

Leveraging business workflows in distributed systems research for the orchestration of reproducible and scalable experiments

Tomasz Buchert Lucas Nussbaum
INRIA, LORIA, Université de Lorraine, CNRS



What is a distributed system?

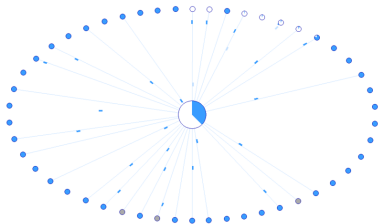
A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.

Leslie Lamport

What is a distributed system? (2)

Examples of well-known distributed systems:

- DNS,
- BitTorrent,
- Gmail.



What is a distributed system? (3)

Distributed systems are:

- complex,
- erroneous,
- difficult to control,
- nondeterministic.

Moreover, they:

- span many networks,
- span many geographical locations,
- consist of thousands of machines.



Can we make better BitTorrent, Gmail, etc.?

Solution: **experimentation**.

The “classical way” is difficult, but other solutions exist:

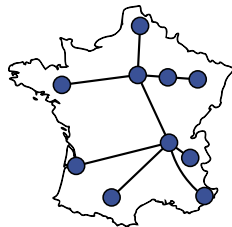
- simulation,
- benchmarking,
- emulation.

	Real program	Modeled program
Real platform	Standard approach	Benchmarking
Modeled platform	Emulation	Simulation

Many solutions exist (hardware and software based):

- Grid'5000, Emulab, PlanetLab,
- OMF, Expo, Plush, g5k-campaign,
- ... among the others.

How can one **evaluate** or **grade** them?



Our final goal

To improve the research, the experimentation framework has to:

- improve **descriptiveness of the experiments**,
- handle **unexpected**, but **inevitable** errors,
- ensure **scalability** of experiments,
- ensure **reproducibility** of the results.

In the end, we want to

**improve experimentation on testbeds
like Grid'5000 and PlanetLab.**

- Introduction
- Issues with distributed systems research
- Approach – a framework to control experiments
- Our contributions:
 - ① Identify requirements for an experimental framework
 - ② A new approach based on *Business Process Management*
- Conclusions

Let's define properties an experimentation framework **must have**.
That way we can evaluate existing and future solutions.
We distinguished 12 features in 3 categories.

To improve the process of designing the experiment we need:

- descriptiveness,
- modularity,
- reusability,
- maintainability,
- support for common patterns.

Required features of experiment execution are:

- snapshotting,
- error handling,
- integration with lower-level tools,
- human interaction.

Finally, to monitor the experiment, we need:

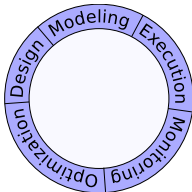
- monitoring,
- instrumentation,
- data analysis.

The partitioning of the properties is not arbitrary.

The three categories:

- design,
- execution,
- monitoring,

map to phases in *Business Process Management Lifecycle*.



None of existing solutions covers all our requirements.

Can we provide them all **at once**?

Our approach is based on *Business Process Management* (BPM).

Traditionally, Business Process Management is about:

- understanding how an organization works,
- modeling its activities in a language of **workflows**,
- **executing** processes and **monitoring** their progress,
- identifying ways to **improve activities** in an organization,
- redesigning processes to make them:
 - cheaper,
 - faster,
 - less defective.



Toyota Corporation is (2010):

- the largest producer of cars world-wide,
- the most profitable car-making company,
- well known for its socio-technical system of practices.

The fundamental objective is to eliminate **waste** or **muda** (無駄): overproduction, waiting, transportation, processing, stock at hand, movement and defective products.



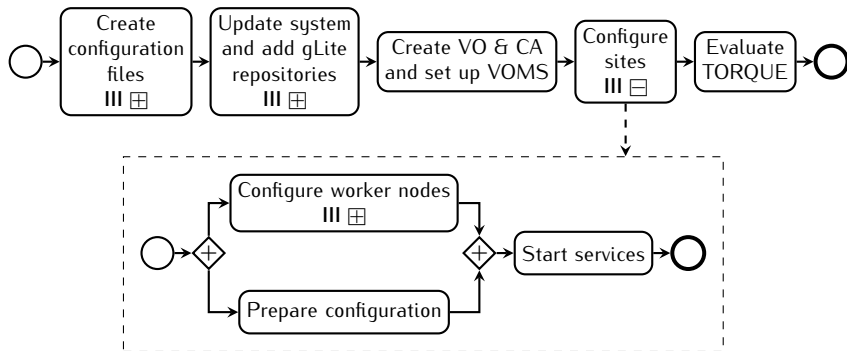
BPM and experimentation

There is an analogy between **production** and **experimentation**:

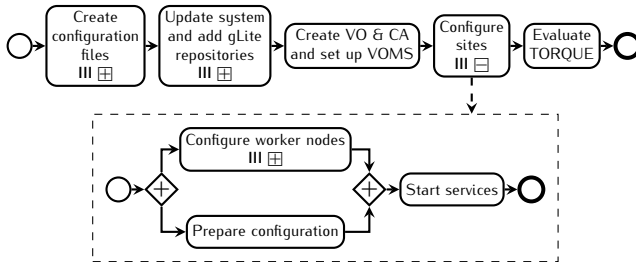
- a car \longleftrightarrow a finished experiment,
- an assembly line \longleftrightarrow execution of an experiment,
- parts of the assembly line \longleftrightarrow parts of an experiment,
- eliminated waste (muda) \longleftrightarrow efficiency and reliability.



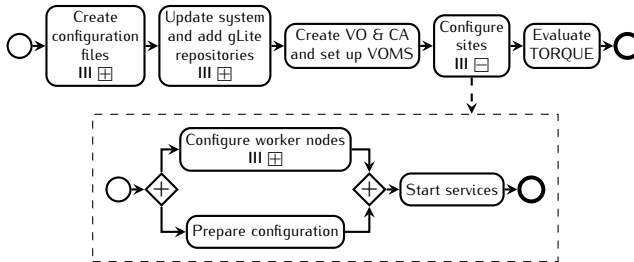
Example of a workflow



How BPM fulfills the requirements?



How BPM fulfills the requirements?



Design

Descriptiveness

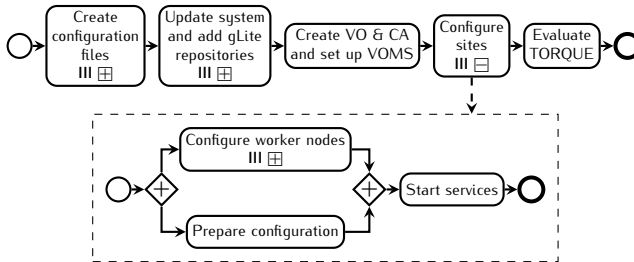
Modularity

Reusability

Maintainability

Support for common patterns

How BPM fulfills the requirements?



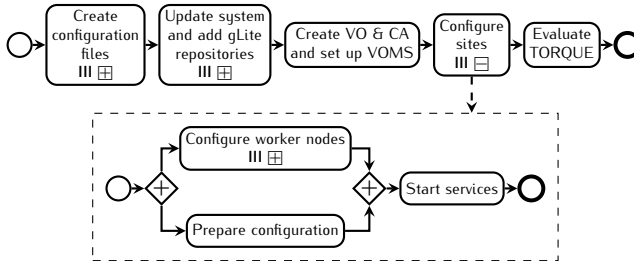
Design

Descriptiveness
Modularity
Reusability
Maintainability
Support for common patterns

Execution

Snapshotting
Error handling
Integration with lower-level tools
Human interaction

How BPM fulfills the requirements?



Design

Descriptiveness
Modularity
Reusability
Maintainability
Support for common patterns

Execution

Snapshotting
Error handling
Integration with lower-level tools
Human interaction

Monitoring

Monitoring
Instrumentation
Data analysis

To sum up, we:

- introduced problems of research in distributed systems,
- described our contributions:
 - ① requirements for an experimentation engine,
 - ② a new approach to experimentation based on BPM,
- showed how our approach meets the defined requirements.

In the near future, we plan to:

- implement a *workflow-based experiment engine*,
- evaluate it using a set of well-chosen experiments,
- find more analogies between BPM and experimental science.

<http://www.loria.fr/~buchert/>

Thank you for your attention.