Information system engineering: the end of “bricoleurs”

The impact of real time, big data and massive parallelism on IT project management

Jacques Printz, CSDM 2014
Professor Emeritus at Cnam
Information engineering

Processes everywhere

Human machinery → BUSINESS processes

Technical engineering → INDUSTRIAL processes

Percolation

Digitalization of all or parts of these processes

Integration of everything through information

System of systems engineering
Interactions between systems and people

Symbiotic relationships

End-users

Engineering

Usages
[State of needs]

Feedbacks

R&D
[State of the art]

Knowledge integration

Open questions

High pressure of the customer

Response time

System as a whole

Response time

Response time

Agile engineering → From WHAT to HOW

Development
Maintenance
Operations

High pressure of the management

Any interaction is becoming ⇨ Real Time

©2014 J.Printz / Information system engineering project management
Human complexity

"Nothing technical here: just choices, decisions, cooperation, soft or hard power, capacity and maturity, authority, etc."

The heart of information system project management

Classical complexity
The "work horse" of IS engineering

Building blocks
→ Transaction processing

That?

Or that?
Layout of the classical TP

Atomic Transaction

\[ \rightarrow \text{with ACID properties} \]

The quantum of information processing

Atomic
Consistency
Isolation
Durability

Rule of thumb:
\[ \text{A piece of code not larger than 1 or 2 thousands lines of code} \]

A coherent memory state

Restore the initial state

Mainly data referenced by the TP

Take a snapshot of the initial state

Initial State

begin

Transaction

commit

roll back

Transaction failed

Transaction completed

Next TP
Examples: Transactional situations from the end-user point of view

Typical long transactions based on short ACID transactions
Evolution of transaction programs concept

70-80
Mainframes, sequential processing
- Classical TP

90-2000
Distributed processing
- Client/Server TP

2010-...
Parallel processing
- Multi-core parallel programming

Short ACID TP
- Fully automatic recovery mechanism

Long TP
- New mechanisms like Autonomic computing are needed

Micro TP
- Multi-core is a black-box

Complexity wall
- New tools and methods are needed

Just evolution
The multicore revolution

Tera-100 Computer

Blue Gene Computer

KALRAY MPPA MANYCORE

©2014 /J.Printz /Information system engineering project management
Organize the parallelization

Processes synchronization & communications

Thousands of threads

N Physical Processors
[but many more virtual processors]

Semaphores

Mail boxes

Priority

Timers

Locking mechanism

©2014 /J.Printz / Information system engineering project management
From one sequential flow to many ...

Sequential flow

Rule: 1 thread → 1 processor
- Can be done by hand

Data processed by that flow

Parallel flow

Rule: 1 thread → Many processors
- Just impossible to do it by hand

Data processed by that flow

Major program transformation to adapt the sequential code to the many cores execution flow especially the data
What is the real status of YOUR code?

Patches

Everything I know about software I learned from spaghetti bolognese

NO MORE SPAGHETTI CODE

Managing complexity
How to transform bad code?

No magic behind!

Just engineering
A new approach for information engineering

David Patterson, Professor UC Berkeley:

“We desperately need new approach to HW & SW based on parallelism since industry has bet its future that parallelism works”

New tools are needed to extract the natural parallelism hidden in data. Data is the new code.

The solution has to model data semantic, explore different data organization, choose one based on simulation result maximizing resource usage and allowing to organize the parallelism in a reversible way to manage the unavoidable programming errors.
The way [道] to system architecting

System Architecting [requirement engineering]

Functional Architecture [Business Computing]

Organic/Physical Architecture

SysML, BPMN/BPEL, WSDL, ...

The singularity is here!

Automated Mapping from the System view → Components view

DSL

Usual Languages : C/C++, Java, ... + Compilers

Final mapping on the available cores for execution

Programs development validation & verification can no more be done by trial & errors, but only by simulation on the functional layer, using models
Facing complexity: layered architecture

Microsoft Office  Internet – Web  Information System  Scientific computing  Signal & Image processing  ...

Applications

Application Programming Interface – Library / Middleware

Operating System API (programming languages)

Operating System

Network stack

Interface Hardware/Software

hardware Functions

OS « Boot »

Machine Architecture and language(s)

Microprocessor architecture and hardware « programming » - Buses

Nanoworld

Physical Architecture (cache memory, pipe-line, …)

Circuits Architecture and wiring (ECC)

Transistors and Silicon – Atoms & Electrons

Deterministic World

Recovery mechanism

Non deterministic

Physical Laws

Recovery mechanism

Quantum world

Recovery mechanism

Silicon cristal

©2014 /J.Printz /Information system engineering project management
Facing complexity: Visible or hidden?

End-users

Business API / Plates-forms & infrastructures

Business processes

- Specialized Applications and processes automation
- Generic Applications – Packages

Visible part
[Added value]

Patrimonial
Business software

Business API

Hidden reality

“Black box”

- Network
- Operating system
- Virtualization (Cloud / multi core computing, )

API associated with programming languages

Middleware and system services → OLTP, SGBD, CORBA, ... Administration, ...

Middleware Integrated like Internet/W3C, ESB, ...

Events ➔ Dynamical control

Dynamical control

• Network
• Operating system
• Virtualization (Cloud / multi core computing, )

Couches Télécoms

OS

Hardware / QOS

Circuits / QOS

Facing complexity: Visible or hidden?

End-users

Business API / Plates-forms & infrastructures

Business processes

- Specialized Applications and processes automation
- Generic Applications – Packages

Visible part
[Added value]

Patrimonial
Business software

Business API

Hidden reality

“Black box”

- Network
- Operating system
- Virtualization (Cloud / multi core computing, )

API associated with programming languages

Middleware and system services → OLTP, SGBD, CORBA, ... Administration, ...

Middleware Integrated like Internet/W3C, ESB, ...

Events ➔ Dynamical control

Dynamical control

• Network
• Operating system
• Virtualization (Cloud / multi core computing, )

Couches Télécoms

OS

Hardware / QOS

Circuits / QOS
Conclusion

- Information engineering, through the pressure of ultra high complexity at all levels, requires now the same rigor as any other traditional engineering.
- There is no alternative to modeling techniques to master complexity.
- If we don’t organize complexity, the complexity will kill us ... like in the Babel tower story !!!