

FITTEST – Future Internet Testing



The Future Internet will be a complex interconnection of services, applications, content and media, on which our society will become increasingly dependent for critical activities such as social services, learning, finance, business, as well as entertainment. This presents challenging problems during testing; challenges that simply cannot be avoided since testing is a vital part of the quality assurance process. With important partners from Spain (Universidad Politecnica de Valencia), UK (University College London), Germany (Berner & Mattner), Israel (IBM), Italy (Fondazione Bruno Kessler), The Netherlands (Utrecht University), France (Softeam) and Finland (Sulake) FITTEST aims to attack the problems of testing the Future Internet with Search Based Testing.

At a glance

Project title (full):

FITTEST – Future Internet Testing

Project coordinator:

Dr. Tanja E.J. Vos
tvos@pros.upv.es
Telf. +34690917971

Partners:

Universidad Politecnica de Valencia (Spain), University College London (United Kingdom), Berner & Mattner (Germany), IBM (Israel), Fondazione Bruno Kessler (Italy), Universiteit Utrecht (The Netherlands), Softeam (France), Sulake (Finland).

Duration:

From September 2010 to September 2013

Total cost:

5.845.000€

Programme:

FP7, ICT, Internet of services, Software and Virtualisation

Further information:

<http://www.pros.upv.es/fittest/>

FITTEST, Why?

The Future Internet (FI) will be a complex interconnection of services, applications, contents and media, possibly augmented with semantic information and based on technologies that offer a rich user experience, extending and improving current hyperlink-based navigation. Our society will be increasingly dependent on services built on top of the FI, for critical activities such as public utilities, social services, government, learning, finance, business, as well as entertainment. As a consequence, the applications running on top of the FI will have to satisfy strict and demanding quality and dependability standards.

FI applications will be characterized by an extreme level of dynamism. Most decisions made at design or deploy time are deferred to the execution time, when the application takes advantage of monitoring (self-observation, as well as data collection from the environment and logging of the interactions) to adapt itself to a changed usage context. The realization of this vision involves a number of technologies, including:

- Observational reflection and monitoring, to gather data necessary for run-time decisions.
- Dynamic discovery/composition of services.
- Hot component loading and update.
- Structural reflection, for self-adaptation.
- Asynchronous communication.
- High configurability and context awareness.
- Composability into large systems of systems.

While offering a major improvement over the currently available Web experience, complexity

of the technologies involved in FI applications makes testing them extremely challenging. Some of the challenges are described in the Table below:

CH1	<i>Dynamism, self-modification and autonomic behaviour</i>	FI applications are highly autonomous; their correct behaviour for testing cannot be specified and modelled precisely at design-time.
CH2	<i>Low observability</i>	FI applications are composed of an increasing number of third-party components and services, accessed as a black box, which are hard to test.
CH3	<i>Asynchronous interactions</i>	FI applications are highly asynchronous and hence harder to test. Each client submits multiple requests asynchronously; multiple clients run in parallel; server-side computations are distributed over the network and concurrent.
CH4	<i>Time and load dependent behaviour</i>	For FI applications, timing and load conditions make it hard to reproduce errors during debugging.
CH5	<i>Huge feature configuration space</i>	FI applications are highly customizable and self-configuring, having a huge number of configurable features, such as context, user, environment-dependent configurable parameters that need to be tested.
CH6	<i>Ultra-large scale</i>	FI applications are often systems of systems; traditional testing adequacy criteria, like coverage, cannot be applied.

FITTEST, What and How?

This project aims at addressing these testing challenges. Search Based Software Testing (SBST) is the FITTEST basis for test case generation, since it can be adopted even in the presence of dynamism and partial observability that characterize FI applications. The key ingredients required by search algorithms are:

- an objective function, that measures the degree to which the current solution (e.g. test case.) achieves the testing goal (e.g., coverage, concurrency problems, etc.);
- operators that modify the current solution, producing new candidate solutions to be evaluated in next iterations of the algorithm.

Such ingredients can usually be obtained even in the presence of high dynamism and low observability. Hence, it is expected that most testing techniques that will be developed to address the project objectives will take advantage of a search based approach.

For example, *model-inference* for model-based testing requires execution scenarios to limit under-approximation. We can generate them through SBST, with the objective of maximizing the portion of equivalent states explored.

Oracle learning needs training data that are used to infer candidate specifications for the system under test. We can select the appropriate executions through SBST, using a different objective function (e.g., maximizing the level of confidence and support for each inferred property).

We will generate **classification-trees** from models and use SBST to generate test cases for classification-trees

Input data generation and *feasibility check* are clearly good candidates to resort to search based algorithms, with an objective function that quantifies the level to which a test case gets close to satisfying a path of interest.

In *concurrency testing*, test cases that produce critical execution conditions are given a higher objective value defined for this purpose.

Coverage targets can be used to define the objective function to be used in *coverage and regression testing* of ultra-large scale FI systems.

SBST represents the unifying conceptual framework for the various testing techniques that will be adapted for FI testing in the project. Implementations of such techniques will be integrated into a unified environment for continuous, automated testing of FI applications. Quantification of the actually achieved project objectives will be obtained by executing a number of case studies, designed in accordance with the best practices of empirical software engineering.

FITTEST, Who?

The FITTEST project brings together, not merely European leaders in this emerging field, but world leaders. **Universidad Politécnica de Valencia** (Spain) is project coordinator and research partner providing expertise in software testing and empirical evaluation. **University College London** (United Kingdom) is a research partner providing expertise in Search based Software Engineering. **Berner & Matter** (Germany) is a specialist in combinatorial testing. They will also play an important role in developing the final testing framework. **IBM** (Israel, an IBM center of competence in testing technologies) is a case study provider of a system that provides highly scalable services for web applications. Moreover, they contribute experience on concurrency testing, coverage criteria and regression testing. The **Fondazione Bruno Kessler** (Italy) will provide expertise on model-inference and model-based test case generation. The **University of Utrecht** (The Netherlands) will work on trace-based testing, code instrumentation and soft-typing. **Softteam** (France) is a software developer and has experience with model-driven technologies. **Sulake** (Finland) is a case study provider of an FI application.

Title

