe unifies inFilter
            }
            else crossOutMultiple(p, eos)
          else if isConnectionUnified(outFilter)
          then { via outFilter send eos. done }
          else done
        }.
    ...
  }
}
```

Unsolved issues in pNets

- No component primitives
- Exceptions
- Overhead in the Specification

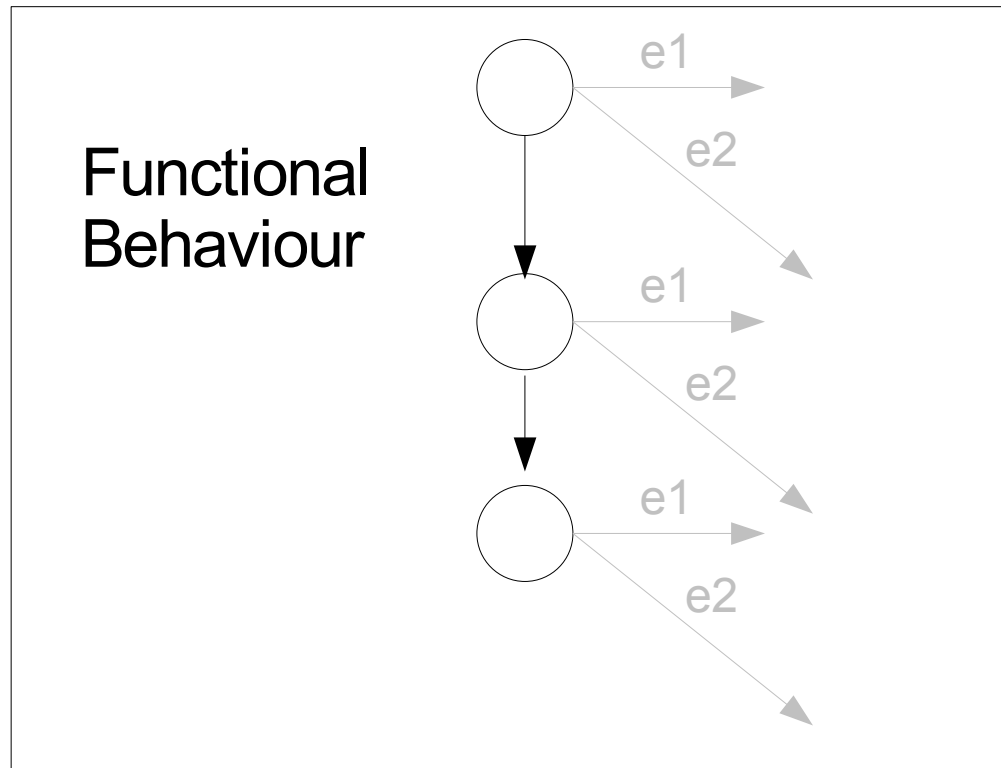
Exceptions

- Add possible transitions to each state
 - What is the user specifying?



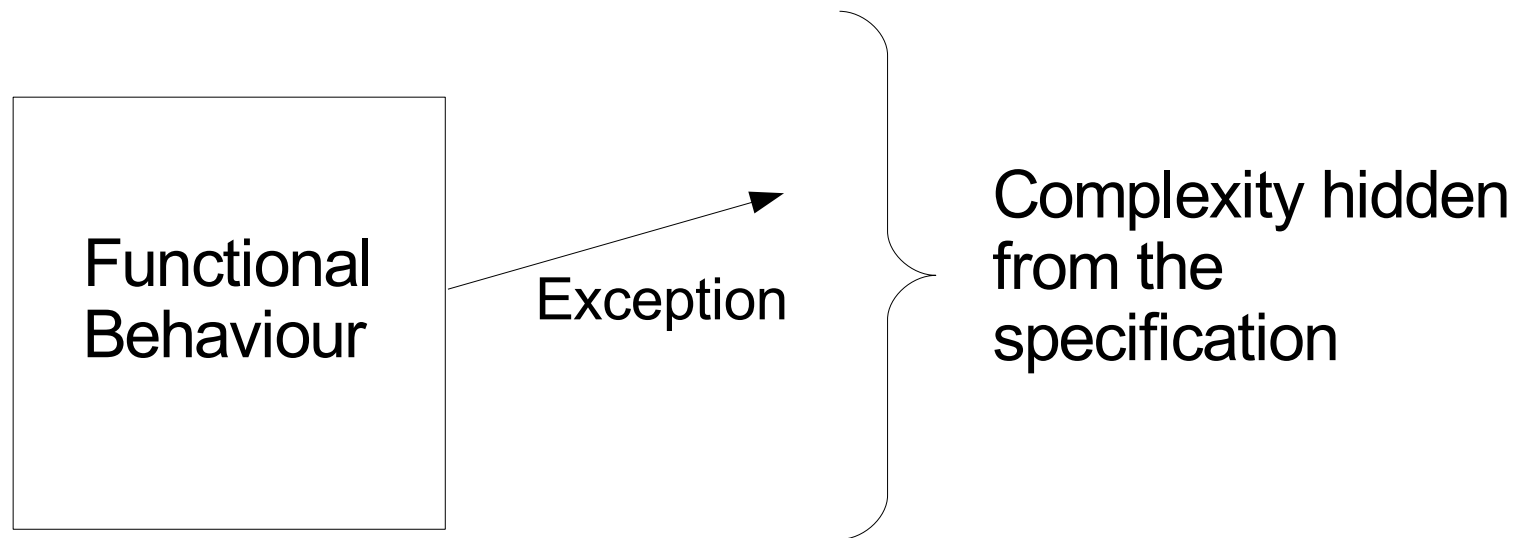
Exceptions

- Add possible transitions to each state
 - What is the user specifying?

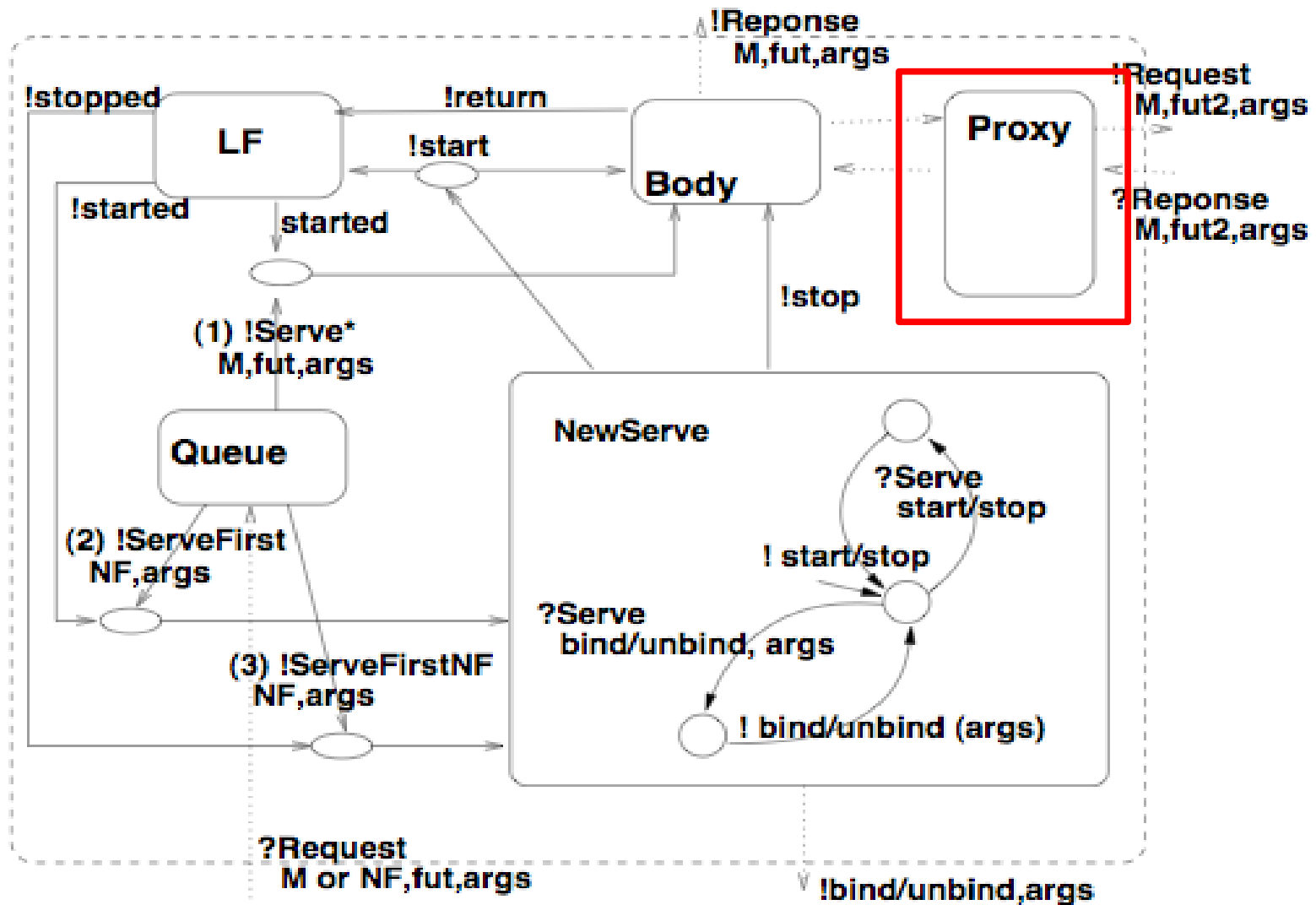


Exceptions

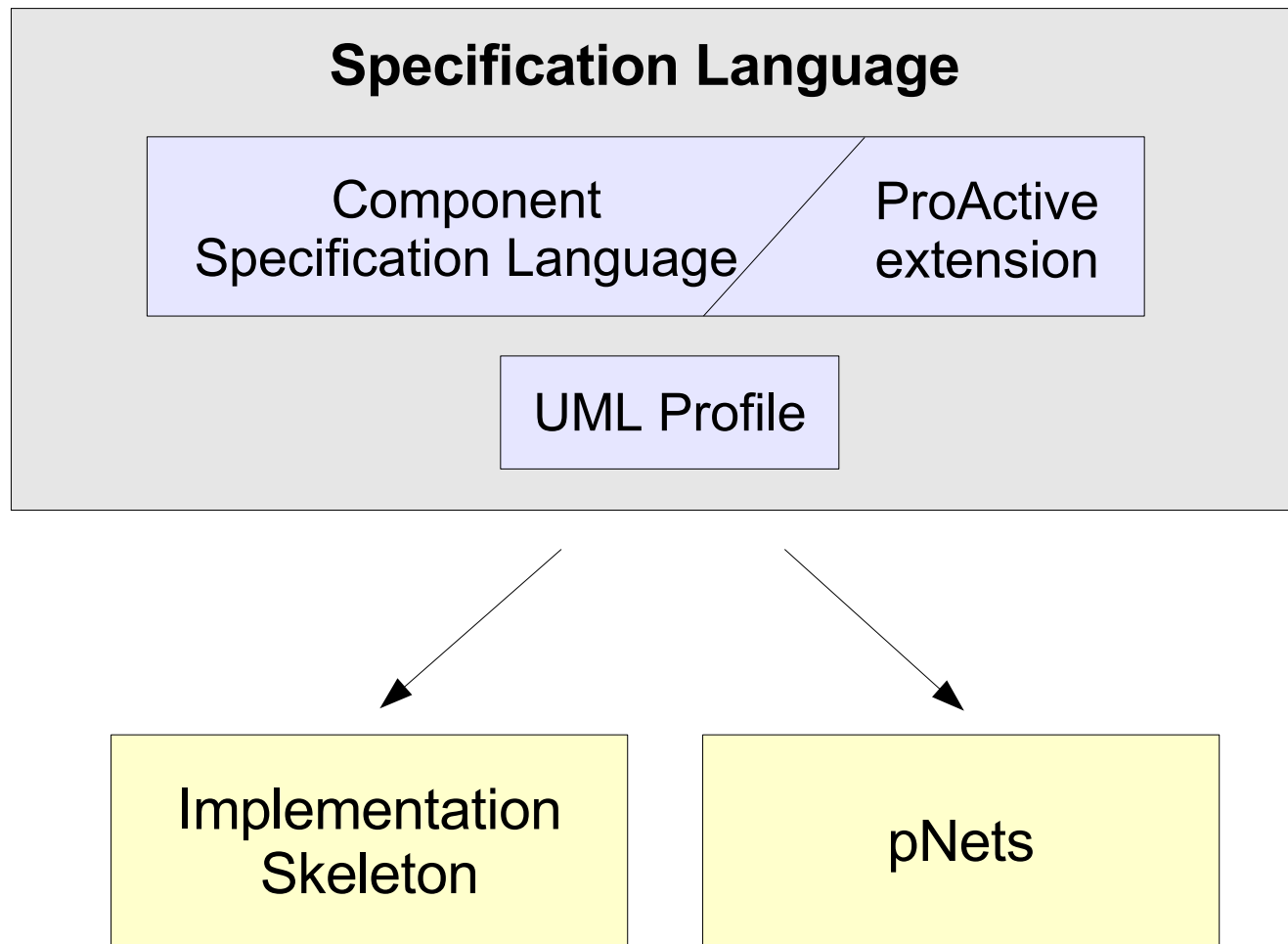
- Outside the functional behaviour
 - LOTOS way? (disable operator, $P \triangleright \text{Exception}$)



Overhead in Specification



Higher-level Specification



Higher-level Specification

- Exceptions
- User-defined controllers (Fractal)
- ProActive extensions
 - Queue, Futures (Proxies, Automatic Continuations)
 - Group communication
 - Skeletons
 - Non-functional exceptions (network?)

Higher-level Specification

- Reconfiguration primitives
 - Dynamic instantiation of components
 - Binding, unbinding
 - Starting, stopping
- Deployment specification (initialisation)