

Experimentally driven research white paper

Version 1, April 2010

On the existence of experimentally-driven research methodology

This white paper attempts to scope the area of *experimentally driven research* by proposing a *methodology* and related terminology that should help research in the ICT domain to understand how this methodology can be applied.

This white paper is intended as the first step in a controversial debate that should take place in the community of FIRE projects (Future Internet Research and Experimentation) and other interested stakeholders.

General information: <http://www.ict-fireworks.eu>

Contact: info@ict-fireworks.eu

See Annex A for a list of contributors

Editor: Anastasius Gavras, gavras@eurescom.eu

1 Introduction

FIRE aims to create a “research environment for investigating and experimentally validating highly innovative and revolutionary ideas” towards new paradigms for future internet architecture by bridging multi-disciplinary long-term research and experimentally-driven large-scale validation. FIRE foundational objectives were:

- a) Creation of a multi-disciplinary, long term research environment for investigating and experimentally validating highly innovative and revolutionary ideas for new networking architectures and service paradigms;
- b) Promotion of experimentally-driven yet long-term research, joining the two ends of academy-driven visionary research and industry-driven testing and experimentation, in a truly multi-disciplinary and innovative approach;
- c) Realization of a large scale European experimental facility, by gradually inter-connecting and federating existing and new “resource clusters” for emerging or future internet architectures and technologies.

These objectives further evolved by refinement of experimentally-driven research as a visionary multidisciplinary research, defining the challenges for and taking advantage of experimental facilities. This was realized by means of iterative cycles of research, oriented towards the design and large-scale experimentation of new and innovative paradigms for the Future Internet - modelled as a complex distributed system. Therefore, research is expected to address all the associated aspects in a holistic vision, at all relevant levels and layers. The refinement of the research directions should be strongly influenced by the data and observations gathered from experimentation in previous iterations thus being “measurement-based” which requires the specification of relevant metrics and measurement tools.

The rationale was thus clear: create a dynamic between elaboration, realization, and validation by means of iterative cycles of experimentation. Nevertheless the “validation by experimentation” objective opens a broad spectrum of experimentation tools (in the broader sense) ranging from simulation¹ to real system experimentation. The selection of the experimental tool selection depends on

- a) the object of experimentation (corpus)
- b) the nature and properties of the results
- c) the cost function which depends on complexity, experimental and running conditions but also on the level of abstraction (referred to as “realism”)

¹ To keep this document simple we do not distinguish between the various classes of simulation from model simulation (macroscopic) to procedural simulation (microscopic) nor distinguish between the various class of simulation techniques

2 Systematic Experimentation

Our *thesis* is that “elaboration” requires validation by means of more abstract tools (not only because their resulting cost is less but because such tools produce results verifying all conditions explained here) followed by progressive addition of realism as part of the experimented system to ultimately reach so called field trials with real systems. Thus, systematic experimentation is a continuum. The following sections describe the dependencies with respect to this experimental chain.

1. “Computer Communication/Networking” is characterized by two fundamental dimensions: distribution of a large number of dynamically interacting (non-atomic) components and the variation of their inner properties that in turn influence these interactions. Thus compared to computer science the distribution/interaction and the large number of elements composing the system add two fundamental dimensions to computer science “paradigms”. A couple of illustrative examples would set the landscape: autonomic networking is the transposition of the autonomic computing concept in the communication space, and “virtualization” is the transposition of the abstraction concept of object-oriented programming in the networking space. More the dynamic nature of these interactions results in modifying the scaling properties of the individual components besides modifying the properties of the global system. Many other examples can be cited, the fundamental observation is: no theoretical **experimental model** exist – or more precisely – the complexity of the resulting system makes its modelling a research discipline on its own. However, this doesn’t mean that a systematic **experimental methodology** could not be defined based on our experience from practicing core experiments in these core disciplines. Such methodology would typically include the following steps (part of each iteration):
 - a) specification of the performance objectives, (technical and non-technical) constraints, and description of expected results
 - b) definition of relevant performance criteria and metrics
 - c) description of the modus operandi including configuration, initialization, and running conditions and (iterative) procedure(s) to be executed
 - d) reporting on observations and the resulting analysis and the feedback on each iteration before reaching (partial) conclusion

A key aspect of a sound experimental methodology is an un-ambiguous **formal description of experiments**, which can be used in all of the above steps to guide the actual instantiation of the experiment setup and to correctly interpret the obtained results. Such a description should capture the abstract nature of the experiment, so that it can be executed on different kinds of experimental facilities as well as in virtual environments like those provided by simulators. This description should, in particular, list all parameters (both inputs and outputs) that are relevant for the experiment. The actual execution of the experiment on a given platform may be eased by tools that automatically translate the experiment description for the target execution environment. Such a translation need to take into account that each specific execution environment provides different degrees of **controllability** and **monitorability** for the relevant experiment parameters.

In particular, the well-known tradeoff between realism and controllability should be under direct control of researchers, who should be able to set up different kinds of experimental testbeds whose isolation from the external environment can be controlled and tuned according to the objectives of the experiments.

2. On the other hand, one shall characterize the output of experimentation: in order to ensure **verifiability**, **reliability**, **repeatability**, and **reproducibility** of the experimental results. Ensuring these properties implies in turn to control the experimental conditions (parameterisation, input/output, and running). The above properties of experimental results are defined later.
3. Different experimental tools can be used. As stated above their selection is neither arbitrary nor religious: it depends on the experimental objective and maturity of the experimented corpus. Nevertheless, each of them needs to ensure that the conditions defined previously in this paper are verified. However it is clear that fulfilling these conditions does not come at the same cost for the same level of abstraction. We can distinguish three types of abstraction:
 - a) abstraction of the network (traffic model, network resource consumption model, processing model, etc.)
 - b) abstraction of the system (processing/memory resource consumption model, computation model, etc.)

- c) abstraction of the environment (user/behaviour model, application model, etc.)

From this decomposition we can associate a level of realism when the abstraction is replaced by a “real” entity. Without entering into the debate about reality or what reality actually represents, we simply consider here a real system as an instantiation of the experimented components on hardware and/or software substrate depending on the expected level of performance. Validation of a new algorithm would be better conducted on a simulation platform (after formal verification) not only because their resulting cost is less but because such tools produce results verifying all conditions explained above. Subsequently, progressive addition of realism as part of the experimented system would consist in instantiating the execution stratum (remove the system abstraction) in order to perform emulation experiments. Emulation experiments can lead to reproducible and repeatable results but only if "conditions" and "executions" can be controlled. Realism can thus be improved compared to simulation (in particular for time-controlled executions of protocol components on real operation system). Nevertheless, experiments are more complex and time consuming to configure and execute; performance evaluation is possible if platforms comprise “sufficient number” of machines (representative of the experiment to conduct). Emulation still requires synthetic network conditions (models) if executed in controlled environment and either injecting real traffic or replay traffic traces (not that even when available "spatial distribution" of traffic remains problematic to emulate).

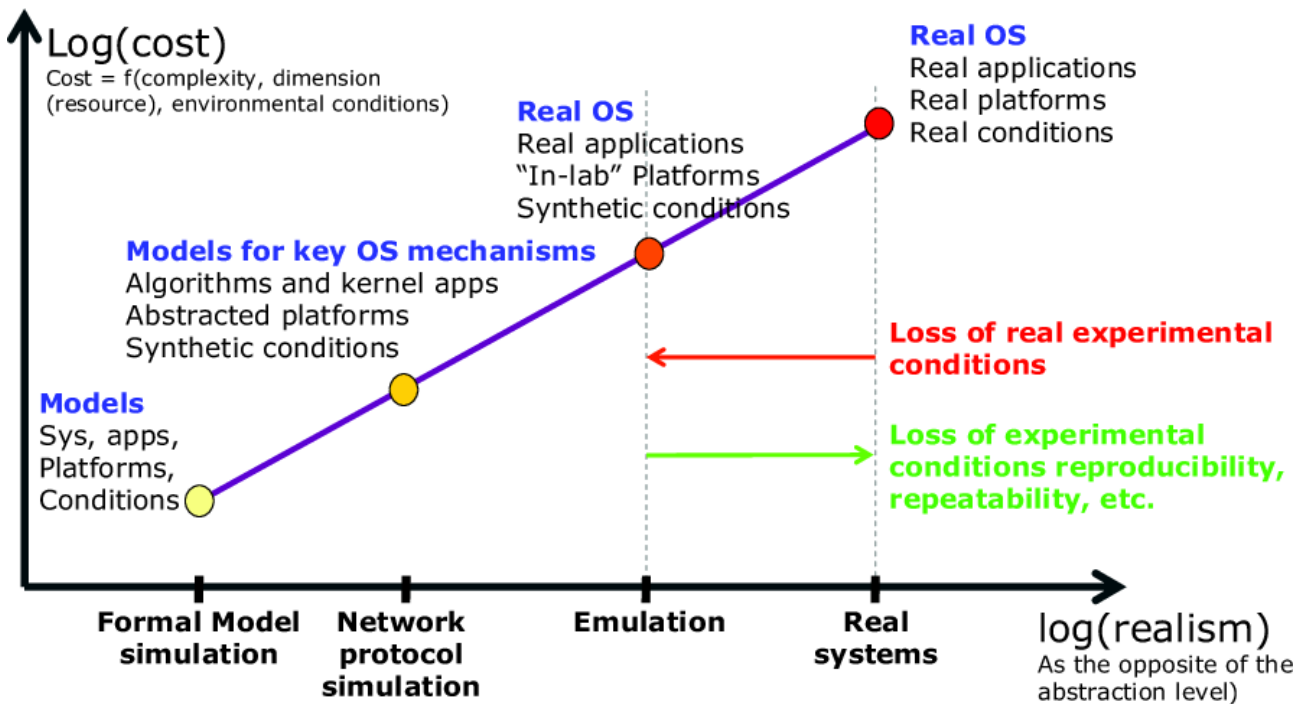


Figure 1 Cost vs. level of realism

Stepping into real system experimentation increases cost but also increases realism. As such the loss of control of experimental conditions in such systems raises the issue of persistence of the properties observed earlier in the experimental chain. In particular, these properties shall already be determined by the earlier experimental stages (leaving them intrinsically part of experimental research activities). Practically, in order to ensure – at least better control of the experimental conditions, the following elements might be considered:

- a) Specify performance analysis methodology together with the necessary mathematical tools to be able to perform data analysis and mining tasks on experimental data coming from various monitoring points (from single or multiple testbeds). This objective also covers specification of the necessary mathematical tools to analyze the sensitivity of the performance measures to changes in the "experimental model" parameters. Sensitivity analysis attempts to identify how responsive the results of an experimental model are to changes in its parameters: it is an important tool for achieving confidence in experimentation and making its results credible. Sensitivity analysis quantifies the dependence of system behaviour on the parameters that affect the modelled process and in particular its dynamics. It is used to determine how sensitive a model is to changes in the numerical value of the model parameters and changes in the model structure.
- b) Specify distributed performance monitoring system (while) allowing experimenters to choose the best tool(s) for their experimentation.

- c) Define a standard experiment description and control interface and wrap existing tools within this API. This standard interface will focus on providing a common programming interface to describe every aspect of a networking experiment but will also attempt to provide robust experiment monitoring and management facilities and will integrate with the data analysis and data mining tools developed in a). The availability of a standard experiment description language can be useful in two ways: i) as an enabler for the execution of the same experiment in different execution environments and the automatic comparisons of the outputs, and ii) as a platform-agnostic description of the experiment to be used by a super-controller to manage the parallel execution of the experiment on several federated heterogeneous environments. The experiment description should also be automatically translated in “virtual infrastructure” descriptions. This approach should allow the automatic setup of an experimental testbed on top of distributed facilities.

Note: sensitivity analysis of the reliability, the performance, and the performability of the monitoring system is a complementary objective.

2.1 Reduce cost

Figure 1 illustrates the dependency of cost, being a function of complexity, dimension (scale) and environmental conditions, vs. the level of experimental realism, being an inverse function of the abstraction level. The introduction of experimental resource sharing might be a mechanism to control cost even when increasing the level of realism. Federation of testbeds is introduced as a mechanism to control cost, while two different classes of federation can be distinguished:

- Heterogeneous federation in which the contributions to the pool of resources are heterogeneous in nature and adhere to different management and control frameworks and possibly provide different monitoring interfaces and formats. However use cases exist that require an experimental environment composed of such heterogeneous resources.
- Homogeneous federation in which the contributions to the pool of resources are homogeneous in nature and usually adhere to or adopt the same management and control framework, providing also the same monitoring interfaces.

Nevertheless in both cases the expectation is that the cost can be controlled in a manner illustrated below.

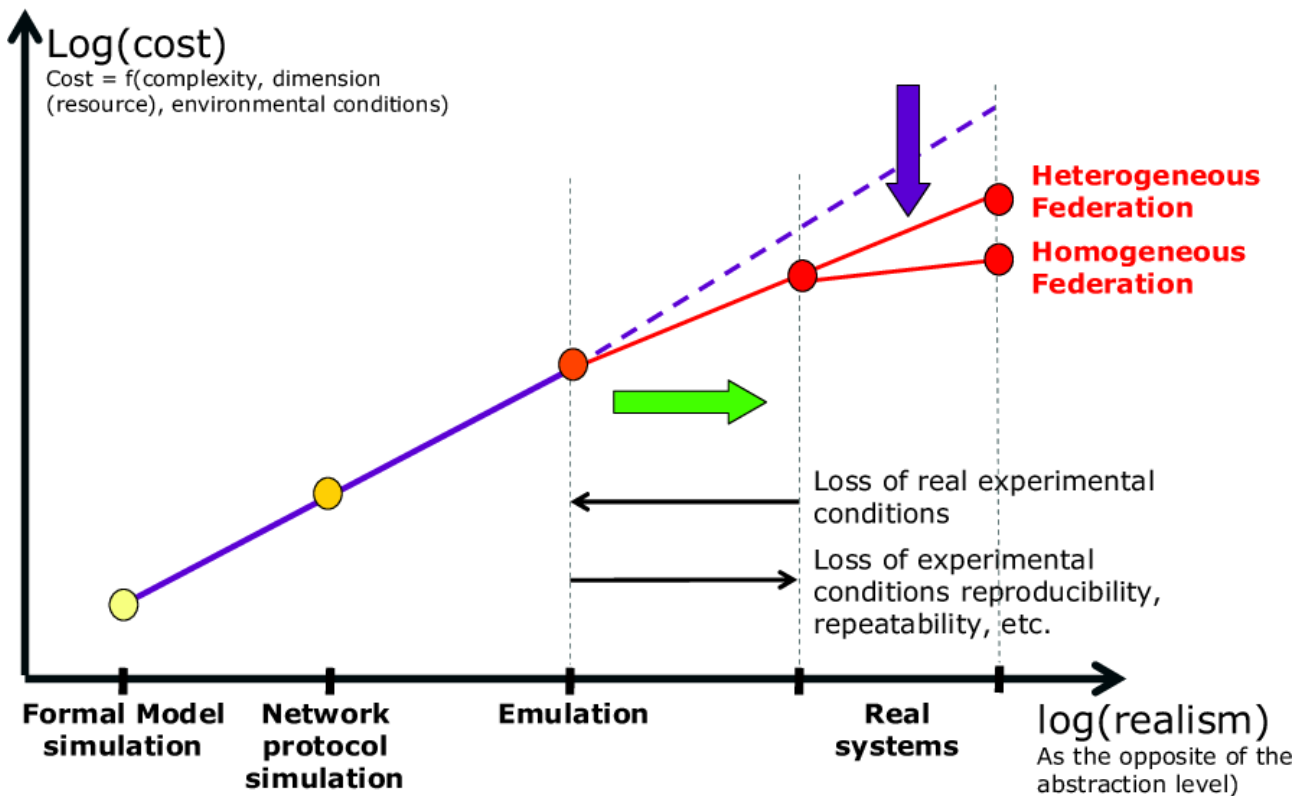


Figure 2 Role of Experimental facilities in controlling cost

2.2 Definitions

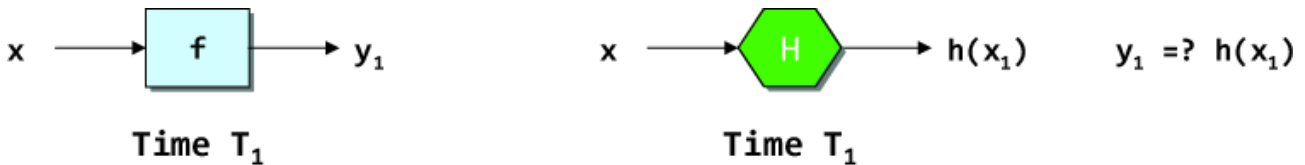
2.2.1 Properties of experimental results

Verifying the repeatability, reproducibility, and reliability conditions ensures generalization of experimental results, and verifiability their credibility.

If we define the experimental model by a function F , with input variables x_1, \dots, x_n and parameters e_1, \dots, e_m such that $F(x_1, \dots, x_n | e_1, \dots, e_m) = y$

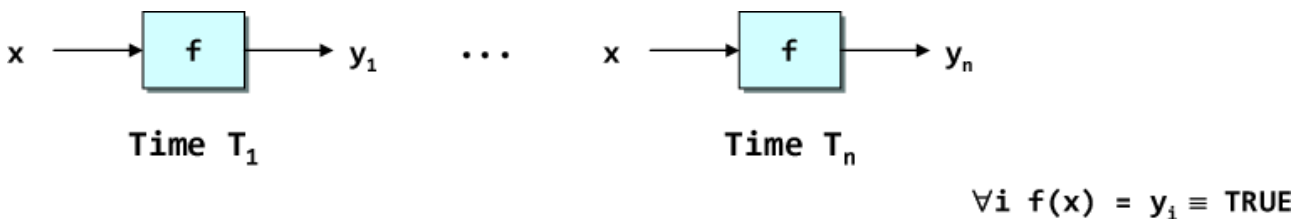
2.2.1.1 Verifiability

Verifiability means that we can find a formal model H of F such that $H(x_1, \dots, x_n | e_1, \dots, e_m) = F(x_1, \dots, x_n | e_1, \dots, e_m)$.



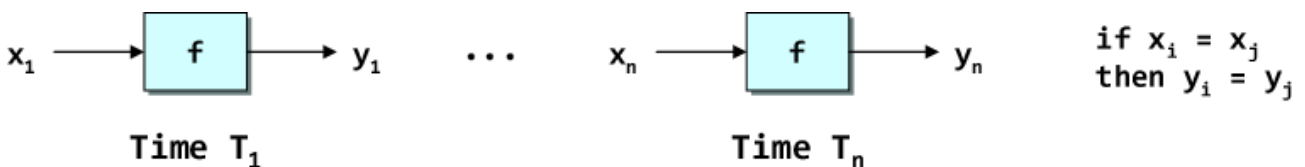
2.2.1.2 Reliability

Reliability (defined as probability that system or component will perform its intended function during a specified period of time under stated conditions) means that output of the model during the pre-defined time interval $[t_{k-1}, t_k]$, $1 \leq k \leq T$, $F(x_1, \dots, x_n | e_1, \dots, e_m)[t_k] = y[t_k]$ exists.



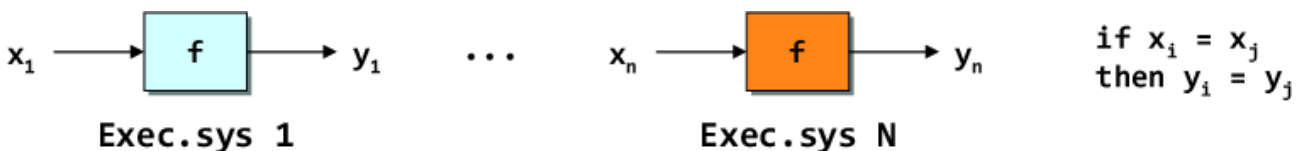
2.2.1.3 Repeatability

Repeatability means that if $(x_1, \dots, x_n | e_1, \dots, e_m)[t_{k-1}] = (x_1, \dots, x_n | e_1, \dots, e_m)[t_k]$ then $y[t_{k-1}] = y[t_k]$ (persistence).



2.2.1.4 Reproducibility

Reproducibility means that the experimental model $F(x_1, \dots, x_n | e_1, \dots, e_m)$ can be executed at the same time on different experimental systems and produce the same output.



3 Experimentation vs. Testing

Trying to define the difference of the terms “experimentation” and “testing” one can lookup the dictionary. However this is not particularly useful since the definition of experimentation according to The American Heritage, Dictionary of the English Language leads to a reciprocal definition.

“**Experiment** is a **test** under controlled conditions that is made to demonstrate a known truth, examine the validity of a hypothesis, or determine the efficacy of something previously untried”.

More useful seems to be a practical approach to consider how certain activities are labelled. As such we can find in ETSI the following definition for **testing**:

- a) Conformance testing
 - Gives a high-level of confidence that key components of a device or system are working as they were specified and designed to do
 - High control and observability but limited scope
 - Tests individual system components but a system is often greater than the sum of its parts! -- Does not prove interoperability
 - Can be automated
- b) Interoperability testing
 - Interoperability testing is system testing (end-to end functionality)
 - Tests at a ‘high’ level and tests the ‘whole’, not the parts
 - Tests functionality and shows function is accomplished (but not how)
 - Does not prove interoperability with other implementations
 - Usually executed manually

In contrast to testing, experimentation, as defined in other scientific disciplines as well, is summarised as:

- a) The orderly or methodical observation of the variation of facts resulting from artificial stimuli in a reproducible environment that confirms a hypothesis (verification) or rejects it (falsification)
- b) In an experiment we manipulate at least one system parameter (variable) according to a pre-defined plan and observe its impact on other (dependent) parameters under controlled conditions.
- c) The validity of the hypothesis is only shown if we can reproduce the experiment under the same conditions and obtain the same result.

In particular this means that we strive to minimise or eliminate “uncontrollable” environmental effects. However controlled conditions means usually a laboratory environment.

If we put again federation in perspective as a mechanism to achieve scale and realism, then we must recall the assertion as illustrated in Figure 1 that this will lead to a less controlled environment. As a consequence it will be impossible to claim that an experiment under real world conditions can be reproduced, in particular due to the external environmental parameters (e.g. users).

Considering the discussion above we could draw the distinction between testing and experimentation according to the knowledge about the right stimuli and observation points of a system. Using this distinction, **testing** has a known list of stimuli and observation points which can be used to determine if a system is working correctly. In contrast **experimentation** is the search for the right stimuli and observation points that are useful for a reasonable assessment of a system.

Another distinction could be drawn according to the level of maturity of the knowledge about the behaviour of a system. This means that we might attempt to draw a line between **research** and **development**, applying experimentation in the research phase and testing the technology development phase of the activities.

As a remark it is important to distinguish **experimentation** and **benchmarking**, the latter being defined as:

Benchmarking is a test procedure executed under controlled (often optimal) conditions to measure functional and/or performance metrics such as number of cycle, access time, etc. and compare the results to an existing specification e.g. well-accepted industry standard.

4 Large-scale trials and pilots

The infrastructure for conducting large-scale experiments and testing is currently constructed via federation of testbeds. Herein we define **testbed** in the context of new information and (tele-) communications technologies for networks and services as follows:

A testbed is an environment which allows experimentation and testing for research and development products. A testbed provides a rigorous, transparent and replicable environment for experimentation and testing.

We define **testbed federation** as follows:

A testbed federation or federated testbeds is the interconnection of two or more independent testbeds for the creation of a richer environment for experimentation and testing, and for the *increased multilateral benefit* of the users of the individual independent testbeds.

Virtualisation technologies allow the concurrent use of shared resources by more than one experimenter and could possibly be the mechanism to introduce a certain level of reproducibility in large-scale (inherently uncontrollable) environments. Such mechanisms are known as “slices” as used in PlanetLab/GENI, Virtual Customer Testbeds (VCT) as used in Panlab/PII or simply virtual machines that can be specified in terms of certain parameters. Slices and VCTs can span across the borders of the individual testbeds of a federation.

4.1 Trial vs. Pilot

The notion of **trial** as is commonly used to describe activities in the ICT discipline, seems not to differ from the act of experimentation and testing as these have been used in this document. Thus it is suggested to drop usage of the term when conducting activities that aim to verify the functionality of a system or parts of it.

However in many cases the experimenter or tester is much more interested in the interaction of the intended end-user of a service or device (a product) and in particular whether and under which conditions the end-user is prepared to engage in a business relationship for using the product. In other words if we add the business dimension in the experimentation or testing activities then we have a **pilot** that could be defined as follows:

The execution of an experiment or test including business relationship assumptions, exemplifying a contemplated added value for the end-user of a product.

4.2 About (large)-scale

In testing and experimentation we often use the term large-scale to denote an environment that exceeds the size, scope and capabilities of a laboratory environment. The notion of scale could not refer to the number of artefacts, whether these are switches, routers, computing nodes, sensors, cars, homes, etc.

Scale can refer to the scope or extend of experimentation and **large** can imply heterogeneity based on the assumption that **large-scale** exceeds the borders of a single laboratory environment. In particular **large** depends on the focus or abstraction level and can be very dynamic. For example in the early days of circuit design we denoted as large-scale 100-5000 circuit elements. When technology outreached that number additional attributes were invented, e.g. very-large-scale (5k-50k), super-large-scale (50k-100k) and finally ultra-large-scale (>100k). Today we know that it was not reasonable to invent new attributes. It is relatively simple to specify large-scale e.g. in the scope of sensors/small devices in the upper thousands, or in the scope of routing nodes in the lower hundreds. However the technology evolution might render these numbers obsolete as was the case in circuit design. It might be more useful to use the **cost** factor as a measure, whereas cost is a function of complexity, dimension and environmental conditions as defined previously. One possibility is to postulate that **large-scale** denotes an environment that exceeds the **cost** of a laboratory environment at any given point in time by one or more levels of magnitude.

5 Application of experimentation methodology

5.1 Self-NET Project

Self-NET project has developed three experimentation facilities for testing Self-NET goals and specifically network elements self-management for: a) coverage and capacity Optimization b) Distributed Protocol Composition, and c) Self-configuration and adaptive routing of wireless access Part.

In all cases the systematic experimentation methodology has been followed. The research follows the characteristic of the Computer Communication/Networking wherein a large number of components are dynamically distributed and interacting with each other. In addition, the variation of their properties and system configuration influence these interactions such as monitoring, decision making and execution. The dynamic nature of these distributed interactions results in modifying the scaling properties of the individual components besides modifying the organizational properties. The systematic experimentation methodology is achieved by the specification of performance objectives and metrics such as end-to-end delay, packet delivery fraction etc.

The research focuses on the third and fourth stage of experimentation, using a real implementation in a laboratory environment. The cost is a function of time, complexity, dimension (number of network elements), environmental conditions such as the physical distance between the network elements, device drivers etc.

As regards heterogeneous federation the contributions to the pool of resources are heterogeneous in nature and adhere to different management and control frameworks and possibly provide different monitoring interfaces and formats. Moreover, Homogeneous federation the contributions to the pool of resources are homogeneous in nature and usually adhere to or adopt the same management and control framework, providing also the same monitoring interfaces. Hence, the cost as function of time and complexity will be higher in the heterogeneous federation than in the homogeneous federation.

The strategic benefits include the verification of end-to-end functionality, ability to perform experiments where in at least one system parameter can be varied according to a pre-defined plan and observe its impact on other (dependent) parameters under controlled conditions and enables the researchers to draw a line between research and development, applying experimentation in the research phase and testing the technology development phase of the activities.

5.2 PERIMETER Project

PERIMETER's main objective is to establish a new paradigm of user centrality for advanced networking. Putting the user in the centre rather than the operator enables the user to control his or her identity, preferences and credentials. This enables the mobile user to be "Always Best Connected" in multiple-access multiple-operator networks of the Future Internet.

For achieving that, PERIMETER develops and implements protocols and a middleware to address requirements for privacy, security, resilience and transparency. The network selection is done based upon a parameter called Quality of Experience. Quality of Experience is a way to evaluate a network connection by user perceivable indexes and parameters (i.e. cost, energy efficiency, security, privacy and connection performance). Network selection can be done manually by the user or can be handled automatically by PERIMETER using the Quality of Experience thus creating a single-sign-in like experience.

The architecture of PERIMETER consists of terminal nodes and support nodes. Terminal nodes are devices which are operated by the user. Currently mobile phones and netbooks based upon Google's Android platform are supported. The support nodes are physical devices that can act as data sources and gateways and are distributed connected with a peer-to-peer approach. In the PERIMETER testbeds support nodes reside in two federated testbed, terminals run on the federated testbeds and the applications run across the federated testbeds.

During all testing and experimentation two key paradigms of PERIMETER must be kept in mind. Firstly, the real world and digital world must be interconnected. Thus, the services must be aware of the users' environment and preferences. And secondly, the user should be Always Best Connected where "Best" is defined by the user. As the two paradigms imply rather vague than formally definable quality indicators, testing and experimentation with user involvement is necessary for the project in addition to standard technical testing mechanisms. Therefore the PERIMETER consortium believes that it is possible to draw a distinction between testing and experimentation. Testing is a continuous process needed in software

development. Experimentation, on the other hand, is needed to guarantee the success of the research and innovation aspects of the project.

5.2.1 Development and Testing Methodology

PERIMETER uses the Agile development methodology and Test Driven Development approach to its development, implementation and testing phases. A large number of supporting tools are used to aid the processes of team collaboration (Subversion, Trac), continuous testing (JUnit, Apache Ant, Cobertura), continuous integration (Hudson) and structured software development (IDEs, Findbugs, PMD). In these phases Unit, Functional and Integration testing is conducted in a testing cycle that is proven to be robust, scalable and secure in order to ensure that the PERIMETER software is brought to a level where it achieves its functional objectives and the objectives of its potential end users.

To effectively validate and demonstrate the results of the PERIMETER project, the innovative aspects of the project are verified in a suitable, and state of the art, federated testbed demonstration facility. Conformance tests are performed on the federated testbeds to ensure that key components of the system are functioning as expected. These tests are complemented with Interoperability tests to ensure the end-to-end functionality of the system is as expected. Both sets of tests are tested against the scenario under analysis, in the scenario-driven approach used in this project in order to demonstrate that the testing process is robust and to ensure the verifiability and reliability of the results. Both testing phases have to be repeatable and reproducible in order to achieve this. Testing of the specific applications running on the federated testbeds is also performed.

This process is further complimented by the employment of a user-driven approach to the requirements specification and the determination of features and subsequent testing phases with the use of Living Labs and dedicated usability sessions. These testing sessions are conducted from a concept point of view and within the federated testbed environment. Performance and scalability issues are addressed with the introduction of emulation and simulation of network conditions.

Finally, the entire process is realised by means of several iterative cycles of this process, where refinements are made, further functionalities are implemented and where testing and experiments become more stringent as the project progresses and naturally matures. This process is illustrated in Figure 3.

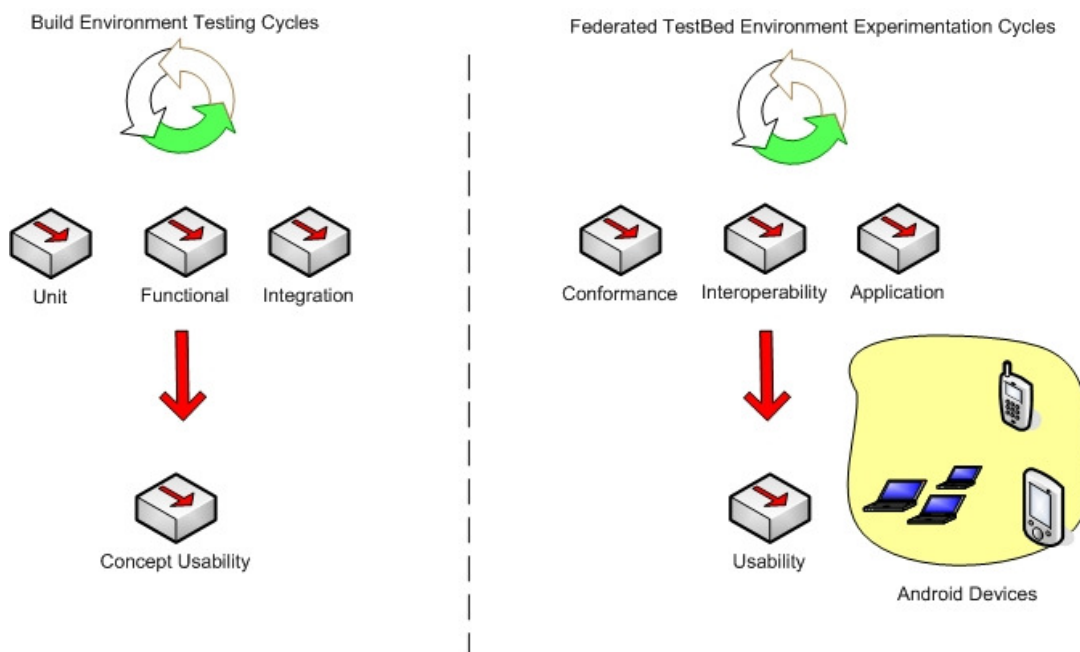


Figure 3 PERIMETER Testing and Experimentation Process

5.2.2 Experimentally Driven Research

The view point of the PERIMETER consortium is that the use of testbeds, single or federated, heterogeneous or homogeneous, provides a vital role in the proposed cost versus realism cost function.

The introduction of the Federated Testbed stage provides a mechanism for the cost to be controlled for this phase of the project, whilst still allowing the level of realism to be increased. A further advantage of this stage is that the conducted experiments can be reproduced and repeated in a close to real world situation.

Further, the issue where experiments depend on the “external environmental parameters (e.g. users)” can be alleviated to a degree with the introduction of a user-driven approach to the research, development, implementation, testing and experimentation phases of the project, such as that already described for the PERIMETER project.

Still the issue remains as to what will constitute the end of testing, or when a certain level of testing has been achieved. This is particularly true for large scale experimental projects and indeed to the PERIMETER project. This is perhaps an issue that the Experimentally Driven Research Whitepaper could also potentially address.

Annex A

A.1 Contributions

Dimitri Papadimitriou, Alcatel-Lucent Bell Labs

Anastasius Gavras, Eurescom GmbH

Roberto Canonico, Universita' di Napoli Federico II

Dev Pramil Audsin, King's College London

David Wagner, Fraunhofer FOKUS

Apostolos Kousaridas, Nancy Alonistioti, University of Athens

Eileen Dillon, Gemma Power, Waterford Institute of Technology/TSSG

Martin Dobler, FH Vorarlberg

ECODE consortium, <http://www.ecode-project.eu/>

PII and Panlab consortia, <http://www.panlab.net>

FIREworks consortium, <http://www.ict-fireworks.eu>

SELF-Net consortium, <https://www.ict-selfnet.eu/>

PERIMETER consortium, <http://www.ict-perimeter.eu>

A.2 Further contributions have been used from

Panel discussion at IPOM 2008, 8th IEEE International Workshop on IP Operations and Management, September 22-26, Samos Island, Greece - held as part of Manweek 2008

Panel discussion at Open NGN Testbeds – Infrastructure as a Service workshop held as part of the IMS Workshop 2008, 6-7 November 2008, Berlin

Panel discussion at NGN test centre launch event, March 2010, Dublin

Several Future Internet Research and Experimentation (FIRE) cluster sessions at the Future Internet Assembly (FIA), Madrid, December 2008, Prague, May 2009, Stockholm, November 2009 (<http://www.future-internet.eu>)

FIREworks workshop on the Baltic sea in preparation of the FIA Stockholm, November 2009