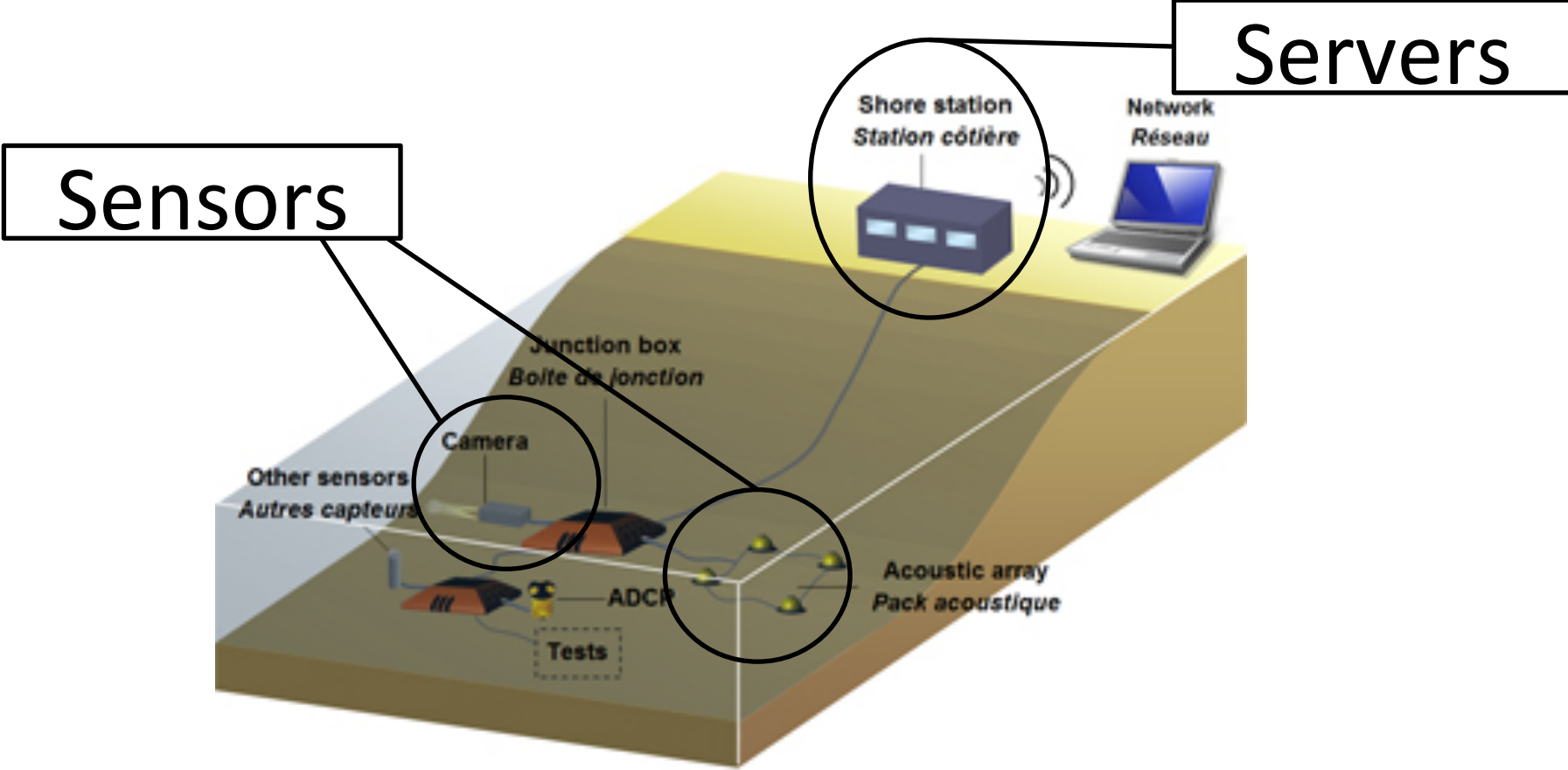# Early exploring design alternatives of smart sensor software with actors

Jean-Philippe Schneider, Zoé Drey, Jean-Christophe Le Lann

IWST 2013 – September, 10th 2013

Cabled Seafloor Observatory

Servers

Sensors

Shore station
Station côtière

Network
Réseau

Junction box
Boîte de jonction

Camera

Other sensors
Autres capteurs

ADCP

Acoustic array
Pack acoustique

Tests
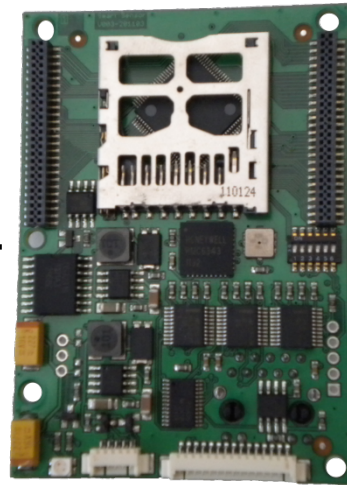
Source: MeDON project (http://medon.info)

⚠️ Large amount of generated data to handle
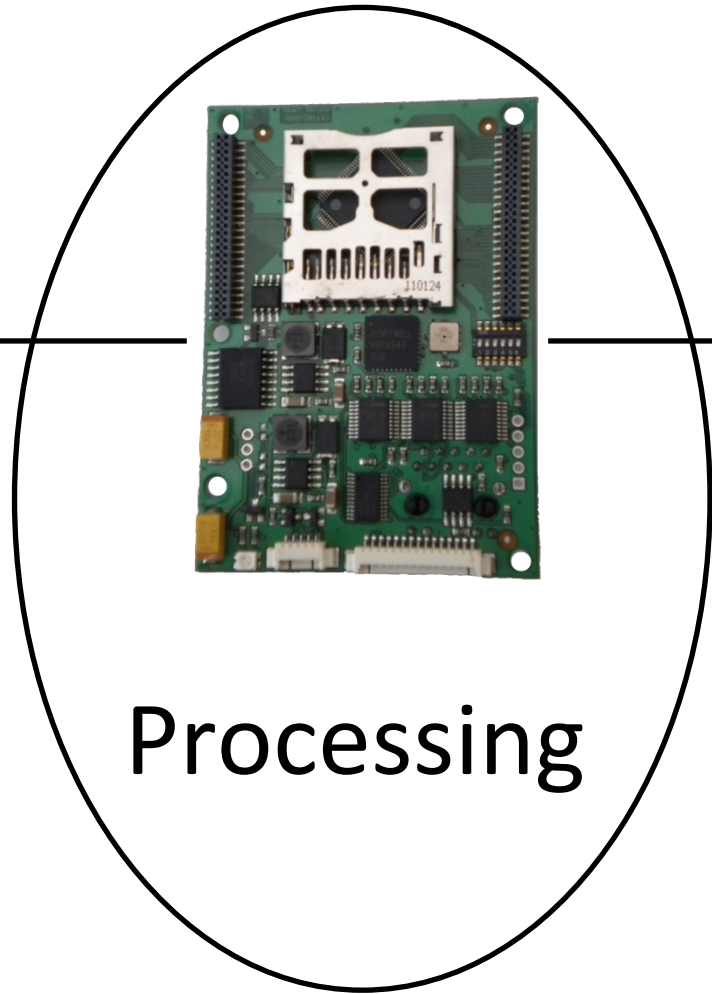
Smart Sensors
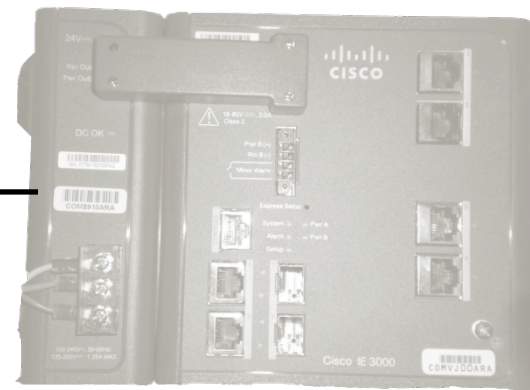

Sensing

Processing

Data dissemination

Smart Sensors

Sensing

Processing

Data dissemination

Embedded software

Prototyping

1. Enable to quickly design various alternatives of architectures:
   - Computation
   - Communication

2. Test the alternatives and help to make a choice

Requirements

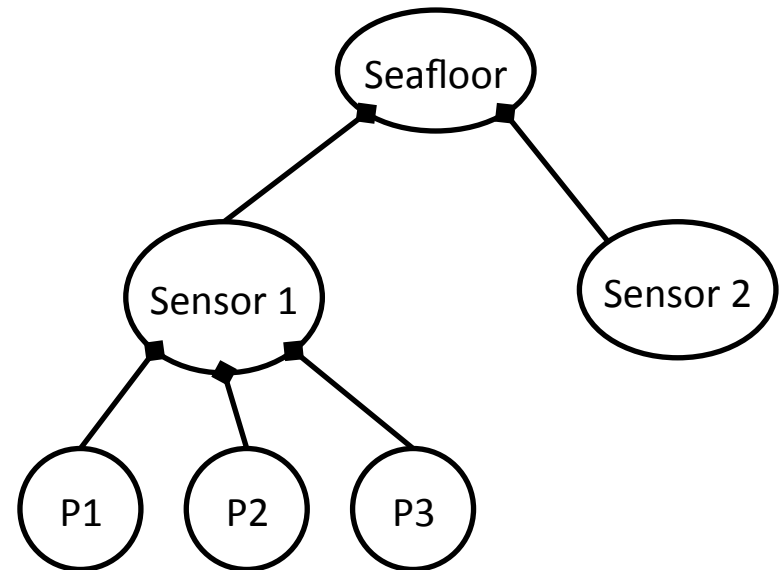Quick modifications: **modularity**

Test: **debug facilities**

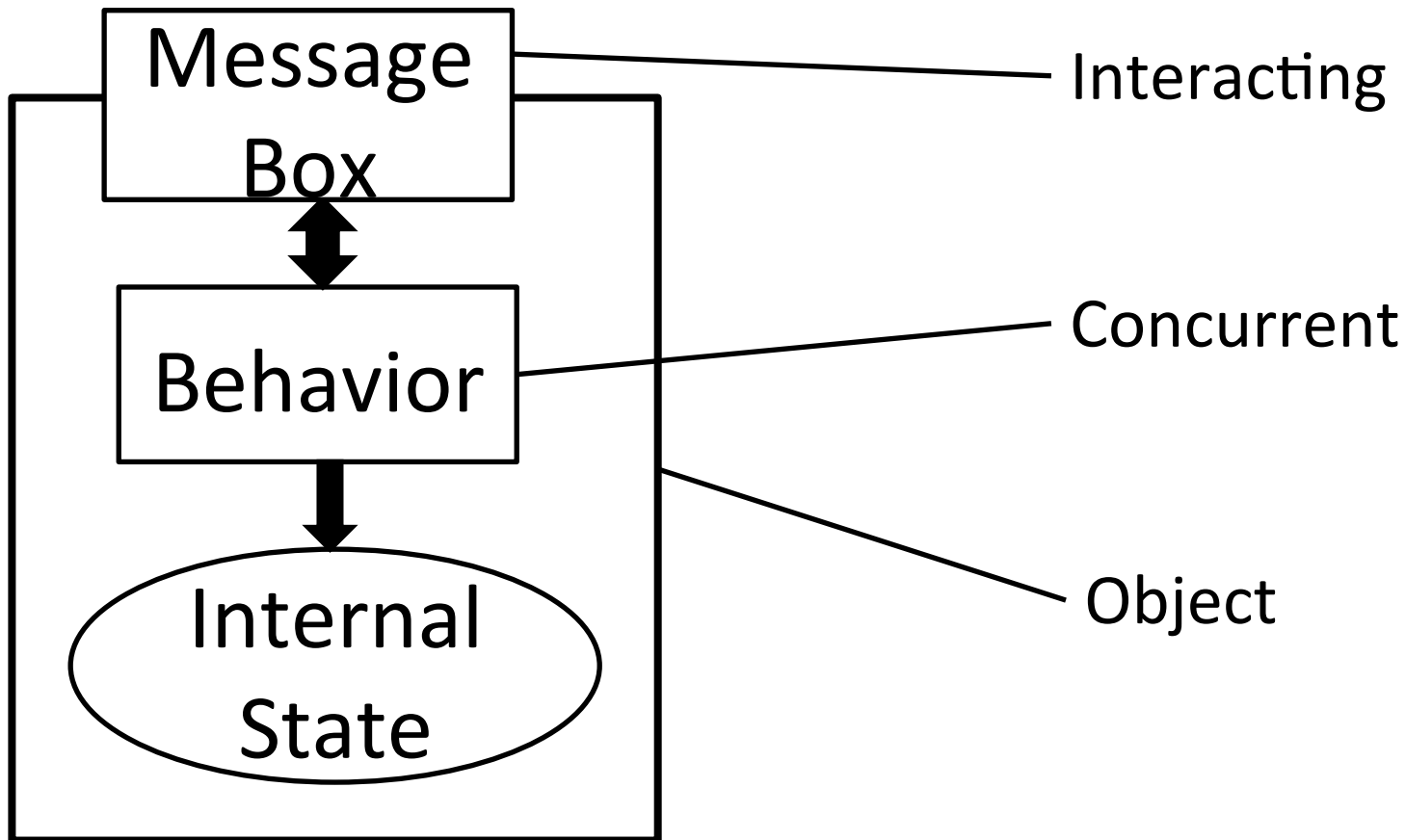# Modularity

✓ Separation of computation and communication

✓ Composite structure

Actor-based approach

Smalltalk for modularity

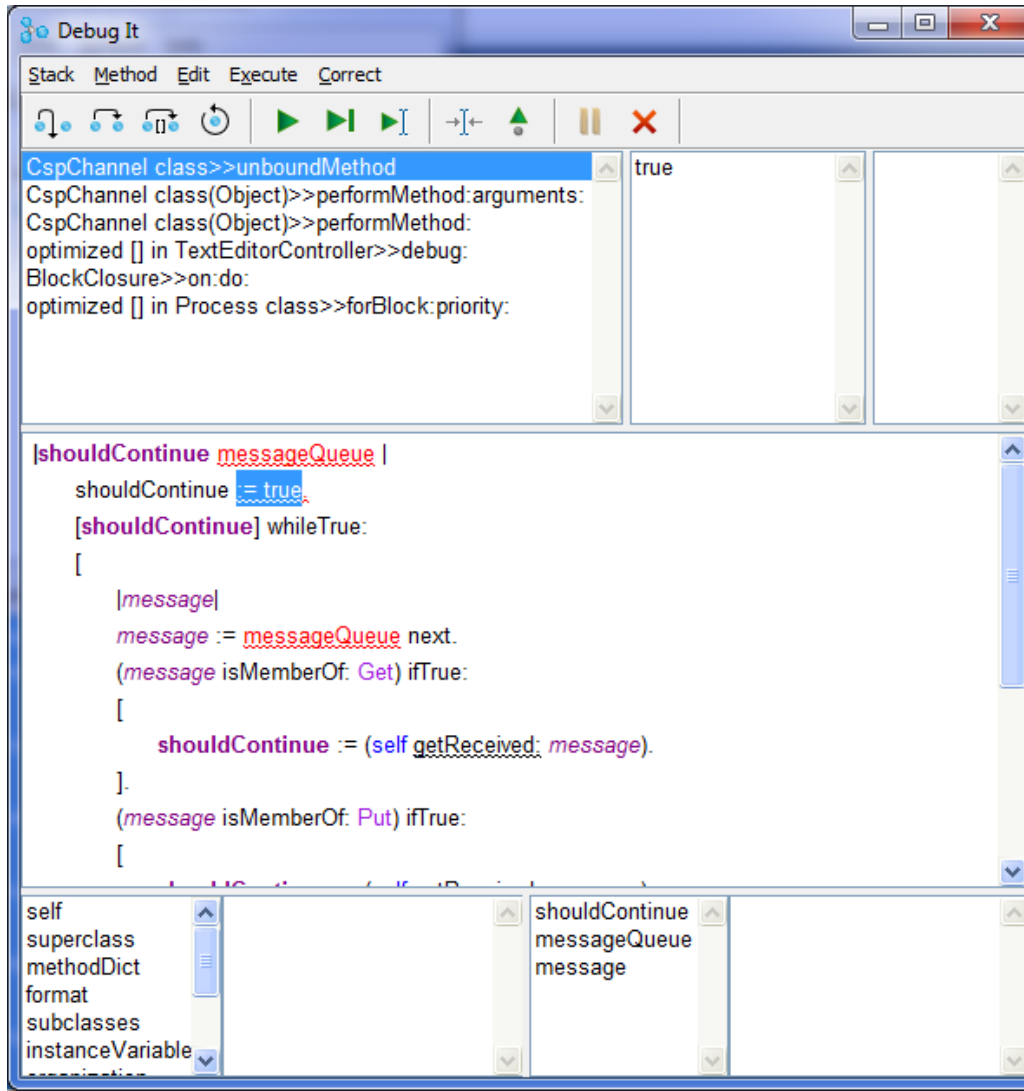- Actors are easy to implement

- Good abstractions for handling composition

Smalltalk for test

- Efficient integrated debug environment

Smalltalk for test

- Efficient integrated debug environment

- Extendable Debug facilities

Biniou

Our framework

Our framework

| Computation Actor 1 | → ← | Communication Actor | ← → | Computation Actor 2 |

✓ Separation of concerns

✓ Agile architecture

Our framework

Put		Get	
Computation Actor 1	→ ←	Communication Actor	← → Computation Actor 2
Ack		Data	

Smalltalk defineClass: #Message

isPut
   ^**false**.

isGet
   ^**false**.

...

Our framework

Smalltalk defineClass: #Get
    superclass: #{Message}
    instanceVariableNames: 'consumer '
isGet
    ^true.

Our framework

```
Put
Get
```

Computation Actor 1	Communication Actor	Computation Actor 2

Ack   Data

```
message := messageQueue next.
(message isGet) ifTrue:
[
    shouldContinue := (self getReceived: message).
].
```

Our framework

Computation Actor 1 — Put → Communication Actor ← Get — Computation Actor 2

Communication Actor ← Ack — Computation Actor 1

Communication Actor — Data → Computation Actor 2

readerMq := **getMessage** consumer.
isPutReceived := false.
[*isPutReceived* not] whileTrue:
[...]

Conclusion and Future Work

Agile architecture for quick design and test

Well-suited environment for fast prototyping

Conclusion and Future Work

Hardware emulation

Composition of models of communication for Sea floor observatories