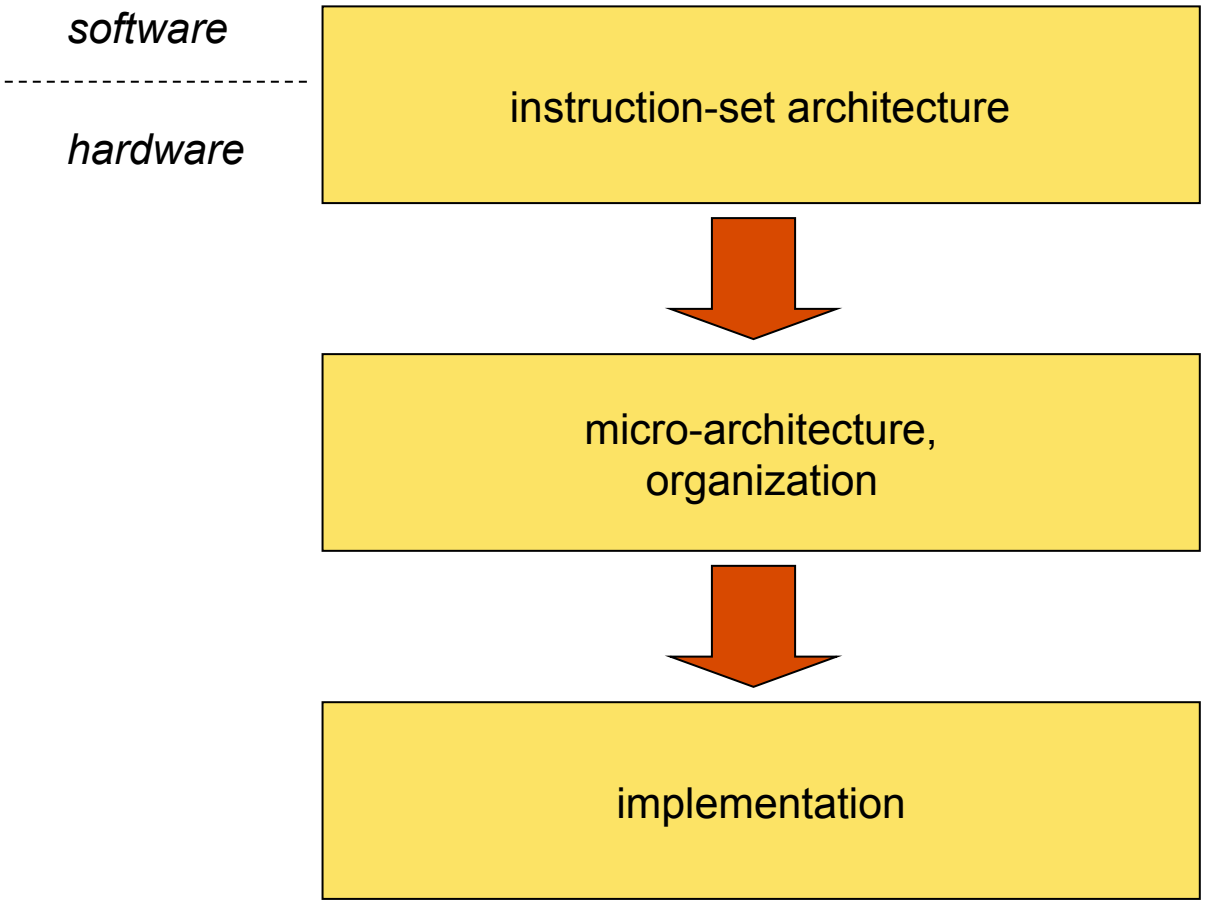
Lecture #0: Introduction

Parallel Computer Systems

Lieven Eeckhout

Academic Year 2014-2015
Ghent University

Computer Architecture



Instruction-Set Architecture

- ISA
- Instruction format/encoding, address modes, memory consistency model, etc.
- Interface between software and hardware
- Examples
  - IA-32, IA-64, Alpha, MIPS, PowerPC, etc.

Organization

- Or micro-architecture
- Internals of the processor
  - Functional units, in-order versus out-of-order execution, speculative execution, pipelinied execution, caches, branch prediction, prefetching, etc.
- Multiple micro-architectures are possible per ISA
  - IA-32: Pentium, Pentium Pro, Pentium III, Pentium 4, Core 2, Nehalem, Sandy Bridge, Ivy Bridge, Haswell
  - Alpha: 21064, 21164, 21264

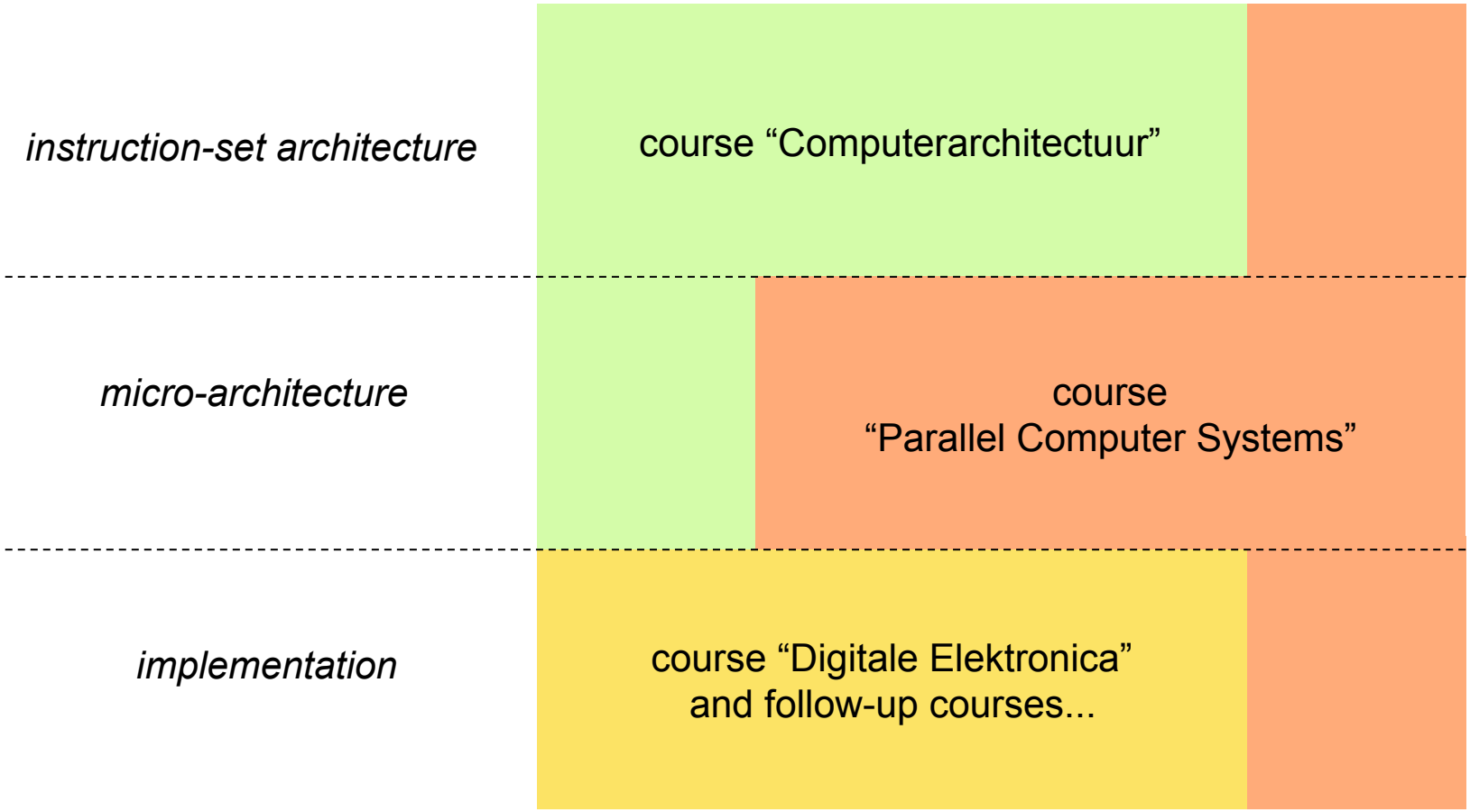
Implementation

- Physical implementation using transistors and interconnects
- Multiple implementations are possible for a given micro-architecture
  - Clock frequency, packaging, VLSI technology, etc.
  - For example: processor in 65nm versus 45nm
    - Currently: 22nm (14nm expected later this year)
  - Micro-architecture vs implementation: Intel's Tick-Tock

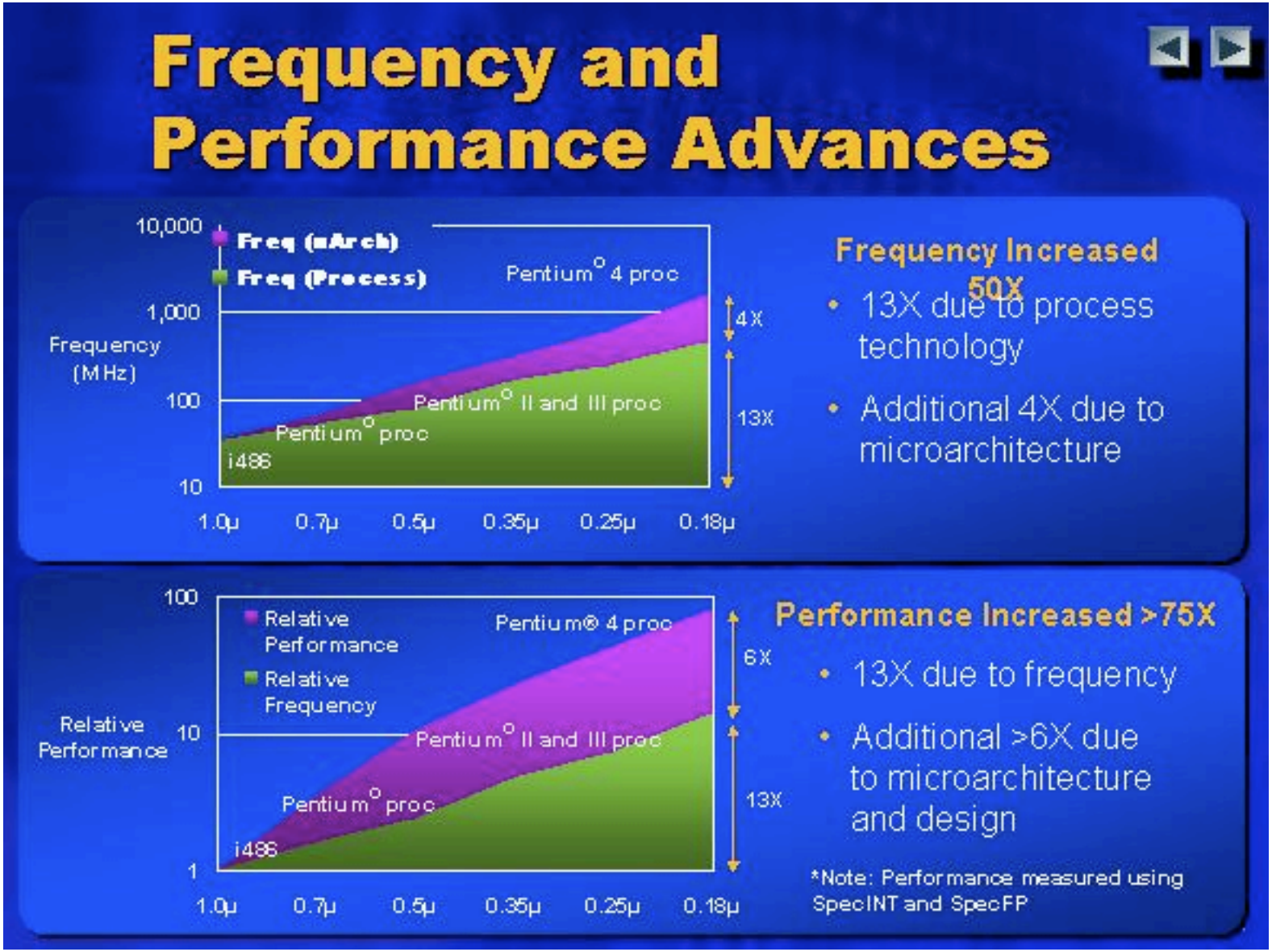
BUT...

- There exist interactions between the different layers
  - ISA has impact on micro-architecture and implementation
  - Micro-architecture has impact on implementation
    - Micro-architecture cannot be super complex!
  - Technology has impact on micro-architecture
    - For example: no. transistors, interconnects, power consumption, etc.

Courses in the curriculum

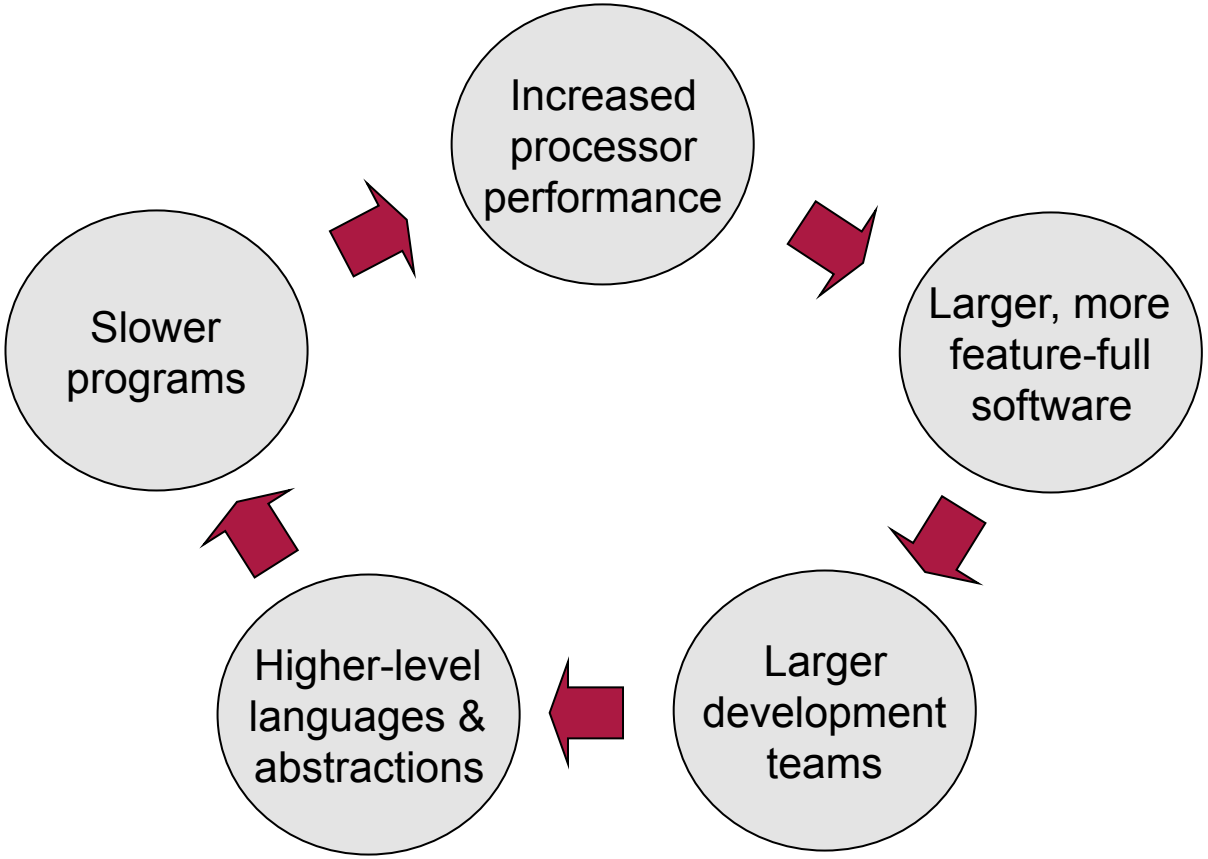


The importance of micro-architecture



[Justin Rattner, Intel, PACT'01] 1.0um Intel486 (1989) – 180nm Pentium 4 / Xeon (2000/2001)

The virtuous cycle



[1950 - 2005, according to James Larus; Mark D. Hill]

Micro-architecture design

- Optimized performance? given
  - ISA
  - no. transistors, technology, cost, power budget, etc.
- Via parallelism
  - At the instruction level
    - *instruction-level parallelism (ILP)*
  - At the thread level
    - *thread-level parallelism (TLP)*
  - At the memory level
    - *memory-level parallelism (MLP)*
  - At the data level
    - *data-level parallelism (DLP)*
  - At the request level
    - *request-level parallelism (RLP)*

Instruction-level parallelism: ILP

- Parallel execution of instructions from a single thread
  - In time: pipelined execution
  - In space: superscalar execution
- Executing multiple instructions per cycle
- (Nearly) All modern processors exploit ILP

Thread-level parallelism: TLP

- Parallel execution of instructions from *multiple* threads
  - instructions are (mostly) independent
- TLP is exploited in multi-processor systems and multi-threaded processors, for example
  - *shared memory multiprocessor (SMP)*: multiple processors (multiple chips) with shared memory
  - c*hip-multiprocessor (CMP)* aka *multi-core processors*: multiple processor core on a single chip (w/ shared memory)
  - *simultaneous multithreading (SMT)* or *Hyper-Threading*: multiple threads execute in parallel on a single processor core

Memory-level parallelism: MLP

- Memory wall: big gap between processor speed and memory speed
  - Processor: <1ns per clock cycle
  - Memory: access time 70ns
- MLP = parallel execution of multiple memory requests (from a single thread or from multiple threads)
  - latency hiding
- Examples: prefetching, non-blocking caches, etc.

Data-level parallelism: DLP

- Subword-parallelism
- Parallel execution of the same operation on different data elements
- SIMD: Single-Instruction, Multiple-Data
  - multimedia-extensions: MMX, AltiVec, VIS, 3DNow!, etc.
- SIMT: Single-Instruction, Multiple-Thread
  - GPU

Request-level parallelism: RLP

- Parallel execution of requests in a datacenter
- We're moving away from the desktop
  - to the cloud and mobile devices
- Interactive Internet services
  - web search, mail, social networking, etc.
- Embarrassingly parallel
  - Hundreds of thousands to millions of independent users

Course Material

- Slides
- Available at minerva.UGent.be
- Also through VTK
- Books & articles: if you're interested
  - Recommended reading
  - Course is based on several books/articles

Books: Available Online

http://www.morganclaypool.com/toc/cac/1/1

- "Processor Microarchitecture: An Implementation Perspective" by A. González, F. Latorre, G. Magklis, Morgan & Claypool Publishers, Dec 2010
  - OR: "Modern Processor Design: Fundamentals of Superscalar Processors" by J. P. Shen en M. H. Lipasti, Mc Graw-Hill, 2005
- "A Primer on Memory Consistency and Cache Coherence" by D. J. Sorin, M. D. Hill and D. A. Wood, Morgan & Claypool Publishers, May 2011
- "Computer Architecture Performance Evaluation Methods" by L. Eeckhout, Morgan & Claypool Publishers, 2010
- "The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Computers", L. A. Barroso and U. Hölzle, Morgan & Claypool publishers, 2009

Books (cont.)

http://www.morganclaypool.com/toc/cac/1/1

- “Multi-Core Cache Hierarchies” by R. Balasubramoniam, N. P. Jouppi, N. Muralimanohar, Morgan & Claypool publishers, May 2011
- “Fault Tolerant Computer Architecture”, D. J. Sorin, Morgan & Claypool publishers, 2009
- “Computer Architecture Techniques for Power-Efficiency”, S. Kaxiras and M. Martonosi, Morgan & Claypool publishers, 2008
- “Chip Multiprocessor Architecture: Techniques to Improve Throughput and Latency”, K. Olukotun. L. Hammond and J. Laudon, Morgan & Claypool publishers, 2007

Theory

- From scalar to superscalar execution (lecture #1)
- Out-of-order micro-architecture
  - Instruction stream (lecture #2)
  - Data stream (lecture #3)
  - Memory stream (lecture #4)
  - Performance analysis (lecture #5)
  - Impact of technology (lecture #6)

Theory (cont.)

- Superscalar in-order architectures (lecture #7)
- Multi-threaded execution
  - Fundamentals: synchronization, coherence and consistency (lecture #8 and #9)
  - Multithreading, transactional memory, GPU (lecture #10)
- Data center technology (lecture #11)

Practical exercises

- On black board
  - 5 sessions
  - By Stijn Eyerman
  - Excellent preparation for the exam!
- Behind the computer
  - 4 sessions
  - By Stijn Eyerman, Sam Van den Steen, Sander De Pestel, Almutaz Adileh
  - Will be quoted!

Agenda

Monday Sept 22	L #0
Friday Sept 26	L #1
Monday Sept 29	L #2
Friday Oct 3	PC #1
Monday Oct 6	E #1
Friday Oct 10	---
Monday Oct 13	E #2
Friday Oct 17	L #3
Monday Oct 20	L #4
Friday Oct 24	L #5
Monday Oct 27	L #6
Friday Oct 31	PC #2

Monday Nov 3	L #7
Friday Nov 7	---
Monday Nov 10	---
Friday Nov 14	L #8
Monday Nov 17	L #9
Friday Nov 21	E #3
Monday Nov 24	L #10
Friday Nov 28	PC #3
Monday Dec 1	E #4
Friday Dec 5	L #11
Monday Dec 8	E #5
Friday Dec 12	PC #4

Exam

- First trial (Jan - Feb)
  - quoted computer exercises: 15% of total
  - written exam with course material
- Second trial (Sept)
  - idem

Questions about the course

- Website
  - minerva.UGent.be
- During the lectures
  - is highly appreciated!
- Via email
  - Lieven.Eeckhout@UGent.be