POLITECNICO DI MILANO

DIPARTIMENTO DI
ELETTRONICA,
INFORMAZIONE
E BIOINGEGNERIA

ASAM
Advanced Software Architectures and Methodologies at Politecnico di Milano

Luca Mottola and the ASAM team

Outline

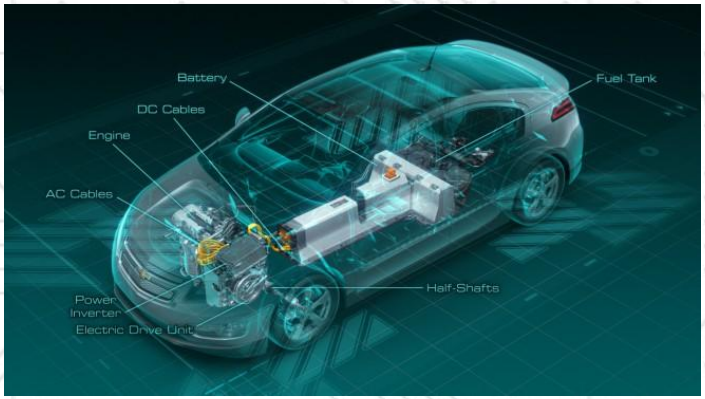
- Introduction to ASAM
- Research in ASAM
- Studying ASAM

Outline

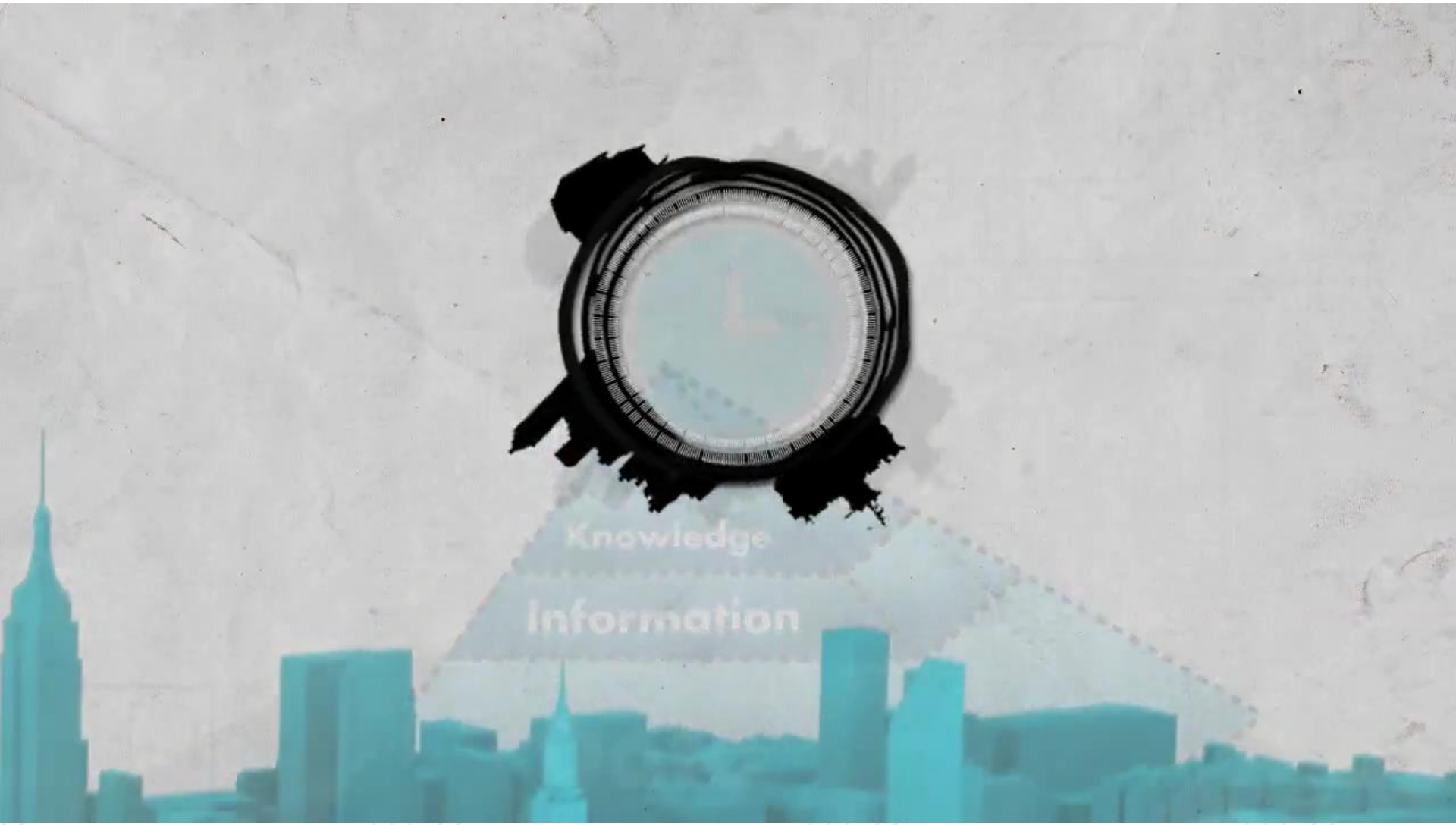
- Introduction to ASAM
- Research in ASAM
- Studying ASAM

ASAM—*software everywhere*

- Our society depends on and relies on **software**
  - software is a key enabling technology of our society
  - nothing in the world works without software
- Do you drive a car?
  - "cars run on code"
    - high-end cars contain close to 100 million LOC
    - running on 70-100 microprocessor-based electronic control units
- Do you travel on planes?
  - "a plane is software with wings"
- Do you have a smartphone?
  - besides talking, you
    - interact in social networks
    - use it for travel instructions
    - …



ASAM—*software everywhere*



ASAM

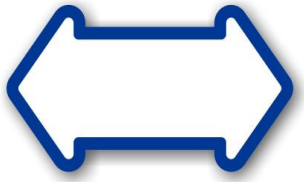
- Software is **key** and **pervasive**, but still **fragile**
  - Ever increasing size (ultra-large systems) and criticality
  - Decentralized control
  - Inherently conflicting requirements
  - Continuous evolution and deployment
  - Ability to self-organize, self-adapt, and self-repair
  - Erosion of the people/system boundary
  - Dependability despite normal failures
  - New paradigms for development and acquisition
- **Goal:** studying, understanding, and improving the underlying principles and techniques of software development

Outline

- Introduction to ASAM
- Research in ASAM
- Studying ASAM

Research in ASAM

Service & Cloud Computing



Middleware

Cyberphysical Systems

Applications

Software Development for Self-adaptive Systems

Formal Specification and Verification of Critical Systems

Self-managing software

- Long term vision
  - continuously-running systems
  - emerging behaviors
  - self-*
- Current focus
  - self adaptation to environment changes
  - non-functional requirements (quality of service)
    - performance, reliability, energy consumption, costs
- Research streams
  - models and verification at run time
  - safe dynamic reconfigurations

Formal design of critical systems

- Critical systems: malfunctions can lead to big losses
  - in terms of money or, worse, human lives
  - airplanes, trains, power plants, production systems ...
- **Goal**: guarantee that system design is correct
- Research focus: develop innovative techniques for modeling and verification of system designs
  - based on sound mathematical models
  - supported by tools
  - usable in practice by domain experts
    - vehicle/plant/... designers
- Target: **real-time, hybrid** systems

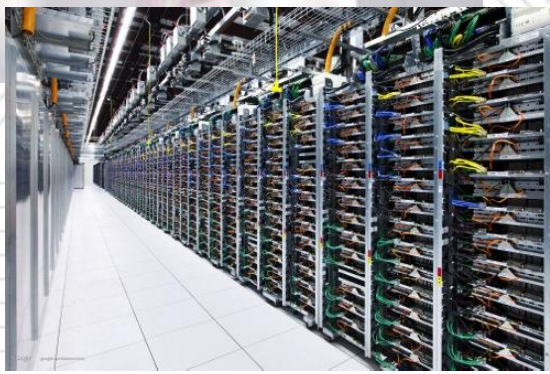
Cyberphysical systems

- Engineered systems deployed in the **physical world**
  - Internet of Things, Wireless Sensor Networks, Drone Sensor Networks
- Research focus
  - programming abstractions
  - service orientation
  - communication support
  - verification and validation

- **Real-world deployments** of prototype systems
  - heritage and archeological sites, road tunnels, energy-efficient buildings, …
- Physical lab about to open for experimentation
  - …more during the openLabs session at the ASAM booth!

Services and cloud

- **Software services**: running software components made available for use by external consumers (Software as a Service – *SaaS*)
- **Cloud computing**: set of storage, computational resources, middleware services available for use (Infrastructure/Platform as a Service – *IaaS/PaaS*)
- Software services + cloud computing constitute a stack of services offered at different levels of abstraction
- Research questions
  - How to evaluate and optimize the performance of IaaS+PaaS+SaaS?
  - How to monitor the services at the various levels of abstraction?
  - How to design applications using services from multiple clouds?
  - Can we see devices in the Internet as new types of services (Things as a Service – TaaS)?



Middleware for self-* systems

- **Middleware** is the layer between the distributed application and the network
  - appropriate abstractions offered at the middleware layer may help in building adaptive software
  - hiding the complexity coming from distribution
- Research focus
  - finding the appropriate middleware models
    - e.g. ,routing messages based on their content, implementing an event-based architecture
  - but also focus on actually implementing them…
    - using parallel (CPUs/GPUs) and distributed programming to improve performance
  - …and testing them in “realistic” settings
    - large scale testbeds, clusters

Compiler technology

- **Goals**:
  - improving compiler development and maintenance cycles with better data structures (intermediate representations)
  - automating performance portability across different kinds of parallel machines
  - automating management of non-functional features (e.g., adaptivity)
  - improving effectiveness through parameter tuning
- **Techniques**: although compiler technology builds over automata theory, most advanced techniques involve specialized algorithms on graph and trees
- **Occupational outcomes**: compiler engineers are a scarce resource, and typically employed by large semiconductor companies or SME specialized in compilers

Applications

Concretely demonstrate the research in action!

Electric vehicle sharing

Smart spaces and mobile apps

Aerial mapping of archeological sites

Nearly-zero energy buildings

VELUX® lab

Outline

- Introduction to ASAM
- Research in ASAM
- Studying ASAM

Studying ASAM

- Master degree in Engineering of Computing Systems
- 120 cfu on two years (four semesters)
- 100 cfu in courses, 20 cfu for Master Thesis
- Three main Methodological Areas:
  - IT Management and Applications (e.g, business information systems)
  - **Software Methodologies** (e.g., distributed systems)
  - Hardware Architectures (e.g., embedded systems)
- Software methodologies have a set of fundamental courses also in the ASAM area and a wide choice of electives, possibly aggregated into specialized tracks

Fundamentals in ASAM

Two key courses:

Software Engineering 2 (compulsory)

Learning goals: To be an advanced software engineer, up-to-date with the most modern software technologies

Core topics: Software life cycle; requirements engineering; software architectures; specification, verification and validation

Principles of Programming Languages (suggested)

Learning goals: Learning concepts and paradigms of programming languages, to master or even design a new language

Core topics: Main Paradigms: object-oriented, functional, logic; concurrent programming; domain-specific languages

Electives

Elective Courses in ASAM sub-areas:

Software Architectures and Cloud

- Distributed Systems
- Middleware Technologies for Distributed Systems

Advanced Software Development

- Distributed Software Development
- Design and Implementation of Mobile Applications

Software Verification and Validation

- Formal Methods For Concurrent and Real Time Systems

Compiler Technologies

- Code Transformation and Optimization

Tracks

Elective Courses in the ASAM area can be combined with courses in other areas for the following tracks:

- **Pervasive systems**
- **ICT for industrial applications**
- **Interactive applications**
- **Internet engineering**

But they also provide methodological support for other tracks, e.g.:

- **Networked enterprises and services**
- **Artificial intelligence**
- **Bioinformatics and e-health, …**

Employment

- Software engineers are highly requested by the market
  - http://finance.yahoo.com/news/10-of-the-best-jobs-for-the-future-191256480.html
  - http://money.usnews.com/careers/best-jobs/rankings/the-100-best-jobs
  - http://money.cnn.com/magazines/moneymag/best-jobs/
  - http://www.careercast.com/jobs-rated/2012-ranking-200-jobs-best-worst

POLITECNICO DI MILANO

DIPARTIMENTO DI
ELETTRONICA,
INFORMAZIONE
E BIOINGEGNERIA

Contacts

Luca Mottola
<luca.mottola@polimi.it>