The Feature Interaction Detection Method of VALISERV

Gilles Bernot / E'vry

Francis Klay / CNET Lannion

Farid Ouabdesselam / Grenoble

Bad interactions, formally?

Our goal: formal methods for computer aided Feature Interaction detection

Characterization of *{Bad Interactions}*
independently of particular features ?
No satisfactory complete modelization exist.
Approximations are necessary,
with many *subjective* considerations.

A fortiori there exist no generic formal/logical specification of bad interactions.

Dynamic interactions with expert people.

Human help is unavoidable for each set of features.

Pessimistic version

OK:

no rigorous definition of $\{Bad\ Interactions\}$

necessary approximations

subjectivity, no completeness ...

Let us do approximative things,
... let us **test** features !

Optimistic version (mine)

Formal testing is not approximative

Testing is incremental

Testing helps to approach the model under test progressively

Testing methods are generic w.r.t. the specifications

Testing allows to reach completeness *to the limit*

Quality can be tuned w.r.t. cost considerations.

⇒ Testing can help us to identify $\{Bad\ Interactions\}$

The landscape (1/3)

We are able to:

specify the POTS

specify features independently

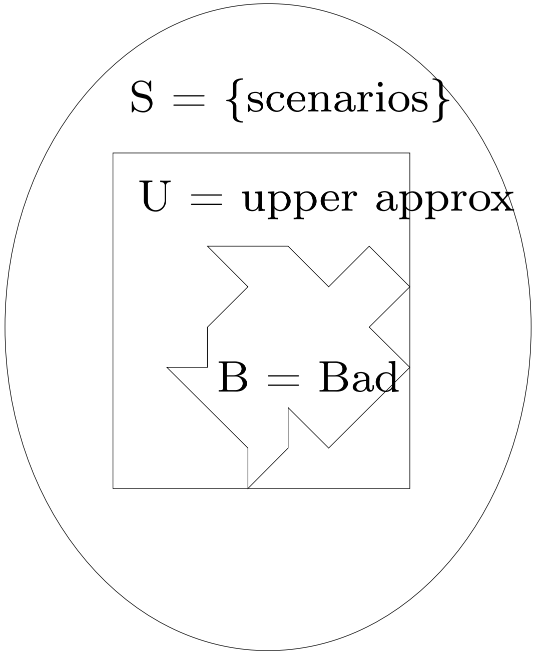
propose integration modes

We are unable to specify *a priori* the set of Bad Interactions

The landscape (2/3)

We are able to deduce from the specifications the set of all possible scenarios

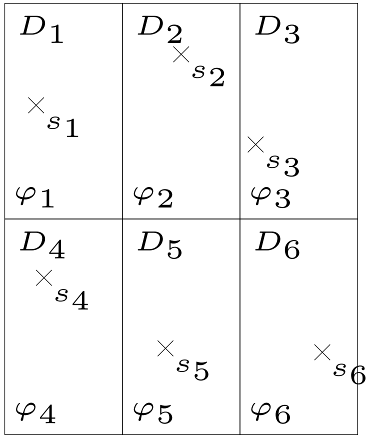
An expert is able to propose an upper approximation of *{Bad Interactions}*



The landscape (3/3)

Testing = – select cases

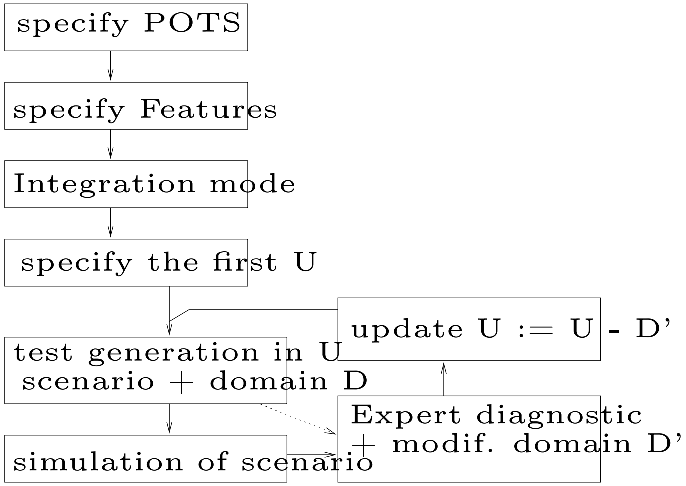
- submit to program
- decide success/failure



Feature integration testing =

- select subdomains of U and one representative scenario
- animation
- decision by the expert

The VALISERV project



Generation step:

a subdomain D of U

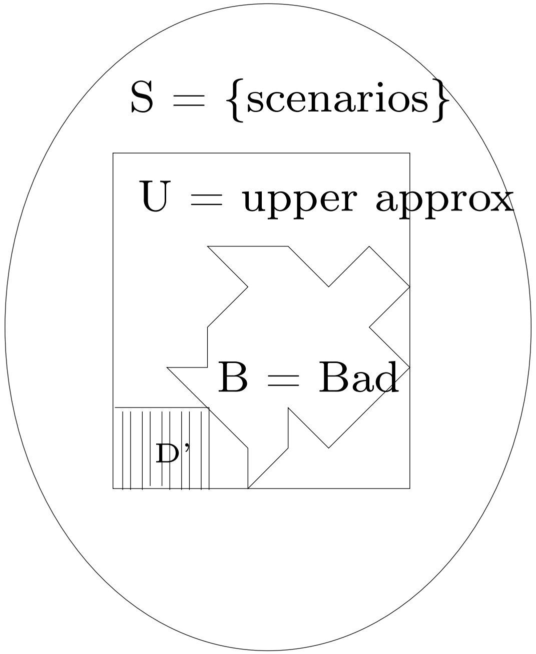
a representative scenario of it

Expert diagnostic:

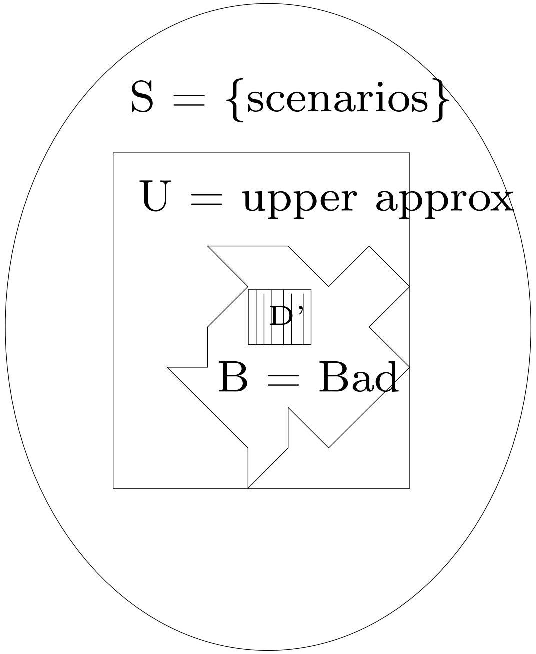
good / bad

modify domain → D'

“Good” diagnostic



“Bad” diagnostic



Critical aspects

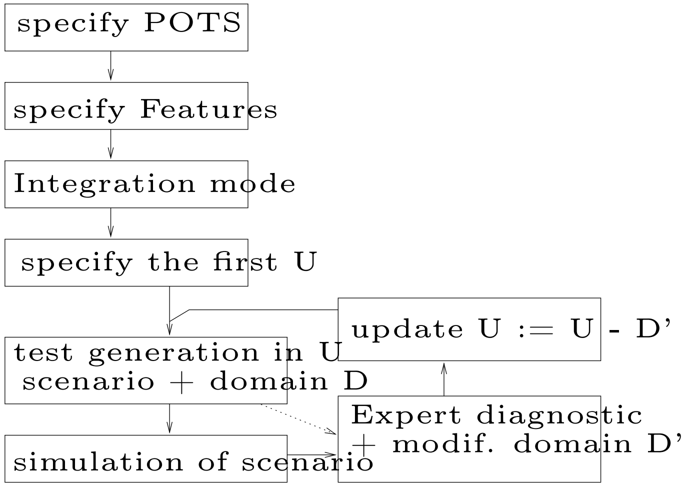
To properly define U at the beginning

To simulate the scenarios in a legible way, with good representation of cause/effect relations

To design a test generator able to dynamically integrate the successive D'

To decide when to stop the loop
$\approx$ to evaluate the progress at each loop
(and ensure a non negative progress)

Conclusion



A good cooperation between formal methods and industrial/expert knowledge.