

# Research lines

Ana Paiva  
apaiva@fe.up.pt



## Research lines

- Pattern Based GUI testing
- Reverse engineering
- Serious games
- Extract knowledge from user interactions

## Pattern Based GUI Testing

### Goals

- Aim to be a effectively applicable MBT approach in industry and to contribute to the construction of a higher quality GUIs and software systems
- For that PBGT focus on
  - Diminishing the time required to build a model
    - Increase the **model abstraction**
    - Build part of the model by **reverse engineering**
  - Test case explosion problem
    - Test only **common recurrent behavior**

## Pattern Based GUI Testing

### Login or Authentication

- They have the same general behavior but slightly different implementations

LOG IN

Email

Password

Stay logged in

[Forgotten Your Password?](#)

Continue >

Authentication

Username:

Password:

Validate

Sign in Google

Username

Password

Sign in  Stay signed in

[Can't access your account?](#)

## User Interface Test Pattern {<Goal,V,A,C,P>}

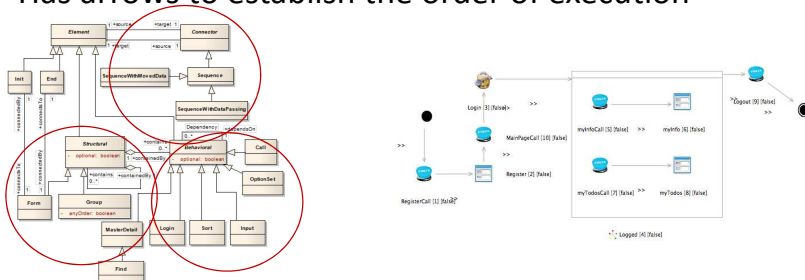
- **Defined by the developer**
  - **Goal:** ID of the test
  - **A:** sequence of actions to perform during test case execution
  - **V.variable:** set of pairs {[variable, inputData]} relating test data with variables of the test
- **Defined by the tester**
  - **V.inputData:** set of pairs {[variable, inputData]} relating test input data with the variables of the test
  - **C:** describes the final purpose (or why) the test should be executed
  - **P:** defines when the test can be executed

## Base UI Test Patterns

Icon	UITP	Set of Test Goals
	Call	{G_Call} – test the result of a call.
	Find	{G_Found, G_notFound} – test searches returning and not returning values.
	Input	{G_IV, G_IINV} – test for valid and invalid inputs.
	Login	{G_LV, G_LINV} – test for valid and invalid authentications.
	Sort	{G_SRTASC, G_SRTDESC} – test the sort for ascending and descending order.
	MasterDetail	{G_MD} – test if exchanging the value of the master, the detail updates accordingly.

## PARADIGM language

- Has nodes
  - User Interface Test Patterns
  - Structural nodes for structuring the model in different levels of abstraction
- Has arrows to establish the order of execution



## Show Results Form

## Pattern Based GUI Testing

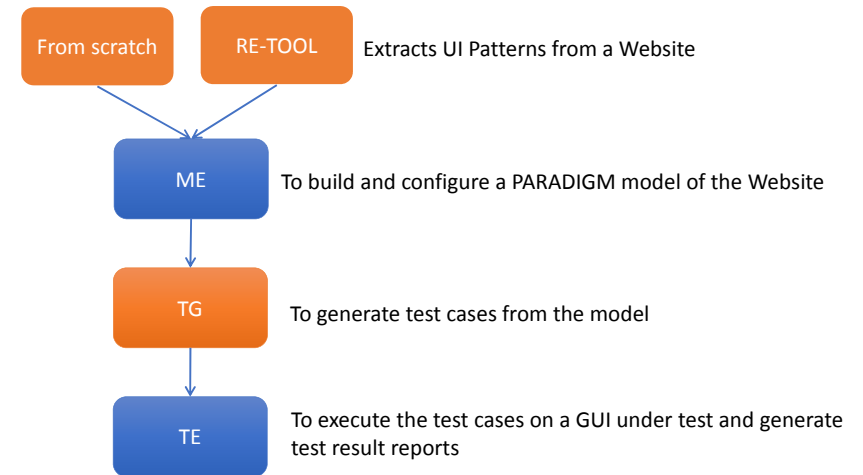
### Show Results Form

The image shows a screenshot of the RE/MAX Portugal website's search results page. A PARADIGM model is overlaid on the page, identifying various UI patterns. Red circles highlight specific elements on the website, and red arrows point from these elements to corresponding nodes in the PARADIGM model. The model includes nodes for 'Limit Search [2.1.1]', 'Filter [2.1.2]', 'Sort [2.4]', 'AdjustSearch [2.2]', and 'ChangeLanguage [2.3]', all contained within a 'ResultsGroup [2.5]' container. A 'Find Entries' dialog box is also visible in the foreground, showing search results for 'Find Entry 22, Find Entry 14'.

9

## Pattern Based GUI Testing

### Overview



10

## Pattern Based GUI Testing

### Front-end

The image shows a screenshot of the Eclipse IDE displaying the front-end of the PARADIGM model for the 'Show Results' form. The main window shows a diagram with nodes for 'Limit Search [2.1.1.1]', 'Filter [2.1.1.2]', and 'AdjustSearch [2.1.2]'. A 'Find Entries' dialog box is open, showing search results for 'Find Entry 22, Find Entry 14'. The 'Properties' window is also visible, showing the 'Core' and 'Appearance' properties for the selected element.

11

## Pattern Based GUI Testing

### Reports

The image shows a screenshot of the test execution reports. The top part displays a test case coverage summary, showing that 85,71% of test cases passed (92,86% executed). Below this, there is a code coverage report for the 'main.paradigm\_diagram' project, showing the coverage percentage for various files. The report includes a table with columns for 'File', 'Coverage', and 'FilePath'.

File	Coverage	FilePath
smarty.php	100%	C:\Users\Public\...
index.php	100%	C:\Users\Public\...
informacoes.php	100%	C:\Users\Public\...
listarobras.php	0%	C:\Users\Public\...
modulosadmin.php	37.5%	C:\Users\Public\...
novobras.php	0%	C:\Users\Public\...
novobraspor.php	0%	C:\Users\Public\...
registro.php	44.4%	C:\Users\Public\...

12

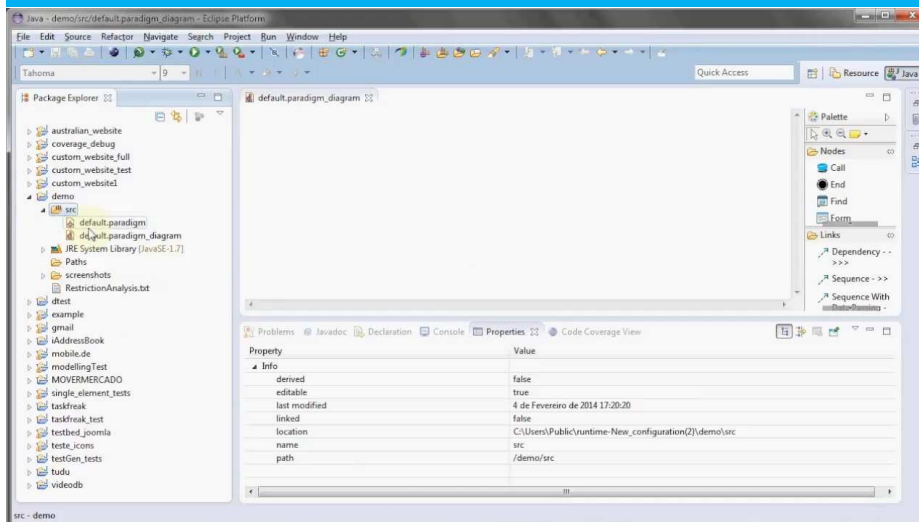
### PBGT framework extensions

- It is possible to add new UTP to the PARADIGM language to deal with new trends.
- It is possible to add new test case generation algorithms
- It is possible to add new test drivers to test other software applications besides web and Android

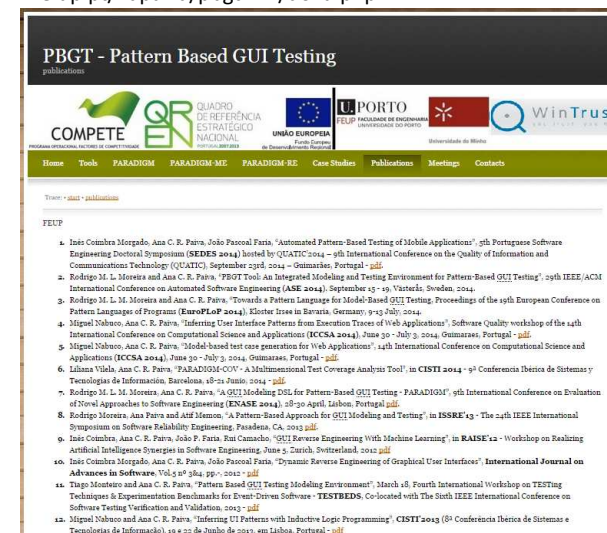
13

### Community impact

- **Reusability concerns** - UI Test Patterns can be reused during the GUI modeling and testing process
- **Reduced efforts** - when compared with other GUI modeling approaches, models can be crafted and configured in short time
- **Goal focus** - typical Model-Based GUI Testing tools are centered in modeling the behavior of the application
- With PBGT the focus is directed towards modeling testing goals
- **Platform independent** - PBGT Tool can be used to model and test web applications and also mobile applications
- **No source code is required** - PBGT Tool does not require access to the source code of the systems under test, in order to create or generate GUI models from them;
- **Low maintenance and evolutionary** - With few steps it is possible to extend the initial set of UI Test Patterns, and also to adjust current test strategies (or create new ones) to support new UI trends;
- **Simple to use** - With few knowledge on testing activities, users can start modeling and testing software in short time.



<http://www.fe.up.pt/~apaiva/pbgtwiki/doku.php>



16

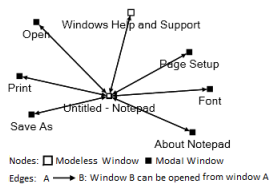
## Future Work

- Improve reverse engineering process
- Extend PARADIGM language with UI test patterns for web, mobile and for security testing
- Generate test data automatically
- Improve the usability of the tool
- Extend the environment to support traceability with requirements

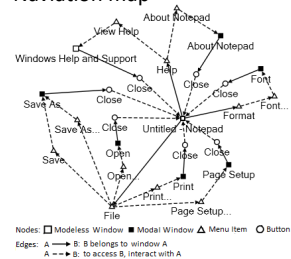
- Pattern Based GUI testing
- Reverse engineering
- Serious games
- Extract knowledge from user interactions

## Extract models from dynamic exploration

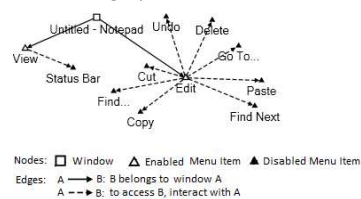
### Windows



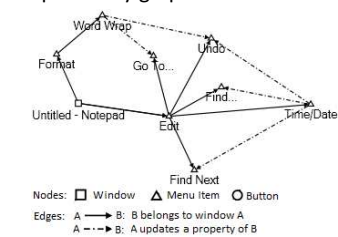
### Navigation map



### Disabled graph



### Dependency graph



- **Inês Coimbra Morgado , Ana C. R. Paiva , João Pascoal Faria**  
[Dynamic Reverse Engineering of Graphical User Interfaces](#)  
International Journal on Advances in Software, Vol.5 n° 3&4, pp.223-235, 2012
- **Miguel Nabuco , Ana Cristina Ramada Paiva , João Carlos Pascoal Faria**  
[Inferring User Interface Patterns from Execution Traces of Web Applications](#)  
in 14th International Conference Computational Science and Its Applications - ICCSA 2014, pp.311-326, 2014
- **Clara Sacramento , Ana C. R. Paiva**  
[Web Application Model Generation through Reverse Engineering and UI Pattern Inferring](#)  
in 9th International Conference on the Quality of Information and Communications Technology (QUATIC 2014), pp.-, 2014
- **Miguel Nabuco , Ana C. R. Paiva , Rui Camacho , João P. Faria**  
[Inferring UI Patterns with Inductive Logic Programming](#)  
in 8th Iberian Conference on Information Systems and Technologies, pp.-, 2013
- **Inês Coimbra , Ana C. R. Paiva , João P. Faria , Rui Camacho**  
[GUI Reverse Engineering With Machine Learning](#)  
in RAISE'12 Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, pp.-, 2012

## Research lines

- Pattern Based GUI testing
- Reverse engineering
- Serious games
- Extract knowledge from user interactions

## Serious games

3.2. Review Process - Exercise Points: 0

Find the activities of a formal review that are characterized by the points presented. If you need you can ask for a new letter (hint button), but remind that you will gain less points for that word.

1

Recording updated status of defects  
Fixing defects found

□ □ □ □ □

Go Back Hint

## Research lines

- Pattern Based GUI testing
- Reverse engineering
- Serious games
- Extract knowledge from user interactions

## Extract knowledge from user interactions

- Examples: Google analytics
- Problems: Difficult to analyse the results
- Solution: build a framework that allows to extract knowledge from that information

