Model-based testing: What’s in it for industry?

Christian Colombo
What is a model?
What is a model?

Drops some details

e.g. actual/relative distances/sizes
What is a model?

Drops some details

\textit{e.g.} actual/relative distances/sizes

...but focuses on others

\textit{e.g.} order of planets, relative positions, etc
What is a model?

- Drops some details e.g. actual/relative distances/sizes
- ...but focuses on others e.g. order of planets, relative positions, etc

Has to be easier to create/understand/deal with than the original system!
What to model in S/W?

- Expected order of actions
- Return values
- Timing
Example
Example

- Empty Cart
  - add
  - remove
- Non-Empty Cart
  - add
  - remove
- Checkout
  - remove
  - checkout
Light switch example

up

On/off

down
How is the model useful?
Test automation

- Testcase generation
- Testcase execution
- Testcase pass/fail

This is usually automated
Test automation

- Testcase generation
- Testcase execution
- Testcase pass/fail

This is automated but assertion is hand-made
Test automation

- Testcase generation
- Testcase execution
- Testcase pass/fail

This is hand-made
Test automation

- Testcase generation
- Testcase execution
- Testcase pass/fail

MBT can automate them all!
Test automation

- Testcase generation
- Testcase execution
- Testcase pass/fail

MBT can automate them all! (once you have the model)
Model-Based Testing

System

= 

Model
Model-Based Testing

System

Model

system action

model action
Model-Based Testing
Example

Empty Cart
- add\(\text{cnt}++\)\(\check\)\text{cnt}
- remove (\text{cnt}==1)\(\text{cnt}--\)\(\check\)\text{cnt}

Non-Empty Cart
- add \(\text{cnt}++\)\(\check\)\text{cnt}
- remove (\text{cnt}>1)\(\text{cnt}--\)\(\check\)\text{cnt}

Check out
- remove \(\text{cnt}--\)\(\check\)\text{cnt}
- remove (\text{cnt}==1)\(\text{cnt}--\)\(\check\)\text{cnt}
- checkout
Model-Based Testing

0 items 🔄 System 🔄 Add book 🔄 System after action 🔄 1 item 🔄

Model 🔄 Inc count 🔄 Model after action

count==0 🔄 count==1
UI example

Model calls content Selenium drives content Website
UI example

Model \rightarrow calls \rightarrow Selenium \rightarrow drives \rightarrow Website

content \rightarrow content \rightarrow content
UI example

```
public void init() {
    driver = new FirefoxDriver();
    driver.get(shoppingCartURL);
    searchPage = new Search(driver);
}

public void addItem() {
    searchPage.searchAndAddToCart(null);
}
```
How do you code the model?

```java
public void init() {
    driver = new FirefoxDriver();
    driver.get(shoppingCartURL);
    search = new Search(driver);
    addButton = new AddButton(driver);
    addItem() {
        searchPage.searchAndAddToCart(null);
    }
```
Tools for MBT

- ModelJUnit (free)
- MaTeLo (commercial)
- Spec Explorer (comes with Visual Studio)

http://mit.bme.hu/~micskeiz/pages/modelbased_testing.html
//States
public enum WebsiteState {
    EMPTY_CART, NON_EMPTY_CART, CHECKOUT, ERROR_STATE
}

// State variables
private int cartSize = 0;
private boolean checkedOut = false;

// Define the states the FSA may be in
@Override
public WebsiteState getState() {
    if (checkedOut)
        return WebsiteState.CHECKOUT;

    if (cartSize==0)
        return WebsiteState.EMPTY_CART;

    else if (cartSize>0)
        return WebsiteState.NON_EMPTY_CART;

    return WebsiteState.ERROR_STATE;
}
ModelJUnit - States

```java
//States
public enum WebsiteState {
    EMPTY_CART,
    NON_EMPTY_CART,
    CHECKOUT,
    ERROR_STATE
}

// State variables
private int cartSize = 0;
private boolean checkedOut = false;

// Define the states the FSA may be in
@Override
public WebsiteState getState() {
    if (checkedOut)
        return WebsiteState.CHECKOUT;

    if (cartSize==0)
        return WebsiteState.EMPTY_CART;

    else if (cartSize>0)
        return WebsiteState.NON_EMPTY_CART;

    return WebsiteState.ERROR_STATE;
}
```

Variables to States
Reminder

Empty Cart

Non-Empty Cart

Check out

add\text{cnt}++\check\text{cnt}

remove (cnt>1)\text{cnt}--\check\text{cnt}

remove \text{cnt}--\check\text{cnt}

remove (cnt==1)\text{cnt}--\check\text{cnt}

add\text{cnt}++\check\text{cnt}
ModelJUnit - Transitions

```java
public boolean addItemGuard() {
    return !getState().equals(WebsiteState.CHECKOUT)
            && !getState().equals(WebsiteState.ERROR_STATE);
}

public @Action void addItem() {

    //update SUT
    sut.addItem();

    //update model
    cartSize++;

    //check correspondence
    assertEquals(cartSize, sut.getNumItems());
}
```
Demo

- With help from Mark Micallef
More advanced stuff

- Exploring the model in different ways
  - Random
  - Greedy
  - Lookahead
More advanced stuff

- Exploring the model in different ways
  - Random
  - Greedy
  - Lookahead

More intelligent ways to cover the model
More advanced stuff

- Exploring the model in different ways
  - Random
  - Greedy
  - Lookahead

- Biasing choice of transitions with probabilities
  - Take more likely transitions more frequently
More advanced stuff

- Exploring the model in different ways
  - Random
  - Greedy
  - Lookahead

- Biasing choice of transitions with probabilities
  - Take more likely transitions more frequently

For example reducing probability of taking Checkout transition
More advanced stuff

- Exploring the model in different ways
  - Random
  - Greedy
  - Lookahead

- Biasing choice of transitions with probabilities
  - Take more likely transitions more frequently

Any ideas of how you would modify the model?
More advanced stuff

- Exploring the model in different ways
  - Random
  - Greedy
  - Lookahead

- Biasing choice of transitions with probabilities
  - Take more likely transitions more frequently

- Timing
  - Timeouts
  - Manipulate timing
More advanced stuff

- Exploring the model in different ways
  - Random
  - Greedy
  - Lookahead

- Biasing choice of transitions with probabilities
  - Take more likely transitions more frequently

- Timing
  - Timeouts
  - Manipulate timing

E.g.: Check website responsiveness
More advanced stuff

- Exploring the model in different ways
  - Random
  - Greedy
  - Lookahead

- Biasing choice of transitions with probabilities
  - Take more likely transitions more frequently

- Timing
  - Timeouts
  - Manipulate timing

E.g.: Vary delay between adding books to cart
Summarising
Benefits of MBT

- Automatic generation of test cases
- Automatic verification of tests
Challenges of MBT

- Writing the model
- Some learning curve
Maintaining the correspondence

System  Model

A change in one requires a change in the other
The End

- Questions