New R&I Challenges in Software Engineering

Lutz Schubert, University of Ulm

Keith Jeffery, Consultant

traditional software engineering

TURING & VON NEUMANN

- oriented towards single core, sequential processors
- sensible decision at the time, now a major problem

data access is free (< local data)

processing is costly

data is not shared

single execution thread

abstraction through stacking

one processor fits all

traditional software engineering

A LOT HAS HAPPENED

- the model is built for scientific computing
- future use cases in the “end user market” were not considered
- the growth in performance need was unforeseen

~~data access is free (< local data)~~ **MEMORY WALL, INTERNET**

~~processing is costly~~ **MEMORY WALL**

~~data is not shared~~ **INTERNET, CLOUDS**

~~single execution thread~~ **MULTI-CORE PROCESSORS**

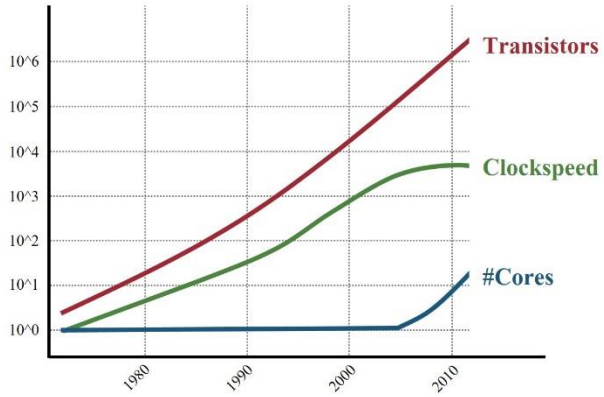
~~abstraction through stacking~~ **PERFORMANCE**

~~one processor fits all~~ **ACCELERATORS, ASICs, FPGAs …**

recent changes (environment, market)

A LOT HAS HAPPENED

and we are aware of these changes and problems for 30+ years, but always dealt with it by adapting rather than rethinking the classical models



Move from

- **Single core to multi-core / single-thread to parallel**
  - No more processing performance
- **Memory wall**
- **Local / centralised to distributed (internet, clouds)**
- **Non-shared to shared data, multiple data centers**
- **Mobility and long distance communication**
- **Homogeneity to increased heterogeneity and form factors**
  - Resource types
  - Usage / users
  - Operating systems / execution environments
- **Strong to weak computers (embedded, mobile)**
- **Few internet users to billions of connected devices**

market relevant aspects

MONEY

with strong competition between providers and offers, the IT market is dictated by the end user requirements and expectations.

- **User satisfaction**
- **Personalisation**
- **Availability in a mobile world**
- **Portability across devices and OS**
- **Performance / more complex applications**
- **Integration with other applications / interoperability**
- **Data mining (big data / information gathering)**

analysis / impact

MORE OF THE SAME?

all factors are in competition and conflict with each other (e.g. performance vs. non-shared vs heterogeneity)

- **Developers cannot cater for all aspects on their own**
- **The framework (including compiler) must become smarter and more efficient, but to do this, it is lacking essential information and has to deal with the conflicting parameters (what to optimize for)**
- **Putting it all on the user in the form of SLA specifications and non functional requirements is thereby no long-term solution: neither does she/he know what is possible, nor what the impact is**
- **We need more information from the developer, but this should not be a burden on her or him**

consequences / approach

THE WAY FORWARD

needs to move away from the traditional centralised model

- **Move from compute- to data-centric programming**
  - Rather repeat a calculation than store
  - Know where which data is needed when
  - Exploit asynchronicity, assess communication cost
  - Reconstruct data and computation to cater for delays
  - Data locality/movement/caching
  - Data partitioning and distribution (performance/security)

consequences / approach

THE WAY FORWARD

needs to move away from the traditional centralised model

- **Move from centralised to “volatile” computing and storing:**
  - Co-location of code and data is no longer guaranteed – both need to be freely localisable
  - End-to-end / total time is important, not individual ones

consequences / approach

THE WAY FORWARD

needs to move away from the traditional centralised model

- **The user cannot have or provide every knowledge**
  - With the degree of heterogeneity and potential execution flexibility (placement, timing, form of execution)

consequences / approach

TOWARDS I3

an increased abstraction *without performance degradation* is necessary to improve interoperability, usability, portability …

requires strong rethinking of traditional methods

- **Most computation is about information, no longer about data**
  - Data doesn't always have to be exact
  - Data doesn't always have to be available
  - Computation doesn't always have to be complete

  ➔ *information aspect of the system*

consequences / approach

TOWARDS I3

an increased abstraction *without performance degradation* is necessary to improve interoperability, usability, portability …

requires strong rethinking of traditional methods

- **The “right” choice of deployment, configuration, computation etc. depends on too many factors**
  - Goes beyond just NFR and SLA and depends on use case
  - Personalisation is not just a question of the interface and the data, but of the whole software behaviour stack
  - Interoperability and portability cannot be posed on the developer

  ➔ *incentive of the system*

consequences / approach

TOWARDS I3

an increased abstraction *without performance degradation* is necessary to improve interoperability, usability, portability …

requires strong rethinking of traditional methods

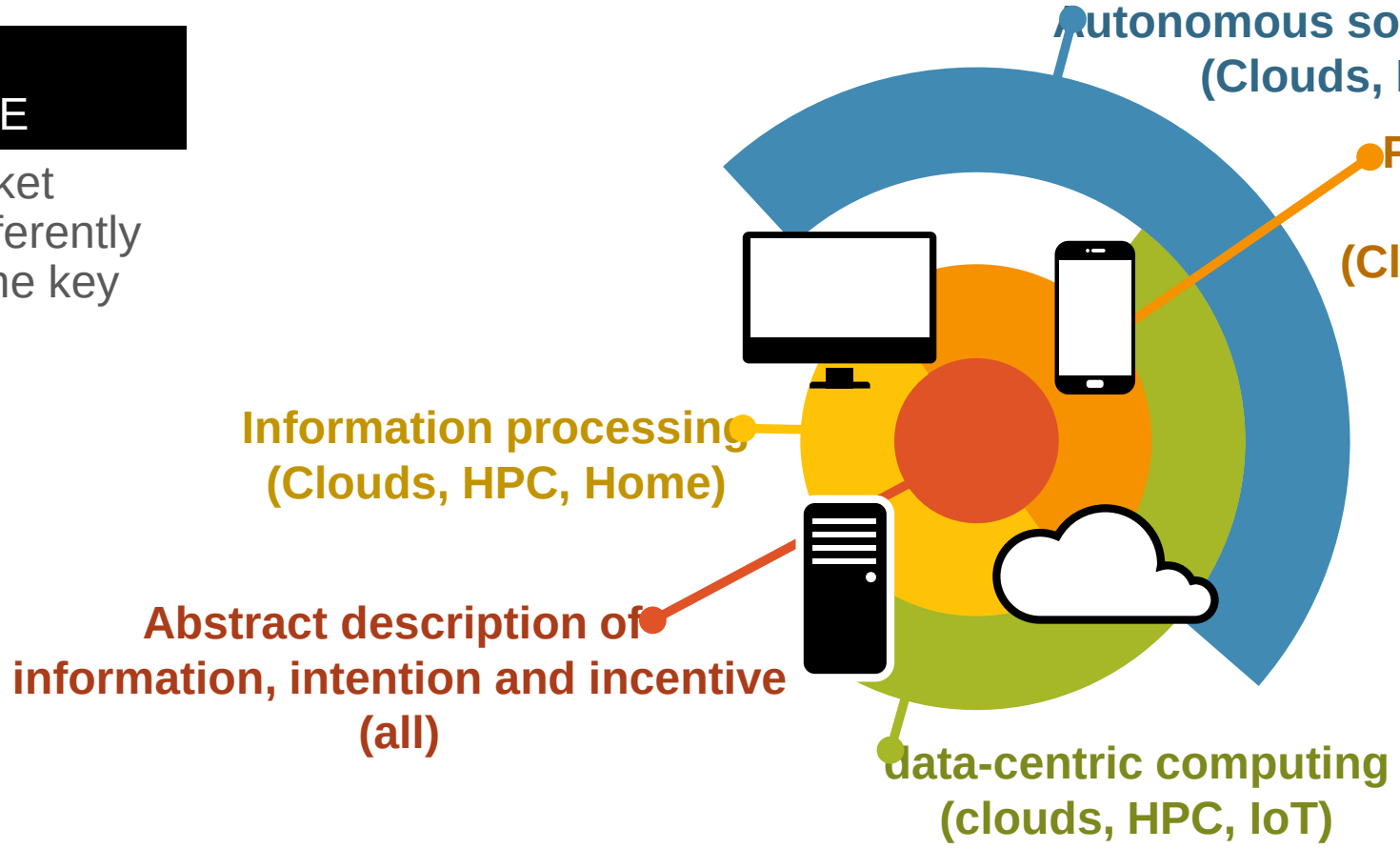
- **The complexity of software increases so much that maintenance and adoption become near impossible**
  - New abstractions are needed that encode the *intention* of the program, much more than its behaviour
  - Must give enough freedom for the execution environment to adapt the actual code
  - Must be close to the hardware, not adding another abstraction layer leading to incompatibilities

➔ *intention of the system*

putting it in context

AREAS OF RELEVANCE

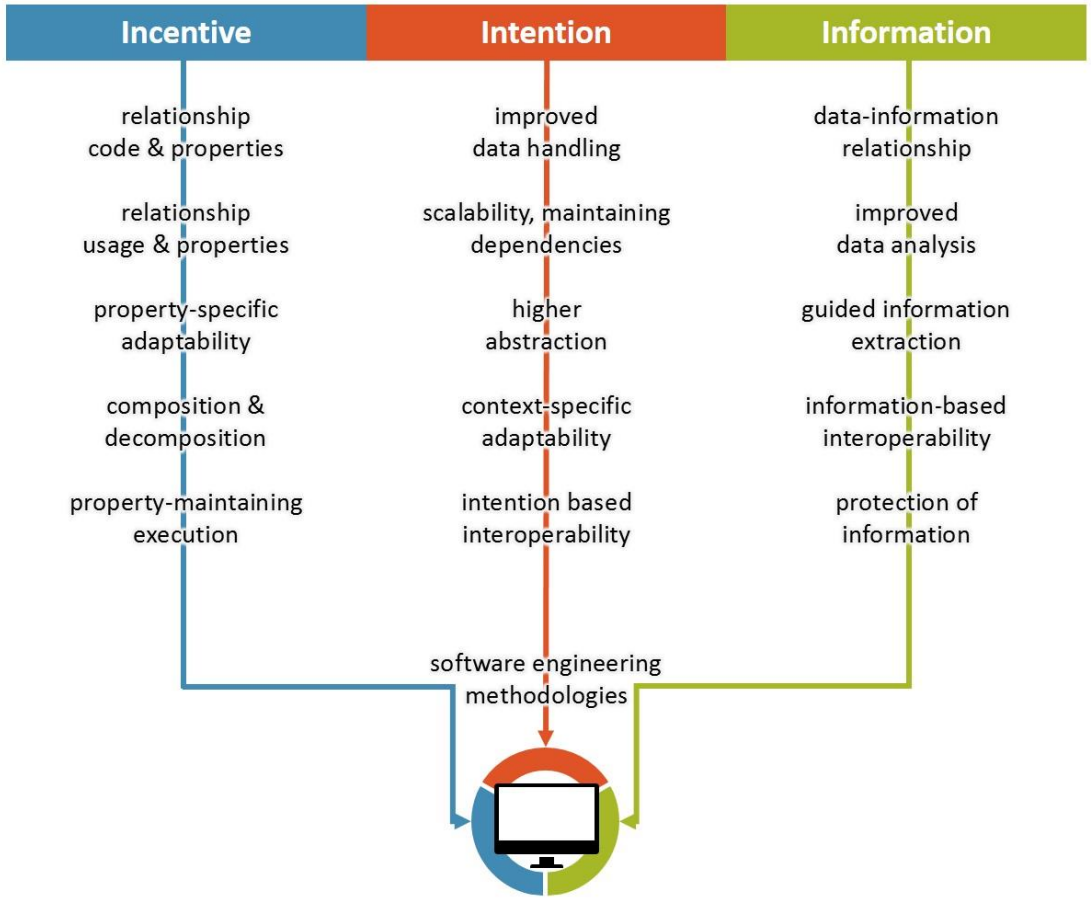
different market areas are differently affected by the key aspects



roadmapping

TOWARD I3

steps towards a richer model of software engineering



More details:

Lutz Schubert, Keith Jeffery, "New Software Engineering Requirements in Clouds and Large-Scale Systems", IEEE Cloud Computing vol. 2 no. 1, p. 48-58, Jan.-Feb., 2015

Keith Jeffery, Lutz Schubert, “Complete Computing: toward information, incentive and intention”, EC report. Available at: http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?action=display&doc_id=6775