Experimenting a Reverse Engineering Technique for Modelling the Behaviour of Rich Internet Applications

Domenico Amalfitano
Anna Rita Fasolino
Porfirio Tramontana

Dipartimento di Informatica e Sistemistica
Universita' degli Studi di Napoli, Federico II
Naples, Italy

{domenico.amalfitano, fasolino, ptramont}@unina.it

Motivation

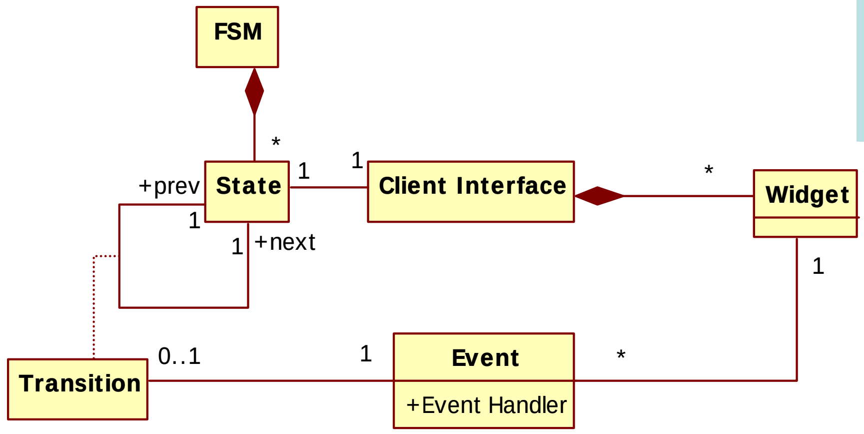
- Modern **Rich Internet Applications** (RIAs) usually offer complex GUIs to their users

- **Reverse Engineering** of models describing the behaviour of existing RIA GUIs can be necessary in different contexts:
  - Testing
  - Comprehension, Maintenance and Reengineering
  - Automatic crawling of RIAs

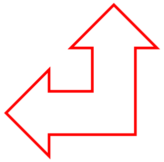
- We proposed and experimented with:
  a tool-supported technique, based on dynamic analysis, for reverse engineering a **Finite State Machine** (FSM) model of the RIA GUI behaviour.

FSM Model and the Reverse Engineering Technique

1. FSM is used to represent an RIA behaviour.
2. FSM represents all the elaboration states where the RIA receives any input solicitation by its user.
3. Each state of the RIA is described by the client Interface shown to the user at that interaction time.
4. Each *client interface* is characterized only by the sub-set of its *active widgets* that are ‘clickable’ or, more in general, that have a *registered event listener* and a corresponding *event handler*.
5. *Transitions* are associated with user interactions that trigger the RIA migration towards the new state.



The UML class diagram showing the information characterizing the proposed FSM model of an RIA



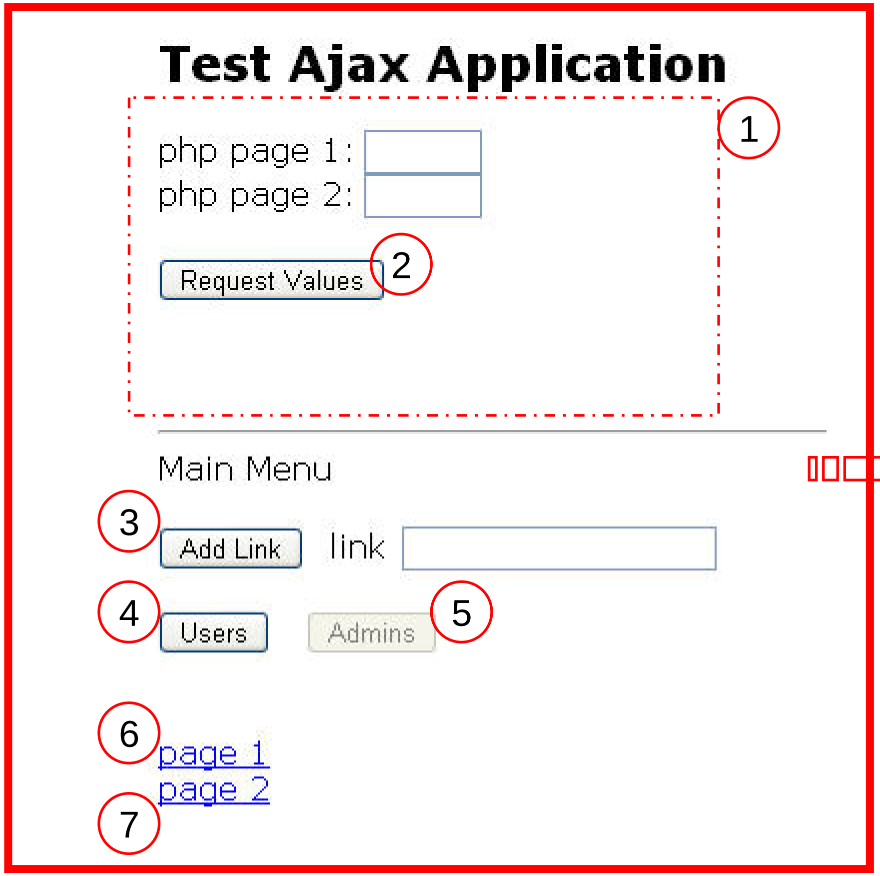
The Reverse Engineering Technique

The proposed Reverse Engineering Technique for obtaining a FSM-based model of the RIA behaviour includes four sequential steps:

1. **Execution trace collection**
   - A set of traces exercising the RIA behaviours is collected;
   - Each trace is made up of Client interface DOMs and user events fired on these DOMs.
2. **Trace analysis and classification**
   - Three interface classification techniques are used to cluster into equivalent states the collected interfaces.
   - User events between consecutive interfaces are associated to transitions between states.
3. **FSM model validation**
   - The abstracted FSM model is compared against a Gold Standard (GS) one expected by an expert.

Client Interface Collection 1/2

During the trace collection, each Client Interface is characterized by the following subsets of widgets :
- All the widgets with an event handler.
- All the widgets on which the user can fire an event handler



An Example of Client Interface and the selected widgets:

1 – Form
2 – Button
3 – Button
4 – Button
5 – Button
6 – Link
7 – Link

Client Interface Collection 2/2

Each widget of the Client Interface is characterized by some of its attributes, its Xpath and unindexed Xpath.

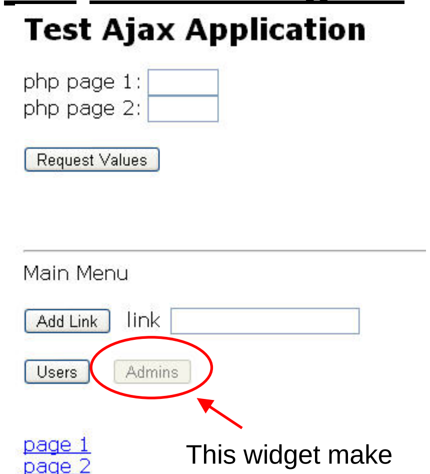
	Id_widget	Attribute	Value	Xpath	Unindexed xpath	Active
1	5482	action	#	/html[2]/body[1]/form[1]	/html/body/form	true
2	5475	type	button	/html[2]/body[1]/form[1]/input[3]	/html/body/form/input	true
	5475	onclick	"avvia()"	/html[2]/body[1]/form[1]/input[3]	/html/body/form/input	true
3	5476	type	button	/html[2]/body[1]/div[1]/div[1]/input[1]	/html/body/div/div/input	true
	5476	onclick	"new_link()"	/html[2]/body[1]/div[1]/div[1]/input[1]	/html/body/div/div/input	true
4	5477	type	button	/html[2]/body[1]/div[1]/div[2]/input[1]	/html/body/div/div/input	true
	5477	onclick	"gest ('users.xml')"	/html[2]/body[1]/div[1]/div[2]/input[1]	/html/body/div/div/input	true
5	5478	type	button	/html[2]/body[1]/div[1]/div[2]/input[2]	/html/body/div/div/input	false
	5478	onclick	"gest (‘admins.xml')"	/html[2]/body[1]/div[1]/div[2]/input[2]	/html/body/div/div/input	false
6	5479	href	page1.html	/html[2]/body[1]/div[2]/a[1]	/html/body/div/a	true
7	5480	href	page2.html	/html[2]/body[1]/div[2]/a[1]	/html/body/div/a	true

Table reporting captured attributes of the Example client interface

Interface Classification Techniques

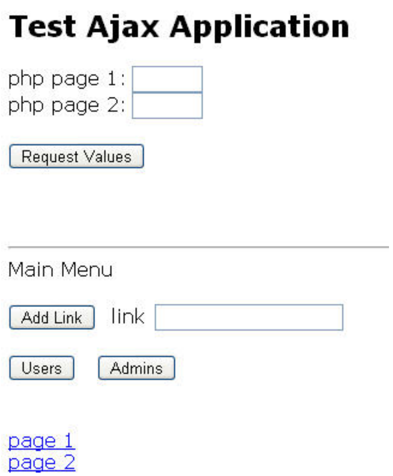
Three interface classification criteria are used to assess the equivalence of two interfaces:

- **C1)** two interfaces are equivalent if they have the same number of widgets with the same subset of attributes.
- **C2)** two interfaces are equivalent if they have the same number of ***enabled*** widgets with the same subset of attributes.
- **C3)** two interfaces are equivalent if they contains ***the same set of lists, containing the same widgets***.



This widget make the difference for C1

Interface I_1



Interface I_2

This widget make the difference for C2

Interface I_3

- Interface I_1 is equivalent to interface I_2 according to C1, while they are not equivalent according to C2

- Interface I_2 is equivalent to interface I_3 according to C3, while they are not equivalent according to C2

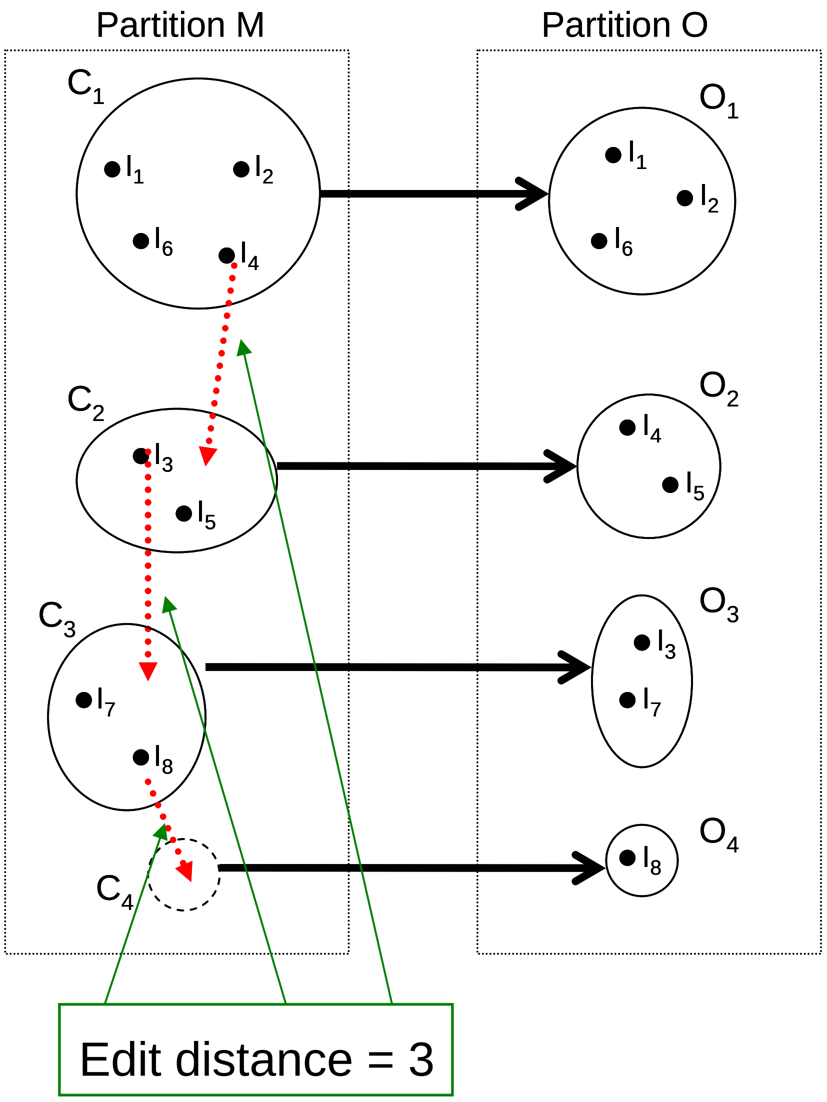
FSM Validation Technique

- The classification of the collected interfaces provides a partition of the set of interfaces
- In order to assess the effectiveness of the clustering criteria, the ***edit distance*** between the produced partition and the one proposed by an expert is automatically evaluated by the CReRIA tool
- The effectiveness of the process of abstracting the correct FSM model is measured by the *edit distance*, using the following metric:

Correct Interface Ratio (CIR) metric:

CIR (M) = 1- d(M, O)/ |I|

where **CIR= 100%** indicates that M and O partitions are identical.



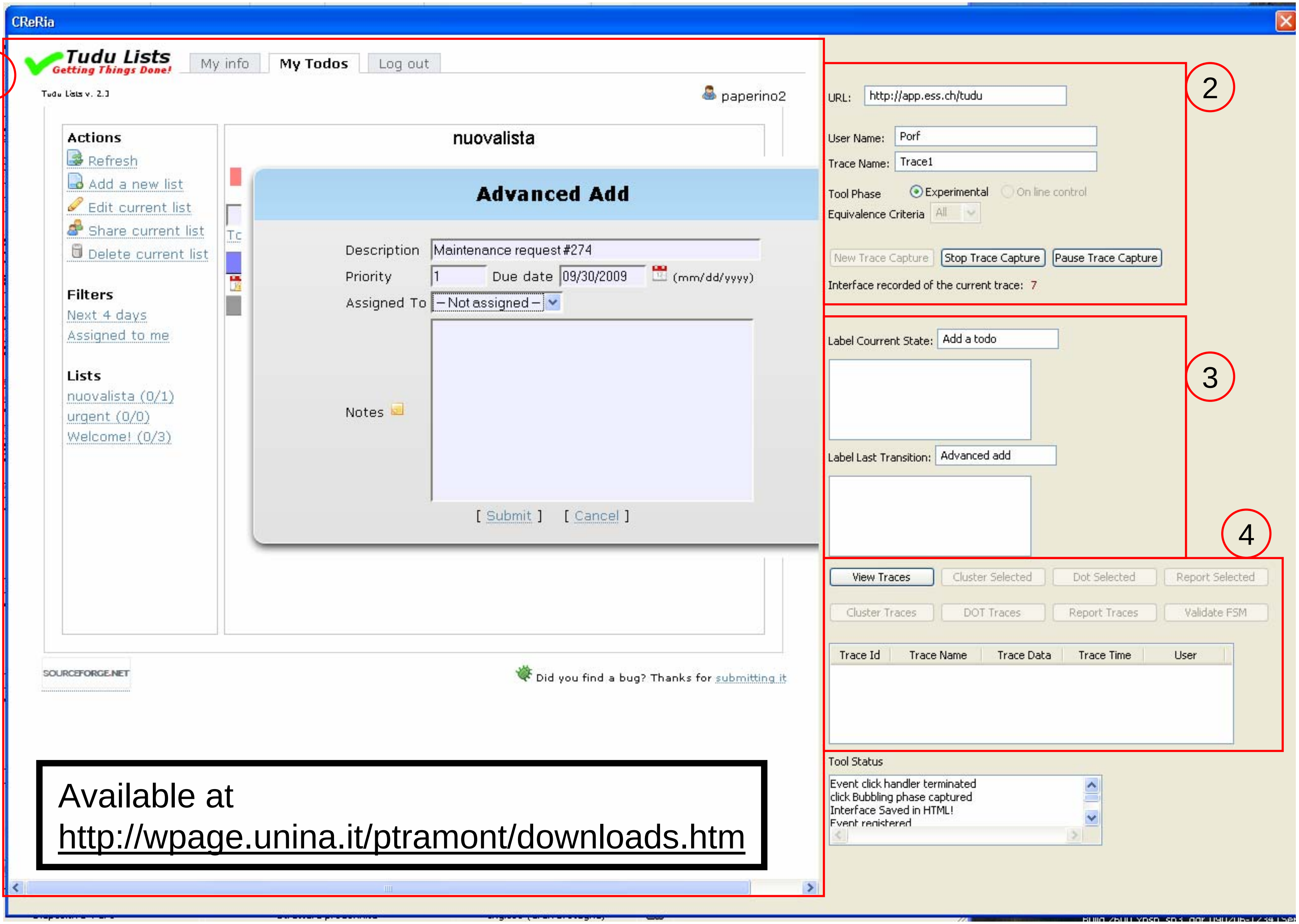
The CReRia tool

- **CReRia** is a Java-based integrated environment, supporting the execution of the proposed reverse engineering technique.

- Its user interface provides:

1. A Browser Emulator – a Mozilla browser for navigating the subject RIA
2. A Panel for Starting and executing the Session of trace collection
3. A Panel for building the GS during the RIA navigation session
4. A Panel for validating the Reverse Engineering results.

The CReRia tool GUI



Available at
http://wpage.unina.it/ptramont/downloads.htm

The CReRia Tool is developed in Java & MySql

Experiment

- **Goals of the experiment**:
  (1) assessing the technique effectiveness
  (2) comparing the effectiveness of different interface equivalence criteria

- **Experimental Materials : (four involved RIAs)**
  W1: http://app.ess.ch/tudu/welcome.action
  W2: http://www.pikipimp.com
  W3: http://www.agavegroup.com/agWork/theList/theListWrapper.php
  W4: http://www.buttonator.com

- **Experimental Procedure**:
  - For each RIA, 2 authors acted as experts and produced the Gold Standard model
  - The reverse engineering process was executed by 5 undergraduate students:
    - Several traces were collected for each RIA
    - The tool classified the trace equivalent interfaces and produced the FSMs
    - The tool compared each obtained FSM against the Gold Standard model using the CIR metric

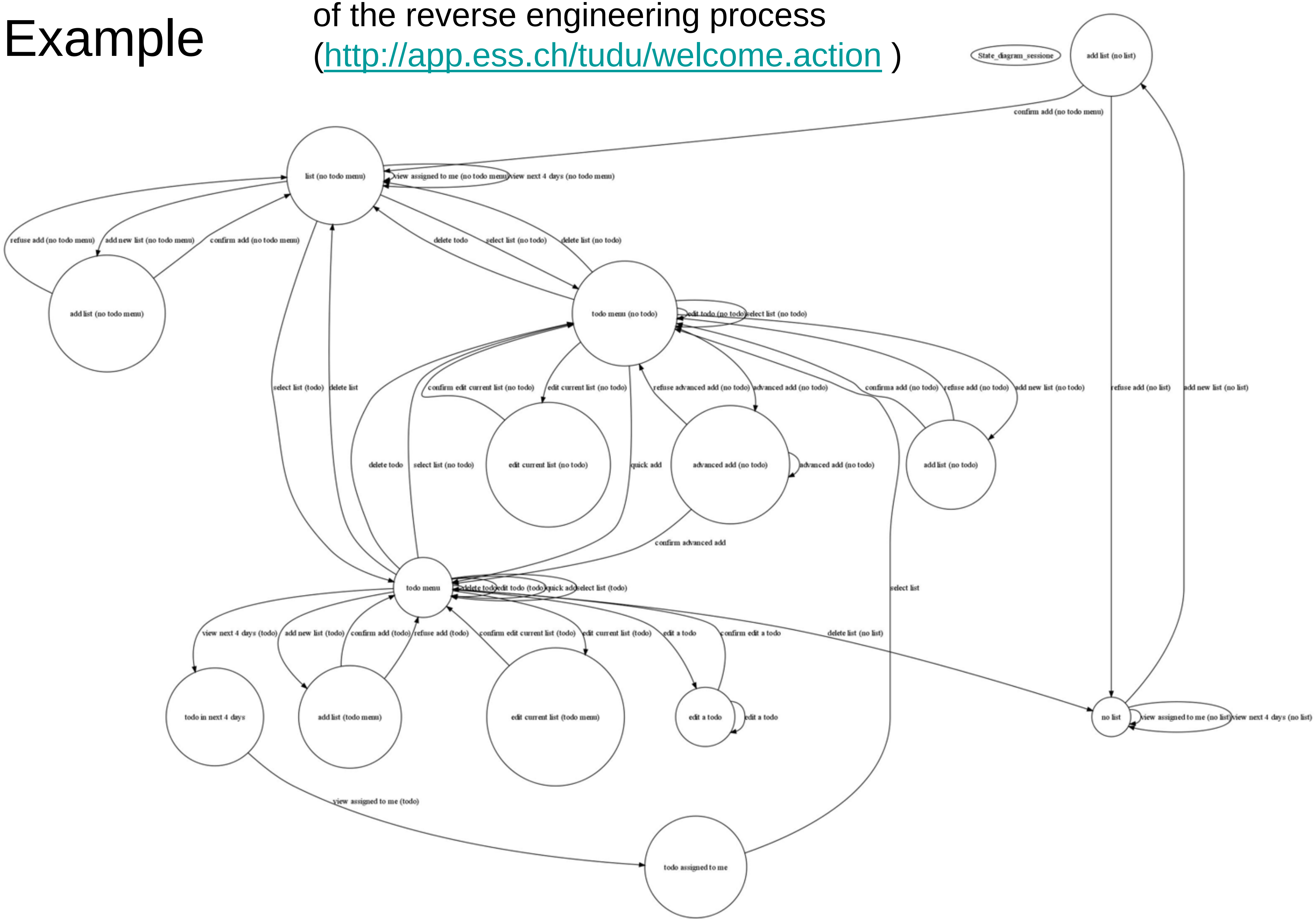
Experimental Results

Subject Application	Use Cases	Scenarios	Gold Standard States	Gold Standard Transitions	Collected User Session Traces	Collected Interfaces	Best Criterion	Edit Distance	Correct Interface Ratio
W1	8	17	15	52	30	1885	C3	14	85%
W2	1	2	4	16	8	533	C3	8	65%
W3	3	10	4	9	11	731	C3	0	100%
W4	1	8	19	54	11	829	C2	0	100%
							C3	23	62%

- In the first two cases (W1 and W2) the FSM and GS models where quite similar (cfr. 85% and 65% values of CIR)
- In the other two cases (W3 and W4) the FSM and GS models where identical (CIR=100%), but they were obtained using different equivalence criteria (C3 and C2)

A FSM Example

The FSM of (W1) Tudu after the execution of the reverse engineering process (http://app.ess.ch/tudu/welcome.action)



Conclusions

- The proposed Reverse Engineering technique showed its effectiveness in reconstructing a FSM model of RIAs which is very similar to the one produced by an expert of the RIA.
- The technique exploits structural/ behavioural interface equivalence criteria which do not require the choice of any similarity threshold
- Experiments showed that the most effective criterion depends on the characteristics of analysed RIA client interfaces.

- Future improvements:
  - An interactive process that puts together the interface collection and validation phases of the process
    - The proposed criteria are used to generate suggestions about the equivalence between FSM states
  - Automatic generation of Test Suites for covering the states of the FSM model