



GUI Reverse Engineering with Machine Learning

Inês Coimbra Morgado, inescm@fe.up.pt

Ana C. R. Paiva, apaiva@fe.up.pt

João Pascoal Faria, jpf@fe.up.pt

Rui Camacho, rcamacho@fe.up.pt

RAISE'12, Zurich, 5th June 2012

Software Reverse Engineering

"the process of analysing a subject system to identify the system's components and interrelationships and to create representations of the system in another form or at a higher level of abstraction"

Chikofsky and Cross, 1990

Software Reverse Engineering

Exploration of the system

*“the process of analysing a subject system to **identify the system’s components** and interrelationships and to create representations of the system in another form or at a higher level of abstraction”*

Chikofsky and Cross, 1990

Software Reverse Engineering

Representation of the information

*"the process of analysing a subject system to identify the system's components and interrelationships and to **create representations** of the system in another form or at a higher level of abstraction"*

Chikofsky and Cross, 1990

Motivation & Goal

- Hard to manually build a model

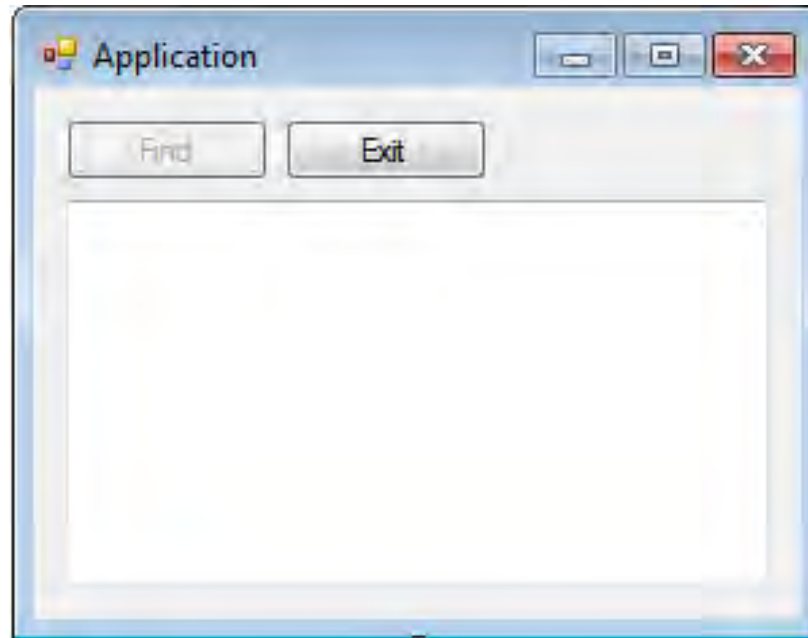


- Extract (part of) the model of a GUI automatically and dynamically (in run time)

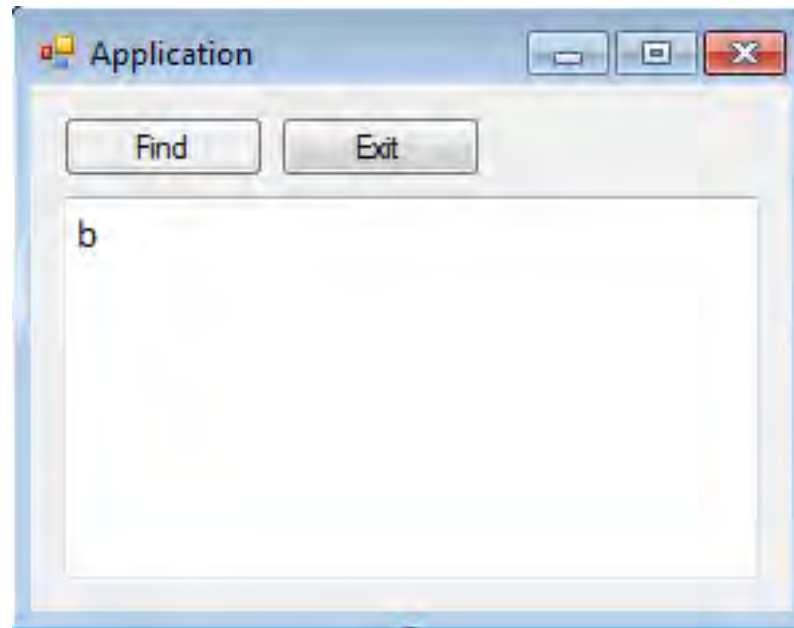


- Reduce the effort of building a formal model

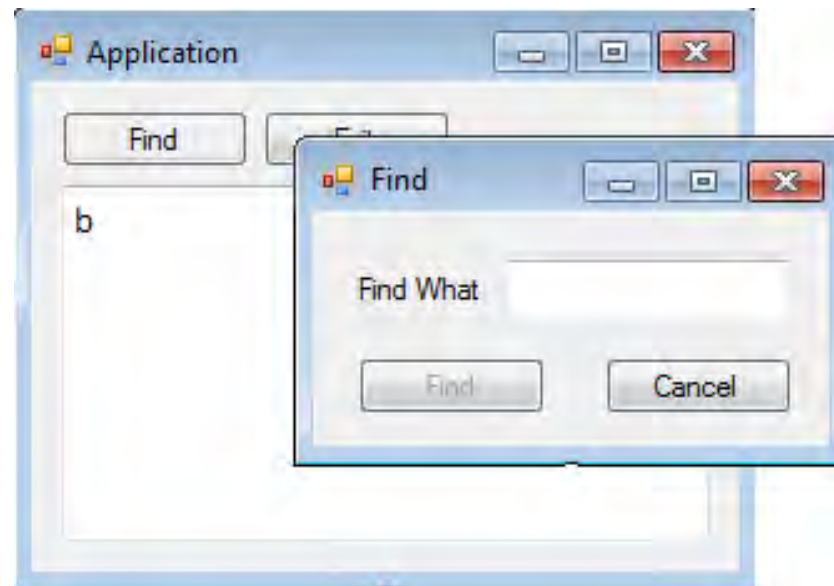
Application Under Analysis



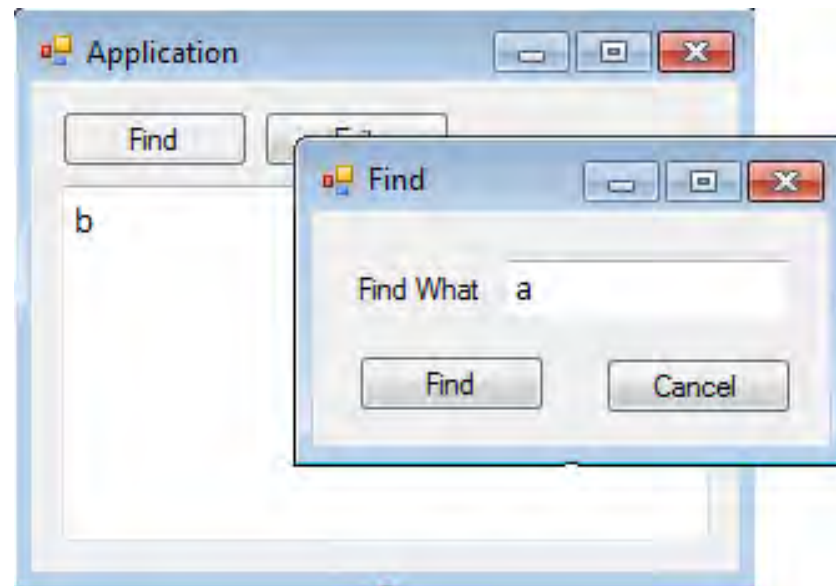
Application Under Analysis



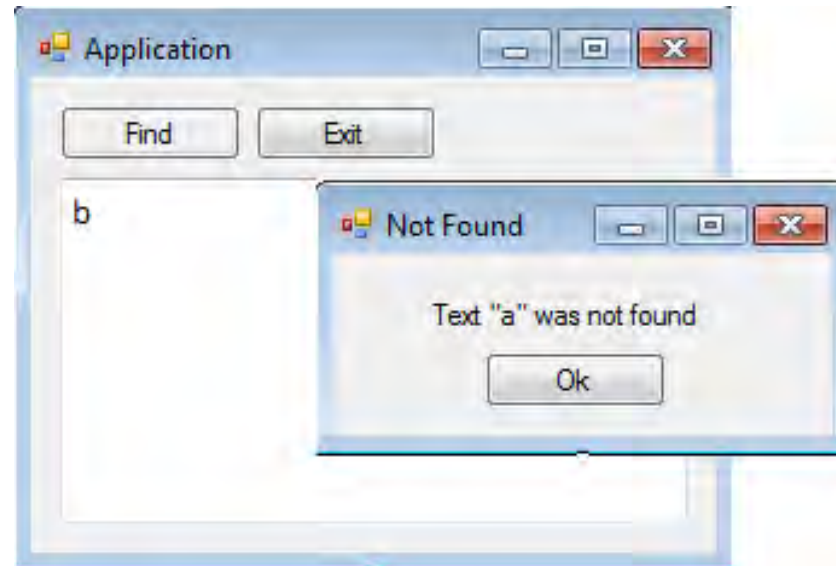
Application Under Analysis



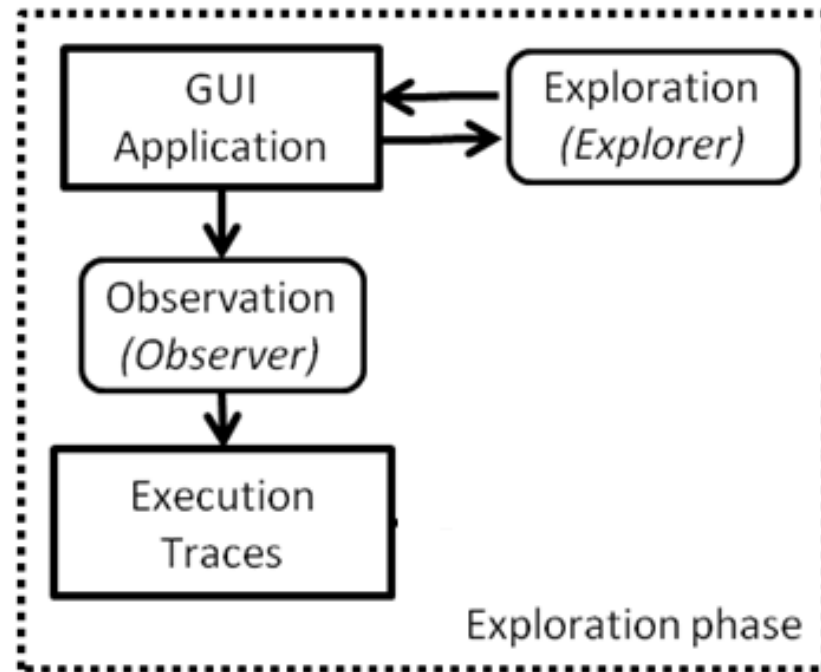
Application Under Analysis



Application Under Analysis



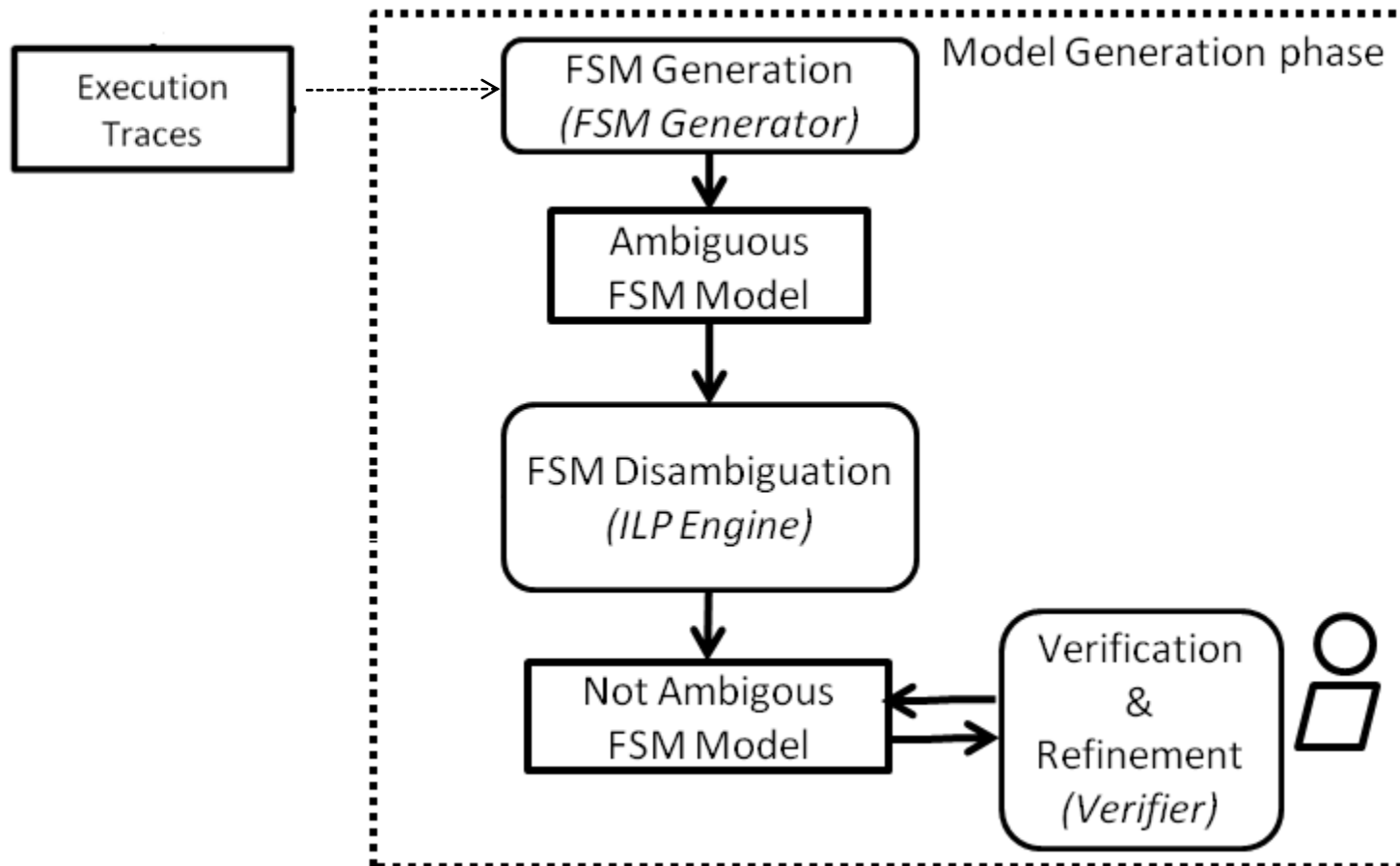
Exploration Process



Execution Traces

| TraceId/StepId | Event (User Action) | Next GUI State | | | | | | | | |
|----------------|-----------------------|----------------|--------------|------|------|----------|------------------|------|--------|----|
| | | W1 | | | | W2 | | | | W3 |
| | | Text | EnterText(X) | Find | Exit | FindWhat | EnterFindWhat(Y) | Find | Cancel | Ok |
| 0.0 | Start | [] | E | D | E | - | - | - | - | - |
| 0.1 | W1.Exit | - | - | - | - | - | - | - | - | - |
| 1.0 | Start | [] | E | D | E | - | - | - | - | - |
| 1.1 | W1.EnterText("a") | [a] | E | E | E | - | - | - | - | - |
| 1.2 | W1.Find | [a] | D | D | D | [] | E | D | E | - |
| 1.3 | W2.EnterFindWhat("a") | [a] | D | D | D | [a] | E | E | E | - |
| 1.4 | W2.Find | [a] | E | E | E | - | - | - | - | - |
| 1.5 | W1.Exit | - | - | - | - | - | - | - | - | - |
| 2.0 | Start | [] | E | D | E | - | - | - | - | - |
| 2.1 | W1.EnterText("b") | [b] | E | E | E | - | - | - | - | - |
| 2.2 | W1.Find | [b] | D | D | D | [] | E | D | E | - |
| 2.3 | W2.EnterFindWhat("a") | [b] | D | D | D | [a] | E | E | E | - |
| 2.4 | W2.Find | [b] | D | D | D | [a] | D | D | D | E |
| 2.5 | W3.Ok | [b] | E | E | E | - | - | - | - | - |
| 2.6 | W1.Exit | - | - | - | - | - | - | - | - | - |

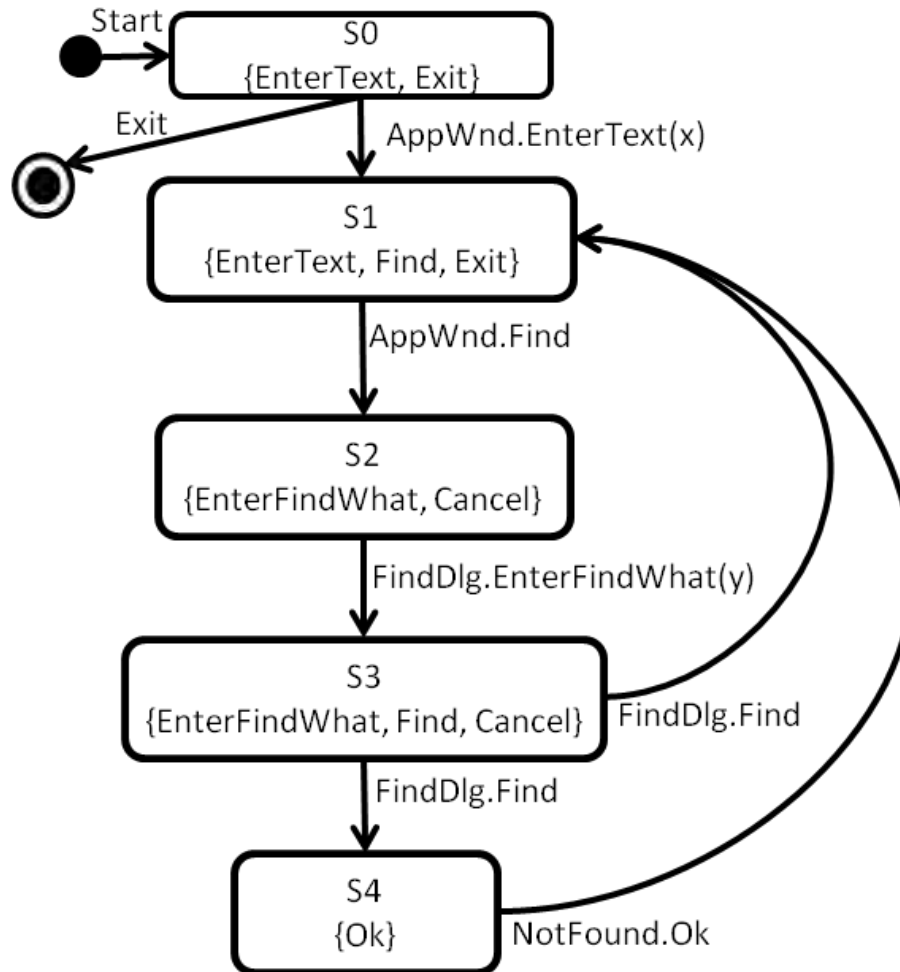
Model Generation Process



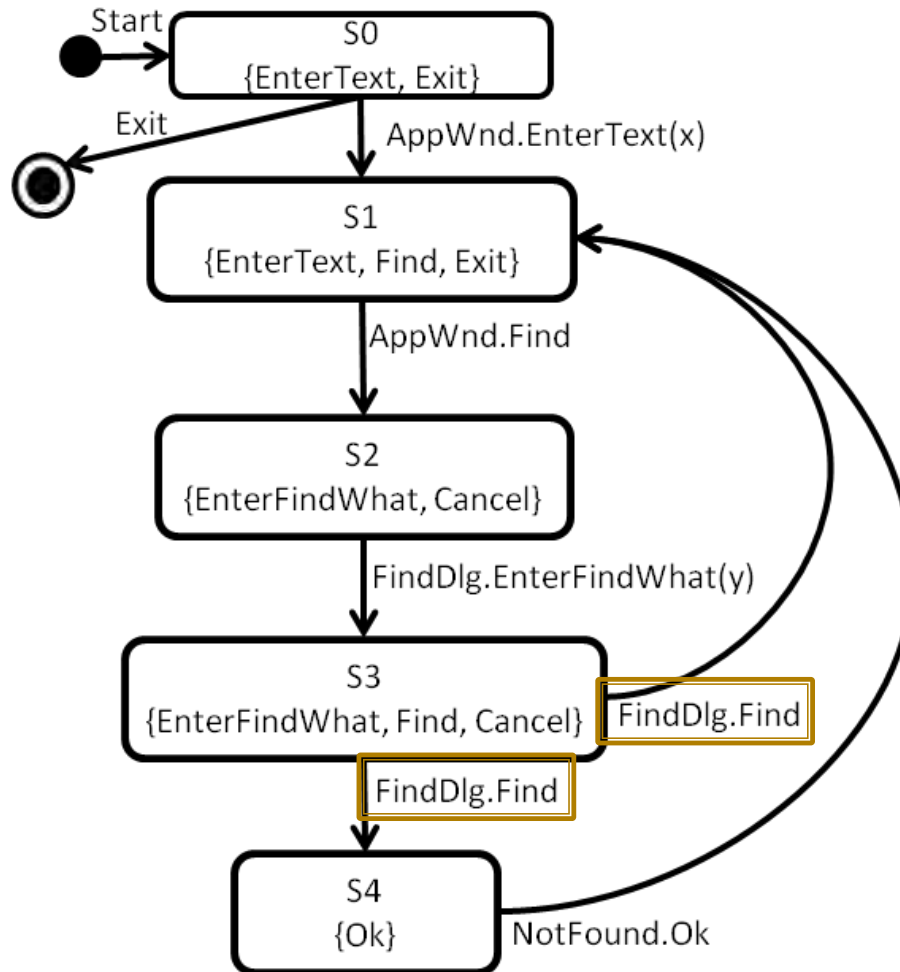
Execution Traces & FSM States

| TraceId,StepId | Event (User Action) | Next GUI State | | | | | | | | | Next FSM State |
|----------------|-----------------------|----------------|--------------|------|------|----------|------------------|------|--------|----|----------------|
| | | W1 | | | | W2 | | | | W3 | |
| | | Text | EnterText(X) | Find | Exit | FindWhat | EnterFindWhat(Y) | Find | Cancel | Ok | |
| 0.0 | Start | [] | E | D | E | - | - | - | - | - | S0 |
| 0.1 | W1.Exit | - | - | - | - | - | - | - | - | - | End |
| 1.0 | Start | [] | E | D | E | - | - | - | - | - | S0 |
| 1.1 | W1.EnterText("a") | [a] | E | E | E | - | - | - | - | - | S1 |
| 1.2 | W1.Find | [a] | D | D | D | [] | E | D | E | - | S2 |
| 1.3 | W2.EnterFindWhat("a") | [a] | D | D | D | [a] | E | E | E | - | S3 |
| 1.4 | W2.Find | [a] | E | E | E | - | - | - | - | - | S1 |
| 1.5 | W1.Exit | - | - | - | - | - | - | - | - | - | End |
| 2.0 | Start | [] | E | D | E | - | - | - | - | - | S0 |
| 2.1 | W1.EnterText("b") | [b] | E | E | E | - | - | - | - | - | S1 |
| 2.2 | W1.Find | [b] | D | D | D | [] | E | D | E | - | S2 |
| 2.3 | W2.EnterFindWhat("a") | [b] | D | D | D | [a] | E | E | E | - | S3 |
| 2.4 | W2.Find | [b] | D | D | D | [a] | D | D | D | E | S4 |
| 2.5 | W3.Ok | [b] | E | E | E | - | - | - | - | - | S1 |
| 2.6 | W1.Exit | - | - | - | - | - | - | - | - | - | End |

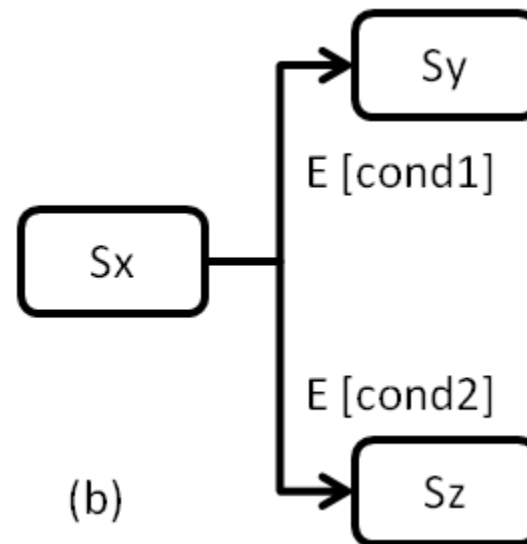
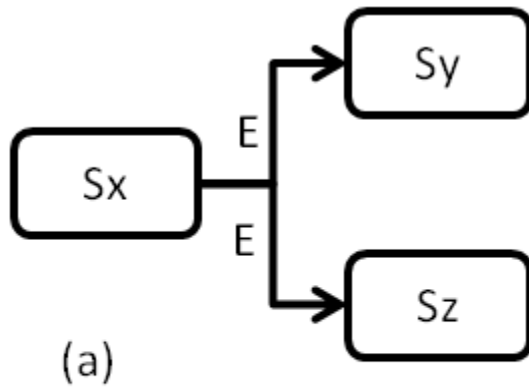
Finite State Machine



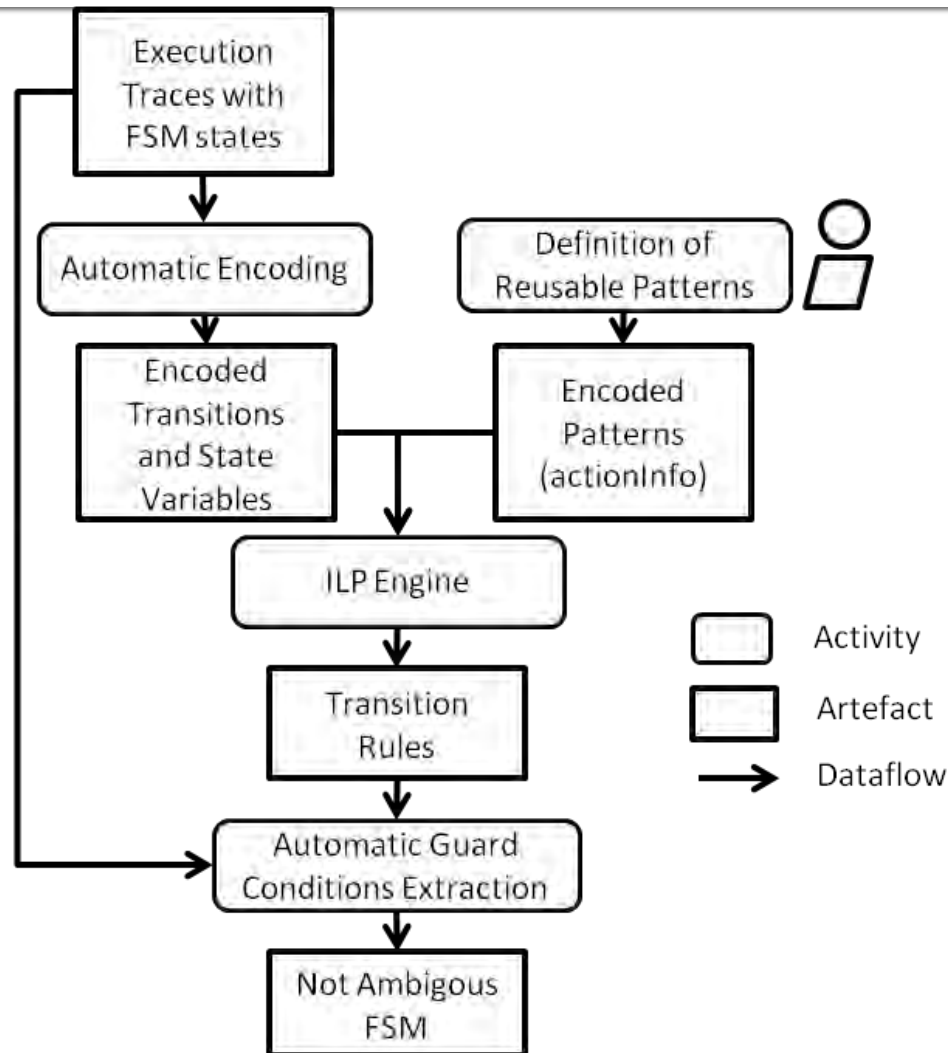
Finite State Machine



Ambiguity



Inductive Logic Programming Process



Encoding of states and transitions

stateVariable(Control, Traceld, StepId, Value).
stateVariable(text, trace1, step3, [b]).

transition (Source, Action, Target, Traceld, StepId).
transition(so, enterText, s1, trace1, step1).

Encoding of patterns

- Manual encoding of reusable patterns (once)
 - Login, RangeValidation, Mandatory Field, Find...

`actionInfo(Action, Traceld, StepId, Result).`

*actionInfo(find, Traceld, StepId, notFound):-
stateVariable(text, Traceld, StepId, Text),
stateVariable(findWhat, Traceld, StepId, FindWhat),
not member(FindWhat, Text).*

Inferring transition rules

- Returns the disambiguated transitions
*transition(Source, Action, Target, Traceld, StepId):-
stateName(Source, s3),
stateName(Target, s4),
actionInfo(Action, Traceld, StepId, notFound).*

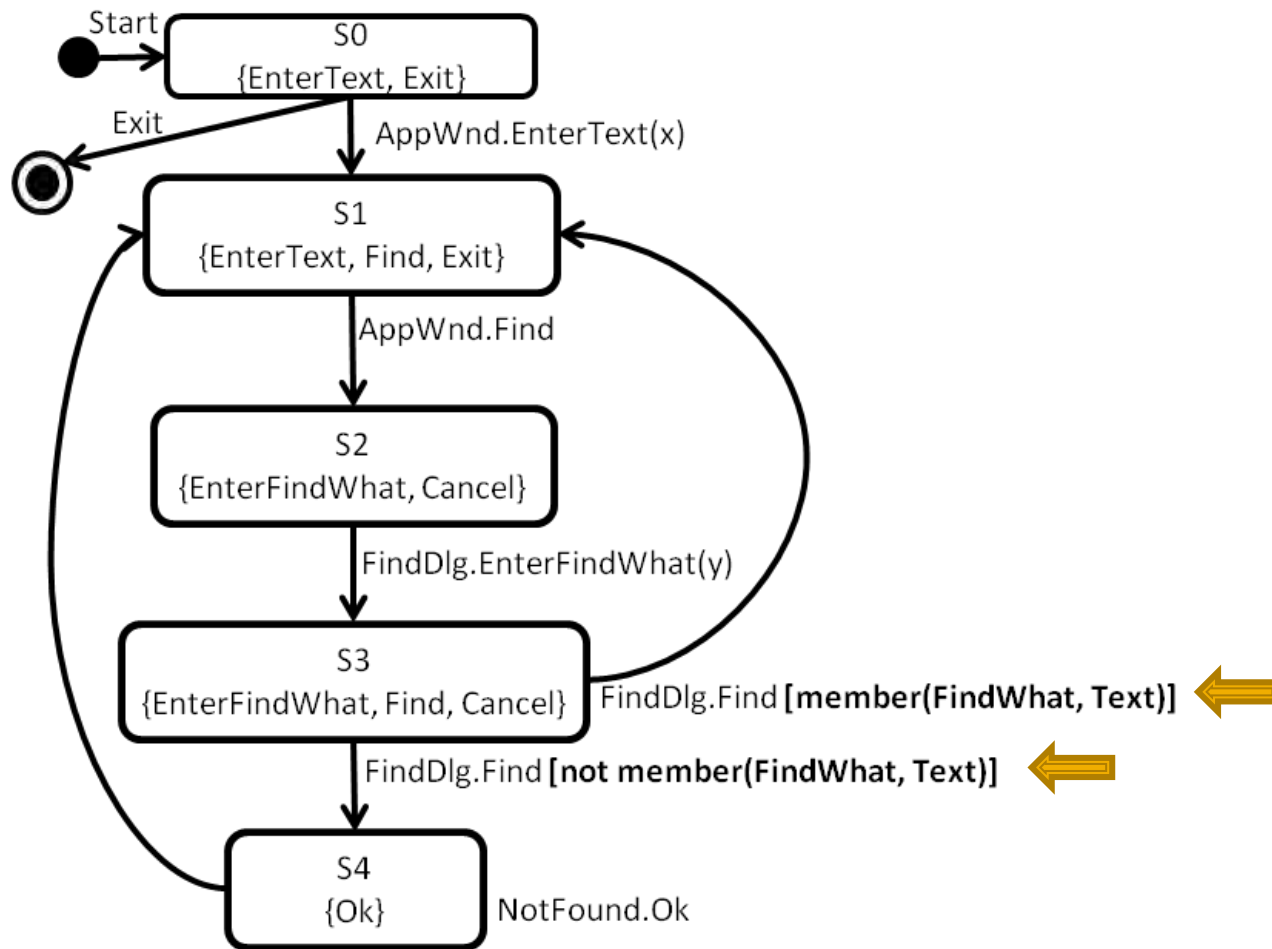
Guard Conditions

- Extract the guard conditions:

cond1 = member(FindWhat, Text).

cond2 = not member(FindWhat, Text).

Not Ambiguous FSM



Conclusions

- Approach to extract model
- Approach to solve ambiguities
- Combines machine learning with software engineering

Future Work

- Explore the encoding of more powerful patterns
- Improve the automatic reuse of patterns
- Transform in iterative process
 - Complement the model at each iteration
 - Use the extracted information to guide the exploration
 - Extract more information for ILP
 - Provide a more complete and intelligent exploration



Universidade do Porto
Faculdade de Engenharia
FEUP

GUI Reverse Engineering with Machine Learning

Thank You!

Inês Coimbra Morgado, inescm@fe.up.pt

Ana C. R. Paiva, apaiva@fe.up.pt

João Pascoal Faria, jpf@fe.up.pt

Rui Camacho, rcamacho@fe.up.pt

RAISE'12, Zurich, 5th June 2012