Software & Language Engineering

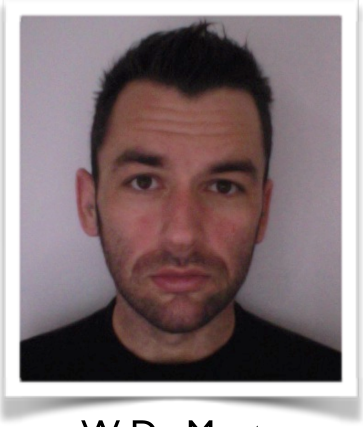
Prof. Dr. Wolfgang De Meuter

Software Languages Lab

Faculteit Wetenschappen en Bio-Ingenieurswetenschappen
Vrije Universiteit Brussel

Our Team

- Programming Languages & Software Engineering
- 4 Professors; ±35 researchers (±8 Post-doc)
- 60+ PhD theses finished, ±20 ongoing
- 150+ Master theses



W. De Meuter

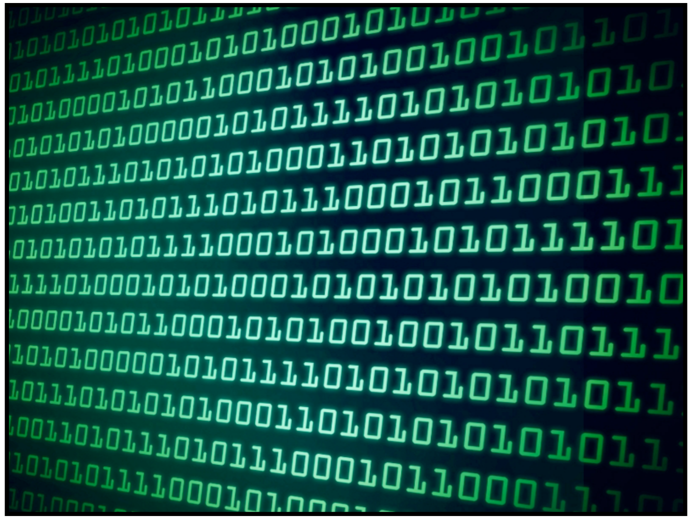
T. D'Hondt

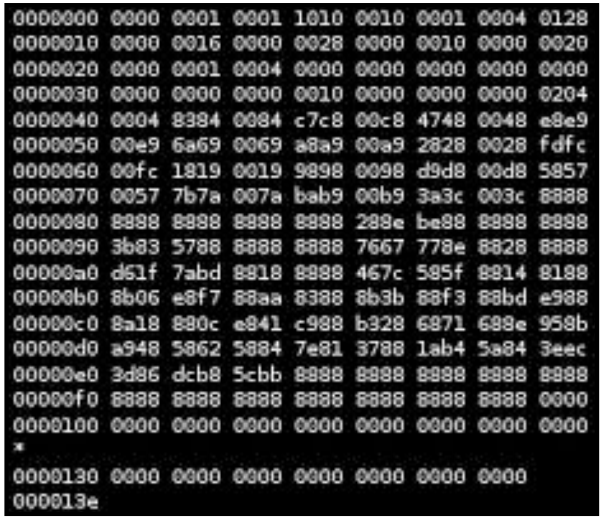
V. Jonckers

?

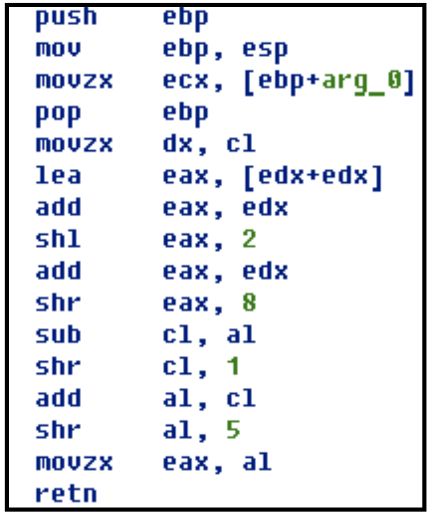
X

Abstractions

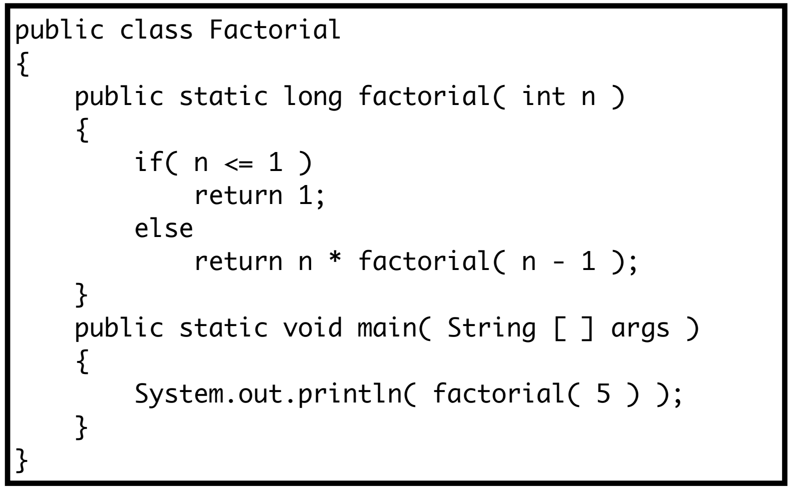




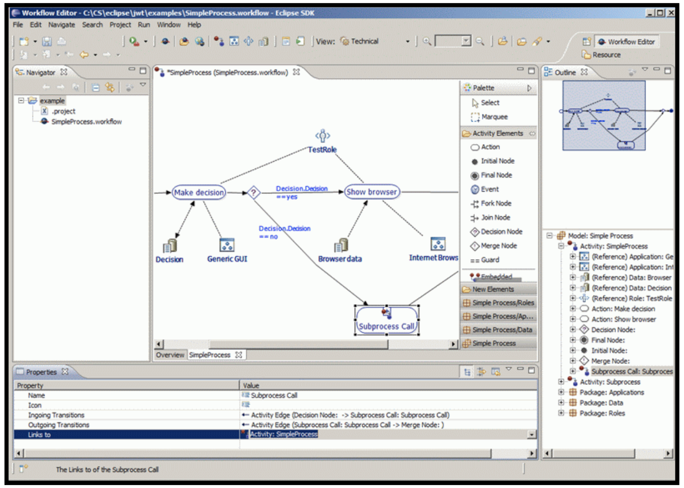
```
0000000 0000 0001 0001 1010 0010 0001 0004 0128
0000010 0000 0016 0000 0028 0000 0010 0000 0020
0000020 0000 0001 0004 0000 0000 0000 0000 0000
0000030 0000 0000 0000 0010 0000 0000 0000 0204
0000040 0004 8384 0084 c7c8 00c8 4748 0048 e8e9
0000050 00e9 6a69 0069 a8a9 00a9 2828 0028 fdfc
0000060 00fc 1819 0019 9898 0098 d9d8 00d8 5857
0000070 0057 7b7a 007a bab9 00b9 3a3c 003c 8888
0000080 8888 8888 8888 8888 288e be88 8888 8888
0000090 3b83 5788 8888 8888 7667 778e 8828 8888
00000a0 d61f 7abd 8818 8888 467c 585f 8814 8188
00000b0 8b06 e8f7 88aa 8388 8b3b 88f3 88bd e988
00000c0 8a18 880c e841 c988 b328 6871 688e 958b
00000d0 a948 5862 5884 7e81 3788 1ab4 5a84 3eec
00000e0 3d86 dcb8 5cbb 8888 8888 8888 8888 8888
00000f0 8888 8888 8888 8888 8888 8888 8888 0000
0000100 0000 0000 0000 0000 0000 0000 0000 0000
*
0000130 0000 0000 0000 0000 0000 0000 0000
000013e
```



```
push    ebp
mov     ebp, esp
movzx   ecx, [ebp+arg_0]
pop     ebp
movzx   dx, cl
lea     eax, [edx+edx]
add     eax, edx
shl     eax, 2
add     eax, edx
shr     eax, 8
sub     cl, al
shr     cl, 1
add     al, cl
shr     al, 5
movzx   eax, al
retn
```



```
public class Factorial
{
    public static long factorial( int n )
    {
        if( n <= 1 )
            return 1;
        else
            return n * factorial( n - 1 );
    }
    public static void main( String [ ] args )
    {
        System.out.println( factorial( 5 ) );
    }
}
```



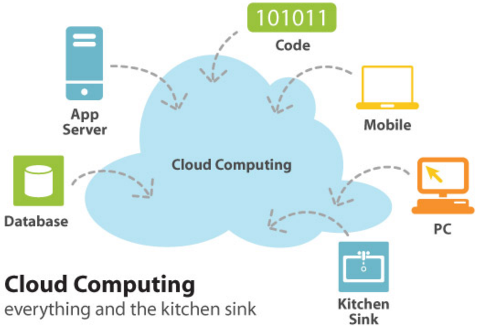
Property	Value
Name	Subprocess Call
Icon	
Ingoing Transitions	Activity Edge (Decision Node: -> Subprocess Call: Subprocess Call)
Outgoing Transitions	Activity Edge (Subprocess Call: Subprocess Call -> Merge Node:)
Links to	Activity: SimpleProcess

The Links to of the Subprocess Call

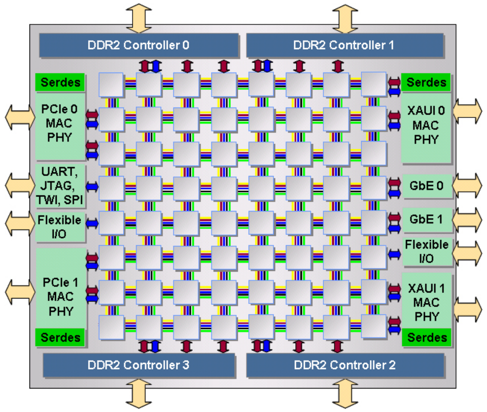
How can we make languages, techniques and tools that allow us to write software in a better way?

The World is Changing!

Mobile Platforms



The Cloud



Multicores/Manycores

Supercomputers

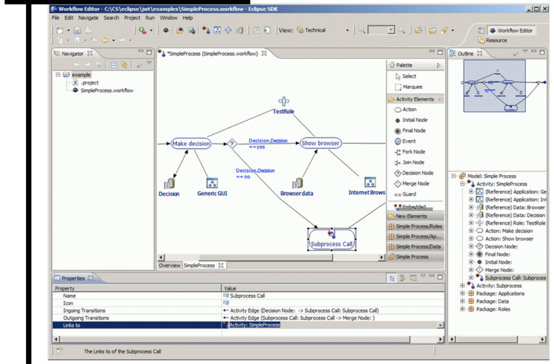
Hardware Evolutions Require Novel Abstractions

Language Related Research

Language Design

```
quicksort []     = []
quicksort (p:xs) = smallerSorted ++ [p] ++ biggerSorted
  where smallerSorted = quicksort [x | x <- xs, x <= p]
        biggerSorted  = quicksort [x | x <- xs, x > p ]
```

Tools



Property	Value
Name	Subprocess Call
Icon	
Ingoing Transitions	Activity Edge (Decision Node: -> Subprocess Call: Subprocess Call)
Outgoing Transitions	Activity Edge (Subprocess Call: Subprocess Call -> Merge Node:)
Links to	Activity: SimpleProcess

The Links to of the Subprocess Call

$$\frac{C \vdash_s \sigma <: \tau \qquad C, E \vdash_s M : \sigma}{C, E \vdash_s M : \tau}$$

$$\frac{\hat{C}, \hat{E} \vdash_s f : \sigma \to \tau \;\; and \;\; \rho \;\models_s\; \hat{C}, \hat{E}}{C, E \vdash_s \langle f, \rho \rangle : \sigma_\rho \to \tau_\rho}$$

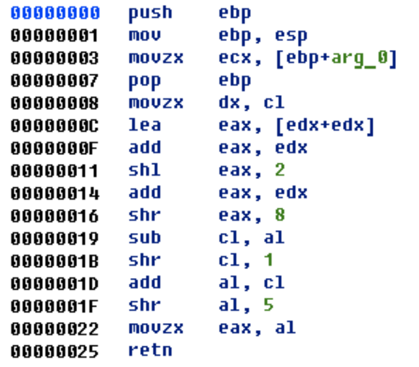
$$\frac{\hat{C}, \hat{E} \vdash_s f : \forall t . \tau \;\; and \;\; \rho \;\models_s\; \hat{C}, \hat{E}}{C, E \vdash_s \langle f, \rho \rangle : \forall t . \tau_\rho}$$

$$\frac{\hat{C}, \hat{E} \vdash_s f : \forall t \; <\# \; \gamma . \tau \;\; and \;\; \rho \;\models_s\; \hat{C}, \hat{E}}{C, E \vdash_s \langle f, \rho \rangle : \forall t \; <\# \; \gamma_\rho . \tau_\rho}$$

$$\frac{\hat{C}, \hat{E} \vdash_s f : \forall t <: \gamma . \tau \;\; and \;\; \rho \;\models_s\; \hat{C}, \hat{E}}{C, E \vdash_s \langle f, \rho \rangle : \forall t <: \gamma_\rho . \tau_\rho}$$

Formalisation

Implementation

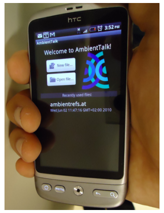


```
00000000  push   ebp
00000001  mov    ebp, esp
00000003  movzx  ecx, [ebp+arg_0]
00000007  pop    ebp
00000008  movzx  dx, cl
0000000C  lea    eax, [edx+edx]
0000000F  add    eax, edx
00000011  shl    eax, 2
00000014  add    eax, edx
00000016  shr    eax, 8
00000019  sub    cl, al
0000001B  shr    cl, 1
0000001D  add    al, cl
0000001F  shr    al, 5
00000022  movzx  eax, al
00000025  retn
```

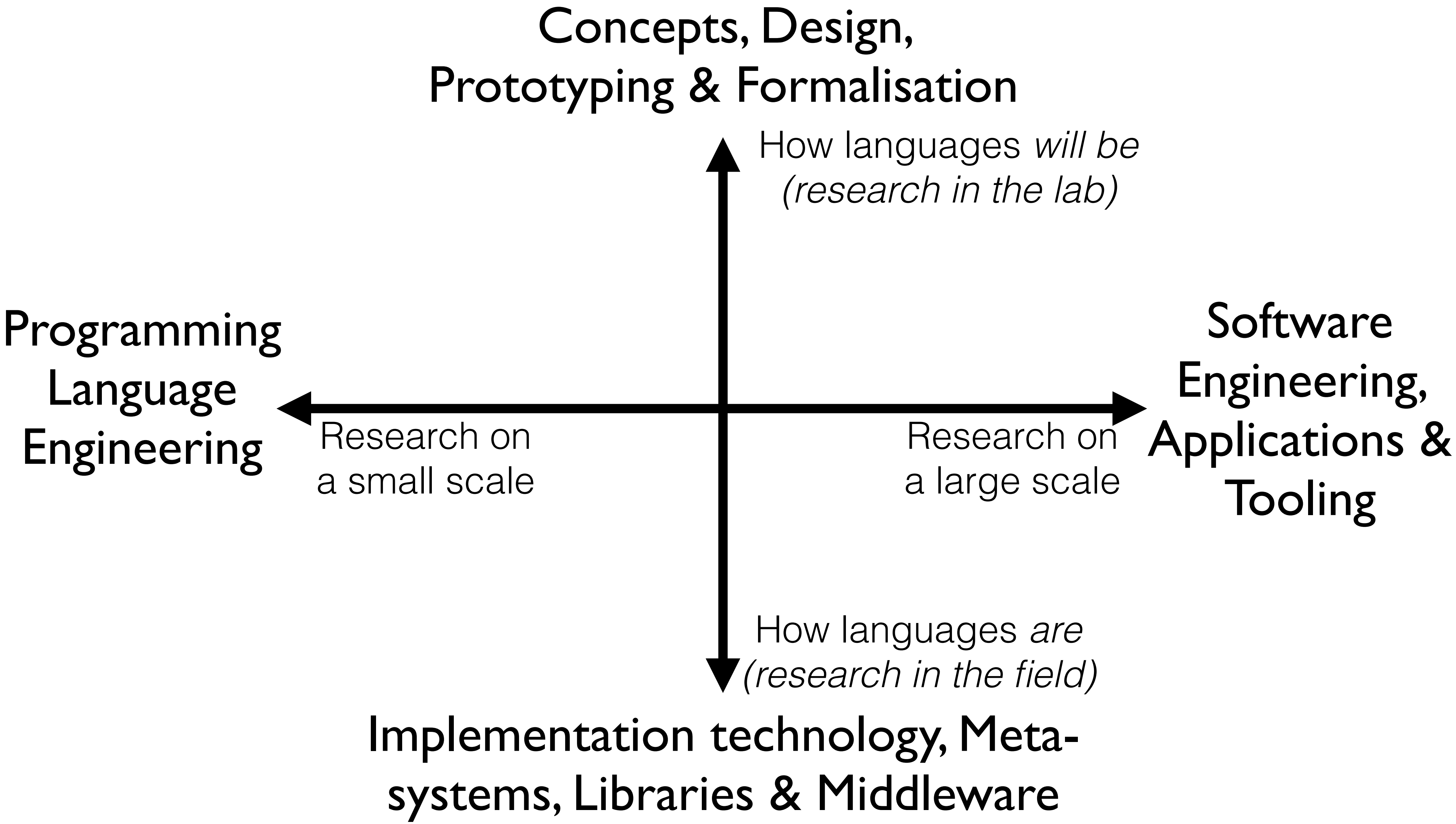
```
network.online();

deftype Pulsar;

when: Pulsar
discovered: { |far|
  whenever: far
  disconnected: {
    print("gone again") };
  whenever: far
  reconnected:  {
    print("here again") }}
```

Middleware

Applications



Our Research Spectrum



Concepts, Design,
Prototyping & Formalisation

How languages *will be*
(research in the lab)

Programming
Language
Engineering

Research on
a small scale

Research on
a large scale

Software
Engineering,
Applications &
Tooling

How languages *are*
(research in the field)

Implementation technology, Meta-
systems, Libraries & Middleware

Middleware >< Languages

Middleware

```
{ Network n = java.myLib.getNetwork();
  Object o = n.smellObject("Printer");
  Message m = new Message();
  m.setReceiver(o);
  Object[] a = new Object[0];
  m.setArgs(a);
  m.setSelector("ping");
  m.send()
}
```

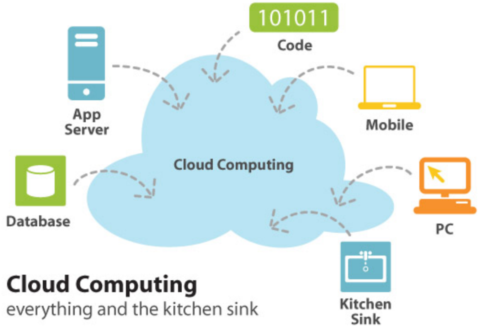
Languages

```
{ o = network ? Printer;
  o ← ping() }
```

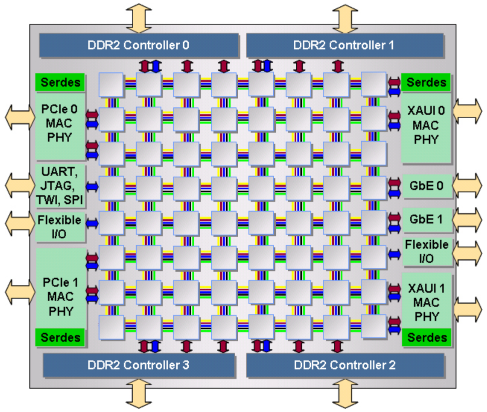
The *concepts* are (often) identical, but with a difference in responsibility (compiler >< programmer)

The World is Changing!

Mobile Platforms



The Cloud

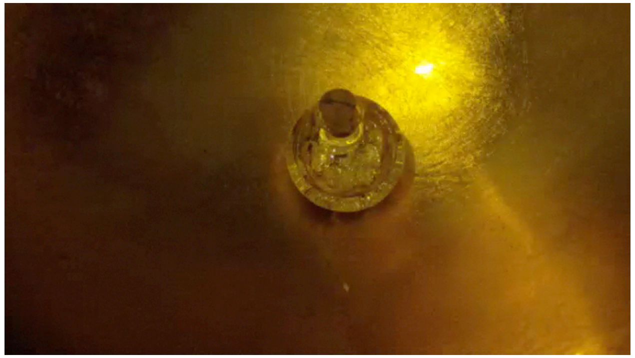


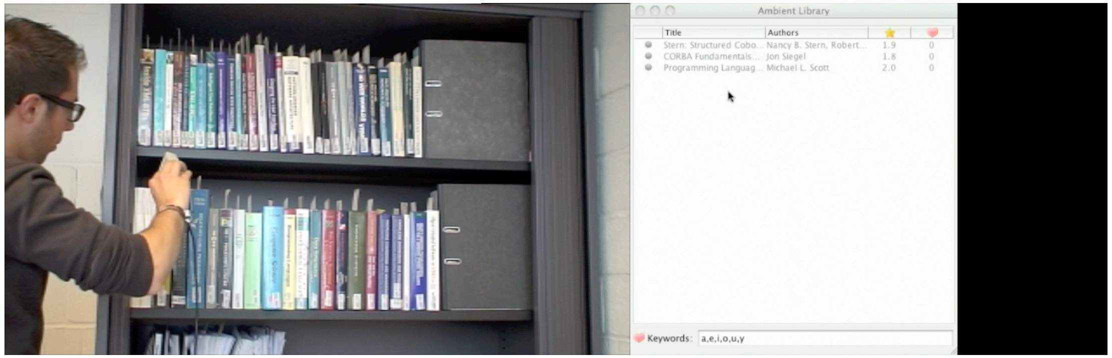
Multicores/Manycores

Supercomputers

Hardware Evolutions Require Novel Abstractions

Track #1: Mobile Systems





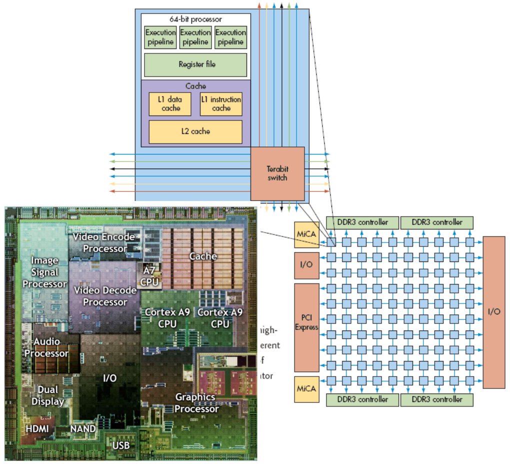
Ambient Library

	Title	Authors	★	♥
●	Stern: Structured Cobo...	Nancy B. Stern, Robert...	1.9	0
●	CORBA Fundamentals...	Jon Siegel	1.8	0
●	Programming Languag...	Michael L. Scott	2.0	0

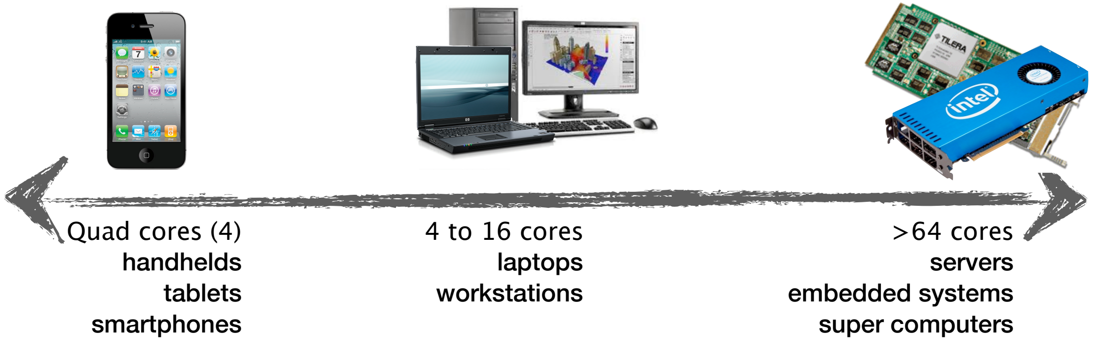
♥ Keywords: a,e,i,o,u,y

Track #2: Multicore & Co

New architectures



Multicore popularization

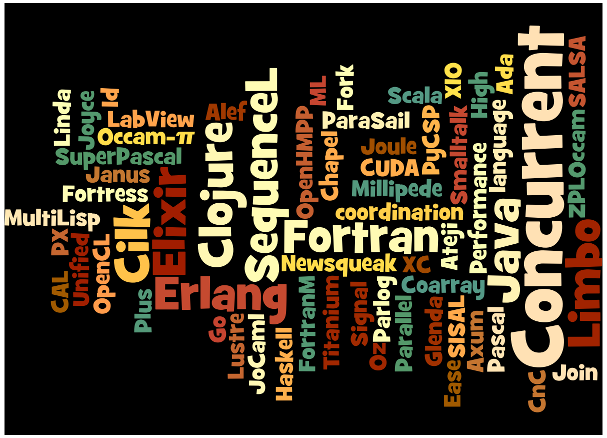


Quad cores (4)
handhelds
tablets
smartphones

4 to 16 cores
laptops
workstations

>64 cores
servers
embedded systems
super computers

Languages



We need languages/middleware that allow programmers to program concurrency more easily.

Track #3:Manycore & HPC

Supercomputing @ Home/Company

Traditionally:

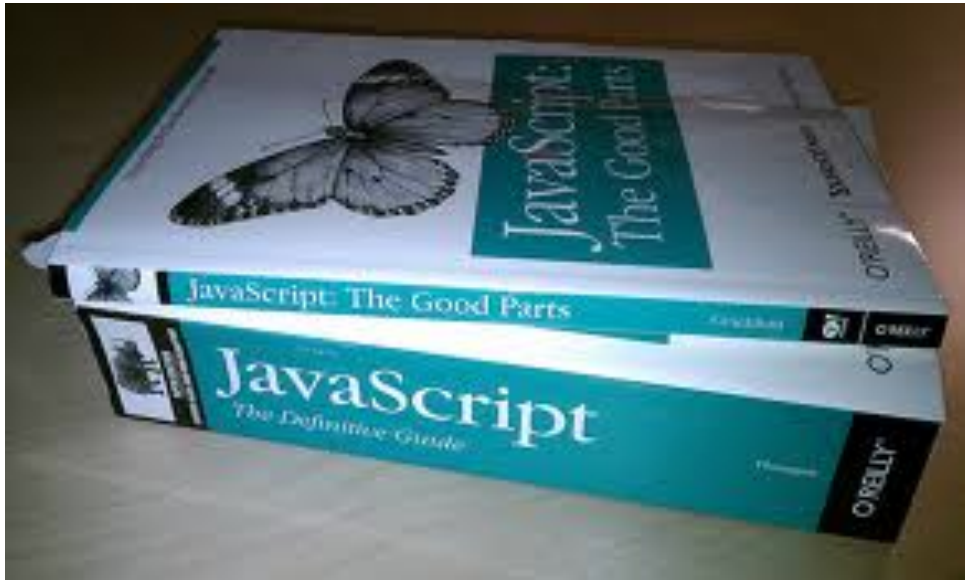
- Moving data = cheap
- Logic = expensive

HPC:

- Moving data = expensive
- Logic = cheap

We need languages/middleware that allow programmers to express how [“near” … “far”] data is w.r.t. a processing unit.

Track #4: Cloud Computing



ANGULARJS by Google

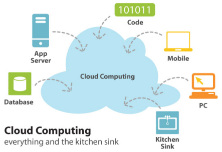
node JS

JavaScript, the next General Purpose Programming Language (c.f. Titanium)

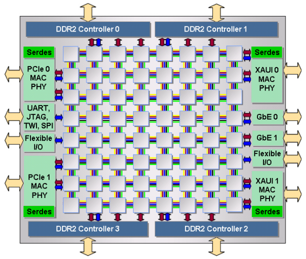
The browser *IS* the virtual machine.
Worldwide language-switch <u>is happening NOW</u>.

Take Home Message

Mobile Platforms



Cloud Platform



Multicore Manycore

Supercomputers

- Platforms are changing and getting more complex. Unfortunately, the code "is always right" !
- Programs are read far more often than they are written. Simple & powerful languages are the solution.
- But middleware & frameworks can be a good substitute.
- The Software Languages Lab is specialised in this process