# Using business workflows to improve quality of experiments in distributed systems research

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

#### 1 Problem: unreproducible experiments

Research in distributed systems suffers from many difficulties and technical issues that threaten the quality of results:

- large-scale, complex and unreliable communication networks,
- software and hardware errors,
- non-isolated experimental environment,
- bad methodology and lack of scientific rigor,
- incomplete description of the experiment.

With ever-growing complexity of computer systems, it becomes more evident.

## 3 Approach: leveraging business workflows

Our method consists in harnessing methods of Business Process Management:

- the experiment is expressed using a domain specific language (DSL) using experiment patterns and patterns identified in the domain of BPM; the experiment description maps to workflow representation,
- the functional parts of the experiment, or activities, implement actions that realize a concrete and well-defined goal (e.g., installation of a package),
- finally, an **experiment engine** executes the workflow and takes care of communication, data collection, monitoring, exceptional situations, etc.



### 2 Goal: make experiments reproducible

The experiment needs a high-level description that:

- is abstract and descriptive,
- is modular and composed of reusable parts,
- can be easily maintained and improved,
- uses common patterns instead of reinventing them,
- integrates analysis of experimental results,
- handles failures and timeouts transparently.

## 4 Benefits: improved quality of the experiment

- Understanding:
- better structure and ease of understanding,
- unobvious optimization opportunities,
- improved maintainability of the experiment.
- Modularity:
- reusable components of the experiment,
- well-defined and loosely coupled components,
- logical separation between software layers used in the experiment.
- Monitoring:
- the state of experiment constantly monitored,
- automatic recovery from failures and exceptional situations,
- timing information used to find bottlenecks and performance issues.
- Workflow patterns:
- well-tested, building blocks of the experiment,
- parallel execution of activities,
- deadlocks avoided by implicit synchronization.

### 5 Example: description of an experiment involving Amazon EC2 instances

```
engine process do
      forall nodes do |n| # ......
          run :install software, n
          <u>run</u>:check node, n
                                                                         Instal
                                                                        software
                                                    Reserve
                                                   EC2 nodes
                                                                         Check
                                                                         node
          result = <u>run</u> :experiment, nodes
          store : results, result
                                                     Deploy
          prec = precision :results
          return on (prec < 0.1)
                                                   (in parallel)
      run :analyze
                                                                         Run the
                                                                        experiment
14 end
                                                   Experiment
                                                                        Store the
result
      results = data :results
                                                    Analyze
                                                                        Precision
      value = results.average
                                                   the results
                                                                        reached?
      delta = results.confidence
      puts "The result is #{value}."
      puts "The conf. interval is #{delta}."
      results.save "results.txt"
23 end
```







